Interference Subspace Tracking for Network Interference Alignment in Cellular Systems

Bo Niu, Osvaldo Simeone, and Alexander M. Haimovich
Department of Electrical and Computer Engineering
New Jersey Institute of Technology
Newark, NJ 07102, USA

***Abstract*— In this paper, a practical solution to implement the distributed interference alignment (IA) algorithm of [1] in a cellular communication system is proposed. In the training period of the proposed strategy, the User Equipment (UE) and the Base Station (BS), without knowing *a priori* the interference covariance matrix, update directly the precoding and interference suppression matrices based on the received signals by a minor subspace tracking algorithm. At the end of the training phase, the precoding and interference suppression matrices are then used at the UE and BS, respectively, for the transmission period. A special spatial reuse method is also proposed for the training period to lower the system overhead. Numerical system performance results are provided, showing that the algorithm, referred to as Interference Subspace Tracking IA (IST-IA), yields a good trade-off between throughput gains, on one side, and training overhead and computational complexity, on the other. It is also argued that IST-IA is a promising solution not only for training but also for the tracking phase, if applicable.**

I. Introduction

Interference management is a critical issue in next generation cellular networks. To address this problem, a solution that relies on base station (BS) cooperation was first proposed in [2], and then further studied by a large number of authors (see review in [3]). In this class of strategies, BS cooperation is realized via a high speed backbone network connecting the BSs to a central unit for the purpose of exchanging channel state information. Recently, Cadambe and Jafar proposed a different approach to interference management to which they referred as *interference alignment* (IA), and which does not require deployment of a backbone network [4]. IA is based on the idea that interfering signals emitted by different sources can be designed so as to "cast overlapping shadows" on the signal subspace of interfered receivers. This approach, by beamforming across multiple antennas when IA is constructed in space (IA could also be built based on the dimensions of time, frequency and code), leaves part of the signal space at the receiver free from interference. It is noted that IA requires global channel knowledge at all the transmitters, which is hard to realize. To alleviate this problem, an iterative distributed algorithm in [1] (referred to here as Distributed IA) is proposed based on the *network duality property* [5], which is provided by the reciprocity of the channel. The algorithm exploits the duality property that the signal subspace with the least interference for a certain receiver is also the subspace that causes the least interference to other users during the reciprocal network transmission. The algorithm works by iteratively adapting the beamforming matrices at both the transmitter and the receiver sides (alternatively, user equipments (UEs) or BSs), in order to monotonically decrease the total interference in the system. At each iteration, each user (UE or BS) attempts to find the subspace over which it observes the least interference. This operation, by duality, translates in producing the least interference to the receivers. The Distributed IA algorithm in [1] requires local channel knowledge as well as knowledge of the interference covariance matrix, which consists of the complex additive white Gaussian noise (AWGN) and the interference from all other users.

In this work, we propose the interference subspace tracking IA (IST-IA), a distributed implementation of IA based on [1] that accounts for the necessary training to estimate the beamforming matrices, without assuming *a priori* knowledge of the interference covariance matrices. We investigate the training overhead and propose a training strategy based on spatial reuse that reduces the overhead with minor performance degradation. The proposed strategy updates the beamforming matrices for both BS and UE based on the *minor subspace tracking* (MST) technique [6].

The rest of the paper is organized as follows. In the next section, the system model for the IST-IA is introduced. In Section III, we briefly review the Distributed IA algorithm of [1], then focus on the IST-IA algorithm, which is described in detail including the analysis of the system overhead and the case for spatial reuse. Numerical results are then provided in Section IV to validate the proposed technique. Concluding remarks are offered in Section V.

Notation: Throughout the paper, we denote matrices and vectors with bold face type, using capital letters for matrices and lower case letters for vectors. For any matrix $\mathbf{A}$, the superscript "T" denotes transpose and "$\dagger$" complex conjugate transpose. Finally, $\mathbf{I}_d$ represents the $d \times d$ identity matrix.

