Distributed Tracking with Consensus on Noisy Time-varying Graphs with Incomplete Data

Sudharman K. Jayaweera, Yongxiang Ruan and R. Scott Erwin

***Abstract*—In this paper we address the problem of distributed tracking with consensus on a time-varying graph with incomplete data and noisy communication links. We develop a distributed and collaborative tracking with consensus algorithm by combining distributed Kalman filtering with consensus updates to handle a time-varying network topology in which not every node has local observations to generate own local tracking estimates. We introduce 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. Simulation results and performance analysis of the proposed algorithm are given and compared with that of distributed local Kalman filtering with centralized fusion.**

I. Introduction

Distributed consensus estimation in sensor networks has gained considerable attention in recent years [1]. The problem has been studied with both fixed as well as time varying communication topologies, taking into account issues such as link failure, packet losses, quantization noise or additive channel noise [2]–[5]. Recent work in [4], [6], [8] have also considered the distributed consensus tracking over networks with noiseless communication links among nodes.

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. For instance, space-object tracking with a satellite surveillance network could benefit from distributed tracking with consensus, due to the fact that individual sensor nodes may not have enough observations of sufficient quality and different sensor nodes may arrive at different local estimates regarding the same space object of interest [6]. Information exchange among nodes may improve the quality of local estimates and help avoids conflicting and inefficient decisions. Other examples include flocking and formation control, real-time monitoring, target tracking and GPS systems [6], [7].

In this work, we consider 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 have active links with other nodes only within their communication radii. We develop a framework by combining distributed Kalman filtering with consensus updates to handle a time-varying network topology in which not every node has local observations to generate own local tracking estimates (incomplete data). We introduce the concepts of an *active node set* and a *connectivity graph* to characterize such a network. By merging the active node set with the connectivity graph, an *effective network graph* is obtained.

Following notation will be used in this paper: At time k, an undirected graph is denoted by $G(k) = (V, E(k))$, where $V = 1, 2, \cdots, n$ and $E(k) \subseteq V \times V$ for $k \geq 0$. The neighborhood of node i at time k is denoted by $\Omega_i(k) = \{l \in V | (i,l) \in E(k)\}$. Node i has degree $d_i(k) = |\Omega_i(k)|$. Let the degree matrix at time k be the diagonal matrix $D(k) = \text{diag}(d_1(k), \cdots, d_n(k))$, where $\text{diag}(d_1, \cdots, d_n)$ represents a diagonal matrix with $d_1, \cdots, d_n$ on its main diagonal. The adjacency matrix at time k is $A(k) = [A_{il}(k)]$, $A_{il}(k) = 1$, if $(i,l) \in E(k)$, 0 otherwise. The graph Laplacian matrix is $L(k) = D(k) - A(k)$. The Laplacian is a positive semidefinite matrix so that its eigenvalues can be ordered as $0 = \lambda_1(L)) \leq \lambda_2(L) \leq \cdots \leq \lambda_n(L)$. For a connected graph, $\lambda_2(L) > 0$ [5]. We will use the notation $G(n,p)$ to denote a random graph with n vertices, in which each edge is taken randomly and independently with probability $p \in (0,1]$.

II. Problem Formulation

A. System Model

Consider an n-node sensor network with a *connectivity graph* $G(k) = (V, E(k))$ at time k. Assume that the graph $G(k)$ is undirected and time-varying due to relative motion or battery constraints of nodes. The objective is to perform distributed tracking of a target state that is modelled as a linear, finite-dimensional system [9]

$$\mathbf{x}(k+1) = F\mathbf{x}(k) + \mathbf{u}(k); \quad \mathbf{x}(0) \sim \mathcal{N}(\overline{\mathbf{x}}(0), P_0), \tag{1}$$

where $\mathbf{x}(0) \in \mathbb{R}^N$ is the initial state of the target assumed to be Gaussian distributed. The sensing model of the ith sensor is

