Proceedings of the
47th IEEE Conference on Decision and Control
Cancun, Mexico, Dec. 9-11, 2008

Convergence Properties of a Decentralized Kalman Filter

Maryam Kamgarpour and Claire Tomlin

Abstract— **We consider the problem of decentralized Kalman filtering in a sensor network. Each sensor node implements a local Kalman filter based on its own measurements and the information exchanged with its neighbors. It combines the information received from other sensors through using a consensus filter as proposed in [14]. For a time-invariant process and measurement model, we show that this algorithm guarantees that the local estimates of the error covariance matrix converge to the centralized error covariance matrix and that the local estimates of the state converge in mean to the centralized Kalman filter estimates. However, due to the use of the consensus filter, the local estimates of the state do not converge to the least-squares estimate that would be obtained from a centralized Kalman filter.**

I. INTRODUCTION

A. Background and Problem Description

The problem of decentralized Kalman filtering refers to Kalman filtering in a sensor network in which there is no central data fusion and computation unit. For example, due to communication constraints in large sensor networks, it may be impossible for all sensors to send their observations to a central unit. In addition, having one unit responsible for data fusion and estimation will make the network fragile and may be computationally infeasible. Without having access to all sensors' observations and measurement models, one can only approximate a centralized Kalman filter. The problem is how to perform this decentralization so that the approximate Kalman filters result in estimates that are close to those obtained from a centralized Kalman filter.

In this work we consider a network of sensors making observations of a process and we require that each sensor maintains an estimate of the state of the process and the error covariance matrix. We assume that each sensor can exchange messages with its nearest neighbors. Since the sensors have computation and communication capabilities, we refer to them as a node or a sensor node interchangeably. An example of the need for the sensor nodes to maintain an estimate of the error covariance matrix in addition to the estimate of the state is target tracking. Here, the mobile sensors need to make motion control decisions based on their estimates of the target position and velocity [10], [17], [5], [9]. The error covariance matrix represents the amount of uncertainty in their estimated target state. They plan their motions in directions of decreasing this uncertainty. Using measurement information from all sensors would lead to maintaining a better estimate of the error covariance matrix at each node and a more cooperative motion.

The Kalman filter algorithm is widely used in estimation because it provides an optimal, in the mean-square-error sense, estimate of the state under assumptions of linearity of the process and observation model and additive Gaussian noise. A large class of systems may be approximated by a Gaussian linear model. Hence, this algorithm provides the basis for studying estimation in sensor networks and can provide insight for decentralized estimation in the presence of nonlinear or non-Gaussian models.

B. Previous Work and Contributions of the Paper

There have been many studies on different methods of implementing an exact or approximate decentralized Kalman filter. See, for example, [11], [1], [20], [21], [8], [12], [24], [22]. There are also methods developed for decentralization of the Kalman filter considering a specific sensor network application as in [19], [2], [13], [6], to name a few. The aforementioned studies differ based on the assumptions made on the interaction topology and the messages exchanged between the sensors, as well as the method for combining the messages that arrive at each node. If all-to-all communication exists between the sensors, it is possible to implement an exact Kalman filter at each node through appropriately defining messages that get broadcast from the nodes [18]. All-to-all communication is not a reasonable assumption for many large scale sensor networks. Many authors consider a communication topology in which each node exchanges information with its nearest neighbors. In such cases, previous research has studied exchanging the estimate mean and possibly the error covariance matrix found locally. Each node then updates its estimate using a weighted sum of all messages it has received from other nodes [2], [19], [22], [24]. One characteristic of this approach is that estimates from different nodes are not necessarily independent as they may contain the same process noise and measurement information. To optimally combine them, one needs to know the mutual information between them. Computing this quantity for a general communication topology is difficult [2]. Under certain assumptions on the communication topology, authors in [8] introduce a so-called Channel filter to find the mutual information between the observations.

The algorithm we consider has been proposed in [20]. Here, the information filter form of the Kalman filter is utilized to decompose the centralized Kalman filter equations based on the individual sensor contributions. Then, consensus algorithms are used to combine the messages from neighboring sensors in order to implement an approximate Kalman filter locally. The method can be applied at each node without it knowing the communication topology of the entire network. Different consensus algorithms appropriate for the decentralized Kalman filtering have been proposed and their stability properties have been analyzed. For example, authors in [17] have taken this approach and have developed consensus algorithms that are best suited for decentralized Kalman filtering [7]. However, the performance of the consensus-based decentralized Kalman filter has not been studied in enough detail. Considering a specific example, the authors in [14] show in simulations that both centralized and approximate decentralized Kalman filters provide almost identical estimates. Our contribution is to provide a theoretical result on the convergence of the local Kalman filters. We show that the local estimates of the error covariance matrix converge to the central error covariance matrix. The local estimates of state converge in mean to the central estimates. However, the error covariance matrices associated with the local estimates of the state do not correspond to the estimated error covariance matrix. These results are independent of the specific consensus algorithms used.

