Performance Analysis of Distributed Tracking with Consensus on Noisy Time-varying Graphs

Yongxiang Ruan, Sudharman K. Jayaweera and Carlos Mosquera

***Abstract*—This paper considers a problem of distributed tracking with consensus on a time-varying graph with noisy communications links. A distributed tracking with consensus algorithm is proposed to handle a time-varying network topology in which every node generates own local tracking estimates and communicates over noisy links. The conditions on the connectivity graph are established so that distributed consensus can be achieved in the presence of noisy communication links when the graph topology is time-varying. The steady-state performance of the proposed distributed tracking with consensus algorithm is analyzed and compared with that of distributed local Kalman filtering with centralized fusion. Simulation results and performance analysis of the proposed algorithm are also given.**

I. INTRODUCTION

Multi-Sensor Tracking problems have attracted the attention of many researchers in robotics, systems, and control theory over the past three decades [1]. Modern target tracking problems are of great importance in surveillance, security, and information systems for monitoring the behavior of agents using sensor networks, such as tracking pallets in warehouses, vehicles on roadways, or firefighters in burning buildings. With the introduction of the concept of consensus, distributed tracking and coordination without any fusion center has also received considerable attention in recent years [2], [3].

Recent work in [4], [5] considers the distributed consensus tracking over a fixed graph with noiseless communication among nodes. A distributed Kalman filter with embedded consensus filters were proposed in [4] and further extended to heterogeneous and nonlinear sensing models in [5]. [6] studies a distributed discrete-time coordinated tracking problem under a fixed communication graph and proposes a PD-like discrete-time consensus algorithm to address the problem. In [7], the authors proposed a greedy stepsize sequence design to guarantee the convergence of distributed estimation consensus over a network with noisy links.

Distributed tracking with consensus, addressed in this paper, refers to the problem that a group of nodes that 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 estimators regarding the same space object of interest [8]. Information exchange among nodes may improve the quality of local estimators and help avoid conflicting and inefficient decisions. Other examples of application of tracking with consensus include flocking and formation control, real-time monitoring, target tracking and GPS systems [6], [8].

The contributions of this work are as follows: 1) model the problem of distributed tracking with consensus on a time-varying graph with noisy communications links, 2) develop a framework by combining distributed Kalman filtering with consensus updates to handle the issue of time-varying network topology and noisy communication links, 3) establish the conditions on the *connectivity graph* so that the distributed consensus can be achieved in the presence of noisy communication links when the graph topology is time-varying, and 4) analyze the steady-state performance of the distributed tracking with consensus and compare with that of distributed local Kalman filtering with centralized fusion.

The outline of the paper is as follows. Section II introduces our assumed system model. A distributed tracking with consensus algorithm is introduced in Section III and conditions for consensus are discussed and the rate of convergence is quantified. The steady-state performance of the distributed tracking with consensus is also analyzed in Section III. Section IV provides detailed simulation results and performance comparison of the proposed distributed tracking with consensus and distributed Kalman filtering with centralized fusion. Finally, concluding remarks are made in Section V.

II. SYSTEM MODEL

A network of n sensors is deployed to track the state of a target of interest. Let network topology at time k denoted by an undirected graph $G(k) = (V, E(k))$, 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 $L(k) = D(k) - A(k)$. Let $\overline{L} = \mathbb{E}[L(k)]$ denote the mean Laplacian for $k \geq 0$. The eigenvalues of the Laplacian can be ordered as $0 = \lambda_1(L) \leq \lambda_2(L) \leq \cdots \leq \lambda_n(L)$. For a connected graph, $\lambda_2(L) > 0$ [9]. A random graph in which the existence of an edge between a pair of

Yongxiang Ruan, Sudharman K. Jayaweera are with the Communications and Information Sciences Lab (CISL), Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM 87131-0001, USA. E-mail: yruan@unm.edu and jayaweera@ece.unm.edu

Carlos Mosquera is with Departamento de Teoría de la Señal y Comunicaciones, Universidad de Vigo, 36310 Vigo, Spain. E-mail: mosquera@gts.tsc.uvigo.es

vertices in the set $V = \{1, 2, \cdots, n\}$ is determined randomly and independent of other edges with probability $p \in (0, 1]$ is denoted by $G(n, p)$. Let $p(l, i)$ be the probability that link (l, i) of the graph exists (for random graphs considered in this paper we will always assume that $p(l, i) = p$ for $\forall l, i \in V$). The direct sum of an $N \times N$ matrix B and an $M \times M$ matrix C will be an $(N+M) \times (N+M)$ matrix, denoted by $B \oplus C$, whereas the Kronecker product of an $N \times N$ matrix B and an $M \times M$ matrix C will be an $NM \times NM$ matrix, denoted by $B \otimes C$.