$$\mathbf{y}_i(k) = H_i\mathbf{x}(k) + \mathbf{v}_i(k); \quad \mathbf{y}_i \in \mathbb{R}^M, \tag{2}$$

where $\mathbf{y}_i(k)$ represents the i-th node's measurement for $1 \leq i \leq n$ and we assume that the H_i's can be different for each node. Both $\mathbf{u}(k)$ and $\mathbf{v}_i(k)$ are zero-mean white Gaussian noise (WGN). The statistics of the process and measurement noise are given by $\mathbb{E}[\mathbf{u}(k)\mathbf{u}(k')^T] = Q\delta_{kk'}$, $\mathbb{E}[\mathbf{v}_i(k)\mathbf{v}_l(k')^T] = R_i\delta_{kk'}\delta_{il}$, where $\delta_{kk'} = 1$ if $k = k'$ and $\delta_{kk'} = 0$, otherwise.

At the end of the k-th tracking update step, node i which has an observation of the target will have a filtered local estimate

Sudharman K. Jayaweera and Yongxiang Ruan are with the Communications and Information Sciences Lab (CISL), Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA Email: jayaweera@ece.unm.edu and yruan@unm.edu.

R. Scott Erwin is with the Air Force Research Laboratory (AFRL), Space Vehicles Directorate, Kirtland AFB, NM 87117.

$\hat{\mathbf{x}}_i(k|k)$ with associated covariance matrix $\hat{P}_i(k|k)$. In order to improve the consistency of its estimate against those of the other nodes, node i will exchange this filtered estimate with other nodes over noisy communication links. Due to time-varying topology of the network, at any given time k not all nodes may observe the target. These nodes will not have filtered local tracking estimates. Note that, here the goal is to obtain a consensus tracking estimate over the local estimates at each k, and thus, the consensus problem is essentially a problem of consensus estimation.

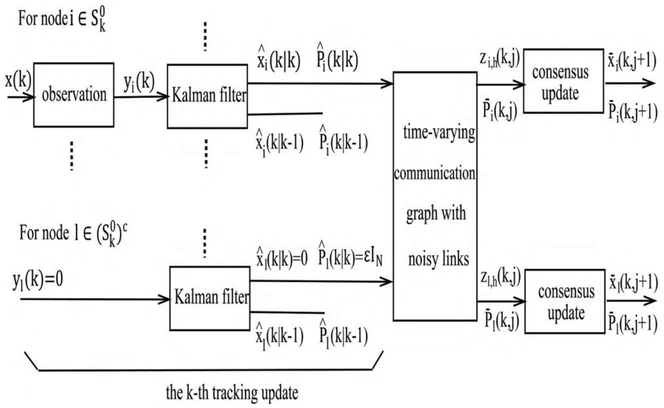

Fig. 1. Block diagram of distributed tracking with consensus on a time-varying graph with incomplete data and noisy communication links.

Figure 1 shows the system model of distributed tracking with consensus on a time-varying graph with incomplete data and noisy communication links. Let $\overline{\mathbf{x}}_i(k,j)$ denote the node i's updated tracking estimate at the j-th consensus iteration that follows the k-th tracking update step with $\overline{\mathbf{x}}_i(k,0) = \hat{\mathbf{x}}_i(k|k)$. The received data at node i from node l at iteration j can be written as $\mathbf{z}_{i,l}(k,j) = \overline{\mathbf{x}}_l(k,j)+\mathbf{w}_{i,l}(j)$, for $0 \leq j \leq J$, where $\mathbf{w}_{i,l}(j)$ denotes the receiver noise at the node i in receiving the estimates of node l at iteration j with $\mathbb{E}[\mathbf{w}_{i,l}(j)] = \mathbf{0}_N$ and $\mathbb{E}[\mathbf{w}_{i,l}(j)\mathbf{w}_{i,l}^T(j)] = \Sigma_{i,l}$, $\mathbf{z}_{i,i}(k,j) = \overline{\mathbf{x}}_i(k,j)$ and J is the number of iterations in each consensus update cycle.