C. Organization of this Report

In section II we review the Kalman filter equations in their equivalent information filter form and pose the problem of decentralized Kalman filtering. In section III, following the approach of [14], we describe how by using consensus algorithms, each node can implement a decentralized Kalman filter. Section IV consists of our contribution where we derive conditions under which the local state estimates and error covariance matrices converge to those obtained from the centralized Kalman filter. We illustrate these results with some simulations in section V.

II. Review of Information Filter Equations

Consider a system whose dynamics can be described using the equations

$$\begin{aligned} x_{k+1} &= A_k x_k + w_k \\ z_k &= H_k x_k + n_k, \end{aligned} \tag{1}$$

where $x_k \in \mathbb{R}^n$, $A_k \in \mathbb{R}^{n\times n}$, $H_k \in \mathbb{R}^{M\times n}$. Here, $w_k \in \mathbb{R}^n$, $n_k \in \mathbb{R}^M$ are the process and measurement noises. We assume x_0, w_l, n_l, for $l = 0,1,\ldots k$ are independent random vectors with zero mean and positive definite covariance matrices $P_0 \in \mathbb{R}^{n\times n}$, $Q_l \in \mathbb{R}^{n\times n}$, $R_l \in \mathbb{R}^{M\times M}$ respectively. Let $x_{k|k-1} = E\{x_k|z_0,\ldots,z_{k-1}\}$ and $P_{k|k-1} = Cov(x_k - x_{k|k-1})$ denote the one-step prediction of the state and the prediction error covariance at time k. Similarly, let $x_{k|k} = E\{x_k|z_0,\ldots,z_k\}$ denote the the state estimate after receiving measurement at time k, and $P_{k|k} = Cov(x_k - x_{k|k})$ denote the updated error covariance matrix at time k. Define $Y_{k|k-1} = (P_{k|k-1})^{-1}$, $Y_{k|k} = (P_{k|k})^{-1}$, $y_{k|k-1} = Y_{k|k-1}x_{k|k-1}$, and $y_{k|k} = Y_{k|k}x_{k|k}$. The positive definiteness assumptions on Q_k and P_0 are sufficient to ensure that the above inverses exist. The Kalman filter equations can be written as two sets of recursive equations, the first set pertains to prediction:

$$\begin{aligned} P_{k|k-1} &= A_k P_{k-1|k-1} {A_k}^T + Q_k \\ x_{k|k-1} &= A_k x_{k-1|k-1}. \end{aligned} \tag{2}$$

The second set pertains to update and is most easily understood using the transformations defined above:

$$\begin{aligned} Y_{k|k} &= Y_{k|k-1} + {H_k}^T (R_k)^{-1} H_k \\ y_{k|k} &= y_{k|k-1} + {H_k}^T (R_k)^{-1} z_k. \end{aligned} \tag{3}$$

The matrix $Y_{k|k}$ and the vector $y_{k|k}$ are referred as the information matrix and the information vector respectively. This is because the contributions of the sensor observation at each time step are apparent in these quantities. The information filter equations (3) are utilized in scenarios where there are multiple sensors making measurements. Suppose there are M sensors with measurement model of sensor i given by[1]

$$z^i{}_k = H^i{}_k x_k + n^i{}_k, \tag{4}$$

where $n^i{}_l$ has zero mean and variance $R^i{}_l \in \mathbb{R}$, $l = 0,1,\ldots,k$. If the measurement noise of different sensors are independent, the update equations (3) can be written such that individual sensor contributions can be seen:

$$\begin{aligned} Y_{k|k} &= Y_{k|k-1} + \sum_{i=1}^{M} (H^i{}_k)^T (R^i{}_k)^{-1} H^i{}_k \\ y_{k|k} &= y_{k|k-1} + \sum_{i=1}^{M} (H^i{}_k)^T (R^i{}_k)^{-1} z^i{}_k. \end{aligned} \tag{5}$$

