Information Weighted Consensus

A. T. Kamal, J. A. Farrell and A. K. Roy-Chowdhury
University of California, Riverside, CA-92521

***Abstract*— Consensus-based distributed estimation schemes are becoming increasingly popular in sensor networks due to their scalability and fault tolerance capabilities. In a consensus-based state estimation framework, multiple neighboring nodes iteratively communicate with each other, exchanging their own local estimates of a target's state with the goal of converging to a single state estimate over the entire network. However, the state estimation problem becomes challenging when a node has limited observability of the state. In addition, the consensus estimate is sub-optimal when the cross-covariances between the individual state estimates across different nodes are not incorporated in the distributed estimation framework. The cross-covariance is usually neglected because the computational and bandwidth requirements for its computation grow exponentially with the number of nodes. These limitations can be overcome by noting that, as the state estimates at different nodes converge, the information at each node becomes redundant. This fact can be utilized to compute the optimal estimate by proper weighting of the prior state and measurement information. Motivated by this idea, we propose information-weighted consensus algorithms for distributed maximum a posteriori parameter estimates, and their extension to the information-weighted consensus filter (ICF) for state estimation. We show both theoretically and experimentally that the proposed methods asymptotically approach the optimal centralized performance. Simulation results show that ICF is robust even when the optimality conditions are not met and has low communication requirements.**

I. INTRODUCTION

Distributed estimation schemes are becoming increasingly popular in the sensor networks community due to their scalability for large networks and high fault tolerance. Unlike centralized schemes, distributed schemes usually rely on peer-to-peer communication between sensor nodes and the task of information fusion is distributed across multiple nodes. In a sensor network, each sensor may get multiple measurements of a target's state. The objective of a distributed estimation framework is to maintain an accurate estimate of the target's state using all the measurements in the network without requiring a centralized node for information fusion.

Among many types of distributed estimation schemes, *consensus algorithms* [1] are schemes where each node, by iteratively communicating with its network neighbors, can compute a function of the measurements at each node (e.g. average). The consensus estimates asymptotically converge to the global result. In practice, only a limited number of iterations can be performed due to limited bandwidth and target dynamics. Thus, true convergence may not be always reached. In the presence of state dynamics, usually a predictor-corrector model is used for state estimation, where a state prediction is made from the prior information and corrected using new measurements. The Kalman Consensus Filter (KCF) [2] is a popular distributed state estimation framework based on the average consensus algorithm. KCF works well in situations where each node gets a measurement of the target.

In a sensor network, a node might have limited observability when it does not have any measurement of a target available in its *local neighborhood* (consisting of the node and its immediate network neighbors). Due to limited observability and limited number of iterations, the node becomes *naive* about the target's state. A naive node contains less information about the state. If a naive node's estimate is given an equal weight in the information fusion scheme (as in KCF), the performance of the overall state estimation framework may decrease. The effect of naivety is severe in sparse networks where the total number of edges is much smaller than the maximum possible number of edges. The Generalized Kalman Consensus Filter (GKCF) [3], was proposed to overcome this issue by utilizing a weighted-averaging consensus scheme where the priors of each node were weighted by their covariance matrices.

The reason that these distributed schemes are usually sub-optimal is that the cross-covariances between the priors across different nodes are not incorporated in the estimation framework. As the consensus progresses, the errors in the information at each node become highly correlated with each other. Thus, to compute the optimal state estimate, the error cross-covariances cannot be neglected. However, it is difficult to compute the cross-covariance in a distributed framework. We note that in a consensus-based framework, the state estimates at different nodes achieve reasonable convergence over multiple iterations. At this point, each node contains almost identical/redundant information. This fact can be utilized to compute the optimal estimate in a distributed framework without explicitly computing the cross-covariances.

Motivated by this idea, we propose information-weighted consensus algorithms for distributed state and parameter estimation which are guaranteed to converge to the optimal centralized estimates as the prior state estimates become equal at different nodes i.e., the total number of iterations approach to infinity at the previous time step. We also show experimentally that even with limited number of iterations, the proposed algorithms achieve near-optimal performance.

This work was partially supported by ONR award N00014091066 titled Distributed Dynamic Scene Analysis in a Self-Configuring Multimodal Sensor Network.

