Virtual Clustering for Distributed Consensus-based Estimation in Cooperative Networks

Guang Xu, Shengdi Wang, Henning Paul, and Armin Dekorsy
Department of Communications Engineering, University of Bremen, Bremen, Germany
Email: {xu,wang,paul,dekorsy}@ant.uni-bremen.de

***Abstract*—This paper presents a new approach for distributed consensus-based estimation with low communication effort in cooperative networks, where a group of nodes cooperates to estimate source messages with information exchange achieving consensus among all nodes. A new distributed estimation algorithm is developed by adopting the novel approach of virtual clustering, in which the size of exchanged data is reduced during the distributed processing by the partial transmission after data clustering. Moreover, the required communication effort and estimation performance of the algorithm are evaluated for networks with different topology and varying system parameters. We show that communication cost is reduced considerably by the proposed algorithm while keeping the estimation performance.**

I. Introduction

The study of consensus-based estimation has been an active topic for the applications of monitoring, tracking, and control in cooperative networks, e.g., wireless sensor networks [1]. As a common scenario, several nodes are distributed over an area and are connected with inter-node links forming a certain topology. Those nodes are deployed to detect signals or quantities from some common sources within the network. Conventionally, either centralized processing, e.g., in a fusion center [2], or distributed processing among cooperated nodes can be applied [3]. Here, we focus on the distributed estimation, where each node performs local processing iteratively with additional information exchanged between neighboring nodes. Plenty of work has investigated such distributed estimation with associated algorithms, e.g., the diffusion based algorithms [4], [5], which achieve a consensus on the average of observations over the network. However, in our system the observations are distorted by, e.g., a fading channel, thus source data cannot be properly recovered by an averaging process. To achieve satisfactory estimation performance, primal and dual decomposition method based algorithms [6], [7], [8], [9] can be applied, which achieve a consensus of the estimate on all sources by solving a constrained convex optimization problem [10].

For the distributed estimation, one critical issue is about the communication effort required during the distributed processing. In order to reduce the overhead over inter-node links, some algorithms in [11], [12] have been proposed to achieve efficient transmission where each node exchanges variables of reduced dimension using sub-graphing and coloring scheme to specify the transmission. Inspired by those work, we adopt a new approach termed virtual clustering (VC) to reduce the dimension of exchanged data during the distributed processing.

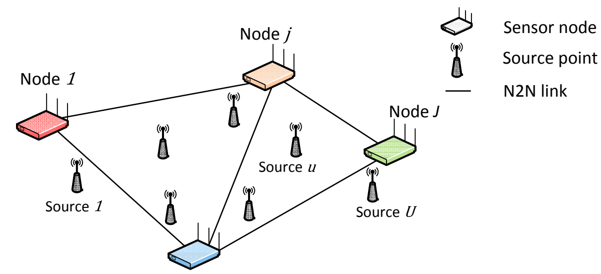

Fig. 1. A sensor network with J nodes receiving and detecting the common messages from U sources in a cooperative way.

Whereas in contrast to [11], [12] where each node only recovers the source information within a predefined local domain, here we aim to achieve a consensus of estimates on all sources over the whole network. By applying the VC approach to one fundamental distributed consensus-based estimation (DiCE) algorithm [7], we achieve a novel VC-DiCE algorithm correspondingly. The derivation of the proposed algorithm is detailed in the following sections, and performance of the algorithm in terms of convergence and communication overhead is evaluated considering different system parameters.

The remainder of this paper is structured as follows: The system model is described in Section II. A detailed derivation and discussion of our novel algorithm are given in Section IV. In the subsequent Section V, the proposed algorithm is simulated numerically, and the corresponding results are evaluated. Finally, the paper is concluded in Section VI.

Notation: Throughout this paper, we use uppercase bold letters $\mathbf{A}$ for matrices, while keep lowercase bold letters $\mathbf{a}$ for vectors and normal letter a for scalars. Vector $\mathbf{a}_{n,m}$ denotes a sub-vector of vector $\mathbf{a}_n$. $\mathbf{I}$ is an identity matrix and $\mathbf{0}$ is a zero matrix or vector whose entries are all zeros. A diagonal matrix is represented as $\mathrm{diag}(\cdot)$. We denote the transpose operator with $(\cdot)^{\mathrm{T}}$ and Hermitian with $(\cdot)^{\mathrm{H}}$.