Clearly, if all-to-all communication between sensor nodes exists at all times, node i can broadcast $(H^i{}_k)^T (R^i{}_k)^{-1} H^i{}_k$ and $(H^i{}_k)^T (R^i{}_k)^{-1} z^i{}_k$ to all nodes. It can implement an exact Kalman filter locally by receiving other nodes' values of the above quantities. This was shown in [18]. In case all-to-all communication does not exist or is not a reasonable assumption, an approach in implementation of a decentralized Kalman filter is that each node would estimate the quantities $F_k = \sum_{i=1}^{M} (H^i{}_k)^T (R^i{}_k)^{-1} H^i{}_k$ and $f_k = \sum_{i=1}^{M} (H^i{}_k)^T (R^i{}_k)^{-1} z^i{}_k$, where we defined F_k and f_k for ease of notation. In order to perform such estimation, one needs to appropriately define the messages that the communicating nodes exchange. The resulting local Kalman filters would be approximations of the centralized one and the approximations improve as the local estimates of F_k, f_k approach the exact summation. Using this method to implement a decentralized Kalman filter, we have not done any prior local processing in terms of estimation of state. Consequently, we are not faced with the problem of accounting for mutual information in local estimates. In addition, the problem of estimation and data fusion have become decoupled; once an appropriate method for locally estimating the required summations is devised, the local estimation of state and the error covariance matrix can be updated exactly using equations (5).

Additive dependence of the update equations to the individual sensors' contributions in (5) motivates use of consensus algorithms to approximate the quantities F_k and f_k locally [20]. In the next section, we describe how a consensus algorithm can be used for estimating these quantities and describe the resulting decentralized Kalman filter algorithm.

[1] We use superscript i to denote sensor i's parameters.

III. Decentralized Kalman Filter with Consensus

The interaction topology of the sensor nodes represents the set of nodes that can exchange information at any given time. It can be represented as a directed graph (digraph) $G = (V, E)$, where V is the set of vertices denoting the sensor nodes, and $E \subseteq V \times V$ is the set of edges: $E = \{(i, j) | \ i \text{ communicates with } j\}$. $N_i = \{j \in V | (j,i) \in E\}$ is called the set of neighbors of node i. In the case that $(i, j) \in E$ implies $(j,i) \in E$ the graph is undirected. The graph is called connected if there is a path connecting any two nodes. Similarly, the digraph is called strongly connected if there is a directed path connecting any two arbitrary nodes of the graph.

By a consensus algorithm we mean a local update law that is implemented on each node. The algorithm is local in the sense that its inputs can only depend on the information available to a given node or its neighboring nodes. Asymptotically, it ensures that all nodes agree on a certain quantity of interest [15].[2] Let ν^i denote node i's estimate of a given quantity of interest and μ^i an external input to node i that can affect ν^i. An update law, g, that results in consensus on sum of (possibly changing) inputs is stated as

Dynamic Consensus

$$\begin{array}{l} \nu^i{}_{k+1} = g(\nu^i{}_k, \mu^i{}_k, \nu^j{}_k, \mu^j{}_k, \forall j \in N_i) \\ \text{such that } \nu^i{}_k \to \sum_{i=1}^{M} \mu^i{}_k, \forall i \in \{1, 2, \ldots, M\} \end{array} \tag{6}$$

The word *dynamic* consensus is used to distinguish from the more traditional consensus algorithms in which nodes agree on a constant value, such as average of their initial states. For the Kalman filter application, we consider a dynamic consensus algorithm that will guarantee the local estimates, $\nu^i{}_k$, asymptotically converge to the summation of the steady-state input values. If the inputs are time-varying, the consensus algorithm guarantees that the local estimates of the sum of the inputs will have a bounded error: $|\nu^i{}_k - \sum_{i=1}^{M} \mu^i{}_k| < \varepsilon, \varepsilon > 0$. The bound would depend on how fast the inputs are changing and on the specific algorithm used [7]. Consensus can be achieved given that certain connectivity conditions on the communication graph are satisfied. For example, a sufficient condition for consensus is that the graph is connected so that the messages from each node eventually get propagated through the entire network. There has been much previous work on how to design consensus algorithms such that consensus is reached quickly and the algorithm is robust with respect to time-delays [23], [16]. Low-pass, band-pass and high-pass consensus algorithms have also been developed [14], [7] and can be used in cases in which inputs contain noise.

