Distributed Tracking with Consensus on Noisy Time-varying Graphs: Convergence Results and Applications

Sudharman K. Jayaweera and Y. Ruan
ECE Deaprtment
University of New Mexico
Albuquerque, NM, USA
Email: jayaweera@ece.unm.edu

R. Scott Erwin
Air Force Research Laboratory
Space Vehicles Directorate
Kirtland AFB, NM 87117

Jed Carty
ECE Deaprtment
University of New Mexico
Albuquerque, NM, USA

***Abstract*—In this paper we consider the problem of distributed tracking with consensus on a time-varying graph with noisy communications links and sensing constraints. We develop a framework to handle the time-varying network topology in which not every node has local observations to generate own local tracking estimates. Our approach introduces the concepts of active node set and connectivity graph to characterize such a network, and by merging these two, an effective network graph is obtained. Then we propose a distributed tracking-with-consensus algorithm for such a network model. We establish the conditions on the connectivity graph so that distributed consensus can be achieved in the presence of noisy communication links when the effective network graph is time-varying. We also discuss how this problem is motivated by the problem of distributed tracking of space-borne Objects of Interests (OoI's) in a hybrid space surveillance network (SSN) formed by both ground and satellite nodes. Simulation results of the proposed distributed tracking with consensus algorithm are given for a two-dimensional hybrid sensor network. They show that our algorithm performs almost the same as the distributed local Kalman filtering with centralized fusion on a noisy time-varying graphs with incomplete data, while the proposed algorithm has the additional advantages of flexibility and scalability.**

I. Introduction

Distributed tracking-with-consensus, addressed in this paper, refers to the problem that a group of nodes need to achieve an agreement over the state of a dynamical system by exchanging tracking estimates over a network. Note that, our distributed tracking-with-consensus problem is different from the *consensus-in-tracking* problem found in literature: The latter refers to the problem in which at each tracking step a consensus exchange of local tracking estimates is performed and the consensus is aimed to be achieved asymptotically in tracking time steps [1]. The former, on the other hand, refers to the problem in which at each tracking step there is an asymptotically large number of consensus iterations among nodes in the network with the aim of achieving consensus on distributed tracking estimates at each tracking time step. This is the problem considered in this paper, and later we will discuss how this problem arises in the important context of *space situational awareness* (SSA) when distributed tracking of space-borne Objects of Interests (OoI's) using a hybrid *space surveillance network* (SSN) is of interest [2], [3].

In this paper, we formulate the problem of distributed tracking with consensus on a time-varying graph with incomplete data and noisy communication links, where the time-varying graph consists of fixed nodes that are connected together and mobile nodes that could only have active links with other nodes within their communication radius. Next, we propose an algorithm by combining distributed Kalman filtering with consensus in estimation [4], [5] to handle a time-varying network in which not every node has local observations to generate own local tracking estimates (incomplete data). We then establish the conditions on the so-called connectivity graph of the sensor network so that the distributed consensus can be achieved even in the presence of noisy communication links when the graph topology is time-varying. We detail an application scenario from SSA in which the considered problem of tracking-with-consensus naturally arise. We provide simulation examples to show the convergence performance of the proposed distributed tracking with consensus algorithm compared to that with the distributed local Kalman filtering with centralized fusion and centralized Kalman filter.

The remainder of this paper is organized as follows: Section II introduces our assumed system and network models, and Section III presents the proposed distributed tracking with consensus algorithm. In Section IV conditions for achieving distributed consensus are discussed and the rate of convergence is quantified. In Section V we describe how the problem addressed here arises in the context of SSA and the use of the the proposed algorithm in that context. The simulation results and performance comparisons are given in Section VI. Finally, Section VII concludes the paper.

II. Problem Formulation

A. System Model

Consider an N-node sensor network with a *connectivity graph* $G(k) = \big(V, E(k)\big)$ at time k made of both mobile and fixed nodes. Assume that the graph $G(k)$ is undirected, but time varying due to nodes moving in and out of communication ranges of each other (or terminating transmissions to save battery power). There is a set of multiple, independent targets. A particular target's dynamics evolves according to

$$\mathbf{x}(k+1) = F\mathbf{x}(k) + \mathbf{w}(k); \quad \mathbf{x}(0) \sim \mathcal{N}\big(\bar{\mathbf{x}}(0), P_0\big). \tag{1}$$

where $\mathbf{w}(k)$ is zero-mean white Gaussian process noise (WGN), $\mathbf{x}(0) \in \mathbb{R}^M$ is the initial state of the target, and $\mathbb{E}\big[\mathbf{w}(k)\mathbf{w}(k')^T\big] = Q\delta_{kk'}$ with $\delta_{ii'} = 1$ if $i = i'$ and $\delta_{ii'} = 0$, otherwise.

The problem facing the sensor network is to track these targets. However, due to relative motion of nodes and targets etc not every node can observe a given target at a given time. Hence, nodes perform local distributed Kalman filtering based on their own local observations. This, of course, leads to a set of possibly different distributed estimates corresponding to the same target state, which is undesirable in practice. To counter this, the nodes exchange distributed tracking estimates over noisy communication links and try to reach consensus over the network. Since targets are independent, in the following we limit our discussion to tracking a single target.