The system dynamics of the target and the sensing model of the ith sensor are as follows:

$$\mathbf{x}(k+1) = F\mathbf{x}(k) + \mathbf{w}(k), \quad \mathbf{x}(0) \sim \mathcal{N}(\bar{\mathbf{x}}(0), P_0).$$
$$\mathbf{y}_i(k) = H_i\mathbf{x}(k) + \mathbf{v}_i(k), \quad \mathbf{y}_i \in \mathbb{R}^m, \tag{1}$$

and we assume that the H_i's can be different for each node. Both $\mathbf{w}(k)$ and $\mathbf{v}_i(k)$ are zero-mean white Gaussian noise (WGN) and $\mathbf{x}(0) \in \mathbb{R}^N$ is the initial state of the target. The statistics of the process and measurement noise are given by

$$\mathbb{E}[\mathbf{w}(k)\mathbf{w}(k')^T] = Q\delta_{kk'}, \mathbb{E}[\mathbf{v}_i(k)\mathbf{v}_{i'}(k')^T] = R_i\delta_{kk'}\delta_{ii'}, \tag{2}$$

where $\delta_{kk'} = 1$ if $k = k'$ and $\delta_{kk'} = 0$, otherwise.

At the end of tracking update, node i will have its filtered estimate $\hat{\mathbf{x}}_i(k|k)$ with associated covariance matrix $\hat{P}_i(k|k)$. In order to improve the tracking estimation accuracy, it will exchange its filtered estimate through noisy communication links and try to reach consensus over the network. 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 in estimation.

Figure 1 shows the system model of distributed tracking with consensus on a time-varying graph with noisy communications links. Let $\bar{\mathbf{x}}_i(k, j)$ denote the node i's updated tracking estimate at the j-th consensus iteration that follows the k-th tracking update step with $\bar{\mathbf{x}}_i(k, 0) = \hat{\mathbf{x}}_i(k|k)$, where $\hat{\mathbf{x}}_i(k|k)$ is the i-th node's filtered tracking estimate in the k-th tracking update. The received data at node l from node i at iteration j can be written as

$$\mathbf{z}_{l,i}(k, j) = \bar{\mathbf{x}}_i(k, j) + \boldsymbol{\phi}_{l,i}(j), \text{ for } 0 \le j \le J, \tag{3}$$

where $\boldsymbol{\phi}_{l,i}(j)$ denotes the receiver noise at the node l in receiving the estimator of node i at iteration j with $\mathbb{E}[\boldsymbol{\phi}_{l,i}(j)] = \mathbf{0}_N$ and $\mathbb{E}[\boldsymbol{\phi}_{l,i}(j)\boldsymbol{\phi}_{l,i}^T(j)] = \Sigma_{l,i}$, $\mathbf{z}_{i,i}(k, j) = \bar{\mathbf{x}}_i(k, j)$ and J is the number of iterations in consensus update. The distributed tracking with consensus problem as formulated above may have other applications beyond the space object tracking problem treated in [8] such as in multi-target tracking with a group of autonomous robots, battlefield life signs detection by using UAVs (Unmanned Aerial Vehicles), package tracking in warehouses using sensor networks, etc.

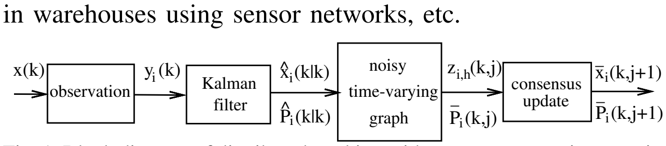

Fig. 1. Block diagram of distributed tracking with consensus on a time-varying graph with noisy communications links.

III. Distributed Tracking with Consensus Algorithm

A. Algorithm

In this section, we propose a distributed tracking and consensus algorithm for the above distributed tracking problem over a time-varying graph with noisy communications links. This is based on the architecture that was first proposed in [8] in the special context of consensus tracking in a satellite sensor network for situational awareness. At the tracking update step, node i passes its observation $\mathbf{y}_i(k)$ through its local Kalman filter as follows:

$$\hat{\mathbf{x}}_i(k|k-1) = F\bar{\mathbf{x}}_i(k-1|k-1),$$
$$\hat{P}_i(k|k-1) = F\overline{P}_i(k-1|k-1)F^T + Q,$$
$$K_i(k) = \hat{P}_i(k|k-1)H_i^T(H_i\hat{P}_i(k|k-1)H_i^T + R_i)^{-1},$$
$$\hat{\mathbf{x}}_i(k|k) = \hat{\mathbf{x}}_i(k|k-1) + K_i(k)(\mathbf{y}_i(k) - H_i\hat{\mathbf{x}}_i(k|k-1)),$$
$$\hat{P}_i(k|k) = (I - K_i(k)H_i)\hat{P}_i(k|k-1), \tag{4}$$

where $\bar{\mathbf{x}}_i(k-1|k-1) = \bar{\mathbf{x}}_i(k-1, J)$ with $\bar{\mathbf{x}}_i(-1, J) = \bar{\mathbf{x}}(0)$ 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 $\overline{P}(k-1, J)$.

