Distributed Tracking in Multi-hop Sensor Networks with Communication Delays

Rashedur Rahman, Murat Alanyali, *Member, IEEE,* and Venkatesh Saligrama, *Member, IEEE*

***Abstract*—We describe distributed tracking of a nonlinear dynamical system via networked sensors. The sensors communicate with each other by means of a multi-hop protocol over a communication network. In the linear setting MMSE-optimal solution is Kalman filtering when measurements are available centrally, but new methods are required to account for communication constraints. We derive in-network processing algorithms to deal with arbitrary network topology and then extend these results to account for communication delays and packet losses. We show that these algorithms are optimal in the linear setting and achieve centralized performance. The proposed techniques differ from existing techniques in two important aspects: a) there is no designated leader/fusion node and each sensor attempts to optimally track the system trajectory based on its local observations and time-dependent information available from other sensors in the network; b) the message computation at each sensor is structurally identical, where the computed message from each sensor is the innovation in the state conditioned on all the information available upto that time at each sensor. Consequently, the sensor network can be queried at any time and at any node to obtain optimal estimates for the state of the dynamical system. We then present two multihop protocols—one based on Gossip and another token-based—for distributed implementation of the in-network processing techniques. We show that the latter token-based approach leads to significant energy savings, i.e., message complexity for achieving a given tracking error performance is significantly smaller than the conventional approaches. Furthermore, we show that token-based approaches are particularly well-suited for situations where target and the network time-scales are comparable. While gossip algorithms may require to wait until aggregating all of data from the whole network to produce accurate estimates, token based approaches can provide useful estimates even in the short term since its "aggregation diameter" scales with the available time.**

I. Introduction

In recent years, there has been significant interest in sensor networks research motivated by a broad range of applications relating to national security, the environment, energy and manufacturing. Sensor networks consist of a large number of inexpensive wireless devices densely distributed over the region of interest. They are typically battery powered with limited communication and computation abilities. Each node is equipped with a variety of sensing modalities, such as acoustic, seismic, and infrared sensors.

We consider a set of sensors that collectively track a non-linear dynamical system driven by noise. The sensor measurements are noisy functions of the system state. When the system dynamics and observation models are linear, it is well-known that the centralized optimal minimum mean squared error (MMSE) estimate is given by the canonical Kalman Filter.

Sensor network applications differ in two respects. First, in many applications of interest the system dynamics and observation equations are nonlinear. In particular, the observation equations are non-linear because there is natural power law decay of sensed information with distance from the target. Second, sensor measurements are not centrally available and need to be aggregated over a communication network. Both of these aspects have been studied in the literature. A number of different techniques such as Extended Kalman Filter (EKF) [3], Unscented Kalman Filter (UKF) [16], and Particle Filtering [14] deal with target tracking in non-linear settings.

A number of papers have dealt with networked sensor measurements in the linear dynamical setting. Data accumulation techniques at a fusion center are investigated in [17]. However, the fact that sensor networks are energy limited together with the fact that transmitting state estimates consumes significantly less energy motivates development of techniques which involve transmission of a summary of past measurements (such as a state estimates) rather than the entire data trace.

Nevertheless, the main difficulty in a dynamical setting is that the measurements are not conditionally independent at the different sensors when conditioned on the current state unlike the parametric estimation case [27], [8]. It is well-known that in the static case conditional independence ensures that the centralized estimate is a weighted average of localized estimates. The output of a centralized Kalman filter can also be represented as a linear combination involving the localized estimates, however this combination has a rather delicate form that entails dynamically updating the coefficients and knowledge of global quantities [24], [25]. The lack of conditional independence has in turn motivated development of practically-appealing approximate linear filtering approaches wherein current state estimates are optimally fused but their past correlations are ignored. These techniques aim to obtain good tracking performance by judicious choice of fixed combination coefficients [4], [5], [9], [18]. Apart from being suboptimal these techniques also assume a completely connected communication infrastructure that interconnects the sensors. Motivated by these results [23], [10] have recently described approximate linear filtering for multihop networks.

However, ignoring past correlations can result in arbitrarily bad performance especially in slow dynamical situations (or small process noise). This has motivated a number of researchers to develop a decentralized fusion centric ap-

This work was supported by PECASE grant no. N00014-02-100362, NSF grants CCF 0430983, CCF 0530572, CNS 0238397.

Department of Electrical and Computer Engineering, Boston University, Boston MA 02215,Email: {rahmanr,alanyali,srv}@bu.edu

proach [24], [25], [20]. The main idea here is to compensate the state estimate either at the encoder [24] or at the decoder [25] through state augmentation, which is then shown to lead to optimal centralized Kalman filter. Our preliminary results [1] have extended these techniques to a non-fusion centric approach over multihop communication networks with packet losses and delays. Similar results were later obtained independently in [19] for multihop networks under ideal channel conditions. Tracking over multihop networks introduced in our preliminary work [1] differ from decentralized techniques in two important aspects: a) there is no designated leader/fusion node and each sensor attempts to optimally track the system trajectory based on its local observations and time-dependent information available from other sensors in the network; b) the message computation at each sensor is structurally identical, where the computed message from each sensor is the innovation in the state conditioned on all the information available upto that time at each sensor.

In the nonlinear setting distributed particle filtering approaches have been studied in [11], [22], [12] and distributed implementation has been addressed in [7]. However, these approaches generally either require substantial computational power, memory or communication overhead for each sensor.

Consequently, there is a need for energy efficient techniques for target tracking in the non-linear setting with networked sensor systems. In this paper we build on our work [1] and describe energy efficient multihop protocols for non-linear distributed target tracking problems. Our paper addresses these aspects through:

- In-network processing algorithms for realizing centralized performance
- Distributed implementation through energy efficient multihop protocols
- Accounting for situations where the target and network messaging time-scales are comparable.

For the first task we consider two scenarios, (a) fast networks where the network time-scale is significantly faster than the system dynamics; (b) slow networks where the two time scales are comparable. For the fast network scenario we present message passing schemes based on generalizations of [25], [1] for arbitrary networks and realize centralized EKF filters through distributed processing. Our scheme has the pleasing feature of *transmitting only the local innovations*. Next, we generalize these protocols to slow network scenarios. The main feature here is that the optimal centralized estimate that respects such delays differs from sensor to sensor(since sensors with superior SNRs will always have better estimates). Again we derive optimal algorithms in this context. Here at each stage a sensor computes the optimal estimate conditioned on locally available information and the new information in the form of innovations received from adjacent sensors. We next extend these schemes to account for packet losses. Consequently, the sensor network can be queried at any time and at any node to obtain optimal estimates for the state of the dynamical system.

Our next task is to develop energy efficient protocols for implementing these distributed algorithms. It turns out that a critical step in all of the aforementioned in-network processing schemes amounts to estimating the sum of sensor-node variables. One implementation is based on so called gossip algorithms [8] that compute the average of all the sensor observations. Gossip algorithms accomplish this task by randomly choosing two neighboring sensor nodes at each time and replacing their current values by their average. It turns out that this process over time converges to the average of all sensor values at all the sensors, i.e., all the sensors achieve a consensus.

Our second implementation is token-based protocols such as simple random walk/coalescing random walks [2]. In a token based algorithm a transmitting node becomes inactive and does not transmit further messages until it is reactivated by a message reception from another node. An active node generates messages at constant rate and sends each message to a randomly selected neighbor. It turns out that token-based protocols hold several advantages over gossip-protocols. First, token-based protocols lead to higher energy efficiencies, i.e., number of messages/node is significantly smaller for achieving similar performance as compared to gossip. The fundamental reason is that in gossip-protocols ad-hoc message passing results in redundant computations, i.e., the same set (or largely similar set) of nodes repeatedly fuse their information at different points in time. This problem is eliminated in token-based protocols leading to exponential savings in energy. A second advantage is that gossip protocols, unlike token-based protocols, are inexact. Averages are computed with small errors while sums can be highly inaccurate. These advantages are further amplified in a dynamic setting particularly for slow networks since it becomes all the more important to quickly fuse time relevant information. One advantage of a token-based approach is in slow networks, where the target dynamics changes faster than the time required to aggregate information from all of the sensors. While gossip algorithms may require to wait until aggregating all of data from the whole network to produce accurate estimates, token based approaches can provide useful estimates even in the short term since its “aggregation diameter” scales with the available time.

The organization of the paper is as follows. In Section II we describe the distributed target tracking problem. Section III gives a decomposition of optimal state estimate that leads to distributed in-network processing algorithms. These algorithms and their implementation through multihop communication protocols are presented in Section IV. These results are then extended to non-ideal channel conditions such as delays and packet losses in Section V. Section VI presents simulation results.

II. Problem statement

Let $\mathbb{R}^n$ denote n-dimensional real vectors, and $\mathbb{M}^{n\times n}$ denote the set of symmetric positive-definite matrices of dimension $n \times n$. We consider the discrete-time system with state vector $\mathbf{x}_t \in \mathbb{R}^n$ evolving according to the non-linear stochastic difference equation,

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t) + \mathbf{w}_t, \quad \mathbf{x}_0 \sim \mathcal{N}(0, \Sigma_0),$$

where $\mathbf{w} = (\mathbf{w}_t : t = 0, 1, 2, \cdots)$ is an IID sequence that is independent of $\mathbf{x}_0$, such that each $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q})$ for some

$\mathbf{Q} \in \mathbb{M}^{n\times n}$.

In this paper we consider tracking the sequence $(\mathbf{x}_t : t = 0, 1, 2, \cdots)$ based on measurements taken by a collection V of sensors. The measurement of sensor $v \in V$ taken at time t is denoted by $\mathbf{y}_t^v \in \mathbb{R}^m$ and it satisfies

$$\mathbf{y}_t^v = h_t^v(\mathbf{x}_t) + \mathbf{u}_t^v, \quad v \in V,$$

where $\mathbf{u} = (\mathbf{u}_t^v : t = 0, 1, 2, \cdots)$ is an IID sequence such that $\mathbf{u}_t^v \sim \mathcal{N}(0, \mathbf{R}^v)$ where $\mathbf{R}^v \in \mathbb{M}^{m\times m}$. Here $h_t^v(.)$ is a time- and sensor-dependent measurement function that accounts for possible spatio-temporal environment dependency of SNR.