II. System description

Fig. 1 depicts the system scenario, where a sensor network composed of a set of nodes $\mathcal{J} = \{1, ..., J\}$ monitors the data $\boldsymbol{x}_u$ broadcasted from $u = 1, .., U$ source points. Those nodes are connected through assumedly error-free links resulting in a set of edges $\mathcal{E}$ between node j and the nodes i in its neighborhood $\mathcal{N}_j$. Here, each node and source point is assumed to be equipped with N_{R} receive and N_{T} transmit antennas, respec-

978-1-5090-5932-4/17/$31.00 ©2017 IEEE

tively. Thus, a stacked data vector $\boldsymbol{x} = [\boldsymbol{x}_1^{\mathrm{T}}, .., \boldsymbol{x}_U^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{C}^{UN_{\mathrm{T}} \times 1}$ can be modeled as the system input, which is received by each node j leading to a local observation $\mathbf{y}_j \in \mathbb{C}^{N_{\mathrm{R}} \times 1}$:

$$\mathbf{y}_j = \mathbf{H}_j \boldsymbol{x} + \mathbf{n}_j, \tag{1}$$

where the channel matrix $\mathbf{H}_j \in \mathbb{C}^{N_{\mathrm{R}} \times UN_{\mathrm{T}}}$ is assumed to be perfectly known at each node, and additive white Gaussian noise (AWGN) $\mathbf{n}_j \in \mathbb{C}^{N_{\mathrm{R}} \times 1}$ with a variance σ_n^2 is considered. In this network, each node aims to recover the total source information $\boldsymbol{x}$. However, it cannot be properly detected only based on the local information, since an under-determined system i.e., $UN_{\mathrm{T}} > N_{\mathrm{R}}$ is assumed here. To reach a consensus of estimates between nodes, one possible way is joint estimation in a central node where information from all nodes is collected. Whereas, in our scenario, no central node is deployed. Thus, distributed processing among nodes is desired, where each node performs local processing and shares information with its neighboring nodes through inter-node links to achieve a consensus of estimates in an iterative fashion.

III. Consensus-Based Distributed Estimation

For the consensus-based estimation, we adopt, e.g., the Least-Square (LS) criterion to minimize the sum estimation errors of all nodes with additional constraints to keep the consistency on local estimates $\mathbf{x}_j = \mathbf{x}_i,\ i, j = 1, .., J$ over the whole network. To this end, we obtain our target constrained LS problem given by

$$\begin{aligned} &\underset{\mathbf{x}_j \in \mathbb{C}^{UN_{\mathrm{T}} \times 1}}{\text{minimize}} \ \frac{1}{2} \sum_{j=1}^{J} \left\| \mathbf{y}_j - \mathbf{H}_j \mathbf{x}_j \right\|^2 \\ &\text{s.t.} \quad \mathbf{x}_j = \mathbf{x}_i, \quad \forall i \in \mathcal{N}_j, \forall j \in \mathcal{J}. \end{aligned} \tag{2}$$

Such a constrained convex problem can be solved by, e.g., the alternating direction method of multipliers (ADMM) [13], like the DiCE algorithm [7], where the constraints $\mathbf{x}_j = \mathbf{x}_i$ in (2) are decoupled for parallel implementation by introducing auxiliary variables $\mathbf{z}_j \in \mathbb{C}^{UN_{\mathrm{T}} \times 1}$ per node j, i.e., $\mathbf{x}_j = \mathbf{z}_j$ and $\mathbf{x}_j = \mathbf{z}_i, j \in \mathcal{J}, i \in \mathcal{N}_j$. In the DiCE algorithm an iterative solution of the local estimate $\mathbf{x}_j^{k+1}$ as well as the variable $\mathbf{z}_j^{k+1}$ of each node j in iteration $k+1$ is given by

$$\mathbf{x}_j^{k+1} = \arg\min_{\mathbf{x}_j} \mathcal{L}_j(\mathbf{x}_j; \mathbf{z}_i^k, \boldsymbol{\lambda}_{ji}^k) \tag{3a}$$

$$\mathbf{z}_j^{k+1} = \arg\min_{\mathbf{z}_j} \mathcal{L}_j(\mathbf{z}_j; \mathbf{x}_i^{k+1}, \boldsymbol{\lambda}_{ij}^k) \tag{3b}$$

$$\boldsymbol{\lambda}_{ji}^{k+1} = \boldsymbol{\lambda}_{ji}^k - \frac{1}{\mu}(\mathbf{x}_j^{k+1} - \mathbf{z}_i^{k+1}),\ i \in \mathcal{N}_j', \tag{3c}$$

with the update of Lagrange multiplier $\boldsymbol{\lambda}_{ji} \in \mathbb{C}^{UN_{\mathrm{T}} \times 1}$ enforcing the constraint between node j and i. Function $\mathcal{L}_j(\cdot)$ denotes the Lagrangian function of the constrained problem (2) separated for each local node j (details in [7]). μ is a step-size parameter. The set $\mathcal{N}_j' = \mathcal{N}_j \cup \{j\}$ is an extension of the neighbor set $\mathcal{N}_j$ by node j.

According to the update equations (3) of the DiCE algorithm, the local variables $\mathbf{x}_j, \mathbf{z}_j$ and multipliers $\boldsymbol{\lambda}_{ji}$ of node j are updated in an iterative way with the vectors $\mathbf{x}_i, \mathbf{z}_i$ and $\boldsymbol{\lambda}_{ij}$ received from neighboring nodes i. A considerable communication effort is required during the iterative processing, particularly for the case when the dimension of the exchanged vector is large. Hence, in practice high communication cost becomes a critical issue to the distributed estimation.

