

Blind Unitary Source Separation using a Multidimensional Phase-Locked Loop

Anuj Batra and John R. Barry

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

Abstract — Blind source separation cannot generally be performed using second-order statistics alone because of a unitary matrix ambiguity. We present the multidimensional phase-locked loop (MPLL) as a blind algorithm for resolving this ambiguity. The MPLL is a multidimensional generalization of the scalar decision-directed PLL for resolving phase rotations in scalar digital communication systems, and as such is applicable only to discrete-alphabet sources. We compare the MPLL to other known unitary source separation algorithms, and find that the MPLL compares favorably in terms of both performance and complexity.

I. INTRODUCTION

We consider blind source separation for a multiple-input, multiple-output (MIMO) channel in which n independent digital source signals are observed by a receiver with m sensors, as shown in Fig. 1(a). The channel is characterized by an $m \times n$ convolutive transfer function $\mathbf{H}(z)$. The channel input \mathbf{x}_k is a vector sequence representing the discrete symbol sequences of n different users. The objective of a blind source separator is to recover these sequences from the channel output \mathbf{r}_k without *a priori* knowledge of the channel or its input.

It is often convenient to decompose the source separation process into three steps, as illustrated in Fig. 1(b): *eliminate memory* using an $m \times m$ filter $\mathbf{B}(z)$, *whiten* using an $n \times m$ matrix \mathbf{W} , and *rotate* using a unitary matrix \mathbf{U} (or $\mathbf{U}_d \mathbf{U}$ when necessary). The purpose of the first filter $\mathbf{B}(z)$ is to eliminate memory, so that the cascade $\mathbf{H} = \mathbf{B}(z)\mathbf{H}(z)$ is a memoryless

mixing matrix. The purpose of the memoryless whitener \mathbf{W} is to eliminate the mixing matrix up to an arbitrary unitary matrix, so that the cascade $\mathbf{F} = \mathbf{W}\mathbf{B}(z)\mathbf{H}(z)$ is a unitary matrix. The first two steps can be accomplished by either a one-step linear predictor (if $m > n$) followed by a spatial whitener [1] or by the vector CMA algorithm [2].

The first two steps reduce the problem to a blind unitary source separation problem, which is the topic of this paper. Since both \mathbf{H} and $\mathbf{H}\mathbf{U}$ produce the same covariance matrix for any unitary matrix \mathbf{U} , second-order statistics are insufficient for resolving the unitary ambiguity [3-5]; higher-order statistics are necessary.

Cardoso [3,4] showed that the columns of the unitary matrix are the eigenvectors of the $n \times n$ cumulant matrix $\mathbf{Q}(\mathbf{M})$ for a particular matrix \mathbf{M} . The cumulant matrix can be estimated by time-averaging the moments of the output of the whitening matrix. Cardoso and Souloumiac [6] have proposed an improved algorithm called the joint approximate diagonalization of eigenmatrices (JADE) algorithm, that jointly and approximately diagonalizes a set of matrices $\mathbf{Q}(\mathbf{M})$ for different values of \mathbf{M} . JADE is a batch-oriented algorithm with a relatively high computational complexity of $\mathcal{O}(n^5)$ [7].

Cardoso and Laheld [8] have recently introduced a class of algorithms called equivariant adaptive separation via independence (EASI) that are based upon a serial update equation:

$$\mathbf{C}_{k+1} = \mathbf{C}_k - \mu \mathbf{G}(\tilde{\mathbf{z}}_k) \mathbf{C}_k, \quad (1)$$

where $\tilde{\mathbf{z}}_k$ is the output of the separating matrix \mathbf{C}_k , $\mathbf{G}(\tilde{\mathbf{z}}_k)$ is a function defined to be the “relative gradient,” and μ is a small positive constant. These EASI algorithms perform both the whitening and rotation functions, so that $\mathbf{C} = \mathbf{U}\mathbf{W}$ in Fig. 1(b). The EASI algorithms are attractive because they are adaptive and have low complexity, $\mathcal{O}(n^3)$.