We denote the discrete time index by k. Due to sensing constraints, the observations are only taken at each $k = iJ$, for $i = 0, 1, \cdots$ and $J > 0$. The sensing model of the n-th sensor, for $k = iJ$ and $i = 0, 1, \cdots$, is

$$\mathbf{y}_n(i) = H_n\mathbf{x}(k) + \mathbf{v}_n(i), \mathbf{y}_n \in \mathbb{R}^L, \tag{2}$$

where $\mathbf{v}_n(i)$ is zero-mean WGN and $\mathbb{E}\big[\mathbf{v}_n(i)\mathbf{v}_{n'}(i')^T\big] = R_n\delta_{ii'}\delta_{nn'}$. Note that, the observation matrices H_n's can be different for each node.

Tracking updates are performed at $k = iJ$ instances, and i denotes the tracking time step ($i = 0, 1, \cdots$). As shown in Fig. 2, at the end of the i-th tracking update, each node n who had an observation of the target will have a filtered estimate $\hat{\mathbf{x}}_n(i|i)$ with an associated covariance matrix $\hat{P}_n(i|i)$.

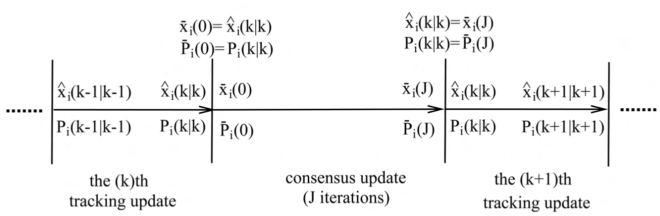


Fig. 1: Timing diagram of tracking and consensus updates in the proposed algorithm for distributed tracking with consensus.

In order to improve the tracking estimate accuracy and, more importantly, to reach at a consensus estimate among all nodes, the nodes then perform a J number of consensus exchanges of their local tracking estimates between any two tracking steps. i.e. for each $i > 0$, consensus updates are performed at $k = iJ + j$ where $0 \leq j < J$ denotes the consensus iteration number as shown in Fig. 5. The exchanges starts by sharing the filtered estimates with neighbors over noisy communication links. Note that, the goal here is to reach at a consensus estimate over the distributed local estimates at

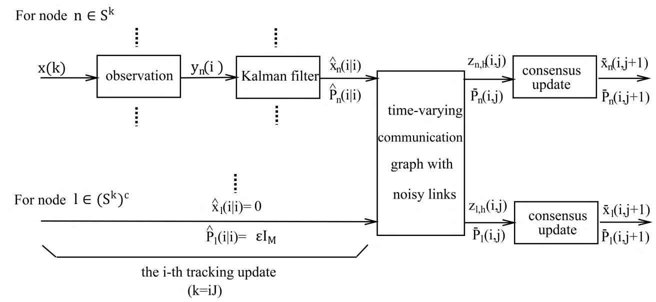


Fig. 2: Block diagram of distributed tracking with consensus on a time-varying graph with incomplete data and noisy communication links.

each tracking time step i. Hence, the consensus problem is essentially a problem of *consensus in estimation*.

Let $\bar{\mathbf{x}}_n(i, j)$ denote the node n's updated tracking estimate at the j-th consensus iteration that follows the i-th tracking update step with $\bar{\mathbf{x}}_n(i, 0) = \hat{\mathbf{x}}_n(i|i)$, where $\hat{\mathbf{x}}_n(i|i)$ is the n-th node's filtered tracking estimate in the i-th tracking update. The received data at node n from node l, for $n \neq l$, at iteration j is written as $\mathbf{z}_{n,l}(i, j) = \bar{\mathbf{x}}_l(i, j) + \boldsymbol{\phi}_{n,l}(j)$, for $0 \leq j < J$, where $\boldsymbol{\phi}_{n,l}(j)$ denotes the communications noise at the node n in receiving the estimator of node l at iteration j. We assume that $\mathbb{E}\big[\boldsymbol{\phi}_{n,l}(j)\big] = \mathbf{0}_M$, $\mathbb{E}\big[\boldsymbol{\phi}_{n,l}(j)\boldsymbol{\phi}_{n,l}^T(j)\big] = \Sigma_{n,l}$ and $\mathbf{z}_{n,n}(i, j) = \bar{\mathbf{x}}_n(i, j)$.

B. Network Model

As mentioned above, the sensor network of interest consists of both mobile and fixed nodes. Due to time-varying topology of the network, at any given tracking time step i some nodes may not be able to observe the target, and thus will not have filtered local tracking estimates. We define the *active node set* S^k in a time-varying graph $G(k)$ as the set of nodes that have updated local estimates to be shared with others in the j-th consensus iteration after the i-th tracking update [3], where $k = iJ + j$, $i = 0, 1, \cdots$ and $0 \leq j < J$. Define *effective network graph* of a network $G(k) = \big(V(k), E(k)\big)$ with active node set S^k as $\tilde{G}(k) = \big(V(k), \tilde{E}(k)\big)$, where $\tilde{E}(k) = E(k) \cap \big(\cup_{n \in S^k} \Upsilon_n^{\text{out}}(k)\big)$ and $\Upsilon_n^{\text{out}}(k) = \{\overrightarrow{(n,l)} | \overrightarrow{(n,l)} \in E(k)\}$ denotes the set of directed edges with initial vertex as n at time k. The effective network graph $\tilde{G}(k)$ is a *directed graph*, which is obtained by removing the outgoing edges of the nodes that do not have data in $G(k)$. For a static graph $G(k) = G(V, E)$, $\tilde{E}(k)$ can be written as $\tilde{E}(k) = \tilde{E}(k-1) \cup_{l \in S^{k-1}} \big(\cup_{n \in \Omega_l} \Upsilon_n^{\text{out}}\big)$, where $\tilde{E}(0) = E \cap \big(\cup_{n \in S^0} \Upsilon_n^{\text{out}}\big)$. The nodes that do not observe the target will not have updated local estimates to share at the beginning of consensus update process (at $j = 0$). However, as information exchange among nodes progresses, some of these nodes may be able to form their own updated local estimates at a consensus iteration j for $j > 0$. Therefore, the active node set S^k is time-varying and $S^k = S^{k-1} \cup_{l \in S^{k-1}} \Omega_l(k-1)$, where S^0 is the set of nodes that have observations of the target in the 0-th tracking update step. Figure 3 shows the relation between the connectivity graph $G(k)$ and the effective network graph $\tilde{G}(k)$ for a graph of 6 nodes with active node set $S^k = (1, 2, 4, 6)$, where solid circles denote active nodes.

