A Factor Graph-Based Distributed Consensus Kalman Filter

Shengdi Wang, *Student Member, IEEE*, and Armin Dekorsy, *Senior Member, IEEE*

Abstract—**The Kalman filter as an effective tool to solve the state estimation problem for linear dynamic systems can be derived from a generalized perspective by applying the sum-product message passing over a factor graph. This viewpoint encourages us to visualize the state estimation problem over a network where all the nodes aspire to obtain consensus-based state estimates of a dynamic system by collecting sequential measurements over time. In this work, nodes process in a distributed and cooperative fashion and exchange Gaussian messages among neighbors resulting in a Gaussian belief propagation algorithm. We discuss and illustrate the performance of our proposed method under acyclic and cyclic network typologies.**

Index Terms—**Consensus, distributed state estimation, factor graph, Gaussian belief propagation, Kalman filtering.**

I. Introduction

DISTRIBUTED Kalman filtering for state estimation of dynamic systems is investigated in the past decade in many areas such as parameter estimation in power systems, cooperative localization and tracking in multi-agent systems and networked spacecraft formation control [1]. Varieties of these applications in a distributed fashion aim to overcome the disadvantages of centralized processing, e.g., single point of failure of central node, congestion of massive data, thereby to enhance system robustness and flexibility. In this work, we study distributed Kalman filter (KF) over a network where all nodes process data cooperatively and desire accurate consensus-based state estimates for further processing, decision making or control.

Different distributed consensus-based Kalman filtering algorithms were proposed in [2]–[6]. Among them, [2] and [3] fused data using one step average to save computation and latency but leading to suboptimal solutions. The estimation accuracy can be improved by processing iterative average consensus [7] on specific terms of Kalman filter updates presented in [4], which has a more general discussion in [5]. From a different viewpoint, the authors of [6] constructed an equivalent maximum-a-posteriori estimation problem subject to a consensus constraint on state estimate. The problem was solved distributedly with the augmented Lagrangian method and the performance outperforms [4] in terms of convergence speed and communication effort for sparse networks. In this paper, we aim to further reduce communication consumption. On the one hand, we try to achieve a fast consensus among nodes to reduce information exchange and processing delay. On the other hand, for some devices enabling point-to-point (P2P) physical connections, we compress the network redundant links, which also reduces the complexity of system design. For example, in networked satellite formation tasks, a sparse communication network means simple hardware structures of satellites for laser line-of-sight (LOS) inter-satellite links [8].

Inspired by [9], we solve the consensus-based state estimation problem in a distributed and visualized way by message passing over a factor graph (FG). Accordingly, a Gaussian belief propagation (GaBP) algorithm is derived. In [10]–[12], belief propagation (BP) algorithms over FGs for distributed state estimation were studied, where sampling-based particles are transmitted over networks increasing computation and communication burdens. For some applications, simple Gaussian approximations are sufficient to obtain acceptable estimates. Thus, this work combines consensus with Gaussian beliefs leading to a concrete algorithm as a special case of BP.

In this paper, we first concentrate on a probabilistic description of the centralized KF (CKF) over a FG. Then, visually split the FG over a communication network by introducing coupling factor nodes to fuse measurements distributedly. We analyse the performance of our algorithm by tuning the coupling parameters for acyclic and cyclic networks. Due to the limited space of this letter, only the linear state estimation is discussed. The extension to nonlinear filtering, e.g., extended Kalman filter [13] or unscented Kalman filter [14], is possible.

II. System Model

Consider the following state-space model of a discrete-time linear dynamic system, where the hidden state vector $\boldsymbol{x}_k \in \mathbb{R}^m$ at time instant k is observed by a set of J nodes:

$$\boldsymbol{x}_{k+1} = \boldsymbol{F}_k \boldsymbol{x}_k + \boldsymbol{w}_k, \tag{1}$$

$$\boldsymbol{y}_{j,k} = \boldsymbol{H}_{j,k} \boldsymbol{x}_k + \boldsymbol{v}_{j,k}. \tag{2}$$

Here, $\boldsymbol{F}_k \in \mathbb{R}^{m\times m}$, $\boldsymbol{H}_{j,k} \in \mathbb{R}^{n\times m}$ and $\boldsymbol{y}_{j,k} \in \mathbb{R}^n$ denote the state transition matrix, local measurement matrix and local measurement, respectively. The process noise $\boldsymbol{w}_k \in \mathbb{R}^m \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_k)$ and local measurement noise $\boldsymbol{v}_{j,k} \in \mathbb{R}^n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R}_{j,k})$ are time and node uncorrelated both following zero mean multivariate Gaussian distributions with covariance matrices $\boldsymbol{Q}_k$ and $\boldsymbol{R}_{j,k}$, respectively. These J nodes form a time-invariant connected communication network described by an undirected graph $\mathcal{T} = \{\mathcal{J}, \mathcal{E}\}$ with the set of nodes $\mathcal{J} = \{1, \ldots, J\}$ and the set of edges $\mathcal{E}$. For algorithm derivation, inter-node communications is assumed to be error-free here. We are aware that practical