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 $\bar{\mathbf{x}}_i(k, 0) = \hat{\mathbf{x}}_i(k|k)$ 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. During the $(j+1)$-th consensus update, each node i forms a linear estimate of the following form as its consensus estimate:

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

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

$$\overline{\mathbf{X}}(k, j+1) = \overline{\mathbf{X}}(k, j) - \gamma(j)[(L(j) \otimes I_N)\overline{\mathbf{X}}(k, j) + \overline{\Phi}(j)], \tag{6}$$

where $\boldsymbol{\phi}_i(j) = -\sum_{l=1}^{n} A_{i,l}(j)\boldsymbol{\phi}_{i,l}(j)$ and $\overline{\Phi}(j) = [\boldsymbol{\phi}_1(j)^T, \cdots, \boldsymbol{\phi}_n(j)^T]^T$. Let us define $\bar{\mathbf{e}}(k, j)$ be the error vector at the j-th consensus iteration after the k-th tracking update:

$$\bar{\mathbf{e}}(k, j) \triangleq \overline{\mathbf{X}}(k, j) - (\mathbf{1} \otimes I_N)\mathbf{x}(k). \tag{7}$$

From (6) and (7), it follows that

$$\begin{aligned}\bar{\mathbf{e}}(k, j+1) = (\mathbf{A}(j) \otimes I_N)\bar{\mathbf{e}}(k, j) - (\gamma(j) \otimes I_N)\overline{\Phi}(j)\\ + \big((\mathbf{A}(j) \otimes I_N) - I\big)(\mathbf{1} \otimes I_N)\mathbf{x}(k),\end{aligned} \tag{8}$$

where $\mathbf{A}(j) = I_n - \gamma(j)L(j)$.

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 $\overline{\mathbf{X}}(k,0)$ is unbiased. From (8) it can be shown that the unbiasedness in the consensus estimator $\overline{\mathbf{X}}(k,j)$ can be maintained if matrix $\mathbf{A}$ satisfies $\big((\mathbf{A}\otimes I_N)-I\big)(\mathbf{1}\otimes I_N)=\mathbf{0}$, which is equivalent to requiring $\big((\mathbf{A}-I_n)\mathbf{1}\big)\otimes I_N=\mathbf{0}$. It follows that the unbiasedness in consensus estimator $\overline{\mathbf{X}}(k,j)$ requires 0 is an eigenvalue of the Laplacian matrix $L(j)$ with the associated eigenvector $\mathbf{1}$. Then, it can be easily seen that

$$\begin{aligned}\overline{P}(k,j+1) &= (\mathbf{A}(j)\otimes I_N)\overline{P}(k,j)(\mathbf{A}(j)\otimes I_N)^T \\ &\quad + \gamma^2(j)\mathbb{E}\{\overline{\Phi}(j)\overline{\Phi}(j)^T\}.\end{aligned} \tag{9}$$

After J consensus iterations, each node i will feed $\overline{\mathbf{x}}_i(k,J)$ back to their local Kalman filters by setting $\overline{\mathbf{x}}_i(k|k)=\overline{\mathbf{x}}_i(k,J)$ and $\overline{P}_i(k|k)=\overline{P}_i(k,J)$ before starting the next tracking update for $k+1$. Figure 2 shows the timing diagram of tracking and consensus updates process in the proposed distributed tracking with consensus algorithm.

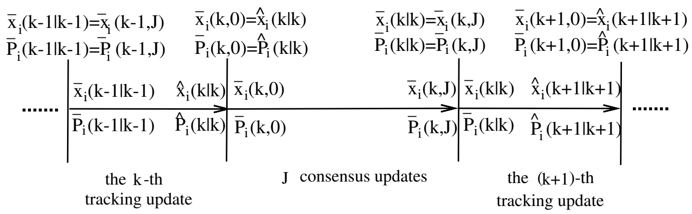

Fig. 2. Timing diagram of tracking and consensus updates in the proposed algorithm for distributed tracking with consensus.

Algorithm 1 Distributed 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 $j\le J-1$ **do**
      $\overline{\mathbf{x}}_i(k,j+1)=\overline{\mathbf{x}}_i(k,j)+\gamma(j)\sum_{l=1}^{n}A_{i,l}(j)(\mathbf{z}_{i,l}(k,j)-\mathbf{z}_{i,i}(k,j))$
      $j\leftarrow j+1$
    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

B. Conditions for achieving consensus

In this section, we summarize results on convergence of the proposed distributed tracking with consensus algorithm and its convergence rate. The proofs are omitted due to space limitations. For a fixed k and $J\gg 1$, the consensus update process can be considered as a consensus in estimation problem discussed in previous literature. In the following, to simplify the notation, we omit the tracking time step k in $\overline{\mathbf{X}}(k,j)$.

