Distributed Kalman Filtering for Sensor Networks

R. Olfati-Saber

***Abstract*— In this paper, we introduce three novel distributed Kalman filtering (DKF) algorithms for sensor networks. The first algorithm is a modification of a previous DKF algorithm presented by the author in CDC-ECC '05. The previous algorithm was only applicable to sensors with identical observation matrices which meant the process had to be observable by every sensor. The modified DKF algorithm uses two identical consensus filters for fusion of the sensor data and covariance information and is applicable to sensor networks with different observation matrices. This enables the sensor network to act as a collective observer for the processes occurring in an environment. Then, we introduce a continuous-time distributed Kalman filter that uses local aggregation of the sensor data but attempts to reach a consensus on estimates with other nodes in the network. This peer-to-peer distributed estimation method gives rise to two iterative distributed Kalman filtering algorithms with different consensus strategies on estimates. Communication complexity and packet-loss issues are discussed. The performance and effectiveness of these distributed Kalman filtering algorithms are compared and demonstrated on a target tracking task.**

***Index Terms*— sensor networks, distributed Kalman filtering, consensus filtering, sensor fusion**

I. INTRODUCTION

Distributed estimation and tracking is one of the most fundamental collaborative information processing problems in wireless sensor networks (WSN). Multi-sensor fusion and tracking problems have a long history in signal processing, control theory, and robotics [1], [2], [3], [6], [16]. Moreover, estimation issues in wireless networks with packet-loss have been the center of much attention lately [18], [7].

Decentralized Kalman filtering [21], [15] involves state estimation using a set of local Kalman filters that communicate with all other nodes. The information flow is all-to-all with communication complexity of $O(n^2)$ which is not scalable for WSNs. Here, we focus on scalable or *distributed Kalman Filtering algorithms* in which each node only communicates messages with its neighbors on a network.

Control-theoretic consensus algorithms have proven to be effective tools for performing network-wide distributed computation tasks such as computing aggregate quantities and functions over networks [13], [12], [17], [8], [22], [23], [24], [20], [5]. These algorithms are closely related to gossip-based algorithms in computer science literature [9], [4].

Recently in [10], the author introduced a distributed Kalman Filtering (DKF) algorithm that uses dynamic consensus algorithms [14], [19]. The DKF algorithm consists of a network of micro-Kalman Filters (MKFs) each embedded with a low-pass and a band-pass consensus filter. The role of consensus filters is fusion of sensor and covariance data obtained at each node.

The existing DKF algorithm of the author suffers from a key weakness: the algorithm is only valid for sensors with identical sensing models. In other words, it is not applicable to heterogeneous multi-sensor fusion. To be more precise, let

$$z_i(k) = H_i(k)x(k) + v_i(k)$$

be the sensing model of node i in a sensor network. Here, $x(k)$ denotes the state of a dynamic process

$$x(k+1) = A_k x(k) + B_k w(k)$$

driven by zero-mean white Gaussian noise $w(k)$. Then, the DKF algorithm in [10] is applicable to sensors with identical H_i's. The reason is that under the assumption of identical observation matrices, or $H_i = H, \forall i$, we get

$$z_i(k) = H(k)x(k) + v_i(k) = s(k) + v_i(k)$$

and z_i's possess the structure necessary for the distributed averaging feature of the low-pass consensus filter in [14].

This limitation of the existing DKF algorithm motivates us to develop novel distributed Kalman filtering algorithms for sensor networks that have broader range of applications. In particular, we are interested in DKF algorithms capable of performing the following tracking and estimation tasks:

1) The H_i's are different across the entire network[1]and the process with state $x(k)$ is collectively observable by all the sensors.
2) Extended Kalman filtering (EKF) for sensors with different nonlinear sensing models

$$z_i(k) = h_i(x(k)) + v_i(k)$$

which after linearization leads to the case above (up to trivial terms) with H_i's being the Jacobian of the partial derivatives of $h_i(x)$ w.r.t. x.

We refer to each node of the distributed Kalman filter that provides a state estimate as a *Microfilter*. A DKF is a *Networked System* (or Swarm) of interacting microfilters with identical architectures. Each microfilter is computationally implemented as an embedded module in a sensor.

The main results of this paper are as follows: i) resolving the limitation of the existing DKF by introducing a novel microfilter architecture with identical high-pass consensus filters. The revised DKF algorithm is applicable to sensors with different H_i's, ii) presenting an alternative distributed Kalman filtering strategy which does not involve consensus

Reza Olfati-Saber is an Assistant Professor at Thayer School of Engineering, Dartmouth College, Hanover, NH 03755, email: olfati@dartmouth.edu.

[1]Some of the H_i's could be equal, but all of them are not the same.

filtering and instead uses consensus on state estimates, iii) introducing a continuous-time distributed Kalman filter, and iv) providing a base performance standard for distributed estimation and comparison between DKF algorithms on target tracking.

The outline of the paper is as follows: a DKF algorithm with identical consensus filters is introduced in Section II. Our main results including the DKF algorithms with consensus on state estimates are presented in Section III. Simulation results are given in Section IV. Finally, concluding remarks are made in Section V.