Manuscript received April 3, 2020; revised October 16, 2020; accepted October 23, 2020. Date of publication November 6, 2020; date of current version December 1, 2020. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xun Cao. *(Corresponding author: Shengdi Wang.)*

The authors are with the Department of Communications Engineering, University of Bremen, 28359 Bremen, Germany (e-mail: wang@ant.uni-bremen.de; dekorsy@ant.uni-bremen.de).

Digital Object Identifier 10.1109/LSP.2020.3036337

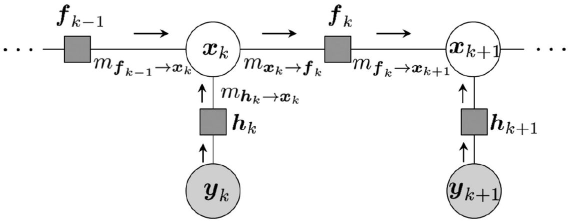

Fig. 1. A factor graph factorization depicts (4) in a Markov chain.

communication uncertainties, e.g., time delays, packet losses, link failures, influence the designed algorithm and need to be considered in future work.

To acquire a state estimate based on the global measurement data, we define a stacked vector which contains all available measurements in (2) as $\boldsymbol{y}_k = [\boldsymbol{y}_{1,k}^{\mathrm{T}}, \ldots, \boldsymbol{y}_{J,k}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{nJ}$. Thus, the global measurement model is formulated by

$$\boldsymbol{y}_k \ = \ \boldsymbol{H}_k \boldsymbol{x}_k + \boldsymbol{v}_k, \tag{3}$$

where $\boldsymbol{H}_k = [\boldsymbol{H}_{1,k}^{\mathrm{T}}, \ldots, \boldsymbol{H}_{J,k}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{nJ\times nJ}$ is the stacked measurement matrix and $\boldsymbol{v}_k = [\boldsymbol{v}_{1,k}^{\mathrm{T}}, \ldots, \boldsymbol{v}_{J,k}^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{nJ}$ is the stacked noise vector with a block diagonal covariance matrix $\boldsymbol{R}_k = \text{blkdiag}(\boldsymbol{R}_{1,k}, \ldots, \boldsymbol{R}_{J,k}) \in \mathbb{R}^{nJ\times nJ}$.

The state-space model (1) and (3) describes a system with a hidden Markov structure [9] and the joint probability density function (pdf) of variables $\boldsymbol{x}$ and $\boldsymbol{y}$ can be factorized as

$$p(\boldsymbol{x}_0, \cdots, \boldsymbol{x}_K, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_K) = p(\boldsymbol{x}_0) \prod_{k=1}^{K} p(\boldsymbol{x}_k|\boldsymbol{x}_{k-1}) p(\boldsymbol{y}_k|\boldsymbol{x}_k), \tag{4}$$

where $p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) = \mathcal{N}(\boldsymbol{F}_k\boldsymbol{x}_k, \boldsymbol{Q}_k)$ is the state transition pdf and $p(\boldsymbol{y}_k|\boldsymbol{x}_k) = \mathcal{N}(\boldsymbol{H}_k\boldsymbol{x}_k, \boldsymbol{R}_k)$ is the likelihood pdf, due to model (1) and (3) respectively. Then, the factorization of joint pdf in (4) can be depicted using a FG shown in Fig. 1, where $\boldsymbol{f}_k$ and $\boldsymbol{h}_k$ are factor nodes indicating state transition and likelihood function, respectively. Thus, messages are Gaussian pdfs passing over this FG following the sum-product rule [9]:

message from function f **to variable** x:

$$m_{f\to x}(x) = \int_{X\backslash x} f(X) \prod_{y\in n(f)\backslash x} m_{y\to f}(y) d(X\backslash x) \tag{5}$$

message from variable x **to function** f:

$$m_{x\to f}(x) = \prod_{h\in n(x)\backslash f} m_{h\to x}(x) \tag{6}$$

where $n(v)$ denotes the set of neighbors of a given node v in a FG and $X = n(f)$ is the set of arguments of the function f. $X\backslash x$ denotes the set of elements in X without x.