Assume that we have chosen a desired g, the consensus algorithm, such that dynamic consensus on the sum of inputs as described above is achieved. The algorithm can be employed in decentralized Kalman filtering as follows: Consider node i using a consensus filter to estimate $F_k = \sum_{i=1}^{M} (H^i{}_k)^T (R^i{}_k)^{-1} H^i{}_k$. Each element of F_k is summation of the corresponding elements of the symmetric matrices $(H^i{}_k)^T (R^i{}_k)^{-1} H^i{}_k$. Let $\nu^i{}_k$ be sensor i's estimate of the l^{th} element of this matrix. Define the input $\mu^i{}_k$ to the consensus algorithm as the l^{th} element of $(H^i{}_k)^T (R^i{}_k)^{-1} H^i{}_k$. The consensus filter will guarantee that the local estimates of this element converge asymptotically to the l^{th} element of F_k. To estimate F_k a total of $\frac{n(n+1)}{2}$ consensus filters is implemented on each node. Similarly, n consensus filters are implemented to estimate $f_k = \sum_{i=1}^{M} (H^i{}_k)^T (R^i{}_k)^{-1} z^i{}_k$. Since the term $(H^i{}_k)^T (R^i{}_k)^{-1} z^i{}_k$ contains noise it is desirable to use a low-pass consensus filter for estimation of this quantity. We denote node i's estimate of F_K and f_k by $F^i{}_k$ and $f^i{}_k$ respectively. The decentralized Kalman filter algorithm is run on each node and is given by:

Algorithm 1 Decentralized Kalman Filter Algorithm

1: Initialize: $P^i{}_{0|0} = P_0$, $x^i{}_{0|0} = x_{0|0}$.
   $Y^i{}_{0|0} = P_0{}^{-1}$, $y^i{}_{0|0} = Y^i{}_{0|0} x^i{}_{0|0}$, $k = 1$.
2: **repeat**
3: Prediction: $P^i{}_{k|k-1} = A_k P^i{}_{k-1|k-1} A_k{}^T + Q_k$,
   $x^i{}_{k|k-1} = A_k x^i{}_{k-1|k-1}$.
   $Y^i{}_{k|k-1} = (P^i{}_{k|k-1})^{-1}$, $y^i{}_{k|k-1} = Y^i{}_{k|k-1} x^i{}_{k|k-1}$.
4: Observation: $z^i{}_k = H^i{}_k x_k + n^i{}_k$.
5: Consensus: estimate $\sum_{i=1}^{M} (H^i{}_k)^T (R^i{}_k)^{-1} H^i{}_k$,
   and $\sum_{i=1}^{M} (H^i{}_k)^T (R^i{}_k)^{-1} z^i{}_k$.
   Store results in $F^i{}_k$, $f^i{}_k$ respectively.
6: Update: $Y^i{}_{k|k} = Y^i{}_{k|k-1} + F^i{}_k$, $y^i{}_{k|k} = y^i{}_{k|k-1} + f^i{}_k$
   $P^i{}_{k|k} = (Y^i{}_{k|k})^{-1}$, $x^i{}_{k|k} = P^i{}_{k|k} y^i{}_{k|k}$.
7: $k = k + 1$.
8: **until** local estimates are needed.

In applications where communication is done at a faster rate than making observations, many consensus filter steps can be run between each observation. This will ensure local estimates of F_k and f_k converge to the true values more quickly. We note that for $Y^i{}_{k|k}$ in the above algorithm to be invertible it is sufficient to assume that the estimate, $F^i{}_k$, obtained from the consensus algorithm is positive semi-definite for all k. In practice, if this is not the case, one can approximate $F^i{}_k$ by a positive definite matrix $\tilde{F}^i_k$ by zeroing out its negative eigenvalues. In this case, if $F^i{}_k \to F$ where F is positive definite, then the positive definite approximations, $\tilde{F}^i_k$, also converges to F. Hence, this substitution does not change the steady-state convergence of the consensus filter.

A question that arises is whether the local error covariance matrices and the state estimates that are computed using the above algorithm converge to those of the centralized Kalman filter. In the next section, we address this question.

IV. Convergence studies of the Decentralized Kalman Filter

To study convergence properties of the decentralized Kalman filter we assume a time invariant process dynamics

[2] This reference contains a survey of consensus theory and applications.