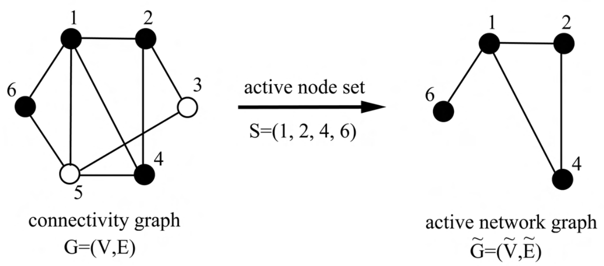


Fig. 3: Connectivity graph and effective network graph.

Let I_{S^k} denote an $N \times N$ matrix generated from the active node set S^k as follows:

$$[I_{S^k}]_{nn'} = \begin{cases} 1 & \text{if } n = n' \quad \text{and} \quad n' \in S^k \\ 0 & \text{else} \end{cases} .$$

Note that, I_{S^k} is a diagonal matrix with n'-th diagonal element equal to zero for $n' \in (S^k)^c$, where $(\cdot)^c$ denotes the set complement. By combining the connectivity graph $G(k)$ with the active node set S^k, we obtain the effective network graph $\tilde{G}(k)$ for $k \geq 0$. Thus, the adjacency matrix [6] of the effective network graph is given by $\mathbb{A}(k) = A(k)I_{S^k}$, where $A(k) = [A_{nl}(k)]$ is the adjacency matrix of the graph $G(k)$ at time k. Note that $A_{nl}(k) = 1$, if $(n,l) \in E(k)$, and $A_{nl}(k) = 0$ otherwise. The corresponding degree matrix $\mathtt{D}(k)$ can then be obtained from $\mathbb{A}(k)$, and the Laplacian matrix becomes $\mathtt{L}(k) = \mathtt{D}(k) - \mathbb{A}(k)$.

III. Proposed Distributed Tracking with Consensus Algorithm

Based on the architecture that was first proposed in [3] in the context of consensus tracking in a satellite sensor network for situational awareness, in the following we propose a distributed tracking and consensus algorithm for the above distributed tracking problem over a time-varying graph with incomplete data and noisy communication links. As in Fig. 5, at tracking time step i, node n is assumed to have completed its consensus iterations corresponding to time $i-1$. If the output of this consensus update following the $(i-1)$-th tracking update step is $\overline{\mathbf{x}}_n(i-1,J)$ with the associated covariance matrix $\overline{P}_n(i-1,J)$, then node n sets $\overline{\mathbf{x}}_n(i-1|i-1) = \overline{\mathbf{x}}_n(i-1,J)$ and $\overline{P}_n(i-1|i-1) = \overline{P}_n(i-1,J)$. Next, at the i-th tracking update step, each node n where $n \in S^k$ for $k = iJ$, passes its observation $\mathbf{y}_n(i)$ through its local Kalman filter as follows [7]:

$$\begin{aligned}
\hat{\mathbf{x}}_n(i|i-1) &= F\overline{\mathbf{x}}_n(i-1|i-1), \\
\hat{P}_n(i|i-1) &= F\overline{P}_n(i-1|i-1)F^T + Q, \\
K_n(i) &= \hat{P}_n(i|i-1)H_n^T\left(H_n\hat{P}_n(i|i-1)H_n^T + R_n\right)^{-1}, \\
\hat{\mathbf{x}}_n(i|i) &= \hat{\mathbf{x}}_n(i|i-1) + K_n(i)\Big(\mathbf{y}_n(i) - H_n\hat{\mathbf{x}}_n(i|i-1)\Big), \\
\hat{P}_n(i|i) &= \Big(I - K_n(i)H_n\Big)\hat{P}_n(i|i-1), \qquad (3)
\end{aligned}$$

where $\overline{\mathbf{x}}_n(i-1|i-1) = \overline{\mathbf{x}}_n(i-1,J)$ with $\overline{\mathbf{x}}_n(-1,J) = \overline{\mathbf{x}}(0)$ and $\overline{P}_n(i-1|i-1) = \overline{P}_n(i-1,J)$ with $\overline{P}_n(-1,J) = P_0$. Let $\overline{\mathbf{X}}(i-1,j) = \left[\overline{\mathbf{x}}_1(i-1,j)^T, \overline{\mathbf{x}}_2(i-1,j)^T, \cdots, \overline{\mathbf{x}}_N(i-1,j)^T\right]^T$. Denote $\overline{P}(i-1,j)$ as the covariance matrix corresponding to $\overline{\mathbf{X}}(i-1,j)$. Note that, $\overline{P}_n(i-1,J)$ in (3) can be obtained by extracting the n-th $M \times M$ main diagonal block of $\overline{P}(i-1,J)$.