III. Approach

Before we derive a distributed algorithm, we first focus on the CKF over FG. At each time k, measurements of all nodes are fused at a variable node $\boldsymbol{y}_k$. Then, forward propagating messages in Fig. 1 can be calculated recursively based on (5) and (6) over the FG providing the forms of posterior marginal distribution $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ on state $\boldsymbol{x}_k$ and predicted distribution $p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_{1:k})$ on state $\boldsymbol{x}_{k+1}$ given measurements up to k as:

$$p_{k|k}(\boldsymbol{x}_k) = p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) \propto m_{\boldsymbol{x}_k\to\boldsymbol{f}_k} = m_{\boldsymbol{f}_{k-1}\to\boldsymbol{x}_k} m_{\boldsymbol{h}_k\to\boldsymbol{x}_k}, \tag{7}$$

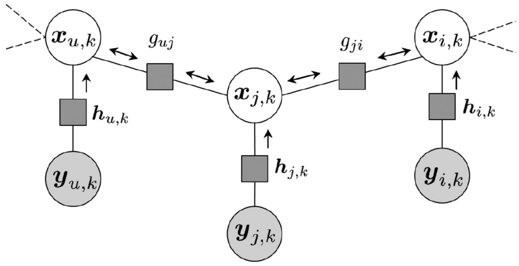

Fig. 2. Message passing over a factor graph representing measurement fusion at each time instant k.

$$p(\boldsymbol{x}_{k+1}|\boldsymbol{y}_{1:k}) \propto m_{\boldsymbol{f}_k\to\boldsymbol{x}_{k+1}} = \int p(\boldsymbol{x}_{k+1}|\boldsymbol{x}_k) m_{\boldsymbol{x}_k\to\boldsymbol{f}_k} d\boldsymbol{x}_k. \tag{8}$$

By defining $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k-1}) = \mathcal{N}(\hat{\boldsymbol{x}}_k^-, \boldsymbol{P}_k^-)$ and $p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k}) = \mathcal{N}(\hat{\boldsymbol{x}}_k, \boldsymbol{P}_k)$, (7) and (8) decide the mean and covariance of the posterior and predicted distribution. The resulting updates are identical to the conventional KF [9] with the state estimate at each time k: $\hat{\boldsymbol{x}}_k = \mathrm{E}\{\boldsymbol{x}_k|\boldsymbol{y}_{1:k}\} = \arg\max_{\boldsymbol{x}_k} p(\boldsymbol{x}_k|\boldsymbol{y}_{1:k})$ [6].

To design a distributed algorithm, the key issue is how to collect measurements over a network, called *measurement fusion*. Inspired by [15], [16], we introduce a coupling factor node g_{ji} to express the relation of two local copied state variables $\boldsymbol{x}_{j,k}$ and $\boldsymbol{x}_{i,k}$ of the global one at neighboring node j and i, where $\forall j \in \mathcal{J}, i \in \mathcal{N}_j$ and $\mathcal{N}_j$ denotes the neighboring set of j. We define g_{ji} as an exponential function to make the calculation of Gaussian messages mathematically tractable:

$$g_{ji}(\boldsymbol{x}_{j,k}, \boldsymbol{x}_{i,k}) = \exp\left\{ -\frac{\beta}{2} \|\boldsymbol{x}_{j,k} - \boldsymbol{x}_{i,k}\|_2^2 \right\}, \tag{9}$$

with coupling parameter $\beta > 0$. Thus, the distributed *measurement fusion* can be conducted by message passing over a communication network at each time k without any routing protocol as shown in Fig. 2, where $\{u, i\} \in \mathcal{N}_j$. Along with increasing β, the neighboring variables $\boldsymbol{x}_{j,k}$ and $\boldsymbol{x}_{i,k}$ become more and more correlated [15], which promotes a consensus on state distributions in the network. When $\beta \to \infty$, we obtain $g_{ji}(\boldsymbol{x}_{j,k}, \boldsymbol{x}_{i,k}) \propto \delta(\boldsymbol{x}_{j,k} - \boldsymbol{x}_{i,k})$ and thus distributions on local state variables conditioned on all measurements at time k will be the same over the whole network, i.e., $p(\boldsymbol{x}_{j,k}|\boldsymbol{y}_{1,k}, \ldots, \boldsymbol{y}_{J,k}) = p(\boldsymbol{x}_{i,k}|\boldsymbol{y}_{1,k}, \ldots, \boldsymbol{y}_{J,k})$, which are identical to the central one $p(\boldsymbol{x}_k|\boldsymbol{y}_k)$. By defining this coupling factor, each node has its own processing Markov chain as Fig. 1 and cooperates with neighboring nodes by information exchange as Fig. 2. Therefore, the posterior marginal distribution on $\boldsymbol{x}_{j,k}$ at node $j \in \mathcal{J}$ at time k is