II. Distributed Kalman Filter: Type I: Consensus-based Fusion of Sensory Data

First, we present a modified version of the DKF algorithm in [10]. The key modification is to replace the low-pass and band-pass consensus filters in the architecture of the microfilter by high-gain versions of the high-pass consensus filter in [19]. The resulting microfilter is shown in Fig. 1. In the following, we provide the equations of the micro-Kalman filter (MKF) and the high-pass consensus filters (CFs).

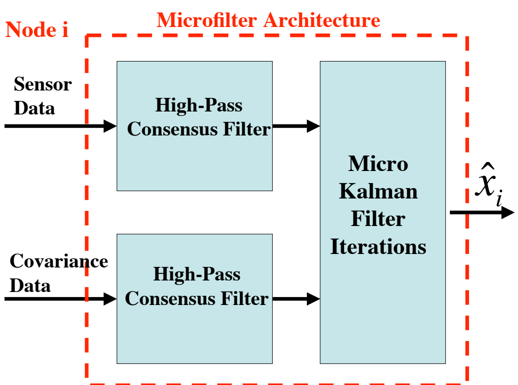


Fig. 1. The architecture of the microfilter of the type-I DKF algorithm.

Similar to the problem set up in [10], consider a sensor network with an ad hoc topology $G = (V, E)$ and n nodes. The graph G is undirected, $V = \{1, 2, \dots, n\}$, and $E \subset V \times V$. The objective is to perform distributed state estimation (or tracking) for a process/target that evolves according to

$$x(k+1) = A_k x(k) + B_k w(k); \quad x(0) \sim \mathcal{N}(\bar{x}(0), P_0). \quad (1)$$

The sensing model of the ith sensor is

$$z_i(k) = H_i(k)x(k) + v_i(k), \quad z_i \in \mathbb{R}^p \quad (2)$$

and we assume the H_i's are different (see Section I). Both w_k and v_k are zero-mean white Gaussian noise (WGN) and $x(0) \in \mathbb{R}^m$ is the initial state of the target. The statistics of the measurement noise is given by

$$E[w(k)w(l)^T] = Q(k)\delta_{kl}, \quad (3)$$
$$E[v_i(k)v_j(l)^T] = R_i(k)\delta_{kl}\delta_{ij}. \quad (4)$$

where $\delta_{kl} = 1$ if $k = l$, and $\delta_{kl} = 0$, otherwise.

Let $\mathbf{z}(k) = \text{col}(z_1(k), \dots, z_n(k)) \in \mathbb{R}^{np}$ be the collective sensor data of the entire sensor network at time k. Given the information $Z_k = \{\mathbf{z}(0), \mathbf{z}(1), \dots, \mathbf{z}(k)\}$, the estimates of the state of the process can be expressed as

$$\hat{x}_k = E(x_k|Z_k), \bar{x}_k = E(x_k|Z_{k-1}), \quad (5)$$
$$P_k = \Sigma_{k|k-1}, M_k = \Sigma_{k|k} \quad (6)$$

where $\Sigma_{k|k-1}$ and $\Sigma_{k|k}$ are the estimation error covariance matrices and $\Sigma_{0|-1} = P_0$. Defining an output matrix $H = \text{col}(H_1, H_2, \dots, H_n)$, one can define a *central estimate* $\hat{x}(k)$ associated with the data $\mathbf{z}(k)$ given by

$$\hat{x}(k) = \bar{x}(k) + K_k(\mathbf{z}(k) - H\bar{x}(k)). \quad (7)$$

Assuming that the measurement noise of the sensors are uncorrelated, the covariance matrix of the noise $v = \text{col}(v_1, \dots, v_n)$ is $R = \text{diag}(R_1, \dots, R_n)$ (the time indices are dropped). Let us define two network-wide aggregate quantities: the fused inverse-covariance matrices

$$S(k) = \frac{1}{n}\sum_{i=1}^{n} H_i'^T(k)R_i^{-1}(k)H_i(k) \quad (8)$$

and the fused sensor data

$$y(k) = \frac{1}{n}\sum_{i=1}^{n} H_i^T(k)R_i^{-1}(k)z_i(k). \quad (9)$$

Theorem 1. *(Micro-KF Iterations, [10]) Suppose every node of the network applies the following iterations*

$$\begin{aligned} M_i(k) &= (P_i(k)^{-1} + S(k))^{-1}, \\ \hat{x}(k) &= \bar{x}(k) + M_i(k)[y(k) - S(k)\bar{x}(k)], \\ P_i(k+1) &= A_k M_i(k) A_k^T + B_k Q_i(k) B_k^T, \\ \bar{x}(k+1) &= A_k \hat{x}(k). \end{aligned} \quad (10)$$

where $Q_i(k) = nQ(k)$ and $P_i(0) = nP_0$. Then, the local and central state estimates for all nodes are the same, i.e. $\hat{x}_i(k) = \hat{x}(k)$ for all i.

The above theorem holds regardless of how the networked-wide fusion task necessary to compute $y(k)$ and $S(k)$ is performed. Now, if one can (approximately) compute the averages $y(k)$ and $S(k)$, a distributed Kalman filtering algorithm emerges. We propose to use a high-gain version of the dynamic consensus algorithm of Spanos *et al.* [19] to perform distributed averaging—hence, the microfilter architecture in Fig. 1.