Let $\mathbf{x}^*_{t|t}$ MMSE estimate of $\mathbf{x}_t$ using the measurements up to and including time t, i.e. $\mathbf{Y}_t = (\mathbf{y}_\tau^v : v \in V, \tau \leq t)$, and let $\mathbf{P}^*_{t|t}$ denote the covariance of $\mathbf{x}^*_{t|t}$. If both the system dynamics and observation models are linear then the MMSE estimate is given by $\mathbf{x}^*_{t|t} = E[\mathbf{x}_t|\mathbf{Y}_t]$, and it admits recursive computation via the Kalman filter. This practically appealing feature disappears if the linearity assumption on the functions $f(.)$ and $h_t^v(.),\ v \in V$, is relaxed, since computation of MMSE then entails the complete distribution of $\mathbf{x}_t$ conditioned on $\mathbf{Y}_t$. In this latter case a convenient approximation for the MMSE estimate can be obtain via the *Extended Kalman Filter* (EKF) provided that $f(.)$ and $h_t^v(.),\ v \in V$, are differentiable. In principle the EKF technique adopts an approximate linear system model based on first-order Taylor expansions of nonlinear components. Namely let $\mathbf{C}_t^v$ and $\mathbf{A}_t$ be matrices defined as

$$\mathbf{A}_t = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}_{t-1|t-1}}, \quad \mathbf{C}_t^v = \left.\frac{\partial h_t^v}{\partial \mathbf{x}}\right|_{\mathbf{x}^v_{t|t-1}},$$

where the dependence on state estimates $\mathbf{x}_{t-1|t-1}, \mathbf{x}^v_{t|t-1}$ is suppressed for notational convenience. The *approximate* MMSE estimate of $\mathbf{x}_t$ based on $(\mathbf{y}_\tau^v : v \in V, \tau \leq t)$ then satisfies

$$\mathbf{x}_{t|t} = \mathbf{x}_{t|t-1} + \mathbf{P}_{t|t} \sum_{v\in V} \mathbf{C}_t^{v\mathrm{T}} (\mathbf{R}^v)^{-1} (\mathbf{y}_t^v - h_t^v(\mathbf{x}_{t|t-1})), \quad (1)$$

and the covariance, $\mathbf{P}_{t|t}$ is given by the recursion:

$$\mathbf{P}^{-1}_{t|t} = \mathbf{P}^{-1}_{t|t-1} + \sum_{v\in V} \mathbf{C}_t^{v\mathrm{T}} (\mathbf{R}^v)^{-1} \mathbf{C}_t^v. \quad (2)$$

For the sake of completeness we mention the prediction steps of the Extended Kalman Filter here as well:

$$\mathbf{x}_{t+1|t} = f(\mathbf{x}_{t|t}), \quad (3)$$
$$\mathbf{P}_{t+1|t} = \mathbf{A}_t \mathbf{P}_{t|t} \mathbf{A}_t^{\mathrm{T}} + \mathbf{Q}. \quad (4)$$

Our objective in this paper is to address the tracking problem, i.e. recursive and distributed computation of $\mathbf{x}_{t|t}$, when sensors are connected via a communication network. Specifically, we consider a communication infrastructure represented by a directed graph $G = (V, E)$ whose vertex set is comprised of the sensors and each edge $(v, v') \in E$ indicates a directed communication link from sensors v to v'. In order to avoid trivialities we shall assume throughout the paper that G is strongly connected . A sensor is called a *neighbor* of sensor v if it has a communication link to v. The main goal in the following section is to study distributed algorithms to compute $\mathbf{x}_{t|t}$, where each such algorithm amounts to specification of locally generated messages that are exchanged by neighboring sensors. Obviously, this will not be possible if the network suffers from communication delays. We study distributed algorithms with communication delays in Section V.

III. In-Network Processing

A brute force approach to compute the centralized estimate would be to flood the network with raw measurements and let sensor nodes (or a designated node) construct the estimate after collecting all of the relevant values. This scheme is communication-intensive, and arguably unsuitable for energy limited sensor networks. Furthermore, its performance should be viewed in the tracking context and the impact of inherent delays in a multi-hop network on tracking performance needs to be quantified. Rather than such brute force techniques we focus on in-network data processing, which, in essence, amounts to refinement of data as it progresses in the sensor network.

The main difficulty that renders the task at hand nontrivial is that sensor measurements are not conditionally independent when conditioned on the current state. When measurements do satisfy conditional independence as in the parametric estimation case (static case) the global MMSE estimate can be recovered as a weighted average of local MMSE estimates. In other words, if $\mathrm{p}(\mathbf{y}_\tau^v, \mathbf{y}_\tau^u|\mathbf{x}_t) = \mathrm{p}(\mathbf{y}_\tau^v|\mathbf{x}_t)\ \mathrm{p}(\mathbf{y}_\tau^u|\mathbf{x}_t)$, then, we can write the global MMSE estimate, $\mathbf{x}_{t|t}$ as,

$$\mathbf{x}_{t|t} = \sum_v w_v(t) \mathbf{x}^v_{t|t}$$

where, $w_v(t)$ is a sensor and time dependent weight.

In the dynamic case although the current measurements $(\mathbf{y}_t^v : v \in V)$ are conditionally independent given the current state, the entire history is not. In fact, the past measurements are coupled through previous states, i.e.

$$\mathrm{p}(\mathbf{y}_\tau^v, \mathbf{y}_\tau^u|\mathbf{x}_t) \neq \mathrm{p}(\mathbf{y}_\tau^v|\mathbf{x}_t)\ \mathrm{p}(\mathbf{y}_\tau^u|\mathbf{x}_t),$$

where $\tau \leq t$ and $\{v, u\} \in V, v \neq u$. The main idea as we will show shortly is that if the network reached consensus on $\mathbf{x}_{t|t-t}$ it is possible to compute the optimal estimate using only $\mathbf{x}^v_{t|t} - \mathbf{x}^v_{t|t-1}$.

First, we derive a relationship between the centralized estimate and coarser estimates that are based on local observations:

Lemma 3.1: For each sensor v let $\mathbf{x}^v_{t|t} = E[\mathbf{x}_t|\mathbf{y}_\tau^v : \tau \leq t]$ be the MMSE estimate of $\mathbf{x}_t$ based on local observations $(\mathbf{y}_\tau^v : \tau \leq t)$ and let $\mathbf{P}^v_{t|t}$ be the covariance of that estimate. The fused sensor update is given by:

$$\mathbf{x}_{t|t} = \mathbf{x}_{t|t-1} + \mathbf{P}_{t|t} \sum_{v\in V} (\mathbf{P}^v_{t|t})^{-1} (\mathbf{x}^v_{t|t} - \mathbf{x}^v_{t|t-1}) \quad (5)$$
$$- \sum_{v\in V} \mathbf{C}_t^{v\mathrm{T}} (\mathbf{R}^v)^{-1} \left(h_t^v(\mathbf{x}_{t|t-1}) - h_t^v(\mathbf{x}^v_{t|t-1}) \right) (6)$$

where $\mathbf{P}_{t|t}$ is the centralized error covariance given by equality (2).

Proof: For a single sensor v, the update equation for the Kalman filter is

$$\mathbf{x}^v_{t|t} = \mathbf{x}^v_{t|t-1} + \mathbf{P}^v_{t|t} \mathbf{C}_t^{v\mathrm{T}} (\mathbf{R}^v)^{-1} (\mathbf{y}_t^v - h_t^v(\mathbf{x}^v_{t|t-1})),$$

which can be rearranged to give

$$\mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}(\mathbf{y}_t^v - h_t^v(\mathbf{x}_{t|t-1}^v)) = (\mathbf{P}_{t|t}^v)^{-1}(\mathbf{x}_{t|t}^v - \mathbf{x}_{t|t-1}^v).$$

Note that the state update for the centralized Kalman filter is given by Equation (1). It now follows that,

$$\begin{aligned}
\mathbf{P}_{t|t}^{-1}\mathbf{x}_{t|t} &= \mathbf{P}_{t|t}^{-1}\mathbf{x}_{t|t-1} + \sum_{v\in V}\mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}(\mathbf{y}_t^v - h_t^v(\mathbf{x}_{t|t-1}))\\
&= \mathbf{P}_{t|t}^{-1}\mathbf{x}_{t|t-1} + \sum_{v\in V}\mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}(\mathbf{y}_t^v - h_t^v(\mathbf{x}_{t|t-1}^v))\\
&\quad - \sum_{v\in V}\mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}(h_t^v(\mathbf{x}_{t|t-1}) - h_t^v(\mathbf{x}_{t|t-1}^v))\\
&= \mathbf{P}_{t|t}^{-1}\mathbf{x}_{t|t-1} + \sum_{v\in V}(\mathbf{P}_{t|t}^v)^{-1}(\mathbf{x}_{t|t}^v - \mathbf{x}_{t|t-1}^v)\\
&\quad - \sum_{v\in V}\mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}(h_t^v(\mathbf{x}_{t|t-1}) - h_t^v(\mathbf{x}_{t|t-1}^v)).
\end{aligned}$$

■

To interpret this lemma from an algorithmic viewpoint, suppose for a moment that the network is in a state of consensus so that the centralized state estimate $\mathbf{x}_{t-1|t-1}$ (and therefore $\mathbf{x}_{t|t-1}$) and error covariance $\mathbf{P}_{t-1|t-1}$ (and therefore $\mathbf{P}_{t|t-1}$) are available at every sensor node at time t. This implies that the last term in (6) is identically zero. In turn, a given sensor node can construct the next estimate, $\mathbf{x}_{t|t}$, if it is provided with the second term in (6). This term involves the new error covariance $\mathbf{P}_{t|t}$ and the sum of local innovations scaled by associated error covariances. We are therefore interested in distributed algorithms that lead to computation of these two quantities.

The intuition of the previous paragraph can perhaps be better formalized in the *information form* [3] of the Kalman filter. Towards this end let $\mathbf{I}_{t|t} \triangleq \mathbf{P}_{t|t}^{-1}$, define $\mathbf{z}_{t|t} \triangleq (\mathbf{P}_{t|t})^{-1}\mathbf{x}_{t|t}$ and $\mathbf{z}_{t|t-1} \triangleq (\mathbf{P}_{t|t-1})^{-1}\mathbf{x}_{t|t-1}$, and denote by $\mathbf{z}_{t|t}^v$ and $\mathbf{z}_{t|t-1}^v$ as local versions of the last two quantities respectively at sensor v. Under the aforementioned consensus condition, equality (2) can be rewritten as:

$$\begin{aligned}
\mathbf{z}_{t|t} &= \mathbf{P}_{t|t}^{-1}\mathbf{x}_{t|t-1} + \sum_{v\in V}(\mathbf{P}_{t|t}^v)^{-1}(\mathbf{x}_{t|t}^v - \mathbf{x}_{t|t-1}^v)\\
&= \left(\mathbf{P}_{t|t-1}^{-1} + \sum_{v\in V}\mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}\mathbf{C}_t^v\right)\mathbf{x}_{t|t-1}\\
&+ \sum_{v\in V}\left[(\mathbf{P}_{t|t}^v)^{-1}\mathbf{x}_{t|t}^v - \left((\mathbf{P}_{t|t-1}^v)^{-1} + \mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}\mathbf{C}_t^v\right)\mathbf{x}_{t|t-1}^v\right] \qquad (7)\\
&= \mathbf{z}_{t|t-1} + \sum_{v\in V}\mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}\mathbf{C}_t^v\mathbf{x}_{t|t-1}\\
&\quad + \sum_{v\in V}\left(\mathbf{z}_{t|t}^v - \mathbf{z}_{t|t-1}^v\right) - \sum_{v\in V}\mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}\mathbf{C}_t^v\mathbf{x}_{t|t-1}^v \qquad (8)\\
&= \mathbf{z}_{t|t-1} + \sum_{v\in V}\left(\mathbf{z}_{t|t}^v - \mathbf{z}_{t|t-1}^v\right).
\end{aligned}$$

Note also that, with $\mathbf{I}_{t|t-1} \triangleq \mathbf{P}_{t|t-1}^{-1} = (\mathbf{A}_t\mathbf{I}_{t-1|t-1}^{-1}\mathbf{A}_t^{\mathrm{T}} + \mathbf{Q})^{-1}$, equality (2) yields

$$\mathbf{I}_{t|t} = \mathbf{I}_{t|t-1} + \sum_{v\in V}\mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}\mathbf{C}_t^v. \qquad (9)$$

Therefore, in order to achieve centralized Kalman filter estimate, the sensors need to compute two sums, namely $\sum_{v\in V}\left(\mathbf{z}_{t|t}^v - \mathbf{z}_{t|t-1}^v\right)$ and $\sum_{v\in V}\mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}\mathbf{C}_t^v$ in a distributed fashion.

