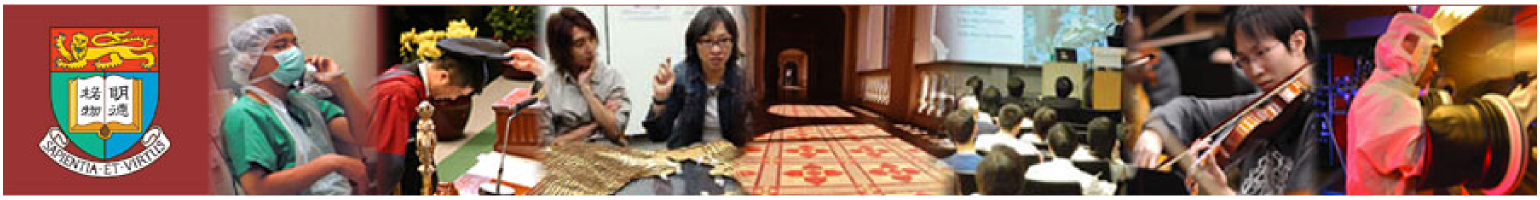

Title	Distributed CFOs Estimation and Compensation in Multi-cell Cooperative Networks
Author(s)	Du, J; Wu, YC
Citation	International Conference on Information and Communication Technology Convergence (ICTC), Jeju, South Korea, 14-16 October 2013. In International Conference on Information and Communication Technology Convergence, 2013, p. 117-121, article no. 6675320
Issued Date	2013
URL	http://hdl.handle.net/10722/199394
Rights	International Conference on Information and Communication Technology Convergence. Copyright © I E E E.

Distributed CFOs Estimation and Compensation in Multi-cell Cooperative Networks

Jian Du and Yik-Chung Wu
Department of EEE, The University of Hong Kong, Pokfulam Road, Hong Kong
Email:{dujian, ycwu}@eee.hku.hk

***Abstract*—In this paper, we propose a fully distributed algorithm for frequency offsets estimation in multi-cell cooperative networks. The idea is based on belief propagation, resulting in that each base station or mobile user estimates its own frequency offsets by local computations and limited exchange of information with its direct neighbors in the cellular network. Such algorithm does not require any centralized information processing or knowledge of global network topology, thus is scalable with network size. Simulation results demonstrate the fast convergence of the algorithm and show that estimation mean-squared-error at each node touches the centralized Cramér-Rao bound within a few iterations of message exchange.**

I. Introduction

In traditional cellular systems, geographical area is divided into cells and a base station is dedicated to serve users within each cell. Frequency reuse pattern that forbids adjacent cells using the same frequency is adopted to avoid excessive intercell interference. Unfortunately, frequency reuse also leads to the fact that each cell is only using a small portion of the whole system bandwidth. Recent breakthrough in multi-cell cooperative networks allows fully frequency reuse among the cells. Despite different users interfere with each other, multiple base stations could coordinate their coding and decoding. It was shown that such joint-processing significantly outperforms a network with individual cell processing [1]–[3].

Since frequencies synthesized from independent oscillators will be different from each other due to variation of oscillator circuits, frequency offset exists at each antenna in the cellular system, as shown in Fig. 1. Multi-cell cooperation requires frequency synchronization over the whole cellular network, otherwise there would be capacity degradation [4], offsetting the benefits of cooperation. Despite the relative CFO between each base station and its users can be optimally estimated by existing methods [5]–[12], network-wide CFOs correction is difficult since each base station needs to synchronize with multiple users with different relative CFOs at the same time. Making the problem more challenging is the fact that synchronization should be accomplished by local operations without knowing the global network structure since users move around and join different parts of the network randomly.

Pioneering works for multi-cell CFOs correction have been proposed in [4]. By gathering all the information in a central processing unit, CFOs are estimated at the receiver and then fedback to corresponding transmitters to adjust the offsets. However, this method is centralized, and is not suitable for large-scale network.

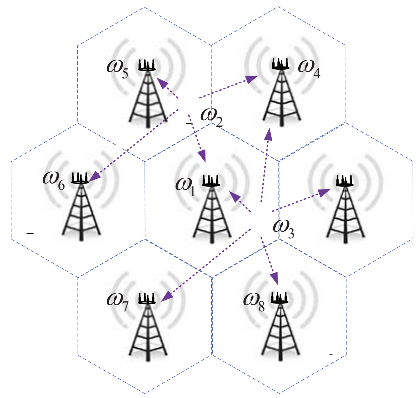

Fig. 1. Multi-cell cooperative networks.

In this paper, we propose a network-wide fully distributed CFOs estimation and compensation method which only involves local processing and information exchange between direct neighbors. The frequency offset of each oscillator is estimated and corrected locally by the base station or mobile user. After synchronization, the mean-square-error (MSE) for each frequency offset approaches the corresponding Cramér-Rao bound (CRB) asymptotically. Moreover, the proposed algorithm is scalable with network size, and robust to topology changes.