Both JADE and EASI generate a rotation matrix \mathbf{U} which diagonalizes \mathbf{F} but may result in a residual phase error for each component. To compensate, a bank of scalar phase-locked loops (represented by the diagonal matrix \mathbf{U}_d in Fig. 1(b)) is needed. Both JADE and EASI are flexible blind unitary separators because they make no assumptions about the structure of the input signals, only that they are independent. In contrast, we propose an adaptive blind unitary separator algorithm that exploits the discrete nature of digital

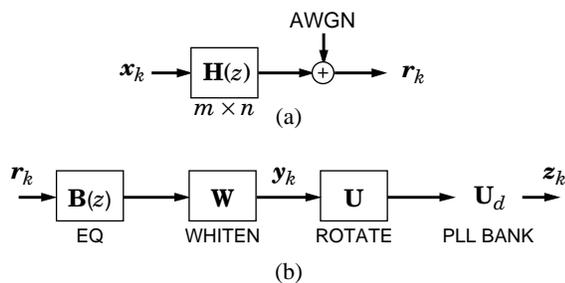


Fig. 1. (a) MIMO communications channel model;
(b) the three-step blind source separator.

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

communications signals [9]. We view the unitary matrix as a constant rotation akin to a constant phase offset in a scalar communication system.

In section II we review the multidimensional phase-locked loop. In section III we compare the performance and complexity of the multidimensional phase-locked loop to that of JADE and EASI.

II. MULTIDIMENSIONAL PHASE-LOCKED LOOP

In analogy to a decision-directed phase-locked loop (PLL) for eliminating phase ambiguity in scalar digital communications systems, the multidimensional PLL (MPLL) is a blind decision-directed equalizer that generates an estimate \mathbf{U}_k of the unitary matrix ambiguity $\mathbf{F} = \mathbf{W}\mathbf{B}(z)\mathbf{H}(z)$ [9]. The block diagram of the MPLL is shown in Fig. 2. The input \mathbf{y}_k to the MPLL is the whitener output, which we assume has the form:

$$\mathbf{y}_k = \mathbf{F}\mathbf{x}_k + \mathbf{n}_k, \quad (2)$$

where \mathbf{F} is unitary and \mathbf{n}_k represents noise. If \mathbf{U}_k is an accurate estimate of \mathbf{F} , then the product $\mathbf{z}_k = \mathbf{U}_k^* \mathbf{y}_k$ [where $(\cdot)^*$ denotes conjugate transpose] reduces to $\mathbf{z}_k \approx \mathbf{x}_k$ at high SNR; the receiver can then produce an accurate estimate $\hat{\mathbf{x}}_k$ of \mathbf{x}_k by quantizing \mathbf{z}_k . The MPLL generates \mathbf{U}_k in a way similar to the scalar PLL. The key component of the MPLL is the rotation detector which generates a unitary rotation matrix \mathbf{T}_k that approximately rotates $\hat{\mathbf{x}}_k$ to \mathbf{z}_k . This rotation is necessarily approximate because the noise causes $\hat{\mathbf{x}}_k$ and \mathbf{z}_k to have different norms.

Because of noise and occasional decision errors, a first-order scalar PLL typically employs a loop filter with gain λ less than unity; this choice improves robustness and stability at the expense of slower convergence. Extending the loop filter to multiple dimensions requires that we raise the rotation matrix \mathbf{T}_k to the power $\lambda \in (0, 1)$. \mathbf{T}_k^λ is a partial rotation matrix, rotating $\hat{\mathbf{x}}_k$ a fraction of the way to \mathbf{z}_k . Later in this section we will explicitly determine \mathbf{T}_k^λ to avoid the computationally complex task of raising a matrix to a fractional power.

The MPLL accumulates the partial rotation matrices, so that the estimate of the overall rotation matrix at time k is given by:

$$\mathbf{U}_k = \prod_{i=0}^{k-1} \mathbf{T}_i^\lambda, \quad (3)$$

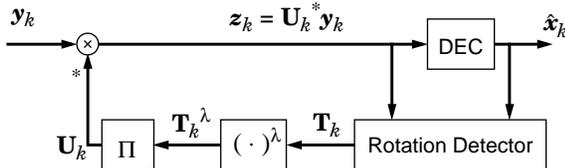


Fig. 2. The decision-directed multidimensional PLL.

which can be implemented recursively:

$$\begin{aligned} \mathbf{U}_{-1} &= \mathbf{I}, \\ \mathbf{U}_k &= \mathbf{U}_{k-1} \mathbf{T}_{k-1}^\lambda. \end{aligned} \quad (4)$$

