Energy-Efficient Diffusion Kalman Filtering for Multi-agent Networks in IoT

Azam Khalili, Vahid Vahidpour, Amir Rastegarnia, Wael M. Bazzi, and Saeid Sanei, *Senior Member, IEEE*

***Abstract*—Increasing the energy efficiency of an Internet of Things (IoT) system is a major challenge for its successful implementation. To reduce the computation and storage burden and enhance the efficiency of traditional IoT, an energy-efficient diffusion-based algorithm for state estimation in multi-agent networks is proposed in this paper. In the proposed algorithm (referred to as reduced-link diffusion Kalman filter (RL-diffKF)) the nodes (agents) can communicate only with a fraction of their neighbors and each node runs a local Kalman filter to estimate the state of a linear dynamic system. This algorithm results in a significant reduction in communication cost during both adaptation and aggregation processes albeit at the expense of possible degradation in the network performance. To justify the stability and convergence of the RL- diffKF algorithm, an in-depth analysis of the performance is reported. We also consider the problem of optimal selection of combination weights and use the idea of minimum variance estimation to analytically derive the adaptive combiners. The theoretical findings are verified through numerical simulations.**

***Index Terms*—Communication cost, diffusion strategy, IoT, Kalman filtering, reduced-link, state estimation.**

I. Introduction

The IoT refers to a system of interrelated physical devices and their virtual implementations that are able to receive and diffuse data over an Internet-like network [1]. Such a framework is based on sensing, communication, networking, and information processing technologies. The wireless IoT devices enable these physical objects to see, hear, think, and execute tasks in a cooperative manner to diffuse information across the network and to coordinate decisions [2]. There are numerous applications benefited from IoT such as industrial automation [3], intelligent agriculture [4], environmental monitoring [5], and mobile healthcare [6]

Distributed state estimation problem appears in many applications such as power systems, sensor networks, smart grid, and networked navigation systems which are parts of an IoT system. So far, different algorithms have been reported in the literature for solving this problem. Distributed Kalman filtering has been extensively used in many IoT applications, including target localization and tracking [7], [8], privacy protection for cloud platform in IoT [9], [10], robust estimation for asynchronous networks [11], outlier detection [12], [13], and state estimation for wireless power transfer systems [14]. This wide range of applications stems from the fact that dynamical performance of the physical system controlled by IoT can be implemented mathematically by a state-space model.

In this work, we focus on the diffusion Kalman filtering as it has been shown that the diffusion-based solutions exhibit superior tracking and stability performance. A successful diffusion implementation requires a connected network and a collaboration mechanism in which, only the agents from a predefined neighborhood are allowed to exchange information with each other [15]–[20]. In this framework, the information is collected by all the agents, which use a local adaptive algorithm, and are scattered through an online sharing mechanism that ripples through the network frequently. This collaboration mechanism makes a diffusion based algorithm scalable and increases its robustness to node and link failures [15]. Diffusion of the parameter vectors among nodes, however; requires a high amount of communication resources. Additionally, in a multi-agent network (e.g. wireless sensor network), the degree of node (i.e. the number of connections it has to other nodes) also plays an important role in the overall communications. Therefore, in order to keep the benefits of diffusion networks, it is necessary to modify their collaboration mechanism, and develop energy-efficient diffusion networks with reduced communication cost [21].

Efforts have been made to reduce the communication burden in diffusion networks. In [22]–[27], some partial diffusion strategies have been proposed where a subset of the entries of intermediate estimate vectors are allowed to share among the neighbors. In [28], the Krylov subspace projection method has been used to perform dimensionality reduction. Quantization is another useful technique to reduce the bandwidth usage [29]. In this method, the local estimates are quantized before being shared among the nodes. In [30], the nodes transmit the sign of innovation (SOI) sequence in the distributed state estimation framework. In [31], a probability is assigned to the link between every two agents and then a probabilistic diffusion adaptation considered when the network topology is dynamic. Takahashi *et al.* in [32] proposed an algorithm to control the link probabilities by minimizing the estimation error in order to improve the estimation performance [31]. In [33] and [34] every agent is allowed to select a subset of its neighbors to aggregate the data based on some extra information that portrays the quality of the nodes. Clearly, these approaches require extra communication to select the neighbors for sharing the data. In [35], a novel method is proposed to reduce the communication overhead, where the

A. Khalili, V. Vahidpour, and A. Rastegarnia are with the Department of Electrical Engineering, Malayer University, Malayer 65719-95863, Iran (email: a-khalili@malayeru.ac.ir; v.vahidpour@stu.malayeru.ac.ir; rastegarnia@malayeru.ac.ir).

W. M. Bazzi is with the Electrical and Computer Engineering Department, American University in Dubai Dubai, United Arab Emirates, (email: wbazzi@aud.edu).

S. Sanei is with the School of Science and Technology, Nottingham Trent University, Clifton Lane, Nottingham, U.K. (e-mail: saeid.sanei@ntu.ac.uk)

Digital Object Identifier 2021/XX

nodes are allowed to dynamically update their estimates and diffuse after each update. The given algorithms in [36] and [37] are able to exploit the sparsity in the observation model and reduce the communication cost by resorting to a convex regularization method.

This paper proposes a novel reduced-link diffusion Kalman filter (RL-diffKF) algorithm to further release the communication density of diffusion Kalman filtering (diffKF). In this algorithm, each individual node updates its local estimate and sends it to only a subset of its neighbors. Moreover, total amount of inter-node communication is effectively decreased in the network with less performance degradation compared to the diffKF algorithm. It is worth mentioning that the RL-diffKF algorithm has not been previously proposed in [21]. Indeed, in [21] a reduced-communication diffusion LMS algorithm has been developed which is obviously different from our proposed algorithm since the given algorithm in [21] considers a linear regression model, whereas in this paper we assume a linear state-space model. The algorithm in [21] uses the LMS algorithm as the learning rule for the agents; whereas here the agents use Kalman filter for adaptation and learning.

A reduced-communication based diffKF algorithm has been reported in [38], this algorithm differs from our proposed approach in several ways. First, the algorithm in [38] is derived by using the information form of the Kalman filter, while our distributed schemes are motivated by time and measurement frameworks. Our work relies on a modified version of diffKF where the nodes do not exchange the information matrices and pseudo-observation vectors with their neighbors in the incremental step and the algorithm relies on the transmission of intermediate estimates from selected immediate neighbors. It is noteworthy to mention that sending and receiving the information matrices and pseudo-observation vectors involve a significant amount of communications between individual nodes. Hence, our main contributions are as follows:

- The RL-diffKF algorithm is developed and its performance is verified in terms of different metrics (including steady-state mean-square deviation (MSD), communication saving) via numerical simulations.
- Under some simplifying assumptions (that are traditionally adopted in the analysis of diffusion-based algorithms) the required conditions for stability of the RL-diffKF algorithm are derived.
- To predict the performance of RL-diffKF algorithm at the steady-state, a theoretical expression is derived for the MSD metric. It is demonstrated that the results from the theoretical analysis (i.e. theoretical MSD values) match the results obtained by simulations.
- It is verified (theoretically) that the RL-diffKF algorithm facilitates the usage of network transmission resources and provides a trade-off between estimation-accuracy and communication-load.
- In order to improve the estimation performance of RL-diffKF algorithm, the combination coefficients are used as design parameters. More specifically, an adaptive combination rule is proposed in which the combination coefficients are adjusted (based on the data statistics) throughout the learning process. It is shown that RL-diffKF with the proposed adaptive weights outperforms those with static combiners (providing smaller steady-state MSD values).

In Section II, the work background is presented. The proposed RL-diffKF algorithm is formulated in Section III. The performance analyses are examined in Section IV. In Section VI, the problem of controlling the combination weights is formulated and an adaptive combination rule is established. Simulation results are presented in Section VII, and the paper is concluded in Section VIII.

Notations: Small boldface letters are adopted for vectors and bold capital letters for matrices. Normal font letters denote scalars. The transposition operator is denoted by superscript $(\cdot)^T$. We also use $\otimes$ for the Kronecker product, $\operatorname{vec}\{\cdot\}$ for a vector formed by stacking the columns of its matrix argument, and $\operatorname{Tr}\{\cdot\}$ for the trace of its matrix argument. We further use $\operatorname{Blkcol}\{\cdot\}$ to denote a column vector formed by stacking its arguments on top of each other. The $\operatorname{Blkdiag}\{\cdot\}$ denotes the block-diagonal matrix and $\odot$ the Hadamard product. The expected value of a random quantity $\mathbf{x}$ is denoted by $\mathbb{E}[\mathbf{x}]$. The weighted Euclidean norm of a vector $\mathbf{x}$ with a weighting matrix $\mathbf{\Sigma}$ is defined as $\|\mathbf{x}\|_{\mathbf{\Sigma}} = \sqrt{\mathbf{x}^T\mathbf{\Sigma}\mathbf{x}}$.

II. Preliminaries

A. Network Model

Consider a connected multi-agent network[1] with K nodes scattered in a geographical space. Two nodes are neighbors if they can exchange information with each other, i.e. only single-hop communications are allowed. The neighborhood of node k is denoted by $\mathcal{N}_k$ (notice that $k \in \mathcal{N}_k$, See Fig. 1). It is common to assume that the inter-node communication links are symmetric, thus the topology of the network can be modeled as an undirected graph $\mathcal{G} = \{\mathcal{K}, \mathcal{E}, \mathcal{A}\}$, where $\mathcal{K} = \{1, 2, \ldots, K\}$ is the set of vertices corresponding to the agents or nodes, $\mathcal{E} \subseteq \mathcal{K} \times \mathcal{K}$ is the set of edges representing available communication links and the symmetric adjacency matrix $\mathcal{A} = [a_{lk}] \in \mathbb{R}^{K\times K}$ represents the presence and absence of connections. For any $\{k, l\} \in \mathcal{K}$, $a_{lk} = a_{kl}$ and $a_{lk} > 0$, if and only if $l \in \mathcal{N}_k$, i.e. there is an edge $(l, k) \in \mathcal{E}$ that nodes k and l can exchange information through communication link. The graph $\mathcal{G}$ is called a connected graph, in the sense of a topological space, if there exists a path between every pair of vertices $\{l, k\} \in \mathcal{K}$.

B. System Description

We consider a linear state-space model as:

$$\mathbf{x}_{t+1} = \mathbf{F}_t\mathbf{x}_t + \mathbf{G}_t\mathbf{u}_t, \tag{1a}$$

$$\mathbf{y}_{k,t} = \mathbf{H}_{k,t}\mathbf{x}_t + \mathbf{v}_{k,t}, \tag{1b}$$

where $\mathbf{x}_t \in \mathbb{R}^M$ denotes the state of the system at time instants $t = \{0, 1, 2, \cdots\}$ and $\mathbf{y}_{k,t}$ denotes the measurement vector of the system at node k at time t. Moreover, in (1), $\mathbf{F}_t \in \mathbb{R}^{M\times M}$, $\mathbf{G}_t \in \mathbb{R}^{M\times M}$ and $\mathbf{H}_{k,t} \in \mathbb{R}^{P\times M}$ denote the model matrix, the state noise matrix and the local data matrix, respectively.

[1] The connectivity condition is requited to ensure diffusion of information through the entire network.

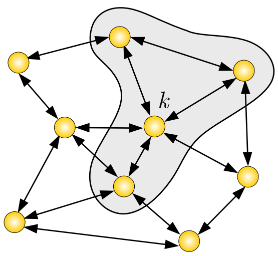

Fig. 1. A network with K nodes with a neighborhood of node k highlighted in gray.