The following notations are used throughout this paper. Boldface uppercase and lowercase letters will be used for matrices and vectors, respectively. Superscripts H and T denote Hermitian and transpose, respectively. The symbol $\boldsymbol{I}_N$ represents the $N \times N$ identity matrix, while $\mathbf{1}_K$ is an all one K dimensional vector. The symbol $\otimes$ denotes the Kronecker product and $\odot$ denotes the Hadamard product. Notation $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{R})$ stands for the probability density function (pdf) of a Gaussian random vector $\boldsymbol{x}$ with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{R}$. The symbol $\propto$ represents the linear scalar relationship between two real valued functions. $\text{diag}\{[a_1, \ldots, a_N]\}$ corresponds to an $N \times N$ diagonal matrix with diagonal components a_1 through a_N, while $\text{blkdiag}\{[\boldsymbol{A}_1, \ldots, \boldsymbol{A}_N]\}$ corresponds to a block diagonal matrix with $\boldsymbol{A}_1$ through $\boldsymbol{A}_N$ as diagonal blocks.

II. System Model

We consider a general network consisting of K nodes (where each node could be a base station or a mobile user)

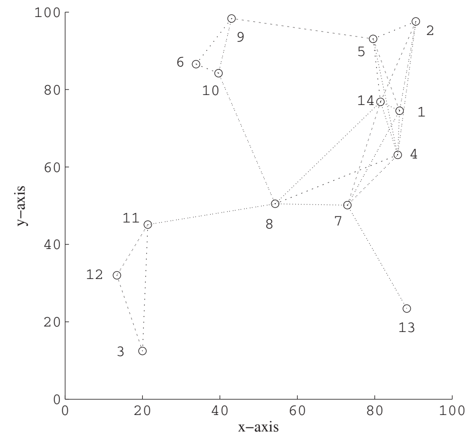

Fig. 2. An example of a network topology with 14 nodes.

distributed in a field as shown in Fig. 2. The topology of the network is described by a communication graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of order K, where $\mathcal{V} = \{1, \ldots, K\}$ is the set of graph vertexes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of graph edges. In the example shown in Fig. 2, the vertices are depicted by circles and the edges by lines connecting these circles. The neighborhood of node i is the set of nodes $\mathcal{I}(i) \subset \mathcal{V}$ defined as $\mathcal{I}(i) \triangleq \{j \in \mathcal{V} | \{i, j\} \in \mathcal{E}\}$, i.e., those nodes that are connected via a direct communication link to node i. It is also assumed that any two distinct nodes can communicate with each other through finite hops, such graph is named strongly connected graph.

In general, relative CFOs exist between any pair of neighboring nodes, and can be estimated by traditional CFOs estimation methods. Let nodes i and j equipped with N_i and N_j antennas, respectively. Denote the frequency offsets[1] (with respect to a reference frequency) of the q^{th} antenna on node i as ω_q^i, while that of k^{th} antenna of node j as ω_k^j. Then, the relative CFO between the q^{th} and k^{th} antenna of nodes i and j is $\epsilon_{q,k}^{i,j} \triangleq \omega_q^i - \omega_k^j$. Here we consider the general case where each antenna can be associated with separate oscillator circuit. Therefore, for the Multiple Input Multiple Output (MIMO) system between node i and node j, there are $N_i N_j$ relative CFOs denoted as $\boldsymbol{\epsilon}^{i,j} \triangleq [\epsilon_{1,1}^{i,j}, \ldots, \epsilon_{N_i,1}^{i,j}, \ldots, \epsilon_{1,N_j}^{i,j}, \ldots, \epsilon_{N_i,N_j}^{i,j}]^T$. Such relative CFOs estimation in MIMO systems can be decomposed into N_j parallel Multiple Input Single Output (MISO) CFOs estimation problem [7]. For example, considering a flat-fading MISO system, for the k^{th} receive antenna of node j, the received signal can be written as

$$y_k^{i,j}(t) = \sum_{q=1}^{N_i} h_{q,k}^{i,j} e^{\jmath \epsilon_{q,k}^{i,j} t} z_q^i(t) + \xi_k^j(t), \quad t = 1, \ldots, N, \quad (1)$$

where $h_{q,k}^{i,j}$ is the unknown channel gain between the q^{th} antenna of node i and k^{th} antenna of node j; $\jmath \triangleq \sqrt{-1}$; $\{z_q^i(t)\}_{t=1}^N$ is the training sequence transmitted from the q^{th} antennas of node i; and $\xi_k^j(t)$ is the observation noise at the k^{th} antenna of node j. By stacking (1) with $t = 1, \ldots, N$ in vector form and omitting superscript i, j without confusion, the received vector $\boldsymbol{y}_k \triangleq [y_k(1), \ldots, y_k(N)]^T$ can be written as

$$\boldsymbol{y}_k = \boldsymbol{\Gamma}_k(\boldsymbol{\epsilon}_k) \odot \boldsymbol{Z}_k \boldsymbol{h}_k + \boldsymbol{\xi}_k \quad k = 1, \ldots, N_j, \quad (2)$$

where $\boldsymbol{\Gamma}_k(\boldsymbol{\epsilon}_k)$ is an N-by-N_i Vandermonde matrix with its t^{th} row given by $[e^{\jmath t \epsilon_{1,k}}, e^{\jmath t \epsilon_{2,k}}, \cdots, e^{\jmath t \epsilon_{N_i,k}}]$; $\boldsymbol{Z}_k$ is the N-by-N_i training sequence matrix with its t^{th} row $[z_1(t), z_2(t), \cdots, z_{N_i}(t)]$; and $\boldsymbol{\xi}_k = [\xi_k(1), \ldots, \xi_k(N)]^T$ is the observation noise. The parameters $\boldsymbol{\epsilon}_k \triangleq [\epsilon_{1,k}, \epsilon_{2,k}, \ldots, \epsilon_{N_i,k}]^T$ and $\boldsymbol{h}_k \triangleq [h_{1,k}, \ldots, h_{N_i,k}]^T$ are the parameters need to be estimated.