Define the consensus subspace $\mathcal{C}$ as, $\mathcal{C}=\{\mathbf{X}\in\mathbb{R}^{nN}|\mathbf{X}=\mathbf{1}_n\otimes\mathbf{a},\mathbf{a}\in\mathbb{R}^N\}$, meaning that if $\overline{\mathbf{X}}(j)\in\mathcal{C}$, then $\overline{\mathbf{x}}_i(j)=\overline{\mathbf{x}}_l(j)=\mathbf{a}$ for $1\le i,l\le n$.

Following two results characterize the convergence behavior of the proposed algorithm.

Theorem 1: Consider the consensus algorithm in (6) with initial state $\overline{\mathbf{X}}(0)\in\mathbb{R}^{nN}$. If the connectivity graph Laplacian $L(j)$ with mean $\overline{L}=\mathbb{E}[L(j)]$ is such that $\lambda_2(\overline{L})>0$, and if $p(l,i)>0$ for $(l,i)\in E(j)$, then there exists an almost sure finite real random vector $\mathbf{\Theta}$ such that

$$\mathbb{P}[\lim_{j\to\infty}\overline{\mathbf{X}}(j)=\mathbf{1}_n\otimes\mathbf{\Theta}]=1. \tag{10}$$

Proof: Omitted due to space constraints. ■

Theorem 2: Consider the consensus algorithm in (6) with initial state $\overline{\mathbf{X}}(0)\in\mathbb{R}^{nN}$. If the connectivity graph Laplacian $L(j)$ with mean $\overline{L}=\mathbb{E}[L(j)]$ is such that $\lambda_2(\overline{L})>0$, and if $p(l,i)>0$ for $(l,i)\in E(j)$, the convergence rate of the proposed consensus algorithm is bounded by $-\lambda_2(\overline{L})(\sum_{0\le j\le\infty}\gamma(j))$.

Proof: Omitted due to space constraints. ■

From the asymptotic unbiasedness of $\mathbf{\Theta}$, we have $\lim_{j\to\infty}\mathbb{E}[\overline{\mathbf{X}}(j)]=\mathbf{1}_n\otimes\mathbf{r}$, where $\mathbf{r}=\frac{1}{n}(\mathbf{1}^T\otimes I_N)\overline{\mathbf{X}}(0)$. Thus, for J large enough, at the end of the consensus update, all the node estimates $\overline{\mathbf{x}}_i(J)$ will converge to the average $\mathbf{r}$. Theorem 2 shows that the convergence rate depends on the topology through the algebraic connectivity $\lambda_2(\overline{L})$ and through the weights $\gamma(j)$, for $j\ge 0$. Thus, we can speed up the convergence process by optimizing these two parameters.

C. Steady-state analysis for noiseless time-varying graphs

In this part, we analyze the steady-state performance of the proposed distributed tracking with consensus algorithm. When the filter reaches steady-state, the error covariance matrix is time-invariant and the corresponding filter gain is constant. Therefore, finding the steady-state of the proposed algorithm will help understanding its asymptotic behavior, analyzing error covariance and filter design. From (9), it can be seen that the propagation of communication noise implies the non-existence of an upper bound to the covariance matrix. Therefore, the covariance matrix in Kalman filter may not also converge and the filter may not reach steady state. However, time-varying graph assumption does not affect the existence of steady-state. From the results of Section II-B, for noiseless time-varying graphs, average consensus is achieved over the network if the connectivity graph Laplacian $L(j)$ with mean $\overline{L}=\mathbb{E}[L(j)]$ is such that $\lambda_2(\overline{L})>0$, and if $p(l,i)>0$ for $(l,i)\in E(j)$. The outputs of the consensus update $\overline{\mathbf{X}}_i(k,J)$ and $\overline{P}(k,J)$ depend only on the inputs $\overline{\mathbf{X}}_i(k,0)$ and $\overline{P}(k,0)$. Hence, the combined system of distributed tracking with consensus can be transformed into a Kalman filter with time-invariant parameters. Therefore, steady-state can still be reached [10].

