Open access • Proceedings Article • DOI:10.1109/ICC.1996.533621

A multidimensional phase-locked loop for blind equalization of multi-input multi-output channels — Source link

John R. Barry, A. Batra

Institutions: Georgia Institute of Technology

Published on: 23 Jun 1996 - International Conference on Communications

Topics: Blind equalization, Phase-locked loop, MIMO, Multidimensional systems and Carrier frequency offset

A Multidimensional Phase-Locked Loop for Blind Multiuser Detection

John R. Barry, *Member, IEEE,* and Anuj Batra, *Member, IEEE*

***Abstract*—This paper concerns the problem of blind multiuser detection, a special case of the blind source separation problem in which the source signals have finite alphabets. Specifically, we address the problem of identifying and resolving the $n \times n$ unitary matrix ambiguity $\mathbf{U}$ that results from whitening the receiver observations, where n is the number of sources. We propose the multidimensional phase-locked loop (MPLL) as a generalization of a scalar decision-directed PLL to vector-valued signals. The MPLL adapts an estimate of $\mathbf{U}$ according to the recursion $\hat{\mathbf{U}}_{k+1} = \hat{\mathbf{U}}_k \mathbf{R}_k$, where $\mathbf{R}_k$ is an n-dimensional Householder-like rotation depending on only the kth receiver observation. The $O(n^2)$ complexity of an efficient implementation of the algorithm is extremely low. Nevertheless, simulation results demonstrate good convergence properties and superior steady-state performance when compared with prior techniques. The algorithm is also able to accommodate large alphabets and shaped alphabets.**

***Index Terms*—Adaptive unitary filtering, finite alphabets, source separation.**

I. Introduction

EARLY strategies for managing interference in wireless communication networks were based on a philosophy of interference avoidance, whereby multiple users competing for the same medium were allocated orthogonal channels in space, time, or frequency. In contrast, modern networks utilizing code- or space-division multiple access are (by design) subject to a significant amount of multiuser interference. *Multiuser detection* is the process of mitigating this interference through signal processing at the receiver with the aim of recovering the information transmitted by each user.

A partially blind multiuser detector exploits partial knowledge of the matrix channel that maps transmitted symbols to receiver observations. For example, the generalized sidelobe canceler [1], MUSIC [2], and ESPRIT [3] algorithms exploit knowledge of the array geometry, and the code-division-multiple-access detectors of [4]–[6] exploit knowledge of the signature sequence of the desired user. In contrast to the partially blind problem, this paper concerns the fully blind problem in which the receiver has no *a priori* knowledge of the channel matrix. Instead, the receiver can exploit only its knowledge of the statistics of the channel inputs. The blind multiuser detection problem is a special case of the blind source separation problem [7] with the restriction that the sources are finite-alphabet digital communications signals having the same signaling rate (and often the same alphabet).

A common and effective strategy for implementing a blind multiuser detector is to decompose the process into two steps by first whitening and then rotating [8]–[21]. The whitening step requires only second-order statistics and is well suited for blind implementation. The effect of whitening is to transform the unknown channel matrix into an unknown *unitary* matrix of dimension $n \times n$, where n is the number of users. The focus of this paper is on the second step of rotation, which blindly identifies and resolves this unknown unitary matrix.

Among the previously proposed blind source separation algorithms, many do not constrain the alphabets to be discrete. For example, Cardoso and Souloumiac proposed the joint approximate diagonalization of eigenmatrices (JADE) algorithm [10], which is an effective separator that places few restrictions on the source signals. (See also [20]). It estimates the columns of $\mathbf{U}$ from the eigenvectors of an $n^2 \times n^2$ sample cumulant matrix. The main drawbacks of JADE are its batch-oriented nature, making it ill-suited for time-varying channels, and its high $O(n^5)$ computational complexity [22]. Another example is the equivariant adaptive source separation (EASI) algorithm [23] of Cardoso and Laheld. It is an adaptive algorithm whose $O(n^3)$ complexity is significantly lower, albeit with slower convergence. Lower still in complexity is the $O(n^{2.5})$ adaptive algorithm of Zarzoso and Nandi [24], but its convergence speed is comparable to that of EASI, and it is untested in noise.

Other separation algorithms require knowledge of a specific property of the alphabet at the receiver. For example, the clustering algorithms of [19] and [25] exploit the finite nature of the symbol alphabets, but they are batch-oriented and restricted to binary alphabets, and their exponential $O(2^n)$ complexity is often prohibitive. A related approach with $O(n^4)$ complexity is proposed by van der Veen [26]. Constant-modulus algorithms [21], [27], [28] have low complexity and can be effective for finite alphabets but suffer from slow convergence and can fail with shaped alphabets.

This paper proposes a *decision-directed* rotation strategy that is substantially different from prior techniques. Our approach is motivated by the single-user problem of blind carrier recovery in which the receiver must blindly resolve an unknown unitary scalar of the form $e^{j\theta}$. In that setting, the phase-locked loop (PLL) is almost universally adopted. By analogy, we propose an extension of the PLL to vector-valued signals. The resulting multidimensional PLL (MPLL) is an *ad hoc* algorithm in the

Manuscript received March 29, 2001; revised May 29, 2002. This work was supported in part by the Center for Research in Applied Signal Processing (CRASP) and by National Science Foundation under Grant CCR-0082329. The associate editor coordinating the review of this paper and approving it for publication was Dr. Abdelhak Zoubir.

J. R. Barry is with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, Georgia, 30332-0250 USA (e-mail: barry@ece.gatech.edu).

A. Batra is with Texas Instruments, Inc., Dallas, TX 75243 USA (e-mail: batra@ti.com).

Publisher Item Identifier 10.1109/TSP.2002.801934.

sense that it is not derived according to any criterion of optimality. Although the scalar PLL has been tied to Kalman filtering [29], we make no such claims for the MPLL. Nevertheless, we will see that many of the same features of the scalar PLL that make it a workhorse in the scalar setting carry over into the higher dimensional problem of adapting unitary matrices. Specifically, the MPLL algorithm has the following attributes.

- It is adaptive.
- Its $O(n^2)$ complexity is significantly lower than prior techniques.
- It has good convergence properties.
- Its decision-directed nature leads to excellent steady-state performance.
- It can accommodate higher order alphabets and shaped alphabets.

The MPLL algorithm is also conceptually simple, with a useful geometric interpretation that will facilitate its application to new areas. Finally, the MPLL is parameterized by only a single parameter, which helps contribute to its robustness.

The remainder of this paper is organized as follows. In Section II, we present our system model and problem statement. In Section III, we review the scalar PLL and then present its multidimensional extension. In Section IV, we present numerical results, illustrating the effectiveness of the MPLL algorithm.

II. Channel Model and Problem Statement

We consider a memoryless unitary channel in which the receiver observation $\mathbf{y}_k = \left[y_k{}^{(1)}, \ldots y_k{}^{(n)}\right]^T$ at time k is given by

$$\mathbf{y}_k = \mathbf{U}\mathbf{x}_k + \mathbf{n}_k \tag{1}$$

where $\mathbf{x}_k = \left[x_k{}^{(1)}, \ldots x_k{}^{(n)}\right]^T$ represents a vector of transmitted symbols, $\mathbf{U}$ is an unknown $n \times n$ unitary matrix, and where the noise $\mathbf{n}_k$ is a complex circularly symmetric Gaussian noise sequence with diagonal autocorrelation matrix $E[\mathbf{n}_k\mathbf{n}_k^*] = \mathbf{\Sigma}^2$ and $E[\mathbf{n}_{k+m}\mathbf{n}_k^*] = \mathbf{0}$ for $m \neq 0$, where $(\cdot)^*$ denotes complex conjugate transpose. We assume that the n symbol sequences are mutually independent, with symbols in each sequence being chosen independently from n possibly distinct (although often identical) finite alphabets. The alphabets are assumed to be known *a priori* at the receiver and are normalized to have unit energy so that $E[\mathbf{x}_k\mathbf{x}_k^*] = \mathbf{I}$. The problem addressed in this paper is straightforward: Given the unitary model (1), blindly and adaptively identify and resolve $\mathbf{U}$. We remark that while not strictly necessary, it is advantageous for the receiver to exploit the fact that the channel is unitary because this reduces the number of degrees of freedom and thus may lead to better performance, faster convergence, or lower complexity.