If the noise is white and Gaussian, i.e., $\boldsymbol{\xi}_k \sim \mathcal{CN}(\boldsymbol{\xi}_k; \boldsymbol{0}, \sigma_k^2 \boldsymbol{I}_N)$, joint relative CFOs and channels estimation have been extensively studied in the past two decades and the optimal estimates $\hat{\boldsymbol{\epsilon}}_k$ and $\hat{\boldsymbol{h}}_k$ have been proposed in [6]–[10], with the MSEs approaching the corresponding CRBs in medium and high signal-to-noise ratio (SNR) ranges. From (2), the CRB of $\boldsymbol{\epsilon}_k$ can be shown to be [7]

$$\boldsymbol{B}_{\boldsymbol{\epsilon}_k}(\boldsymbol{\epsilon}_k, \boldsymbol{h}_k) = \frac{\sigma_k^2}{2} \left\{ \mathrm{Re}[\boldsymbol{V}_k - \boldsymbol{T}_k^H (\boldsymbol{\Lambda}_k^H \boldsymbol{\Lambda}_k)^{-1} \boldsymbol{T}_k] \right\}^{-1}, \quad (3)$$

where $\boldsymbol{V}_k \triangleq \mathrm{diag}\{\boldsymbol{h}_k\} \boldsymbol{\Lambda}_k^H \boldsymbol{D}^2 \boldsymbol{\Lambda}_k \mathrm{diag}\{\boldsymbol{h}_k\}$, $\boldsymbol{T}_k \triangleq \boldsymbol{\Lambda}_k^H \boldsymbol{D} \boldsymbol{\Lambda}_k \mathrm{diag}\{\boldsymbol{h}_k\}$, with $\boldsymbol{\Lambda}_k \triangleq \boldsymbol{\Gamma}_k(\boldsymbol{\epsilon}_k) \odot \boldsymbol{Z}_k$ and $\boldsymbol{D} \triangleq \mathrm{diag}\{[1, 2, \ldots, N]\}$. Since there are N_j independent MISO estimation problems as in (2), the CR-B for frequency estimation in MIMO system between n-odes i and j is given by $\boldsymbol{B}_{\boldsymbol{\epsilon}}^{\{i,j\}}(\{\boldsymbol{\epsilon}_k\}_{k=1}^{N_j}, \{\boldsymbol{h}_k\}_{k=1}^{N_j}) = \mathrm{blkdiag}\{[\boldsymbol{B}_{\boldsymbol{\epsilon}_1}(\boldsymbol{\epsilon}_1, \boldsymbol{h}_1), \ldots, \boldsymbol{B}_{\boldsymbol{\epsilon}_{N_j}}(\boldsymbol{\epsilon}_{N_j}, \boldsymbol{h}_{N_j})]\}$.

After joint estimation of relative CFOs and channels, the relative CFOs between nodes i and j can be obtained as

$$\boldsymbol{r}_{i,j} = \boldsymbol{A}_{i,j} \boldsymbol{\omega}_i + \boldsymbol{A}_{j,i} \boldsymbol{\omega}_j + \boldsymbol{n}_{i,j}, \quad (4)$$

where $\boldsymbol{r}_{i,j} \triangleq [\hat{\boldsymbol{\epsilon}}_1^T, \hat{\boldsymbol{\epsilon}}_2^T, \ldots, \hat{\boldsymbol{\epsilon}}_{N_j}^T]^T$ are the $N_i N_j$ relative CFOs estimates; $\boldsymbol{A}_{i,j} \triangleq \boldsymbol{I}_{N_i} \otimes \boldsymbol{1}_{N_j}$ and $\boldsymbol{A}_{j,i} \triangleq -\boldsymbol{1}_{N_i} \otimes \boldsymbol{I}_{N_j}$; and $\boldsymbol{n}_{i,j}$ is the estimation error. It is known that for the maximum likelihood (ML) estimates, $\boldsymbol{r}_{i,j}$ is asymptotically Gaussian distributed with mean $[\boldsymbol{\epsilon}_1^T, \boldsymbol{\epsilon}_2^T, \ldots, \boldsymbol{\epsilon}_{N_j}^T]^T = \boldsymbol{A}_{i,j} \boldsymbol{\omega}_i + \boldsymbol{A}_{j,i} \boldsymbol{\omega}_j$ and covariance matrix $\boldsymbol{B}_{\boldsymbol{\epsilon}}^{\{i,j\}}(\{\boldsymbol{\epsilon}_k\}_{k=1}^{N_j}, \{\boldsymbol{h}_k\}_{k=1}^{N_j})$ [13]. That is, $\boldsymbol{r}_{i,j} \sim \mathcal{N}(\boldsymbol{r}_{i,j}; \boldsymbol{\epsilon}_{i,j}, \boldsymbol{B}_{\boldsymbol{\epsilon}}^{\{i,j\}}(\{\boldsymbol{\epsilon}_k\}_{k=1}^{N_j}, \{\boldsymbol{h}_k\}_{k=1}^{N_j}))$. Notice that the CRB depends on the true value of $\{\boldsymbol{\epsilon}_k\}_{k=1}^{N_j}$ and $\{\boldsymbol{h}_k\}_{k=1}^{N_j}$, but since we have obtained the ML estimate $\{\hat{\boldsymbol{\epsilon}}\}_{k=1}^{N_j}$ and $\{\hat{\boldsymbol{h}}_k\}_{k=1}^{N_j}$, $\boldsymbol{B}_{\boldsymbol{\epsilon}}^{\{i,j\}}(\{\boldsymbol{\epsilon}_k\}_{k=1}^{N_j}, \{\boldsymbol{h}_k\}_{k=1}^{N_j})$ can be closely approximated by $\boldsymbol{R}_{i,j} = \boldsymbol{B}_{\boldsymbol{\epsilon}}^{\{i,j\}}(\{\hat{\boldsymbol{\epsilon}}_k\}_{k=1}^{N_j}, \{\hat{\boldsymbol{h}}_k\}_{k=1}^{N_j})$.