In the following, we assume a scalar target state $x \in \mathbb{R}^1$ and noiseless time-varying graphs, where the connectivity graph Laplacian $L(j)$ with mean $\overline{L} = \mathbb{E}[L(j)]$ is such that $\lambda_2(\overline{L}) > 0$, and $p(l,i) > 0$ for $(l,i) \in E(j)$. Note that, since a closed form equation for $\hat{P}_i(k+1|k)$ can not be easily obtained when the target state $x \in \mathbb{R}^M$ for $M > 1$, the following derivation would not apply to vector state. With these assumptions, the covariance matrix (9) in the j-th consensus iteration after the k-th tracking update simplifies to

$$\overline{P}(k, j+1) = \mathbf{A}(j)\overline{P}(k,j)\mathbf{A}(j)^T. \quad (11)$$

Since $\mathbf{1}^T L(j) = 0$ for $j \geq 0$, we have that $\mathbf{1}^T \overline{P}(k, j+1)\mathbf{1} = \mathbf{1}^T \overline{P}(k,j)\mathbf{1}$. Then, it follows that

$$\lim_{J\to\infty} \overline{P}(k,J) = \frac{\mathbf{1}^T\overline{P}(k,0)\mathbf{1}}{n^2}\mathbf{1}\mathbf{1}^T = \frac{\sum_{i=1}^n \hat{P}_i(k|k)}{n^2}\mathbf{1}\mathbf{1}^T. \quad (12)$$

From the result of Theorem 1 and the asymptotic unbiasedness of $\boldsymbol{\Theta}$, with noiseless communication assumption, it can be shown that for $1 \leq i \leq n$

$$\lim_{J\to\infty} \overline{x}_i(k,J) = \frac{1}{n}\sum_{i=1}^n \overline{x}_i(k,0) = \frac{1}{n}\sum_{i=1}^n \hat{x}_i(k|k). \quad (13)$$

By feeding the outputs of consensus update $\overline{x}_i(k,J)$ and $\overline{P}(k,J)$ back to the local Kalman filter of node i, from (4), (12) and (13), we have for $1 \leq i \leq n$

$$\hat{P}_i(k+1|k) = Q + \frac{1}{n^2}\sum_{q=1}^n F(I - K_q(k)H_q)\hat{P}_q(k|k-1)F^T,$$

$$\hat{x}_i(k+1|k) \quad (14)$$

$$= F\frac{1}{n}\sum_{q=1}^n \left[\hat{x}_q(k|k-1) + K_q(k)\left(y_q(k) - H_q\hat{x}_q(k|k-1)\right)\right].$$

From (14), we have $\hat{x}_i(k+1|k) = \hat{x}_l(k+1|k)$ and $\hat{P}_i(k+1|k) = \hat{P}_i(k+1|k)$ for $1 \leq i, l \leq n$. Let $\hat{x}_i(k+1|k) = \hat{x}(k+1|k)$ and $\hat{P}_i(k+1|k) = \hat{P}(k+1|k)$. Then, the combined system of distributed tracking with consensus can be transformed into a single Kalman filter as follows:

$$\begin{aligned}
\hat{x}(k+1|k) &= F\hat{x}(k|k-1) \\
&\quad + \frac{F}{n}\sum_{i=1}^n K_i(k)\left(y_i(k) - H_i\hat{x}_i(k|k-1)\right), \\
K_i(k) &= \hat{P}(k|k-1)H_i^T[H_i\hat{P}(k|k-1)H_i^T + R_i]^{-1}, \\
\hat{P}(k+1|k) &= Q + \frac{1}{n^2}\sum_{i=1}^n [F\hat{P}(k|k-1)F^T - FK_i(k) \\
&\quad \times \left(H_i\hat{P}(k|k-1)H_i^T + R_i\right)K_i(k)^T F^T]. \quad (15)
\end{aligned}$$

Theorem 3: Consider the system dynamics in (1) and the Kalman filter in (15). Assume that the connectivity graph Laplacian $L(j)$ with mean $\overline{L} = \mathbb{E}[L(j)]$ is such that $\lambda_2(\overline{L}) > 0$, and $p(l,i) > 0$ for $(l,i) \in E(j)$. If the pair (F, I) is controllable and the pair (F, H_i) is observable for $1 \leq i \leq n$, then the prediction covariance matrix $\hat{P}(k|k-1)$ converges to a constant matrix

$$\lim_{k\to\infty} \hat{P}(k|k-1) = P,$$

where P is the unique definite solution of the discrete algebraic Riccati equation (DARE)

$$P = Q + \frac{1}{n^2}\sum_{i=1}^n [FPF^T - FPH_i^T\left(H_iPH_i^T + R_i\right)^{-1}H_iPF^T]. \quad (16)$$

Proof: The proof is a generalization of the proof in [10]. ∎

Remark 1: As a consequence of Theorem 2, the filter gain converges to

$$\lim_{k\to\infty} K_i(k) = PH_i^T[H_iPH_i^T + R_i]^{-1}. \quad (17)$$

From (16), it can be seen that $\lim_{n\to\infty} P = Q$. i.e. as the size of the sensor network increases, the steady-state covariance matrix is reduced. This implies that if the network size is large enough, asymptotically the tracking is noiseless and follows the target exactly. For $H_i = H$ and $R_i = R$ for $1 \leq i \leq n$, we have $P = \frac{-B+\sqrt{B^2+4H^2QR}}{2H^2}$, where $B = (1 - \frac{F^2}{n})R - H^2Q$.