The state transition noise vector $\mathbf{u}_t \in \mathbb{R}^M$ and measurement noise $\mathbf{v}_{k,t} \in \mathbb{R}^P$ are assumed to be the realization of zero-mean white process with covariance matrices $\mathbf{Q}_t$ and $\mathbf{R}_{k,t}$, respectively,

$$\mathbb{E}\left[\begin{bmatrix}\mathbf{u}_t\\ \mathbf{v}_{k,t}\end{bmatrix}\begin{bmatrix}\mathbf{u}_s\\ \mathbf{v}_{k,s}\end{bmatrix}^T\right] \triangleq \begin{bmatrix}\mathbf{Q}_t & 0\\ 0 & \mathbf{R}_{k,t}\end{bmatrix}\delta_{ts} \tag{2}$$

where δ_{ts} is the Kronecker delta, i.e. $\delta_{ts} = 1$ when $t = s$ and $\delta_{ts} = 0$ when $t \neq s$. It is assumed that the measurement noise vectors $\mathbf{v}_{k,t}$ are spatially and temporally uncorrelated, i.e.,

$$\mathbb{E}\left[\mathbf{v}_{k,t}\mathbf{v}_{l,s}^T\right] = \mathbf{R}_{k,t}\delta_{kl}\delta_{ts},\ \mathbf{R}_{k,t} > 0 \tag{3}$$

Moreover, the initial state $\mathbf{x}_0$ is zero-mean with covariance matrix $\mathbb{E}\left[\mathbf{x}_0\mathbf{x}_0^T\right] = \mathbf{\Omega}_0 > 0$ and uncorrelated with noise signals $\mathbf{v}_{k,t}$ and $\mathbf{u}_t$ for all t and k. The parameter matrices $\{\mathbf{F}_t, \mathbf{G}_t, \mathbf{Q}_t, \mathbf{R}_{k,t}, \mathbf{\Omega}_0, \mathbf{H}_{k,t}\}$ are considered known by each node k.

The objective is to develop a fully distributed algorithm to estimate the state vectors $\{\mathbf{x}_t\}$ in a time-sequential manner [39], based on sequences of noisy measurements $\{\mathbf{y}_{k,j}\}, j \leq t$.

C. Diffusion Kalman Filter

Let $\hat{\mathbf{x}}_{k,t|s}$ denote the linear minimum mean-square error estimate of $\mathbf{x}_t$ that node k computes at time t using the available information and measurements up to and including time s. The covariance matrix of the estimation error $\tilde{\mathbf{x}}_{t|s} \triangleq \mathbf{x}_t - \hat{\mathbf{x}}_{k,t|s}$ is denoted by $\mathbf{P}_{k,t|s}$. The diffKF algorithm in its time-and-measurement update form begins with $\hat{\mathbf{x}}_{k,0|-1} = \mathbf{0}$ and $\mathbf{P}_{k,0|-1} = \mathbf{\Omega}_0$, where $\mathbf{P}_{k,0|-1} \in \mathbb{R}^{M\times M}$. The algorithm includes two steps, namely the *incremental update* and *diffusion update*. In the incremental step, at every time instant t each node shares local data $\{\mathbf{y}_{k,t}, \mathbf{H}_{k,t}, \mathbf{R}_{k,t}\}$ with its neighbors and computes $\boldsymbol{\phi}_{k,t} \leftarrow \hat{\mathbf{x}}_{k,t|t-1}$ and $\mathbf{P}_{k,t} \leftarrow \mathbf{P}_{k,t|t-1}$. Then, each node calculates the intermediate estimates $\boldsymbol{\phi}_{k,t}$ by performing the KF algorithm as follows

```
for   $l \in \mathcal{N}_k$   repeat
    $\mathbf{R}_{e,t} \leftarrow \mathbf{R}_{\ell,t} + \mathbf{H}_{\ell,t}\mathbf{P}_{k,t}\mathbf{H}_{\ell,t}^T$
    $\boldsymbol{\phi}_{k,t} \leftarrow \boldsymbol{\phi}_{k,t} + \mathbf{P}_{k,t}\mathbf{H}_{\ell,t}^T\mathbf{R}_{e,t}^{-1}[\mathbf{y}_{\ell,t} - \mathbf{H}_{\ell,t}\boldsymbol{\phi}_{k,t}]$
    $\mathbf{P}_{k,t} \leftarrow \mathbf{P}_{k,t} - \mathbf{P}_{k,t}\mathbf{H}_{\ell,t}^T\mathbf{R}_{e,t}^{-1}\mathbf{H}_{\ell,t}\mathbf{P}_{k,t}$        (4)
end
    $\hat{\mathbf{x}}_{k,t|t} \leftarrow \boldsymbol{\phi}_{k,t}$
    $\mathbf{P}_{k,t|t} \leftarrow \mathbf{P}_{k,t}$
    $\hat{\mathbf{x}}_{k,t+1|t} = \mathbf{F}_t\hat{\mathbf{x}}_{k,t|t}$
    $\mathbf{P}_{k,t+1|t} = \mathbf{F}_t\mathbf{P}_{k,t|t}\mathbf{F}_t^T + \mathbf{G}_t\mathbf{Q}_t\mathbf{G}_t^T$
```

The symbol ← denotes a sequential assignment. In the diffusion step, the nodes share $\boldsymbol{\phi}_{k,t}$ with their immediate neighbors and calculate an updated state vector $\hat{\mathbf{x}}_{k,t|t}$ as:

$$\hat{\mathbf{x}}_{k,t|t} = \sum_{\ell\in\mathcal{N}_k} c_{\ell k}\boldsymbol{\phi}_{\ell,t} \tag{5}$$

The scalars $\{c_{\ell k}\}$ denote real non-negative convex combination coefficients where $c_{\ell k} = 0$ if $\ell \notin \mathcal{N}_k$ and:

$$\sum_{\ell=1}^{K} c_{\ell k} = 1, \quad \mathbb{1}^T\mathbf{C} = \mathbb{1}^T \tag{6}$$

where $\mathbb{1}$ denotes $K \times 1$ vector with unit entries. The combination weight $c_{\ell k}$ is the (ℓ, k) element of matrix $\mathbf{C} \in \mathbb{R}^{N\times N}$, named *combination* matrix. The property above conveys that $\mathbf{C}$ is a left stochastic matrix, i.e., the magnitude of any of the eigenvalues of $\mathbf{C}$ is bounded by one.

III. Proposed Algorithm

A. Algorithm Extraction

Although diffKF algorithm exhibits proper estimation performance in a connected network, it requires each node to share its intermediate state estimates with all of its neighbors at any iteration. In order to reduce the communication load, we consider the case in which each node is allowed to diffuse the update estimate with only a fraction of nodes within its neighborhood $\mathcal{N}_k$. While this procedure decreases the total amount of communication burden in the network, it is at the expense of slight degradation in terms of MSD performance compared to the diffKF.

Let $d_k = |\mathcal{N}_k|$ be the degree or valency of node k, where $|\cdot|$ is the cardinality operator. To achieve this, assume node k communicates at each time instant t to receive the intermediate estimate $\boldsymbol{\phi}_{\ell,t}$ from $0 < \eta_k \leq d_k$ nodes. To-be-selected neighboring nodes of node k at iteration t are characterized by a link-selection variable as $\pi_{\ell k,t}$. This variable determines the status of link, being active or inactive, between node k and ℓ at time instant t. The link-selection variable is defined as follows:

$$\pi_{\ell k,t} = \begin{cases} 1 & \text{if } \ell \in \mathcal{N}_{k,t}^{'} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $\mathcal{N}_{k,t}^{'} \subseteq \mathcal{N}_k$ consists those nodes of $\mathcal{N}_k$ which are allowed to exchange information.

Regarding the proposed link selection scheme, the following remark is made [21]:

Remark 1. *The link-selection variable $\{\pi_{\ell k,t}\}$ is mutually independent of each element of set $\{\mathbf{u}_t, \mathbf{v}_{k,t}, \mathbf{x}_t\}$. Moreover, let denote by ρ_k the link-selection probability. It is assumed that ρ_k is to be identical for all the neighbors:*

$$\rho_k = \mathbb{E}\left[\pi_{\ell k,t}\right] = \frac{\eta_k}{d_k}$$

In diffKF algorithm, the nodes exchange local data $\{\mathbf{y}_{k,t}, \mathbf{H}_{k,t}, \mathbf{R}_{k,t}\}$ to achieve the intermediate estimates $\boldsymbol{\phi}_{k,t}$. Obviously, considerable communication and computation resources are required to implement this algorithm. In order to reduce the communication load as much as possible, here an algorithm, based on a modified version of diffKF is used where the nodes do not receive $\{\mathbf{y}_{k,t}, \mathbf{H}_{k,t}, \mathbf{R}_{k,t}\}$ from their neighbors in the incremental phase. So, in the diffKF algorithm, the incremental step (4) should be replaced by the following *Adaptation Phase*:

$$\begin{aligned}
\mathbf{R}_{e,t} &\longleftarrow \mathbf{R}_{k,t} + \mathbf{H}_{k,t}\mathbf{P}_{k,t}\mathbf{H}_{k,t}^{T} \\
\boldsymbol{\phi}_{k,t} &\longleftarrow \boldsymbol{\phi}_{k,t} + \mathbf{P}_{k,t}\mathbf{H}_{k,t}^{T}\mathbf{R}_{e,t}^{-1}[\mathbf{y}_{k,t} - \mathbf{H}_{k,t}\boldsymbol{\phi}_{k,t}] \qquad (8) \\
\mathbf{P}_{k,t} &\longleftarrow \mathbf{P}_{k,t} - \mathbf{P}_{k,t}\mathbf{H}_{k,t}^{T}\mathbf{R}_{e,t}^{-1}\mathbf{H}_{k,t}\mathbf{P}_{k,t}
\end{aligned}$$

As the intermediate estimates from the selected neighbors are available at node k, we instead propose a new aggregation method that uses the node's own intermediate estimate as a proxy [35] for the missing data and changes (5) to:

$$\hat{\mathbf{x}}_{k,t|t} = c_{kk}\boldsymbol{\phi}_{k,t} + \sum_{\ell\in\mathcal{N}_k'} c_{\ell k}[\pi_{\ell k,t}\boldsymbol{\phi}_{\ell,t} + (1-\pi_{\ell k,t})\,\boldsymbol{\phi}_{k,t}] \quad (9)$$

Accordingly, the considered RL-diffKF algorithm utilizes (8) in the adaptation phase and (9) for aggregation phase. It is noteworthy to say that (5) and (9) have the same computational complexity. The method summary can be seen in Algorithm 1.

Remark 2. *The only required condition in the network for using the proposed algorithm is that the network should be connected. So, as the proposed algorithm is naturally distributed, in the case of configuration changes (e.g., addition/removal of agents or sensors), this paves the way to a plug-and-play (PnP) implementation.*

B. A Typical Application

An increasing IoT trend is mobile crowd-sensing (MCS) where carriers of sensing and computing devices obtain and diffuse vital information for different types of application [40]. MCS mainly occurs in three phases, i.e., data collection, data storage, and data upload, involves two categories of users, i.e., task requesters and mobile participants, and three layers, i.e., the sensing layer, application layer and cloud platform, respectively [41]. The overall process of distributed MCS is shown on Fig. 2. The process is briefly explained as follows: (1) A task requester submits its requirements to the cloud server, such as monitoring air quality; (2) These tasks are transmitted to the users willing to assess a particular task T in their smart devices; (3) The mobile users can collaborate as long as their smart devices operate. These devices are employed to sense the surrounding environment all the times; (4) A diffusion Kalman filter, diff-KF, strategy is used for data processing among mobile devices. This process includes two steps, an incremental update followed by a diffusion update; (5) Finally, the users diffuse their estimate values to task providers.