The unitary model of (1) may arise in a wide variety of applications, including blind multiuser detection, blind beamforming, array-to-array (MIMO) communications, and multicarrier modulation. For example, consider the problem of blind multiuser detection for an n-user system in which the kth observation at an array of m sensors is

$$\mathbf{r}_k = \mathbf{H}\mathbf{x}_k + \mathbf{w}_k \tag{2}$$

where $\mathbf{x}_k$ is as before, where $\mathbf{H}$ is an unknown (nonunitary) complex-valued $m \times n$ channel matrix with $m > n$ and full column rank, and where $\mathbf{w}_k$ is a complex circularly symmetric white-Gaussian noise sequence satisfying $E[\mathbf{w}_k\mathbf{w}_k^*] = \sigma^2\mathbf{I}$. If $\mathbf{H} = \mathbf{Q}_{m\times n}\mathbf{S}_{n\times n}\mathbf{U}_{n\times n}$ is a truncated SVD for which $\mathbf{Q}^*\mathbf{Q} = \mathbf{U}^*\mathbf{U} = \mathbf{I}$ and $\mathbf{S}$ is diagonal, then $\mathbf{Q}$, $\mathbf{S}$, and σ may be recovered blindly using second-order statistics only, namely, from an eigendecomposition of $E[\mathbf{r}_k\mathbf{r}_k^*] = \mathbf{Q}\mathbf{S}^2\mathbf{Q}^* + \sigma^2\mathbf{I}$. It follows that $\mathbf{W} = \mathbf{S}^{-1}\mathbf{Q}^*$ is an $n \times m$ whitening matrix and that $\mathbf{y}_k = \mathbf{W}\mathbf{r}_k$ accomplishes the whitening step. Then, the unitary model of (1) describes the whitened data $\mathbf{y}_k = \mathbf{W}\mathbf{r}_k$, where the noise autocorrelation matrix is $\mathbf{\Sigma}^2 = \sigma^2\mathbf{S}^{-2}$.

For the convolutive case of channels with memory, a combination of spatial and temporal whitening will also lead to the unitary model (1) [13], [17], [30]. Alternatively, the whitening step need not be based on second-order statistics; for example, a whitener based on higher order statistics and a vector constant-modulus algorithm has been proposed [21], [28].

III. Phase-Locked Loop

We describe the multidimensional PLL (MPLL), a low-complexity technique for adapting a unitary matrix. Since the MPLL can be viewed as a generalization of a scalar PLL to vector-valued signals, we begin by reviewing a first-order scalar PLL for carrier phase synchronization.

A. Scalar PLL

A single-user quadrature-amplitude modulation (QAM) communication system with carrier-phase offset is a special case of (1) in which all signals are scalars. In this case, (1) reduces to

$$y_k = e^{j\theta}x_k + n_k$$

where θ is the unknown phase, x_k is the transmitted symbol chosen from a finite alphabet, and n_k is the noise. The vast majority of practical receivers mitigate such phase offset using a PLL. In Fig. 1(a), we depict the traditional first-order discrete-time PLL.

The structure of the PLL is easily motivated by first neglecting noise. Let $\hat{\theta}_k$ denote the receiver's estimate of θ at time k. In an attempt to cancel the phase offset, the receiver multiplies y_k by $e^{-j\hat{\theta}_k}$, producing $z_k = e^{-j\hat{\theta}_k}y_k$. In the absence of noise, this reduces to $z_k = e^{j\varepsilon_k}x_k$, where $\varepsilon_k = \theta - \hat{\theta}_k$ denotes the *estimation error*. With knowledge of x_k (i.e., with training), the receiver may then recover ε_k by measuring the angle between x_k and z_k. If this angle is known exactly at time k, the receiver can force the estimation error to zero at time $k+1$ by choosing the next phase estimate to be $\hat{\theta}_{k+1} = \hat{\theta}_k + \varepsilon_k$.

The impact of noise is to prevent the receiver from knowing ε_k exactly. Let $\hat{\varepsilon}_k$ denote the receiver's estimate of ε_k at time k, as estimated by measuring the angle between x_k and z_k using the *phase detector*

$$\hat{\varepsilon}_k = \sin^{-1}\left(\mathrm{Im}\left\{\frac{x_k^* z_k}{|x_k z_k|}\right\}\right).$$

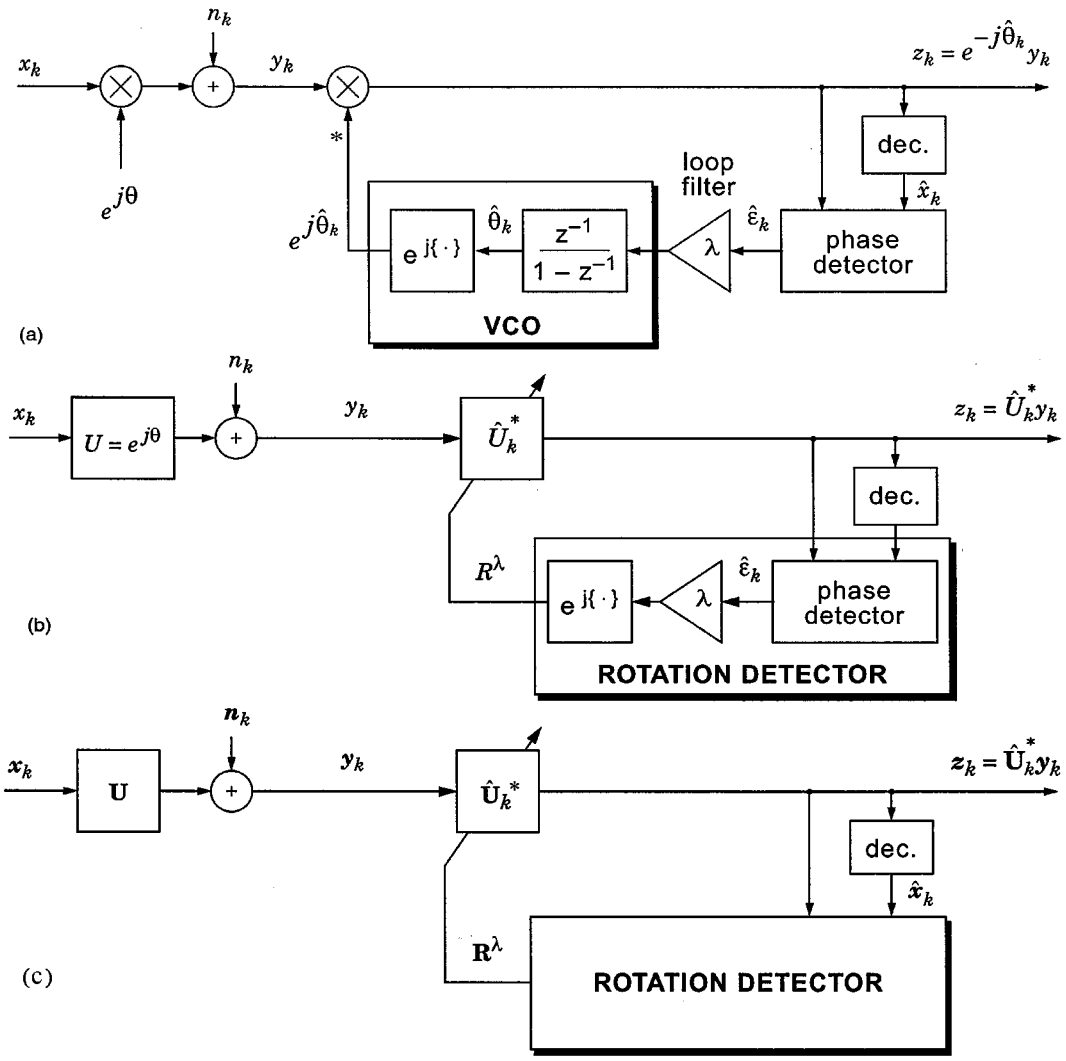

Fig. 1. (a) Phase-locked loop for scalar channels. (b) Its equivalent model. (c) MPLL for matrix channels.

Because of noise, this estimate differs from ε_k. As illustrated in Fig. 1(a), a first-order PLL updates the phase estimate according to [31]:

$$\hat{\theta}_{k+1} = \hat{\theta}_k + \lambda \hat{\varepsilon}_k \tag{3}$$

where the parameter $\lambda \in (0, 1]$ is referred to as the loop-filter gain or step size. Choosing $\lambda = 1$ is appropriate only in the absence of noise. A smaller gain will mitigate noise at the expense of slower convergence and poorer tracking (for the case when θ varies with time). A proper choice of λ trades off convergence speed and tracking capability for noise immunity.

The block diagram of Fig. 1(a) shows the traditional view of the PLL described by (3), consisting of a phase detector with output $\hat{\varepsilon}_k$, a loop filter with output $\lambda\hat{\varepsilon}_k$, an accumulator with output $\hat{\theta}_k = \lambda \sum_{i=0}^{k-1} \hat{\varepsilon}_i$, and a complex exponentiator with output $e^{j\hat{\theta}_k} = \exp\left(j\lambda \sum_{i=0}^{k-1} \hat{\varepsilon}_i\right)$. The cascade of the accumulator and complex exponentiator is commonly referred to as a VCO, in deference to its analog origins. In the absence of training, the PLL can be operated in a *decision-directed* mode by substituting decisions $\hat{x}_k$ in place of x_k in the phase detector, where $\hat{x}_k$ is the alphabet symbol closest to z_k. The PLL shown in Fig. 1(a) is decision directed.