IV. Numerical Examples

In this section, we consider the performance of the distributed tracking with consensus algorithm and compare it with centralized Kalman filter and the distributed local Kalman filtering with centralized fusion. The performance of the centralized Kalman filter is well-understood [11] and provides a benchmark performance for distributed local Kalman filtering with centralized fusion. In local Kalman filtering with centralized fusion, all nodes send their filtered estimates to a fusion center. The fusion center then generates a fused estimate $\hat{\mathbf{x}}_{\text{fusion}}(k) = \frac{1}{n}\sum_{i=1}^n \hat{\mathbf{x}}_i(k|k)$.

The assumed parameters in the first simulation setup are as follows: $F = 1$, $Q = 1$, $x(0) = 0$, $P_0 = 0$, $R_i = 0.25$, $\Sigma_{l,i} = \Sigma = 0$, $n = 6$, $J = 30$, $H_i = 1$ for $i = 1, 3, 5$ and $H_i = 0.5$ for $i = 2, 4, 6$. The connectivity graph Laplacian $L(j) = \begin{cases} L_1 & j = 4m \\ L_2 & j = 4m+1 \\ L_3 & j = 4m+2 \\ L_4 & j = 4m+3 \end{cases}$ for $m = 0, 1, 2, \cdots$, which is described in Figure 3. As we can see, the graph is connected on average and $p(l,i) > 0$ for $(l,i) \in E(j)$. Thus, it satisfies the condition in Theorem 1 on the connectivity graph Laplacian.

Figure 4 shows the node consensus estimates $\overline{\mathbf{x}}_i(k,J)$ over a graph with noiseless communication links and switching topologies. It can be seen that all node estimates $\overline{\mathbf{x}}_i(k,J)$ converge to the same value and follow the target state, as asserted by Theorem 1. Figures 5 and 6 show the node estimates $\overline{\mathbf{x}}_i(k,j)$ in the consensus update after the 16-th tracking update and the variance of all the node estimates, respectively. From Figure 5, it can be seen that the node estimates converge to the average which is also confirmed in Figure 6, where the variance decreases as consensus iteration number increases and becomes static (around 10^{-8}) after consensus is reached. Figures 7 and 8 show the prediction covariance matrix $\hat{P}_i(k|k-1)$ and Kalman gain $K_i(k)$ of the filter in (15), respectively. It can be seen that as the Kalman filter reaches the steady state, both the prediction covariance matrix and Kalman gain converge, as asserted by Theorem 3. Note that the limit of the Kalman gain is different for different nodes in Figure 8 because the observation matrix H_i is different for each node.

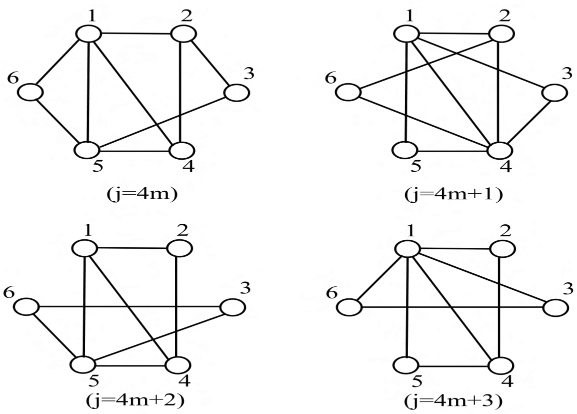

Fig. 3. A time-varying graph with switching topologies.

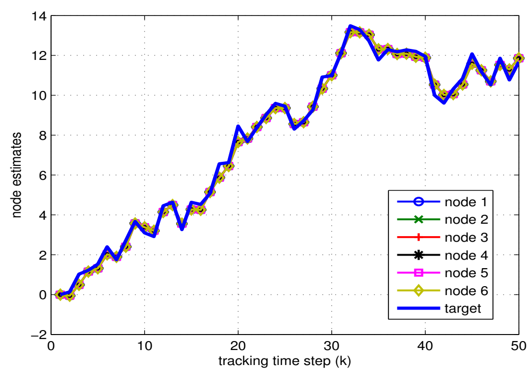

Fig. 4. Node consensus estimates $\bar{\mathbf{x}}_i(k, J)$ over a graph with switching topologies.

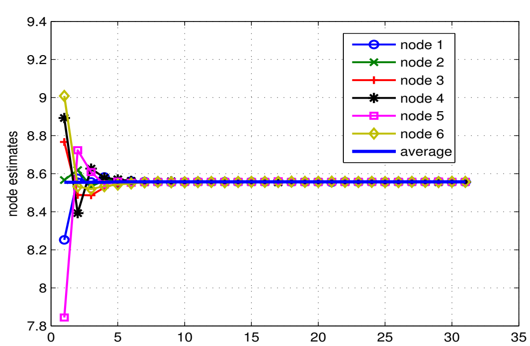

Fig. 5. Node estimates $\bar{\mathbf{x}}_i(k, j)$ in consensus update.