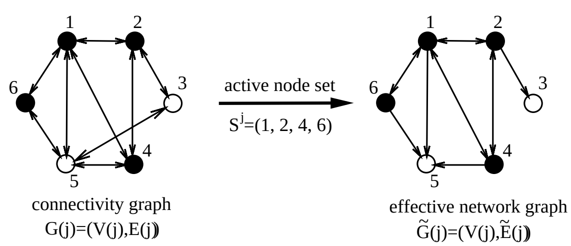

Fig. 2. Connectivity graph and effective network graph.

B. Network Model

We define the *active node set* S^j in a time-varying graph $G(j)$ as the set of nodes that have local estimates to be shared with others at the beginning of the j-th consensus iteration [6]. Define the *effective network graph* $\tilde{G}(j) = \big(V(j), \tilde{E}(j)\big)$ of a network $G(j)$ with active node set S^j as a graph $G(j)$ with the outgoing edges of the nodes that do not have data removed. Note that, the nodes that do not observe the target will not have filtered estimates to share at the beginning of consensus update process. However, as information exchange among nodes progresses, some of these nodes may be able to form their own local estimates to be shared with others at the beginning of iteration j for $j > 0$. Therefore, the active node set S^j is time-varying and is a function of S^{j-1} and $G(j-1)$. Figure 2 shows the relation between the connectivity graph $G(j)$ and the effective network graph $\tilde{G}(j)$ for a graph of 6 nodes with active node set $S^j = (1,2,4,6)$, where solid circles denote active nodes.

Let I_{S^j} denote an $n \times n$ diagonal matrix generated from the active node set S^j , where

$$[I_{S^j}]_{ii'} = \begin{cases} 1 & \text{if } i = i' \quad \text{and} \quad i' \in S^j \\ 0 & \text{else} \end{cases} . \tag{3}$$

By combining the connectivity graph $G(j)$ and active node set S^j, we obtain the effective network graph $\tilde{G}(j)$ for $j \geq 0$. The adjacency matrix of the effective network graph is $\mathtt{A}(j) = A(j)I_{S^j}$. The corresponding degree matrix $\mathtt{D}(j)$ can then be generated from the $\mathtt{A}(j)$, leading to the Laplacian matrix $\mathtt{L}(j) = \mathtt{D}(j) - \mathtt{A}(j)$.

III. Distributed and Collaborative Tracking with Consensus Algorithm

For the above distributed tracking problem over a time-varying graph with incomplete data and noisy communication links, we propose a distributed and collaborative tracking and consensus algorithm which is based on the architecture that was first proposed in [6]. At the k-th tracking update step, node i is assumed to observe the target and passes its observation $\mathbf{y}_i(k)$ through a Kalman filter as follows [9]:

$$\begin{aligned}
\hat{\mathbf{x}}_i(k|k-1) &= F\overline{\mathbf{x}}_i(k-1|k-1), \\
\hat{P}_i(k|k-1) &= F\overline{P}_i(k-1|k-1)F^T + Q, \\
K_k &= \hat{P}_i(k|k-1)H_i^T(H_i\hat{P}_i(k|k-1)H_i^T + R_i)^{-1}, \\
\hat{\mathbf{x}}_i(k|k) &= \hat{\mathbf{x}}_i(k|k-1) + K_k(\mathbf{y}_i(k) - H_i\hat{\mathbf{x}}_i(k|k-1)), \\
\hat{P}_i(k|k) &= (I - K_kH_i)\hat{P}_i(k|k-1),
\end{aligned} \tag{4}$$