Let $N_i = \{j : (i, j) \in E\}$ be the set of neighbors of node i on graph G. Moreover, let $L = D - \mathcal{A}$ be the Laplacian matrix of G and $\lambda_2 = \lambda_2(L)$ denote its algebraic connectivity. The high-pass consensus filter is a linear system in the form

$$\begin{cases} \dot{q}_i = \beta \sum_{j \in N_i}(q_j - q_i) + \beta \sum_{j \in N_i}(u_j - u_i); \ \beta > 0 \\ y_i = q_i + u_i \end{cases} \quad (11)$$

where u_i is the input of node i, q_i is the state of the consensus filter, and y_i is its output. The gain $\beta > 0$ is relatively large ($\beta \sim O(1/\lambda_2)$) for randomly generated ad hoc topologies that are rather sparse. The collective dynamics of this CF is given by

$$\begin{cases} \dot{q} = -\beta \hat{L} q - \beta \hat{L} u \\ p = q + u \end{cases} \tag{12}$$

where $\hat{L} = L \otimes I_m$ is the m-dimensional graph Laplacian. *For a connected network, $p_i(t)$ asymptotically converges to* $1/n \sum_i u_i(t)$ as $t \to \infty$ (see [19]). Expressing this filter in discrete-time is trivial (see [12] for hints on the right choice of the step-size).

Remark 1. In [19], the equation of the dynamic consensus algorithm is given as $\dot{p} = -Lp + \dot{u}$ which reduces to (12) for $m = 1$ by defining $q = p - u$ and assuming the graph is weighted with weights in $\{0, \beta\}$.

Algorithm 1 Distributed Kalman Filtering Algorithm with high-pass consensus filtering of the sensed data.

Initialization: $q_i = 0, X_i = 0_{m \times m}, P_i = nP_0, \bar{x}_i = x(0)$

while new data exists **do**

Update the state of the data CF:

$$u_j = H_j^T R_j^{-1} z_j, \ \forall j \in N_i \cup \{i\}$$
$$q_i \leftarrow q_i + \epsilon\beta \sum_{j \in N_i} [(q_j - q_i) + (u_j - u_i)]$$
$$y_i = q_i + u_i$$

Update the state of the covariance CF:

$$U_j = H_j^T R_j^{-1} H_j, \ \forall j \in N_i \cup \{i\}$$
$$X_i \leftarrow X_i + \epsilon\beta \sum_{j \in N_i} [(X_j - X_i) + (U_j - U_i)]$$
$$S_i = X_i + U_i$$

Estimate the target state using Micro-KF:

$$M_i = (P_i^{-1} + S_i)^{-1}$$
$$\hat{x}_i = \bar{x}_i + M_i(y_i - S_i \bar{x}_i)$$

Update the state of the Micro-KF:

$$P_i \leftarrow A M_i A^T + nBQB^T$$
$$\bar{x}_i \leftarrow A \hat{x}_i$$

end while

To compute $y(k)$ and $S(k)$ in a distributed way, we use the discrete-time versions of both consensus filters. Each node uses the inputs $u_i(k) = H_i^T(k) R_i^{-1}(k) z_i(k)$ and $U_i(k) = H_i^T(k) R_i^{-1}(k) H_i(k)$ with zero initial states $q_i(0) = 0$ and $X_i(0) = 0$. The outputs $y_i(k)$ and $S_i(k)$ of the high-pass consensus filters asymptotically converge to $y(k)$ and $S(k)$ (up to minor conditions given in [19]). Algorithm 1 is the new type-I DKF algorithm with identical consensus filters.

According to Algorithm 1, node i sends the *message*

$$\text{msg}_i = (q_i(k), X_i(k), u_i, U_i)$$

to all of its neighbors. The message consists of the state and input of its consensus filters. This communication scheme is fully compatible with *packet-based communication in broadcast mode* in real-world wireless sensor networks. The information in each message can be contained in one or multiple packets (in case the message size is large). The message size is $O(m(m+1))$ with m being the dimension of the state x of the process/target. *Packet-loss* can be treated as loss of a communication link. Under mild connectivity conditions, packet-loss will not affect consensus algorithms.

III. Distributed Kalman Filter: Type II: Consensus on Estimates

In this section, we discuss an alternative approach to distributed Kalman filtering that relies on communicating state estimates between neighboring nodes. We refer to this second class of the DKF algorithms as type-II algorithms. Before presenting the type-II DKF algorithms, we first need to discuss a more primitive DKF algorithm that involves local Kalman filtering and forms the basis of our main results.

A. Local Kalman Filtering

Assume that each node i of the sensor network can communicate its measurement z_i, covariance information R_i, and output matrix H_i with its neighbors N_i. Should one avoid using any form of consensus (i.e. without further information exchange regarding states/estimates), *what is the optimal state estimate by each node?* The answer to this question is rather simple. In fact, one can use local Kalman filtering. The resulting algorithm can act as a base (or minimum) performance standard for any current (or future) DKF algorithms for sensor networks. Intuitively, local Kalman filtering (LKF) does not perform well for a minority of nodes and their neighbors that make relatively poor observations due to environmental or geometric conditions.