Notice that traditional CFO estimation for point-to-point link only estimates $N_i N_j$ relative CFOs given by $\boldsymbol{r}_{i,j}$ in (4). However, in order to compensate the offset of individual oscillator, we need to estimate $N_i + N_j$ absolute CFOs in $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_j$. For simple MIMO systems, [4] provides a method to resolve $N_i + N_j$ absolute CFOs from $N_i N_j$ relative CFOs. In this paper, we take a significant step further to resolve all

[1]The frequency offset in this paper is the normalized CFO, defined as $2\pi \Delta f T_s$, where Δf is the CFO in Hz, while T_s is the sampling period.

absolute CFOs in a distributed network. That is, to estimate and compensate $\boldsymbol{\omega}_i$ in each node based on estimation results of local relative CFOs $\boldsymbol{r}_{i,j}$.

III. Distributed CFOs Estimation

A. Distributed CFOs Estimation via Belief Propagation

The optimal CFO estimator at each node is the ML estimator, which finds the maximum points of the global likelihood function:

$$[(\hat{\boldsymbol{\omega}}_2^{\mathrm{ML}})^T, \ldots, (\hat{\boldsymbol{\omega}}_K^{\mathrm{ML}})^T]^T = \arg\max_{\boldsymbol{\omega}_2,\ldots,\boldsymbol{\omega}_K} p\big(\{\boldsymbol{r}_{i,j}\}_{\{i,j\}\in\mathcal{E}} | \boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_K\big). \tag{5}$$

Here, without loss of generality, node 1 is assumed to be the reference node, so $\boldsymbol{\omega}_1$ is known. The global likelihood function is given by

$$p\big(\{\boldsymbol{r}_{i,j}\}_{\{i,j\}\in\mathcal{E}} | \boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_K\big) \propto \delta(\boldsymbol{\omega}_1) \prod_{\{i,j\}\in\mathcal{E}} p(\boldsymbol{r}_{i,j}|\boldsymbol{\omega}_i, \boldsymbol{\omega}_j), \tag{6}$$

where $p(\boldsymbol{r}_{i,j}|\boldsymbol{\omega}_i, \boldsymbol{\omega}_j) \sim \mathcal{N}(\boldsymbol{r}_{i,j}; \boldsymbol{A}_{i,j}\boldsymbol{\omega}_i + \boldsymbol{A}_{j,i}\boldsymbol{\omega}_j, \boldsymbol{R}_{i,j})$ is the local likelihood function. Notice that since the likelihood function in (6) depends on interactions among all unknown variables, the computation of $\hat{\boldsymbol{\omega}}_i^{\mathrm{ML}}$ in (5) requires gathering of all information in a central processing unit. However, such centralized processing is not favorable in large-scale networks.

In order to compute the optimal estimate (5) in a distributed way, one can exploit the conditional independence structure of the joint distribution (6), which is conveniently revealed by factor graph (FG). FG is an undirected bipartite graphical representation of a joint distribution that unifies direct and undirected graphical models. An example of FG in the context of network-wide synchronization is shown in Fig. 3. In the FG, there are two distinct kinds of nodes. One is variable nodes representing local synchronization parameters $\boldsymbol{\omega}_i$. If there is a communication link between node i and node j, the corresponding variable nodes $\boldsymbol{\omega}_i$ and $\boldsymbol{\omega}_j$ are linked by the other kind of node, factor node $f_{i,j} = p(\boldsymbol{r}_{i,j}|\boldsymbol{\omega}_i, \boldsymbol{\omega}_j)$ representing the local likelihood function[2]. On the other hand, the factor node $f_1 = \delta(\boldsymbol{\omega}_1)$ denotes value of frequency offsets of node 1, and is connected only to the variable node $\boldsymbol{\omega}_1$. Note that the FG is bipartite which means neighbors of a factor node must be variable nodes and vice versa.