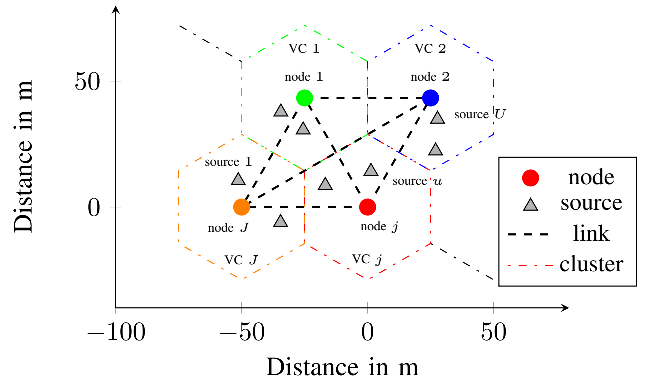

Fig. 2. A network of J nodes with the virtual cluster for each. All nodes cooperate through inter-node links to detect the common messages from U sources located in different virtual clusters.

IV. Virtual Data Clustering for Distributed Estimation

In order to reduce the communication effort, a novel virtual clustering approach is applied to DiCE resulting in a new algorithm VC-DiCE, which aims to decrease the amount of data exchanged over the inter-node links. Note that in each iteration of the DiCE algorithm, the complete vectors of $\mathbf{x}_i, \mathbf{z}_i$ and $\boldsymbol{\lambda}_{ij}$ need to be exchanged, while in VC-DiCE only a part of the entire vector is designed to be transmitted, which is determined by the partial consensus constraints from virtual clustering.

A. Virtual clustering for partial consensus constraint

The virtual clustering is defined to classify the sources corresponds to local estimation data for partial transmission between neighboring nodes. For virtual clustering, each source is assumed to be covered by one or multiple virtual clusters depending on the size of each cluster. An example of regular virtual clustering is depicted in Fig. 2, where J hexagonal clusters with same size are defined for each node in a fully meshed network. We define that each source point u located in the cluster $\kappa \in \{1, .., J\}$ belongs to set $\mathcal{V}_\kappa$. Correspondingly, the local estimation data can be classified by the virtual clustering, i.e., each estimate vector $\mathbf{x}_j$ on node j is decomposed into a set of sub-vectors $\{\mathbf{x}_{j,1}, \mathbf{x}_{j,2}, ..., \mathbf{x}_{j,J}\}$ with each $\mathbf{x}_{j,\kappa}$ covering the data $\boldsymbol{x}_u$ of sources $u \in \mathcal{V}_\kappa, \kappa = 1, .., J$.

Considering the distributed consensus-based estimation problem (2), the constraint on entire local estimate vectors $\mathbf{x}_j = \mathbf{x}_i, i, j \in \mathcal{J}$ has to be satisfied. Furthermore, following the principle of virtual clustering, we can extract sub-vectors $\mathbf{x}_{j,\kappa}$ from the entire vector $\mathbf{x}_j$ resulting in partial consensus constraints between neighboring nodes j and $i \in \mathcal{N}_j$, i.e., $\mathbf{x}_j = \mathbf{x}_i$ is specified into $\mathbf{x}_{j,\kappa} = \mathbf{x}_{i,\kappa}, \kappa \in \{j, i\}$. For example, in Fig. 2 assuming $J = 4$, the local estimate vector $\mathbf{x}_j$ on node j is thus composed of four sub-vectors $[\mathbf{x}_{j,1}^{\mathrm{T}}, \mathbf{x}_{j,2}^{\mathrm{T}}, \mathbf{x}_{j,3}^{\mathrm{T}}, \mathbf{x}_{j,4}^{\mathrm{T}}]^{\mathrm{T}}$.

Then, the consensus constraints over the whole network can be split into several partial constraints:

$$\mathbf{x}_1 = \mathbf{x}_2 = \mathbf{x}_3 = \mathbf{x}_4 \Rightarrow \begin{cases} \mathbf{x}_{1,1} = \mathbf{x}_{2,1} = \mathbf{x}_{3,1} = \mathbf{x}_{4,1}, \\ \mathbf{x}_{1,2} = \mathbf{x}_{2,2} = \mathbf{x}_{3,2} = \mathbf{x}_{4,2}, \\ \mathbf{x}_{1,3} = \mathbf{x}_{2,3} = \mathbf{x}_{3,3} = \mathbf{x}_{4,3}, \\ \mathbf{x}_{1,4} = \mathbf{x}_{2,4} = \mathbf{x}_{3,4} = \mathbf{x}_{4,4}. \end{cases} \tag{4}$$

Note that, in order to achieve such a consensus of estimates during the distributed processing, partial information needs to be exchanged between neighboring nodes, which is determined by the partial constraints. Thus, by using the virtual clustering approach, we can reduce the size of data to be transmitted.