In local Kalman filtering, node i can assume that no nodes other than its neighbors N_i exist as the information flow from non-neighboring nodes to node i is prohibited in this case. Therefore, node i can use a central Kalman filter that only utilizes the observations and output matrices of the nodes in $J_i = N_i \cup \{i\}$. This leads to the following primitive DKF algorithm with no consensus on data/states/estimates.

Proposition 1. *(LKF Iterations) Let $N_i^c = V \setminus J_i$ be the set of non-neighboring nodes of node i. Moreover, assume node i receives no information from its non-neighbors. Then, the local Kalman filtering iterations for node i are in the form*

$$\begin{aligned} S_i(k) &= \sum_{j \in J_i} H_j^T(k) R_j^{-1}(k) H_j(k), \\ y_i(k) &= \sum_{j \in J_i} H_j^T(k) R_j^{-1}(k) z_j(k), \\ M_i(k) &= (P_i(k)^{-1} + S_i(k))^{-1}, \\ \hat{x}_i(k) &= \bar{x}_i(k) + M_i(k)[y_i(k) - S_i(k)\bar{x}_i(k)], \\ P_i(k+1) &= A_k M_i(k) A_k^T + B_k Q(k) B_k^T, \\ \bar{x}_i(k+1) &= A_k \hat{x}_i(k). \end{aligned} \tag{13}$$

where node i locally computes $y_i(k)$ and $S_i(k)$.

Proof: The proof follows from algebraic manipulation of the equations of the information form of the Kalman filter with restricted information to the observations of the inclusive neighbors J_i of node i. □

According to this LKF algorithm, there is no guarantee that the state estimates remain *cohesive* (or close to each other). This form of group disagreement regarding the state estimates is highly undesirable for a peer-to-peer network of estimators.

Let $\hat{L} = L \otimes I_m$ be the m-dimensional Laplacian of G. The *disagreement potential* [13] of the state estimates is defined as

$$\Psi_G(\hat{\mathbf{x}}) = \hat{\mathbf{x}}^T \hat{L} \hat{\mathbf{x}} = \frac{1}{2} \sum_{(i,j) \in E} \|\hat{x}_j - \hat{x}_i\|^2$$

where $\hat{\mathbf{x}} = \mathrm{col}(\hat{x}_1, \ldots, \hat{x}_n)$.

Algorithm 2 is a type-II DKF algorithm that attempts to reduce the disagreement regarding the state estimates in local Kalman filtering using an *ad hoc* approach by implementing a consensus step right after the estimation step. Later, in Algorithm 3, we provide a rigorous way of performing this "consensus on estimates."

Algorithm 2 Distributed Kalman Filtering Algorithm with an ad hoc consensus step on estimates.

Initialization: $P_i = P_0, \xi_i = x(0)$

while new data exists **do**

  Locally aggregate data and covariance matrices:

$$\begin{aligned} & J_i = N_i \cup \{i\} \\ & u_j = H_j^T R_j^{-1} z_j,\ \forall j \in J_i,\ y_i = \sum_{j \in J_i} u_j \\ & U_j = H_j^T R_j^{-1} H_j,\ \forall j \in J_i,\ S_i = \sum_{j \in J_i} U_j \end{aligned}$$

  Compute the intermediate Kalman estimate of the target state:

$$\begin{aligned} M_i &= (P_i^{-1} + S_i)^{-1} \\ \varphi_i &= \xi_i + M_i(y_i - S_i \xi_i) \end{aligned}$$

  Estimate the target state after a Consensus step:

$$\hat{x}_i = \varphi_i + \epsilon \sum_{j \in N_i} (\varphi_j - \varphi_i)$$

  {This is equivalent to "moving towards the average intermediate estimate of the neighbors."}

  Update the state of the local Kalman filter:

$$\begin{aligned} P_i &\leftarrow A M_i A^T + BQB^T \\ \xi_i &\leftarrow A \hat{x}_i \end{aligned}$$

end while

Let us refer to φ_i as the *intermediate estimate* of the sate of the target. Each node sends the following message

$$\mathrm{msg}_i = (u_i, U_i, \varphi_i)$$

to its neighbors.

In the following, we provide a more rigorous derivation of a type-II DKF algorithm that aims at reducing *disagreement of estimates* due to local Kalman filtering over the set of inclusive neighbors J_i. Our algorithm design method relies on solving the problem in continuous-time and then "inferring" the discrete-time version of the distributed Kalman filter with cohesive estimates.

B. Continuous-Time Distributed Kalman Filter

Consider a continuous-time (CT) linear system representing a target

$$\dot{x} = A(t)x + B(t)w$$

and a sensing model

$$z = H(t)x + v.$$

The noise statistics is rather analogous to the discrete-time problem setup. The continuous-time Kalman filter is in the form[2]:

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + K(z - H\hat{x}) \\ K &= PH^T R^{-1} \\ \dot{P} &= AP + PA^T + BQB^T - PH^T R^{-1} HP \end{aligned} \tag{14}$$

The following lemma is the key in derivation of the DKF in continuous-time:

Lemma 1. *Let $\eta = x - \hat{x}$ denote the estimation error of the Kalman filter in continuous-time. The estimation error dynamics is in the form*

$$\dot{\eta} = (A - KH)\eta + w_e \tag{15}$$

with the input noise $w_e = Bw + Kv$. The error dynamics without noise is a stable linear system with a Lyapunov function

$$V(\eta) = \eta^T P(t)^{-1} \eta \tag{16}$$

Proof: By direct differentiation, we have

$$\begin{aligned} \dot{V} &= \dot{\eta}^T P^{-1} \eta + \eta^T P^{-1} \dot{\eta} - \eta^T P^{-1} \dot{P} P^{-1} \eta \\ &= \eta^T [(A - KH)^T P^{-1} + P^{-1}(A - KH)]\eta \\ &\quad - \eta^T [P^{-1} A + A^T P^{-1} + P^{-1} BQB^T P^{-1} - H^T R^{-1} H]\eta \\ &= -\eta^T [H^T R^{-1} H + P^{-1} BQB^T P^{-1}]\eta < 0 \end{aligned}$$

for all $\eta \neq 0$. Thus, $\eta = 0$ is globally asymptotically stable and $V(\eta)$ is a valid Lyapunov function for the error dynamics. □

Consider a network with n sensors with the following sensing model:

$$z_i(t) = H_i(t)x + v_i \tag{17}$$

with $E[v_i(t) v_i^T(s)] = R_i \delta(t - s)$. Assume the pair (A, H) with $H = \mathrm{col}(H_1, \ldots, H_n)$ is observable. Here is our main result:

[2]For simplicity of notations, the time-dependence of many time-varying matrices is dropped.

Proposition 2. *(Kalman-Consensus filter) Consider a sensor network with continuous-time sensing model in* (17)*. Suppose each node applies the following distributed estimation algorithm*

$$\begin{aligned} \dot{\hat{x}}_i &= A\hat{x}_i + K_i(z_i - H_i\hat{x}_i) + \gamma P_i \sum_{j\in N_i}(\hat{x}_j - \hat{x}_i) \\ K_i &= P_i H^T R_i^{-1},\ \gamma > 0 \\ \dot{P}_i &= AP_i + P_i A^T + BQB^T - K_i R_i K_i^T \end{aligned} \tag{18}$$

with a Kalman-Consensus estimator and initial conditions $P_i(0) = P_0$ *and* $\hat{x}_i(0) = x(0)$*. Then, the collective dynamics of the estimation errors* $\eta_i = x - \hat{x}_i$ *(without noise) is a stable linear system with a Lyapunov function* $V(\eta) = \sum_{i=1}^n \eta_i^T P_i^{-1}\eta_i$*. Furthermore,* $\dot{V} \le -2\Psi_G(\eta) \le 0$ *and asymptotically all estimators agree* $\hat{x}_1 = \cdots = \hat{x}_n = x$*.*

Proof: Define vectors

$$\eta = \mathrm{col}(\eta_1,\ldots,\eta_n), \mathbf{x} = \mathrm{col}(x,\ldots,x) \in \mathbb{R}^{mn}$$

and let $\hat{\mathbf{x}} = \mathrm{col}(\hat{x}_1,\ldots,\hat{x}_n)$ be the vector of all the estimates of the nodes. Note that $\hat{x}_j - \hat{x}_i = \eta_i - \eta_j$ and thus, the estimator dynamics can be written as

$$\dot{\hat{x}}_i = A\hat{x}_i + K_i(z_i - H_i\hat{x}_i)) - \gamma P_i \sum_{j\in N_i}(\eta_j - \eta_i)$$

which gives the following error dynamics for the ith node

$$\dot{\eta}_i = (A - K_iH_i)\eta_i + \gamma P_i \sum_{j\in N_i}(\eta_j - \eta_i)$$

or

$$\dot{\eta}_i = F_i\eta_i + \gamma P_i \sum_{j\in N_i}(\eta_j - \eta_i)$$

with $F_i = A - K_iH_i$. By calculating $\dot{V}(\eta)$, we get

$$\dot{V} = \sum_i \eta_i^T P_i^{-1}\dot{\eta}_i + \dot{\eta}_i^T P_i^{-1}\eta_i - \eta_i^T P_i^{-1}\dot{P}_i P_i^{-1}\eta_i.$$

But

$$\eta_i^T P_i^{-1}\dot{\eta}_i = \eta_i^T P_i^{-1} F_i\eta_i + \eta_i^T \sum_{j\in N_i}(\eta_j - \eta_i)$$

and after transposition, the last term remains the same. Hence

$$\dot{\eta}_i^T P_i^{-1}\eta_i = \eta_i^T F_i^T P_i^{-1}\eta_i + \eta_i^T \sum_{j\in N_i}(\eta_j - \eta_i)$$

Note that the evolution of P_i is the same as the one for a standard Kalman filter (only the estimator is modified). Adding the three terms in $\dot{V}(\eta)$ and using Lemma 1 gives

$$\begin{aligned} \dot{V} &= -\sum_i \eta_i^T[H_i^T R_i^{-1} H_i^T + P_i^{-1}BQB^T P^{-1}]\eta_i \\ &\quad + 2\sum_i \sum_{j\in N_i} \eta_i^T(\eta_j - \eta_i) \end{aligned}$$