From the FG, two kinds of messages are passed around: One is the message from factor node f (likelihood function $f_{i,j}$ or prior distribution f_1) to its neighboring variable node $\boldsymbol{\omega}_i$, defined as the product of the function f with messages received from all neighboring variable nodes except $\boldsymbol{\omega}_i$, and then marginalized for $\boldsymbol{\omega}_i$ [14]

$$m_{f\to i}^{(l)}(\boldsymbol{\omega}_i) = \int \cdots \int f \times \prod_{\boldsymbol{\omega}_j \in \mathcal{B}(f)\setminus\boldsymbol{\omega}_i} m_{j\to f}^{(l-1)}(\boldsymbol{\omega}_j) d\{\boldsymbol{\omega}_j\}_{\boldsymbol{\omega}_j\in\mathcal{B}(f)\setminus\boldsymbol{\omega}_i}, \tag{7}$$

where $\mathcal{B}(f)$ denotes the set of variable nodes that are direct neighbors of the factor nodes f on the FG and $\mathcal{B}(f)\setminus\boldsymbol{\omega}_i$

[2]Note that $f_{i,j}$=$f_{j,i}$.

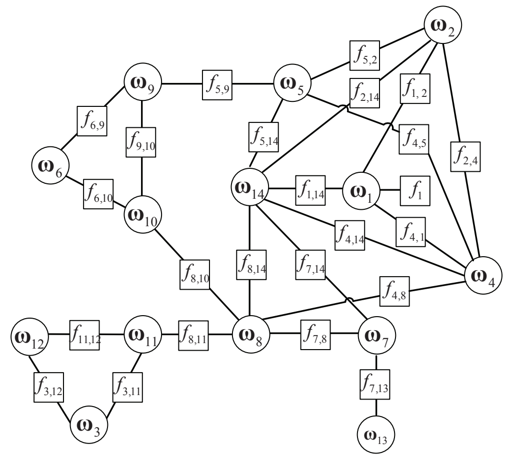

Fig. 3. The factor graph representation of the network in Fig. 2.

denotes the same set but with $\boldsymbol{\omega}_i$ removed. In (7), $m_{j\to f}^{(l-1)}(\boldsymbol{\omega}_j)$ is the other kind of message from variable node to factor node which is simply the product of the incoming messages on other links, i.e.,

$$m_{j\to f}^{(l)}(\boldsymbol{\omega}_j) = \prod_{\tilde{f}\in\mathcal{B}(\boldsymbol{\omega}_j)\setminus f} m_{\tilde{f}\to j}^{(l)}(\boldsymbol{\omega}_i), \tag{8}$$

where $\mathcal{B}(\boldsymbol{\omega}_j)$ denotes the set of factor nodes that are direct neighbors of the variable nodes $\boldsymbol{\omega}_j$ on the FG.

These two kinds of messages are iteratively updated at variable nodes and factor nodes, respectively. In any round of message exchange, a belief of $\boldsymbol{\omega}_i$ can be computed at variable node i as the product of all the incoming messages from neighboring factor nodes, which is given by

$$b^{(l)}(\boldsymbol{\omega}_i) = \prod_{f\in\mathcal{B}(\boldsymbol{\omega}_i)} m_{f\to i}^{(l)}(\boldsymbol{\omega}_i). \tag{9}$$

Thereupon, the estimate of $\boldsymbol{\omega}_i$ in the l^{th} iteration is simply

$$\hat{\boldsymbol{\omega}}_i^{(l)} = \int \boldsymbol{\omega}_i b^{(l)}(\boldsymbol{\omega}_i) d\boldsymbol{\omega}_i. \tag{10}$$

B. Message Computation

In the BP framework, messages are passed and updated iteratively. In order to start the recursion, in the first round of message passing, it is reasonable to set the initial messages from factor nodes to variable nodes $m_{f_{i,j}\to i}^{(0)}(\boldsymbol{\omega}_i)$ as non-informative message $\mathcal{N}(\boldsymbol{\omega}_i; \boldsymbol{v}_{f_{i,j}\to i}^{(0)}, \boldsymbol{C}_{f_{i,j}\to i}^{(0)})$, where $\boldsymbol{v}_{f_{i,j}\to i}^{(0)}$ can be arbitrarily chosen and $[\boldsymbol{C}_{f_{i,j}\to i}^{(0)})]^{-1} = \mathbf{0}$. On the other hand, the message from f_1 to $\boldsymbol{\omega}_1$ is always $\delta(\boldsymbol{\omega}_1)$, which can be viewed as a Gaussian distribution with mean $\boldsymbol{\omega}_1$ and covariance $\mathbf{0}$. Thereupon, based on the fact that the likelihood function $f_{i,j}$ is also Gaussian, according to (7), $m_{f_{i,j}\to i}^{(1)}(\boldsymbol{\omega}_i)$ is a Gaussian function. In addition, $m_{j\to f_{i,j}}^{(1)}(\boldsymbol{\omega}_j)$ being the product of Gaussian functions in (8) is also a Gaussian function. Thus during each round of message exchange, all the messages are Gaussian functions and only the mean vectors and covariance matrices need to be exchanged between factor nodes and variable nodes.