B. Virtual data selection matrix

To select the partial information, e.g., sub-vector $\mathbf{x}_{j,i}$ from the entire vector $\mathbf{x}_j$, we define a virtual data selection (VDS) matrix $\mathbf{C}_{ji}$ on each node j to select the part of estimate data related to the source in cluster $i \in \mathcal{N}_j$ as $\mathbf{x}_{j,i} = \mathbf{C}_{ji}\mathbf{x}_j$. Thus, in the example of (4) above, those sub-vectors can be represented by using the VDS matrices defined as $\mathbf{C}_{j1} = [\mathbf{I}, \mathbf{0}, \mathbf{0}, \mathbf{0}], \mathbf{C}_{j2} = [\mathbf{0}, \mathbf{I}, \mathbf{0}, \mathbf{0}], \mathbf{C}_{j3} = [\mathbf{0}, \mathbf{0}, \mathbf{I}, \mathbf{0}], \mathbf{C}_{j4} = [\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{I}], j = 1, .., 4$. Whereas, the size of each cluster is not always the same. For a randomly connected network, the clusters can be in different sizes depending on the node connection. Taking an example of $J = 4$ nodes network in Fig. 2, where we additionally assume that node 4 is disconnected with node 2 and 3, then node 1 becomes a "bridge" to keep the whole network connected. Note that local information is only exchanged between neighboring nodes. Thus, to achieve a consensus of estimate among all nodes, the information on all sources must be passed through node 1. Hence, the virtual cluster 1 on node 1 has to include all the sources $1, .., U \in \mathcal{V}_1$, such that the corresponding VDS matrices become $\mathbf{C}_{j1} = \mathrm{diag}(\mathbf{I}, \mathbf{I}, \mathbf{I}, \mathbf{I}), \mathbf{C}_{j2} = [\mathbf{0}, \mathbf{I}, \mathbf{0}, \mathbf{0}], \mathbf{C}_{j3} = [\mathbf{0}, \mathbf{0}, \mathbf{I}, \mathbf{0}], \mathbf{C}_{j4} = [\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{I}]$. The VDS matrix on each node is determined in a learning phase following some rules summarized in [14], which is beyond scope of this paper.

C. Virtual clustering based DiCE algorithm

Recalling the DiCE algorithm, the decoupled constraints $\mathbf{x}_j = \mathbf{z}_j$ and $\mathbf{x}_j = \mathbf{z}_i$ can be incorporated into $\mathbf{x}_j = \mathbf{z}_i$ for $i \in \mathcal{N}'_j$. Moreover, by applying VC approach with VDS matrix, the constrained optimization problem (2) can be rewritten with partial consensus constraints given by

$$\begin{aligned} \underset{\mathbf{x}_j}{\text{minimize}} \quad & \tfrac{1}{2}\sum_{j=1}^{J} \|\mathbf{y}_j - \mathbf{H}_j\mathbf{x}_j\|^2 \\ \text{subject to} \quad & \mathbf{C}_{ji}\mathbf{x}_j = \mathbf{C}_{ii}\mathbf{z}_i, \\ & \mathbf{C}_{jj}\mathbf{x}_j = \mathbf{C}_{ij}\mathbf{z}_i, \qquad \forall i \in \mathcal{N}'_j, \forall j \in \mathcal{J}. \end{aligned} \tag{5}$$

Then we apply the ADMM approach to solve the new constrained LS problem (5), such that the estimates $\mathbf{x}_j$ as well as other variables $\mathbf{z}_j, \boldsymbol{\lambda}_{ji}$ can be updated in an iterative way.

Furthermore, for a distributed implementation, we firstly decompose the Lagrangian function $\mathcal{L}(\,\cdot\,)$ of problem (5) into individual sub-functions $\mathcal{L}_j(\mathbf{x}_j, \,\cdot\,)$ with respect to $\mathbf{x}_j$:

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\nu}) &= \frac{1}{2}\sum_{j=1}^{J} \|\mathbf{y}_j - \mathbf{H}_j\mathbf{x}_j\|^2 \\ &- \sum_{j=1}^{J}\sum_{i\in\mathcal{N}'_j} \Big[\boldsymbol{\lambda}_{ji}^{\mathrm{T}}(\mathbf{C}_{ji}\mathbf{x}_j - \mathbf{C}_{ii}\mathbf{z}_i) + \boldsymbol{\nu}_{ji}^{\mathrm{T}}(\mathbf{C}_{jj}\mathbf{x}_j - \mathbf{C}_{ij}\mathbf{z}_i)\Big] \\ &+ \sum_{j=1}^{J}\sum_{i\in\mathcal{N}'_j} \frac{1}{2\mu}\left[\|\mathbf{C}_{ji}\mathbf{x}_j - \mathbf{C}_{ii}\mathbf{z}_i\|^2 + \|\mathbf{C}_{jj}\mathbf{x}_j - \mathbf{C}_{ij}\mathbf{z}_i\|^2\right] \\ &= \sum_{j=1}^{J} \mathcal{L}_j(\mathbf{x}_j, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\nu}), \end{aligned} \tag{6}$$

where $\mathbf{x}$ and $\mathbf{z}$ represent the set of all variables $\mathbf{x}_j, \mathbf{z}_j, j \in \mathcal{J}$ and the set $\boldsymbol{\lambda}, \boldsymbol{\nu}$ includes all multipliers $\boldsymbol{\lambda}_{ji}, \boldsymbol{\nu}_{ji}, j \in \mathcal{J}, i \in \mathcal{N}'_j$. Note that the local estimate $\mathbf{x}_j$ of node j in $\mathcal{L}_j(\mathbf{x}_j, \,\cdot\,)$ is independent from the estimates $\mathbf{x}_i$ of other nodes. Therefore, estimate $\mathbf{x}_j$ can be achieved in a parallel and iterative fashion for all nodes $j \in \mathcal{J}$.