We finally note for linear dynamical systems, transmitting local innovation instead of measurements has communication advantages since it requires less energy as innovation power is small and stable. We present the fact as the following lemma,

Lemma 3.2: Let $\tilde{\mathbf{S}}_t = \mathbf{E}[(\mathbf{y}_t - \mathbf{C}_t\mathbf{x}_{t|t-1})(\mathbf{y}_t - \mathbf{C}_t\mathbf{x}_{t|t-1})^{\mathrm{T}}]$ be the innovation power for some sensor (where we have dropped the superscript v identifying the sensor here for simplicity of exposition). The dynamical system and the measurements are assumed to be linear. Then

$$\mathbf{E}[\mathbf{y}_t\mathbf{y}_t^{\mathrm{T}}] \geq \tilde{\mathbf{S}}_t. \qquad (10)$$

Proof: Using the definition of innovation power we have:

$$\begin{aligned}
\tilde{\mathbf{S}}_t &= \mathbf{E}[(\mathbf{y}_t - \mathbf{C}_t\mathbf{x}_{t|t-1})(\mathbf{y}_t - \mathbf{C}_t\mathbf{x}_{t|t-1})^{\mathrm{T}}]\\
&= \mathbf{C}_t\mathbf{P}_{t|t-1}\mathbf{C}_t^{\mathrm{T}} + \mathbf{R},\\
\mathbf{P}_{t+1|t} &= \mathbf{A}_t\mathbf{P}_{t|t}\mathbf{A}_t^{\mathrm{T}} + \mathbf{Q}\\
&= \mathbf{A}_t\mathbf{P}_{t|t-1}\mathbf{A}_t^{\mathrm{T}} + \mathbf{Q} - \mathbf{A}_t\mathbf{P}_{t|t-1}\mathbf{C}_t^{\mathrm{T}}\tilde{\mathbf{S}}_t^{-1}\mathbf{C}_t\mathbf{P}_{t|t-1}\mathbf{A}_t^{\mathrm{T}}.
\end{aligned}$$

Let $\mathbf{S}_t = \mathbf{E}[\mathbf{x}_t\mathbf{x}_t^{\mathrm{T}}]$, then data power is given by:

$$\begin{aligned}
\mathbf{E}[\mathbf{y}_t\mathbf{y}_t^{\mathrm{T}}] &= \mathbf{C}_t\mathbf{S}_t\mathbf{C}_t^{\mathrm{T}} + \mathbf{R}\\
\mathbf{S}_{t+1} &= \mathbf{A}_t\mathbf{S}_t\mathbf{A}_t^{\mathrm{T}} + \mathbf{Q}.
\end{aligned}$$

Using the initial condition $\mathbf{S}_0 = \mathbf{P}_{0|-1}$ and by noting that the second part of $\mathbf{P}_{t+1|t}$ is always positive semi-definite, we have,

$$\mathbf{S}_{t+1} \geq \mathbf{P}_{t+1|t} \Rightarrow \mathbf{E}[\mathbf{y}_t\mathbf{y}_t^{\mathrm{T}}] \geq \tilde{\mathbf{S}}_t.$$

This also hold true for nonlinear systems assuming the linearization error is small compared to state covariance. ■

IV. Distributed Algorithms

In this section we consider distributed tracking algorithms in the case when the network is significantly faster than the dynamics of the system. Namely, in the scope of this section, neighboring sensors can exchange messages multiple times between consecutive measurements. The complementary case when the network and the system have a common time-scale will be considered in Section V.

The present scenario can be conveniently abstracted by a time-scale separation argument in which the system time is frozen and the network has infinite time to share the data. We shall continue to use t as the system time, and consistently refer to the network time with k. Hence, for example, $\tilde{\mathbf{m}}_t^v(k)$ will denote the k^{th} message sent by sensor v in the t^{th} system time (that is, between taking the t^{th} and $(t+1)^{st}$ measurements).

The algorithms studied in the next two subsections are characterized as follows. At each system time t each sensor v makes a local observation and computes the state estimate $\mathbf{x}_{t|t}^v$ based on the current observation and past information. The objective of the algorithms is to reach consensus by exchanging messages before the next sensing interval and thus maintain the optimal global estimate available at any time at all sensors. Without loss of generality assume scalar messages and denote their vector form by

$$\tilde{\mathbf{M}}_t(k) = \left[\tilde{\mathbf{m}}_t^1(k) \quad \tilde{\mathbf{m}}_t^2(k) \quad \cdots \quad \tilde{\mathbf{m}}_t^v(k) \quad \cdots \quad \tilde{\mathbf{m}}_t^V(k)\right]^{\mathrm{T}}.$$

At each network time messages are updated linearly as

$$\tilde{\mathbf{M}}_t(k) \quad = \quad \mathbf{B}_k \tilde{\mathbf{M}}_t(k-1), \tag{11}$$

where $\mathbf{B}_k = [b_k(u,v)]_{V \times V}$ is a weight matrix for the k^{th} messaging round such that $b_k(u,v) > 0$ only if sensor u is a neighbor of sensor v. Each algorithm is specified via a different choice of weight matrices. Note that if this choice leads to

$$\lim_{k\to\infty} \tilde{\mathbf{m}}_t^v(k) = \sum_{u \in V} \tilde{\mathbf{m}}_t^u(0), \tag{12}$$

for some sensor v, then either sum $\sum_{v\in V}\left(\mathbf{z}^v_{t|t} - \mathbf{z}^v_{t|t-1}\right)$ or $\sum_{v \in V} \mathbf{C}_t^{v\mathrm{T}} (\mathbf{R}^v)^{-1} \mathbf{C}_t^v$ can be constructed at each sensor if $\tilde{\mathbf{m}}_t^v(0)$ is initialized as $\left(\mathbf{z}^v_{t|t} - \mathbf{z}^v_{t|t-1}\right)$ or $\mathbf{C}_t^{v\mathrm{T}} (\mathbf{R}^v)^{-1} \mathbf{C}_t^v$ respectively at each sensor v. Thus using (9) and (9), we have the consensus of centralized Kalman filter estimate at each sensor at every time t. We shall show that (12) can be achieved simultaneously at all sensors by proper choice of $\mathbf{B}_k$.

A. Generalized Belief Propagation

We start by seeking weight matrices $\mathbf{B}_k$ such that each entry of $\tilde{\mathbf{M}}_t(k)$ converges to a scaled version of the sum $\sum_{v\in V} \tilde{\mathbf{m}}_t^v(0)$ of initial values. That is,

$$\lim_{k\to\infty} \tilde{\mathbf{M}}_t(k) = \lim_{k\to\infty} \mathbf{B}_k \mathbf{B}_{k-1} \ldots \mathbf{B}_1 \tilde{\mathbf{M}}_t(0) = c(\mathbf{1}\mathbf{1}^{\mathrm{T}})\tilde{\mathbf{M}}_t(0),$$

for some *known* constant c (here $\mathbf{1}$ is the vector of all 1's). It will be convenient to choose $c = |V|^{-1}$. Let us further consider adopting the same communication rule in each messaging round by setting $\mathbf{B}_k$ to some suitable matrix $\mathbf{B}$. This latter matrix then needs to satisfy $\lim_{k\to\infty} \mathbf{B}^k = \mathbf{1}\mathbf{1}^{\mathrm{T}}/|V|$, which holds if

$$\mathbf{1}^{\mathrm{T}}\mathbf{B} = \mathbf{1}^{\mathrm{T}}, \quad \mathbf{B}\mathbf{1} = \mathbf{1}, \quad \rho(\mathbf{B} - \mathbf{1}\mathbf{1}^{\mathrm{T}}/|V|) < 1,$$

where $\rho(.)$ denotes the spectral radius of its argument. The following two specifications for $\mathbf{B}$ are doubly stochastic and therefore both satisfy the conditions above, so, at all time-steps k, the weights can be chosen according to one of the following rules:

Maximum-degree weights

$$B(u,v) = \begin{cases} \frac{1}{|V|}, & u \neq v,\ (u,v) \in E \\ 1 - \frac{deg(u)}{|V|}, & u = v \\ 0, & \text{otherwise,} \end{cases}$$

Metropolis weights

$$B(u,v) = \begin{cases} \frac{1}{1+\max(deg(u),deg(v))}, & u \neq v,\ (u,v) \in E \\ 1 - \sum_{v:(u,v)\in E} b(u,v), & u = v \\ 0, & \text{otherwise.} \end{cases}$$

Here $deg(u)$ refers to the out-degree of (i.e. the number of edges emanating from) sensor u in the communication graph G.

Theorem 4.1: Consider two parallel instantiations $\tilde{\mathbf{M}}\mathbf{1}_t, \tilde{\mathbf{M}}\mathbf{2}_t$ of the decentralized updating scheme presented above. Suppose that initial messages are set to $\tilde{\mathbf{m}\mathbf{1}}_t^v(0) = \mathbf{C}_t^{v\mathrm{T}} (\mathbf{R}^v)^{-1} \mathbf{C}_t^v$ and $\tilde{\mathbf{m}\mathbf{2}}_t^v(0) = \left(\mathbf{z}^v_{t|t} - \mathbf{z}^v_{t|t-1}\right)$ for each sensor v. Then

$$\begin{aligned} \lim_{k\to\infty} \left(\mathbf{P}^{-1}_{t|t-1} + |V|\tilde{\mathbf{m}\mathbf{1}}_t^v(k)\right)^{-1} &= \mathbf{P}_{t|t}, \\ \lim_{k\to\infty} \mathbf{P}^v_{t|t}(k)\left(\mathbf{x}_{t|t-1} + |V|\tilde{\mathbf{m}\mathbf{2}}_t^v(k)\right) &= \mathbf{x}_{t|t}, \end{aligned}$$

where $\mathbf{x}_{t|t}, \mathbf{P}_{t|t}$ are the centralized state estimate and the error covariance matrix respectively conditioned on sensor data upto time t for all of the sensors.

Proof: The weight matrix $\mathbf{B}$ is chosen such that, $\lim_{k\to\infty} \mathbf{B}^k = \mathbf{1}\mathbf{1}^{\mathrm{T}}/|V|$; so for $i = 1, 2$ we have

$$\lim_{k\to\infty} \tilde{\mathbf{M}}\mathbf{i}_t(k) \quad = \quad \lim_{k\to\infty} \mathbf{B}^k \tilde{\mathbf{M}}\mathbf{i}_t(0) = \left(\sum_{v\in V} \tilde{\mathbf{m}\mathbf{i}}_t^v(0)/|V|\right)\mathbf{1}.$$

In turn, by choice of the initial messages,

$$\lim_{k\to\infty} \left(\mathbf{P}^{-1}_{t|t-1} + |V|\tilde{\mathbf{m}\mathbf{1}}_t^v(k)\right)^{-1} = \left(\mathbf{P}^{-1}_{t|t-1} + \sum_{v\in V} \mathbf{C}_t^{v\mathrm{T}} (\mathbf{R}^v)^{-1} \mathbf{C}_t^v\right)^{-1} = \mathbf{P}_{t|t}$$

$$\begin{aligned} \lim_{k\to\infty} \mathbf{P}^v_{t|t}(k)\left(\mathbf{x}_{t|t-1} + |V|\tilde{\mathbf{m}\mathbf{2}}_t^v(k)\right) \quad &= \quad \mathbf{P}_{t|t}\left(\mathbf{P}^{-1}_{t|t-1}\mathbf{x}_{t|t-1} + \sum_{v\in V} \mathbf{z}^v_{t|t} - \mathbf{z}_{t|t} \right. \\ &= \quad \mathbf{x}_{t|t}. \end{aligned}$$

■

Remark 4.1: Convergence rates of both algorithms are determined by the second largest eigenvalue of the weight matrix $\mathbf{B}$ (see, for example [15, Theorem 8.5.1]).