The issue of naivety and optimality is handled by proper information weighting of the prior and measurement information. The communication bandwidth requirement is also low for the proposed methods.

Related works

Consensus algorithms [1] are one of the many types of distributed information fusion schemes. Consensus algorithms are protocols that are run individually by each agent where each agent communicates with just its network neighbors and corrects its own information iteratively using the information sent by its neighbors. The protocol, over multiple iterations, ensures the convergence of all the agents in the network to a single consensus.

Consensus algorithms have been extended to perform various tasks in a network of agents such as linear algebraic operations like SVD, least squares, PCA, GPCA [4], distributed state and parameter estimation frameworks such as the KCF [2], the GKCF [3] and the distributed maximum likelihood estimator (DMLE) [5]. A detailed review of distributed state estimation methods and comparisons with centralized and decentralized approaches can be found in [6]. These distributed state and parameter estimation frameworks have been applied in various fields including camera networks for distributed implementations of 3-D point triangulation, pose estimation [4], tracking [7], action recognition [7], [8], collaborative tracking and camera control [9] etc.

The issue of limited observability of the individual nodes has been considered previously in distributed estimation frameworks. In [10], the authors proposed a hybrid peer-to-peer/hierarchical framework for state estimation requiring fusion centers. Thus, the solution was not fully distributed. In this paper, we propose a fully distributed framework without the requirement of fusion centers.

Average consensus: Review

Average consensus [1] is a popular distributed algorithm for computing the arithmetic mean of some values. Suppose, there are N nodes and each node C_i has the state a_i. Using average consensus, the average value of these states i.e. $\frac{1}{N}\sum_{i=1}^{N} a_i$ can be computed in a distributed manner. Here, a_i can be a scalar, a vector or a matrix.

In average consensus algorithm, each node initializes its consensus state as $a_i^0 \leftarrow a_i$. At the beginning of iteration k, a node $\mathcal{C}_i$ sends its previous state a_i^{k-1} to its immediate network neighbors $\mathcal{C}_j \in \mathcal{N}_i$ and similarly receives the neighbors' previous states a_j^{k-1}. Then it updates its own state as

$$a_i^k \leftarrow a_i^{k-1} + \epsilon \sum_{j\in\mathcal{N}_i} (a_j^{k-1} - a_i^{k-1}). \tag{1}$$

By iteratively doing so, the values of the states at all the nodes converge to the average of the initial values. Here ϵ is the rate parameter which should be chosen between 0 and $\frac{1}{\Delta_{max}}$, where Δ_{max} is the maximum degree of the network graph $\mathcal{G}$. Using a higher value of ϵ would give a higher rate of convergence. However, choosing a value greater than or equal to $\frac{1}{\Delta_{max}}$ would render the system unstable.

II. Problem Formulation

Consider a sensor network with N sensors. The communication in the network can be represented using an undirected connected graph $\mathcal{G} = (\mathcal{C}, \mathcal{E})$. The set $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_N\}$ contains the vertices of the graph and represents the sensor nodes. The set $\mathcal{E}$ contains the edges of the graph which represents the available communication channels between different nodes. The set of nodes having direct communication channel with node $\mathcal{C}_i$ (sharing an edge with $\mathcal{C}_i$) is represented by $\mathcal{N}_i$.

The true state of the target(s) is represented by $\mathbf{x}(t) \in \mathcal{R}^p$. For multiple targets $\mathbf{x}(t)$ is the concatenation of the individual state vectors. A data association scheme might be necessary for multiple targets. As our focus is on the distributed state estimation problem, we would assume that the data association is given. For simplicity of notation, time index t will be dropped where the issue under consideration can be understood without it. Each node has a prior estimate of $\mathbf{x}$ as $\mathbf{x}_i^- \in \mathcal{R}^p$. The error in the prior estimate at C_i is $\boldsymbol{\eta}_i = \mathbf{x}_i^- - \mathbf{x} \in \mathcal{R}^p$ with covariance $\mathbf{P}_i^- \in \mathcal{R}^{p\times p}$. The information form of the estimators will be used throughout this paper. Thus, we will have notations in the inverse covariance form which is also known as the information/precision matrix. We denote the prior information matrix of node C_i as $\mathbf{W}_i \in \mathcal{R}^{p\times p}$, where

$$\mathbf{W}_i \quad = \quad (\mathbf{P}_i^-)^{-1}. \tag{2}$$

The observation of node C_i is denoted by $\mathbf{z}_i \in \mathcal{R}^{m_i}$ with noise covariance $\mathbf{R}_i \in \mathcal{R}^{m_i\times m_i}$, where m_i is the length of the measurement vector at node C_i. The observations from all the nodes are modeled as,

$$\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{H}}\mathbf{x} + \boldsymbol{\nu}. \tag{3}$$