We now present an alternative model of the scalar PLL that will facilitate its generalization to vector-valued signals. The alternative model is shown in Fig. 1(b) and is precisely equivalent to that of Fig. 1(a). The basic difference here is that processing is performed on $U = e^{j\theta}$ directly instead of indirectly via θ. In particular, let $\hat{U}_k = e^{j\hat{\theta}_k}$ denote the receiver's estimate of $U = e^{j\theta}$ at time k. Rather than viewing the VCO as an accumulator followed by a complex exponentiator, we opt instead to view it as a complex exponentiator followed by a product accumulator so that the VCO output can be rewritten as $\hat{U}_k = \prod_{i=0}^{k-1} \exp(j\lambda\hat{\varepsilon}_i)$. By definition of the phase detector, we may interpret the term $\exp(j\hat{\varepsilon}_i)$ as the unique unitary scalar that rotates $\hat{x}_i/|\hat{x}_i|$ to $z_i/|z_i|$. To emphasize this interpretation, we introduce the new notation

$$R\{\hat{x}_i \to z_i\} \triangleq \exp(j\hat{\varepsilon}_i).$$

The VCO output may now be expressed as $\hat{U}_k = \prod_{i=0}^{k-1} R^\lambda\{\hat{x}_i \to z_i\}$, where $R^\lambda\{\hat{x}_i \to z_i\}$ may be interpreted as the unique unitary scalar that rotates a fraction λ of the way from $\hat{x}_i/|\hat{x}_i|$ to $z_i/|z_i|$.

As shown in Fig. 1(b), we define the *rotation detector* as the cascade of the phase detector, loop filter, and exponentiator. The rotation detector takes two inputs ($\hat{x}_k$ and z_k) and produces the output $R^\lambda\{\hat{x}_k \to z_k\}$. Simply stated, the rotation detector produces the unique unitary scalar that rotates a fraction λ of the way from one of its normalized inputs to the other. The VCO output $\hat{U}_k = \prod_{i=0}^{k-1} R^\lambda\{\hat{x}_i \to z_i\}$ can then be generated recursively using

$$\hat{U}_{k+1} = \hat{U}_k R^\lambda\{\hat{x}_k \to z_k\}. \tag{4}$$

This recursion is a compact and exact description of a first-order decision-directed PLL.

That the PLL always converges is not immediately evident, given that it is both blind and decision directed, and initial decisions will almost surely be incorrect. (A decision-directed LMS-based linear equalizer is known to be a poor blind equalizer, for example.) Indeed, with a 16-QAM alphabet and in the absence of noise, Simon and Smith [32] have shown that the output of the phase detector is identically zero when the phase error is $\varepsilon_k = 22.5°$ or $E[\hat{\varepsilon}_k|\varepsilon_k = 22.5°] = 0$. For sufficiently small step size λ, therefore, ε_k may linger near $\varepsilon_k = 22.5°$ for a long time. In theory [32], however, such misconvergence will not last indefinitely; even for zero noise and infinitesimal λ, an appropriate sequence of symbols will eventually occur that allows the PLL to escape this local minimum, and ε_k is guaranteed to converge eventually to a multiple of 90°. Fortunately, in practice, there is always noise, and λ is always significantly larger than zero, both of which prevent the PLL from getting trapped indefinitely in an undesirable stable point. For this reason, the existence of local minima rarely causes a problem in practice.

B. Multidimensional PLL

In this section, we generalize the scalar decision-directed phase-locked loop of the Section II to vector-valued signals. Assume the model (1), and let $\hat{\mathbf{U}}_k$ denote the receiver's estimate of the unitary matrix $\mathbf{U}$ at time k. Let $\mathbf{z}_k = \hat{\mathbf{U}}_k^* \mathbf{y}_k$ denote the MPLL output, and let $\hat{\mathbf{x}}_k$ denote the receiver's estimate of $\mathbf{x}_k$, as found through either quantizing $\mathbf{z}_k$ or via training. (In the former case, the ith component of $\hat{\mathbf{x}}_k$ is taken as the element of the ith alphabet closest to the ith component of $\mathbf{z}_k$.) Generalizing (4) to higher dimensions leads to the following recursion for updating the receiver's estimate of $\mathbf{U}$:

$$\hat{\mathbf{U}}_{k+1} = \hat{\mathbf{U}}_k \mathbf{R}^\lambda\{\hat{\mathbf{x}}_k \to \mathbf{z}_k\} \tag{5}$$

where $\mathbf{R}^\lambda$ is the rotation detector output at time k, as shown in Fig. 1(c). The key question is how to define this rotation detector. Extending the scalar definition, we will require that it partially rotate $\hat{\mathbf{x}}_k/\|\hat{\mathbf{x}}_k\|$ to $\mathbf{z}_k/\|\mathbf{z}_k\|$. Unlike the scalar case, however, such an $\mathbf{R}^\lambda$ is not unique. We define a suitable $\mathbf{R}^\lambda$ in the following.

Because the two inputs to $\mathbf{R}\{\hat{\mathbf{x}}_k \to \mathbf{z}_k\}$ lie within the subspace spanned by $\hat{\mathbf{x}}_k$ and $\mathbf{z}_k$, the rotation detector at time k can learn nothing about vectors orthogonal to this subspace, and hence, it is intuitively reasonable to limit the action of the rotation detector to this subspace. This observation suggests that $\mathbf{R}\{\hat{\mathbf{x}}_k \to \mathbf{z}_k\}\mathbf{w} = \mathbf{w}$ for any $\mathbf{w} \perp \text{Span}\{\hat{\mathbf{x}}_k, \mathbf{z}_k\}$. We will thus require that the unitary $\mathbf{R}$ satisfy the following two conditions:

i) $\mathbf{R}\{\hat{\mathbf{x}}_k \to \mathbf{z}_k\}\hat{\mathbf{x}}_k/\|\hat{\mathbf{x}}_k\| = \mathbf{z}_k/\|\mathbf{z}_k\|$.
ii) $\mathbf{R}\{\hat{\mathbf{x}}_k \to \mathbf{z}_k\}\mathbf{w} = \mathbf{w}$ for any $\mathbf{w} \perp \text{Span}\{\hat{\mathbf{x}}_k, \mathbf{z}_k\}$.

Even with the second constraint, $\mathbf{R}\{\hat{\mathbf{x}}_k \to \mathbf{z}_k\}$ is not unique. We can make it unique by choosing the $\mathbf{R}$ satisfying conditions i) and ii) that minimizes the Frobenius distance between $\mathbf{R}$ and the identity matrix; this strategy is intuitively pleasing because we expect $\mathbf{R}$ to approximate the identity near convergence. As shown in Appendix A, the unitary matrix $\mathbf{R}$ closest to the identity and satisfying conditions i) and ii) is

$$\mathbf{R}\{\hat{\mathbf{x}}_k \to \mathbf{z}_k\} = \mathbf{I} + \begin{bmatrix} \mathbf{x} & | & \mathbf{v} \end{bmatrix} \times \begin{bmatrix} e^{j\alpha}c - 1 & -e^{j\alpha}s \\ s & c - 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \hline \mathbf{v}^* \end{bmatrix} \tag{6}$$

where $p = \mathbf{x}^*\mathbf{z}$ is the inner product between $\mathbf{x} = \hat{\mathbf{x}}_k/\|\hat{\mathbf{x}}_k\|$ and $\mathbf{z} = \mathbf{z}_k/\|\mathbf{z}_k\|$, where $c = |p| \le 1$, where $s = \sqrt{1 - |p|^2}$, where $e^{j\alpha} = p/|p|$, and where $\mathbf{v} = (\mathbf{z} - p\mathbf{x})/s$ is chosen so that $\{\mathbf{x}, \mathbf{v}\}$ forms an orthonormal basis for the span of $\mathbf{x}$ and $\mathbf{z}$. For the singular and rare case of $|p| = 1$, we set $\mathbf{v} = \mathbf{0}$ in (6). As expected, $\mathbf{R}\{\hat{\mathbf{x}}_k \to \mathbf{z}_k\}$ reduces to the Householder matrix $\mathbf{I} - 2\mathbf{x}\mathbf{x}^*$ when $\mathbf{z} = -\mathbf{x}$.