To calculate the local estimate $\mathbf{x}_j^{k+1}$ in iteration $k+1$, we have to minimize the cost function $\mathcal{L}_j(\mathbf{x}_j; \mathbf{z}^k, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)$ with respect to $\mathbf{x}_j$ while assuming the rest of variables $\mathbf{z}^k, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k$ from last iteration k as fixed. Then the estimate $\mathbf{x}_j^{k+1}$ can be obtained by setting $\partial\mathcal{L}_j(\mathbf{x}_j, \mathbf{z}^k, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)/\partial\mathbf{x}_j = 0$:

$$\begin{aligned} \mathbf{x}_j^{k+1} &= \underset{\mathbf{x}_j}{\text{minimize}}\, \mathcal{L}_j(\mathbf{x}_j; \mathbf{z}_i^k, \boldsymbol{\lambda}_{ji}^k, \boldsymbol{\nu}_{ji}^k) \\ &= \left[\mathbf{H}_j^{\mathrm{H}}\mathbf{H}_j + \frac{1}{\mu}\sum_{i\in\mathcal{N}'_j}\left(\mathbf{C}_{jj}^{\mathrm{T}}\mathbf{C}_{jj} + \mathbf{C}_{ji}^{\mathrm{T}}\mathbf{C}_{ji}\right)\right]^{-1} \\ &\quad \left[\mathbf{H}_j^{\mathrm{H}}\mathbf{y}_j + \sum_{i\in\mathcal{N}'_j}\left(\mathbf{C}_{ji}^{\mathrm{T}}\boldsymbol{\lambda}_{ji}^k + \mathbf{C}_{jj}^{\mathrm{T}}\boldsymbol{\nu}_{ji}^k\right)\right. \\ &\quad \left. + \sum_{i\in\mathcal{N}'_j}\frac{1}{\mu}\left(\mathbf{C}_{ji}^{\mathrm{T}}\mathbf{C}_{ii}\mathbf{z}_i^k + \mathbf{C}_{jj}^{\mathrm{T}}\mathbf{C}_{ij}\mathbf{z}_i^k\right)\right]. \end{aligned} \tag{7}$$

Once the estimate $\mathbf{x}_j^{k+1}$ is updated, each node j has to share partial information on $\mathbf{x}_j^{k+1}$ with neighboring nodes $i \in \mathcal{N}_j$. Then the auxiliary variable $\mathbf{z}_j$ on each node j can be updated according to the ADMM approach. For the distributed implementation, we also reformulate the Lagrangian function $\mathcal{L}(\,\cdot\,)$ and decompose it into sub-functions $\mathcal{L}'_j(\mathbf{z}_j, \,\cdot\,)$ with respect to $\mathbf{z}_j$:

$$\begin{aligned} \mathcal{L}(\mathbf{x}, \mathbf{z}, \boldsymbol{\lambda}, \boldsymbol{\nu}) &= \frac{1}{2}\sum_{j=1}^{J} \|\mathbf{y}_j - \mathbf{H}_j\mathbf{x}_j\|^2 \\ &- \sum_{j=1}^{J}\sum_{i\in\mathcal{N}'_j} \Big[\boldsymbol{\lambda}_{ij}^{\mathrm{T}}(\mathbf{C}_{ij}\mathbf{x}_i - \mathbf{C}_{jj}\mathbf{z}_j) + \boldsymbol{\nu}_{ij}^{\mathrm{T}}(\mathbf{C}_{ii}\mathbf{x}_i - \mathbf{C}_{ji}\mathbf{z}_j)\Big] \\ &+ \sum_{j=1}^{J}\sum_{i\in\mathcal{N}'_j} \frac{1}{2\mu}\left[\|\mathbf{C}_{ij}\mathbf{x}_i - \mathbf{C}_{jj}\mathbf{z}_j\|^2 + \|\mathbf{C}_{ii}\mathbf{x}_i - \mathbf{C}_{ji}\mathbf{z}_j\|^2\right] \\ &= \sum_{j=1}^{J} \mathcal{L}'_j(\mathbf{z}_j, \mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu}). \end{aligned} \tag{8}$$