and measurement model given by

$$x_{k+1} = Ax_k + w_k \qquad (7)$$
$$z^i{}_k = H^i x_k + n^i{}_k,$$

where the covariance matrix of the process noise is denoted by Q and the covariance matrix of sensor i's noise is denoted by R^i. Let $H^T = (H^{1^T}, \dots, H^{M^T})$, $R = \mathrm{diag}(R^1, \dots, R^M)$. For the convergence analysis to be followed, we assume that $M \geq n$ and (A,H) is observable. These conditions are not difficult in general to satisfy given that decentralized Kalman filter is usually applied in large sensor networks, where there are more sensors than the dimension of the state. For notational compactness let $P_k = P_{k|k-1}$ and $\bar{P}_k = P_{k|k}$. The prediction error covariance matrix given by the centralized Kalman filter satisfies the following recursive equation

$$P_{k+1} = A(P_k - P_k H^T (H P_k H^T + R)^{-1} H P_k) A^T + Q. \qquad (8)$$

It is known that if (A,H) is observable and (A,D) is controllable, where $Q = DD^T$, the above recursion converges to the unique positive definite fixed-point of the Riccati equation

$$P = A(P - PH^T (HPH^T + R)^{-1} HP) A^T + Q. \qquad (9)$$

Also, $\bar{P}_k \to \bar{P}$ where $\bar{P} = P - PH^T(HPH^T + R)^{-1}HP$ (see for example, [4], [3]).

In the decentralized Kalman filter algorithm defined in the previous section, at each time step, each node maintains an estimate of the information matrix $F = H^T R^{-1} H$ through using appropriate consensus filters. Node i's estimate at time k is denoted by $F^i{}_k$. As discussed in the previous section, without loss of generality we can assume that $F^i{}_k$ is positive semi-definite. The local information matrix, $Y^i{}_k = (P^i{}_{k|k-1})^{-1}$, satisfies the following recursive equation

$$Y^i{}_{k+1} = (A(Y^i{}_k + F^i{}_k)^{-1} A^T + Q)^{-1}. \qquad (10)$$

For the rest of the discussion, we drop the superscript i for simplicity in notation and study the convergence of local estimates of the covariance matrix for an arbitrary node. Let $S_k \in \mathbb{R}^{M \times n}$ be such that $F_k = S_k{}^T S_k$. Define $C_k = R^{\frac{1}{2}} S_k \in \mathbb{R}^{M \times n}$. From one of the matrix inversion lemmas, namely, $(V + U^T R^{-1} U)^{-1} = V^{-1} - V^{-1} U^T (U V^{-1} U^T + R)^{-1} U V^{-1}$, we see that the recursive equation of (10) is equivalent to the following Riccati equation

$$P_{k+1} = A(P_k - P_k C_k{}^T (C_k P_k C_k{}^T + R)^{-1} C_k P_k) A^T + Q. \qquad (11)$$

One can think of equation (11) as the Riccati equation for the least-squares estimate covariance matrix associated with the following measurement model

$$z_k = C_k x_k + n_k, \qquad (12)$$

where n_k is zero mean with covariance $R = \mathrm{diag}(R^1, \dots, R^M)$ and x_k is governed by the same process model of (7). Studying convergence of the local estimates of the error covariance matrix to that of the centralized one simplifies to studying whether the Riccati equation in (11) converges.

Lemma 1. *Let (C_k) be a sequence converging to C, where (A,C) is observable. For any positive definite initial condition P_0 the discrete-time Riccati equation (11) converges to the unique positive definite fixed point of (9).*

Proof: We consider the equivalent Riccati equation in optimal control. Let $B_k = C_k{}^T$ and consider the optimal control problem of a linear system with dynamics

$$x_{k+1} = Ax_k + B_k u_k, \qquad (13)$$

and the quadratic cost

$$J = min \sum_{i=0}^{N-1} \{x_i{}^T Q x_i + u_i{}^T R u_i\} + x_N{}^T Q_N x_N, \qquad (14)$$

where $Q_N \in \mathbb{R}^{n \times n}$ is symmetric and positive definite. It is known (see for example, [4]) that the optimal cost is given by $x_0{}^T P_N x_0$ where P is governed by the following Riccati equation

$$P_{k+1} = A^T (P_k - P_k B_k (B_k{}^T P B_k + R)^{-1} B_k{}^T P_k) A + Q, \qquad (15)$$

and $P_0 = Q_N$. The optimal cost-to-go is $x_0^T P_k x_0$. Following the approach in [4], where the convergence of the Riccati equation for constant system matrices A and B is shown, we show that as $N \to \infty$ the above recursion converges.