At this point, we can compute the messages at any iteration. In general, for the l^{th} $(l = 2, 3, \cdots)$ round of message exchange, factor node $f_{i,j}$ receive messages $m^{(l-1)}_{j\to f_{i,j}}(\boldsymbol{\omega}_j)$ from its neighboring variable nodes and then compute messages using (7). After some derivations, it can be obtained that

$$\begin{aligned} m^{(l)}_{f_{i,j}\to i}(\boldsymbol{\omega}_i) &= \int p(\boldsymbol{A}_{i,j}, \boldsymbol{A}_{j,i}|\boldsymbol{\omega}_i, \boldsymbol{\omega}_j) m^{(l-1)}_{j\to f_{i,j}}(\boldsymbol{\omega}_j) d\boldsymbol{\omega}_j \\ &\propto \mathcal{N}(\boldsymbol{\omega}_i; \boldsymbol{v}^{(l)}_{f_{i,j}\to i}, \boldsymbol{C}^{(l)}_{f_{i,j}\to i}), \end{aligned} \tag{11}$$

where the inverse of covariance matrix is

$$\left[\boldsymbol{C}^{(l)}_{f_{i,j}\to i}\right]^{-1} = \boldsymbol{A}^T_{i,j}\left[\boldsymbol{R}_{i,j} + \boldsymbol{A}_{j,i}\boldsymbol{C}^{(l-1)}_{j\to f_{i,j}}\boldsymbol{A}^T_{j,i}\right]^{-1}\boldsymbol{A}_{i,j}, \tag{12}$$

and the mean vector is

$$\begin{aligned} \boldsymbol{v}^{(l)}_{f_{i,j}\to i} =& \boldsymbol{C}^{(l)}_{f_{i,j}\to i}\boldsymbol{A}^T_{i,j}\left[\boldsymbol{R}_{i,j} + \boldsymbol{A}_{j,i}\boldsymbol{C}^{(l-1)}_{j\to f_{i,j}}\boldsymbol{A}^T_{j,i}\right]^{-1} \\ &\times (\boldsymbol{r}_{i,j} - \boldsymbol{A}_{j,i}\boldsymbol{v}^{(l-1)}_{j\to f_{i,j}}). \end{aligned} \tag{13}$$

On the other hand, using (8), the messages passed from variable nodes to factor nodes can be computed as

$$\begin{aligned} m^{(l)}_{i\to f_{i,j}}(\boldsymbol{\omega}_i) &= \prod_{f\in\mathcal{B}(\boldsymbol{\omega}_i)\setminus f_{i,j}} m^{(l)}_{f\to i}(\boldsymbol{\omega}_i) \\ &\propto \mathcal{N}(\boldsymbol{\omega}_i; \boldsymbol{v}^{(l)}_{i\to f_{i,j}}, \boldsymbol{C}^{(l)}_{i\to f_{i,j}}), \end{aligned} \tag{14}$$

where

$$\left[\boldsymbol{C}^{(l)}_{i\to f_{i,j}}\right]^{-1} = \sum_{f\in\mathcal{B}(\boldsymbol{\omega}_i)\setminus f_{i,j}} \left[\boldsymbol{C}^{(l)}_{f\to i}\right]^{-1}, \tag{15}$$

and

$$\boldsymbol{v}^{(l)}_{i\to f_{i,j}} = \boldsymbol{C}^{(l)}_{i\to f_{i,j}} \sum_{f\in\mathcal{B}(\boldsymbol{\omega}_i)\setminus f_{i,j}} \left[\boldsymbol{C}^{(l)}_{f\to i}\right]^{-1}\boldsymbol{v}^{(l)}_{f\to i}. \tag{16}$$

Furthermore, during each round of message passing, each node can compute the belief for $\boldsymbol{\omega}_i$ using (9), which can be easily shown to be $b^{(l)}_i(\boldsymbol{\omega}_i) \sim \mathcal{N}(\boldsymbol{\omega}_i; \boldsymbol{\mu}^{(l)}_i, \boldsymbol{P}^{(l)}_i)$, with the inverse of covariance matrix

$$\left[\boldsymbol{P}^{(l)}_i\right]^{-1} = \sum_{j\in\mathcal{I}(i)} \left[\boldsymbol{C}^{(l)}_{f_{i,j}\to i}\right]^{-1}, \tag{17}$$

and mean vector

$$\boldsymbol{\mu}^{(l)}_i = \boldsymbol{P}^{(l)}_i \sum_{j\in\mathcal{I}(i)} \left[\boldsymbol{C}^{(l)}_{f_{i,j}\to i}\right]^{-1}\boldsymbol{v}^{(l)}_{f_{i,j}\to i}. \tag{18}$$

When the algorithm converges or the maximum number of message exchange is reached, each node computes the CFOs according to (10) as

$$\hat{\boldsymbol{\omega}}^{(l)}_i = \int \boldsymbol{\omega}_i b^{(l)}(\boldsymbol{\omega}_i) d\boldsymbol{\omega}_i = \boldsymbol{\mu}^{(l)}_i. \tag{19}$$

The iterative algorithm based on BP is summarized as follows. The algorithm is started by setting the message from factor node to variable node as $m^{(0)}_{f_1\to 1}(\boldsymbol{\omega}_1) = \delta(\boldsymbol{\omega}_1)$ and $m^{(0)}_{f_{i,j}\to i}(\boldsymbol{\omega}_i) = \mathcal{N}(\boldsymbol{\omega}_i; \boldsymbol{v}^{(0)}_{f_{i,j}\to i}, \boldsymbol{C}^{(0)}_{f_{i,j}\to i})$ with $\boldsymbol{v}^{(0)}_{f_{i,j}\to i} = \boldsymbol{0}$ and $[\boldsymbol{C}^{(0)}_{f_{i,j}\to i})]^{-1} = \boldsymbol{0}$. At each round of message exchange, every variable node computes the output messages to factor nodes according to (15) and (16). After receiving the messages from its neighboring variable nodes, each factor node computes its output messages according to (12) and (13). Such iteration is terminated when (18) converges (e.g., when $\|\boldsymbol{\mu}^{(l)}_i - \boldsymbol{\mu}^{(l-1)}_i\| < \eta$, where η is a threshold) or the maximum number of iteration is reached. Then the estimate of CFOs of each node is obtained as in (19).