To deal with the noisy case when $\hat{\mathbf{x}}_k$ and \mathbf{z}_k may have different lengths, we normalize both to have unit norm before estimating the rotation matrix. Define \mathbf{x} and \mathbf{z} by:

$$\mathbf{x} = \hat{\mathbf{x}}_k / \|\hat{\mathbf{x}}_k\| \quad \text{and} \quad \mathbf{z} = \mathbf{z}_k / \|\mathbf{z}_k\|. \quad (5)$$

We then define \mathbf{T}_k as a unitary matrix mapping \mathbf{x} to \mathbf{z} . It is important to note that \mathbf{T}_k is not unique. A good choice for \mathbf{T}_k was derived in [9], which we review here in a slightly modified form. Let p denote the inner product:

$$p = \mathbf{x}^* \mathbf{z}. \quad (6)$$

Because \mathbf{x} and \mathbf{z} are of unit length, p satisfies $|p| \leq 1$. We consider the cases $|p| = 1$ and $|p| < 1$ separately.

If $|p| = 1$ then $\mathbf{z} = p\mathbf{x}$, and we have:

$$\mathbf{T}_k = \mathbf{I} + (p - 1)\mathbf{x}\mathbf{x}^* \quad (\text{when } |p| = 1). \quad (7)$$

Because \mathbf{x} and \mathbf{z} convey information about the one-dimensional subspace spanned by \mathbf{x} only, we have chosen not to affect the vectors orthogonal to this subspace. The partial rotation matrix for $|p| = 1$ is then given by:

$$\mathbf{T}_k^\lambda = \mathbf{I} + (p^\lambda - 1)\mathbf{x}\mathbf{x}^*. \quad (8)$$

On the other hand, if $|p| < 1$, then the span of \mathbf{x} and \mathbf{z} is a two-dimensional subspace. Let us introduce a basis $\{\mathbf{x}, \mathbf{y}\}$ for this subspace, where the Gram-Schmidt procedure yields:

$$\mathbf{y} = (\mathbf{z} - p\mathbf{x}) / \sqrt{1 - |p|^2}. \quad (9)$$

In terms of this basis, \mathbf{x} and \mathbf{z} are given by the vectors $[1, 0]^T$ and $[p, \sqrt{1 - |p|^2}]^T$, respectively. It can be shown that any unitary matrix mapping $[1, 0]^T$ to $[p, \sqrt{1 - |p|^2}]^T$ must have the form:

$$\tilde{\mathbf{R}} = \begin{bmatrix} p & -\sqrt{1 - |p|^2} \\ \sqrt{1 - |p|^2} & p^* \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & e^{j\beta} \end{bmatrix}, \quad (10)$$

for some $\beta \in (-\pi, \pi]$. As argued in [9], a good choice for β is $\angle p$, which leads to:

$$\mathbf{T}_k = \mathbf{I} + \begin{bmatrix} \mathbf{x}, \mathbf{y} \end{bmatrix} \begin{bmatrix} p - 1 & \frac{-p}{|p|} \sqrt{1 - |p|^2} \\ \sqrt{1 - |p|^2} & |p| - 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \mathbf{y}^* \end{bmatrix} \quad (|p| < 1). \quad (11)$$

Finally, the partial rotation matrix for $|p| < 1$ becomes:

$$\mathbf{T}_k^\lambda = \mathbf{I} + \begin{bmatrix} \mathbf{x}, \mathbf{y} \end{bmatrix} \begin{bmatrix} d_1^\lambda |v|^2 + d_2^\lambda - 1 & (d_1^\lambda - d_2^\lambda)v \\ (d_1^\lambda - d_2^\lambda)v^* & d_1^\lambda + d_2^\lambda |v|^2 - 1 \end{bmatrix} \begin{bmatrix} \mathbf{x}^* \\ \mathbf{y}^* \end{bmatrix}, \quad (12)$$

where $d_1 = 0.5(p + |p| + s)$, $d_2 = 0.5(p + |p| - s)$, $v = \frac{d_1}{\sqrt{1 - |p|^2}}$, and $s = \frac{p}{\sqrt{|p|^2 + 2p|p| + p^2 - 4\frac{p}{|p|}}}$.

In summary, the MPLL of Fig. 2 is defined by (4), (8), and (12), together with (5), (6), and (9), and is parameterized by the constant λ .