For undirected graphs, the second term is the negative of the disagreement function $-\Psi_G(\eta)$ in consensus theory [13]. Thus

$$\dot{V}(\eta) = -\eta^T\Lambda\eta - 2\eta^T\hat{L}\eta \le -2\Psi_g(\eta) \le 0 \tag{19}$$

where Λ is a positive definite block-diagonal matrix with diagonal blocks $H_i^T R_i^{-1} H_i^T + P_i^{-1}BQB^T P^{-1}$. Due to the fact that $\dot{V}(\eta) = 0$ implies all η_i's are equal and $\eta = 0$, asymptotically $\hat{x}_i = x, \forall i$ as $t \to \infty$. □

Remark 2. All H_i's in Proposition 2 are, in fact, $H_{i,l}$'s (l means "local") and $H_{i,l} = \mathrm{col}\{H_j\}_{j\in J_i}$ meaning that the aggregate observation of all sensors in J_i are used as z_i. Interestingly, H_i can also be the observation matrix of node i and the result will still hold.

C. Iterative Kalman-Consensus Filter

From the continuous-time DKF algorithm in Proposition 2, Algorithm 3 can be inferred. This is our main type-II distributed Kalman filtering algorithm.

Algorithm 3 *Kalman-Consensus filter:* DKF Algorithm with an estimator that has a rigorously derived consensus term.

Initialization: $P_i = P_0, \bar{x}_i = x(0)$
while new data exists **do**
  Locally aggregate data and covariance matrices:

$$\begin{aligned} J_i &= N_i \cup \{i\} \\ u_j &= H_j^T R_j^{-1} z_j,\ \forall j \in J_i,\ y_i = \sum_{j\in J_i} u_j \\ U_j &= H_j^T R_j^{-1} H_j,\ \forall j \in J_i,\ S_i = \sum_{j\in J_i} U_j \end{aligned}$$

  Compute the Kalman-Consensus estimate:

$$\begin{aligned} M_i &= (P_i^{-1} + S_i)^{-1} \\ \hat{x}_i &= \bar{x}_i + M_i(y_i - S_i\bar{x}_i) + \epsilon M_i \sum_{j\in N_i}(\bar{x}_j - \bar{x}_i) \end{aligned}$$

  Update the state of the Kalman-Consensus filter:

$$\begin{aligned} P_i &\leftarrow AM_iA^T + BQB^T \\ \bar{x}_i &\leftarrow A\hat{x}_i \end{aligned}$$

end while

In Algorithm 3, the message broadcasted to all the neighbors by node i is

$$\mathrm{msg}_i = (u_i, U_i, \bar{x}_i)$$

and the nodes communicate their predictions $\bar{x}_i$'s as well as their sensed data. In simulation results, we will see that Algorithm 3 has the best performance on a tracking task. Algorithms 2 and 3 can be effectively applied to mobile sensor networks by assuming the time-dependence of the set of neighbors $N_i(t)$ as in [11].

IV. Simulation Results

Consider a target with dynamics

$$\dot{x} = A_0x + B_0w$$

with

$$A_0 = 2\begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

and $B_0 = c_w^2 I_2$ with $c_w = 5$ (i.e. a point moving on noisy circular trajectories). We use the discrete-time model

$$x(k+1) = Ax(k) + Bw(k)$$

of this target with parameters

$$A = I_2 + \epsilon A_0 + \frac{\epsilon^2}{2} A_0^2 + \frac{\epsilon^3}{6} A_0^3, B = \epsilon B_0.$$

The step-size is $\epsilon = 0.015$ (≈ 70 Hz). The initial conditions are

$$x_0 = (15, -10)^T, \ P_0 = 10 I_2.$$

A sensor network with randomly located nodes is used in this experiment (see Fig. 2). The nodes make noisy measurement of the position of the target either along the x-axis, or along the y-axis, i.e.

$$z_i = H_i x + v_i$$

where either $H_i = H_x = (1, 0)$, or $H_i = H_y = (0, 1)$. Moreover, $R_i = c_v^2 \sqrt{i}$ for $i = 1, 2, \dots, 50$ with $c_v = 30$. Clearly, the target is not observable by individual sensors, but is observable by all the sensors. Furthermore, we assume that each set of inclusive neighbors J_i of node i contains nodes with observation matrices H_x and H_y.

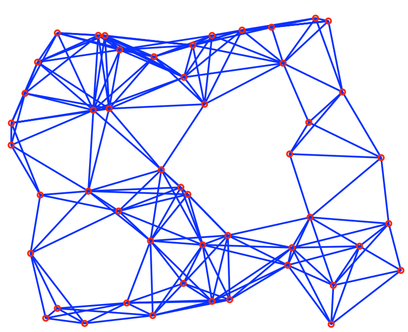
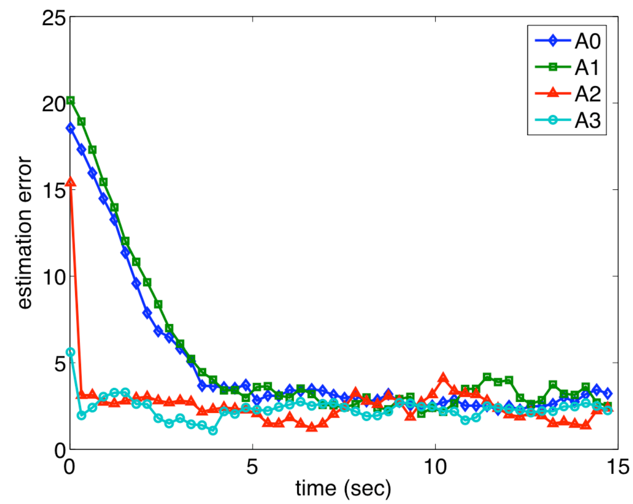


Fig. 2. A sensor network with 50 nodes and 242 links. Half of the nodes sense along the x-axis and the other half sense along the y-axis.