where $\overline{\mathbf{x}}_i(k-1|k-1) = \overline{\mathbf{x}}_i(k-1,J)$ with $\overline{\mathbf{x}}_i(-1,J) = \overline{\mathbf{x}}(0)$ and $\overline{P}_i(k-1|k-1) = \overline{P}_i(k-1,J)$ with $\overline{P}_i(-1,J) = P_0$. Denote $\overline{P}(k-1,j)$ as the covariance matrix in the j-th consensus iteration after the $(k-1)$-th tracking update. The $\overline{P}_i(k-1,J)$ in (4) can be obtained by extracting the i-th $N \times N$ main diagonal block of $\overline{P}(k-1,J)$. Then, node i will have its filtered estimate $\hat{\mathbf{x}}_i(k|k)$ in tracking update and uses it as initial estimate for consensus update exchange by setting $\overline{\mathbf{x}}_i(k,0) = \hat{\mathbf{x}}_i(k|k)$ with initial covariance matrix $\overline{P}(k,0) = \hat{P}_1(k|k) \oplus \hat{P}_2(k|k) \oplus \cdots \oplus \hat{P}_n(k|k)$, where $\oplus$ denotes the direct sum of matrices. For nodes $i \in (S^0)^c$, it will only have its predicted estimate $\hat{\mathbf{x}}_i(k|k-1)$ and $\hat{P}_i(k|k-1)$ from the Kalman filter in (4) without observation. It may arbitrarily set $\hat{\mathbf{x}}_i(k|k) = \mathbf{0}$ and $\hat{P}_i(k|k) = \epsilon I_N$ for some $\epsilon > 0$ and use them as the initial estimate for the consensus stage.

During the j-th consensus update, each node i forms its consensus estimate by combing received noisy estimates from its neighbors [5]:

$$\begin{aligned}
&\overline{\mathbf{x}}_i(k,j+1) \\
=&\overline{\mathbf{x}}_i(k,j) + \gamma_i(j)\sum_{l=1}^{n} \mathtt{A}_{i,l}(j)(\overline{\mathbf{z}}_{i,l}(k,j) - \overline{\mathbf{z}}_{i,i}(k,j)),
\end{aligned} \tag{5}$$

where $\gamma_i(j)$ is the i-th node's weight coefficient at iteration j. We set $\gamma_i(j) = \gamma(j)$ for $i \in S^j$ and $\gamma_i(j) = \frac{1}{\sum_{l=1}^{n} \mathtt{A}_{i,l}(j)}$ for $i \in (S^j)^c$ and $\sum_{l=1}^{n} \mathtt{A}_{i,l}(j) \neq 0$. Here (5) is distributed average consensus with imperfect communication, where each sensor receives noise corrupted versions of its neighbors' states and the weight coefficient is different for each node [5].

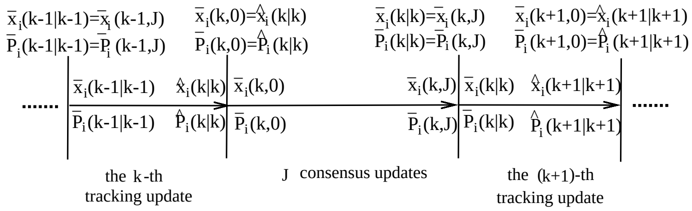

Fig. 3. Timing diagram of tracking and consensus updates in the proposed algorithm for distributed tracking with consensus.

Let $\overline{\mathbf{X}}(k,j) = [\overline{\mathbf{x}}_1(k,j)^T \ \overline{\mathbf{x}}_2(k,j)^T \cdots \overline{\mathbf{x}}_n(k,j)^T]^T$. Then, the consensus update dynamics can be written in vector form as follows:

$$\begin{aligned} &\overline{\mathbf{X}}(k,j+1) \\ =&\overline{\mathbf{X}}(k,j) - [(\Gamma(j)\mathtt{L}(j)) \otimes I_N]\overline{\mathbf{X}}(k,j) - (\Gamma(j) \otimes I_N)\overline{\mathbf{W}}(j), \end{aligned} \tag{6}$$