B. Asynchronous Algorithms and Packet Losses

The synchronous nature of the algorithms in the previous section is to some extent an artifact of the formulation, since these algorithms can in fact be implemented in an event-driven manner by programming each sensor to send out its k^{th} message only *after* receiving the $(k-1)^{th}$ messages from all of its neighbors. Despite its practical appeal, such an implementation has drawbacks since the overall rate of message exchange in the network would be determined by the slowest sensor therein. Furthermore, packet losses would have significant negative effect on the network-wide computation. Our goal in this section is to consider a generalized version of network operation that addresses these shortcomings.

Given sensors u, v we shall say that link $u \to v$ is *functional* in round k if sensor v receives a message from sensor u in that round. The link is then non-functional if either sensor u does not send a message or such a message is lost in transmission. Note that since in an asynchronous setting not all links are necessarily functional at each messaging round, the weight matrix $\mathbf{B}_k$ cannot be taken constant for all rounds k. Let us express the evolution (11) of messages in a form that is convenient for the purposes of this section:

$$\tilde{\mathbf{M}}_t(k) = (\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_2^{\mathrm{T}} \cdots \mathbf{B}_k^{\mathrm{T}})^{\mathrm{T}} \tilde{\mathbf{M}}_t(0), \quad k \geq 1, \tag{13}$$

where $^{\mathrm{T}}$ indicates matrix transpose. We shall consider the distributed algorithm specified by choosing $\mathbf{B}_k = (\mathbf{I} + \mathbf{F}_k)\mathbf{D}_k$ where $\mathbf{F}_k = [f_k(u,v)]_{V\times V}$ is a binary matrix with

$$f_k(u,v) = \begin{cases} 1 & \text{if link } v \to u \text{ is functional at round } k \\ 0 & \text{else,} \end{cases}$$

and $\mathbf{D}_k = [d_k(u,v)]_{V\times V}$ is a diagonal matrix with

$$d_k(v,v) = \left(1 + \sum_u f_k(u,v)\right)^{-1}.$$

In particular columns of $\mathbf{B}_k$ are probability vectors; and thus each $\mathbf{B}_k^{\mathrm{T}}$ is a stochastic matrix.

The system (13) describes the evolution of local messages when each transmitted message is normalized by the number of outgoing functional links (i.e., the number of receivers of the message) in the same round. Inspection of the matrices $\mathbf{B}_k$ yields that each transmitting sensor distributes its current value equally to its recipients who simply add the message to their own values. In particular the sum of all sensor values is invariant under the algorithm. Note that the identity matrix in $\mathbf{B}_k$ indicates a self-loop at each sensor node that is always functional; so a transmitting sensor is also a recipient of its own message.

Theorem 4.2: Suppose that the matrices $(\mathbf{F}_k : k \geq 1)$ are IID, and that $E[\mathbf{F}_1]$ is irreducible. Then for each $v \in V$ there exists a random sequence $(\gamma_k^v : k \geq 0)$ such that

$$\lim_{k\to\infty} \frac{\tilde{\mathbf{m}}_t^v(k)}{\gamma_k^v} = \sum_{u\in V} \tilde{\mathbf{m}}_t^u(0), \quad \text{almost surely.}$$

We prove the theorem via an adaptation of the techniques in [26] for asymptotic analysis of stochastic-matrix products. We start with some auxiliary results. Given a square matrix $\mathbf{P} = [p_{nm}]$, define

$$\lambda(\mathbf{P}) = 1 - \min_{n_1,n_2} \sum_m \min(p_{n_1 m}, p_{n_2 m})$$

Lemma 4.1: Under the hypothesis of Theorem 4.2, for each $\epsilon > 0$ there exists $k(\epsilon)$ such that for $k \geq k(\epsilon)$,

$$P(\lambda(\mathbf{B}_{k_0+1}^{\mathrm{T}}\mathbf{B}_{k_0+2}^{\mathrm{T}}\cdots\mathbf{B}_{k_0+k}^{\mathrm{T}}) < 1) > 1-\epsilon, \quad k_0 \geq 0$$

Proof: It suffices to show that for large enough k all entries of the matrix product

$$\mathbf{B}_{k_0+1}^{\mathrm{T}}\mathbf{B}_{k_0+2}^{\mathrm{T}}\dots\mathbf{B}_{k_0+k}^{\mathrm{T}}$$

are positive with probability at least $1-\epsilon$. By definition of $\mathbf{B}_k^{\mathrm{T}}$s, entries of this product are positive if and only if all entries in

$$(\mathbf{I}+\mathbf{F}_{k_0+1})^{\mathrm{T}}(\mathbf{I}+\mathbf{F}_{k_0+2})^{\mathrm{T}}\cdots(\mathbf{I}+\mathbf{F}_{k_0+k})^{\mathrm{T}} \tag{14}$$

are positive. The $(v,v')^{th}$ entry in the product (14) is positive if and only if a hypothetical message that originates at node v in round k_0 can reach node v' by round k_0+1 by traversing a functional link in each round. Note that a self-looping link is always functional due to the identity matrix contained in each factor of (14). Let $q(v,v')$ be the probability that link (v,v') is functional at a round, so that without loss of generality $E = \{(v,v') : q(v,v') > 0\}$, and let $(\phi(v,v') : (v,v') \in E)$ be independent geometric random variables where $\phi(v,v')$ has parameter $q(v,v')$. Since $E[\mathbf{F}_1]$ is irreducible by hypothesis, the time to reach any node from any other node via functional links is stochastically dominated by $\sum_{(v,v')\in E}\phi(v,v')$. Define the random variable κ as

$$\kappa = \min\{k : (\mathbf{I}+\mathbf{F}_{k_0+1})^{\mathrm{T}}(\mathbf{I}+\mathbf{F}_{k_0+2})^{\mathrm{T}}\cdots(\mathbf{I}+\mathbf{F}_{k_0+k})^{\mathrm{T}} \text{ has positive entries}\}$$

Since there are $|V|^2$ node pairs, κ is stochastically dominated by $|V|^2\sum_{(v,v')\in E}\phi(v,v')$. Let μ be the mean of this latter variable so that,

$$P(\lambda(\mathbf{B}_{k_0+1}^{\mathrm{T}}\mathbf{B}_{k_0+2}^{\mathrm{T}}\cdots\mathbf{B}_{k_0+k}^{\mathrm{T}}) < 1) \geq 1 - P(\kappa > k) \geq 1 - \frac{\mu}{k},$$

where last inequality is an application of Markov's inequality. The lemma follows by choosing $k(\epsilon) = \mu/\epsilon$. ■

Corollary 4.1: Since each $\mathbf{B}_k$ takes values from a finite set, there exists a positive number $d < 1$ such that $\lambda(\mathbf{B}_{k_0+1}^{\mathrm{T}}\mathbf{B}_{k_0+2}^{\mathrm{T}}\dots\mathbf{B}_{k_0+k}^{\mathrm{T}}) < 1$, for $k_0 \geq 0$.

For a square matrix $\mathbf{P} = [p_{nm}]$ define

$$\delta(\mathbf{P}) = \max_m \max_{n_1,n_2} |p_{n_1,m} - p_{n_2,m}|$$

The following lemma is a recitation of [29,Lemma 2] and is given without proof:

Lemma 4.2: For $k \geq 1$

$$\delta(\mathbf{B}_k^{\mathrm{T}}\mathbf{B}_{k-1}^{\mathrm{T}}\cdots\mathbf{B}_1^{\mathrm{T}}) \leq \prod_{l=1}^{k}\lambda(\mathbf{B}_l^{\mathrm{T}})$$

Proof of Theorem 4.2 Fix $\sigma, \epsilon > 0$ and $k > k(\epsilon)$. Appeal to Lemma 4.2 to write,

$$\delta(\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_2^{\mathrm{T}}\cdots\mathbf{B}_k^{\mathrm{T}}) \leq \prod_{l=1}^{k \bmod k(\epsilon)}\lambda(\mathbf{B}_l^{\mathrm{T}})\prod_{l=1}^{k/k(\epsilon)}\lambda(\mathbf{B}_{k-lk(\epsilon)+1}^{\mathrm{T}}\mathbf{B}_{k-lk(\epsilon)+2}^{\mathrm{T}}\cdots\mathbf{B}_k^{\mathrm{T}}$$

Since each factor of the product on the left hand side is at most 1, Corollary 4.1 implies that the product is larger than σ only if there are more than $\lfloor\log_d(\sigma)\rfloor$ values of l with

$$\lambda(\mathbf{B}_{k-lk(\epsilon)+1}^{\mathrm{T}}\mathbf{B}_{k-lk(\epsilon)+2}^{\mathrm{T}}\dots\mathbf{B}_{k-(l-1)k(\epsilon)}^{\mathrm{T}}) > d$$

Lemma 4.1 now implies that for $k > k(\epsilon)\lfloor\log_d(\sigma)\rfloor$

$$\begin{aligned} P(\delta(\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_2^{\mathrm{T}}\dots\mathbf{B}_k^{\mathrm{T}}) > \sigma) &\leq \sum_{l=1}^{\lfloor\log_d(\sigma)\rfloor}\binom{\lfloor k/k(\epsilon)\rfloor}{l}(1-\epsilon)^{\lfloor k/k(\epsilon)\rfloor - l}\epsilon^l \\ &\leq (1-\epsilon)^{k/k(\epsilon)}k^{\lfloor\log_d(\sigma)\rfloor}c, \end{aligned}$$

where c does not depend on k. The left hand side is thus summable in k; in turn

$$\lim_{k\to\infty}\sup\delta(\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_2^{\mathrm{T}}\dots\mathbf{B}_k^{\mathrm{T}}) \leq \sigma, \quad \text{almost surely}$$

due to the Borel-Cantelli Lemma. Arbitrariness of σ implies that $\delta(\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_2^{\mathrm{T}}\dots\mathbf{B}_k^{\mathrm{T}})$ converges, hence by definition the rows of the product $\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_2^{\mathrm{T}}\dots\mathbf{B}_k^{\mathrm{T}}$ almost surely become identical (though they do not necessarily settle to a fixed vector). In light of equality (13) the theorem follows by identifying γ_k^v with the vth entry of an arbitrary row of $\mathbf{B}_1^{\mathrm{T}}\mathbf{B}_2^{\mathrm{T}}\dots\mathbf{B}_k^{\mathrm{T}}$. □

Theorem 4.2 states that the sum $\sum_{u\in V}\tilde{\mathbf{m}}_t^u(0)$ of interest can be obtained at a sensor v provided that the random sequence $(\gamma_k^v : k \geq 0)$ associated with that sensor is known. Statistics of this sequence depend on the network topology as well as the frequency of message transmissions at each node in the network, and furthermore the sequence itself need not be convergent. An appealing technique to compute γ_k^v with little overhead involves a pilot signal, which entails augmenting messages with a redundant entry that has a globally known initialization. Specifically, let $\hat{\mathbf{m}}_t^v(k)$ be an augmented message that is initialized to, say, $\hat{\mathbf{m}}_t^v(0) = 1$ at each sensor v

and is transmitted along with each message $\tilde{\mathbf{m}}_t^v(k)$. Then, by Theorem 4.2, $\hat{\mathbf{m}}_t^v(k)/\gamma_k^v \to |V|$; and therefore $\hat{\mathbf{m}}_t^v(k)/|V|$ can be adopted as an asymptotically accurate proxy to γ_k^v.