Algorithm 1 Reduced-Link Diffusion Kalman Filter

```
Initialization: $\hat{\mathbf{x}}_{k,0|-1} = 0$ and $\mathbf{P}_{k,0|-1} = \boldsymbol{\Omega}_0$
For every time instant $t$, every node $k$ computes

Step1: Adaptation phase
$\boldsymbol{\phi}_{k,t} \leftarrow \hat{\mathbf{x}}_{k,t|t-1}$
$\mathbf{P}_{k,t} \leftarrow \mathbf{P}_{k,t|t-1}$
        $\mathbf{R}_{e,t} \leftarrow \mathbf{R}_{k,t} + \mathbf{H}_{k,t}\mathbf{P}_{k,t}\mathbf{H}_{k,t}^{T}$
        $\boldsymbol{\phi}_{k,t} \leftarrow \boldsymbol{\phi}_{k,t} + \mathbf{P}_{k,t}\mathbf{H}_{k,t}^{T}\mathbf{R}_{e,t}^{-1}[\mathbf{y}_{k,t} - \mathbf{H}_{k,t}\boldsymbol{\phi}_{k,t}]$
        $\mathbf{P}_{k,t} \leftarrow \mathbf{P}_{k,t} - \mathbf{P}_{k,t}\mathbf{H}_{k,t}^{T}\mathbf{R}_{e,t}^{-1}\mathbf{H}_{k,t}\mathbf{P}_{k,t}$

Step2: Aggregation Phase
    $\hat{\mathbf{x}}_{k,t|t} = c_{kk}\boldsymbol{\phi}_{k,t} + \sum_{\ell\in\mathcal{N}_k'} c_{\ell k}\left[\pi_{\ell k,t}\boldsymbol{\phi}_{\ell,t} + (1-\pi_{\ell k,t})\,\boldsymbol{\phi}_{k,t}\right]$
    $\mathbf{P}_{k,t|t} \leftarrow \mathbf{P}_{k,t}$
    $\hat{\mathbf{x}}_{k,t+1|t} = \mathbf{F}_t\hat{\mathbf{x}}_{k,t|t}$
    $\mathbf{P}_{k,t+1|t} = \mathbf{F}_t\mathbf{P}_{k,t|t}\mathbf{F}_t^{T} + \mathbf{G}_t\mathbf{Q}_t\mathbf{G}_t^{T}$
```

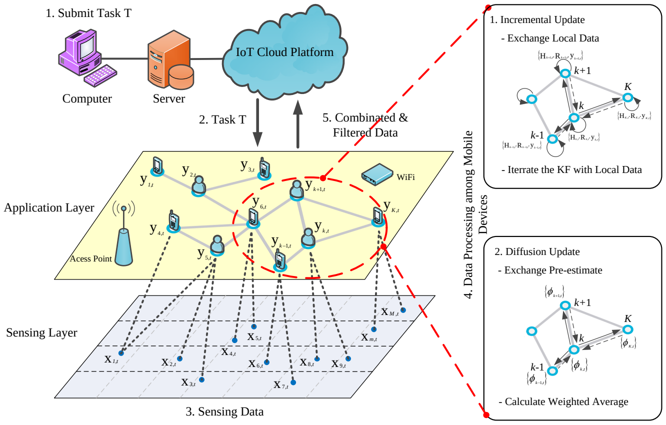

Fig. 2. IoT framework for MCS based on diffKF.

Remark 3. *As we mentioned earlier, in many practical applications, such as distributed power systems, smart grid and sensor networks, diffusion based algorithms have been developed as promising solutions for distributed state estimation [42]–[44]. In this paper, we have proposed an IoT based energy-efficient algorithm to solve the state estimation problems in a fully distributed manner. This means that with proper data modeling and conditioning, the proposed algorithm can be used at the same applications.*

IV. Performance Analysis

In this section, different aspects of RL-diffKF algorithm are examined. Defining the following error signals

$$\widetilde{\mathbf{x}}_{k,t|t} = \mathbf{x}_t - \hat{\mathbf{x}}_{k,t|t} \tag{10}$$

$$\widetilde{\boldsymbol{\phi}}_{k,t|t} = \mathbf{x}_t - \boldsymbol{\phi}_{k,t|t} \tag{11}$$

and subtracting $\mathbf{x}_t$ from both sides of the expression for $\boldsymbol{\phi}_{k,t}$ in (8) yields

$$\begin{aligned}\tilde{\boldsymbol{\phi}}_{k,t} &= \tilde{\mathbf{x}}_{k,t|t-1} - \mathbf{P}_{k,t}\mathbf{H}_{k,t}^T\mathbf{R}_{e,t}^{-1}(\mathbf{H}_{k,t}\tilde{\mathbf{x}}_{k,t|t-1} + \mathbf{v}_{k,t}) \\ &= (\mathbf{I}_M - \mathbf{P}_{k,t}\mathbf{H}_{k,t}^T\mathbf{R}_{e,t}^{-1}\mathbf{H}_{k,t})\tilde{\mathbf{x}}_{k,t|t-1} \\ &\qquad - \mathbf{P}_{k,t}\mathbf{H}_{k,t}^T\mathbf{R}_{e,t}^{-1}\mathbf{v}_{k,t}\end{aligned} \tag{12}$$

Noting that $\mathbf{P}_{k,t}\mathbf{H}_{k,t}^T\mathbf{R}_{e,t}^{-1} = \mathbf{P}_{k,t}\mathbf{H}_{k,t}^T\mathbf{R}_{k,t}^{-1}$, the above equation can be rewritten as

$$\tilde{\boldsymbol{\phi}}_{k,t} = (\mathbf{I}_M - \mathbf{P}_{k,t}\mathbf{S}_{k,t})\tilde{\mathbf{x}}_{k,t|t-1} - \mathbf{P}_{k,t}\mathbf{H}_{k,t}^T\mathbf{R}_{k,t}^{-1}\mathbf{v}_{k,t} \tag{13}$$

where $\mathbf{S}_{k,t} = \mathbf{H}_{k,t}^T\mathbf{R}_{k,t}^{-1}\mathbf{H}_{k,t}$. Using

$$\widetilde{\mathbf{x}}_{k,t|t-1} = \mathbf{F}_{t-1}\tilde{\mathbf{x}}_{k,t-1|t-1} + \mathbf{G}_{t-1}\mathbf{u}_{t-1} \tag{14}$$

the following recursion is obtained:

$$\begin{aligned}\widetilde{\boldsymbol{\phi}}_{k,t} &= (\mathbf{I}_M - \mathbf{P}_{k,t}\mathbf{S}_{k,t})\mathbf{F}_{t-1}\tilde{\mathbf{x}}_{k,t-1|t-1} \\ &\quad + (\mathbf{I}_M - \mathbf{P}_{k,t}\mathbf{S}_{k,t})\mathbf{G}_{t-1}\mathbf{u}_{t-1} - \mathbf{P}_{k,t}\mathbf{H}_{k,t}^T\mathbf{R}_{k,t}^{-1}\mathbf{v}_{k,t}\end{aligned} \tag{15}$$

Using the combination step from (9) gives

$$\tilde{\mathbf{x}}_{k,t|t} = \left(1 - \sum_{\ell\in\mathcal{N}_k} \pi_{\ell k,t}c_{\ell k}\right)\tilde{\boldsymbol{\phi}}_{k,t} + \sum_{\ell\in\mathcal{N}_k'} \pi_{\ell k,t}c_{\ell k}\tilde{\boldsymbol{\phi}}_{\ell,t} \tag{16}$$

Now, the following supporting block-diagonal and Kronecker product matrices are introduced:

$$\begin{aligned}\mathbf{U}_{t-1} &\triangleq \mathbb{1}_K \otimes \mathbf{u}_{t-1} \\ \mathbf{V}_t &\triangleq \text{Blkcol}\left\{\mathbf{v}_{1,t}, \mathbf{v}_{2,t}, \cdots, \mathbf{v}_{K,t}\right\} \\ \boldsymbol{\mathcal{P}}_t &\triangleq \text{Blkdiag}\left\{\mathbf{P}_{1,t}, \mathbf{P}_{t,2}, \cdots, \mathbf{P}_{K,t}\right\} \\ \boldsymbol{\mathcal{H}}_t &\triangleq \text{Blkdiag}\left\{\mathbf{H}_{1,t}, \mathbf{H}_{2,t}, \cdots, \mathbf{H}_{K,t}\right\} \\ \boldsymbol{\mathcal{S}}_t &\triangleq \text{Blkdiag}\left\{\mathbf{S}_{1,t}, \mathbf{S}_{2,t}, \cdots, \mathbf{S}_{K,t}\right\} \\ \boldsymbol{\mathcal{R}}_t &\triangleq \text{Blkdiag}\left\{\mathbf{R}_{1,t}, \mathbf{R}_{2,t}, \cdots, \mathbf{R}_{K,t}\right\} \\ \boldsymbol{\mathcal{B}}_t &= \mathbf{B}_t \otimes \mathbf{I}_M \\ \mathbf{B}_t &= \begin{bmatrix} b_{11,t} & \cdots & b_{1K,t} \\ \vdots & \ddots & \vdots \\ b_{K1,t} & \cdots & b_{KK,t}\end{bmatrix}.\end{aligned}$$

where

$$b_{pq,t} = \begin{cases} 1 - \sum_{\ell\in\mathcal{N}_p\setminus\{p\}} \pi_{\ell p,t}c_{\ell p} & \text{if}\, p = q \\ \pi_{pq,t}c_{qp} & \text{if}\, q \in \mathcal{N}_p\setminus\{p\} \\ 0 & \text{otherwise}\end{cases}$$

Thus, (15) and (16) are expressed in a global form that captures the evolution of entire network as follows:

$$\widetilde{\boldsymbol{\mathcal{X}}}_{t|t} = \boldsymbol{\mathcal{B}}_t\widetilde{\boldsymbol{\Phi}}_t \tag{17}$$

$$\begin{aligned}\widetilde{\boldsymbol{\Phi}}_t &= \left(\mathbf{I}_{KM} - \boldsymbol{\mathcal{P}}_{t|t}\boldsymbol{\mathcal{S}}_t\right)\left[\left(\mathbf{I}_K \otimes \mathbf{F}_{t-1}\right)\widetilde{\boldsymbol{\mathcal{X}}}_{t-1|t-1}\right. \\ &\quad \left. + \left(\mathbf{I}_K \otimes \mathbf{G}_{t-1}\right)\mathbf{U}_t\right] - \boldsymbol{\mathcal{P}}_{t|t}\boldsymbol{\mathcal{H}}_t^T\boldsymbol{\mathcal{R}}_t^{-1}\mathbf{V}_t\end{aligned} \tag{18}$$

These relations demonstrate that $\widetilde{\boldsymbol{\mathcal{X}}}_{t|t}$ evolves over time as:

$$\begin{aligned}\widetilde{\boldsymbol{\mathcal{X}}}_{t|t} &= \boldsymbol{\mathcal{B}}_t\left(\mathbf{I}_{KM} - \boldsymbol{\mathcal{P}}_{t|t}\boldsymbol{\mathcal{S}}_t\right)\left(\mathbf{I}_K \otimes \mathbf{F}_{t-1}\right)\widetilde{\boldsymbol{\mathcal{X}}}_{t-1|t-1} \\ &\quad + \boldsymbol{\mathcal{B}}_t\left(\mathbf{I}_K \otimes \mathbf{G}_{t-1}\right)\mathbf{U}_t - \boldsymbol{\mathcal{B}}_t\boldsymbol{\mathcal{P}}_{t|t}\boldsymbol{\mathcal{H}}_t^T\boldsymbol{\mathcal{R}}_t^{-1}\mathbf{V}_t\end{aligned} \tag{19}$$

which can also be rewritten as the following compact form:

$$\widetilde{\boldsymbol{\mathcal{X}}}_{t|t} = \boldsymbol{\mathcal{B}}_t\boldsymbol{\mathcal{F}}_t\widetilde{\boldsymbol{\mathcal{X}}}_{t-1|t-1} + \boldsymbol{\mathcal{B}}_t\boldsymbol{\mathcal{G}}_t\mathbf{U}_t - \boldsymbol{\mathcal{B}}_t\boldsymbol{\mathcal{D}}_t\mathbf{V}_t \tag{20}$$

where

$$\begin{aligned}\boldsymbol{\mathcal{F}}_t &\triangleq \left(\mathbf{I}_{KM} - \boldsymbol{\mathcal{P}}_{t|t}\boldsymbol{\mathcal{S}}_t\right)\left(\mathbf{I}_K \otimes \mathbf{F}_{t-1}\right) \\ \boldsymbol{\mathcal{G}}_t &\triangleq \left(\mathbf{I}_{KM} - \boldsymbol{\mathcal{P}}_{t|t}\boldsymbol{\mathcal{S}}_t\right)\left(\mathbf{I}_K \otimes \mathbf{G}_{t-1}\right) \\ \boldsymbol{\mathcal{D}}_t &\triangleq \boldsymbol{\mathcal{P}}_{t|t}\boldsymbol{\mathcal{H}}_t^T\boldsymbol{\mathcal{R}}_t^{-1}\end{aligned}$$

A. Mean Performance

The following proposition guarantees the stability and asymptotic unbiasedness of the RL-diffKF algorithm.