Similarly, $\mathbf{z}_j$ is also calculated by minimizing the local cost function $\mathcal{L}'_j(\mathbf{z}_j, \mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\nu})$ with respect to $\mathbf{z}_j$. For iteration $k+1$, since $\mathbf{x}_j^{k+1}$ is already updated, then we assume $\mathbf{x}^{k+1}, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k$ in cost function $\mathcal{L}'_j(\mathbf{z}_j; \mathbf{x}^{k+1}, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)$ are fixed, and $\mathbf{z}_j^{k+1}$ is obtained by solving $\partial \mathcal{L}'_j(\mathbf{z}_j; \mathbf{x}^{k+1}, \boldsymbol{\lambda}^k, \boldsymbol{\nu}^k)/\partial \mathbf{z}_j = 0$:

$$\begin{aligned}\mathbf{z}_j^{k+1} &= \underset{\mathbf{z}_j}{\text{minimize}}\, \mathcal{L}_j(\mathbf{z}_j; \mathbf{x}_j^{k+1}, \boldsymbol{\lambda}_{ij}^k, \boldsymbol{\nu}_{ij}^k) \\ &= \left[\frac{1}{\mu} \sum_{i \in \mathcal{N}'_j} \left(\mathbf{C}_{jj}^{\mathrm{T}} \mathbf{C}_{jj} + \mathbf{C}_{ji}^{\mathrm{T}} \mathbf{C}_{ji}\right)\right]^{-1} \\ &\quad \left[-\sum_{i \in \mathcal{N}'_j} \left(\mathbf{C}_{jj}^{\mathrm{T}} \boldsymbol{\lambda}_{ij}^k + \mathbf{C}_{ji}^{\mathrm{T}} \boldsymbol{\nu}_{ij}^k\right)\right. \\ &\quad \left. + \sum_{i \in \mathcal{N}'_j} \frac{1}{\mu} \left(\mathbf{C}_{jj}^{\mathrm{T}} \mathbf{C}_{ij} \mathbf{x}_i^{k+1} + \mathbf{C}_{ji}^{\mathrm{T}} \mathbf{C}_{ii} \mathbf{x}_i^{k+1}\right)\right]. \end{aligned} \tag{9}$$

Afterwards, each node j shares the partial information on $\mathbf{z}_j^{k+1}$ with neighboring nodes $i \in \mathcal{N}_j$ and prepares for the update of the multipliers $\boldsymbol{\lambda}_{ji}^{k+1}$ and $\boldsymbol{\nu}_{ji}^{k+1}$. The update equations are given according to ADMM method and read:

$$\boldsymbol{\lambda}_{ji}^{k+1} = \boldsymbol{\lambda}_{ji}^k - \frac{1}{\mu}\left(\mathbf{C}_{ji} \mathbf{x}_j^{k+1} - \mathbf{C}_{ii} \mathbf{z}_i^{k+1}\right) \tag{10a}$$

$$\boldsymbol{\nu}_{ji}^{k+1} = \boldsymbol{\nu}_{ji}^k - \frac{1}{\mu}\left(\mathbf{C}_{jj} \mathbf{x}_j^{k+1} - \mathbf{C}_{ij} \mathbf{z}_i^{k+1}\right). \tag{10b}$$

During the distributed estimation of the VC-DiCE algorithm, the update of $\mathbf{x}_j, \mathbf{z}_j, \boldsymbol{\lambda}_{ji}, \boldsymbol{\nu}_{ji}$ on each node j requires additional information $\mathbf{x}_i, \mathbf{z}_i, \boldsymbol{\lambda}_{ij}, \boldsymbol{\nu}_{ij}$ from neighboring nodes i over the inter-node links according to (7)-(10). In contrast to DiCE where the entire local vectors are exchanged, in VC-DiCE only partial information is transmitted. For the exchange of information on $\mathbf{x}_j$ and $\mathbf{z}_j$, only sub-vectors $\mathbf{x}_{j,i} = \mathbf{C}_{ji}\mathbf{x}_j, \mathbf{z}_{j,i} = \mathbf{C}_{ji}\mathbf{z}_j$ selected by the VDS matrices $\mathbf{C}_{ji}, i \in \mathcal{N}'_j$ are transmitted by node j to its neighboring nodes i. For the exchange of multipliers $\boldsymbol{\lambda}_{ji}, \boldsymbol{\nu}_{ji}$, sub-vectors $\mathbf{C}_{ii}\boldsymbol{\lambda}_{ji}, \mathbf{C}_{ij}\boldsymbol{\nu}_{ji}$ are transmitted from node j to nodes $i \in \mathcal{N}_j$. Note that when the VDS matrices are not all quadratic, i.e., $\text{rank}(\mathbf{C}_{ji}) < UN_{\mathrm{T}}, j \in \mathcal{J}, i \in \mathcal{N}'_j$, then the dimension of exchanged vectors in VC-DiCE decreases compared to DiCE. Therefore, a reduction on the communication effort during the distributed estimation can be achieved. The whole procedure of the VC-DiCE algorithm is summarized in Algorithm 1.

Algorithm 1 VC-DiCE algorithm

Require: Define maximum iteration K, choose initial values for Lagrange multipliers $\boldsymbol{\lambda}_{ji}^0 = \boldsymbol{\nu}_{ji}^0 = \mathbf{0}$ and variables $\mathbf{x}_j^0 = \mathbf{z}_j^0 = \mathbf{0}$ for all $j \in \mathcal{J}, i \in \mathcal{N}'_j$, $k = 0$.