$$\begin{aligned} p_{k|k}(\boldsymbol{x}_{j,k}) &\propto m_{\boldsymbol{f}_{k-1}\to\boldsymbol{x}_{j,k}} m_{\boldsymbol{h}_{j,k}\to\boldsymbol{x}_{j,k}} \prod_{u\in\mathcal{N}_j} m_{g_{uj}\to\boldsymbol{x}_{j,k}} \\ &\propto \mathcal{N}(\boldsymbol{\mu}_{j,k}, \boldsymbol{P}_{j,k}) \prod_{u\in\mathcal{N}_j} \mathcal{N}(\boldsymbol{\mu}_{uj}, \boldsymbol{\Lambda}_{uj}^{-1}) \propto \mathcal{N}(\hat{\boldsymbol{x}}_{j,k}, \boldsymbol{\Lambda}_{j,k}^{-1}), \end{aligned} \tag{10}$$

where an augmented product term shows up compared to (7) representing the Gaussian incoming messages $m_{g_{uj}\to\boldsymbol{x}_{j,k}}$ with mean $\boldsymbol{\mu}_{uj}$ and precision $\boldsymbol{\Lambda}_{uj}$ transmitted by neighbors. The term $\mathcal{N}(\boldsymbol{\mu}_{j,k}, \boldsymbol{P}_{j,k})$ indicates the local KF updated distribution on

$\boldsymbol{x}_{j,k}$ based on system model (1) and (2) as:

$$\begin{aligned} & m_{\boldsymbol{f}_{k-1}\to\boldsymbol{x}_{j,k}} m_{\boldsymbol{h}_{j,k}\to\boldsymbol{x}_{j,k}} \\ & \propto \mathcal{N}(\hat{\boldsymbol{x}}^-_{j,k}, \boldsymbol{P}^-_{j,k})\mathcal{N}(\boldsymbol{H}^{-1}_{j,k}\boldsymbol{y}_{j,k}, (\boldsymbol{H}^T_{j,k}\boldsymbol{R}^{-1}_{j,k}\boldsymbol{H}_{j,k})^{-1}) \\ & \propto \mathcal{N}(\boldsymbol{\mu}_{j,k}, \boldsymbol{P}_{j,k}). \end{aligned} \tag{11}$$

By calculating the products of two Gaussian pdfs in (11), $\boldsymbol{P}_{j,k}$ and $\boldsymbol{\mu}_{j,k}$ can be obtained by

$$\boldsymbol{P}_{j,k} = \left((\boldsymbol{P}^-_{j,k})^{-1} + \boldsymbol{H}^T_{j,k}\boldsymbol{R}^{-1}_{j,k}\boldsymbol{H}_{j,k}\right)^{-1}, \tag{12}$$

$$\boldsymbol{\mu}_{j,k} = \boldsymbol{P}_{j,k}\left((\boldsymbol{P}^-_{j,k})^{-1}\hat{\boldsymbol{x}}^-_{j,k} + \boldsymbol{H}^T_{j,k}\boldsymbol{R}^{-1}_{j,k}\boldsymbol{y}_{j,k}\right). \tag{13}$$

To calculate the Gaussian posterior marginal distribution (10), parameters $\boldsymbol{\mu}_{uj}$ and $\boldsymbol{\Lambda}_{uj}$ are required at node j. Next, we first derive the message from variable node $\boldsymbol{x}_{j,k}$ to its neighboring factor node g_{ji} as

$$\begin{aligned} m_{\boldsymbol{x}_{j,k}\to g_{ji}} &= m_{\boldsymbol{h}_{j,k}\to\boldsymbol{x}_{j,k}} \prod_{u\in\mathcal{N}_j\backslash i} m_{g_{uj}\to\boldsymbol{x}_{j,k}} \\ & \propto \mathcal{N}(\boldsymbol{H}^{-1}_{j,k}\boldsymbol{y}_{j,k}, (\boldsymbol{H}^T_{j,k}\boldsymbol{R}^{-1}_{j,k}\boldsymbol{H}_{j,k})^{-1}) \prod_{u\in\mathcal{N}_j\backslash i} \mathcal{N}(\boldsymbol{\mu}_{uj}, \boldsymbol{\Lambda}^{-1}_{uj}). \end{aligned} \tag{14}$$

The message from factor node g_{ji} to variable node $\boldsymbol{x}_{i,k}$ is

$$\begin{aligned} m_{g_{ji}\to\boldsymbol{x}_{i,k}} &= \int g_{ji}(\boldsymbol{x}_{j,k}, \boldsymbol{x}_{i,k})\, m_{\boldsymbol{x}_{j,k}\to g_{ji}} d\boldsymbol{x}_{j,k} \\ & \propto \mathcal{N}(\boldsymbol{\mu}_{ji}, \boldsymbol{\Lambda}^{-1}_{ji}). \end{aligned} \tag{15}$$