Let $(B_k) \to B$ where (A,B) is controllable. From the fact that the function *rank* is lower semi-continuous, it follows that for l large enough $\mathrm{rank}([B_l, AB_{l+1}, \dots, A^{n-1} B_{l+n-1}]) = n$. This implies that there exists a control sequence that drives x_l to zero in n steps. Hence, for fixed x_0 the sequence $x_0^T P_k x_0$ is bounded above by a cost corresponding to a control sequence that is arbitrary for finitely many steps $k < l$, drives x_l to zero within n steps, and applies zero from that time on. Since x_0 is arbitrary, we conclude that (P_k) is bounded. The fact that (P_k) is monotone can be shown exactly as done in [4] for the time-invariant system matrices. Hence, P_k converges. To show that it converges to the unique positive definite fixed point of (9), let P denote the steady-state value of the recursion

$$P_{k+1} = A^T (P_k - P_k B (B^T P B + R)^{-1} B^T P_k) A + Q, \qquad (16)$$

and P_2 denote the steady-state solution of recursion (15). We see that in steady-state, P and P_2 satisfy (16). This equation has a unique positive definite fixed point for any arbitrary positive definite initial condition and hence $P_2 = P$.

Now, we assume that the consensus algorithm chosen guarantees convergence of local estimates to the steady-state value of inputs to the nodes. From the above lemma, we can show the following result:

Theorem 1. *For the process and measurement model of (7) the local estimates of the error covariance matrix converge to the centralized error covariance matrix.*

Proof: For the time-invariant process and measurement model of (7), the input to node i's consensus filter for estimating F is the constant matrix $(H^i)^T (R^i)^{-1} H^i$. The consensus algorithm guarantees that $F^i{}_k$ converges to $F = \sum_{i=1}^{M} (H^i)^T (R^i)^{-1} H^i$. As $F^i{}_k \to F = H^T R^{-1} H$, $C_k \to C$ for some constant matrix C. Here, C is not necessarily equal to H but we have $H^T R^{-1} H = C^T R^{-1} C$. Observability assumption on (A,H) implies that (A,C) is also observable (Nullspace of $H^T H$ is equal to Nullspace of H). Now, the local error covariance matrix, $P_k = {Y_k}^{-1}$, is governed by the Riccati equation in (11) which converges by the previous lemma. Hence, in steady-state, the local error covariance matrices and the centralized error covariance matrix are equal to the fixed point of the Riccati equation (9).

Practically, since the local estimates of the error covariance matrix converge, one can use consensus filters required for estimating this quantity for some finite time until the required convergence is met and then use consensus filters to maintain the state estimates only. From the above result, we can also show the following:

Theorem 2. *If A is stable, then the local state estimates converge in mean to the centralized state estimate.*

Proof: We first express the steady-state dynamics of the centralized state estimate using the information vector form $y_{k|k} = Y_k x_{k|k}$,

$$y_{k+1|k+1} = P^{-1} A \bar{P} y_{k|k} + f_k, \tag{17}$$

where, $f_k = H^T R^{-1} z_k$ as defined in previous section. From convergence of the local estimates of the error covariance matrix, we have that in steady-state, node i's local information vector satisfies

$$y^i{}_{k+1|k+1} = P^{-1} A \bar{P} y^i{}_{k|k} + f^i{}_k, \tag{18}$$

where $f^i{}_k$ is the local estimate of f_k. From stability of A, it follows that the terms $(H^i)^T (R^i)^{-1} x_k$, which are the inputs to the consensus filters for estimating f_k, converge in mean. Hence, the estimates $f^i{}_k$ converge in mean to f_k. From the fact that the Kalman gain, $A - APH^T(HPH^T + R)^{-1}H$, is stable [4], we can check that $P^{-1}A\bar{P}$ is also a stable. From these two results, the local estimate of information vector given by $y^i{}_{k|k}$ converges in mean to the centralized information vector $y_{k|k}$. Now, the local estimate of the state is $x^i{}_{k|k} = P y^i{}_{k|k}$ and consequently it also converges in mean to $x_{k|k} = P y_{k|k}$, the centralized estimate of the state as desired.

One implication of this result is that even though each sensor's measurement model may not observe all states, i.e. (A, H_i) not observable, as long as the set of all sensor measurement models H and A form an observable pair, each node can maintain the mean of all the states. Also, notice that $f^i{}_k$, the local estimate of the f_k, is output of a linear filter and it has a different covariance than f_k. This implies that the local sample error covariance matrix differs from the estimated error covariance matrix. Hence, regardless of the specific consensus filters chosen, the local estimates would not converge to the least-squares estimate. This is an undesirable limitation of the decentralized Kalman filter using the consensus filters. In the next section, we show the consistency of few simulations with the above results.