We finally note that the proof of Theorem 4.2 relies only on Lemma 4.2; hence the conclusion of the theorem holds under more relaxed assumptions on the statistics of $(\mathbf{F}_k : k \geq 1)$. From a deterministic perspective, conclusion of the theorem holds if each link is functional infinitely often, provided that the incidence matrix of the communication graph is irreducible and aperiodic.

C. Gossip Algorithms

For purposes of comparison we also consider *gossip algorithms* that have origins database applications and have recently been proposed for sensor networks [8]. In broad terms, gossip algorithms refer to distributed randomized algorithms that are based on pairwise relaxations between randomly chosen node pairs. In the context of the present section a pairwise relaxation refers to averaging the two values available at the associated nodes. For completeness we briefly describe the algorithm here. A more detail description of this algorithm can be found in [8]. The algorithm is specified by a stochastic matrix $P = [P_{uv}]_{V\times V}$ such that $P_{uv} > 0$ only if nodes u and v are neighbors in G. At round k each node makes a randomized to relax its value in that round, and then chooses a neighbor u with respect to the distribution $(P_{vu} : u \in V)$. Both nodes, say nodes u and v, then update their local variables by setting them to the common value $\tilde{\mathbf{m}}_t^v(k+1) = \tilde{\mathbf{m}}_t^u(k+1) = (\tilde{\mathbf{m}}_t^v(k) + \tilde{\mathbf{m}}_t^u(k))/2$. It has been shown that under this procedure $\lim_{k\to\infty} \tilde{\mathbf{m}}_t^v(k) = |V|^{-1}\sum_u \tilde{\mathbf{m}}_t^u(0)$, at each node v in the network [8]. Note that gossip algorithms bear qualitative similarities to the asynchronous algorithms of the previous section. In fact the two algorithms have very similar convergence properties.

D. Token Based Algorithms

Wireless sensor nodes are typically equipped with limited power source and, in most realistic scenarios, replenishment of power resources might be impractical. Hence, energy-efficiency is of additional importance when designing distributed algorithms for sensor networks. Another apparent advantage of token-based schemes is that one can trade space for time, i.e., the scheme aggregates information only from nodes reachable in short time. This is particularly advantageous in slow networks where the target dynamics and network messaging time-scales are comparable. In this section we consider distributed algorithms to compute a global estimate with emphasis on energy-efficiency.

We introduce two algorithms that have identical dynamic specification but differ in their initialization. Similar to the previously considered algorithms, each algorithm is executed after the sensors make a new observation at time t and until the next observation at time $t+1$. At this time all the sensor update their local estimate using the information received so far and the new measurement, and the same procedure is repeated in the next sensing period.

Our main objective here is to regulate message transmissions in a distributed fashion so as to reduce unnecessary transmissions and thereby to achieve energy gains. This goal is achieved by deactivating each transmitting sensor until the time it becomes reactivated by reception of a message. Formal specification of the algorithms is as follows: We continue to refer to the network time by k. Each node $v \in V$ maintains a status variable $\xi_t^v(k)$ which is either *'active'* (i.e. 1) or *'idle'* (i.e. 0). The value of $\xi_t^v(k)$ indicates whether sensor v is eligible to transmit a message in round k, as will be explained next. The initial value $\xi_t^v(0)$ depends on the particular algorithm in the following fashion: Under *simple-walk* $\xi_t^v(0) = 1$ for exactly one node, say node v_0, whereas $\xi_t^v(0) = 1$ for all nodes v under *coalescent*.

Let $\tilde{\mathbf{m}}_t^v(k)$ continue to denote the value (equivalently the message) maintained at node v. At each messaging round each active node sends a message, say with some probability $\lambda \in (0,1)$. Such a sensor, say sensor u, sends its message to a randomly chosen neighbor and sets its own value to $\tilde{\mathbf{m}}_t^u(k+1) = 0$ and declares itself as idle $\xi_t^u(k+1) = 0$. For clarity of exposition we impose that a node is either a sender or a receiver in a given round, but not both. The receiving neighbor, say sensor v, simply adds the message to its own value and declares itself active $\xi_t^v(k+1) = 1$. Note that idle sensors have value 0, therefore the value of the sender passes to the receiver if the latter is idle, or its added to the present value otherwise:

$$\tilde{\mathbf{m}}_t^v(k+1) = \begin{cases} \tilde{\mathbf{m}}_t^u(k) + \tilde{\mathbf{m}}_t^v(k) & \text{if } \xi_t^v(k) = 1 \\ \tilde{\mathbf{m}}_t^u(k) & \text{else.} \end{cases}$$

In both algorithms an active node can be interpreted to be holding a transmit token that moves with the transmitted message. Reflecting on the algorithm specification reveals that two such tokens coalesce into one if they meet at the same node. Clearly, *simple-walk* maintains a single such token in the network. An illustrative scenario that depicts the evolution of token locations (equivalently of active nodes) under *coalescent* is given by Figure 1.

1) Correctness and complexity: Note that under both algorithms, the value of an idle node is 0 if it has ever been active in the past. Both algorithms then maintain the network-wide sum of values invariant:

Proposition 4.1: Under both *simple-walk* and *coalescent*,

$$\sum_{v\in V} \tilde{\mathbf{m}}_t^v(k) = \sum_{v\in V} \tilde{\mathbf{m}}_t^v(0), \qquad \text{for all } k \geq 0.$$

Proof: The proof is straightforward via induction on k, and is omitted here. ■

To determine the stopping criteria for the two algorithms, let us define the random times τ_S, τ_C as follows:

$$\begin{aligned} \tau_S &= \inf\{k : \text{ each node becomes active by time } k\ \} \\ \tau_C &= \inf\{k : \sum_{v\in V} \xi_t^v(k) = 1\}. \end{aligned}$$

Note that under algorithm *coalescent*, there exists a single active node in the network for $k > \tau_C$. Since all nodes start out active under this algorithm, the values of the remaining

Fig. 1. An illustration of evolution of active nodes in algorithm *coalescent*. The white circles represent the active nodes at some arbitrary time t_0 and the arrow represent the next message transmission after t_o. After transmission, Node 20 becomes active and the value of the transmitting node 8 passes onto the receiver. Active number of nodes in the network remains the same. If the next transmission is from node 17 to node 20, then number of active nodes decrease by one.

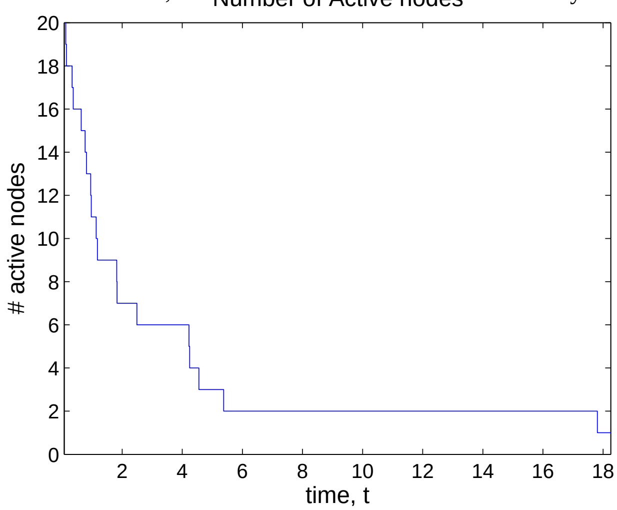


Fig. 2. A typical sample path of the number of active nodes under the algorithm *coalescent*. The algorithm terminates when a single active node remains, which occurs around time 18.

$n-1$ idle nodes are all 0 for such k . Under *simple-walk* there is always one active node, but all nodes become active at least once by time τ_S. Therefore the $n-1$ idle nodes again have 0 values for $k > \tau_S$. In turn Proposition 4.1 has the following corollary:

Corollary 4.2: Let $k \geq \tau_S$ (respectively $k \geq \tau_C$) and let $v(k)$ be the unique active node at time k under algorithm *simple-walk* (resp. *coalescent*). For such k,

$$\tilde{\mathbf{m}}_t^{v(k)}(k) = \sum_{v \in V} \tilde{\mathbf{m}}_t^v(0).$$

We regard τ_S and τ_C as termination times of respectively *simple-walk* and *coalescent* in the sense that the value of $\sum_{v \in V} \tilde{\mathbf{m}}_t^v(0)$ is known to a single node from there on.

Definition 4.1: *Average time complexity* of algorithm *simple-walk* (resp. *coalescent*) is $E[\tau_S]$ (resp. $E[\tau_C]$).

	COALESCENT	SIMPLE-WALK	GOSSIP
Clique	$\Theta(\log n)$	$\Theta(\log n)$	$\Theta(n)$
Ring	$\Theta(n)$	$\Theta(n)$	$\Theta(n^2)$
Torus ($d = 2$)	$O((\log n)^2)$	$\Theta((\log n)^2)$	$\Theta(n)$
Torus ($d \geq 3$)	$O(\log n)$	$\Theta(\log n)$	$\Theta(n^{2/d})$

TABLE I

MESSAGE COMPLEXITIES FOR THREE DIFFERENT PROTOCOLS AND THREE DIFFERENT TOPOLOGIES FOR AN n NODE NETWORK.

In adopting a measure of messaging complexity, let $\eta_S(k)$ and $\eta_C(k)$ be the total number of transmitted messages in the network by time k under algorithms *simple-walk* and *coalescent* respectively:

Definition 4.2: *Average per-node message complexity* of algorithm *simple-walk* (resp. *coalescent*) refers to $E[\eta_S(\tau_S)]/|V|$ (resp. $E[\eta_C(\tau_C)]/|V|$).

For each network time k let the vector valued variable $\xi_t(k)$ be defined by $\xi_t(k) \triangleq (\xi_t^v(k) : v \in V)$. Note that $(\xi_t(k) : k \geq 0)$ is a time-homogeneous Markov process under both algorithms. More precisely, $(\xi_t(k) : k \geq 0)$ is a random walk on G under *simple-walk*, and a coalescing random walk [13] on G under *coalescent*. In particular the average time complexity of *simple-walk* $E[\tau_S]$ is the cover time of G, which is the expected time that a random walk visits all nodes of G [28]. Since only one node is active under *simple-walk* the mean number of messages transmitted in the whole network is $\lambda E[\tau_S]$ (note that λ is the rate at which an active node transmits); and the average per-node message complexity of the algorithm is $\lambda E[\tau_S]/|V|$. The average message complexity of *coalescent*, $E[\eta_C(\tau_C)]$, is the mean area under the sample path trajectory of the number of active sensors shown in Fig. 2, over the time interval $[0, \tau_C]$.