Then, we substitute (9) and (14) into (15) resulting in an iterative GaBP loop for *measurement fusion* by initializing $\boldsymbol{\Lambda}^0_{uj}$ and $\boldsymbol{\mu}^0_{uj}$. Note that the initialization may influence the convergence speed of GaBP and hence can be chosen properly. Each node $j \in \mathcal{J}$ calculates parameters $\boldsymbol{\Lambda}_{ji}$ and $\boldsymbol{\mu}_{ji}$ in an inner iteration l:

$$\boldsymbol{\Lambda}^l_{ji} = \left[(\boldsymbol{H}^T_{j,k}\boldsymbol{R}^{-1}_{j,k}\boldsymbol{H}_{j,k} + \sum_{u\in\mathcal{N}_j\backslash i}\boldsymbol{\Lambda}^{l-1}_{uj})^{-1} + \beta^{-1}\boldsymbol{I}\right]^{-1}, \tag{16}$$

$$\begin{aligned} \boldsymbol{\mu}^l_{ji} &= \left(\boldsymbol{H}^T_{j,k}\boldsymbol{R}^{-1}_{j,k}\boldsymbol{H}_{j,k} + \sum_{u\in\mathcal{N}_j\backslash i}\boldsymbol{\Lambda}^{l-1}_{uj}\right)^{-1} \\ & \times \left(\boldsymbol{H}^T_{j,k}\boldsymbol{R}^{-1}_{j,k}\boldsymbol{y}_{j,k} + \sum_{u\in\mathcal{N}_j\backslash i}\boldsymbol{\Lambda}^{l-1}_{uj}\boldsymbol{\mu}^{l-1}_{uj}\right), \end{aligned} \tag{17}$$

and transmits both parameters to neighboring nodes $i \in \mathcal{N}_j$. After a certain number of inner iteration L, each node j receives $\boldsymbol{\Lambda}^L_{uj}$ and $\boldsymbol{\mu}^L_{uj}$ from neighboring nodes $u \in \mathcal{N}_j$. Thus, the precision $\boldsymbol{\Lambda}_{j,k}$ and mean $\hat{\boldsymbol{x}}_{j,k}$ of the posterior marginal distribution on $\boldsymbol{x}_{j,k}$ can be computed by (10) and we obtain

$$\boldsymbol{\Lambda}_{j,k} = \boldsymbol{P}^{-1}_{j,k} + \sum_{u\in\mathcal{N}_j}\boldsymbol{\Lambda}^L_{uj}, \tag{18}$$

$$\hat{\boldsymbol{x}}_{j,k} = \boldsymbol{\Lambda}^{-1}_{j,k}\left(\boldsymbol{P}^{-1}_{j,k}\boldsymbol{\mu}_{j,k} + \sum_{u\in\mathcal{N}_j}\boldsymbol{\Lambda}^L_{uj}\boldsymbol{\mu}^L_{uj}\right). \tag{19}$$

Then, the predict step is performed by calculating message:

$$\begin{aligned} m_{\boldsymbol{f}_k\to\boldsymbol{x}_{j,k+1}} &= \int p(\boldsymbol{x}_{j,k+1}|\boldsymbol{x}_{j,k}) m_{\boldsymbol{x}_{j,k}\to\boldsymbol{f}_k} d\boldsymbol{x}_{j,k} \\ &= \int \mathcal{N}(\boldsymbol{F}_k\boldsymbol{x}_{j,k}, \boldsymbol{Q}_k)\mathcal{N}(\hat{\boldsymbol{x}}_{j,k}, \boldsymbol{\Lambda}^{-1}_{j,k}) d\boldsymbol{x}_{j,k} \\ & \propto \mathcal{N}(\hat{\boldsymbol{x}}^-_{j,k+1}, \boldsymbol{P}^-_{j,k+1}) \end{aligned}$$

and we get the predicted mean and covariance matrix at node j for the next time instant $k+1$:

$$\hat{\boldsymbol{x}}^-_{j,k+1} = \boldsymbol{F}_k\hat{\boldsymbol{x}}_{j,k}, \tag{20}$$

$$\boldsymbol{P}^-_{j,k+1} = \boldsymbol{F}_k\boldsymbol{\Lambda}^{-1}_{j,k}\boldsymbol{F}^T_k + \boldsymbol{Q}_k. \tag{21}$$

The general algorithm is summarized in Algorithm 1.