Notice that after convergence, the belief $b^{(l)}(\boldsymbol{\omega}_i)$ at each variable node corresponds to the marginal distribution of that variable exactly when the underlying FG is loop free [14]. However, for the FG with loops, it is generally difficult to know if BP will converge [15]. Despite the lack of general results on BP, the convergence and optimality of BP for network-wide CFO estimation algorithm are analytically proved in [16].

Remark 1: In practical networks, there is neither factor nodes nor variable nodes. The two kinds of messages $m^{(l)}_{i\to f_{i,j}}(\boldsymbol{\omega}_i)$ and $m^{(l)}_{f_{i,j}\to j}(\boldsymbol{\omega}_j)$ are computed locally at node i, and only mean vector $\boldsymbol{v}^{(l)}_{f_{i,j}\to j}(\boldsymbol{\omega}_j)$ and covariance matrix $\boldsymbol{C}^{(l)}_{f_{i,j}\to j}(\boldsymbol{\omega}_j)$ are passed from node i to node j during each round of message exchange of BP. It can be seen the algorithm is fully distributed and each node only needs to exchange limited information with neighboring nodes.

IV. Simulation Results

This section presents numerical results to assess the performance of the proposed algorithm. In each trial, the normalized CFO of each antenna on each node (except node 1 where CFO is zero) is generated independently and is uniformly distributed in the range $2\pi[-0.2, 0.2]$. Besides, the channel between each pair of nodes is Rayleigh flat-fading. The relative CFOs and channels are first estimated based on the algorithm in [8], with training length N. Then the BP algorithm is executed for network-wide CFOs estimation and compensation. 5000 simulation runs were performed to obtain the average performance for each point in the figures.

In order to provide a performance benchmark for the proposed distributed algorithm, the centralized CRB is computed. The CRB can be easily derived by stacking all the pair-wise information denoted by (4) as

$$\boldsymbol{r} = \boldsymbol{A}\boldsymbol{\omega} + \boldsymbol{n}, \tag{20}$$

where $\boldsymbol{r}$ is a vector containing $\boldsymbol{r}_{i,j}$ with ascending indexes first on i and then on j; and $\boldsymbol{n}$ containing $\boldsymbol{n}_{i,j}$ with the indexes i, j ordered in the same way as in $\boldsymbol{r}$. Since $\boldsymbol{n} \sim \mathcal{N}(\boldsymbol{n}; \boldsymbol{0}, \boldsymbol{R})$, where $\boldsymbol{R}$ is a block diagonal matrix with $\boldsymbol{R}_{i,j}$ as block diagonal and with the same order as $\boldsymbol{r}_{i,j}$ in $\boldsymbol{r}$, and (20) is a standard linear model, the CRB for $\boldsymbol{\omega}$ is given by $\text{CRB}(\boldsymbol{\omega}) = \left(\boldsymbol{A}^T\boldsymbol{R}^{-1}\boldsymbol{A}\right)^{-1}$ [13].

First consider the fixed network shown in Fig. 2 and each node equipped with two antennas. We employ training with length $N = 16$ for relative CFOs estimation. The SNR during training stage and BP message passing are the same. Fig. 4 shows the sum MSE over the two antennas of $\boldsymbol{\omega}_i$ for nodes

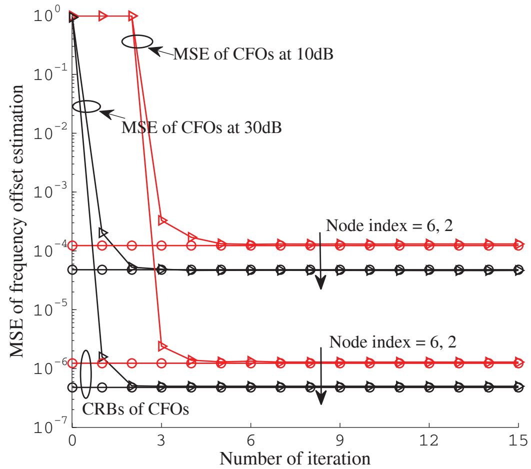

Fig. 4. Convergence performance of the proposed algorithm at Node 2 and 6.

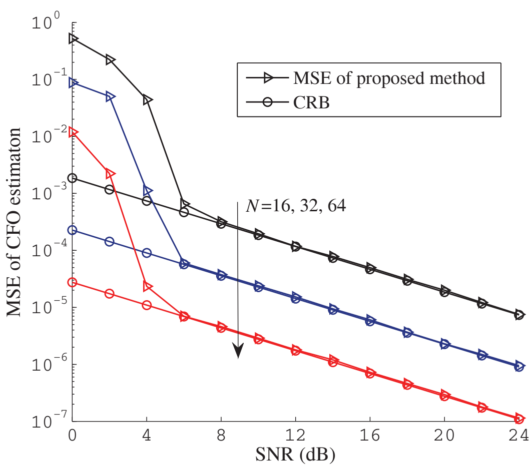

Fig. 5. MSE of CFOs $\{\boldsymbol{\omega}_i\}_{i\in\mathcal{V}}$ averaged over the whole network with respect to SNRs.