Node n uses its filtered estimate $\hat{\mathbf{x}}_n(i|i)$ obtained by the above tracking update step as the initial estimate for consensus update exchanges by setting $\overline{\mathbf{x}}_n(i,0) = \hat{\mathbf{x}}_n(i|i)$ with initial covariance matrix $\overline{P}(i,0) = \hat{P}_1(i|i) \oplus \hat{P}_2(i|i) \oplus \cdots \oplus \hat{P}_N(i|i)$, where $\oplus$ denotes the direct sum. On the other hand, for nodes $n \in (S^k)^c$, for $k = iJ$, we may arbitrarily set $\hat{\mathbf{x}}_n(i|i) = \mathbf{0}$ and $\hat{P}_n(i|i) = \epsilon I_M$ for some $\epsilon > 0$.

During the $(j+1)$-th consensus update, each node n forms a linear estimate of the following form as its consensus estimate where $k = iJ + j$:

$$\begin{aligned}
&\overline{\mathbf{x}}_n(i,j+1) \\
=&\overline{\mathbf{x}}_n(i,j) + \gamma_n(j)\sum_{l=1}^{N}\mathbb{A}_{n,l}(k)\Big(\overline{\mathbf{z}}_{n,l}(i,j) - \overline{\mathbf{z}}_{n,n}(i,j)\Big), \qquad (4)
\end{aligned}$$

where $\gamma_n(j)$ is the n-th node's weight coefficient at iteration j and $0 \leq j < J$. For $k = iJ + j$, we set $\gamma_n(j) = \gamma(j)$ for $n \in S^k$ and $\gamma_n(j) = \frac{1}{\sum_{l=1}^{N}\mathbb{A}_{n,l}(k)}$ for $n \in (S^k)^c$ and $\sum_{l=1}^{N}\mathbb{A}_{n,l}(k) \neq 0$, so that the consensus update dynamics can be written in vector form as follows for $k = iJ + j$:

$$\overline{\mathbf{X}}(i,j+1) = \overline{\mathbf{X}}(i,j) - \Big[\big(\Gamma(j)\mathtt{L}(k)\big) \otimes I_M\Big]\overline{\mathbf{X}}(i,j) - \big(\Gamma(j) \otimes I_M\big)\overline{\Phi}(j), \qquad (5)$$

where $\Gamma(j) = \text{diag}\big(\gamma_1(j), \cdots, \gamma_N(j)\big)$, $\overline{\Phi}(j) = \big[\boldsymbol{\phi}_1(j)^T \cdots \boldsymbol{\phi}_N(j)^T\big]^T$ and $\boldsymbol{\phi}_n(j) = -\sum_{l=1}^{N}\mathbb{A}_{n,l}(k)\boldsymbol{\phi}_{n,l}(j)$.

Let us define $\overline{\mathbf{e}}(i,j)$ to be the error vector at the j-th consensus iteration after the i-th tracking update: $\overline{\mathbf{e}}(i,j) \triangleq \overline{\mathbf{X}}(i,j) - (\mathbf{1} \otimes I_M)\mathbf{x}(iJ)$. From (5), it follows that

$$\begin{aligned}
\overline{\mathbf{e}}(i,j+1) &= \big(\mathbf{A}(j) \otimes I_M\big)\overline{\mathbf{e}}(i,j) - \big(\Gamma(j) \otimes I_M\big)\overline{\Phi}(j) \\
&\quad + \Big(\big(\mathbf{A}(j) \otimes I_M\big) - I\Big)\big(\mathbf{1} \otimes I_M\big)\mathbf{x}(iJ), \qquad (6)
\end{aligned}$$

where $\mathbf{A}(j) = I_N - \Gamma(j)\mathtt{L}(k)$ is a coefficient matrix and $k = iJ + j$.

Note that, if the filtered estimate $\hat{\mathbf{x}}_n(i|i)$ at the end of the measurement update stage is an unbiased estimate, then $\overline{\mathbf{x}}_n(i,0)$ is also unbiased for all $n \in S^k$ and $k = iJ$. From (4), since $\overline{\mathbf{x}}_n(i,j+1) = \frac{1}{\sum_{l=1}^{N}\mathbb{A}_{n,l}(k)}\sum_{l=1}^{N}\mathbb{A}_{n,l}(k)\big(\overline{\mathbf{x}}_l(i,j) + \boldsymbol{\phi}_{n,l}(j)\big)$ for $n \in (S^k)^c$, then $\overline{\mathbf{x}}_n(i,j+1)$ is also unbiased for $n \in (S^k)^c$ if $\overline{\mathbf{x}}_l(i,j)$ is unbiased for $l \in S^k$. From (6), it can be shown that the unbiasedness in the consensus estimator $\overline{\mathbf{X}}(i,j)$ can be maintained if matrix $\mathbf{A}(j)$ satisfies the condition $\Big(\big(\mathbf{A}(j) \otimes I_M\big) - I\Big)\big(\mathbf{1} \otimes I_M\big) = \mathbf{0}$, which is equivalent to requiring $\Big(\big(\mathbf{A}(j) - I_N\big)\mathbf{1}\Big) \otimes I_M = \mathbf{0}$. It follows that the unbiasedness in consensus estimator $\overline{\mathbf{X}}(i,j)$ requires 0 to be an eigenvalue of the Laplacian matrix $\mathtt{L}(k)$ with the associated eigenvector $\mathbf{1}$[1]. Then, it can be easily seen that

$$\begin{aligned}
\overline{P}(i,j+1) &= \big(\mathbf{A}(j) \otimes I_M\big)\overline{P}(i,j)\big(\mathbf{A}(j) \otimes I_M\big)^T \qquad (7) \\
&\quad + \mathbb{E}\Big\{\big(\Gamma(j) \otimes I_M\big)\overline{\Phi}(j)\overline{\Phi}(j)^T\big(\Gamma(j) \otimes I_M\big)^T\Big\}.
\end{aligned}$$

[1]Note that, similar results on the unbiasedness of consensus estimator was obtained in [3].