In principle, we may use (6) directly in (5). However, this would require that $\mathbf{R}$ be raised to a fractional power λ at each iteration. In practice, we can avoid this computationally intensive task by solving for $\mathbf{R}^\lambda$ directly as a function of $\hat{\mathbf{x}}_k$ and $\mathbf{z}_k$. As derived in Appendix A, with $\mathbf{R}$ given by (6), $\mathbf{R}^\lambda$ may be expressed as

$$\mathbf{R}^\lambda\{\hat{\mathbf{x}}_k \to \mathbf{z}_k\} = \mathbf{I} + \begin{bmatrix} \mathbf{x} & | & \mathbf{v} \end{bmatrix} \times \begin{bmatrix} d_2^\lambda + \gamma s - 1 & \gamma(d_1 - p) \\ \gamma(d_1 - p)^* & d_1^\lambda - \gamma s - 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \hline \mathbf{v}^* \end{bmatrix} \tag{7}$$

where p, s, and $\mathbf{v}$ are as defined in (6); where $d_1 = \varepsilon e^{j\Delta}$ and $d_2 = \varepsilon e^{-j\Delta}$ with $\varepsilon = \mu/|\mu|$, $\Delta = \cos^{-1}(|\mu|)$, and $\mu = (p + |p|)/2$; and where $\gamma = js\varepsilon^\lambda \sin(\lambda\Delta)/(\sin(\Delta)|d_1 - p|)$.

The expression for $\mathbf{R}^\lambda$ in (7), together with (5), defines the MPLL algorithm. It is parameterized by the step size $\lambda \in (0, 1]$. We emphasize that the MPLL is a true generalization of a scalar PLL and that (5) and (7) collapse to a conventional scalar decision-directed first-order PLL when $n = 1$.

We remark that because matrix multiplication is not commutative, we had to make a choice about the ordering of $\hat{\mathbf{U}}_k$ and $\mathbf{R}_k = \mathbf{R}^\lambda\{\hat{\mathbf{x}}_k \to \mathbf{z}_k\}$ in (5). Our choice to postmultiply $\hat{\mathbf{U}}_k$ by $\mathbf{R}_k$ is easily motivated on heuristic grounds. By definition of the rotation detector, $\mathbf{R}_k^*\mathbf{z}_k$ should be closer to $\mathbf{x}_k$ than was $\mathbf{z}_k$, or equivalently, $(\hat{\mathbf{U}}_k\mathbf{R}_k)^*\mathbf{y}_k$ should be closer to $\mathbf{x}_k$ than was $\hat{\mathbf{U}}_k^*\mathbf{y}_k$, where we used $\mathbf{z}_k = \hat{\mathbf{U}}_k^*\mathbf{y}_k$. However, from (1), we have $\mathbf{y}_k = \mathbf{U}\mathbf{x}_k + \mathbf{n}_k$; therefore, another equivalent statement is that $\hat{\mathbf{U}}_k\mathbf{R}_k$ should be closer to $\mathbf{U}$ than was $\hat{\mathbf{U}}_k$. It thus follows that the postmultiplication in (5) is the natural choice.

The multiplicative recursion of (5) can be implemented as an additive recursion with lower complexity since the difference $\mathbf{D}_k = \mathbf{R}^\lambda - \mathbf{I}$ is a rank-two matrix consisting of the sum of four outer products. Specifically, substituting (7) into (5) yields

$$\hat{\mathbf{U}}_{k+1} = \hat{\mathbf{U}}_k + \hat{\mathbf{U}}_k\mathbf{D}_k. \tag{8}$$

The matrix multiplication in (8) has only $O(n^2)$ complexity, which is significantly less than the $O(n^3)$ complexity of a direct implementation of the matrix multiplication in (5). Both re-

cursions preserve orthogonality exactly, without the need for an extra normalization step (like that used in [23]).

While an analytical study of the convergence of the MPLL has proven difficult (see [21]), the numerical results of Section IV demonstrate that, like the scalar PLL, the MPLL exhibits good convergence properties.

C. Reduced-Complexity Approximations to the Rotation Detector

The rotation detector of (7) is expressed as a function of the angle $\Delta = \cos^{-1}(|p+|p||/2)$ and the factor $\gamma = js\varepsilon^{\lambda}\sin(\lambda\Delta)/(\sin(\Delta)|d_1 - p|)$, neither of which possesses a straightforward geometric interpretation. The complexity required for computing Δ and γ may also be significant in certain applications. As alternatives, in this section, we present two approximations to the rotation detector (7); both approximations have reduced complexity and are somewhat easier to interpret than (7).

First, from (17) in Appendix A, we have that

$$\tilde{\mathbf{R}} = \begin{bmatrix} e^{j\alpha} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

the product of two matrices, where $\theta = \cos^{-1}(|p|)$. Raising each of the two matrices to the power λ *individually* leads to the approximation

$$\tilde{\mathbf{R}}^{\lambda} \approx \begin{bmatrix} e^{j\lambda\alpha} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\lambda\theta) & -\sin(\lambda\theta) \\ \sin(\lambda\theta) & \cos(\lambda\theta) \end{bmatrix}.$$

This approximation possesses a simple geometric interpretation as a partial plane rotation followed by a complex scalar rotation. It also requires fewer operations than (19). The approximate rotation detector that results is given by

$$\mathbf{R}\{\hat{\mathbf{x}}_k \to \mathbf{z}_k\} \approx \mathbf{I} + \left[\mathbf{x} \mid \mathbf{v}\right] \times \begin{bmatrix} e^{j\lambda\alpha}\cos(\lambda\theta) - 1 & -e^{j\lambda\alpha}\sin(\lambda\theta) \\ \sin(\lambda\theta) & \cos(\lambda\theta) - 1 \end{bmatrix} \left[\frac{\mathbf{x}^*}{\mathbf{v}^*}\right]. \quad (9)$$

Second, we can derive another approximation to the rotation detector (7) by introducing an intermediate vector $\mathbf{z}_\lambda = \lambda\mathbf{z}_k + (1-\lambda)\hat{\mathbf{x}}_k$ between $\mathbf{z}_k$ and $\hat{\mathbf{x}}_k$. Based on the observation that the true rotation detector $\mathbf{R}^{\lambda}\{\hat{\mathbf{x}}_k \to \mathbf{z}_k\}$ will approximately rotate $\hat{\mathbf{x}}_k/\|\hat{\mathbf{x}}_k\|$ to $\mathbf{z}_\lambda/\|\mathbf{z}_\lambda\|$, we may approximate (7) using

$$\mathbf{R}^{\lambda}\{\hat{\mathbf{x}}_k \to \mathbf{z}_k\} \approx \mathbf{R}\{\hat{\mathbf{x}}_k \to \mathbf{z}_\lambda\} \quad (10)$$

where $\mathbf{R}\{\hat{\mathbf{x}}_k \to \mathbf{z}_\lambda\}$ is computed from (6) with $\mathbf{z}_\lambda$ replacing $\mathbf{z}_k$. For the special case in which all alphabets are real, it is shown in [21] that (10) is equivalent to (7), albeit with a different value for λ.

Although (9) and (10) only approximate (7), they are equally valid extensions of the PLL to higher dimensions in the sense that they both reduce to a conventional scalar decision-directed first-order PLL when $n = 1$. The performance and complexity of the two approximations (9) and (10) will be compared with the true rotation detector (7) in the Section IV.

IV. Numerical Results

At first glance, it might seem that the goal of a blind multiuser detector is to force the channel-whitener-rotation cascade $\hat{\mathbf{U}}^*\mathbf{WH}$ to the identity matrix. However, there are two ambiguities in the blind setting that make this goal unattainable. First, the ordering of the users is arbitrary and unobservable, and second, rotating each user's alphabet by certain angles may also be unobservable. These two ambiguities are inherent to any blind source separation problem and are captured by a *complex permutation matrix* $\mathbf{P}$, which we define as any matrix expressible as the product of a permutation matrix and a diagonal unitary matrix. Thus, the actual goal of a blind multiuser detector is to force the channel-whitener-rotation cascade to *any* complex permutation matrix $\mathbf{P}$.

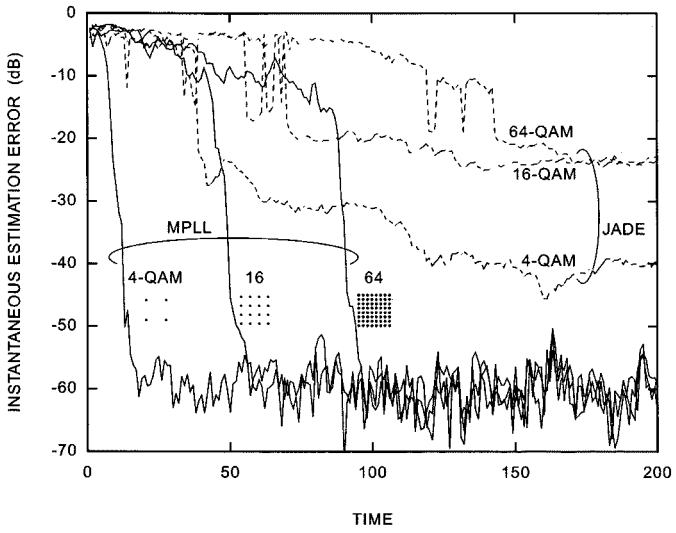

Fig. 2. Convergence slows with increasing alphabet size.