Asymptotic time and message complexities of *coalescent* and *simple-walk* in specific graph topologies can be found in [2] and are summarized in Table I.

In particular, token based algorithms yield drastic gains in energy relative to the algorithms discussed in the previous sections. Furthermore, token based algorithms compute the exact sum in finite time, whereas belief propagation and gossip can only guarantee a bounded error in finite time. It should also be noted that the latter algorithms compute averages, and small errors in the average may translate to intolerable errors in the sum when the number of sensors is large.

V. COMMUNICATION DELAYS

In this section we relax the assumption that the network is much faster than the system, and consider message passing and measurement on a common time scale. This operational regime has significant implications on tracking performance since communication delays dictate restrictions on the quality of the optimal estimate that can be obtained. We focus on the linear setting with linear system dynamics as well as observation models for simplicity of exposition. With some modifications in message composition it is possible to realize EKFs for non-linear problems.

We determine the optimal estimate subject to the communications delays, which typically differs from sensor to sensor.

We also give a distributed message passing algorithm that computes the optimal estimate at each sensor. The optimal estimate at sensor v when messages from different sensors arrive with delay k is:

$$\mu_t^v = \mathrm{E}[\mathbf{x}_t|\mathbf{y}_t^v,\, \mathbf{y}_{t-1}^v,\, \ldots,\, \mathbf{y}_{t-k+1}^v,\, \mathbf{Y}_{t-k}]$$

Here we focus on a completely connected network topology for simplicity of exposition. These techniques can be extended to arbitrary topologies with a more elaborate message passing scheme. We first consider a network with unit delay, i.e., the $k = 1$ case, since this setting can be handled by a minor modification of the scheme of Section III.

Note that since the network and the system have a common time scale we use the same symbol t to indicate time. We shall denote the message transmitted by sensor v at time t by $\mathbf{m}_t^v$, and define it as an encoding of the optimal prediction of the state at time t based on data available at the originating sensor subject to communication delays. Namely, the message,

$$\mathbf{m}_t^v = (\tilde{\mathbf{m}}_t^v, \tilde{\mathbf{P}}_t^v) \tag{15}$$

where

$$\begin{aligned} \tilde{\mathbf{m}}_t^v &= \mu_t^v - \mathbf{x}_{t|t-1} && (16)\\ \mu_t^v &= \mathrm{E}[\mathbf{x}_t|\mathbf{y}_t^v, \mathbf{Y}_{t-1}] && (17)\\ \tilde{\mathbf{P}}_t^v &= \mathrm{E}[(\mathbf{x}_t - \mu_t^v)(\mathbf{x}_t - \mu_t^v)^{\mathrm{T}}]. \end{aligned}$$

with $\mathbf{Y}_{t-1} = (\mathbf{y}_\tau^v : \tau \le t-1, v \in V)$ as defined earlier.

In a completely connected network this message is simultaneously received by all sensors at time $t + 1$. The same message composition rule is adopted by all sensors, though it is clear that messages differ from sensor to sensor since they are adapted to different filtrations. At time $t + 1$ each sensor v constructs its next message, $\mathbf{m}_{t+1}^v$, based on the received messages $\mathbf{m}_t^u,\ u \ne v$, as well as the local measurement $\mathbf{y}_{t+1}^v$ taken after the last transmitted message. We specify this construction as follows: Note that

$$\mu_{t+1}^v = \mathbf{x}_{t+1|t} + \mathbf{K}_{t+1}^v(\mathbf{y}_{t+1}^v - \mathbf{C}_{t+1}^v\mathbf{x}_{t+1|t}) \tag{18}$$

where $\mathbf{K}_{t+1}^v = \tilde{\mathbf{P}}_{t+1}^v(\mathbf{C}_{t+1}^v)^{\mathrm{T}}(\mathbf{R}^v)^{-1}$ is the Kalman gain matrix. The conditional error covariance $\tilde{\mathbf{P}}_{t+1}^v$ is given by

$$(\tilde{\mathbf{P}}_{t+1}^v)^{-1} = (\mathbf{P}_{t+1|t})^{-1} + (\mathbf{C}_{t+1}^v)^{\mathrm{T}}(\mathbf{R}^v)^{-1}\mathbf{C}_{t+1}^v, \tag{19}$$

where $\mathbf{P}_{t+1|t}$ can be constructed from $\mathbf{P}_{t|t-1}$ and received information $\tilde{\mathbf{P}}_t^u, u \ne v$, via the recursion (4) and the representation

$$\mathbf{P}_{t|t}^{-1} = \sum_{u\in V}(\tilde{\mathbf{P}}_t^u)^{-1} - (V-1)\mathbf{P}_{t|t-1}^{-1} \tag{20}$$

Note that the last term in (19) is local information at sensor v. Manipulation of equality (1), as outlined in Section III, yields that

$$\mathbf{x}_{t|t} = \mathbf{x}_{t|t-1} + \mathbf{P}_{t|t}\sum_{u\in V}(\tilde{\mathbf{P}}_t^u)^{-1}\tilde{\mathbf{m}}_t^u \tag{21}$$

therefore $\mathbf{x}_{t+1|t}$ can be computed based on $\mathbf{x}_{t|t-1}$ and $\mathbf{m}_t^u, u \ne v$ via (4), in turn μ_{t+1}^v is calculated via (18). We collect these observations in the following theorem.

Theorem 5.1: Consider the message passing scheme of Equation (15) and suppose each node upon reception fuses information according to Equation (21) with initial conditions: $\mathbf{x}_{0|0} = \mathbf{x}_0,\ \mathbf{P}_{0|0} = \Sigma_0,\ \tilde{\mathbf{m}}_0^v = 0,\ (\tilde{\mathbf{P}}_0^v)^{-1} = 0,\quad \forall\, v \in V$ then each sensor, v, at any time t has the optimal estimate $\mu_t^v = \mathrm{E}[\mathbf{x}_t|\mathbf{y}_t^v, \mathbf{Y}_{t-1}]$.

Proof: Let us assume that the sensors reached consensus of $(\mathbf{x}_{t-1|t-1}, \mathbf{P}_{t-1|t-1})$ by the time t. Then at time t, each sensor obtains measurement y_t^v and uses it to compute μ_t^v using standard Kalman filter update equation. As Kalman filter produces the optimal estimate for linear systems, we have the equality (17). As we have assumed that the sensors receive the messages $(\tilde{\mathbf{m}}_t^v, \tilde{\mathbf{P}}_t^v)$ from all other sensors transmitted at time t before the time $t + 1$, then each sensor maintains the consensus of $(\mathbf{x}_{t|t}, \mathbf{P}_{t|t})$ at the start of the time $t+1$ by using (18,21). Hence, starting with the same $(\mathbf{x}_{0|0}, \mathbf{P}_{0|0})$ at time 0, by method of induction (17) is true at each timestep t. ■

Note that the pair $(\mathbf{x}_{t|t-1}, \mathbf{P}_{t|t-1})$ is an internal state for the sensor at time t. Namely, $\mathbf{m}_t^v$ is determined by $\mathbf{x}_{t|t-1}, \mathbf{P}_{t|t-1}$ together with the local information $\mathbf{y}_t^v$ and $\mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}\mathbf{C}_t^v$, whereas $(\mathbf{x}_{t+1|t}, \mathbf{P}_{t+1|t})$ is determined by new information obtained at the end of time t.

The algorithm guarantees that each sensor v dynamically constructs μ_t^v, i.e. the optimal estimate of $\mathbf{x}_t$ that can be obtained *subject to the delay constraints* imposed by the network. In general, the error covariances $\tilde{\mathbf{P}}_t^v$ associated with these estimates are different due to non-identical observation models. These covariance matrices are also constructed locally; in turn the network can be queried to identify a sensor with a highest quality estimate.

Theorem 5.1 can be generalized to sensor network system with k-delays by modifying the message passing algorithm. These ideas are based on our related work on asynchronous distributed Kalman filtering [21]. In this setting, if a message is transmitted at time t, its reception is completed by another sensor at time $t+k$. One way to generalize Theorem 5.1 is to transmit message vectors composed of all of the intermediate state estimates from $t + 1$ to $t + k$. However this scheme does not scale well to large delays. An alternative scheme is to compensate for the cross-correlations and transmit a compensated estimate as described next.

First, notice that the global KF estimate can be written in the information form:

$$\begin{aligned} \mathbf{P}_{t|t}^{-1}\mathbf{x}_{t|t} &= \mathbf{P}_{t|t-1}^{-1}\mathbf{x}_{t|t-1} + \sum_{v\in V}\mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}\mathbf{y}_t^v\\ &= \mathbf{P}_{t|t-1}^{-1}A\mathbf{P}_{t-1|t-1}\mathbf{P}_{t-1|t-1}^{-1}\mathbf{x}_{t-1|t-1} + \sum_{v\in V}\mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}\mathbf{y}_t^v \end{aligned}$$

Denoting $\mathbf{s}_t = \mathbf{P}_{t|t}^{-1}\mathbf{x}_{t|t}$ and $\tilde{A}_t = \mathbf{P}_{t|t-1}^{-1}A\mathbf{P}_{t-1|t-1}$ we get:

$$\mathbf{s}_t = \tilde{A}_t\mathbf{s}_{t-1} + \sum_{v\in V}\mathbf{C}_t^{v\mathrm{T}}(\mathbf{R}^v)^{-1}\mathbf{y}_t^v$$

The main observation is that the state evolution of the information, $\mathbf{s}_t$, is linear. The main property of a linear system is that the superposition of the inputs leads to superposition of the corresponding outputs. Consequently, it follows that if each sensor transmits a local compensated estimate their superposition leads to the required global estimate. In particular, suppose that at time t all of the sensors have global estimates $\mathbf{x}_{t-k|t-k}$

then each sensor computes the following compensated estimate (note that this is unrelated to the local optimal KF estimate):

$$\mathbf{s}^v_{t-k+j+1} = \tilde{A}_t \mathbf{s}^v_{t-k+j} + \mathbf{C}^{v\mathrm{T}}_t (\mathbf{R}^v)^{-1} \mathbf{y}^v_{t-k+j+1}; \;\; j = 0, 1, \ldots, k \tag{22}$$

with initial condition $\mathbf{s}^v_{t-k} = \mathbf{P}^{-1}_{t-k|t-k} \mathbf{x}_{t-k|t-k}$

It follows by linear superposition that:

$$\mathbf{s}_t = \sum_{v \in V} \mathbf{s}^v_t = \mathbf{P}^{-1}_{t|t} \mathbf{x}_{t|t} \tag{23}$$

Now each sensor at time t can transmit its local compensated estimate, $\mathbf{s}^v_t$ (or the innovation in $\mathbf{s}^v_t$). This can be locally computed because each sensor has the consensus global estimate for $t-k$ and the update in Equation 22 only requires knowledge of local sensor data. This arrives at all the other sensors at time $t+k$. Now each sensor fuses this information using the above fusion rule. Consequently, a consensus for time t at time $t+k$ at each sensor is achieved. Each sensor, v, can then produce its local optimal estimate by running a local KF to obtain $\mu^v_t = \mathrm{E}[\mathbf{x}_t | \mathbf{y}^v_t, \mathbf{y}^v_{t-1}, \ldots, \mathbf{y}^v_{t-k+1}, \mathbf{Y}_{t-k}]$. We collect this result in the following theorem:

Theorem 5.2: Consider the message update scheme of Equation (22) and suppose the estimate $\mathbf{s}^v_t$ is transmitted by each sensor and arrives at all the other sensors at time $t+k$. Suppose each node upon reception fuses information according to Equation (23). Then at any time t each sensor can compute the optimal estimate $\mu^v_t = \mathrm{E}[\mathbf{x}_t | \mathbf{y}^v_t, \mathbf{Y}_{t-1}]$.