Proposition 1. (**Mean Stability**) *The RL-diffKF is convergent in the mean and asymptotically unbiased.*

Proof. Applying expectations to both sides of (14) and (16), the following recursions for the estimate expectations are obtained:

$$\mathbb{E}\left[\widetilde{\mathbf{x}}_{k,t|t-1}\right] = \mathbf{F}_{t-1}\mathbb{E}\left[\tilde{\mathbf{x}}_{k,t-1|t-1}\right] \tag{21}$$

$$\begin{aligned}\mathbb{E}\left[\widetilde{\mathbf{x}}_{k,t|t}\right] &= \bar{\beta}_{kk}(\mathbf{I}_M - \mathbf{P}_{k,t}\mathbf{S}_{k,t})\tilde{\mathbf{x}}_{k,t|t-1} \\ &\quad + \sum_{\ell\in\mathcal{N}_k'} \bar{\beta}_{\ell k}(\mathbf{I}_M - \mathbf{P}_{\ell,t}\mathbf{S}_{\ell,t})\mathbb{E}\left[\tilde{\mathbf{x}}_{\ell,t|t-1}\right]\end{aligned} \tag{22}$$

where

$$\bar{\beta}_{pq} = \begin{cases} 1 - \sum_{\ell\in\mathcal{N}_p\setminus\{p\}} \rho_\ell c_{\ell p} & \text{if}\, p = q \\ \rho_p c_{qp} & \text{if}\, q \in \mathcal{N}_p\setminus\{p\} \\ 0 & \text{otherwise}\end{cases}$$

Since we assume $\hat{\mathbf{x}}_{k,0|-1} = 0$ and $\mathbb{E}\left[\mathbf{x}_0\right] = 0$, we have $\mathbb{E}\left[\hat{\mathbf{x}}_{k,0|-1}\right] = 0$ for all k. Then, we obtain

$$\begin{aligned}\mathbb{E}\left[\widetilde{\mathbf{x}}_{k,0|0}\right] &= \bar{\beta}_{kk}(\mathbf{I}_M - \mathbf{P}_{k,0}\mathbf{S}_{k,0})\tilde{\mathbf{x}}_{k,0|-1} \\ &\quad + \sum_{\ell\in\mathcal{N}_k'} \bar{\beta}_{\ell k}(\mathbf{I}_M - \mathbf{P}_{0,\ell}\mathbf{S}_{0,\ell})\mathbb{E}\left[\tilde{\mathbf{x}}_{\ell,0|-1}\right] = 0\end{aligned} \tag{23}$$

Thus, iterating (21) and (22), it is concluded that the estimates $\mathbb{E}\left[\widetilde{\mathbf{x}}_{k,t|t}\right]$ provided by the RL-diffKF is unbiased for all $t \geq 0$. ■

B. Mean-Square Performance

If $\boldsymbol{\mathcal{P}}_{\boldsymbol{\mathcal{X}},t}$ denotes the covariance matrix of the network error vectors

$$\boldsymbol{\mathcal{P}}_{\tilde{\boldsymbol{\mathcal{X}}},t} \triangleq \mathbb{E}\left[\tilde{\boldsymbol{\mathcal{X}}}_{t|t}\tilde{\boldsymbol{\mathcal{X}}}_{t|t}^T\right] \tag{24}$$

then, it follows from (20) that this matrix satisfies the following Lyapunov recursion:

$$\begin{aligned}\boldsymbol{\mathcal{P}}_{\tilde{\boldsymbol{\mathcal{X}}},t} = \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{F}}_t\boldsymbol{\mathcal{P}}_{\tilde{\boldsymbol{\mathcal{X}}},t-1}\boldsymbol{\mathcal{F}}_t^T\boldsymbol{\mathcal{B}}^T + \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{G}}_t\left(\mathbb{1}_K\mathbb{1}_K^T\otimes\mathbf{Q}_{t-1}\right)\boldsymbol{\mathcal{G}}_t^T\boldsymbol{\mathcal{B}}^T\\ + \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{D}}_t\boldsymbol{\mathcal{R}}_t\boldsymbol{\mathcal{D}}_t^T\boldsymbol{\mathcal{B}}^T\end{aligned} \tag{25}$$

where $\boldsymbol{\mathcal{B}} = \mathbb{E}\left[\boldsymbol{\mathcal{B}}_t\right]$. In order to analyze the stability and performance of the RL-diffKF algorithm, the following assumption is considered.

Assumption 1. *The matrices in the linear state-space model (i.e., the matrices* $\mathbf{F}$*,* $\mathbf{G}$*,* $\mathbf{H}$*,* $\mathbf{R}$ *and* $\mathbf{Q}$*) are time-invariant. Furthermore, it is assumed that the pair* $(\mathbf{F}, \mathbf{H}_k)$ *is detectable and the pair* $\left(\mathbf{F}, \mathbf{G}\mathbf{Q}^{1/2}\right)$ *is stabilizable and*

Note that under Assumption 1 and the convergence properties of the discrete Riccati recursions, each entry $\mathbf{P}_{k,t}$ of $\boldsymbol{\mathcal{P}}_t$ converges to $\mathbf{P}_k^{\dagger}$ where [39]:

$$\mathbf{P}_k^{\dagger} = \left(\mathbf{P}_k^{-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k\right) = \mathbf{P}_k^{-1} + \mathbf{S}_k \tag{26}$$

where matrix $\mathbf{P}_k$ is the unique stabilizing solution of the following discrete algebraic Riccati equation (DARE):

$$\begin{aligned}\mathbf{P}_k &= \mathbf{F}\mathbf{P}_k^{\dagger}\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T\\ &= \mathbf{F}\mathbf{P}_k\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T - \mathbf{\Gamma}_{p,k}\mathbf{R}_{e,k}\mathbf{\Gamma}_{p,k}^T\end{aligned} \tag{27}$$

where $\mathbf{\Gamma}_{p,k} \triangleq \mathbf{F}\mathbf{P}_k\mathbf{H}_k^T\mathbf{R}_{e,k}^{-1}$ and $\mathbf{R}_{e,k} \triangleq \mathbf{R}_k + \mathbf{H}_k\mathbf{P}_k\mathbf{H}_k^T$. Consequently, the matrices $\left\{\boldsymbol{\mathcal{P}}_{t|t}, \boldsymbol{\mathcal{F}}_t, \boldsymbol{\mathcal{G}}_t, \boldsymbol{\mathcal{D}}_t\right\}$ converge respectively to the following steady-state values:

$$\begin{aligned}\boldsymbol{\mathcal{P}}^{\dagger} &\triangleq \lim_{t\to\infty}\boldsymbol{\mathcal{P}}_{t|t}^{\dagger} \;\; \text{(block diagonal)}\\ \boldsymbol{\mathcal{F}} &\triangleq \lim_{t\to\infty}\boldsymbol{\mathcal{F}}_t = \left(\mathbf{I}_{KM} - \boldsymbol{\mathcal{P}}^{\dagger}\boldsymbol{\mathcal{S}}\right)(\mathbf{I}_K\otimes\mathbf{F})\\ \boldsymbol{\mathcal{G}} &\triangleq \lim_{t\to\infty}\boldsymbol{\mathcal{G}}_t = \left(\mathbf{I}_{KM} - \boldsymbol{\mathcal{P}}^{\dagger}\boldsymbol{\mathcal{S}}\right)(\mathbf{I}_K\otimes\mathbf{G})\\ \boldsymbol{\mathcal{D}} &\triangleq \lim_{t\to\infty}\boldsymbol{\mathcal{D}}_t = \boldsymbol{\mathcal{P}}^{\dagger}\boldsymbol{\mathcal{H}}^T\mathbf{R}^{-1}\end{aligned} \tag{28}$$

Assumptions 1 is sufficient to guarantee the convergence of the RL-diffKF. It is shown in [18] that matrix $\boldsymbol{\mathcal{F}}$ is stable (the stability condition is discussed in more detail in [18]). Therefore, the error covariance matrix $\boldsymbol{\mathcal{P}}_{\tilde{\boldsymbol{\mathcal{X}}},t}$ converges to the unique solution of the Lyapunov equation:

$$\begin{aligned}\boldsymbol{\mathcal{P}}_{\tilde{\boldsymbol{\mathcal{X}}}} = \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{F}}\boldsymbol{\mathcal{P}}_{\tilde{\boldsymbol{\mathcal{X}}}}\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{B}}^T + \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{G}}\left(\mathbb{1}_K\mathbb{1}_K^T\otimes\mathbf{Q}\right)\boldsymbol{\mathcal{G}}^T\boldsymbol{\mathcal{B}}^T\\ + \boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{D}}^T\boldsymbol{\mathcal{B}}^T\end{aligned} \tag{29}$$

Applying $\operatorname{vec}\{\cdot\}$ operator to (29) together with using the property $\operatorname{vec}\{\mathbf{A}\mathbf{\Sigma}\mathbf{B}\} = \left(\mathbf{B}^T\otimes\mathbf{A}\right)\operatorname{vec}\{\mathbf{\Sigma}\}$, we can solve for $\boldsymbol{\mathcal{P}}_{\tilde{\boldsymbol{\mathcal{X}}}}$ and write

$$\begin{aligned}\operatorname{vec}\{\boldsymbol{\mathcal{P}}_{\tilde{\boldsymbol{\mathcal{X}}}}\} = \left[\mathbf{I}_{K^2M^2} - \mathfrak{B}\left(\boldsymbol{\mathcal{F}}\otimes\boldsymbol{\mathcal{F}}\right)\right]^{-1}\\ \times\,\mathfrak{B}\operatorname{vec}\left\{\boldsymbol{\mathcal{G}}\left(\mathbb{1}_K\mathbb{1}_K^T\otimes\mathbf{Q}\right)\boldsymbol{\mathcal{G}}^T + \boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{D}}^T\right\}\end{aligned} \tag{30}$$

where $\mathfrak{B} = \boldsymbol{\mathcal{B}}\otimes\boldsymbol{\mathcal{B}}$. $\mathfrak{B}$ in Appendix A is calculated. The steady-state MSD is defined in terms of state variance

$$\zeta_k = \mathbb{E}\left[\|\mathbf{x}_t - \hat{\mathbf{x}}_{k,t|t}\|^2\right] \tag{31}$$

So, the steady-state MSD at node k is given by:

$$\zeta_k = \lim_{t\to\infty}\mathbb{E}\left[\|\tilde{\mathbf{x}}_{k,t}\|\right] = \operatorname{Tr}\left\{\boldsymbol{\mathcal{P}}_{\tilde{\boldsymbol{\mathcal{X}}}}\boldsymbol{\mathcal{I}}_k\right\} \tag{32}$$

where $\boldsymbol{\mathcal{I}}_k$ is a $K\times K$ block diagonal matrix with blocks of size $M\times M$. It contains the identity matrix at block (k,k) and zero elsewhere. Eventually, the network MSD, defined as the average steady-state MSDs across the network, is given by

$$\zeta^{\text{net}} = \frac{1}{K}\operatorname{Tr}\left\{\boldsymbol{\mathcal{P}}_{\tilde{\boldsymbol{\mathcal{X}}}}\right\} \tag{33}$$

The above allegation is validated by demonstrating the stability of matrix $\boldsymbol{\mathcal{F}}$, in view of the fact that the Lyapunov recursion (25) converges to the unique solution of the Lyapunov equation (29). We summarize our results with the following proposition.