As shown in Fig. 5, after J consensus iterations each node n will feed $\overline{\mathbf{x}}_n(i, J)$ back to their local Kalman filters by setting $\overline{\mathbf{x}}_n(i|i) = \overline{\mathbf{x}}_n(i, J)$ and $\overline{P}_n(i|i) = \overline{P}_n(i, J)$ before starting next tracking update at time $i+1$. Recall that here $\overline{P}_n(i, J)$ is the n-th $M \times M$ main diagonal block of $\overline{P}(i, J)$.

IV. Convergence Analysis of the Proposed Algorithm

In this section, we analyze the convergence of the proposed distributed tracking with consensus algorithm and characterize its rate of convergence. We assume that the information exchange rate during the consensus updates is much higher compared to the data sampling rate for the tracking updates. Hence, we can assume that $J \gg 1$, guaranteeing enough time for information to be exchanged over the network so that consensus can be reached if the weights $\{\gamma(j)\}$ are chosen properly. As mentioned above, for a fixed i and $J \gg 1$, the problem is that of consensus in estimation. Thus, to simplify notation, in the following we drop the tracking time step index i in $\overline{\mathbf{X}}(i, j)$.

In the following theorem, we state a result that establishes *almost-sure (a.s.)* convergence of the sequence of component-wise averages $\{\overline{\mathbf{X}}_{\text{avg}}(j)\}_{j\geq 0}$ of the proposed distributed tracking with consensus algorithm to a finite random variable $\mathbf{\Theta}$, where $\overline{\mathbf{X}}_{\text{avg}}(j) = \frac{1}{N}(\mathbf{1}^T \otimes I_M)\overline{\mathbf{X}}(j)$. The proof is lengthy, and thus due to space limitations, we omit the proof.

Theorem 1 (a.s. convergence to a finite random vector): Consider the consensus algorithm in (5) with initial state $\overline{\mathbf{X}}(0) \in \mathbb{R}^{NM}$. If the connectivity graph Laplacian $L(k)$ with mean $\overline{L} = \mathbb{E}\big[L(k)\big]$ is such that $\lambda_2(\overline{L}) > 0$, and if $p(l, n) > 0$ for $(l, n) \in E(k)$, then there exists an almost sure finite real random vector $\mathbf{\Theta}$ such that

$$\mathbb{P}\left[\lim_{j\to\infty} \overline{\mathbf{X}}(j) = \mathbf{1}_N \otimes \mathbf{\Theta}\right] = 1.$$

Proof: Omitted due to space, but steps follow closely those of the proof in the scalar case given in [9]. ∎

Theorem 1 essentially states that the proposed distributed tracking with consensus algorithm will reach consensus almost surely and the consensus estimate $\lim_{j\to\infty} \overline{\mathbf{x}}(j)$ is a finite random vector $\mathbf{\Theta}$. Since the consensus algorithm in (5) falls in the framework of stochastic approximation, we may also analyze the convergence rate of the consensus algorithm based on the ODE method (Ordinary Difference Equation) [8]. The next theorem characterizes an upper bound to the convergence rate of the proposed distributed tracking with consensus algorithm.

Theorem 2 (convergence rate): Consider the consensus algorithm in (5) with initial state $\overline{\mathbf{X}}(0) \in \mathbb{R}^{NM}$. For a fixed i, let $J_i = \inf\{j | (S^k)^c = \emptyset$ and $k = iJ + j, j \geq 0\}$. For $k = iJ + j$ and $j \geq J_i$, the effective network graph Laplacian is $\mathtt{L}(k) = \overline{\mathtt{L}} + \tilde{\mathtt{L}}(k)$ with mean $\overline{\mathtt{L}} = \mathbb{E}\big[\mathtt{L}(k)\big]$. If the connectivity graph Laplacian $L(k)$ with mean $\overline{L} = \mathbb{E}\big[L(k)\big]$ is such that $\lambda_2(\overline{L}) > 0$, and if $p(l, n) > 0$ for $(l, n) \in E(k)$, the convergence rate of the proposed consensus algorithm is bounded by $-\lambda_2(\overline{\mathtt{L}})\left(\sum_{J_i\leq j\leq\infty}\gamma(j)\right)$.