II. System Model

We consider a linear cellular network as shown in Fig. 1, where one UE is active at any given time in any cell [2]. Each BS, besides the received signal from the UE in its own cell, receives inter-cell interference from all other BSs' transmissions. We assume that d_{ki} $(i, k = 1, \cdots, K)$ is the

This work was supported in part by National Science Foundation (NSF) under CNS-0626611.

distance from the UE in the i-th cell to the k-th BS, and d_B is the distance between two adjacent BSs. The distance-dependent path-loss gain between the k-th BS and the i-th UE is $g_{ki} = \left(\frac{d_{ii}}{d_{ki}}\right)^{-\alpha}$, where α is the propagation exponent. We assume that all the UEs and BSs are equipped with M and N multiple antennas, respectively. All UE channels in the network experience independent and identically distributed block Rayleigh fading, i.e., the fading channel gain is assumed to be constant during training and data transmission phases. Therefore, for the uplink, the received signal at the k-th BS is:

$$\mathbf{y}^{[k]} = \sum_{i=1}^{K} g_{ki}\mathbf{H}^{[ki]}\mathbf{x}^{[i]} + \mathbf{w}^{[k]}, \quad (1)$$

where $\mathbf{H}^{[ki]}$ is the $N \times M$ matrix of fading channel coefficients between the i-th UE and k-th BS (every entry is i.i.d. $\mathcal{CN}(0,1)$); $\mathbf{y}^{[k]}$ is the $N \times 1$ received signal vector, and $\mathbf{w}^{[k]}$ is AWGN noise with zero mean and unit variance; $\mathbf{x}^{[i]}$ is the $M \times 1$ transmit Gaussian distributed signal vector by UE i, with transmit power $E\|\mathbf{x}^{[i]}\|^2 = P$. Note that the fading channel coefficient matrix $\mathbf{H}^{[ki]}$ and the path-loss gain g_{ki} do not have to be explicitly known at neither the transmitter nor receiver side. This is explained with more details in Sec. III. For the downlink transmission, we assume that the channel is reciprocal to the uplink channel, as we are focusing on a Time Division Duplex (TDD) system. The received signal at the i-th UE is

$$\mathbf{z}^{[i]} = \sum_{k=1}^{K} g_{ki}(\mathbf{H}^{[ki]})^T\bar{\mathbf{x}}^{[k]} + \bar{\mathbf{w}}^{[i]}, \quad (2)$$

where $\mathbf{z}^{[i]}$ and $\bar{\mathbf{w}}^{[i]}$ are respectively the $N \times 1$ received signal vector and the AWGN noise with zero mean and unit variance; $\bar{\mathbf{x}}^{[k]}$ is the $N \times 1$ transmit Gaussian distributed signal vector by BS k, with transmit power $E\|\bar{\mathbf{x}}^{[k]}\|^2 = \overline{P}$.

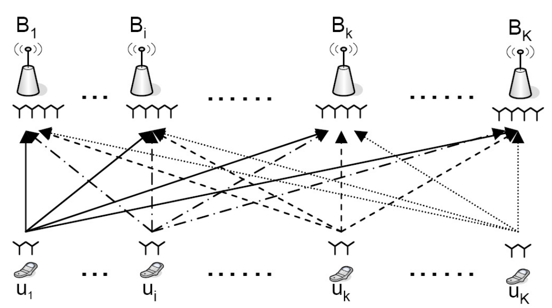

Fig. 1. A distance dependent path-loss model of a cellular network

III. Interference Subspace Tracking for IA (IST-IA)

In this section, we firstly review the Distributed IA algorithm of [1], then we detail a proposed implementation of Distributed IA that does not assume *a priori* knowledge of the interference covariance matrix (5). The algorithm is based on the MST algorithm [6], and it updates the precoding and interference matrices.