Algorithm 1: Factor Graph-Based Distributed Consensus Kalman Filter.

```
Initialize: for all $j \in \mathcal{J}$, $\hat{\boldsymbol{x}}^-_{j,1} = \mathbf{0}$, $\boldsymbol{P}^-_{j,1} = \boldsymbol{\Pi}$, $\beta$
for $k = 1, \cdots K$, node $j$ do
    Initialize: $l = 0$, $\boldsymbol{\mu}^0_{uj}$, $\boldsymbol{\Lambda}^0_{uj}$, $\forall u \in \mathcal{N}_j$
    % distributed measurement fusion
    while(GaBP not converge)do
        $l = l + 1$
        calculate (16) and (17) and transmit $\boldsymbol{\Lambda}^l_{ji}$, $\boldsymbol{\mu}^l_{ji}$ to
        neighbors $i \in \mathcal{N}_j$
    end while
    calculate (18) and (19) for posterior distribution
    % local prediction
    calculate (20) and (21) for predicted distribution
end for
```

IV. Discussions on Network Topology and Communication Overhead

A. Acyclic Network Topology

An acyclic network is a tree structure without any loop. We will show that the mean and precision on state estimates are consensus-based and identical to the central ones when $\beta \to \infty$ by performing $L = D$ iterations. Here, D is defined as the network diameter describing the greatest distance between any pair of nodes. With $\beta \to \infty$, the term $\beta^{-1}\boldsymbol{I}$ in (16) vanishes. With any initialization $\boldsymbol{\mu}^0_{uj}$ and $\boldsymbol{\Lambda}^0_{uj}$, it can be derived that after D-th inner iteration, each node j gathers all messages on measurements $\boldsymbol{H}^T_{i,k}\boldsymbol{R}^{-1}_{i,k}\boldsymbol{H}_{i,k}$ and $\boldsymbol{H}^T_{i,k}\boldsymbol{R}^{-1}_{i,k}\boldsymbol{y}_{i,k}$, $i \in \mathcal{J}\backslash j$, from neighbors $u \in \mathcal{N}_j$ over the acyclic FG by terms

$$\sum_{u\in\mathcal{N}_j}\boldsymbol{\Lambda}^D_{uj} = \sum_{i\in\mathcal{J}\backslash j}\boldsymbol{H}^T_{i,k}\boldsymbol{R}^{-1}_{i,k}\boldsymbol{H}_{i,k}, \tag{22}$$

$$\sum_{u\in\mathcal{N}_j}\boldsymbol{\Lambda}^D_{uj}\boldsymbol{\mu}^D_{uj} = \sum_{i\in\mathcal{J}\backslash j}\boldsymbol{H}^T_{i,k}\boldsymbol{R}^{-1}_{i,k}\boldsymbol{y}_{i,k}. \tag{23}$$

Substituting (12), (13), (22), (23) into (18) and (19), we get

$$\boldsymbol{\Lambda}_{j,k} = (\boldsymbol{P}^-_{j,k})^{-1} + \sum_{i\in\mathcal{J}}\boldsymbol{H}^T_{i,k}\boldsymbol{R}^{-1}_{i,k}\boldsymbol{H}_{i,k}, \tag{24}$$

$$\hat{\boldsymbol{x}}_{j,k} = \boldsymbol{\Lambda}^{-1}_{j,k}\left((\boldsymbol{P}^-_{j,k})^{-1}\hat{\boldsymbol{x}}^-_{j,k} + \sum_{i\in\mathcal{J}}\boldsymbol{H}^T_{i,k}\boldsymbol{R}^{-1}_{i,k}\boldsymbol{y}_{i,k}\right), \tag{25}$$

which are identical to the CKF results [6] with prior knowledge as the central case $\boldsymbol{P}^-_{j,k} = \boldsymbol{P}^-_k$ and $\hat{\boldsymbol{x}}^-_{j,k} = \hat{\boldsymbol{x}}^-_k$, $\forall j$. Under $\beta \to \infty$, we can also explain (17) as an iterative weighted averaging on the means of incoming messages at node j and the weights are decided by the corresponding precisions.

B. Cyclic Network Topology

For cyclic networks, we tune the parameter β as a relaxation of consensus on state variables to make GaBP converge. The central solution can not be achieved, because the term $\beta^{-1}\boldsymbol{I}$ in (16) leads to a small precision matrix which further influences the accuracy of the mean in (17). The choice of β offers a trade-off between fast convergence (small β) and accurate estimate (large β) of GaBP. However, the convergence analysis and the choice of optimal β are still open. For iterative KF updates, a poor result of the inner iteration means estimates are not effectively propagated along with the time. This may make our algorithm diverge. To overcome this limit, a minimum spanning tree (MST) algorithm, e.g., [17], [18], can be applied as a pre-processing step to break loops in a network and obtain the smallest network diameter. Thus, our algorithm converges fast towards the central solution meanwhile reducing redundant inter-node communication effort in P2P transmission. For time-invariant networks, a pre-processing MST is performed only once. Here, we assume that radio propagation over all inter-node links has no difference. Definitely, communication qualities such as the link path-loss effect can be considered associate with link weights in MST. Reminding (9), a broken link can be regarded as setting $\beta = 0$ for this specific link, which means there is no coupling between state densities of two nodes. If $\beta = 0$ for all links, all communication will be suppressed among nodes, i.e., nodes are isolated.