Distributed processing among nodes for iteration $k+1$:

1: Each node j updates the local estimate $\mathbf{x}_j^{k+1}$ according to (7), and transmits the partial estimates $\mathbf{C}_{jj}\mathbf{x}_j^{k+1}$, $\mathbf{C}_{ji}\mathbf{x}_j^{k+1}$ to its neighboring nodes $i \in \mathcal{N}_j$;
2: Each node j updates the auxiliary variable $\mathbf{z}_j^{k+1}$ according to (9), and transmits the partial variables $\mathbf{C}_{jj}\mathbf{z}_j^{k+1}, \mathbf{C}_{ji}\mathbf{z}_j^{k+1}$ to neighboring nodes $i \in \mathcal{N}_j$;
3: Each node j updates the multipliers $\boldsymbol{\lambda}_{ji}^{k+1}$ and $\boldsymbol{\nu}_{ji}^{k+1}$ according to (10) respectively, then it sends the partial multipliers $\mathbf{C}_{ii}\boldsymbol{\lambda}_{ji}^{k+1}, \mathbf{C}_{ij}\boldsymbol{\nu}_{ji}^{k+1}$ to its neighboring nodes $i \in \mathcal{N}_j$.
4: If $k = K$, then stop, otherwise increase k by 1 and go to step 1.

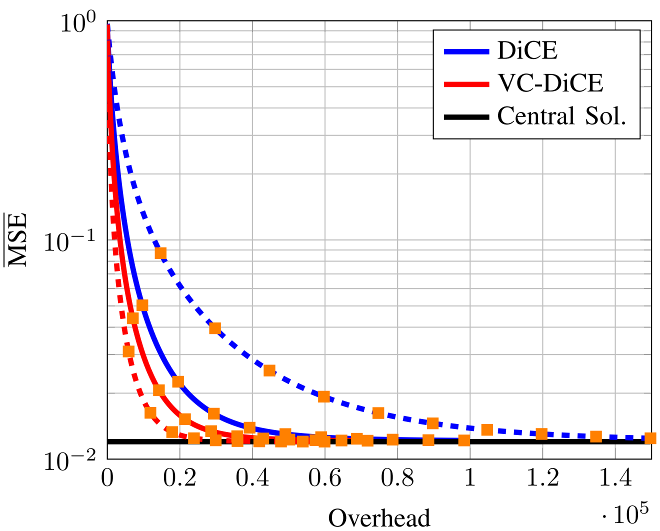

Fig. 3. $\overline{\text{MSE}}$ vs. number of total communication overhead measured in vector elements, $J = U = 5, N_{\mathrm{T}} = 1, N_{\mathrm{R}} = 2$, the network is randomly connected (—) and fully connected (····), each marker represents an increment of 50 iterations

V. Performance evaluation

In this section, we present numerical examples to corroborate the proposed algorithm. For the evaluation, we conduct simulation by means of the Monte Carlo method over 1000 trials. The channels between sources and nodes are i.i.d. complex Gaussian, and a path-loss in between is also considered, which is inversely proportional to the distance with loss exponent of 1. We consider a network of J nodes connected with a connectivity ratio r, which is defined as the ratio between actual number of node to node (N2N) links (edges $\mathcal{E}$) and the maximum number of links when all nodes are connected:

$$r = \frac{|\mathcal{E}|}{J(J-1)/2}. \tag{11}$$

To evaluate the distributed algorithms, we use the averaged mean square error ($\overline{\text{MSE}}$) as a metric to measure the estimation performance, which is defined by

$$\overline{\text{MSE}} = \frac{1}{J}\sum_{j=1}^{J} \mathrm{E}\{\|\mathbf{x}_j^k - \boldsymbol{x}\|^2\}. \tag{12}$$

In addition, the communication overhead produced during the distributed processing is also of interest. Here, we define the overhead as a number of elements in exchanged vectors (or dimension of the vector) over the N2N links.

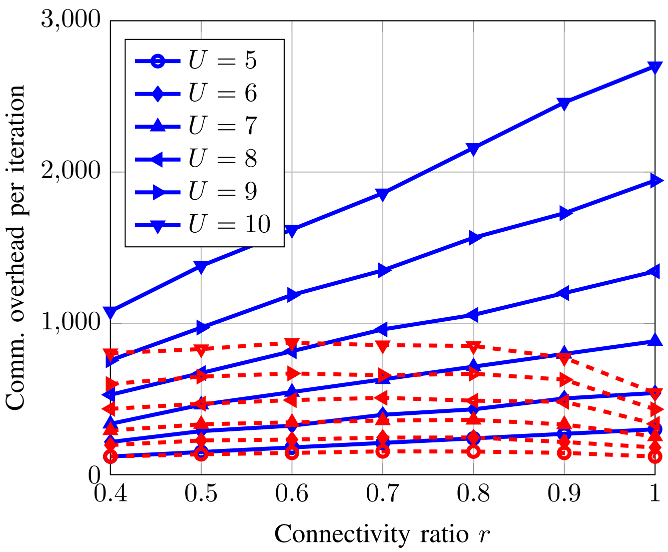

Fig. 4. Number of all N2N communication overhead per iteration measured in vector elements generated by the VC-DiCE algorithm (····) and the DiCE algorithm (—) for a network with various connectivity ratio r, $J = U = 5, .., 10, N_{\mathrm{T}} = 1, N_{\mathrm{R}} = 2$.