A. Distributed IA: A Brief Review

The basic idea behind the Distributed IA scheme of [1] is that, in a K users' interference network (which corresponds to K cells in our model), each user, rather than trying to maximize its own transmission signal-to-interference-and-noise ratio (SINR) at the desired receiver ("selfish" transmission strategy), chooses its transmission strategy so as to cause the least interference to other concurrent transmissions ("altruistic" transmission strategy). By network duality, this equals to find the subspace over which it observes the least interference when as a receiver. More specifically, the Distributed IA algorithm starts with a training phase which operates in T iterations. In each of the iteration, we have forward and backward transmissions. During the first forward transmission (say, from UEs to BSs), the k-th UE, $k = 1, \cdots, K$, precodes D independent symbols $\hat{x}_d^{[k]}, d = 1, \cdots, D$, at the transmitter side with an arbitrary $M \times D$ precoding matrix $\mathbf{V}^{[k]}$. Each of the symbols $\hat{x}_d$ is a zero-mean complex Gaussian random variable with power $\frac{P}{D}$, so that the transmit signal vector $\mathbf{x}^{[k]}$ in (1) is

$$\mathbf{x}^{[k]} = \mathbf{V}^{[k]}\hat{\mathbf{x}}^{[k]}, \quad (3)$$

where $\hat{\mathbf{x}}^{[k]} = [\hat{x}_1^{[k]}, \hat{x}_2^{[k]}, \cdots \hat{x}_D^{[k]}]^T$. At the receiver side, an $N \times D$ interference suppression matrix $\mathbf{U}^{[k]}$ is applied at the k-th receiver, so that the signal after interference suppression is given by,

$$\hat{\mathbf{y}}^{[k]} = \mathbf{U}^{[k]\dagger}\mathbf{y}^{[k]}. \quad (4)$$

The interference suppression matrix $\mathbf{U}^{[k]}$ is chosen in a way such that the signal $\hat{\mathbf{y}}^{[k]}$ lies in the D-dimensional subspace with the least interference. In order to accomplish this, the k-th receiver (BS) is assumed to know the interference covariance matrix (including the AWGN)

$$\mathbf{Q}^{[k]} = \sum_{i=1, i\neq k}^{K} \frac{P}{D}\mathbf{H}^{[ki]}\mathbf{V}^{[i]}\mathbf{V}^{[i]\dagger}\mathbf{H}^{[ki]\dagger} + \mathbf{I}_N, \quad (5)$$

so that the interference suppression matrix $\mathbf{U}^{[k]}$ is chosen as the D eigenvectors corresponding to the smallest D eigenvalues of $\mathbf{Q}^{[k]}$. Similar steps are then used in the backward transmission (from BSs to UEs), where the transmitters and receivers switch roles. Specifically, denoting as $\bar{\mathbf{U}}^{[k]}$ and $\bar{\mathbf{V}}^{[k]}$ the precoding and interference suppression matrices of the backward transmission, respectively, the transmitters (BSs) set $\bar{\mathbf{V}}^{[k]} = \mathbf{U}^{[k]}$ with $\mathbf{U}^{[k]}$ calculated in the forward phase. After obtaining the interference covariance matrix $\bar{\mathbf{Q}}^{[k]}$, the interference suppression matrix $\bar{\mathbf{U}}^{[k]}$ is updated with the D eigenvectors corresponding to the smallest D eigenvalues of $\bar{\mathbf{Q}}^{[k]}$, and we then set $\mathbf{V}^{[k]} = \bar{\mathbf{U}}^{[k]}$. The algorithm iterates forward and backward transmissions, updating the precoding matrix $\mathbf{U}^{[k]}$ and interference suppression matrix $\mathbf{V}^{[k]}$ for T times. The total interference in the system monotonically decreases, and the convergence of the algorithm can be proved as shown in [1].

B. Operation of IST-IA

Similarly to [1], the IST-IA algorithm starts with a training phase consisting of T iterations to update the precoding and interference suppression matrices. However, instead of *a priori* knowledge of the interference covariance matrix (5), here we consider in a practical cellular system the training structure in Fig. 2 to enable estimation of the beamforming matrices via KL training symbol periods for each forward or backward iteration.