$i = \{6, 2\}$ as a function of BP iteration number l. These two nodes are chosen to represent nodes close to (node 2) and far away (node 6) from the reference node. It can be seen that for both SNR= 10dB and 30dB, the MSEs decrease quickly and touch the corresponding CRBs in only a few iterations.

Fig. 5 shows the average sum MSE of $\{\boldsymbol{\omega}_i\}_{i\in\mathcal{V}}$ versus SNRs for different training length N. The network is randomly generated within the $[0, 100] \times [0, 100]$ area in each trial, with the communication range for each node equals 38 and each node is equipped with 2 antennas. As shown in the figure, the MSEs of the proposed distributed algorithm achieve the best performance as the MSEs touch the corresponding CRBs. Furthermore, with increasing N, the approximation of $\boldsymbol{R}_{i,j}$ to $\boldsymbol{B}_{\boldsymbol{\epsilon}}^{\{i,j\}}(\{\epsilon_k\}_{k=1}^{N_j}, \{\boldsymbol{h}_k\}_{k=1}^{N_j})$ becomes better at lower SNR, and thus the estimation MSEs of $\boldsymbol{\omega}_i$ achieves the corresponding CRBs earlier.

V. Conclusions

In this paper, a fully distributed CFOs estimation algorithm for multi-cell cooperative networks was proposed. The algorithm is based on BP and is easy to be implemented by exchanging limited amount of information between neighboring nodes, thus is scalable with network size. Simulation results showed that the MSE of the proposed method touches the CRB within only a few iterations. Finally, it is worth to point out that while the focus of this paper is on multi-cell cooperative networks, the proposed algorithm is a general one, and can be applied to other distributed networks such as relay networks, heterogenous networks and massive MIMO networks.

References

[1] A. D. Wyner, "Shannon-theoretic approach to a Gaussian cellular multiple-access channel," *IEEE Trans. Inf. Theory*, vol. 40, no. 6, pp. 1713-1727, Nov. 1994.

[2] R. Irmer, H. Droste, P. Marsch, M. Grieger, G. Fettweis, S. Brueck, H.-P. Mayer, L. Thiele, and V. Jungnickel, "Coordinated multipoint: Concepts, performance, and field trial results," *IEEE Commun. Mag.*, vol. 49, no. 2, pp. 102-111, Feb. 2011.

[3] D. Gesbert, S. Hanly, H. Huang, S. Shamai, O. Simeone, and W. Yu, "Multi-cell MIMO cooperative networks: a new look at interference," *IEEE J. Select. Areas Commun.*, vol. 28, no. 9, pp. 1380-1408, Dec. 2010.

[4] B. Zarikoff and J. Cavers, "Coordinated multi-cell systems: Carrier frequency offset estimation and correction," *IEEE Trans. Select. Areas Commun.*, vol. 28, no. 9, pp. 1490–1501, Dec. 2010

[5] L. Sanguinetti, M. Morelli, and G. Imbarlina, "An EM-based frequency offset estimator for OFDM systems with unknown interference," *IEEE Trans. Wireless Commun.*, vol. 8, no. 9, pp. 4470–4475, Sep. 2009.

[6] M. Ghogho and A. Swami, "Training design for multipath channel and frequency-offset estimation in MIMO systems," *IEEE Trans. Signal Process.*, vol. 54, no. 10, pp. 3957–3965, Oct. 2006.

[7] O. Besson and P. Stoica, "On parameter estimation of MIMO at fading channels with frequency offsets," *IEEE Trans. Signal Process.*, vol. 51, no. 3, pp. 602–613, Mar. 2003

[8] M. O. Pun, M. Morelli, and C.-C. J. Kuo, "Maximum-likelihood synchronization and channel estimation for OFDMA uplink transmissions," *IEEE Trans. Commun.*, vol. 54, no. 4, pp. 726–736, Apr. 2006.

[9] T. Pham, A. Nallanathan, and Y. Liang, "Joint channel and frequency offset estimation in distributed MIMO flat-fading channels," *IEEE Trans. Wireless Commun.*, vol. 7, no. 2, pp. 648–656, Feb. 2008

[10] J. Chen, Y.-C. Wu, S. Ma, and T.-S. Ng, "Joint CFO and channel estimation for multiuser MIMO-OFDM systems with optimal training sequences," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 4008-4019, Aug. 2008.

[11] H. Mehrpouyan and S. D. Blostein, "Bounds and algorithms for multiple frequency offset estimation in cooperative networks," *IEEE Trans. Wireless Commun.*, vol. 10, no. 4, pp. 1300–1311, Apr. 2011.

[12] H. G. Jeon, K. S. Kim, and E. Serpedin, "An efficient blind deterministic frequency offset estimator for OFDM systesm," *IEEE Trans on Communications.*, vol. 59, no. 4, Apr. 2011.

[13] S. M. Kay, *Fundamentals of Statistical Signal Processing Estimation Theory*, Upper Saddle River, NJ: Prentice-Hall, 1993.

[14] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory,* vol. 47, no. 2, pp. 498-519, Feb. 2001.

[15] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," *Neural Computation,* vol, 13, no. 10, pp. 2173-2200, 2001.

[16] J. Du and Y.-C. Wu, "Network-wide distributed carrier frequency offsets estimation and compensation via belief propagation," accepted in *IEEE Trans. Signal Process.*