Proof: From the asymptotic unbiasedness of $\mathbf{\Theta}$, we have $\lim_{j\to\infty}\mathbb{E}\big[\overline{\mathbf{X}}(j)\big] = \mathbf{1}_N \otimes \mathbf{r}$, where $\mathbf{r} = \overline{\mathbf{X}}_{\text{avg}}(J_i)$. For $j \geq J_i$, define $\Xi(j) = I_{NM} - \gamma(j)(\overline{\mathtt{L}} \otimes I_M)$, where $\overline{\mathtt{L}} = \mathbb{E}\big[\mathtt{L}(k)\big]$. Using the fact that $\mathtt{L}(k)$ and $\overline{\mathbf{X}}(j)$ are independent, and $\mathbb{E}\big[\overline{\Phi}(j)\big] = \mathbf{0}_{NM}$, from (5), we have that $\forall j \geq J_i$

$$\mathbb{E}\big[\overline{\mathbf{X}}(j+1)\big] = \Xi(j)\mathbb{E}\big[\overline{\mathbf{X}}(j)\big] = \prod_{l=J_i}^{j} \Xi(l)\mathbb{E}\big[\overline{\mathbf{X}}(J_i)\big]. \qquad (8)$$

From the persistence condition $\gamma(j) > 0, \sum_{j\geq 0}\gamma(j) = \infty$ and $\sum_{j\geq 0}\gamma^2(j) \leq \infty$ [9], it follows that $\gamma(j) \to 0$. Hence, without loss of generality, we can assume that $\gamma(j) \leq \frac{2}{\lambda_2(\overline{\mathtt{L}})+\lambda_n(\overline{\mathtt{L}})}, \forall j$ [10]. From the mixed-product property of Kronecker product $(A \otimes B)(C \otimes D) = AB \otimes CD$ and $\big(I_{NM} - \gamma(j)\overline{\mathtt{L}}\big)\mathbf{1}_N = \mathbf{1}_N$ [11], we have

$$\mathbf{1}_N \otimes \mathbf{r} = \Xi(j)\big(\mathbf{1}_N \otimes \mathbf{r}\big). \qquad (9)$$

From (8) and (9), it can be shown that

$$\begin{aligned} &\|\mathbb{E}\big[\overline{\mathbf{X}}(j)\big] - \mathbf{1}_N \otimes \mathbf{r}\|, \\ \leq &\prod_{J_i\leq l\leq j-1} \overline{\rho}\big(1 - \gamma(l)\overline{\mathtt{L}}\big)\|\mathbb{E}\big[\overline{\mathbf{X}}(J_i)\big] - \mathbf{1}_N \otimes \mathbf{r}\|, \\ = &\prod_{J_i\leq l\leq j-1} \big(1 - \gamma(l)\lambda_2(\overline{\mathtt{L}})\big)\|\mathbb{E}\big[\overline{\mathbf{X}}(J_i)\big] - \mathbf{1}_N \otimes \mathbf{r}\|, \end{aligned}$$

where last step follows from Lemma 8 of [4] and $\overline{\rho}(\cdot)$ denotes the spectral radius of a matrix. Since $1 - \gamma \leq e^{-\gamma}$ and $0 \leq \gamma \leq 1$, we then have that

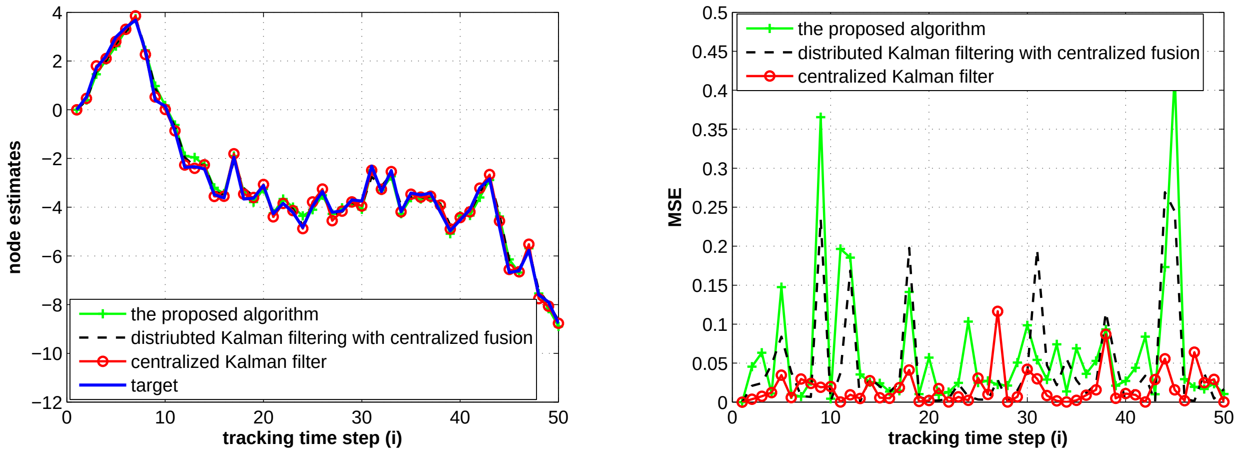
$$\begin{aligned} &\|\mathbb{E}\big[\overline{\mathbf{X}}(j)\big] - \mathbf{1}_N \otimes \mathbf{r}\| \\ \leq &\Big(e^{-\lambda_2(\overline{\mathtt{L}})\big(\sum_{J_i\leq l\leq j-1}\gamma(l)\big)}\Big)\|\mathbb{E}\big[\overline{\mathbf{X}}(J_i)\big] - \mathbf{1}_N \otimes \mathbf{r}\|. \end{aligned} \qquad (10)$$

Therefore, as $j \to \infty$, the convergence rate is bounded by $-\lambda_2(\overline{\mathtt{L}})\left(\sum_{J_i\leq l\leq\infty}\gamma(l)\right)$, which depends on the algebraic connectivity $\lambda_2(\overline{\mathtt{L}})$ and the weights $\gamma(j)$, for $j \geq J_i$. ∎

Theorem 2 shows that the convergence rate of the proposed algorithm depends on the topology through the algebraic connectivity $\lambda_2(\overline{\mathtt{L}})$ of the effective network graph $\tilde{G}(k)$ and through weights $\gamma(j)$, for $j \geq J_i$. Since for $k = iJ + j$ and $j \geq J_i$, $I_{S^k} = I$ and $L(k) = \mathtt{L}(k)$, we have $\overline{\mathtt{L}} = \mathbb{E}\big[\mathtt{L}(k)\big] = \mathbb{E}\big[L(k)\big]$. In (10), $\lambda_2(\overline{\mathtt{L}})$ is the algebraic connectivity of the mean Laplacian corresponding to the time-varying network graphs. For a static network, this reduces to the algebraic connectivity of the static Laplacian $\mathtt{L}$. We can speed up the convergence process by optimizing algebraic connectivity $\lambda_2(\mathbb{E}\big[L(k)\big])$ of the connectivity graph $G(k)$ and weights $\gamma(j)$, for $k = iJ + j$ and $j \geq J_i$.