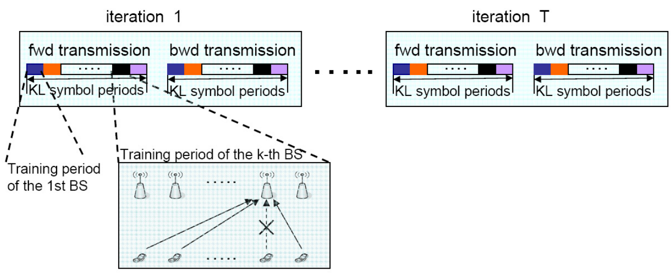

Fig. 2. Training phase structure for IST-IA

In the forward transmission of the first iteration, training symbols $\hat{\mathbf{x}}^{[k]}$ of dimension $D \times 1$ are sent by UEs in the network by using an arbitrary precoding matrix $\mathbf{V}^{[k]}$, as shown in (3). These symbols are used by BSs to update the interference suppression matrix $\mathbf{U}^{[k]}$, $k = 1, \cdots, K$. Specifically, for training of the first BS, the UEs in all other cells transmit L symbols, while the direct transmission from the UE in the first cell is silenced, i.e., in Fig. 1, the network firstly schedules concurrent transmissions from $U_2, U_3, \cdots, U_k$, to B_1, and shuts off the transmission from U_1 to B_1. After B_1 receives the L training signal vectors of the form (1), the network schedules the transmissions from $U_1, U_3, \cdots, U_K$ to B_2 for training of B_2. The procedure continues until every BS receives L training signal vectors from all the UEs of other cells (see Fig. 2 for illustration). The l-th $(l = 1, \cdots, L)$ symbol received by the k-th BS is (recall (1))

$$\mathbf{y}^{[k]}(l) = \sum_{i=1, i\neq k}^{K} g_{ki}\mathbf{H}^{[ki]}\mathbf{x}^{[i]}(l) + \mathbf{w}^{[k]}(l). \tag{6}$$

Different algorithms can then be used to update the interference suppression matrix $\mathbf{U}^{[k]}$. A simple approach would be to estimate the covariance matrix (5) from the L received training vector and then obtain $\mathbf{U}^{[k]}$ via eigenvector decomposition (EVD). We term this scheme EVD-IA. In this paper, we propose an alternative solution, termed IST-IA, which is based on the MST algorithm discussed in Sec. IV-C. Compared to EVD-IA, IST-IA significantly reduces the computational complexity, as discussed in Sec. IV-D, with minor performance degradation. Moreover, IST-IA could be easily deployed for tracking the beamforming matrices when the channel is a time-varying one (e.g., in a decision-directed fashion) during data transmission, following the training phase. Here, we focus on the advantage brought by the IST-IA in the training phase for lack of space, and leave the analysis of the tracking phase to future work. After the BSs have updated their interference suppression matrices $\mathbf{U}^{[k]}$ via MST, downlink transmissions, i.e., backward transmissions from BSs to UEs take place. Each BS precodes D independent symbols with the precoding matrix $\bar{\mathbf{V}}^{[k]} = \mathbf{U}^{[k]}$. Similar to the uplink, each UE receives L interference signal vectors and then performs the MST algorithm to calculate the interference suppression matrix $\bar{\mathbf{U}}^{[k]}$, and updates $\mathbf{V}^{[k]} = \bar{\mathbf{U}}^{[k]}$. The training period continues for the prescribed T training iterations.

After the training period, the equivalent channel $\mathbf{H}_{eq}^{[ki]}$ between the k-th BS and the i-th UE is

$$\mathbf{H}_{eq}^{[ki]} = g_{ki}\mathbf{U}^{[k]\dagger}\mathbf{H}^{[ki]}\mathbf{V}^{[i]}. \tag{7}$$

The k-th BS estimates the equivalent channel $\mathbf{H}_{eq}^{[ki]}$ and the UEs start data transmission. Notice that here we focus on uplink transmission, but the algorithm applies to downlink as well.