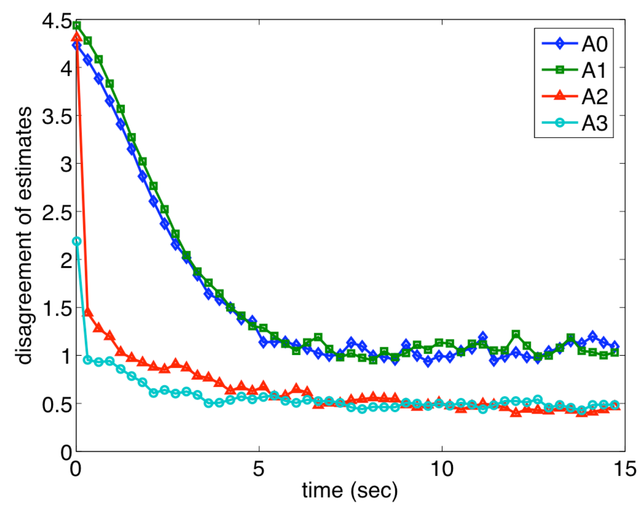
Let us refer to local Kalman filtering with no exchange of state or estimates as *Algorithm 0* (or A0). We compare four algorithms A0, A1, A2, and A3. For Algorithm 1, we set $\beta = 7$. Larger values of β force a smaller step-size ϵ, or a higher rate of information exchange in the network. This is a critical weakness of the high-pass consensus filter and thus Algorithm 1.

To measure the disagreement of the estimates independent of the network topology, we use the following measure $\|\delta\| = (\sum_{i=1}^n \delta_i^2)^{1/2}$ with $\delta_i = \hat{x}_i - \mu$ and $\mu = \frac{1}{n} \sum_i \hat{x}_i$. One can also utilize $\Psi_g(\hat{\mathbf{x}})$ instead.

Fig. 3 demonstrates the comparison of the four distributed Kalman filtering algorithms. A quick look at Fig. 3 reveals that the pairs of algorithms (A0,A1) and (A2,A3) behave in a similar manner (have comparable performances). Furthermore, both A2 and A3 perform significantly better than A0 and A1. Finally, A3 performs better than all other algorithms on this target tracking task.



(a)



(b)

Fig. 3. Comparison of the performance of distributed Kalman filtering algorithms: (a) average estimation error per node and (b) disagreement of the estimates $\|\delta\|$. Each curve is determined by averaging over 10 random runs of each algorithm).

Estimation error by itself is no longer an important measure of performance in distributed estimation in sensor networks. In a peer-to-peer estimation architecture, no particular fusion centers exist and every node is supposed to know the estimate of the state. A relatively high disagreement in the estimates of the different nodes goes against the purpose of performing distributed estimation without leaders. Therefore, only Kalman-Consensus filters (A2,A3) perform in a satisfactory manner.

This result demonstrates the importance of having a base standard (A0) that can be employed to judge the usefulness of more complex distributed estimation algorithms that involve further information exchange in terms of states/estimates.

Fig. 4 compares the estimates of all nodes by A1 vs. A3. Clearly, A3 has a superior performance and provides cohesive estimates. The estimates of A1 are somewhat dispersed for a relatively long period.