We remark that, when $n = 1$, the partial rotation matrix \mathbf{T}_k^λ reduces to $e^{j\lambda\theta}$, where θ is the angle between $\hat{x}_k / |\hat{x}_k|$ and $z_k / |z_k|$, and that the MPLL reduces to the conventional first-order decision-directed scalar PLL.

III. PERFORMANCE AND COMPLEXITY

In this section we compare the performance and complexity of three unitary source separators: JADE [6], EASI [8] (with $\mathbf{G}(\mathbf{z}) = \mathbf{z}\mathbf{z}^* - \mathbf{I} + \mathbf{g}(\mathbf{z})\mathbf{z}^* - \mathbf{z}\mathbf{g}(\mathbf{z})^*$ and $\mathbf{g}(\mathbf{z}) = \mathbf{z} \otimes \mathbf{z}^* \otimes \mathbf{z}$, where \otimes indicates a component-wise product), and the MPLL. The performance criteria used in this comparison is the mean-squared error: $\text{MSE} = \text{E}[\|\mathbf{P}\mathbf{x}_k - \mathbf{z}_k\|^2]$, where \mathbf{P} is a complex permutation matrix that reorders the sources and rotates the phase of each source by a multiple of 90° ; this permutation is necessary to account for the inherent ambiguity that exists in any blind separation problem.

We next present simulation results from three experiments. In all cases we assume that the channel inputs are selected independently and uniformly from a 16-QAM alphabet.

2 × 2 Memoryless Unitary Channel

In the first experiment we consider a noiseless 2×2 system $\mathbf{y}_k = \mathbf{F}\mathbf{x}_k$ with a memoryless unitary channel matrix \mathbf{F} . The step size for EASI decreased with time according to $\mu_k = 0.1/(1 + k/50)$, the step size for the MPLL decreased with time according to $\lambda_k = 0.9/(1 + k/1000)$, and the bank of second-order scalar PLLs (required by both JADE and EASI) had proportional-plus-integral parameters $\alpha_1 = 0.1$ and $\alpha_2 = 0.005$. The parameters for each algorithm were optimized to provide the fastest rate of convergence.

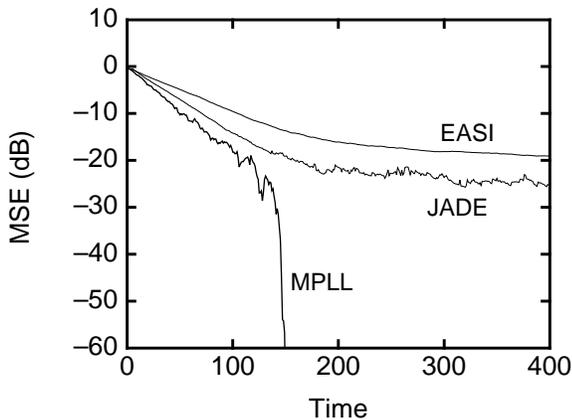


Fig. 3. Performance comparison of the three unitary source separators for experiment one.

In Fig. 3 we plot MSE versus time for each algorithm, averaged over 1000 random unitary channels \mathbf{F} . To produce these random channels, we generated a memoryless Gaussian channel matrix \mathbf{H} having *i.i.d.* real and imaginary components normally distributed with zero mean and unit variance, and then appended the whitening matrix $\mathbf{W} = \mathbf{D}^{-1/2}\mathbf{V}^*$, where the diagonal matrix \mathbf{D} and unitary matrix \mathbf{V} are defined by the eigendecomposition $\mathbf{H}\mathbf{H}^* = \mathbf{V}\mathbf{D}\mathbf{V}^*$. This yields the random unitary matrix $\mathbf{F} = \mathbf{D}^{-1/2}\mathbf{V}^*\mathbf{H}$. Fig. 3 shows that the MPLL converges much faster than the other algorithms. Observe that the steady-state performance of the MPLL is far superior to that of the other blind algorithms; this superiority is because of the decision-directed nature of the MPLL.

3 × 3 Memoryless Unitary Channel

In the second experiment we again consider the noiseless system $\mathbf{y}_k = \mathbf{F}\mathbf{x}_k$ with unitary \mathbf{F} , only this time the dimension of \mathbf{F} is 3×3 . Again the parameters for each algorithm were optimized for fast convergence: the EASI step size at time k was $\mu_k = 0.1/(1 + k/50)$, the MPLL step size was $\lambda_k = 0.65/(1 + k/1000)$, and the scalar PLL bank (used by both JADE and EASI) had parameters $\alpha_1 = 0.1$ and $\alpha_2 = 0.0005$.