Consider a square 3×3 version of the multiuser channel (2), with each of the three symbols in $\mathbf{x}_k$ being chosen i.i.d. uniformly from a unit-energy QAM alphabet, and assume that an ideal whitener $\mathbf{W} = \mathbf{S}^{-1}\mathbf{Q}^*$ is used, where $\mathbf{H} = \mathbf{QSU}$ is an SVD. The performance of the blind MPLL algorithm (7) and (8) is shown in Fig. 2, where we plot the instantaneous estimation error $\|\hat{\mathbf{U}}_k^*\mathbf{WH} - \mathbf{P}_k\|_F^2$ versus time for three different alphabet sizes: 4-QAM, 16-QAM, and 64-QAM. Here, $\mathbf{P}_k$ is computed anew at each time k as the complex permutation matrix closest to $\hat{\mathbf{U}}_k^*\mathbf{WH}$. The channel $\mathbf{H}$ was generated randomly with i.i.d. $\mathcal{N}(0, 1/\sqrt{2})$ real and imaginary components, the SNR was $1/\sigma^2 = 60$ dB, and the MPLL step size was $\lambda = 0.8$. The performance of the JADE algorithm is also shown in Fig. 2.

Based on Fig. 2, we make the following observations. First, the convergence of the blind MPLL slows as the size of the alphabets increases. This is because initial decisions are less likely to be correct when the alphabet is large. Nevertheless, even with 64-QAM, the blind MPLL converges successfully after 95 iterations. Second, the blind MPLL converges faster than JADE, and it offers superior steady-state performance. This latter fact is due primarily to the decision-directed nature of the blind MPLL. In contrast, JADE does not exploit the finite-alphabet property.

The steep slopes of the MPLL curves in Fig. 2 are typical of an individual trial, but the exact instant at which the error drops can vary significantly from one trial to the next. Indeed, there may exist symbol and noise realizations for which the blind MPLL converges slower than JADE. To better illustrate

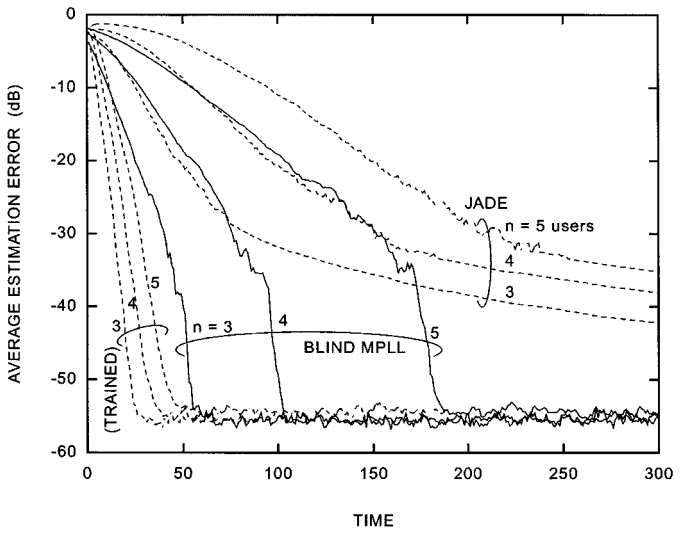

Fig. 3. Convergence slows as the number of users increases.

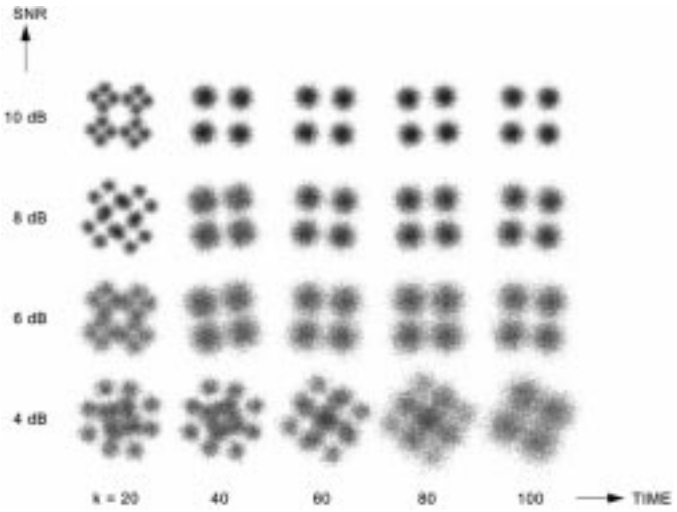

Fig. 4. User 1 constellations at different times (columns) and different SNRs (rows).

average performance, we now average the instantaneous results over 1000 independent trials. This time, we fix the alphabet at $\{\pm 1 \pm j\}/\sqrt{2}$ and vary the number of users n. As before, we assume an ideal whitener $\mathbf{W} = \mathbf{S}^{-1}\mathbf{Q}^*$, where $\mathbf{H} = \mathbf{QSU}$ is an SVD, SNR $= 1/\sigma^2 = 60$ dB, and $\lambda = 0.8$. The average performance of the MPLL algorithm is shown in Fig. 3, where we plot average estimation error $\langle \|\hat{\mathbf{U}}_k^* \mathbf{WH} - \mathbf{P}_k\|_F^2 \rangle$ versus time for $n \in \{3, 4, 5\}$. Here, the operator $\langle \cdot \rangle$ indicates an arithmetic average over the trials, and $\mathbf{P}_k$ is again the complex permutation matrix closest to $\hat{\mathbf{U}}_k^* \mathbf{WH}$. The results were averaged over 1000 independent symbol, noise, and channel realizations, with $\mathbf{H}$ having i.i.d. $\mathcal{N}(0, 1/\sqrt{2})$ real and imaginary components. Two cases are shown: The blind MPLL algorithm generates the symbol estimates $\hat{\mathbf{x}}_k$ in (7) using a memoryless component-wise decision device, and the trained MPLL algorithm uses the actual transmitted symbols $\mathbf{x}_k$ in place of $\hat{\mathbf{x}}_k$.

We should emphasize at this point that the trained MPLL is not a blind receiver since it has access to all of the transmitted symbols. Nevertheless, the trained MPLL is of interest for three reasons. First, it provides an empirical bound on the performance of the blind MPLL. Second, by comparing the blind MPLL to the trained MPLL, we can immediately discern the impact of incorrect decisions on the performance of the blind MPLL. Third, a trained MPLL may be useful in unrelated nonblind applications that are beyond the scope of this paper; for example, a trained MPLL may be used to adapt unitary matrices to implement a singular-value decomposition of the channel matrix [17], [33].

Observe from Fig. 3 that the trained MPLL converges significantly faster than the blind MPLL. The initial sluggishness of the blind MPLL can be blamed on the fact that many of the decisions are initially incorrect. The initial decision errors are not catastrophic, however; the performance of the blind MPLL eventually equals that of the trained MPLL. In addition, observe that for the trained as well as the blind MPLL, the speed of convergence decreases as the number of users increases. This latter fact is not surprising, given that at any iteration, the MPLL is able only to compensate for a rotation within the two-dimensional (2-D) subspace defined by its two inputs at that time, and hence, it must average over many iterations in order to compensate for a rotation in the full n-dimensional space.

Like the scalar PLL, the MPLL is effective at low SNR. Consider the system (2) with $n = 2$ independent 4-QAM users and an ideal whitener $\mathbf{W} = \mathbf{S}^{-1}\mathbf{Q}^*$. The performance of the blind MPLL algorithm is illustrated in Fig. 4, where the first-output constellations are shown after 20, 40, 60, 80, and 100 iterations. (The constellation at time $K \in \{20, 40, 60, 80, 100\}$ was generated by passing 30 000 symbol vectors through the noisy channel while holding $\hat{\mathbf{U}}$ fixed at $\hat{\mathbf{U}}_K$.) The results are shown for SNR $= 1/\sigma^2$ values of 4, 6, 8, and 10 dB. The channel $\mathbf{H}$ was generated randomly with i.i.d. $\mathcal{N}(0, 1/\sqrt{2})$ real and imaginary components, and the MPLL step size was $\lambda = 0.1$. The decision-directed MPLL performs well for SNR as low as 6 dB, despite the occurrence of frequent decision errors.

To examine the impact of SNR on the MPLL performance statistics, consider again the system (2) with $n = 2$ independent 4-QAM users. We applied the MPLL algorithm with $\lambda = 0.2$ over 40 000 independent channel, noise, and symbol realizations, where the components of the channel $\mathbf{H}$ were generated randomly with i.i.d. $\mathcal{N}(0, 1/\sqrt{2})$ real and imaginary components and using an ideal whitener. The performance of the blind MPLL is compared with that of the trained MPLL in Fig. 5, where we plot performance as a function of SNR $= 1/\sigma^2$ after 400 iterations. In this example, we measure performance in three different ways, using mean-squared error, median-squared error, and 90%-percentile squared error. All three performance measures take the form $\langle \|\hat{\mathbf{U}}_k^* \mathbf{WH} - \mathbf{P}_k\|_F^2 \rangle$ with $k = 400$ but with the operator $\langle \cdot \rangle$ appropriately defined in each case.