In Fig. 3, the convergence behavior of the distributed estimation algorithms is illustrated, where the local estimates in both algorithms converge to a centralized solution $\mathbf{x}_{\text{cen}} = (\mathbf{H}^{\mathrm{H}}\mathbf{H})^{-1}\mathbf{H}^{\mathrm{H}}\mathbf{y}, \mathbf{H} = \left[\mathbf{H}_1^{\mathrm{T}}, .., \mathbf{H}_J^{\mathrm{T}}\right]^{\mathrm{T}}$ (details given in [7]) with an increased overhead during the iterative processing. For the same estimation performance, we can observe that the VC-DiCE algorithm reduces the overhead considerably compared to the DiCE algorithm for a randomly connected network. In particular, a significant reduction on the communication effort can be achieved by VC-DiCE for a fully connected network compared to other typologies, since every node can directly communicate with other nodes, thus only a small part of information interested by the neighboring nodes needs to be transmitted.

In Fig. 4, a systematic illustration of the communication overhead of all N2N links produced by distributed algorithms per iteration with respect to varying system parameters is given. The result shows that a significant overhead can be reduced by VC-DiCE compared to the DiCE algorithm. In particular for a highly connected network ($r = 1$) with a large scale of sources ($U = 10$), VC-DiCE saves almost 80% overhead compared to the DiCE algorithm. Moreover, the N2N overhead in DiCE grows with increasing number of sources and connectivity ratio. In contrast, low overhead is produced by VC-DiCE for a large connectivity ratio, since for a highly connected network, each node can only spread a small part of the local estimate vectors to reach the consensus constraint as indicated by the examples given in section IV-B.

VI. Conclusion

In this paper, we apply a novel approach of virtual clustering to the distributed consensus-based estimation in cooperative networks for a reduction of the communication effort. One VC-based distributed estimation algorithm has been developed with exchanged information of a reduced size. The proposed VC-DiCE algorithm can significantly reduce communication effort compared to former algorithm especially for a highly connected network while keeping the estimation accuracy. For verification, the estimation performance and the communication effort of the proposed algorithm have been evaluated in numerical simulation. The results reveal that the VC-DiCE algorithm succeeds in reducing the overhead compared to the DiCE algorithm. Furthermore, it should be noticed that the virtual clustering approach developed here is not limited to one algorithm but is also applicable to other distributed algorithms for consensus-based estimation. The design of virtual cluster or VDS matrix is of importance for reducing the size of exchanged information, thus more investigation needs to be performed to the optimize virtual clustering for data selection and transmission.

Acknowledgment

The work leading to this publication was partially funded by the German Research Foundation (DFG) under grant Pa2507/1.

References

[1] C. Chen, S. Zhu, X. Guan, and X. Shen, *Wireless Sensor Networks: Distributed Consensus Estimation*, 1st ed. Springer International Publishing, Dec. 2014.
[2] Z. Chair and P. K. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-22, no. 1, pp. 98–101, Jan. 1986.
[3] A. Swami, Q. Zhao, Y. Hong, and L. Tong, *Wireless Sensor Networks: Signal Processing and Communications*. Wiley, Nov. 2007.
[4] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.
[5] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *42nd IEEE Conference on Decision and Control*, vol. 5, Dec. 2003, pp. 4997–5002.
[6] H. Zhu, A. Cano, and G. Giannakis, "Distributed Consensus-based Demodulation: Algorithms and Error Analysis," *IEEE Transactions on Wireless Communications*, vol. 9, no. 6, pp. 2044–2054, 2010.
[7] H. Paul, J. Fliege, and A. Dekorsy, "In-Network-Processing: Distributed Consensus-Based Linear Estimation," *IEEE Communications Letters*, vol. 17, no. 1, pp. 59–62, 2013.
[8] I. Schizas, A. Ribeiro, and G. Giannakis, "Consensus in Ad Hoc WSNs With Noisy Links part I: Distributed Estimation of Deterministic Signals," *IEEE Transactions on Signal Processing*, vol. 56, no. 1, pp. 350–364, Jan. 2008.
[9] H. Zhu, G. Giannakis, and A. Cano, "Distributed in-network channel decoding," *IEEE Transactions on Signal Processing*, vol. 57, no. 10, Oct. 2009.
[10] D. P. Bertsekas, A. E. Ozdaglar, and A. Nedi, *Convex analysis and optimization*, ser. Athena scientific optimization and computation series. Belmont (Mass.): Athena Scientific, 2003.
[11] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Pschel, "Distributed optimization with local domains: Applications in mpc and network flows," *IEEE Transactions on Automatic Control*, vol. 60, no. 7, pp. 2004–2009, July 2015.
[12] ——, "Admm for consensus on colored networks," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, Dec. 2012, pp. 5116–5121.
[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2010.
[14] S. Wang, "Virtual clustering based distributed consensus estimation in wireless sensor networks," in *Master Thesis*, University of Bremen, 2016.