In order to evaluate the system performance in terms of rate, the SINR of the k-th BS is calculated as

$$\text{SINR}^{[k]} = \frac{P}{D}\mathbf{H}_{eq}^{[ki]}\mathbf{H}_{eq}^{[ki]\dagger}\left(\mathbf{U}^{[k]\dagger}\mathbf{Q}^{[k]}\mathbf{U}^{[k]}\right)^{-1}, \tag{8}$$

and the corresponding rate is measured as $\log(1 + \text{SINR}^{[k]})$. Notice that in calculating this rate we assume that the receivers have perfect knowledge of the equivalent channel (7) and the final interference covariance matrix (5). These assumptions are made to simplify the analysis but could be easily waived by introducing appropriate estimation noise terms.

The IST-IA algorithm is summarized in Algorithm 1 below.

Algorithm 1 IST-IA

Training Period:
Initialize $\mathbf{U}^{[k]}, \mathbf{V}^{[k]}$ $(k = 1 \cdots K)$: random matrix with orthonormal vectors;
for $t = 1$ to T (IST-IA training iterations) **do**
  Uplink: UE→BS
  for $k = 1$ to K (the k-th BS) **do**
    for $l = 1$ to L **do**
      for $i = 1$ to K, $i \neq k$ (the i-th UE) **do**
        generate transmission vector $\mathbf{x}^{[i]}(l) = \mathbf{V}^{[i]}\hat{\mathbf{x}}^{[i]}(l)$
      end for
      the k-th BS receives concurrent interference signal transmitted from the $K-1$ UEs with $i \neq k$:
      $\mathbf{y}^{[k]}(l) = \sum_{i=1, i\neq k}^{K} g_{ki}\mathbf{H}^{[ki]}(l)\mathbf{x}^{[i]}(l) + \mathbf{w}^{[k]}(l)$
    end for
    Find $\mathbf{U}_t^{[k]}(L)$ by MST in Algorithm 2
  end for
  Downlink: BS→UE, dual to the uplink part: obtain $\mathbf{V}_t^{[k]}(L)$.
end for
Update $\mathbf{U}^{[k]} = \mathbf{U}_T^{[k]}(L)$, $\mathbf{V}^{[k]} = \mathbf{V}_T^{[k]}(L)$
Equivalent Channel Measurement:
$\mathbf{H}_{eq}^{[ki]} = g_{ki}\mathbf{U}^{[k]\dagger}\mathbf{H}^{[ki]}\mathbf{V}^{[i]}$
DATA Transmission Period

C. MST

In this subsection, we consider the problem of tracking the eigenvectors corresponding to the smallest D eigenvalues of the empirical covariance matrix obtained from L received vectors. A number of efficient MST algorithms exist for this purpose [6] [7] [8]. In this paper, we focus on the square-root QR inverse iteration minor subspace tracking algorithm of [6], which provides a good tradeoff between complexity and performance. The algorithm is summarized in Algorithm 2 using the notation of this paper.

Algorithm 2 Minor Subspace Tracking Algorithm

Initialize $\mathbf{U}^{[k]}$: random matrix with orthonormal vectors; $\mathbf{R}_{\mathbf{x}}(0) = \Delta \cdot \mathbf{I}_N$, where Δ is a small positive real number.
for $l = 1$ to L **do**
  Input: $\mathbf{y}^{[k]}(l)$
  Find $\mathbf{R}_{\mathbf{x}}(l)$ ($\mathbf{R}_{\mathbf{x}}(l)$ needs to be a upper-right triangular matrix) such that