Although this result appears to generalize Theorem 5.1 there is an important difference in terms of sensor knowledge. Note that the result in Theorem 5.1 did not require that each sensor has global knowledge of observation models for other sensors. However, it is clear from Equation (22) that this knowledge is required in realizing the linear dynamical system and hence in establishing Theorem 5.2.

VI. Simulation Results

In this section we obtain numerical results to compare performances of the algorithms outlined in previous sections. In order to illustrate main concepts in a plain setting, we first consider a static situation in which each sensor has one measured value and the goal is to compute the sum of these values. We then return to distributed Kalman filtering wherein the main issue is to obtain sums of dynamically generated local innovations and error covariance terms. In both settings we examine the relationship between the square error and the number of message transmissions in the network.

Numerical simulations of this section are conducted on random geometric graphs that have received significant attention in wireless ad-hoc networks. Namely, we randomly generate a sensor network as follows: A total of N nodes are distributed uniformly on the unit square, and each node maintains a communication link with those nodes that are within $\sqrt{\frac{2\log_2(N)}{N}}$ distance.

A. Distributed computation of sum

We first consider a scenario where each node is initialized with a scalar value of zero except for a randomly chosen node which is initialized with N. Such an initialization is of interest for two main reasons. Firstly, in most applications, the spatial distribution of information quality (i.e. SNR) is not homogeneous across the whole network. For example, in target tracking usually the nearby sensors have better observation compared to the distant ones. The scheme we have chosen can be used to characterize the extreme case where only a single sensor has a large SNR compared to all other sensors in the network. Secondly, this initial configuration can be considered as a worst case in terms of the square error in the network.

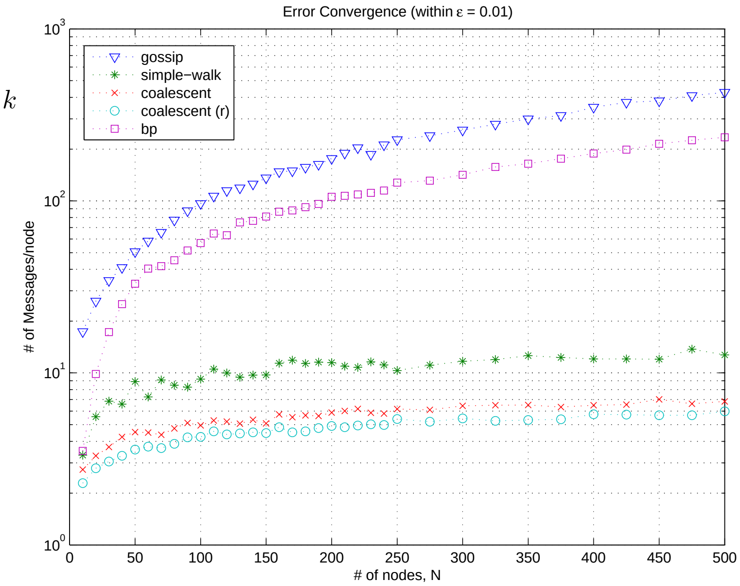


Fig. 3. Average number of messages transmitted per node needed to achieve $\mathcal{E}(\tau) \leq \epsilon = 0.01$ for *gossip*, *simple-walk*, *coalescent* and *belief propagation* algorithms. The curve for *coalescent(r)* shows the performance of *coalescent* when the nodes retains the best estimate using a weight variable that keeps track of the number of contributors to the current value.

Let the value of node v at the start of the algorithm be $\mathbf{m}^v(0)$. We are interested in finding the sum, $\sum_{v \in V} \mathbf{m}^v(0)$ without using a fusion center. Under each algorithm of interest let $\mathbf{m}^v(\tau)$ denote the local estimate of sensor v at time τ. We define the square error of the sensor network at time τ as $\mathcal{E}(\tau) \triangleq \sum_{v=1}^{N} (\mathbf{m}^v(\tau) - N)^2$.

In order to clearly distinguish between the time and the number of message required to reach consensus we consider a continuous time model. Namely, each node has a Poisson clock that ticks at rate $\lambda > 0$. In simulations we choose the Poisson clock rate suitably for each algorithm to make fair comparisons concerning the total number of messages exchanged. If, for example, λ is taken to be unity in all the algorithms, under *simple-walk* the nodes transmit only 1 message on average per unit time (only the active node can transmit and in *simple-walk* at any given time, only one node is active). On the other hand, under *gossip* nodes exchange $2N$ messages per unit time (at a tick of a local Poisson clock, two nodes exchange messages between each other).

Figure 3 shows the required number of messages to get the network error $\mathcal{E}(\tau)$ below $\epsilon = 0.01$ for different al-

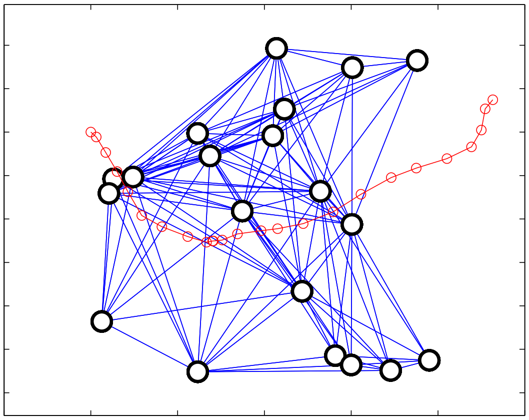


Fig. 4. A typical target trajectory used in simulation. The target is maneuvering through the sensor field of uniformly distributed 20 sensors.

gorithms. The results for *belief propagation* are obtained via the *Metropolis* weights as described in Section IV-A. In algorithms *simple-walk* and *coalescent* the estimate of an idle node is taken to be its value when it was last active. As the figure illustrates, token based algorithms need significantly fewer messages to reach the same consensus condition relative to *belief propagation* and *gossip*. Performance of the two token based algorithms can be improved to some extent by *coalescent(r)* in which the estimate of an idle node is the prior value which was contributed by the largest number of sensors.

B. Target tracking

Next we apply the algorithms in distributed Kalman filtering application. Consider a target, maneuvering through a sensor field whose state space is given by $\mathbf{x}_t = (p_x, v_x, p_y, v_y)^{\mathrm{T}}$ where p_x, p_y and v_x, v_y are the x, y position and velocity resp. and its evolution is given by:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{w}_t$$

where $\mathbf{w}_t \sim \mathcal{N}(0, \mathbf{Q})$ is the process noise independent of $\mathbf{x}_t$ and,

$$\mathbf{A} = \begin{pmatrix} 1 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \delta t \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{Q} = q \begin{pmatrix} \frac{\delta t^3}{3} & \frac{\delta t^2}{2} & 0 & 0 \\ \frac{\delta t^2}{2} & \delta t & 0 & 0 \\ 0 & 0 & \frac{\delta t^3}{3} & \frac{\delta t^2}{2} \\ 0 & 0 & \frac{\delta t^2}{2} & \delta t \end{pmatrix}$$

where q is the covariance scale. At each discrete time step δt, the sensor v takes a noisy measurement of the target's position,

$$y_t^v = \mathbf{C}\mathbf{x}_t + \mathbf{u}_t^v, \quad \mathbf{u}_t^v \sim \mathcal{N}(0, \mathbf{R}^v), \quad \mathbf{C} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Noise covariance matrices, $\mathbf{R}^v$, are chosen so as to make individual measurements highly noisy. For this simulation, we have used $\delta t = 1$, $q = 0.01$ and $\mathbf{R}^v = 10.0\mathbf{I}$. A typical target trajectory is shown in Figure 4. We compare the performances of *gossip*, *belief propagation*, *simple-walk* and *coalescent* algorithms under these settings.

Figure 5 shows a typical sample path in the (system-) time interval $t = 4$ to $t = 8$. As explained in earlier

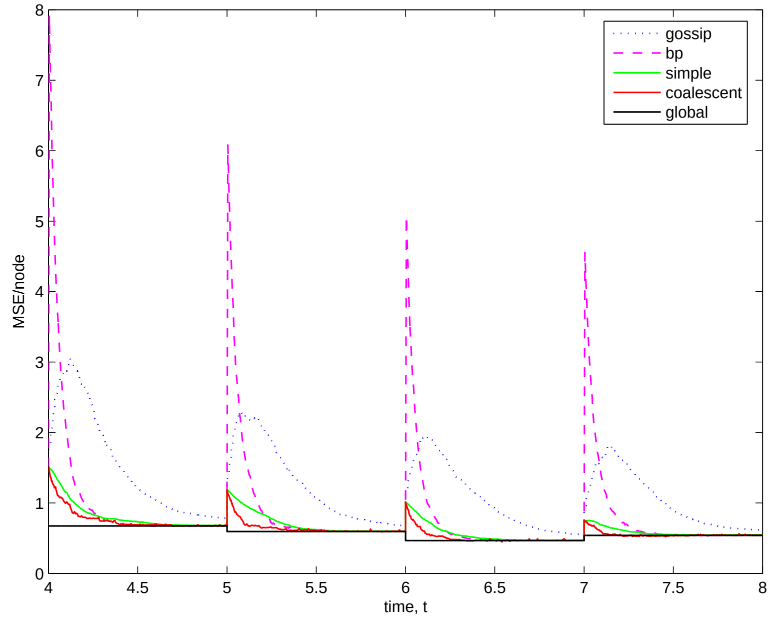
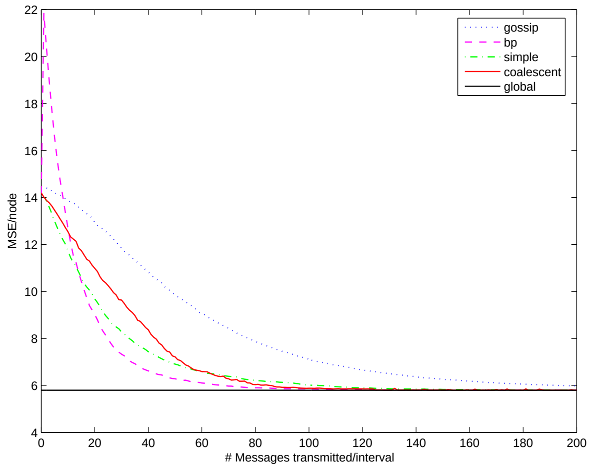


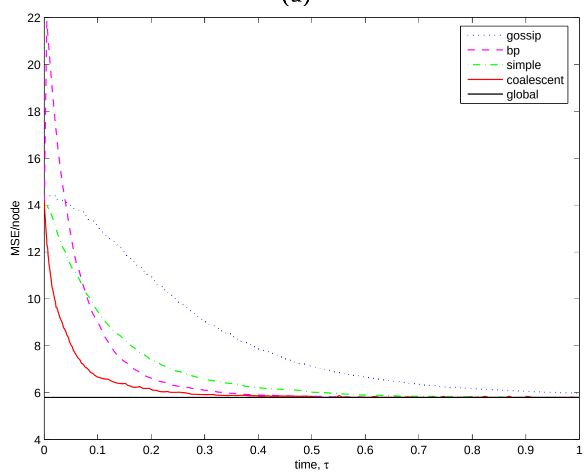
Fig. 5. A typical sample path of error convergence for *gossip*, *simple-walk* and *coalescent* algorithms. The network has 20 nodes with 118 edges. We have used different Poisson clock rates for different algorithms so as to achieve the same message count on average. The solid line indicates the instantaneous (rather than mean) square error of the global optimal estimate, which is constant between measurements.

sections, at the start of each interval, every node uses its local measurement to update the local estimates. Then until the start of the next interval, the nodes run the consensus algorithm. In the simulated setting the network is fast compared to target dynamics; hence the nodes exchange multiple messages within each interval.