where $\Gamma(j) = \text{diag}(\gamma_1(j), \gamma_2(j), \cdots, \gamma_n(j))$, $\mathbf{w}_i(j) = -\sum_{l=1}^{n} \mathtt{A}_{i,l}(j)\mathbf{w}_{i,l}(j)$ and $\overline{\mathbf{W}}(j) = [\mathbf{w}_1(j)^T, \cdots, \mathbf{w}_n(j)^T]^T$. Define $\overline{\mathbf{e}}(k,j)$ as the error vector in the j-th consensus iteration that follows the k-th tracking update: $\overline{\mathbf{e}}(k,j) \triangleq \overline{\mathbf{X}}(k,j) - (\mathbf{1} \otimes I_N)\mathbf{x}(k)$. From (6), it follows that

$$\begin{aligned} \overline{\mathbf{e}}(k,j+1) = &(\mathbf{A}(j) \otimes I_N)\overline{\mathbf{e}}(k,j) - (\Gamma(j) \otimes I_N)\overline{\mathbf{W}}(j) \\ &+ \big((\mathbf{A}(j) \otimes I_N) - I\big)(\mathbf{1} \otimes I_N)\mathbf{x}(k), \end{aligned} \tag{7}$$

where $\mathbf{A}(j) = I_n - \Gamma(j)\mathtt{L}(j)$. Note that, this coefficient matrix $\mathbf{A}(j)$ is slightly different from the one in [6]. In [6], $\mathbf{A}(j) = \tilde{\mathbf{I}}(j) - \gamma(j)\tilde{L}(j)$, where $\tilde{\mathbf{I}}(j)$ and $\tilde{L}(j)$ are the modified identity and Laplacian matrices.

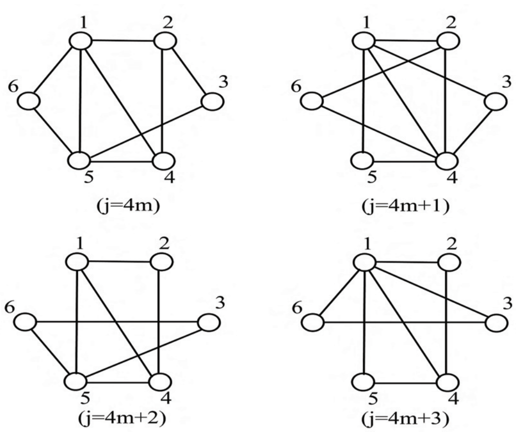

Fig. 4. A time-varying graph with switching topologies.

Assume that the filtered estimate $\hat{\mathbf{x}}_i(k|k)$ at the end of the measurement update stage is an unbiased estimate, so that $\overline{\mathbf{x}}_i(k,0)$ is unbiased for $i \in S^0$. From (5), for $i \in (S^j)^c$, we have $\overline{\mathbf{x}}_i(k,j+1) = \frac{1}{\sum_{l=1}^{n} \mathtt{A}_{i,l}(j)} \sum_{l=1}^{n} \mathtt{A}_{i,l}(j)\big(\overline{\mathbf{x}}_l(k,j) + \mathbf{w}_{i,l}(j)\big)$. Then, $\overline{\mathbf{x}}_i(k,j+1)$ is unbiased for $i \in (S^j)^c$ if $\overline{\mathbf{x}}_l(k,j)$ is unbiased for $l \in S^j$. From (7) the unbiasedness in the consensus estimate $\overline{\mathbf{X}}(k,j)$ can be maintained if matrix $\mathbf{A}(j)$ satisfies $\big((\mathbf{A}(j) \otimes I_N) - I\big)(\mathbf{1} \otimes I_N) = \mathbf{0}$. From this, it follows that the unbiasedness in consensus estimate $\overline{\mathbf{X}}(k,j)$ requires 0 is an eigenvalue of the Laplacian matrix $\mathtt{L}(j)$ with the associated eigenvector $\mathbf{1}$. Similar results on the unbiasedness of the consensus estimate was obtained in [6]. Then, it can easily be seen that