$$\begin{bmatrix} 0 \cdots 0 \\ \mathbf{R}_{\mathbf{x}}(l) \end{bmatrix} = \mathbf{G}(l) \begin{bmatrix} (1-\beta)^{1/2} \mathbf{y}^{[k]\dagger}(l) \\ \beta^{1/2} \mathbf{R}_{\mathbf{x}}(l-1) \end{bmatrix},$$

  where $\mathbf{G}(l)$ is a $(N+1) \times (N+1)$ unitary matrix.
  Find $\mathbf{B}(l)$ such that $\mathbf{R}_{\mathbf{x}}^{\dagger}(l)\mathbf{B}(l) = \mathbf{U}^{[k]}(l-1)$
  Find $\mathbf{A}(l)$ such that $\mathbf{R}_{\mathbf{x}}(l)\mathbf{A}(l) = \mathbf{B}(l)$
  $\mathbf{A}(l) = \mathbf{U}^{[k]}(l)\mathbf{R}(l)$, $N \times D$ QR decomposition
end for
Output: $\mathbf{U}^{[k]}(L)$

D. Overhead and Computational Complexity Analysis

The IST-IA algorithm needs T iterations in the training phase, each of which includes both forward and backward transmissions. During each of the forward and backward transmission, KL training symbol periods are needed. Therefore, the training phase consists of an overhead $t_o = 2KLT$ training symbol periods.

It is pointed out in [6] that the computation complexity of MST is $\mathcal{O}(N^2 D)$, where N is the number of receive antennas. Thus assuming $M = N$ for simplicity, the IST-IA needs $\mathcal{O}(N^2 DKLT)$ operations for the training period. On the other hand, EVD-IA involvs the eigen-decomposition of the covariance matrix, which requires $\mathcal{O}(N^3 + N^2 D)$ operations [9]. Thus compared to EVD-IA, which needs a total of $\mathcal{O}\left((N^3 + N^2 D)KLT\right)$ operations, IST-IA has much less computational complexity, especially when the number of receive antennas is large.

E. Spatial Reuse IST-IA

In order to further reduce the training symbols with minor performance degradation, a form of spatial reuse can be introduced in the IST-IA algorithm. Unlike the standard spatial reuse in which several spatially separated cells transmit at the same time or bandwidth, while adjacent cells are silent, here several spatially separated cells shut off their transmission at the transmitter side while all the other cells' transmitters send the training signals. The receivers of such spatially separated cells receive the L training transmissions and estimate their interference suppression matrices $\mathbf{U}^{[k]}$ using MST. We define as γ the spacing between two silent cells ($\gamma = K$ for the basic IST-IA algorithm described in Algorithm 1). Decreasing γ reduces the complexity order to $\mathcal{O}(N^2 D\gamma LT)$ since the number of training symbols becomes $t_{o,s} = 2\gamma LT$. There is a tradeoff between the spatial-reuse factor γ (and thus the computational complexity) and the performance of IST-IA, as illustrated in the next section.

IV. Numerical Results

We consider at first the same experimental conditions of [1] by looking at a three-users interference channel where each node is equipped with two antennas ($M = N = 2$) and all channel coefficients are i.i.d zero-mean unit variance circularly complex Gaussian, without considering path-loss. For simplicity, we set $P = \bar{P}$. We also set $D = 1$. It can be seen from Fig. 3 that both the Distributed IA of [1] and IST-IA perform significantly better than the time-sharing orthogonal scheme (every link transmits in a TDMA fashion and with transmit power equals to $3P$) and the isotropic transmission scheme (each transmitter sends $D = M = N = 2$ streams of equal power without regard to the channel information). The degradation between IST-IA with $T = 50$, $L = 20$ and Distributed IA with $T = 50$ is minor. We also show as reference the curve for EVD-IA. The performance of EVD-IA with $T = 50$ and $L = 20$ is almost the same as IST-IA.