V. Space-Object Tracking and Scheduling with a Hybrid Space Surveillance Network

The task of a space surveillance network is to keep track of space-born *objects of interest* (OoIs). However, this is a challenging problem especially if we presume an SSN made of a handful of ground-based sensor nodes connected in a *centralized architecture*. The number of possible space OoIs that could be present in the space *region of interest* (RoI), on



(a) (b)

Fig. 4: Comparison of the proposed distributed tracking with consensus algorithm with distributed local Kalman filtering with centralized fusion and centralized Kalman filter. (a) node estimates. (b) mean squared error.

the other hand, may be several orders of magnitudes larger than the number of sensor nodes. There is the possibility that such a fixed-node architecture may not be able to fully cover the space RoI, leaving some space objects never to be detected. As a solution to these constraints we recently proposed to integrate a small number of space-born *mobile sensor nodes* (i.e. satellites) to a ground-based fixed SSN [12].

Even with such a hybrid SSN, due to resource constraints it is important to assign detected targets to the sensor nodes judiciously for the purpose of tracking. This node-target scheduling is also important due to the sensing limitations of nodes and the finite maximum number of targets each node can sense/track at any given time. Traditionally, this tracking and node-target scheduling would be implemented in a centralized processing architecture. While a centralized SSN architecture has its own advantages, a *distributed architecture* may possibly be flexible and more efficient: The distributed architecture is easily scalable and perhaps more robust against node failures. However, since not all nodes may be able to observe a given target with the same quality, in a distributed processing architecture different sensor nodes may arrive at different local estimators regarding the *same* space object of interest. This, of course, would then lead to inconsistent distributed node-target scheduling decisions across the nodes in a hybrid SSN. This is clearly undesirable since it could lead to conflicting node actions leading disastrous outcomes.

The solution to this is to achieve consensus among distributed nodes in an SSN. The problem is then exactly the one we considered in this paper. Following each tracking update step, the nodes in an SSN performs a sequence of consensus exchanges among themselves to arrive at a consensus state estimate for a given target. The node-target scheduling decisions will be based on these consensus estimates, and thus will be consistent across the whole network. The only difference however, is that in the case of SSA, the target dynamics given by orbital equations are highly non-linear. Hence, rather than the standard KF, as discussed above, one needs to employ a suitable nonlinear tracking algorithm for the distributed local tracking step: For example, in [12] the Extended Kalman Filter (EKF) was considered. Indeed, in [12] it was established that a sufficient number of repeated exchanges can diffuse information throughout the whole SSN allowing all distributed estimators to converge to a single *consensus estimator*.

VI. Numerical Examples

In this section, we evaluate the performance of the proposed distributed tracking with consensus algorithm on a simulated sensor network, and compare it with that of the centralized Kalman filter and the distributed local Kalman filtering with centralized fusion. In distributed local Kalman filtering with centralized fusion, all nodes send their filtered estimates to a fusion center (FC). The FC then generates a fused estimate $\hat{\mathbf{x}}_{\text{fusion}}(i) = \frac{1}{|S^k|}\sum_{n\in S^k}\hat{\mathbf{x}}_n(i|i)$ for $k = iJ$ and $i = 0, 1, \cdots$.

In the first simulation we compare the performance of the proposed algorithm with the distributed local Kalman filtering with centralized fusion and the centralized Kalman filter over a random graph with noisy communication links and incomplete data. We consider a random connectivity graph $G(N, p)$ with $N = 20$ and the probability that each link exists $p = 0.5$. The other parameters of the simulation setup are: $F = 1$, $Q = 1$, $x(0) = 0$, $P_0 = 0$, $R_n = 0.25$, $H_n = 1$, $\Sigma_{l,n} = \Sigma = 0.1$, $S^k = \{n | 1 \le n \le 10, n \in \mathbb{Z}\}$ for $k = iJ$, $i = 0, 1, \cdots$ and $J = 30$. Figure 4a shows the node estimates from the three algorithms. As we see from Fig. 4a, the node estimates of the three algorithms follow the target's trajectory very closely.

Figure 4b compares the resulting mean squared error (MSE) of the three algorithms, where the MSE of the distributed tracking with consensus is defined to be the average MSE over all nodes $\frac{1}{N}\sum_{n=1}^{N}\left[\left(\bar{\mathbf{x}}_n(i,J) - \mathbf{x}(iJ)\right)^T\left(\bar{\mathbf{x}}_n(i,J) - \mathbf{x}(iJ)\right)\right]$. From Fig. 4b it can be seen that the MSE of the proposed distributed tracking with consensus algorithm is close to that of the distributed local Kalman filtering with centralized fusion. As expected, both of them are higher than the MSE of the centralized Kalman filter, which acts as a benchmark. The results in Fig. 4b show that the performance of the proposed distributed tracking with consensus algorithm is close to that of the distributed local Kalman filtering with centralized fusion in a time-varying random graph with noisy communication and incomplete data. Essentially, distributed consensus does not loose much compared to combining all estimates at a central fusion node! Of course, the proposed algorithm has the advantages of fully distributed implementation, robustness and scalability.