C. Communication Overhead

We assume that the communication network $\mathcal{T}$ equips with P2P inter-node links with a unicast transmission scheme. Transmitted scalars per time instant k are counted as communication overhead (CommOH) during the algorithm procedure. In general, its value is $2|\mathcal{E}|KL(m+m^2)$ where $|\mathcal{E}|$ is the number of edges of $\mathcal{T}$. For acyclic networks, the CommOH is limited as $2|\mathcal{E}|KD(m+m^2)$ to reach the central optimal solution. For cyclic networks, by applying the MST algorithm to reduce network redundant links, the CommOH is compressed into $2|\mathcal{E}'|KD(m+m^2)$ where $|\mathcal{E}'|$ is the number of edges of simplified tree network $\mathcal{T}' = \{\mathcal{J}, \mathcal{E}'\}$ with $|\mathcal{E}'| < |\mathcal{E}|$.

V. Simulation Results

In this section, we evaluate the performance of our proposed algorithm compared to the optimal consensus-based algorithms in [4] and [6]. A process model with 2-dimensional state is considered with matrices $\boldsymbol{F}_k = [0.992, -0.1247; 0.1247, 0.992]$ and $\boldsymbol{Q}_k = 0.5\boldsymbol{I}_2$, $\forall k$. A set of $J = 16$ nodes in a network $\mathcal{T}$ takes measurements of the system state with local measurement matrix $\boldsymbol{H}_{j,k}$ chosen to be either [1, 0] or [0, 1] randomly and a degraded scalar-valued noise variance $\boldsymbol{R}_{j,k} = 1$. The key performance indicator is the average mean square error aMSE $= \frac{1}{J}\sum_{j=1}^{J} \mathrm{E}\{\|\boldsymbol{x}_k - \hat{\boldsymbol{x}}_{j,k}\|^2\}$ approximated by averaging over M Monte Carlo simulation runs with different realizations on $\mathcal{T}$, $\boldsymbol{w}_k$, $\boldsymbol{v}_{j,k}$ and $\boldsymbol{H}_{j,k}$.

We first simulate aMSE performance of different algorithms for $M = 1000$ randomly generated networks $\mathcal{T}$ along with $K = 50$ time instants. For our proposed algorithm, in each run, we apply MST to construct a tree network $\mathcal{T}'$ with a diameter D. Note that in algorithms [4] and [6], each node transmits the same amount of scalars as our proposed one per inner iteration l. One example of networks $\mathcal{T}$ and $\mathcal{T}'$ is shown in Fig. 3 (left). By fixing $L = D$ for all algorithms and $\beta \to \infty$, the results of aMSE w.r.t. CommOH are presented in Fig. 3 (middle). We observe that the proposed algorithm outperforms the others gaining the highest estimation accuracy only with almost 50% communication cost, while the others have not yet converged to the central solution at each time instant. The steady-state consensus status over nodes is illustrated in Fig. 3 (right), indicating that a consensus-based centralized result has been achieved among nodes by applying our proposed algorithm.

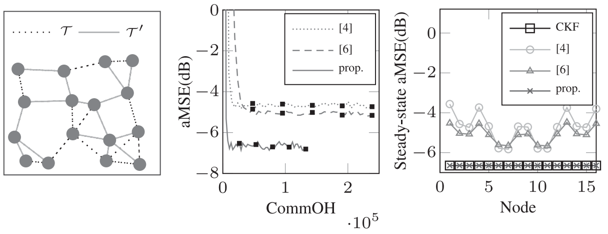

Fig. 3. (Left) an example of $\mathcal{T}$ and $\mathcal{T}'$; (middle) aMSE w.r.t. CommOH, each marker indicates an increment of 10 time instants; (right) steady-state aMSE of different nodes.

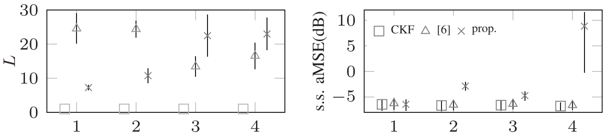

Fig. 4. Mean ± standard deviation of required L (left) and steady-state aMSE (right) when algorithms converge under 4 cases of networks, i.e., 1 = tree, 2 = ring, 3 = loops, 4 = loops + tree.