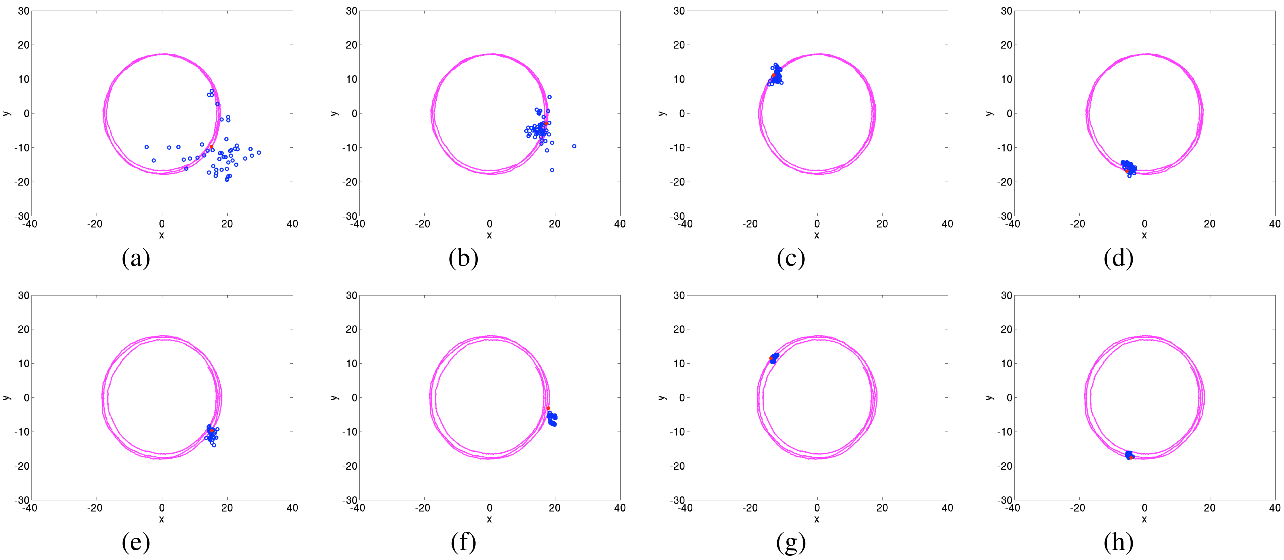


Fig. 4. Comparison between the estimates of all nodes: (a)-(d) Algorithm 1 and (e)-(h) Algorithm 3

V. Conclusions

Three novel distributed Kalman filtering algorithms were introduced. Algorithm 1 is a modification of a previous DKF algorithm [10]. A continuous-time DKF algorithm is rigorously derived and analyzed. Two Kalman-Consensus filtering algorithms in discrete-time were inspired by this continuous-time DKF algorithm. The objective in type-II DKF algorithms is to reduce disagreement of the estimates by different nodes. This led to the addition of a consensus term in the estimator of Algorithm 3 and an ad hoc consensus step in Algorithm 2. On a tracking task, both A2 and A3 perform better than A1 and a primitive local Kalman filtering approach. Algorithm 3 has the best overall performance.

Acknowledgment

Many thanks goes to Jeff Shamma for several useful discussions on distributed estimation and suggesting Lemma 1.

References

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

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

[3] Y. Bar-Shalom and X. R. Li. *Multitarget-Multisensor Tracking: Principles and Techniques*. YBS Publishing, Storrs, CT, 1995.

[4] S. Boyd, A. Ghosh, and D. Prabhakar, B. Shah. Gossip algorithms: design, analysis and applications. *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '05)*, pages 1653–1664, March 2005.

[5] J. Cortés. Distributed algorithms for reaching consensus on arbitrary functions. *Automatica (submitted)*, Oct. 2006.

[6] D. Fox, J. Hightower, L. Liao, D. Schulz, and G. Borriello. Bayesian filtering for Location estimation. *IEEE Pervasive Computing*, 2(3):24–33, July–Sept. 2003.

[7] V. Gupta, D. Spanos, B. Hassibi, and R. M. Murray. On LQG control across a stocjastic packet-dropping link. *Proc. of the 2005 Automatic Control Conference*, pages 360–365, June 2005.

[8] Y. Hatano and M. Mesbahi. Agreement over random networks. *IEEE Trans. on Automatic Control*, 50(11):1867–1872, Nov. 2005.

[9] D. Kempe, A. Dobra, and J. Gehrke. Gossip-based computation of aggregate information. *Proc. of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS '03)*, 8(3):482–491, 2003.

[10] R. Olfati-Saber. Distributed Kalman filter with embedded consensus filters. *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference (CDC-ECC '05)*, pages 8179–8184, Dec. 2005.

[11] R. Olfati-Saber. Distrbuted tracking for mobile sensor networks with information-driven mobility. *Proc. of the 2006 American Control Conference*, July 2007.

[12] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95, Jan. 2007.

[13] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. on Automatic Control*, 49(9):1520–1533, Sep. 2004.

[14] R. Olfati-Saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. *44th IEEE Conference on Decision and Control, 2005 and 2005 European Control Conference (CDC-ECC '05)*, pages 6698–6703, Dec. 2005.

[15] 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*, 12(1):20–44, Feb 1993.

[16] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Trans. on Automatic Control*, 24(6):843–854, Dec 1979.

[17] W. Ren and R. W. Beard. Consensus seeking in multiagent systems under dynamically changing interaction topologies. *IEEE Trans. on Automatic Control*, 50(5):655–661, 2005.

[18] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poola, M. I. Jordan, and S. S. Sastry. Kalman filtering with intermittent observations. *IEEE Trans. on Automatic Control*, 49(9):1453–1464, Sep. 2004.

[19] D. Spanos, R. Olfati-Saber, and R. M. Murray. Dynamic Consensus on Mobile Networks. *The 16th IFAC World Congress, Prague, Czech*, 2005.

[20] D. P. Spanos, R. Olfati-Saber, and R. M. Murray. Approximate distributed Kalman filtering in sensor networks with quantifiable performance. *Fourth International Symposium on Information Processing in Sensor Networks*, pages 133–139, April 2005.

[21] J. L. Speyer. Computation and transmission requirements for a decentralized linear-quadratic-gaussian control problem. *IEEE Trans. on Automatic Control*, 24(2):266–269, Feb 1979.

[22] J. N. Tsitsiklis. *Problems in Decentralized Decision Making and Computation*. PhD thesis, Department of Electrical Engineering and Computer Science, Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, Nov. 1982.

[23] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Trans. on Automatic Control*, 31(9):803–812, Sep. 1986.

[24] L. Xiao and S. Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 52:65–78, 2004.