In the second simulation, we compare the performance of the three algorithms over a random graph with noisy communication links. 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: $F = 1$, $Q = 1$, $x(0) = 0$, $P_0 = 0$, $R_i = 0.25$, $H_i = 1$, $\Sigma_{l,i} = \Sigma = 0.1$ and $J = 30$.

Figure 9 shows the node estimates of the three algorithms in a time-varying graph with noisy communication links.

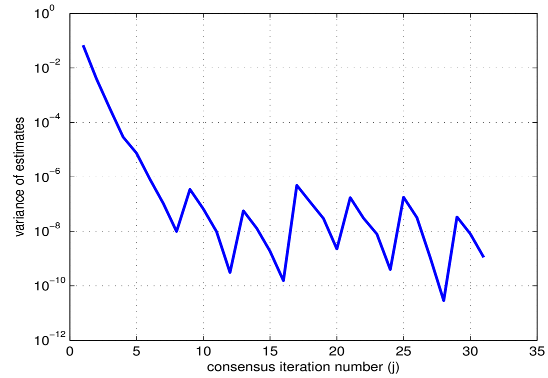

Fig. 6. Variance of all the node estimates $\bar{\mathbf{x}}_i(k, j)$ in consensus update.

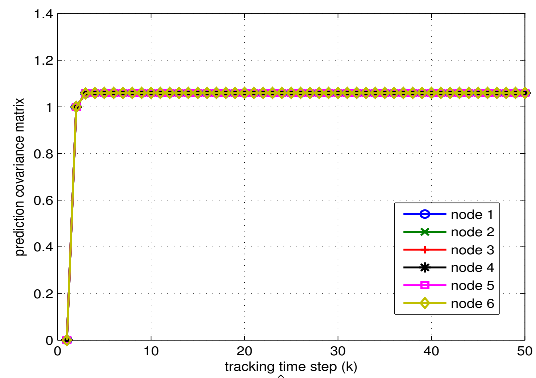

Fig. 7. Prediction covariance matrix $\hat{P}_i(k|k-1)$ versus tracking time step.

As we can see, the node estimates of the three algorithm follow the target's trajectory. In Figure 9, the curve with cross marker denotes the first node's estimate by using distributed tracking with consensus algorithm, the dashed curve denotes the distributed local Kalman filtering with centralized fusion, the curve with circle marker denotes the centralized Kalman filter and the solid curve denotes the target's trajectory. Figure 10 compares the resulting mean squared error (MSE) of the three algorithms, where the MSE of the distributed tracking with consensus is the average MSE of all the nodes $\frac{1}{n}\sum_{i=1}^{n}\left[\left(\bar{\mathbf{x}}_i(k, J) - \mathbf{x}(k)\right)^T\left(\bar{\mathbf{x}}_i(k, J) - \mathbf{x}(k)\right)\right]$. In Fig. 9, 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 Figures 9 and 10 show that the performance of the proposed distributed tracking with consensus algorithm is close to that of the centralized one in a time-varying graph with noisy communication. Since the proposed algorithm has advantages of robustness and scalability, it may be preferable in practical applications.

In the third simulation, we consider the two dimensional tracking problem in [5]. The connectivity graph is a random graph $G(n, p)$ with $n = 50$ and the probability that each link exists $p = 0.5$. The initial location of the target at (15,-10) in meter. The other parameters of the simulation setup are as follows: $F = I_2 + \epsilon F_0 + \frac{\epsilon^2}{2}F_0^2 + \frac{\epsilon^3}{6}F_0^3$, $F_0 = \begin{bmatrix} 0 & -2 \\ 2 & 0 \end{bmatrix}$,

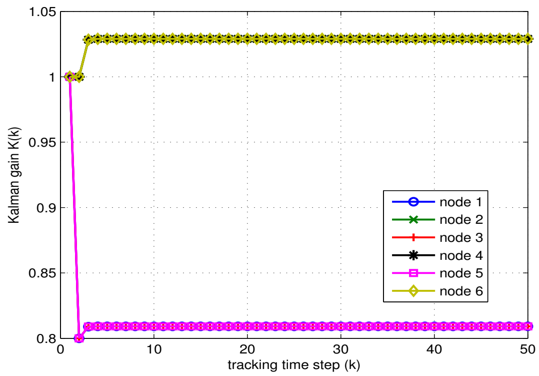

Fig. 8. Kalman gain $K_i(k)$ versus tracking time step.