Proposition 2. (Mean-square Stability) *Under Assumption 1, the RL-diffKF algorithm is unbiased and converges, and the steady-state mean-square deviation for every node is given by* (32).

Proof. The recursion (30) is stable and convergent if and only if, the matrix $\mathfrak{B}\left(\boldsymbol{\mathcal{F}}\otimes\boldsymbol{\mathcal{F}}\right)$ is stable. All the entries of $\mathfrak{B}$ are real and non-negative. Since,

$$\begin{aligned}\mathfrak{B}\mathbb{1}_{K^2M^2} &= \mathbb{E}\left[\boldsymbol{\mathcal{B}}\otimes\boldsymbol{\mathcal{B}}\right]\mathbb{1}_{K^2M^2}\\ &= \mathbb{E}\left[\boldsymbol{\mathcal{B}}\mathbb{1}_{K^2M^2}\otimes\boldsymbol{\mathcal{B}}\mathbb{1}_{K^2M^2}\right] = \mathbb{1}_{K^2M^2}\end{aligned}$$

and all elements of each row of $\mathfrak{B}$ adds up to one. This means that the block matrix $\mathfrak{B}$ is right-stochastic matrix and has unit spectral radius. In steady-state condition, the RL-diffKF algorithm is stable in the mean-square sense if, and only if, $\boldsymbol{\mathcal{F}}\otimes\boldsymbol{\mathcal{F}}$ is stable. Since the eigenvalues of $\left(\boldsymbol{\mathcal{F}}\otimes\boldsymbol{\mathcal{F}}\right)$ are square of the eigenvalues of $\boldsymbol{\mathcal{F}}$, the stability of $\left(\boldsymbol{\mathcal{F}}\otimes\boldsymbol{\mathcal{F}}\right)$ has the same conditions as the stability of $\boldsymbol{\mathcal{F}}$. This means that, as $\boldsymbol{\mathcal{F}}$ is stable (the stability condition is discussed in more detail in [18]) then, the RL-diffKF algorithm is convergent and stable in mean-square sense, and the steady-state mean-square deviation follows from the derivation of (32). ■

V. Communication-performance Trade-off

To achieve a deep and accurate understanding of the performance of RL-diffKF algorithm, we consider its steady-state network MSD under the following assumptions:

Assumption 2.

(i) The degrees of all agents are identical, i.e., the network is fully connected.

(ii) At only η_k *of each* d_k *consecutive iterations, every agent* k *in the network simultaneously receives the intermediate estimates of all its neighbors in a round-robin fashion.*

(iii) The combination matrix $\mathbf{C}$ *is doubly-stochastic.*

In Appendix B it is shown that in view of Assumption 2, $\mathfrak{B}$ is given by the following relation

$$\mathfrak{B} = (1-\rho)\,\mathbf{I}_{M^2K^2} + \rho\boldsymbol{\mathcal{C}}\otimes\boldsymbol{\mathcal{C}} \tag{34}$$

where $\boldsymbol{\mathcal{C}} = \mathbf{C}\otimes\mathbf{I}_M$ and ρ is the selection probability that is identical for all nodes and all node neighbors at each iteration. To proceed, let ζ_m denote the average steady-state network

MSD of the RL-diffKF algorithm with m selected entries. Thus, (33) can be rewritten as

$$\zeta_m = \frac{1}{K}\mathrm{vec}^T\left\{\boldsymbol{\Omega}\right\}\boldsymbol{\mathfrak{B}}^T\left(\mathbf{I}_{M^2K^2} - \left(\boldsymbol{\mathcal{F}}^T \otimes \boldsymbol{\mathcal{F}}^T\right)\boldsymbol{\mathfrak{B}}^T\right)^{-1} \times \mathrm{vec}\left\{\mathbf{I}_{MK}\right\} \tag{35}$$

since the proposed algorithm is mean-square stable, ζ_m can be rewritten as

$$\zeta_m = \frac{1}{K}\mathrm{vec}^T\left\{\boldsymbol{\Omega}\right\}\boldsymbol{\mathfrak{B}}^T\sum_{j=0}^{\infty}\left[\left(\boldsymbol{\mathcal{F}}^T \otimes \boldsymbol{\mathcal{F}}^T\right)\boldsymbol{\mathfrak{B}}^T\right]^j \mathbf{h} \tag{36}$$

where $\mathbf{h} = \mathrm{vec}\left\{\mathbf{I}_{MK}\right\}$ and $\boldsymbol{\Omega} = \boldsymbol{\mathcal{G}}\left(\mathbb{1}_K\mathbb{1}_K^T \otimes \mathbf{Q}\right)\boldsymbol{\mathcal{G}}^T + \boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{R}}\boldsymbol{\mathcal{D}}^T$. Substituting (34) into (36) leads to

$$\begin{aligned}\zeta_m &= \frac{1}{N}\sum_{i=0}^{\infty}(1-\rho)^{i+1}\mathrm{vec}^T\left\{\boldsymbol{\Omega}\right\}\left[(\boldsymbol{\mathcal{F}}^T)^i \otimes (\boldsymbol{\mathcal{F}}^T)^i\right]\mathbf{h} \\ &\quad + \frac{1}{K}\sum_{i=0}^{\infty}\rho^{i+1}\mathrm{vec}^T\left\{\boldsymbol{\Omega}\right\} \\ &\qquad \times \left[\boldsymbol{\mathcal{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{C}}^T)^i \otimes \boldsymbol{\mathcal{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{C}}^T)^i\right]\mathbf{h} \\ &= \frac{1}{K}\sum_{i=0}^{\infty}(1-\rho)^{i+1}\mathrm{Tr}\left\{(\boldsymbol{\mathcal{F}})^i\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^i\right\} \\ &\quad + \frac{1}{K}\sum_{i=0}^{\infty}\rho^{i+1}\mathrm{Tr}\left\{(\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{F}})^i\boldsymbol{\mathcal{C}}\boldsymbol{\Omega}\boldsymbol{\mathcal{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{C}}^T)^i\right\}\end{aligned} \tag{37}$$

It is obvious that

$$\begin{aligned}&\mathrm{Tr}\left\{(\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{F}})^i\boldsymbol{\mathcal{C}}\boldsymbol{\Omega}\boldsymbol{\mathcal{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{C}}^T)^i\right\} = \\ &\qquad \mathrm{Tr}\left\{\left(\mathbf{C}^T\right)^{i+1}\mathbf{C}^{i+1}\right\} \times \mathrm{Tr}\left\{(\boldsymbol{\mathcal{F}})^i\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^i\right\}\end{aligned} \tag{38}$$

where $\mathrm{Tr}\left\{\left(\mathbf{C}^T\right)^{i+1}\mathbf{C}^{i+1}\right\} = \sum_{k=1}^{K}\|c_{k,i+1}\|^2,\ \forall\ i \geq 0$ and $c_{k,i+1}$ is the k-th row of $\mathbf{C}^{i+1}$. As $\mathbf{C}$ is doubly-stochastic $\mathbf{C}^{i+1}$ is also doubly-stochastic. Moreover, since the network is fully-connected, each node is coupled with all the network nodes. This fact leads to $\|c_{k,i+1}\|^2 \leq 1$, $i \geq 0$, $\forall k$. Consequently, the following inequality holds for $0 < \rho < 1$

$$\begin{aligned}&\mathrm{Tr}\left\{(\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{F}})^i\boldsymbol{\mathcal{C}}\boldsymbol{\Omega}\boldsymbol{\mathcal{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{C}}^T)^i\right\} \\ &\qquad < (1-\rho)^{i+1}\mathrm{Tr}\left\{(\boldsymbol{\mathcal{F}})^i\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^i\right\} \\ &\quad + \rho^{i+1}\mathrm{Tr}\left\{(\boldsymbol{\mathcal{C}}\boldsymbol{\mathcal{F}})^i\boldsymbol{\mathcal{C}}\boldsymbol{\Omega}\boldsymbol{\mathcal{C}}^T(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{C}}^T)^i\right\} \\ &\qquad < \mathrm{Tr}\left\{(\boldsymbol{\mathcal{F}})^i\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^i\right\}\end{aligned} \tag{39}$$

In addition, for the non-cooperative ($\rho = 0$) and full-diffusion ($\rho = 1$) cases, we respectively have

$$\zeta_0 = \frac{1}{N}\sum_{i=0}^{\infty}\mathrm{Tr}\left\{(\boldsymbol{\mathcal{F}})^i\boldsymbol{\Omega}(\boldsymbol{\mathcal{F}}^T)^i\right\} \tag{40}$$

$$\zeta_{full} = \frac{1}{N}\sum_{i=0}^{\infty}\mathrm{Tr}\left\{(\boldsymbol{\mathcal{C}}^T\boldsymbol{\mathcal{F}})^i\boldsymbol{\mathcal{C}}^T\boldsymbol{\Omega}\boldsymbol{\mathcal{C}}(\boldsymbol{\mathcal{F}}^T\boldsymbol{\mathcal{C}})^j\right\} \tag{41}$$

Summing (39) for all $j \geq 0$ alongside considering (40) and (41), we arrive at

$$\zeta_{full} < \cdots < \zeta_m < \cdots < \zeta_0 \tag{42}$$

This inequality shows a communication-performance trade-off for RL-diffKF algorithm. As expected, using more neighbors in the aggregation phase gives better steady-state network MSD values.

VI. Adaptive Combiners

The objective of this section is to design the combination weights $\{c_{lk}\}$ in the aggregation phase of Algorithm 1. An optimization problem is formulated that its solution leads to calculation of a set of weights. These weights adapt to variations in the data statistics. In what follows, the optimal combiners is approximated by a stochastic gradient type algorithm. The algorithm runs in real-time, and there is no need to access the global information for its implementation.