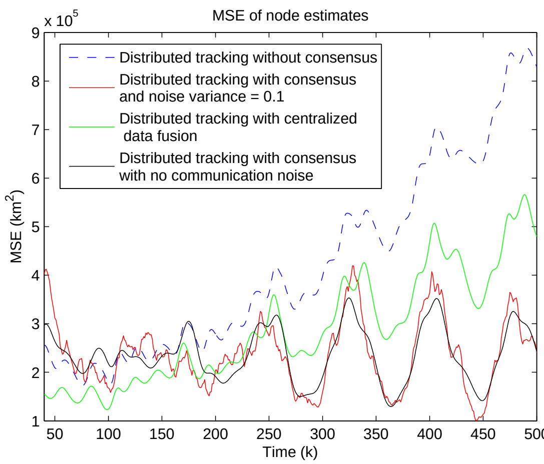


Fig. 5: The MSE performance. $J = 10$ in the consensus algorithm.

As our next example, we choose a two-dimensional SSN with 5 sensors total, 4 of which are ground based with the remaining node orbiting around the Earth. Four ground based sensors are spaced evenly around the Earth and the orbiting sensor is assumed to have an orbital radius of 6320 km. We consider a single target of interest that has a slightly eccentric orbit ($e = 0.0597$) with a mean orbital radius of 11776 km. The sample time $T = 172$s for the simulation, and each simulation lasts one sidereal day (86164s). In our simulation, we model the 2-D nonlinear orbital dynamics of the target and the sensor node in an orbit, and replace the KF with the EKF. We omit the details here. Figure 5 shows the average Mean-Squared Error (MSE) defined as $\frac{1}{n}\sum_{i=1}^{n}||\hat{\mathbf{x}} - \mathbf{x}||^2$, at each time instant k with the proposed distributed consensus algorithm assuming both zero communications noise as well as noise with $\sigma_c^2 = 0.1$. Figure 5 also shows the average MSE if local nodes run their own local EKF's without any data/estimate exchanges, as well as the MSE if these local estimates were fused by a central fusion node. As seen in 5 the consensus exchanges among nodes as in the proposed algorithm reduce the average MSE of distributed local estimates. In some cases, the consensus can help reduce the average MSE even beyond that with central fusion of distributed estimates. This is because with the proposed consensus algorithm there is the likelihood that at some point even a node which does not have any observations might get a better local estimate due to the information received from other nodes. From then on, this node might be able to predict a reasonably good estimate for the target location. Without consensus exchanges, however, these nodes will always have bad estimates and thus fusing them at a central node may not necessarily help.

VII. Conclusions

In this paper, we considered the problem of distributed tracking with consensus on a time-varying graph with incomplete data and noisy communication links. We developed a distributed and collaborative algorithm that performs distributed tracking with consensus in order to obtain a consensus target estimate across all the nodes in the network. We discussed the conditions for achieving consensus and quantified the convergence rate of the proposed algorithm. We discussed how this problem of tracking with consensus over time-varying noisy networks arises naturally in the context of space situational awareness when one needs to track space-borne objects with an SSN. Our simulation results showed that the performance of the proposed distributed tracking with consensus algorithm is close to that of the distributed local Kalman filtering with centralized fusion.

Acknowledgment

This research was supported in part by the Space Vehicles Directorate of the Air Force Research Laboratory (AFRL), and in part by the National Science foundation (NSF) under the grant CCF-0830545.

References

[1] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1520–1533, Sep. 2004.

[2] B. Teixeira, M. A. Santillo, R. S. Erwin, and D. S. Bernstein, "Spacecraft tracking using sampled-data kalman filters," *IEEE Control Sys. Magazine*, pp. 78–94, Aug. 2008.

[3] S. K. Jayaweera, "Distributed space-object tracking and scheduling with a satellite-assisted collaborative space surveillance network," AFRL Summer Faculty Fellowship Program, Final project report, July 2009.

[4] S. Kar and J. M. F. Moura, "Sensor networks with random links: Topology design for distributed consensus," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3315–3326, July 2008.

[5] A. Kashyap, T. Basar, and R. Srikant, "Quantized consensus," *Automatica*, vol. 43, no. 7, pp. 1192–1203, July 2007.

[6] F. R. K. Chung, *Spectral Graph Theory*. Providence, RI: American Mathematical Society, 1997.

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

[8] N. Ghasemi, S. Dey, and J. Baras, "Stochastic average consensus filter for distributed hmm filtering: Almost sure convergence," Institute for Systems Research Technical Reports, Tech. Rep., May 2010.

[9] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks: Link failures and channel noise," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 355–369, Jan. 2009.

[10] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Systems and Control Letters*, vol. 53, pp. 65–78, 2004.

[11] A. J. Laub, *Matrix Analysis for Scientists and Engineers*. SIAM: Society for Industrial and Applied Mathematics, 2004.

[12] S. K. Jayaweera, R. S. Erwin, and J. Carty, "Distributed space situational awareness (d-ssa) with a satellite-assisted collaborative space surveillance network," in *18th World Congress of the International Federation of Automatic Control (IFAC)*, Milan, Italy, Aug. 2011, in review.