V. SIMULATION RESULTS

We consider a system with dynamics

$$x_{k+1} = x_k + w_k, \tag{19}$$

where $x \in \mathbb{R}^3$, $w(k) \sim N(0, \begin{pmatrix} .3 & .1 & 0 \\ .1 & .2 & 0 \\ 0 & 0 & .1 \end{pmatrix})$. There are three sensors making measurements

$$\begin{aligned} z^1{}_k &= \begin{pmatrix} 1 & 1 & 0 \end{pmatrix} x_k + n^1{}_k, \\ z^2{}_k &= \begin{pmatrix} 1 & -1 & 0 \end{pmatrix} x_k + n^2{}_k, \\ z^3{}_k &= \begin{pmatrix} 0 & 0 & 1 \end{pmatrix} x_k + n^3{}_k, \end{aligned} \tag{20}$$

where $n^1{}_k \sim N(0,.5)$, $n^2{}_k \sim N(0,.2)$, $n^3{}_k \sim N(0,.3)$. The initial parameters are set to $x(0) = 0$, $P^i{}_0 = 10 \times I_{3\times3}$, $x^i{}_0 = 0$, $i = 1,2,3$. The communication topology is such that the pairs $(1,2)$ and $(2,3)$ exchange messages. The consensus algorithm used was the discretized version of the low-pass consensus filter proposed in [7]. We used one iteration of the consensus algorithm between each observation.

Fig. 1 shows the local estimates of state 1. Nodes 1 and 2 maintain good estimates since they make measurements of this state and also directly communicate with each other. Even though node 3 is not observing state 1, it maintains an estimate of this state with some time delay. The delay is due to the fact that it takes some time for the observation information of other nodes to reach node 3 through the consensus algorithm. Similarly, local estimates of state 3 are shown in Fig. 2.

We verifed that the local estimates of the covariance matrix converge to the centralized one. Fig. 3 shows the logarithm of the trace-norm of the estimates of the error covariance matrix. We calculated the norm of sample covariance matrices for the nodes by running the algorithm for a longer time horizon. The norm of the sample error covariance matrices for nodes 1, 2, 3, and the centralized least-squares covariance matrix were found to be .2185, .1079, .2458, and .0420 respectively. This is consistent with our analysis in that each node's sample error covariance matrix differs from its estimated error covariance matrix.

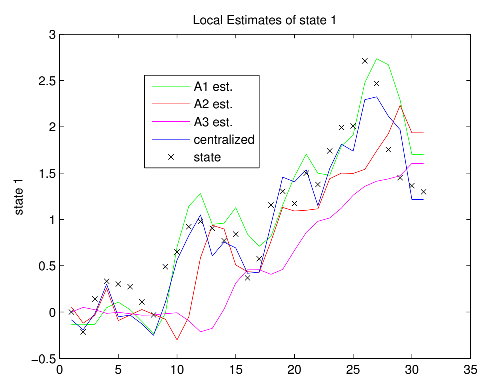

Fig. 1. Local estimates of state 1.

VI. CONCLUSIONS AND FUTURE WORK

The decentralized Kalman filter based on consensus is applicable in large sensor networks where having a central computation unit may not be feasible. Using this algorithm, each sensor node is able to maintain an estimate of the

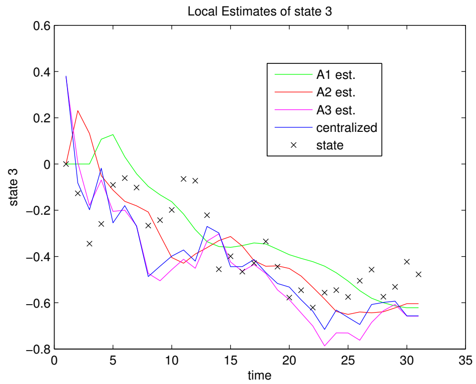

Fig. 2. Local estimates of state 3.

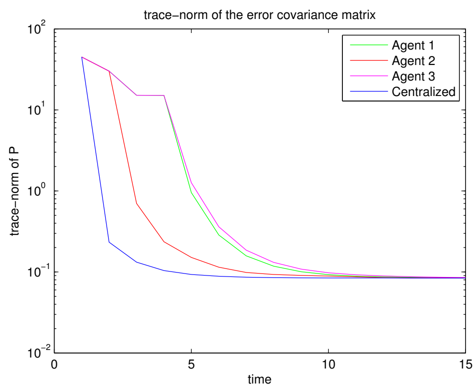

Fig. 3. Local estimates of the error covariance matrix converge to the centralized value. Here, the logarithm of the trace-norm of the matrices is plotted.

centralized error covariance matrix and the states. In addition, one can use results established on consensus algorithms such as design of algorithms that have fast convergence rates and are robust to time-delay in order to maintain better local estimates. We showed that in the time invariant process and measurement model, the local estimates of the error covariance matrix converge to the centralized error covariance matrix. For a stable process, the local estimates of the state converge in mean to the centralized state estimates. However, the local state estimates have different error covariance matrices than their estimates of the centralized error covariance matrix. Consequently, the local estimates do not converge to the least-squares estimates.