In Fig. 4 we plot MSE versus time for each algorithm, averaged over 750 random unitary channels, generated in a manner analogous to experiment one. Fig. 4 shows that both JADE and EASI offer fast initial convergence, but the MPLL outperforms them both after 600 symbols. The slower rate of initial convergence for the MPLL is likely due to the fact that at each iteration the MPLL is able to compensate for a rotation in two dimensions only, whereas the unitary ambiguity is a 3×3 matrix.

Noisy 3 × 2 Memoryless Gaussian Channel

In the third experiment we illustrate the effects of noise by considering a noisy 3×2 system $\mathbf{r}_k = \mathbf{H}\mathbf{x}_k + \mathbf{n}_k$ with a memoryless channel matrix \mathbf{H} . We assume the noise \mathbf{n}_k has independent Gaussian real and imaginary parts, each with

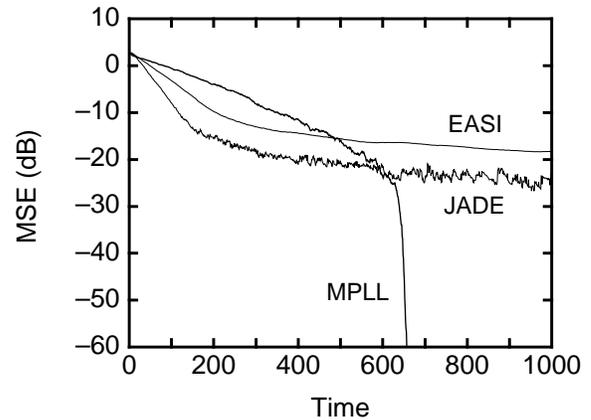


Fig. 4. Performance comparison of the three unitary source separators for experiment two.

covariance matrix $\sigma^2 \mathbf{I}$. The 3×2 Gaussian channel matrix \mathbf{H} is normalized to have unity Frobenius norm. To make a fair comparison, we assume all three unitary source separators use the same whitening filter $\mathbf{W} = [\mathbf{D}_{2 \times 2}]^{-1/2} [\mathbf{V}_{3 \times 2}]^*$, where $\mathbf{H}\mathbf{H}^* = [\mathbf{V}_{3 \times 2}] [\mathbf{D}_{2 \times 2}] [\mathbf{V}_{3 \times 2}]^*$ is a truncated SVD.

In Fig. 5 we plot the convergence time versus the per-user SNR = $E[\|\mathbf{r}_k^{(i)}\|^2] / E[\|\mathbf{n}_k\|^2] = 1 / (2mn\sigma^2)$, where $\mathbf{r}_k^{(i)}$ is the contribution of \mathbf{r}_k due to the i -th source only. We define algorithm convergence as when the signal component of the squared-error $\|\mathbf{F}\mathbf{x}_k - \mathbf{P}\mathbf{x}_k\|^2$ is less than -12 dB for 40 consecutive symbols. This plot is generated by averaging over 100 random Gaussian channels at each SNR. Although the parameters could have been optimized at each value of SNR, the same values were used for all SNR values. The EASI step size decreased with time according to $\mu_k = 0.01 / (1 + k / 500)$, the MPLL step size was $\lambda_k = 0.3 / (1 + k / 500)$, and the scalar PLL bank had parameters $\alpha_1 = 0.1$ and $\alpha_2 = 0.0005$.

At low SNR, the JADE algorithm converges faster than the MPLL and EASI. The convergence speed of the MPLL approaches that of JADE at high SNR. The MPLL convergence time can be improved further, especially at high SNR, by optimizing the step size λ at each SNR value. It should be noted that the behavior of the curves in Fig. 5 depends radically on the MSE threshold that defines convergence. Modifying the threshold from -12 dB to -30 dB, for example, would cause the MPLL to “converge” much faster than the other algorithms, as is evident from Fig. 3 and Fig. 4.