$$\begin{aligned} \overline{P}(k,j+1) = &(\mathbf{A}(j) \otimes I_N)\overline{P}(k,j)(\mathbf{A}(j) \otimes I_N)^T \\ &+ \mathbb{E}\{(\Gamma(j) \otimes I_N)\overline{\mathbf{W}}(j)\overline{\mathbf{W}}(j)^T(\Gamma(j) \otimes I_N)^T\}. \end{aligned} \tag{8}$$

After J consensus iterations, each node i feeds $\overline{\mathbf{x}}_i(k,J)$ back to the Kalman filter by setting $\overline{\mathbf{x}}_i(k|k) = \overline{\mathbf{x}}_i(k,J)$ with covariance matrix $\overline{P}_i(k|k) = \overline{P}_i(k,J)$ before starting the next tracking update for $k+1$. Figure 3 shows the timing diagram of tracking and consensus updates process in the proposed distributed and collaborative tracking with consensus algorithm, summarized in Algorithm 1.

Algorithm 1 Distributed and Collaborative Tracking with Consensus Algorithm

Initialize: $\mathbf{x}(0)$, F, H_i, Q,R_i
 while new data exists **do**
  Kalman filtering in tracking process:
   $\hat{\mathbf{x}}_i(k|k-1) = F\overline{\mathbf{x}}_i(k-1|k-1)$
   $\hat{P}_i(k|k-1) = F\overline{P}_i(k-1|k-1)F^T + Q$
   $K_i(k) = \hat{P}_i(k|k-1)H_i^T(H_i\hat{P}_i(k|k-1)H_i^T + R_i)^{-1}$
   $\hat{\mathbf{x}}_i(k|k) = \hat{\mathbf{x}}_i(k|k-1) + K_i(k)(\mathbf{y}_i(k) - H_i\hat{\mathbf{x}}_i(k|k-1))$
   $\hat{P}_i(k|k) = (I - K_i(k)H_i)\hat{P}_i(k|k-1)$
  update the initial state of consensus process:
   $\overline{\mathbf{x}}_i(k,0) \leftarrow \hat{\mathbf{x}}_i(k|k)$
   $\overline{P}(k,0) \leftarrow \hat{P}_1(k|k) \oplus \hat{P}_2(k|k) \oplus \cdots \oplus \hat{P}_n(k|k)$
   $j \leftarrow 0$
  while $j \leq J-1$ **do**
   $\overline{\mathbf{x}}_i(k,j+1) = \overline{\mathbf{x}}_i(k,j) + \gamma(j)\sum_{l=1}^{n} A_{i,l}(j)(\mathbf{z}_{i,l}(k,j) - \mathbf{z}_{i,i}(k,j))$
   $\overline{P}(k,j+1) = (\mathbf{A}(j) \otimes I_N)\overline{P}(k,j)(\mathbf{A}(j) \otimes I_N)^T + \mathbb{E}\{(\Gamma(j) \otimes I_N)\overline{\mathbf{W}}(j)\overline{\mathbf{W}}(j)^T(\Gamma(j) \otimes I_N)^T\}$
   $j \leftarrow j+1$
  end while
  $\overline{\mathbf{x}}_i(k|k) = \overline{\mathbf{x}}_i(k,J)$
  $\overline{P}_i(k|k) = \overline{P}_i(k,J)$
  $k \leftarrow k+1$
 end while