We make a number of observations based on Fig. 5. As expected, the estimation errors decrease with increasing SNR. In all cases, the blind and trained MPLL curves have the same slope at high SNR. The performance penalty of the blind MPLL relative to the trained MPLL seems to depend on which performance measure is used. In terms of mean-squared error, the blind MPLL never attains the same performance as the trained MPLL, with a SNR gap of about 2 dB. Nevertheless, the blind

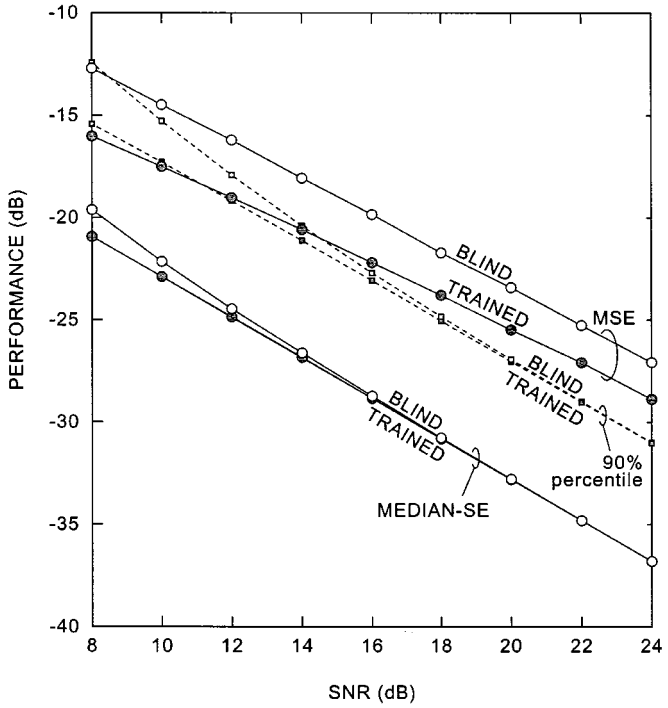

Fig. 5. MPLL performance as a function of SNR, after 400 iterations, averaged over 40 000 trials.

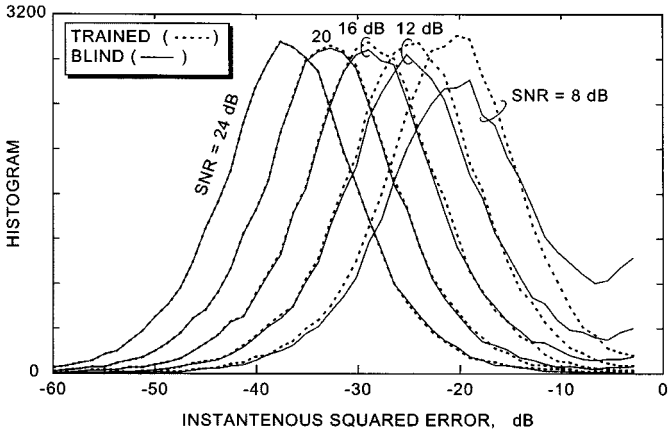

Fig. 6. Histograms of estimation error for different values of SNR, with training and blind.

and trained MPLL do agree at high SNR in terms of median-squared error and 90%-percentile squared error. Apparently, the blind MPLL departs from the trained MPLL only on a small fraction of trials, which nevertheless has a substantial impact of the mean-squared error. This bimodal behavior leads to similar median and percentile measurements but differing mean measurements. The bimodal statistics are illustrated more directly in Fig. 6, where we plot histograms of the squared estimation error from this experiment for different values of SNR. For the most part, the performance histograms for the trained and blind MPLLs coincide, even at low SNR, except for the heavier tail for the blind MPLL at high estimation errors and low SNR.

The blind MPLL is also effective in the presence of shaped alphabets with probability density functions approximating that of a Gaussian distribution. As an illustration, consider the complex alphabet defined by choosing from the hexagonal lattice $\{0.5+$

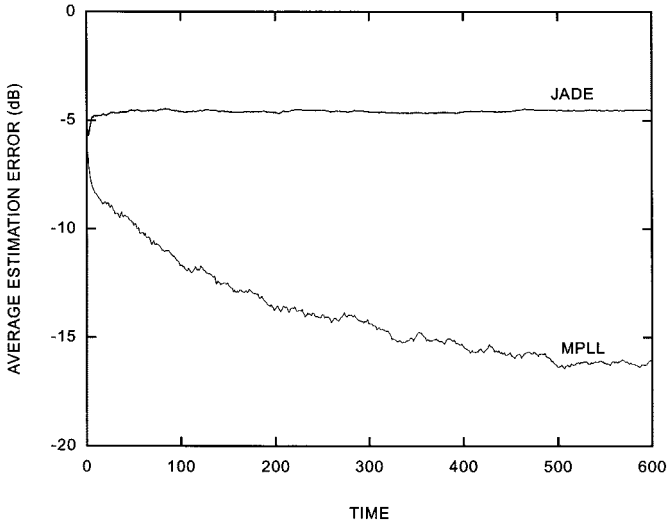

Fig. 7. Performance with a shaped constellation.

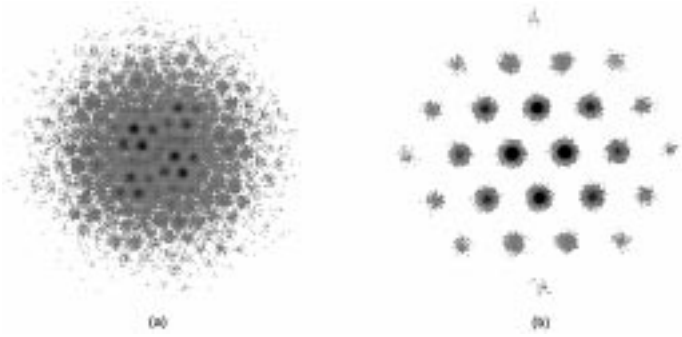

Fig. 8. Constellation of the first output for the (a) JADE and (b) blind MPLL detectors, using a shaped input alphabet.

$k + me^{j\pi/3}: k, m \in Z\}$ the 26 elements closest to the origin. To shape the alphabet, the symbols are chosen i.i.d. according to the pdf $p(a) = ce^{-|a|^2}$, with c chosen so that $\sum_a p(a) = 1$. The resulting alphabet satisfies $E[|a|^4]/\left(E[|a|^2]\right)^2 = 1.97$, which is close to the complex-Gaussian value of 2. Consider a two-user system, with both users independently drawing symbols from this shaped alphabet. The performance of the blind MPLL is compared with that of JADE in Fig. 7, where we plot average estimation error versus time. The results were averaged over 1000 independent symbol, noise, and channel realizations, with $\mathbf{H}$ having i.i.d. $\mathcal{N}(0, 1/\sqrt{2})$ real and imaginary components, with SNR $= 1/\sigma^2 = 20$ dB, and with a time-varying step size $\lambda_k = 1/(1 + k/100)$. The MPLL is seen to consistently converge even with shaped alphabets, which is a by-product of its decision-directed nature. In contrast, the JADE algorithm fails. The results of a single typical trial are shown in Fig. 8, where constellations are shown for the first output of both the MPLL and JADE algorithms, based on the final at time $k = 600$. Only the MPLL is able to faithfully reconstruct the original alphabet.

To test the performance of the MPLL in the presence of frequency offset between the transmitter and receiver local oscillators, consider the system (2) with $n = 2$ independent 4-QAM users. To emulate frequency offset for user 1, the channel matrix $\mathbf{H}$ in (2) is replaced by a time-varying matrix $\mathbf{H}_k = \mathbf{H}_0\mathbf{\Theta}_k$ with $\mathbf{\Theta}_k = \text{diag}\{e^{j2\pi k\Delta f T}, 1\}$, where Δf is the frequency offset

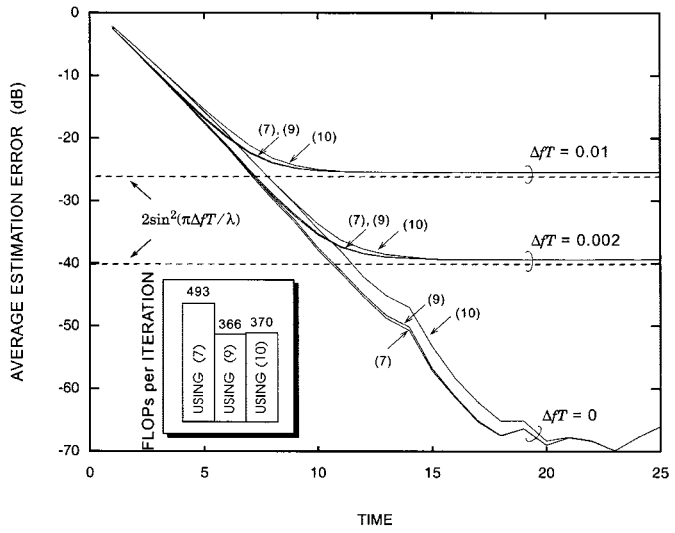

Fig. 9. Performance of a trained MPLL in the presence of frequency offset. For each ΔfT, results are shown for the three MPLLs that result from using (7), (9), and (10) for the rotation detector. The inset shows the corresponding complexity.