In applications in which the nodes need to maintain good estimates of the error covariance matrix, for example, in target tracking where motion is based on reducing the norm of this covariance matrix, consensus-based algorithms described can be used. In order to maintain estimates of state, one could use alternative algorithms such as weighted averaging in which the weights for combining local estimates are chosen such that the norm of each local error covariance matrix is minimized [1]. Such approaches may require more computation or off-line calculations, but could result in better estimates.

References

[1] P. Alriksson and A. Rantzer, "Distributed Kalman filtering using weighted averaging," in *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, Kyoto, Japan, Jul. 2006.

[2] ——, "Experimental evaluation of a distributed Kalman filter algorithm," in *Proceedings of the 46th IEEE Conference on Decision and Control*, Dec. 2007, to appear.

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

[4] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2005, vol. 1.

[5] T. H. Chung, J. W. Burdick, and R. M. Murray, "Decentralized motion control of mobile sensing agents in a network," in *Intl. Conference on Robotics and Automation*. ICRA, May 2006.

[6] J. Cortes, "Distributed kriged Kalman filter for spatial estimation," *IEEE Transactions on Automatic Control*, 2007.

[7] R. A. Freeman, P. Yang, and K. M. Lynch, "Stability and convergence properties of dynamic average consensus estimators," *45th IEEE Conference on Decision and Control*, pp. 338–343, Dec. 2006.

[8] S. Grime and H. F. Durrant-Whyte, "Data fusion in decentralized sensor networks," *Control Engineering Practice*, vol. 2, no. 5, pp. 849–863, 1994.

[9] B. Grocholsky, A. Makarenko, T. Kaupp, and H. F. Durrant-Whyte, "Scalable control of decentralised sensor platforms," in *International Symposium on Information Processing in Sensor Networks, IPSN*, 2003, pp. 96–112.

[10] G. M. Hoffmann and C. J. Tomlin, "Mobile sensor network control using mutual information methods and particle filters," *IEEE Transactions on Automatic Control*, 2007, submitted.

[11] A. Makarenko and H. Durrant-Whyte, "Decentralized data fusion and control in active sensor networks," in *International Conference on Information Fusion*, Stockholm, Sweden, Jun. 2004.

[12] G. O. Mutambara, *Decentralized Estimation and Control for Multisensor Systems*. CRC Press, 1998.

[13] S. Oh and S. Sastry, "Distributed networked control system with lossy links: State estimation and stabilizing communication control," in *45th IEEE Conference on Decision and Control*, 2006.

[14] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *44th IEEE conference on Decision and Control*, Dec. 2005, pp. 6698–6703.

[15] R. Olfati-Saber, J. A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," in *Proceedings of the IEEE*, vol. 95, no. 1, Jan. 2007, pp. 215–233.

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

[17] Y. Peng, R. A. Freeman, and K. M. Lynch, "Distributed cooperative active sensing using consensus filters," in *2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 405–410.

[18] B. S. Rao and H. F. Durrant-Whyte, "Fully decentralised algorithm for multisensor Kalman filtering," in *IEE Proceedings-Control Theory and Applications*, vol. 138, Sep. 1991, pp. 413–420.

[19] S. I. Roumeliotis and G. A. Bekey, "Collective localization: A distributed Kalman filter approach to localization of groups of mobile robots," in *IEEE International Conference on Robotics and Automation*, vol. 3, San Francisco, CA, Apr. 2000, pp. 2958–2965.

[20] D. P. Spanos and R. Olfati-Saber, "Distributed sensor fusion using dynamic consensus," in *16th IFAC World Congress*, Prague, Jul. 2005.

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

[22] A. Speranzon, C. Fischione, and K. H. Johansson, "Distributed and collaborative estimation over wireless sensor networks," in *45th IEEE Conference on Decision and Control*, Dec. 2005.

[23] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," in *42nd IEEE Conference on Decision and Control*, vol. 5, Dec. 2003, pp. 4997–5002.

[24] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Fourth International Symposium on Information Processing in Sensor Networks, 2005. IPSN 2005*, Apr. 2005, pp. 63–70.