A. Problem Formulation

To begin with, stacking all the intermediate state estimates at the end of adaptation phase in a row-wise manner and denoting it by $\boldsymbol{\Psi}_t \in \mathbb{R}^{M \times K}$ gives

$$\boldsymbol{\Psi}_t \triangleq \left[\boldsymbol{\phi}_{1,t}, \boldsymbol{\phi}_{2,t}, \ldots, \boldsymbol{\phi}_{K,t}\right]$$

We further introduce

$$\begin{aligned}\mathbf{c}_k &\triangleq \mathrm{Blkcol}\left\{c_{1k}, c_{2k}, \ldots, c_{Kk}\right\} \\ \boldsymbol{\pi}_{k,t} &\triangleq \mathrm{Blkcol}\left\{\pi_{1k,t}, \pi_{2k,t}, \ldots, \pi_{Kk,t}\right\} \\ \mathbf{f}_{k,t} &\triangleq \mathbf{c}_k \odot \boldsymbol{\pi}_{k,t} = \mathrm{Blkcol}\left\{f_{1k,t}, f_{2k,t}, \ldots, f_{Kk,t}\right\}\end{aligned}$$

where $\mathbf{c}_k$ represents the kth column of combination matrix $\mathbf{C}$, $\boldsymbol{\pi}_{k,t}$ denotes the kth column of link-selection variable vector for node k at time instant t, and $\mathbf{f}_{k,t}$ represents the combination coefficient vector for agent k at time t to be optimized. For each agent k, the objective is to obtain a set of weights $\left\{f_{\ell k,t}\right\},\ \ell = 1, \cdots, K$ that minimizes $\sum_{k=1}^{K}\|\mathbf{x}_t - \hat{\mathbf{x}}_{k,t}\|^2$. Using (9), we define the following optimization problem:

$$\begin{aligned}&\arg\min_{\mathbf{b}_{k,t}} \mathbb{E}\left[\left\|\mathbf{x}_t - ((\boldsymbol{\Psi}_t - \boldsymbol{\phi}_{k,t}\mathbb{1}_K^T)\mathbf{f}_{k,t})\right\|^2\right] \\ &\text{subject to } f_{\ell k,t} = 0 \text{ if } \ell \notin \mathcal{N}_{k,t}^{'} \text{ and } \mathbb{1}^T\mathbf{f}_{k,t} = 1\end{aligned} \tag{43}$$

The constraint $\mathbb{1}^T\mathbf{f}_{k,t} = 1$ states that the coefficients add up to unity. We now introduce an auxiliary variable $\boldsymbol{\Lambda}_k$ to decrease the dimension of (43) from K to η_k unknowns. This dimensionality reduction comes from removing the constraint $f_{lk,t} = 0$ if $\ell \notin \mathcal{N}_{k,t}^{'}$. The $K \times \eta_k$ matrix $\boldsymbol{\Lambda}_k$ is defined as

$$\boldsymbol{\Lambda}_k \triangleq [\ell - \text{th column of identity matrix } \mathbf{I}_K]_{\ell \in \mathcal{N}_{k,t}^{'}}$$

Thus, (43) can be rewritten as

$$\begin{aligned}&\min_{\mathbf{a}_{k,t}} \mathbb{E}\left[\|\mathbf{x}_t - \boldsymbol{\Pi}_{k,t}\mathbf{a}_{k,t}\|^2\right] \\ &\text{subject to } \mathbb{1}_{\eta_k}^T\mathbf{a}_{k,t} = 1\end{aligned} \tag{44}$$

where $\mathbf{a}_{t,k} \in \mathbb{R}^{\eta_k}$ contains the non-zero entries of $\mathbf{f}_{k,t}$, i.e., $\mathbf{f}_{k,t} = \boldsymbol{\Lambda}_k\mathbf{a}_{k,t}$ and $\boldsymbol{\Pi}_{k,t} = ((\boldsymbol{\Psi}_t - \boldsymbol{\phi}_{k,t}\mathbb{1}_K^T)\mathbf{f}_{k,t})\boldsymbol{\Lambda}_k$.

B. Adaptive Solution

We can minimize the cost (44) by constructing a Lagrangian regularization form as

$$\mathbf{J}\left(\mathbf{a}_{k,t},\lambda_k\right)=\mathbb{E}\left[\mathbf{x}_t^T\mathbf{x}_t\right]- \\ 2\mathbf{q}_{k,t}^T+\mathbf{a}_{k,t}^T\boldsymbol{\Theta}_{k,t}\mathbf{a}_{k,t}+2\lambda_k\left(\mathbb{1}_{\eta_k}^T\mathbf{a}_{k,t}-1\right) \quad (45)$$

where $\mathbf{q}_{k,t}=\mathbb{E}\left[\boldsymbol{\Pi}_{k,t}^T\mathbf{x}_t\right]$ and $\boldsymbol{\Theta}_{k,t}=\mathbb{E}\left[\boldsymbol{\Pi}_{k,t}^T\boldsymbol{\Pi}_{k,t}\right]$. So, simply differentiating with respect to $\mathbf{a}_{k,t}$ and setting the result to zero gives

$$-2\mathbf{q}_{k,t}^T+2\mathbf{a}_{k,t}^T\boldsymbol{\Theta}_{k,t}+2\lambda_k\mathbb{1}_{\eta_k}^T=0$$

Assuming $\boldsymbol{\Theta}_{k,t}$ is positive definite, we have

$$\mathbf{a}_{k,t}^o=\boldsymbol{\Theta}_{k,t}^{-1}\left[\mathbf{q}_{k,t}-\lambda_k^o\mathbb{1}_{\eta_k}\right] \quad (46)$$

and employing the constraint $\mathbb{1}_{\eta_k}^T\mathbf{a}_{k,t}=1$, we obtain

$$\lambda_k^o=\frac{\mathbb{1}_{\eta_k}^T\boldsymbol{\Theta}_{k,t}^{-1}\mathbf{q}_{k,t}-1}{\mathbb{1}_{\eta_k}^T\boldsymbol{\Theta}_{k,t}^{-1}\mathbb{1}_{\eta_k}} \quad (47)$$

In the sequel, a steepest-descent solution to problem (44) is derived which is then modified to obtain an adaptive solution. So, let $\mathbb{P}_{V_k}$ denote the metric projection from $\mathbb{R}^{\eta_k}$ onto $V_k=\left\{x\in\mathbb{R}^{\eta_k}:\mathbb{1}_{\eta_k}^Tx=1\right\}$, which is given by

$$\mathbb{P}_{V_k}\left(\nu\right)=\left(\mathbf{I}_{\eta_k}-\frac{\mathbb{1}_{\eta_k}\mathbb{1}_{\eta_k}^T}{\eta_k}\right)\nu+\frac{\mathbb{1}_{\eta_k}}{\eta_k} \quad \forall\nu\in\mathbb{R}^{\eta_k} \quad (48)$$

and let $\boldsymbol{\mathcal{V}}_k\triangleq\mathbf{I}_{\eta_k}-\frac{\mathbb{1}_{\eta_k}\mathbb{1}_{\eta_k}^T}{\eta_k}$, then (44) can be rewritten as the following unconstrained problem:

$$\min_{\mathbf{a}_{k,t}}\mathbb{E}\left[\left\|\mathbf{x}_t-\boldsymbol{\Pi}_{k,t}\mathbb{P}_{V_k}\left(\mathbf{a}_{k,t}\right)\right\|^2\right] \quad (49)$$

Since $\mathbb{P}_{V_k}$ is affine in $\mathbf{a}_{k,t}$, the cost (49) is quadratic and its gradient at $\mathbf{a}_{k,t}$ can be easily calculated as

$$\nabla^T\left(\mathbf{a}_{k,t}\right)=-2\boldsymbol{\mathcal{V}}_k\mathbf{q}_{k,t}+2\boldsymbol{\mathcal{V}}_k\boldsymbol{\Theta}_{k,t}\mathbb{P}_{V_k}\left(\mathbf{a}_{k,t}\right) \quad (50)$$

Therefore, we suggest minimizing (49) employing a gradient-descent algorithm as

$$\mathbf{a}_{k,t+1}=\mathbf{a}_{k,t}+\mu_{k,t}\boldsymbol{\mathcal{V}}_k\left[\mathbf{q}_{k,t}-\boldsymbol{\Theta}_{k,t}\mathbb{P}_{V_k\left(\mathbf{a}_{k,t}\right)}\right] \quad (51)$$

where $\mu_{k,t}\geq 0$ is a step-size parameter. Since $\mathbf{a}_{k,t}\in V_k$ is equivalent to $\mathbb{P}_{V_k}\left(\mathbf{a}_{k,t}\right)=\mathbf{a}_{k,t}$, our recursion is simplified as follows

$$\mathbf{a}_{k,t+1}=\mathbf{a}_{k,t}+\mu_{k,t}\boldsymbol{\mathcal{V}}_k\left[\mathbf{q}_{k,t}-\boldsymbol{\Theta}_{k,t}\mathbf{a}_{k,t}\right] \quad (52)$$

where $\mathbf{a}_{k,0}$ must satisfy $\mathbb{1}_{\eta_k}^T\mathbf{a}_{k,0}=1$. Remember that the desired coefficients can be obtained through $\mathbf{f}_{k,t}$ through $\mathbf{f}_{k,t+1}=\boldsymbol{\Lambda}_k\mathbf{a}_{k,t+1}$.

In order to derive an adaptive version of recursion (52), we replace quantity $\boldsymbol{\Theta}_{k,t}$ and $\mathbf{q}_{k,t}$ by their instantaneous approximation

$$\mathbf{q}_{k,t}=\mathbb{E}\left[\boldsymbol{\Pi}_{k,t}^T\mathbf{x}_t\right]\approx\boldsymbol{\Pi}_{k,t-1}^T\hat{\mathbf{x}}_{k,t-1|t} \quad (53)$$

$$\boldsymbol{\Theta}_{k,t}=\mathbb{E}\left[\boldsymbol{\Pi}_{k,t}^T\boldsymbol{\Pi}_{k,t}\right]\approx\boldsymbol{\Pi}_{k,t-1}^T\boldsymbol{\Pi}_{k,t-1} \quad (54)$$

Substituting these approximations in (52), the adaptive weights algorithm summarized in Algorithm 2 is obtained.

Algorithm 2 Adaptive weights for RL-diffKF

```
For every agent $k$, start with $\mathbf{a}_{k,0}\in\mathbb{R}^{\eta_k}$, such that $\mathbb{1}_{\eta_k}^T\mathbf{a}_{k,0}=1$.
Then, for every time instant $t\geq 0$, repeat
    $\begin{cases}\mathbf{a}_{k,t+1}=\mathbf{a}_{k,t}+\mu_{k,t}\boldsymbol{\mathcal{V}}_k\boldsymbol{\Pi}_{k,t-1}^T\left(\hat{\mathbf{x}}_{k,t-1|t}-\boldsymbol{\Pi}_{k,t-1}\mathbf{a}_{k,t}\right)\\ \mathbf{f}_{k,t+1}=\boldsymbol{\Lambda}_k\mathbf{a}_{k,t+1}\end{cases}$    (55)
```

VII. Empirical Evaluations

In this section, some numerical examples have been presented to evaluate the proposed algorithm[2]. We consider a network with $K=10$ nodes, where $\forall k\in\mathcal{K}$, $\min\{d_k\}=3$, $\max\{d_k\}=8$ and $\frac{1}{K}\sum_{k=1}^K d_k=5$. It is assumed that $\mathbf{x}_t$ denotes the 2-dimensional location of an object, where the time evolution of $\mathbf{x}_t$ is given by (1a) and (1b) with

$$\mathbf{F}=\begin{bmatrix}1&0&0.1&0\\0&1&0&0.1\\0&0&1&0\\0&0&0&1\end{bmatrix},\mathbf{Q}=0.001\mathbf{I}_4,\mathbf{G}=\mathbf{I}_4.$$

At any time instant t, each node obtains the noisy measurements of an unknown position based on the local data matrix $\mathbf{H}_{k,t}=\mathbf{H}_1$ or $\mathbf{H}_2$ where

$$\mathbf{H}_1=\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&0\end{bmatrix},\quad\mathbf{H}_2=\begin{bmatrix}0&1&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix}.$$

It should be noted that to guarantee the local Kalman filter convergence as in Assumption 1 the following condition should hold:

$$\forall k\in\mathcal{K},\ \exists j,\ell\in\mathcal{N}_k\quad\text{where}\quad\mathbf{H}_{j,t}=\mathbf{H}_1,\quad\mathbf{H}_{\ell,t}=\mathbf{H}_2$$

For any agent k, the covariance matrix of measurement noise is $\mathbf{R}_{k,t}=\sigma_{k,t}^2\mathbf{I}_3$, where each $\sigma_{k,t}^2$ is randomly selected in the range [0 0.5]. The observation noise variances for individual nodes are shown in Fig. 3. The curves are generated by ensemble averaging over 200 independent runs. In RL-diffKF algorithm, each node k at any time instant t receives the intermediate state estimates from $\eta_k=\min(L,d_k)$ where $0<L<d_k$. For the combination weights $\{c_{\ell k}\}$ we use the relative-degree rule. The initial values for the state vector is set to zero for all of the nodes.