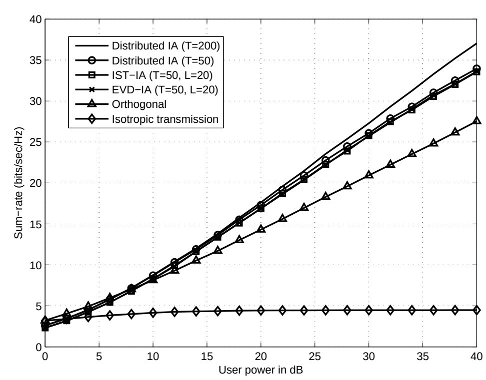

Fig. 3. Sum-rate versus user power P ($K = 3, M = N = 2, P = \bar{P}, \beta = 0.993$).

We then investigate a system with $K = 7$ cells, $N = 4$ and $M = 2$. Path-loss is considered, and we choose the path-loss propogation exponent $\alpha = 3$. We further assume that the ratio between the distance of the BS and the UE in the same cell d_{ii}, and the distance of two adjacent BSs d_B is $\frac{3}{4}$. We examine the effect on the system sum-rate due to the number of transmission iterations T and the number of received signal vectors L. In Fig. 4, we show the curves for different combinations of $T = 2, 50$ and $L = 2, 50$. Reference curves for the Distributed IA with $T = 2, 50$ are also shown. It is seen that, fixing $TL = 100$, it is more convenient to select a sufficiently large L (say $L = 50$), rather than a large T (say $T = 50$). The above results indicates that we should put more emphasis on the training symbols L when designing a real system (see Fig. 5 for another example).

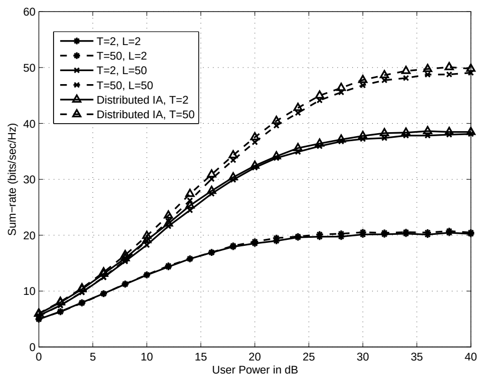

Fig. 4. Sum-rate versus user power P for IST-IA ($K = 7, M = 2, N = 4, D = 1, \frac{d_{ii}}{d_B} = 3/4, P = \bar{P}, \beta = 0.993, \alpha = 3$).

Following the previous example, we compare in Fig. 5 the sum-rate performance under the constraint of a fixed TL, namely, $TL = 20$ (the total overhead of the system for this case is $t_o = 2KTL = 280$ training symbol periods), for different T and L ($T = 10, L = 2$; $T = 4, L = 5$ and $T = 2, L = 10$). The results confirms that the best sum-rate is achieved when one select L to be large enough ($L = 10, T = 2$).

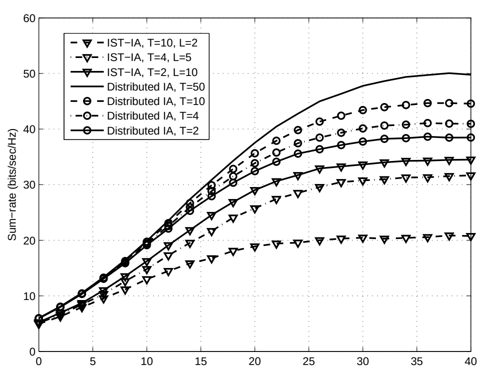

Fig. 5. Sum-rate versus user power P for IST-IA with $TL = 20$ ($K = 7, M = 2, N = 4, D = 1, \frac{d_{ii}}{d_B} = 3/4, P = \bar{P}, \beta = 0.993, \alpha = 3$).