Complexity Comparison

Finally, to demonstrate the low complexity of the MPLL, we plot in Fig. 6 the number of floating point operations required per iteration for each algorithm versus the number of users for a 20 sensor system. The figure shows that EASI and the MPLL have equivalent complexity, $O(n^3)$. The high complexity of the JADE curve is misleading because we made no attempt to optimize the JADE computer code; code optimization can reduce the complexity to $O(n^5)$ [7].

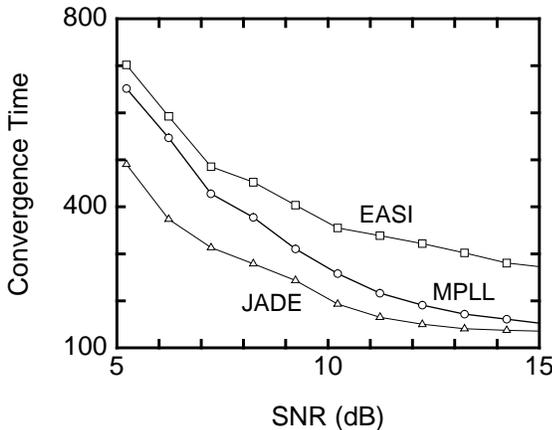


Fig. 5. Convergence time versus SNR for experiment three.

IV. SUMMARY

We have presented the multidimensional PLL as a blind decision-directed algorithm for unitary source separation. The MPLL exploits the discrete nature of digital communication signals and compares favorably in terms of both performance and complexity to other known source-separation algorithms. The MPLL offers fast convergence, excellent steady-state performance, and low complexity. Because the MPLL is a generalization of the scalar PLL, it is subject to false lock when the SNR is high and the step size is small. But just as in the scalar case, the probability of false lock can be minimized through careful choice of the step size.

V. REFERENCES

- [1] A. Gorokhov, P. Loubaton, and E. Moulines, “Second Order Blind Equalization in Multiple Input Multiple Output FIR Systems: A Weighted Least-Squares Approach”, *ICASSP*, vol. 5, pp. 2415-2418, 1996.
- [2] A. Batra and J. R. Barry, “Blind Cancellation of Co-Channel Interference,” *Globecom*, Singapore, vol. 1, pp. 157-162, 1995.
- [3] J. F. Cardoso, “Source Separation using Higher Order Moments”, *ICASSP*, vol. 4, pp. 2109-2112, 1989.
- [4] J. F. Cardoso, “Eigen-Structure of the Fourth-Order Cumulant Tensor with Application to the Blind Source Separation Problem”, *ICASSP*, vol. 5, pp. 2655-2658, 1990.
- [5] P. Comon, “Independent Component Analysis, A New Concept?”, *Signal Processing*, vol. 36, no. 3, pp. 287-314, 1994.
- [6] J. F. Cardoso and A. Souloumiac, “Blind Beamforming for non-Gaussian Signals”, *IEE Proceedings F (Radar and Signal Processing)*, vol. 140, no. 6, pp. 362-370, December 1993.
- [7] J. F. Cardoso and P. Comon, “Tensor-Based Independent Component Analysis,” in *Signal Processing V: Theory and Applications*, Elsevier, 1990.
- [8] J. F. Cardoso and B. H. Laheld, “Equivariant Adaptive Source Separation”, *IEEE Transactions on Signal Processing*, vol. 44, no.12, pp. 3017-3030, December 1996.
- [9] J. R. Barry and A. Batra, “A Multidimensional Phase-Locked Loop for Blind Equalization of Multi-Input Multi-Output Channels,” *ICC*, vol. 3, pp. 1307-1312, 1996.

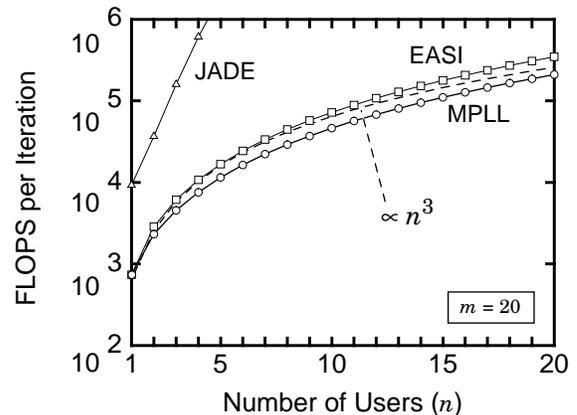


Fig. 6. Floating point operations versus the number of users.