Here, $\boldsymbol{\mathcal{Z}} = [\mathbf{z}_1^T, \mathbf{z}_2^T, \ldots, \mathbf{z}_N^T]^T \in \mathcal{R}^m$ and observation matrix $\boldsymbol{\mathcal{H}} = [\mathbf{H}_1^T, \mathbf{H}_2^T, \ldots, \mathbf{H}_N^T]^T \in \mathcal{R}^{m\times p}$ where, $\mathbf{H}_i \in \mathcal{R}^{m_i\times p}$ and $m = \sum_{i=1}^{N} m_i$. Observation noise $\boldsymbol{\nu}$ is assumed to be Gaussian with $\boldsymbol{\nu} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\mathcal{R}}) \in \mathcal{R}^m$. The inverse of $\boldsymbol{\mathcal{R}} \in \mathcal{R}^{m\times m}$ is denoted by $\boldsymbol{\mathcal{B}} \in \mathcal{R}^{m\times m}$. The measurements are assumed to be uncorrelated across nodes. Thus, the measurement information matrix is block diagonal and can be expressed as,

$$\boldsymbol{\mathcal{B}} = \begin{bmatrix} \mathbf{B}_1 & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_2 & & \\ \vdots & & \ddots & \\ \mathbf{0} & \ldots & & \mathbf{B}_N \end{bmatrix}. \tag{4}$$

Here, $\mathbf{B}_i = \mathbf{R}_i^{-1} \in \mathcal{R}^{m_i\times m_i}$.

III. Distributed MAP estimation (DMAP)

In this section, first we will present the centralized solution for our problem and later will derive the distributed implementation of it.

A. Centralized case

The task in a centralized *a posteriori* estimation process is to estimate the state $\mathbf{x}$ from the measurements $\boldsymbol{\mathcal{Z}}$ and prior state $\mathbf{x}_c^-$ (with information matrix $\mathbf{W}_c^-$). The centralized maximum *a posteriori* (MAP) [11] estimate $\mathbf{x}_c^+$ and its information matrix $\mathbf{W}_c^+$ in information form can be expressed as,

$$\mathbf{x}_c^+ = \Big(\mathbf{W}_c^- + \boldsymbol{\mathcal{H}}^T\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{H}}\Big)^{-1}\Big(\mathbf{W}_c^-\mathbf{x}_c^- + \boldsymbol{\mathcal{H}}^T\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{Z}}\Big), \quad (5)$$

$$\mathbf{W}_c^+ = \Big(\mathbf{W}_c^- + \boldsymbol{\mathcal{H}}^T\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{H}}\Big). \quad (6)$$

Let us define $\mathbf{U}_i = \mathbf{H}_i^T\mathbf{B}_i\mathbf{H}_i$ and $\mathbf{u}_i = \mathbf{H}_i^T\mathbf{B}_i\mathbf{z}_i$. Due to the block diagonal structure of $\boldsymbol{\mathcal{B}}$, we have

$$\boldsymbol{\mathcal{H}}^T\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{H}} = \sum_{i=1}^{N}\mathbf{H}_i^T\mathbf{B}_i\mathbf{H}_i = \sum_{i=1}^{N}\mathbf{U}_i, \quad (7)$$

$$\boldsymbol{\mathcal{H}}^T\boldsymbol{\mathcal{B}}\boldsymbol{\mathcal{Z}} = \sum_{i=1}^{N}\mathbf{H}_i^T\mathbf{B}_i\mathbf{z}_i = \sum_{i=1}^{N}\mathbf{u}_i. \quad (8)$$