Next, we demonstrate in Fig. 6 the performance of the proposed spatial-reuse scheme. We choose $T = 2$, $L = 10$ and compare the sum-rate of the IST-IA for different spatial reuse factors $\gamma = 2, 3$ and with no spatial reuse ($\gamma = K = 7$). The curves of Distributed IA in [1] for different γ and no spatial reuse cases are also generated here for reference. We see from the figure that the sum-rate performance of IST-IA with $\gamma = 3$ is very close to the case of no spatial reuse (1 bits/sec/Hz difference at high SNR), while the overhead of the system is drastically reduced from $t_o = 280$ to $t_{o,s} = 2TL\gamma = 120$.

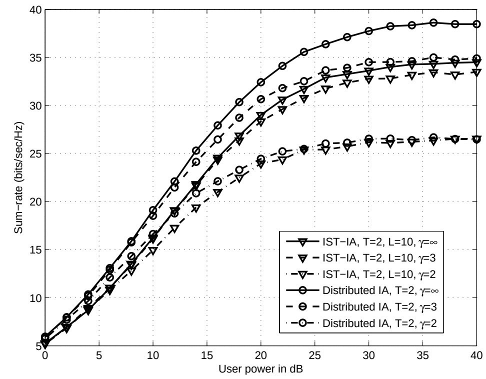

Fig. 6. Sum-rate versus user power P for IST-IA and Distributed IA with spatial reuse ($K = 7, M = 2, N = 4, D = 1, \frac{d_{ii}}{d_B} = 3/4, P = \bar{P}, \beta = 0.993, \alpha = 3$).

V. Concluding Remarks

In this paper, we propose the IST-IA algorithm as a practical solution that can be implemented in a cellular system to reduce inter-cell interference and increase network capacity. Compared to the Distributed IA algorithm of [1], IST-IA does not need *a priori* knowledge of the interference covariance matrix at each of the BS or UE, as well as limited channel state information knowledge. Based on a few received signal vectors, the IST-IA algorithm can track the beamforming matrices that are used at BS and UE, and provides a similar performance to the Distributed IA algorithm with much less system overhead and computational complexity. Future work is needed to fully assess the role of the proposed solution over mobile fading channels.

References

[1] K. S. Gomadam, V. R. Cadambe, and S. A. Jafar, “Approaching the capacity of wireless networks through distributed interference alignment,” *arXiv*, no. 0803.3816, 2008.

[2] A. D. Wyner, “Shannon-theoretic approach to a gaussian cellular multiple-accesschannel,” *IEEE Trans. Inform. Theory*, vol. 40, no. 6, pp. 1713–1727, Nov. 1994.

[3] O. Somekh, O. Simeone, Y. Bar-Ness, A. M. Haimovich, U. Spagnolini, and S. S. (Shitz), *Distributed Antenna Systems: Open Architecture for Future Wireless Communications, ch. An Information Theoretic View of Distributed Antenna Processing in Cellular Systems*. Auerbach Publications, CRC Press, May 2007.

[4] V. R. Cadambe and S. A. Jafar, “Interference alignment and the degrees of freedom for the K user interference channel,” *IEEE Trans. Inform. Theory*, vol. 54, no. 8, pp. 3425–3441, Aug. 2008.

[5] B. Babadi and V. Tarokh, “A distributed dynamic frequency allocation algorthim for ad hoc networks,” *arXiv*, no. 0711.3247, 2007.

[6] P. Strobach, “Square-root QR inverse iteration for tracking the minor subspace,” *IEEE Trans. Signal Processing*, vol. 48, no. 11, pp. 2994–2999, Nov. 2000.

[7] X. Doukopoulos and G. Moustakides, “Fast and stable subspace tracking,” *IEEE Trans. Signal Processing*, vol. 56, no. 1, pp. 1452–1465, Apr. 2008.

[8] R. Badeau, G. Richard, and B. David, “Fast and stable YAST algorithm for principal and minor subspace tracking,” *IEEE Trans. Signal Processing*, vol. 56, no. 8, pp. 3437–3446, Aug. 2008.

[9] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Johns Hopkins Univ. Press, Oct. 1996.