The overall aMSE performance and CommOH highly depends on the number of inner iteration L. Next, we study the required L for each algorithm to reach the best performance for 4 different cases of randomly generated networks, i.e., tree, ring, loops and loops+tree. Note that here the loops network means each node at least has two neighbors. We simulate the steady-state aMSE w.r.t. the increasing L. The simulation is terminated when the absolute value of difference between steady-state aMSEs at $L-1$ and L is less than 10^{-3}. Then the current L is recorded. For cyclic networks, we suggest $\beta = 1$ as the same level of $\boldsymbol{R}_{j,k}^{-1}$ for the proposed algorithm to achieve acceptable performance. The resulting mean and standard deviation of L and the corresponding steady-state aMSE under $M = 50$ for each network case are indicated in Fig. 4. As observed from the results, our proposed algorithm gains the best performance with the lowest CommOH under tree network. For ring, a trade-off between estimation accuracy and CommOH should be taken into account to choose a proper algorithm. For highly cyclic networks, our recommendation is to perform MST to break loops and further save communication cost, which is suitable for time-invariant networks or pre-deployed P2P communication links.

VI. Conclusion

In this work, we propose a distributed consensus-based KF algorithm by applying the GaBP over a FG. It consumes very low communication effort meanwhile converges to the optimal CKF solution for acyclic networks. Convergence and CommOH are also discussed for different networks. For some cyclic networks, it is possible to relax consensus to achieve a fast convergence. Generally, we can apply a pre-processing MST algorithm to break loops of a cyclic network, which further reduces communication cost. In future research, we intend to consider practical communication uncertainties to study their influences on the proposed algorithm.

REFERENCES

[1] M. S. Mahmoud and H. M. Khalid, “Distributed Kalman filtering: A bibliographic review,” *IET Control Theory Appl.*, vol. 7, no. 4, pp. 483–501, Mar. 2013.

[2] R. Olfati-Saber, “Distributed Kalman filtering for sensor networks,” in *Proc. IEEE Conf. Decis. Control*, Dec. 2007, pp. 5492–5498.

[3] F. S. Cattivelli and A. H. Sayed, “Diffusion strategies for distributed Kalman filtering and smoothing,” *IEEE Trans. Autom. Control*, vol. 55, no. 9, pp. 2069–2084, Sep. 2010.

[4] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, “Information weighted consensus filters and their application in distributed camera networks,” *IEEE Trans. Autom. Control*, vol. 58, no. 12, pp. 3112–3125, Dec. 2013.

[5] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano, “Consensus-based linear and nonlinear filtering,” *IEEE Trans. Autom. Control*, vol. 60, no. 5, pp. 1410–1415, May 2015.

[6] S. Wang, H. Paul, and A. Dekorsy, “Distributed optimal consensus-based Kalman filtering and its relation to MAP estimation,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2018, pp. 3664–3668.

[7] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.

[8] Z. Sodnik, B. Furch, and H. Lutz, “Optical intersatellite communication,” *IEEE J. Sel. Topics Quantum Electron.*, vol. 16, no. 5, pp. 1051–1057, Sep. 2010.

[9] 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.

[10] H. Wymeersch, J. Lien, and M. Z. Win, “Cooperative localization in wireless networks,” *Proc. IEEE*, vol. 97, no. 2, pp. 427–450, Feb. 2009.

[11] P. Chavali and A. Nehorai, “Distributed power system state estimation using factor graphs,” *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2864–2876, Jun. 2015.

[12] V. Savic, H. Wymeersch, and S. Zazo, “Belief consensus algorithms for fast distributed target tracking in wireless sensor networks,” *Signal Process.*, vol. 95, no. 3, pp. 149–160, Feb. 2014.

[13] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1979.

[14] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation,” *Proc. IEEE*, vol. 92, no. 3, pp. 401–422, Mar. 2004.

[15] C. C. Moallemi and B. Van Roy, “Consensus propagation,” *IEEE Trans. Inf. Theory*, vol. 52, no. 11, pp. 4753–4766, Nov. 2006.

[16] T. Buchgraber and D. Shutin, “Distributed variational sparse Bayesian learning for sensor networks,” in *Proc. IEEE Mach. Learn. Signal Process.*, Sep. 2012, pp. 1–6.

[17] R. C. Prim, “Shortest connection networks and some generalizations,” *Bell Syst. Tech. J.*, vol. 36, no. 6, pp. 1389–1401, Nov. 1957.

[18] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem,” *Proc. Amer. Math. Soc.*, vol. 7, no. 1, pp. 48–50, 1956.