IV. Numerical Examples

In this section, we consider the performance of the distributed tracking with consensus algorithm and compare it with the distributed local Kalman filtering with centralized fusion, in which all nodes send the filtered estimates to a fusion center, and the fusion center generates a fused estimate $\hat{\mathbf{x}}_{\text{fusion}}(k) = \frac{1}{|S^0|}\sum_{i \in S^0} \hat{\mathbf{x}}_i(k|k)$. The assumed parameters in the first simulation setup are as follows: $F = 1$, $Q = 1$, $x(0) = 0$, $P_0 = 0$, $H_i = 1$, $R_i = 0.25$, $\Sigma_{l,i} = \Sigma = 0.01$, $n = 6$, $J = 30$ and $S^0 = \{1,3,4,6\}$. Figures 5-6 show the performance of the proposed distributed tracking with consensus on a time-varying graph with switching topologies.

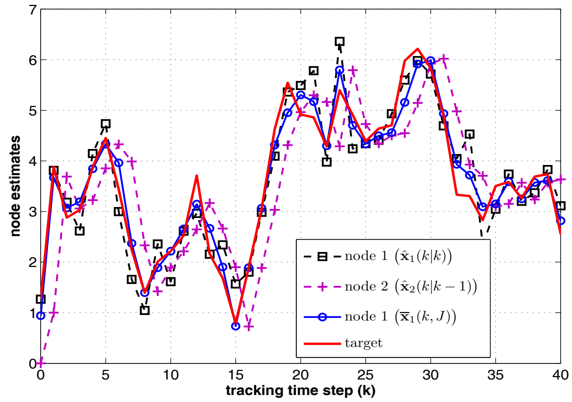

Fig. 5. Node consensus estimates $\overline{\mathbf{x}}_i(k,J)$ over a graph with switching

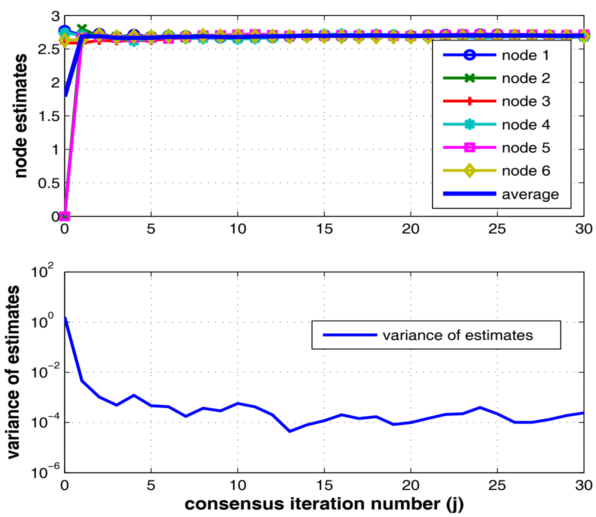

Fig. 6. Node estimates $\overline{\mathbf{x}}_i(k,j)$ and variance of the node estimates during the consensus updates.

The topologies of the graph are described in Fig. 4, where $m = 0, 1, 2, \cdots$. As can be seen for Fig. 5, the proposed algorithm ensures that all nodes very closely follow the target trajectory. Note that, the filtered estimate $\hat{\mathbf{x}}_1(k|k)$ is plotted for node $1 \in S^0$ and the predicted estimate $\hat{\mathbf{x}}_2(k|k-1)$ is also plotted for node $2 \in (S^0)^c$. Moreover, Figs. 6 shows that indeed the consensus algorithm helps bring local estimates closer within very few exchanges even in the presence of noise.