Fig. 4 illustrates the learning curves of instantaneous network MSD for different values of η_k. In Fig. 5, the experimental and theoretical values of the steady-state network MSD have been compared for different values of η_k. From Figs. 4 and 5 it can be easily seen that there is a trade-off between the performance and the communication cost. Moreover, the calculated values from theoretical steady-state MSD have good agreement with the obtained simulation results. Fig. 6, shows the theoretical and experimental values of the steady-state network MSD for the proposed algorithm with different combination rules, including relative-degree, uniform and metropolis. It can be seen that for the same values for L, the proposed adaptive combination rule provides better steady-state estimation performance.

[2] Useful details or implementation of diffusion strategies with MATLAB® can be found at https://asl.epfl.ch/software/

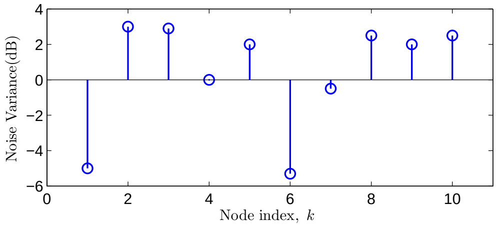

Fig. 3. The observation noise variances for individual nodes.

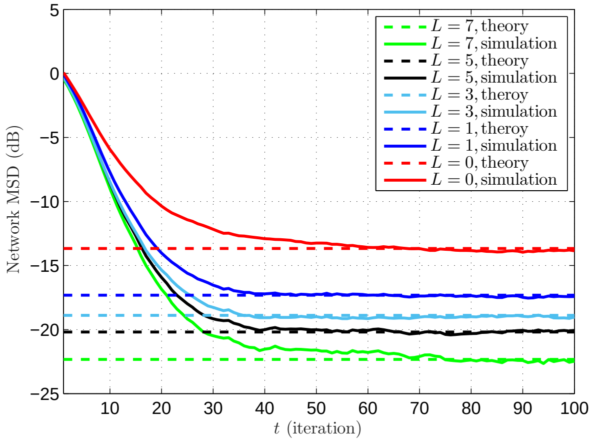

Fig. 4. Experimental and theoretical network MSD curves of the RL-diffKF algorithm with different values L.

To show the effectiveness of RL-diffKF in energy savings, we define a communication saving metric as follows

$$\mathrm{C}_{\text{saving}} = 1 - \frac{\sum_{k=1}^{K} \min\left(L, d_k\right)}{\sum_{k=1}^{K} d_k} \tag{56}$$

This quantity indicates the rate of communications saved by the RL-diffKF algorithm at each time instant t with respect to the diffKF algorithm, where $L = d_k$. Eventually, the expected performance degradation of RL-diffKF algorithm in comparison to diffKF can be described as the following performance loss metric:

$$\mathrm{P}_{\text{loss}} = \frac{\zeta_L - \zeta}{\zeta_0 - \zeta} \tag{57}$$

where ζ, ζ_L and ζ_0 denote respectively the steady-state MSD provided by diffKF, RL-diffKF, and non-cooperative KF algorithms in decibel scale. In Figs. 7 and 8, the values of $\mathrm{C}_{\text{saving}}$ and P_{loss} are plotted in terms of the maximum number of aggregated neighbors at each iteration.

Figs. 7 and 8 demonstrate that receiving intermediate state estimate only from a single neighbor per iteration results in significant reduction in the inter-node communication with respect to the diffKF case. For example, the algorithm gives 79.59% and 40% communication saving (with reference to the diffKF algorithm) for $L = 1$ and $L = 3$, respectively. Note also that the communication saving is achieved at the expense of performance loss (57.85% and 40% for $L = 1$ and $L = 3$, respectively.

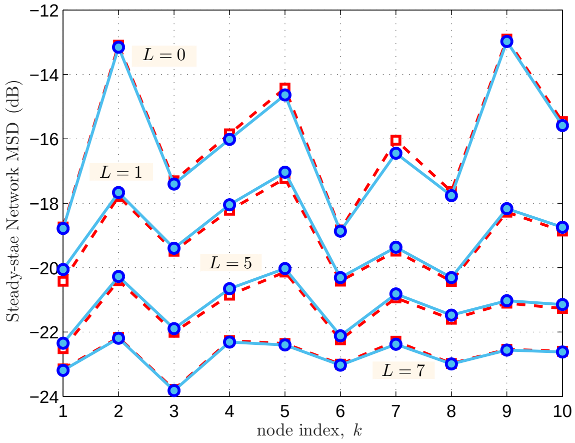

Fig. 5. Theoretical, (dash line), and experimental, (solid line), steady-state MSDs of the RL-diffKF algorithm at each node for different values of L.

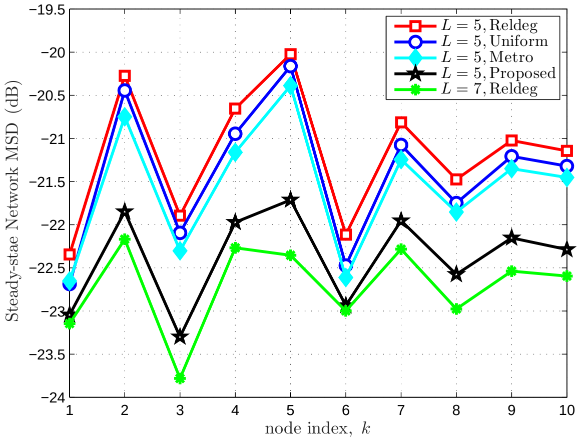

Fig. 6. Theoretical and experimental values of the steady-state network MSD for the proposed algorithm with different combination rules.

Finally, Table I compares the steady-state performance of the proposed algorithm with that of a non-cooperative system, DiffKF, and the algorithms in [24], [27]. As we expected diffKF exhibits better performance since in diffKF nodes share all available local information. The proposed algorithm with adaptive combiner provides better steady-state performance as compared to static combiner. However, it requires further computations in every node to obtain the optimum combination rules.

TABLE I
Steady-state MSD values for different algorithms.

Method	L	MSD (dB)
non-cooperative	0	-13.9
diffKF	d_k	-22.7
[24]	3	-18.8
[27]	3	-21.4
Proposed (static combiner)	3	-18.9
Proposed (adaptive combiner)	3	-21.5

Based on the above results, the following observations can be made:

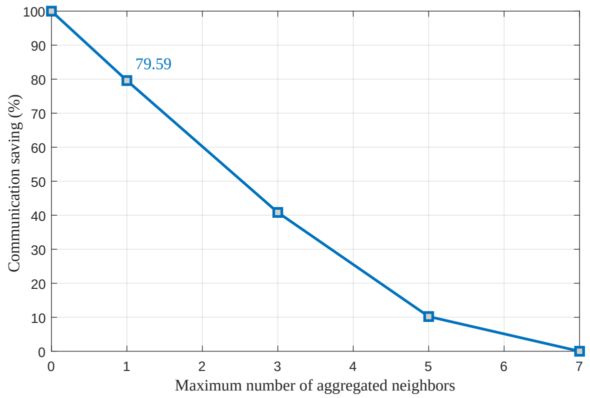

Fig. 7. Communication saving versus the maximum number of aggregated neighbors at each iteration.

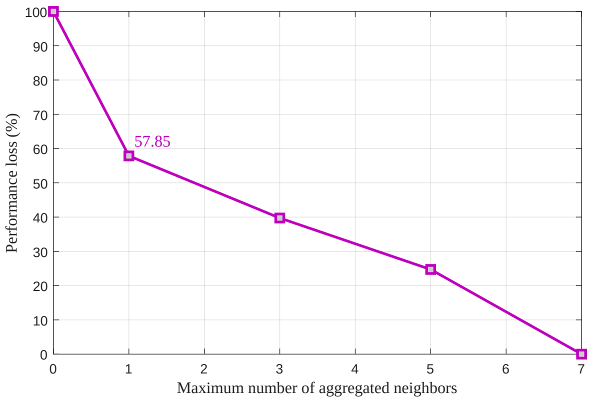

Fig. 8. Performance loss versus the maximum number of aggregated neighbors at each iteration in the experiment of Fig. 3.

- The RL-diffKF algorithm facilitates a convenient trade-off between communication burden and estimation performance.
- The RL-diffKF algorithm provides an impressive mechanism to trade steady-state MSD for decreased power-consumption and bandwidth necessities;
- The performance deterioration caused by RL-diffKF is elegant in comparison with the savings attained in transmission.
- The RL-diffKF with the proposed adaptive weights outperforms those with existing static combiners.
- A good agreement between the experimental and theoretical steady-state MSD values is achieved for a wide range of L .

VIII. Conclusion and Future Work

Here, a reduced-link diffKF (RL-diffKF) algorithm for distributed state estimation has been proposed. This algorithm reduces the communication load by allowing each node to receive the intermediate estimates from a fraction of its neighbors. We have derived expressions for the network MSD as well as the mean and mean-square stability analysis. We have also derived the optimal combination weights which minimize the steady-state MSD. The simulation results confirmed the efficiency of the proposed algorithm and verified the precision of theoretical derivations. As future work, we plan to design a modified version of diffKF algorithm that is robust against the impulse-corrupted observed data.

Appendix A

Based on the definition for $\mathfrak{B}$, we can write

$$\mathfrak{B}_t = \mathbb{E}\left[\boldsymbol{\mathcal{B}}_t \otimes \boldsymbol{\mathcal{B}}_t\right] \tag{A.1}$$

Thus, calculation of $\mathfrak{B}$ adds up to finding A.1 in the expanded form

$$\mathbb{E}\left[\mathbf{B}_t \otimes \mathbf{I}_M \otimes \mathbf{B}_t \otimes \mathbf{I}_M\right] = \begin{bmatrix} \mathbf{I}_M \otimes \mathbb{E}\left[b_{11,t}\mathbf{B}_t\right] & \cdots & \mathbf{I}_M \otimes \mathbb{E}\left[b_{1K,t}\mathbf{B}_t\right] \\ \vdots & \ddots & \vdots \\ \mathbf{I}_M \otimes \mathbb{E}\left[b_{K1,t}\mathbf{B}_t\right] & \cdots & \mathbf{I}_M \otimes \mathbb{E}\left[b_{KK,t}\mathbf{B}_t\right] \end{bmatrix} \tag{A.2}$$

Using $\mathbb{E}\left[b_{qp,t}b_{\ell k,t}\right]$ (see (A.3)) and

$$\mathbb{E}\left[\pi_{qp,t}\pi_{\ell k,t}\right] = \begin{cases} \rho_p & \text{if } p = q \ \& \ q = \ell \\ 0 & \text{if } p = q \ \& \ q \neq \ell \ \& \ d_p = 1 \\ \rho_p \frac{\eta_p - 1}{d_p - 1} & \text{if } p = q \ \& \ q \neq \ell \ \& \ d_p > 1 \\ \rho_p \rho_k & \text{if } p \neq k \end{cases}$$

Appendix B

In view of Assumption 1, we have

$$\mathbb{E}\left[\pi_{qp,t}\pi_{\ell k,t}\right] = \rho \ \forall \ p, q, k, \ell$$