Figure 6 shows the square error $\mathcal{E}(\tau)$ (with N representing the sought sum for each interval) per node in a network of 20 sensors. We have recorded the per-node error $\mathcal{E}(\tau)/20$ after each message exchange for every interval and have taken the average over the entire simulation period starting from $t = 6$ leaving out the initial transient. For *belief propagation*, we have implemented an asynchronous-broadcast version of the algorithm where at each time a randomly chosen node broadcasts its value to all of its neighbors, which in turn update their own values. We have also employed the pilot signal technique to determine the right estimates as described in Section IV-B. Note the initial error increase of *gossip* and *belief propagation* is a result of using average to obtain the sum, as this amounts to multiplying the estimation error for the average by the number of sensor nodes. We notice no such behavior in *coalescent* and *simple-walk* where we can compute the sum directly. The aforementioned features have direct implications for the slow network setting. In a dynamic setting when the network is slow relative to target dynamics it is important to optimally aggregate information only from nodes that can be reached in a short period of time. The behavior of the coalescent walk suggests that mean-square error is not only monotonically decreasing but also is



(a)



(b)

Fig. 6. Convergence to global estimate in the target tracking application. The target is tracked over 25 intervals using different algorithms keeping the messaging rate/node fixed. The simulations suggest that *belief propagation* takes the least number of messages (perhaps due to counting each broadcast message as a single message) for achieving similar square error across all the nodes, while *coalescent* takes the least time to achieve the same goal, and *gossip* has the poorest performance in both measures. Also note that *simple-walk* and *coalescent* reaches exact global estimate whereas no such claim can be made for *gossip* or *belief propagation*.

significantly smaller than BP/gossip in the short term. This means that if the target dynamics changes faster than the time required to aggregate information from all of the sensors, coalescent will only aggregate information from nodes that can be reached within any time-period. In other words, while gossip algorithms may require to wait until aggregating all of data from the whole network to produce accurate estimates, token based approaches can provide useful estimates even in the short term since its "aggregation diameter" scales with the available time.

The simulations suggest that *belief propagation* takes the least number of messages for achieving similar square error across all the nodes, while *coalescent* takes the least time to achieve the same goal, and *gossip* has the poorest performance in both measures. We finally note that *gossip* and *belief propagation* have similar order-wise complexity, so the apparent complexity gain of *belief propagation* may be attributed to its broadcast nature (note that the nodal degrees in Figure 4 are fairly large, so broadcasting has a significant impact on efficiency). This suggest further improvements in *coalescent* due to incorporation of broadcast and this direction is the subject of future research.

VII. Conclusions

We have presented methods for distributed tracking of a target via networked sensors. The networked sensors communicate with each other by means of a multi-hop protocol over a communication network. We presented optimal energy efficient in-network processing algorithms to deal with arbitrary network topology and non-ideal channel conditions where packet transmissions are subject to delays and losses. The techniques presented differ from existing techniques in two important aspects: a) there is no designated leader/fusion node and each sensor attempts to optimally track the system trajectory based on its local observations and time-dependent information available from other sensors in the network; b) the message computation at each sensor is structurally identical, where the transmitted message is related to the state innovation conditioned on the information available. We then present two multihop protocols—one based on Gossip and another token-based—for distributed implementation of the in-network processing techniques. We show that the latter token-based approach leads to significant energy savings, i.e., message complexity for achieving a given tracking error performance is significantly smaller than the conventional approaches.

References

[1] M. Alanyali and V. Saligrama. "Distributed Target Tracking over Multi-hop Networks," *IEEE Statistical Signal Processing Workshop,* Bordeaux, France, June 2005

[2] M. Alanyali, V. Saligrama, O. Savas. A Random-Walk Model for Distributed Computation in Energy-Limited Networks. Inaugural Workshop for the Center for Information Theory and its Applications, University of California San Diego, February 2006.

[3] B.D. Anderson and J.B. Moore, "Optimal Filtering," *Prentice- Hall, New Jersey*, 1979.

[4] Y. Bar-Shalom. "On the Track-to-Track Correlation Problem," *IEEE Trans. Automatic Control,* AC-26:571-572, April 1981

[5] Y. Bar-Shalom and L. Campo, "The effects of the common process noise on the two-sensor fused-track covariance," *IEEE Trans. on Aero. and Elect. Syst.,* vol. AES-22, no. 6, pp. 803-804, 1986.

[6] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation, Numerical Methods*, Prentice-Hall, Inc, 1989.

[7] M. Bolic and P. M. Djuric and S. Hong, Resampling algorithms and architectures for distributed particle filters, *IEEE Trans. Signal Proc.*, 2005.

[8] S. Boyd, A. Ghosh, B. Prabhakar and D. Shah, "Gossip algorithms: Design, analysis and applications," IEEE INFOCOM 2005, 2005.

[9] C. Y. Chong and S. P. Kumar. "Sensor Networks: Evolution, Opportunities, and Challenges," *Proceedings of the IEEE,* VOL. 91, NO. 8, AUGUST 2003

[10] T. H. Chung, V. Gupta, J. W. Burdick, R. M. Murray, "On a Decentralized Active Sensing Strategy using Mobile Sensor Platforms in a Network," Proceedings of the Control and Decision Conference, Paradise Islands, Bahamas, Dec 2004.

[11] M.J. Coates. "Distributed particle filtering for sensor networks," *in Proc. Int. Symp. IPSN, Berkeley,* CA, Apr. 2004.

[12] M. J. Coates and G. Ing. "Sensor Network Particle Filters: Motes as particles," *in IEEE Statistical Signal Processing Workshop,* Bordeaux, France, June 2005.

[13] J.T. Cox. Coalescing random walks and voter model consensus times on the torus. *The Annals of Probability*, 17(4):1333-1366, 1989.
[14] A. Doucet and N. de Freitas and N. Gordon, “Sequential Monte Carlo Methods,” *Springer-Verlag, New York,* 2001.
[15] R.A Horn and C.R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.
[16] S. J. Julier and J. K. Uhlmann. “A New Extension of the Kalman Filter to Nonlinear Systems,” *SPIE*, 1997
[17] F. Martinerie. “Data fusion and tracking using HMMs in a distributed sensor network,” *IEEE Trans. Aerospace and Electronic Systems*, vol. 33, pp. 1128, Jan. 1997.
[18] S. Mori, W.H. Barker, C.Y. Chong and K.C. Chang, ”Track association and track fusion with non-deterministic target dynamics,” *Proceedings of FUSION99*, 1999.
[19] R. Olfati-Saber. “Distributed Kalman Filter with Embedded Consensus Filters,” *44th IEEE Conference on Decision and Control*, Dec. 2005.
[20] B. S. Y. Rao and H. F. Durrant-Whyte and J. A. Sheen. A fully decentralized multi-sensor system for tracking and surveillance. *Int. Journal of Robotics Research*, 12(1):2044, Feb 1993.
[21] V. Saligrama and D. Castanon. “Reliable Tracking with Unrealiable Communications,” ICASSP, Toulouse, France, 2006
[22] X. Sheng and Y-H. Hu. “Distributed particle filter withGMM approximation for multiple target localization and tracking in wireless sensor network,” *in Proc. IEEE/ACM Int. Symp. IPSN*, Los Angeles, CA, Apr. 2005.
[23] D. Spanos, R. Olfati-Saber, R. Murray, “Distributed Kalman Filtering in Sensor Networks with Quantifiable Performance,” In *Proceedings of Information Processing in Sensor Networks*, April 2005
[24] J. Speyer, “Computation and transmission requirements for a decentralized linear-quadratic-Gaussian control problem,” *IEEE Transactions on Automatic Control*, vol. 24, no. 2, pp. 266–269, 1979.
[25] A. Willsky, M. G. Bello, D. A. Castanon, B. C. Levy, G. C. Verghese, “Combining and Updating of Local Estimates and Regional Maps Along Sets of One Dimensional Tracks,” *IEEE Transactions on Automatic Control*, vol. 24, no. 4, pp. 799–813, 1982.
[26] J. Wolfowitz, “Products of indecomposable, aperiodic, stochastic matrices,” *Proceedings of the American Mathematical Society*, vol. 14, pp. 733–737, 1963.
[27] L. Xiao and S. Boyd. “Fast linear iterations for distributed averaging.” *Systems & Control Letters*, 52:6578, 2004.
[28] D. Zuckerman, A technique for lower bounding the cover time, *SIAM Journal on Discrete Mathematics*, no. 5, pp. 81-87, 1992.

PLACE PHOTO HERE

Rashedur Rahman received the B.Sc. and the M.Sc. degree in Applied Physics and Electronics from the University of Dhaka, Dhaka, Bangladesh in 1999 and 2000 respectively. He then recieved the MS degree in Electrical Engineering from Boston University in 2006. He is currently working at the Microsoft Corporation in Redmond, WA. His research interests include Distributed/Network signal processing, Bayesian estimation and tracking and sensor network applications.

PLACE PHOTO HERE

Murat Alanyali is an Assistant Professor at Boston University Department of Electrical and Computer Engineering. He is a member of the Information Systems and Sciences (ISS) Laboratory and the Center for Information Systems and Engineering (CISE). Professor Alanyali’s research interests include communication networks, performance analysis and optimization, and stochastic systems.

Professor Alanyali received his Ph.D. in 1996 from the University of Illinois, Urbana-Champaign and an M.S. in 1990 from Bilkent University, Ankara, Turkey. Previously he was Assistant Professor of Electrical and Electronics Engineering, Bilkent University, Ankara, Turkey, and from 1996-1998, a consultant for Bell Laboratories, Lucent Technologies, Holmdel, New Jersey. Professor Alanyali received an NSF CAREER Award in 2003. He is an IEEE Associate Editor, and a member of the Conference Editorial Board, IEEE Control Systems Society.

PLACE PHOTO HERE

Venkatesh Saligrama received a Ph.D degree from MIT in 1997. He was a Research Scientist at the United Technologies Research Center, Hartford,CT between 1997-2001 and was Visiting Scientist at MIT between 2000-2001. He is currently an Associate Professor in the Department of Electrical and Computer Engineering at Boston University. He has received numerous awards including the Presidential Early Career Award, Office of Naval Research Young Investigator Award, NSF Career Award, and the Outstanding Achievement Award from United Technologies. His team was a finalist in the 2005 Crossbow Sensor Network Challenge Competition. He is currently serving as an Associate Editor for IEEE Transactions on Signal Processing. His research interests are in Information and Control theory, Network Signal Processing with applications to Sensor Networks.