In the second simulation, we consider a random connectivity graph $G(n,p)$ with $n = 50$ and the probability that each link exists $p = 0.5$. The probability of each node having an observation at a given time instant is $p_{\mathrm{s}} = 0.9$. The other parameters of the simulation setup are as follows [8]: $F = I_2 + \epsilon F_0 + \frac{\epsilon^2}{2}F_0^2 + \frac{\epsilon^3}{6}F_0^3$, $F_0 = \begin{bmatrix} 0 & -2 \\ 2 & 0 \end{bmatrix}$, $\epsilon = 0.015$, $Q = (\epsilon c_w^2)^2 I_2$, $c_w = 5$, $x(0) = [15, -10]^T$, $H_i = [1, 0]$ for i is odd and $H_i = [0, 1]$ for i is even, $R_i = c_v^2\sqrt{i}$ for $i = 1, \cdots, n$ with $c_v = 30$, $\Sigma_{l,i} = \Sigma = 0.01$, $J = 10$. Figure 7 shows the node estimates (trajectory) of the two algorithms in this time-varying graph with incomplete data. As we can see, the proposed algorithm performs almost the same as the local Kalman filtering with centralized fusion. In Fig. 7, the solid curve denotes the target's trajectory and the dashed curve denotes the distributed local Kalman filtering with centralized fusion.

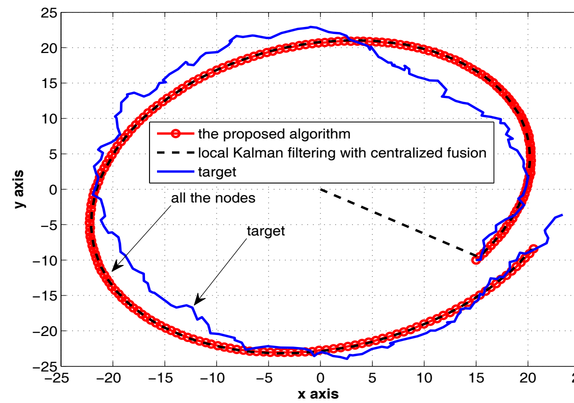

Fig. 7. Comparison of the proposed distributed tracking with consensus algorithm with local Kalman filtering with centralized fusion (trajectory).

V. Concluding Remarks

In this paper, we developed a distributed and collaborative tracking with consensus algorithm to achieve distributed tracking in a sensor network with incomplete data and a noisy time-varying graph. Our simulation results showed the proposed algorithm improves the estimation quality of each node and its performance is close to distributed local Kalman filtering with centralized fusion. The proposed algorithm does not require global knowledge of network topology and shows an advantage in scalability and robustness to dynamic changes of the network topology, which is preferable in practical applications.

Acknowledgment

This research was supported in part by the Space Vehicles Directorate of the Air Force Research Laboratory (AFRL).

References

[1] W. Ren, R. Beard, and E. Atkins, “A Survey of Consensus Problems in Multi-agent Coordination,” in Proc. Amer. Control Conf., 2005.

[2] 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(9), pp. 1520-1533, Sep. 2004.

[3] M. Yildiz and A. Scaglione, “Coding With Side Information for Rate-Constrained Consensus,” IEEE Trans. Signal Proc., Vol. 56, No. 8, 2008.

[4] C. Mosquera, R. Lopez-Valcarce, and S. K. Jayaweera, “Stepsize sequence design for distributed average consensus,” IEEE Signal Proc. Letters, Vol. 17, no. 2, pp. 169-172, Feb. 2010.

[5] S. Kar and J. M. F. Moura, “Distributed Consensus Algorithms in Sensor Networks: Link Failures and Channel Noise,” IEEE Trans. Signal Process., 57:1, pp. 355-369, Jan. 2009.

[6] S. K. Jayaweera, “Distributed Space-object Tracking and Scheduling with a Satellite-assisted Collaborative Space Surveillance Network,” Final project report: AFRL Summer Faculty Fellowship Program, Jul. 2009.

[7] Y. Cao, W. Ren and Y. Li, “Distributed discrete-time coordinated tracking with a time-varying reference state and limited communication,” Automatica, Vol. 45, No. 5, May 2009, pp. 1299-1305.

[8] R. Olfati-Saber, “Distributed Kalman Filter with Embedded Consensus Filters,” CDC 2005, pp. 8179-8184, Dec. 2005.

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