Therefore, using the formulations given in Appendix A, we can show that

$$\mathbb{E}\left[\pi_{qp,t}\mathbf{B}_t\right] = \begin{cases} (1-\rho)\,\mathbf{I}_M + c_{pp}c_{qq}\rho\mathbf{I}_M & \text{if } p = q \ \& \ k = \ell \\ c_{\ell k}c_{pp}\rho\mathbf{I}_M & \text{if } p = q \ \& \ \ell \in \mathcal{N}_k \\ c_{qp}c_{kk}\rho\mathbf{I}_M & \text{if } q \in \mathcal{N}_p \ \& \ k = \ell \\ c_{qp}c_{\ell k}\rho\mathbf{I}_M & \text{if } q \in \mathcal{N}_p \ \& \ \ell \in \mathcal{N}_k \\ \mathbf{O}_M & \text{otherwise} \end{cases}$$

$$\begin{aligned} \mathbb{E}\left[\pi_{qp,t}\boldsymbol{\mathcal{B}}_t\right] &= (1-\rho)\,\mathbf{I}_{MK} + \rho c_{qp}\mathbf{C}^T \otimes \mathbf{I}_M \\ \mathbb{E}\left[\mathbf{B}_t \otimes \boldsymbol{\mathcal{B}}_t\right] &= (1-\rho)\,\mathbf{I}_{M^2K} + \rho c_{qp}\mathbf{I}_M \otimes \mathbf{C}^T \otimes \mathbf{I}_M \\ \mathbb{E}\left[\boldsymbol{\mathcal{B}}_t \otimes \boldsymbol{\mathcal{B}}_t\right] &= (1-\rho)\,\mathbf{I}_{M^2K^2} + \rho\mathbf{C}^T \otimes \mathbf{I}_M \otimes \mathbf{C}^T \otimes \mathbf{I}_M \end{aligned}$$

and finally

$$\mathfrak{B} = (1-\rho)\,\mathbf{I}_{M^2K^2} + \rho\mathbf{C}^T \otimes \mathbf{I}_M \otimes \mathbf{C}^T \otimes \mathbf{I}_M.$$

References

[1] Y. Huang, W. Yu, E. Ding, and A. Garcia-Ortiz, "EPKF: Energy efficient communication schemes based on Kalman filter for iot," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6201–6211, 2019.

[2] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: A survey on enabling technologies, protocols, and applications," *IEEE communications surveys & tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[3] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, "Industrial internet of things: Challenges, opportunities, and directions," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, 2018.

$$\mathbb{E}\left[b_{qp,t}b_{\ell k,t}\right]=\begin{cases}1-\rho_p\left(1-c_{pp}\right)-\rho_k\left(1-c_{kk}\right)\sum_{r\in\mathcal{N}_p}\sum_{s\in\mathcal{N}_k}c_{rp}c_{sk}\mathbb{E}\left[\pi_{rp,t}\pi_{sk,t}\right] & \text{if } p=q\ \&\ k=\ell\\ \rho_p c_{\ell k}-c_{\ell k}\sum_{r\in\mathcal{N}_p}c_{rp}\mathbb{E}\left[\pi_{rp,t}\pi_{\ell k,t}\right] & \text{if } p=q\ \&\ \ell\in\mathcal{N}_k\\ \rho_p c_{qp}-c_{qp}\sum_{r\in\mathcal{N}_k}c_{rk}\mathbb{E}\left[\pi_{qp,t}\pi_{rk,t}\right] & \text{if } q\in\mathcal{N}_p\ \&\ k=\ell\\ c_{qp}c_{\ell k}\mathbb{E}\left[\pi_{qp,t}\pi_{\ell k,t}\right] & \text{if } q\in\mathcal{N}_p\ \&\ \ell\in\mathcal{N}_k\\ 0 & \text{otherwise}\end{cases}\tag{A.3}$$

[4] H. M. Jawad, R. Nordin, S. K. Gharghan, A. M. Jawad, and M. Ismail, “Energy-efficient wireless sensor networks for precision agriculture: A review,” *Sensors*, vol. 17, no. 8, p. 1781, 2017.

[5] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, “Efficient IoT-based sensor big data collection–processing and analysis in smart buildings,” *Future Generation Computer Systems*, vol. 82, pp. 349–357, 2018.

[6] O. Salem, A. Serhrouchni, A. Mehaoua, and R. Boutaba, “Event detection in wireless body area networks using Kalman filter and power divergence,” *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1018–1034, 2018.

[7] N. Yu, X. Zhan, S. Zhao, Y. Wu, and R. Feng, “A precise dead reckoning algorithm based on bluetooth and multiple sensors,” *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 336–351, 2017.

[8] Z. Chen, Q. Zhu, and Y. C. Soh, “Smartphone inertial sensor-based indoor localization and tracking with iBeacon corrections,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 4, pp. 1540–1549, 2016.

[9] J. Wang, R. Zhu, and S. Liu, “A differentially private unscented Kalman filter for streaming data in iot,” *IEEE Access*, vol. 6, pp. 6487–6495, 2018.

[10] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, “Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy,” *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 591–606, 2016.

[11] F. Chen, L. Hu, P. Liu, and M. Feng, “A robust diffusion estimation algorithm for asynchronous networks in iot,” *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9103–9115, 2020.

[12] J. Zhao, M. Netto, and L. Mili, “A robust iterated extended Kalman filter for power system dynamic state estimation,” *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3205–3216, 2016.

[13] H. Wang, H. Li, J. Fang, and H. Wang, “Robust Gaussian Kalman filter with outlier detection,” *IEEE Signal Processing Letters*, vol. 25, no. 8, pp. 1236–1240, 2018.

[14] M. M. Rana and W. Xiang, “IoT communications network for wireless power transfer system state estimation and stabilization,” *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4142–4150, 2018.

[15] C. G. Lopes and A. H. Sayed, “Diffusion least-mean squares over adaptive networks: Formulation and performance analysis,” *Signal Processing, IEEE Transactions on*, vol. 56, no. 7, pp. 3122–3136, 2008.

[16] F. S. Cattivelli and A. H. Sayed, “Diffusion LMS strategies for distributed estimation,” *Signal Processing, IEEE Transactions on*, vol. 58, no. 3, pp. 1035–1048, 2010.

[17] F. S. Cattivelli, C. G. Lopes, and A. H. Sayed, “Diffusion recursive least-squares for distributed estimation over adaptive networks,” *Signal Processing, IEEE Transactions on*, vol. 56, no. 5, pp. 1865–1877, 2008.

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

[19] V. Vahidpour, A. Rastegarnia, A. Khalili, and S. Sanei, “Analysis of partial diffusion recursive least squares adaptation over noisy links,” *IET Signal Processing*, vol. 11, no. 6, pp. 749–757, 2017.

[20] M. O. Sayin, N. D. Vanli, S. S. Kozat, and T. Baar, “Stochastic subgradient algorithms for strongly convex optimization over distributed networks,” *IEEE Transactions on Network Science and Engineering*, vol. 4, no. 4, pp. 248–260, 2017.

[21] R. Arablouei, S. Werner, K. Dogancay, and Y.-F. Huang, “Analysis of a reduced-communication diffusion LMS algorithm,” *Signal Processing*, vol. 117, pp. 355–361, 2015.

[22] R. Arablouei, S. Werner, Y.-F. Huang, and K. Dogancay, “Distributed least mean-square estimation with partial diffusion,” *Signal Processing, IEEE Transactions on*, vol. 62, no. 2, pp. 472–484, 2014.

[23] R. Arablouei, K. Dogancay, S. Werner, and Y.-F. Huang, “Adaptive distributed estimation based on recursive least-squares and partial diffusion,” *Signal Processing, IEEE Transactions on*, vol. 62, no. 14, pp. 3510–3522, 2014.

[24] V. Vahidpour, A. Rastegarnia, A. Khalili, and S. Sanei, “Partial diffusion Kalman filtering for distributed state estimation in multiagent networks,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 12, pp. 3839–3846, 2019.

[25] A. Rastegarnia, “Reduced-communication diffusion RLS for distributed estimation over multi-agent networks,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2019.

[26] V. Vahidpour, A. Rastegarnia, M. Latifi, A. Khalili, and S. Sanei, “Performance analysis of distributed Kalman filtering with partial diffusion over noisy network,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 3, pp. 1767–1782, 2019.

[27] A. Khalili, V. Vahidpour, A. Rastegarnia, W. M. Bazzi, and S. Sanei, “Partial diffusion Kalman filter with adaptive combiners,” *IEEE Transactions on Aerospace and Electronic Systems*, 2020.

[28] S. Chouvardas, K. Slavakis, and S. Theodoridis, “Trading off complexity with communication costs in distributed adaptive learning via Krylov subspaces for dimensionality reduction,” *Selected Topics in Signal Processing, IEEE Journal of*, vol. 7, no. 2, pp. 257–273, 2013.

[29] S. Xie and H. Li, “Distributed LMS estimation over networks with quantised communications,” *International Journal of Control*, vol. 86, no. 3, pp. 478–492, 2013.

[30] A. Ribeiro, G. B. Giannakis, and S. I. Roumeliotis, “SOI-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations,” *IEEE Transactions on signal processing*, vol. 54, no. 12, pp. 4782–4795, 2006.

[31] C. G. Lopes and A. H. Sayed, “Diffusion adaptive networks with changing topologies,” in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008.

[32] N. Takahashi and I. Yamada, “Link probability control for probabilistic diffusion least-mean squares over resource-constrained networks,” in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 3518–3521.

[33] X. Zhao and A. H. Sayed, “Single-link diffusion strategies over adaptive networks,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 3749–3752.

[34] O. L. Rørtveit, J. H. Husøy, and A. H. Sayed, “Diffusion LMS with communication constraints,” in *2010 Conference Record of the Forty Fourth Asilomar Conference on Signals, Systems and Computers*. IEEE, 2010, pp. 1645–1649.

[35] J.-W. Lee, S.-E. Kim, and W.-J. Song, “Data-selective diffusion LMS for reducing communication overhead,” *Signal Processing*, vol. 113, pp. 211–217, 2015.

[36] M. O. Sayin and S. S. Kozat, “Single bit and reduced dimension diffusion strategies over distributed networks,” *Signal Processing Letters, IEEE*, vol. 20, no. 10, pp. 976–979, 2013.

[37] ——, “Compressive diffusion strategies over distributed networks for reduced communication load,” *Signal Processing, IEEE Transactions on*, vol. 62, no. 20, pp. 5308–5323, 2014.

[38] Y. Chen, Y. Li, G. Qi, and A. Sheng, “A neighbor selected diffusion Kalman filter over communication resource constrained networks,” *2016 35th Chinese Control Conference (CCC)*, pp. 5200–5206, 2016.

[39] A. H. Sayed, P. M. Djuri, and F. Hlawatsch, “Chapter 6 - distributed Kalman and particle filtering,” in *Cooperative and Graph Signal Processing*, P. M. Djuri and C. Richard, Eds. Academic Press, 2018, pp. 169 – 207.

[40] N. Pius Owoh and M. Mahinderjit Singh, “Sensecrypt: A security framework for mobile crowd sensing applications,” *Sensors*, vol. 20, no. 11, p. 3280, 2020.

[41] S. Yang, C. Xu, X. Qiu, and D. O. Wu, “Diffusion Kalman filter with quantized information exchange in distributed mobile crowdsensing,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4423–4435, 2018.

[42] R. de Azevedo, M. H. Cintuglu, T. Ma, and O. A. Mohammed, “Multiagent-based optimal microgrid control using fully distributed diffusion strategy,” *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1997–2008, 2017.

[43] M. M. Rana, L. Li, and S. W. Su, “An adaptive-then-combine dynamic state estimation considering renewable generations in smart grids,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3954–3961, 2016.

[44] V.-H. Bui, A. Hussain, and H.-M. Kim, “Diffusion strategy-based distributed operation of microgrids using multiagent system,” *Energies*, vol. 10, no. 7, 2017.