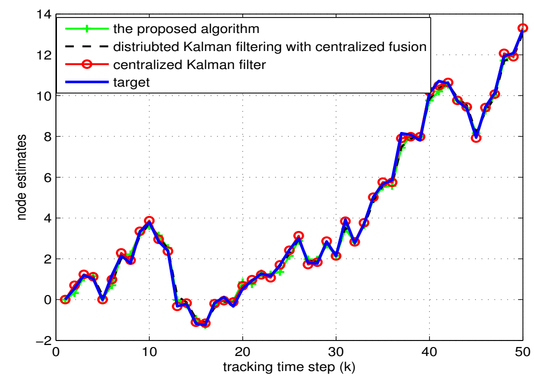

Fig. 9. Comparison of the proposed distributed tracking with consensus algorithm with distributed local Kalman filtering with centralized fusion and centralized Kalman filter (node estimates).

$\epsilon = 0.015$, $Q = (\epsilon c_w^2)^2 I_2$, $c_w = 5$, $x(0) = [15, -10]^T$ in meter, $H_i = [1, 0]$ 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 $c_v = 30$, $\Sigma_{l,i} = \Sigma = 0.01$, $J = 10$. The target is moving on noisy circular trajectories. The target is not fully observable by an individual node, but is collectively observable by all the nodes.

Figure 11 shows the node estimates (trajectory) of the proposed distributed tracking with consensus algorithms over a time-varying graph with noisy communication. As we can see, the distributed tracking consensus algorithm overcomes the impact of partial observation in each node and improves the overall observation quality. Note that the estimates are close to the trajectory of the target but with a small gap. That is because the observation noise covariance is rather large at each node. In all, the performance of the proposed algorithm can still be considered satisfactory.

V. Conclusion and Further Work

In this paper, we considered the problem of distributed tracking with consensus on a time-varying graph with noisy communications links. We developed a framework consisting of tracking and consensus updates to handle the time-varying topology and noisy communication issue. We discussed the conditions of consensus and analyzed the steady state behavior of the algorithm. Our simulation results show the distributed tracking with consensus algorithm improves the estimation quality of each node and its performance is close to distributed Kalman filtering with centralized fusion.

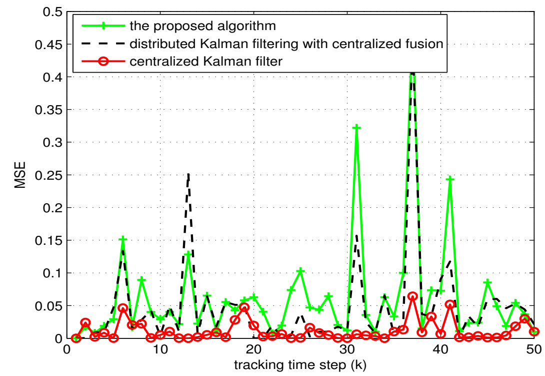

Fig. 10. Comparison of the proposed distributed tracking with consensus algorithm with distributed local Kalman filtering with centralized fusion and centralized Kalman filter (mean squared error).

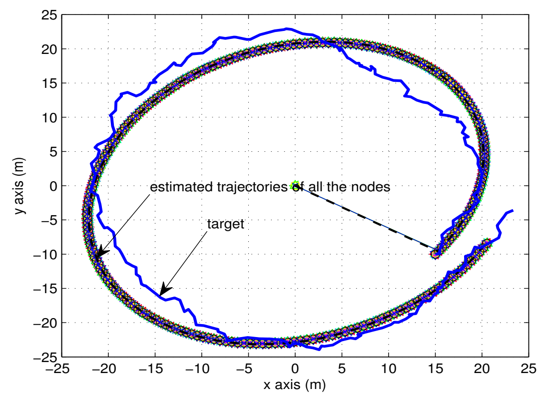

Fig. 11. The proposed distributed tracking with consensus algorithm on a two dimensional tracking problem (trajectory).

Acknowledgment

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

References

[1] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. Academic Press, 1988.

[2] D. Estrin, R. Govindan, J. Heidemann and S. Ku- mar. "Next century challenges: scalable coordination in sensor networks," Proc. of Mob. Comp. and Net., 1999.

[3] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of Groups of Mobile Autonomous. Agents Using Nearest Neighbor Rules,IEEE Transactions on Automatic Control, Vol. 48, No. 6, June 2003, pp. 988-1001.

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

[5] R. Olfati-Saber, "Distributed Kalman Filtering for Sensor Networks," Proc. of the 46th IEEE Conference on Decision and Control, pp. 5492 - 5498, Dec. 2007.

[6] 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.

[7] 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.

[8] 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, July 2009.

[9] 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.

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

[11] B. S. Y. Rao, H. F. Durrant-Whyte, and J. A. Sheen, "A fully decentralized multi-sensor system for tracking and surveillance," *Int. Journal of Robotics Research*, vol. 12, no. 1, pp. 20-44, Feb. 1993.