(in Hertz) between the transmitter and receiver local oscillators, and $1/T$ is the signaling rate. Assume an ideal whitener $\mathbf{W} = \mathbf{S}^{-1}\mathbf{Q}^*$, where $\mathbf{H}_0 = \mathbf{QSU}_0$ and $\mathbf{H}_k = \mathbf{QSU}_k$ are SVDs, with $\mathbf{U}_k = \mathbf{U}_0\mathbf{\Theta}_k$. The ideal whitener is seen to be time-invariant and independent of Δf. In this case, the unitary model of (1) applies but with a time-varying ambiguity $\mathbf{U}_k = \mathbf{U}_0\mathbf{\Theta}_k$.

The performance of the trained MPLL with frequency offset is shown in Fig. 9, where we plot average estimation error $\|\hat{\mathbf{U}}_k - \mathbf{U}_k\|_F^2$ versus time for $\Delta fT \in \{0, 0.001, 0.01\}$, SNR $= 80$ dB, and $\lambda = 0.9$. The results were averaged over 10^5 independent channel, symbol, and noise realizations, with $\mathbf{H}$ having i.i.d. $\mathcal{N}(0, 1/\sqrt{2})$ real and imaginary components. The good performance demonstrated in the figure indicates that the MPLL is able to track the time-varying unitary matrix ambiguity, albeit with a small steady-state error that increases linearly with Δf. Classical linear analysis of a single-user first-order PLL with frequency offset and in the absence of noise predicts a steady-state phase offset of $\phi = 2\pi\Delta fT/\lambda$[31]. The simulation results of the MPLL are seen to match this prediction since $\|\text{diag}\{e^{j\phi}, 1\} - \mathbf{I}\|_F^2 = 2\sin^2(\phi/2)$, as depicted by the dashed lines in Fig. 9.

Fig. 9 also illustrates the impact of using the approximate rotation detectors of (9) and (10) in place of the original rotation detector of (7) in the MPLL update of (8). Close inspection of Fig. 9 reveals that for each value of ΔfT, there are three nearly indistinguishable curves shown in Fig. 9; from lower to upper, these three curves correspond to using (7), (9), and (10). The figure indicates that the approximations (9) and (10) converge slightly slower than the original (7) but with no perceptible performance degradation at steady state. The inset to Fig. 9 illustrates the corresponding complexity reduction, with an MPLL based on the approximations of (9) and (10) requiring 74 and 75% fewer FLOPs (as measured using the MATLAB flops command) per iteration, respectively, than the original based on (7). This reduction in complexity diminishes with increasing n, as the complexity of (8) eventually dominates for large n, regardless of how the rotation detector is implemented.

V. Summary

We have proposed the MPLL as an extension of the scalar first-order PLL to vector-valued signals. We have demonstrated the effectiveness of the MPLL in the context of blind multiuser detection or blind source separation with finite-alphabet sources. Despite its decision-directed nature, the MPLL exhibits good convergence properties, with fast convergence and excellent steady-state performance. The MPLL is able to accommodate shaped alphabets and carrier frequency offset. Furthermore, the complexity of the MPLL is only $O(n^2)$, where n is the number of sources. Further results on the MPLL, including a second-order extension and an empirical convergence study, may be found in [21]. Still lacking is a theoretical understanding of the convergence properties of the MPLL.

Appendix
Derivation of the Rotation Detector

Let $\mathbf{x}, \mathbf{z} \in C^n$ satisfy $\|\mathbf{x}\| = \|\mathbf{z}\| = 1$. Let $\mathbf{R}$ denote the unitary matrix closest in Frobenius distance to the identity matrix that satisfies $\mathbf{Rx} = \mathbf{z}$ and $\mathbf{Rw} = \mathbf{w}$ for any $\mathbf{w} \perp \text{Span}\{\mathbf{x}, \mathbf{z}\}$. In this appendix, we show that $\mathbf{R}$ is given by (6) and that $\mathbf{R}^\lambda$ is given by (7).

Let p denote the inner product

$$p = \mathbf{x}^*\mathbf{z}. \tag{11}$$

Because $\mathbf{x}$ and $\mathbf{z}$ are unit length, p satisfies $|p| \leq 1$. Consider first the case $|p| = 1$, which implies that $\mathbf{z} = p\mathbf{x}$. Let $\mathbf{V} = [\mathbf{x}, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \ldots, \mathbf{v}_n]$ be a unitary matrix whose columns form a basis for C^n. In this case, the set of all unitary matrices mapping $\mathbf{x}$ to $\mathbf{z}$ is given by

$$\mathbf{R} = \mathbf{V}\begin{bmatrix} p & \mathbf{0} \\ \mathbf{0} & \mathbf{J} \end{bmatrix}\mathbf{V}^* \tag{12}$$

for some $(n-1) \times (n-1)$ unitary matrix $\mathbf{J}$. To ensure that $\mathbf{Rw} = \mathbf{w}$ for $\mathbf{w} \in \text{Span}\{\mathbf{x}, \mathbf{z}\}$, we must have $\mathbf{J} = \mathbf{I}$. It follows that $\mathbf{R}^\lambda$ is

$$\mathbf{R}^\lambda = \mathbf{I} + (p^\lambda - 1)\mathbf{x}\mathbf{x}^* \quad (\text{when } |p| = 1). \tag{13}$$

This is identical to (6) with $\mathbf{v} = \mathbf{0}$.

On the other hand, if $|p| < 1$, then $\mathbf{x}$ and $\mathbf{z}$ span a 2-D subspace. Let us introduce the angles $\alpha = \sin^{-1}(\text{Im}\{p/|p|\})$ and $\theta = \cos^{-1}(|p|)$ so that p may be written as $p = e^{j\alpha}\cos(\theta)$. Let $\{\mathbf{x}, \mathbf{v}\}$ be a basis for the span of $\mathbf{x}$ and $\mathbf{z}$, where from the Gram–Schmidt procedure, we have

$$\mathbf{v} = \frac{(\mathbf{z} - p\mathbf{x})}{\sin(\theta)}. \tag{14}$$

Let $\mathbf{V} = [\mathbf{x}, \mathbf{v}, \mathbf{v}_3, \mathbf{v}_4, \ldots \mathbf{v}_n]$ be a unitary matrix whose columns form an orthonormal basis for C^n. In terms of this basis, $\mathbf{x}$ and $\mathbf{z}$ are described by the vectors $[1, 0, \ldots, 0]^T$ and $\left[e^{j\alpha}\cos(\theta), \sin(\theta), 0, 0, \ldots, 0\right]^T$, respectively. However, any

unitary matrix mapping $[1,0]^T$ to $\left[e^{j\alpha}\cos(\theta),\sin(\theta)\right]^T$ must have the Givens-like form

$$\tilde{\mathbf{R}} = \begin{bmatrix} e^{j\alpha} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{j(\beta-\alpha)} \end{bmatrix} \quad (15)$$

for some $\beta \in (-\pi, \pi]$. Therefore, $\mathbf{R}$ must take the form

$$\mathbf{R} = \mathbf{V} \begin{bmatrix} \tilde{\mathbf{R}} & \mathbf{0} \\ \mathbf{0} & \mathbf{J} \end{bmatrix} \mathbf{V}^* \quad (16)$$

where $\beta \in (-\pi, \pi]$ and $\mathbf{J}$ is some $(n-2)\times(n-2)$ unitary matrix. Again, to satisfy $\mathbf{R}\mathbf{w} = \mathbf{w}$ for any $\mathbf{w} \perp \text{Span}\{\mathbf{x}, \mathbf{z}\}$, we must have $\mathbf{J} = \mathbf{I}$. It is easy to see from (15) that choosing $\beta = \alpha$ will minimize the Frobenius distance between $\mathbf{R}$ and the identity matrix, in which case, (15) reduces to

$$\tilde{\mathbf{R}} = \begin{bmatrix} e^{j\alpha} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}. \quad (17)$$

From (17) and (16) with $\mathbf{J} = \mathbf{I}$, $\mathbf{R}$ reduces to (6).

We now derive the expression for $\mathbf{R}^\lambda$ of (7). Since $\tilde{\mathbf{R}}$ of (17) is unitary, its eigendecomposition takes the form $\tilde{\mathbf{R}} = \mathbf{V}\mathbf{D}\mathbf{V}^*$ with $\mathbf{D} = \text{diag}\{d_1, d_2\}$, where both $\mathbf{D}$ and $\mathbf{V}$ are unitary [34]. It can be verified that

$$\mathbf{V} = \begin{bmatrix} \sqrt{1-\rho^2} & -j\rho e^{j\alpha/2} \\ -j\rho e^{-j\alpha/2} & \sqrt{1-\rho^2} \end{bmatrix} \quad (18)$$

and $d_1 = e^{j\alpha/2}e^{j\Delta}$, $d_2 = e^{j\alpha/2}e^{-j\Delta}$, where $\Delta = \cos^{-1}(|p+|p||/2)$, and

$$\rho = \sqrt{\frac{1}{2}\left(1 - \frac{|p|\sin\left(\frac{\alpha}{2}\right)}{\sin(\Delta)}\right)}.$$

We can then compute $\tilde{\mathbf{R}}^\lambda$ as $\tilde{\mathbf{R}}^\lambda = \mathbf{V}\mathbf{D}^\lambda\mathbf{V}^*$, which, after some manipulation, reduces to

$$\tilde{\mathbf{R}}^\lambda = \begin{bmatrix} d_2^\lambda + \gamma s & \gamma(d_1 - p) \\ \gamma(d_1 - p)^* & d_1^\lambda - \gamma s \end{bmatrix} \quad (19)$$

where

$$\gamma = je^{j\lambda\alpha/2}\frac{\sin(\theta)\sin(\lambda\Delta)}{\sin(\Delta)|d_1 - p|}.$$

From (19) and (16) with $\mathbf{J} = \mathbf{I}$, $\mathbf{R}^\lambda$ reduces to (7). Q.E.D.

REFERENCES

[1] L. J. Griffiths and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. Antennas Propagat.*, vol. AP-30, pp. 27–34, Jan. 1982.

[2] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propagat.*, vol. AP-34, pp. 276–280, 1986.

[3] R. Roy and T. Kailath, "ESPRIT estimation of signal parameter via rotational invariance techniques," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, pp. 984–993, July 1989.

[4] M. Honig, U. Madhow, and S. Verd, "Blind adaptive multiuser detection," *IEEE Trans. Inform. Theory*, vol. 41, pp. 944–960, July 1995.

[5] X. Wang and H. V. Poor, "Blind adaptive interference suppression for CDMA communication based on eigenspace tracking," in *Proc. Conf. Inform. Sci. Syst.*, 1997, pp. 468–473.

[6] M. L. Honig, "A comparison of subspace adaptive filtering techniques for DS-CDMA interference suppression," in Military Conf. Commun., Monterey, Ca, Nov. 1997.

[7] J.-F. Cardoso, "Blind signal separation: Statistical principles," *Proc. IEEE*, vol. 9, pp. 2009–2025, Oct. 1998.

[8] M. Gaeta and J. L. Lacoume, "Source separation without a priori knowledge: The maximum likelihood solution," in *Proc. Eur. Signal Process. Conf.*, 1990, pp. 621–624.

[9] L. Tong, Y. Inouye, and R. Liu, "Waveform preserving blind estimation of multiple independent sources," *IEEE Trans. Signal Processing*, vol. 41, pp. 2461–2470, July 1993.

[10] J. F. Cardoso and A. Souloumiac, "Blind beamforming for non-Gaussian signals," *Proc. Inst. Elect. Eng. F, Radar Signal Process.*, vol. 140, no. 6, pp. 362–370, Dec. 1993.

[11] P. Comon, "Independent component analysis, a new concept?," *Signal Process. Special Issue Higher Order Statist.*, vol. 36, no. 3, pp. 287–314, Apr. 1994.

[12] J.-F. Cardoso, "On the performance of orthogonal source separation algorithms," in *Proc. Eur. Signal Process. Conf.*, Edinburgh, U.K., Sept. 1994, pp. 776–779.

[13] Gorokhov, P. Loubaton, and E. Moulines, "Second-order blind equalization in multiple input multiple output FIR systems: A weighted least squares approach," in *Proc. ICASSP*, vol. 5, Atlanta, GA, 1996, pp. 2417–2420.

[14] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," *IEEE Trans. Signal Processing*, vol. 45, pp. 434–444, Feb. 1997.

[15] R. T. Causey and J. R. Barry, "A project-whiten-rotate blind multiuser detector," in *Proc. Commun. Theory Mini-Conf.*, Sydney, Australia, Nov. 1998, pp. 125–130.

[16] ——, "Blind multiuser detection using linear prediction," *IEEE J. Select. Areas Commun.*, vol. 16, no. 8, pp. 1702–1710, Dec. 1998.

[17] R. T. Causey, "Blind multiuser detection based on second-order statistics," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, 1999.

[18] J. Xavier and V. Barroso, "A channel order independent method for blind equalization of MIMO systems," in *Proc. ICASSP*, vol. 5, 1999, pp. 3897–2900.

[19] M. Meng, N. H. Dowlut, and T. N. Davidson, "A derotation approach to the blind separation of synchronous cochannel BPSK signals," in *Proc. ICASSP*, vol. 5, 1999, pp. 2957–2959.

[20] M. Wax and Y. Anu, "A least squares approach to blind beamforming," *IEEE Trans. Signal Processing*, vol. 47, pp. 231–234, Jan. 1999.

[21] A. Batra, "Extensions of the constant modulus algorithm and the phase-locked loop for blind multiuser detection," Ph.D. dissertation, Georgia Inst. Technol., Atlanta, GA, 2000.

[22] J. F. Cardoso and P. Comon, "Tensor-based independent component analysis," in *Signal Processing V: Theory and Applications*. New York: Elsevier, 1990.

[23] J.-F. Cardoso and B. H. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Processing*, vol. 44, pp. 3017–3030, Dec. 1996.

[24] V. Zarzoso and A. Nandi, "Adaptive blind source separation for virtually any source probability density function," *IEEE Trans. Signal Processing*, vol. 48, pp. 477–488, Feb. 2000.

[25] K. Anand and V. U. Reddy, "Blind separation of multiple cochannel BPSK signals arriving at an antenna array," *IEEE Signal Processing Lett.*, vol. 2, no. 9, pp. 176–178, Sept. 1995.

[26] A.-J. van der Veen, "Analytical method for blind binary signal separation," *IEEE Trans. Signal Processing*, vol. 45, pp. 1078–1082, Apr. 1997.

[27] H. Oda and Y. Sato, "A method of multidimensional blind equalization," in *Proc. IEEE Int. Symp. Inform. Theory*, San Antonio, TX, 1993, p. 327.

[28] A. Batra and J. R. Barry, "Blind cancellation of co-channel interference," in *Proc. IEEE Global Telecommun. Conf.*, vol. 1, Singapore, Nov. 13–17, 1995, pp. 157–162.

[29] A. Patapoutian, "On phase-locked loops and kalman filters," *IEEE Trans. Commun.*, vol. 47, pp. 670–672, May 1999.

[30] J. K. Tugnait and B. Huang, "Multistep linear predictors-based blind identification and equalization of multiple-input multiple-output channels," *IEEE Trans. Signal Processing*, vol. 48, pp. 26–38, Jan. 2000.

[31] E. A. Lee and D. G. Messerschmitt, *Digital Communication*, 2nd ed. Boston, MA: Kluwer, 1994.

[32] M. K. Simon and J. G. Smith, "Carrier synchronization and detection of QASK signal sets," *IEEE Trans. Commun.*, vol. COM–22, pp. 98–106, Feb. 1974.

[33] J.-H. Sung and J. R. Barry, "Space-time processing with channel knowledge at the transmitter," in *Proc. Eur. Conf. Commun.*, vol. 1, Bratislava, Slovakia, July 5–7, 2001, pp. 26–29.

[34] C. R. Johnson and R. A. Horn, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

John R. Barry (M'92) received the B.S. degree in electrical engineering from the State University of New York, Buffalo, in 1986 and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley in 1987 and 1992, respectively.

Since 1992, he has been with the Georgia Institute of Technology, Atlanta, where he is an Associate Professor with the School of Electrical and Computer Engineering. His research interests include wireless communications, equalization, and multiuser communications. He is the author of *Wireless Infrared Communications* (Boston, MA: Kluwer, 1994).

Anuj Batra (M'00) received the B.S. degree, with distinction, in electrical engineering from Cornell University, Ithaca, NY, in 1992, the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 1993, and the Ph.D. degree in electrical engineering from Georgia Institute of Technology, Atlanta, in 2000.

In 1992, he was with Raytheon E-Systems, Falls Church, VA, where he designed algorithms for mobile cellular phones based on the AMPS standard. He is currently a Member of Technical Staff with the Digital Signal Processing Solutions Research and Development Center, Texas Instruments, Inc., Dallas, TX. His current research interests are in the areas of wireless communications, in particular, the design of high-speed wireless networks, multiuser detection theory, and coexistence between unlicensed wireless devices. Since joining Texas Instruments, he has also been involved in standardization activities for IEEE 802.11g, IEEE 802.15.2, and Bluetooth.

Dr. Batra is a member of Eta Kappa Nu and Tau Beta Pi.