Thus, we have the centralized MAP estimate as,

$$\begin{aligned}\mathbf{x}_c^+ &= \left(\mathbf{W}_c^- + \sum_{i=1}^{N}\mathbf{U}_i\right)^{-1}\left(\mathbf{W}_c^-\mathbf{x}_c^- + \sum_{i=1}^{N}\mathbf{u}_i\right)\\ &= \left(\sum_{i=1}^{N}\left(\frac{\mathbf{W}_c^-}{N} + \mathbf{U}_i\right)\right)^{-1}\sum_{i=1}^{N}\left(\frac{\mathbf{W}_c^-}{N}\mathbf{x}_c^- + \mathbf{u}_i\right), \quad (9)\end{aligned}$$

$$\mathbf{W}_c^+ = \sum_{i=1}^{N}\left(\frac{\mathbf{W}_c^-}{N} + \mathbf{U}_i\right). \quad (10)$$

B. Derivation of Distributed MAP

Now, we will derive the distributed implementation of the centralized MAP estimates of (9-10). In the centralized case, after the estimation at time $t-1$, we have the state estimate $\mathbf{x}_c^+(t-1)$ that is used as a prior ($\mathbf{x}_c^-(t)$) for the estimation at time t. For the distributed case, each node will have its own prior $\mathbf{x}_i^-(t)$. Ideally, for all i, $\mathbf{x}_i^-(t)$ should be equal to $\mathbf{x}_c^-(t)$. However, in practice, due to limited number of consensus iterations at previous time steps, there may be some discrepancies among the priors in the distributed case. Here, we will derive the distributed MAP estimation framework for the case where the priors in the distributed framework have converged to the prior in the centralized framework. Later in Sec IV, we will discuss the importance of this case for consensus-based estimation frameworks.

Under this condition, for all i, we have $\mathbf{x}_i^- = \mathbf{x}_c^-$ and $\mathbf{W}_i^- = \mathbf{W}_c^-$. Thus, from (9) and (10) we have

$$\mathbf{x}_c^+ = \left(\sum_{i=1}^{N}\left(\frac{\mathbf{W}_i^-}{N} + \mathbf{U}_i\right)\right)^{-1}\sum_{i=1}^{N}\left(\frac{\mathbf{W}_i^-}{N}\mathbf{x}_i^- + \mathbf{u}_i\right) \quad (11)$$

$$\mathbf{W}_c^+ = \sum_{i=1}^{N}\left(\frac{\mathbf{W}_i^-}{N} + \mathbf{U}_i\right) \quad (12)$$

Intuitively, this division by the number of nodes N is very important because when all the nodes have the same prior state, from the centralized perspective, the state information matrix should only be used once in the calculation of the MAP estimate. However, in the distributed case, if this division is not performed, the prior information gets N times more weight than it should. As a result, the estimator gets more biased towards the prior states and gives less weight to the new measurement information. Let,

$$\mathbf{V}_i^0 = \frac{\mathbf{W}_i^-}{N} + \mathbf{U}_i, \quad (13)$$

$$\mathbf{v}_i^0 = \frac{\mathbf{W}_i^-}{N}\mathbf{x}_i^- + \mathbf{u}_i. \quad (14)$$

Each node can compute $\mathbf{V}_i^0$ and $\mathbf{v}_i^0$ from the information available to it i.e. $\mathbf{x}_i^-$, $\mathbf{W}_i^-$, $\mathbf{u}_i$, $\mathbf{U}_i$ and N. Then, each node communicates to its neighbors with its own information matrix $\mathbf{V}_i^k \in \mathcal{R}^{p\times p}$ and information vector $\mathbf{v}_i^k \in \mathcal{R}^p$, using average consensus algorithm as described in Sec I to asymptotically compute the global averages of each of these two quantities as

$$\lim_{k\to\infty}\mathbf{V}_i^k = \frac{\sum_{i=1}^{N}\mathbf{V}_i^0}{N}, \quad (15)$$

$$\lim_{k\to\infty}\mathbf{v}_i^k = \frac{\sum_{i=1}^{N}\mathbf{v}_i^0}{N}. \quad (16)$$

Therefore, from (11-16) we have

$$\begin{aligned}\mathbf{x}_c^+ &= \lim_{k\to\infty}\left(N\mathbf{V}_i^k\right)^{-1}\left(N\mathbf{v}_i^k\right)\\ &= \lim_{k\to\infty}\left(\mathbf{V}_i^k\right)^{-1}\mathbf{v}_i^k \quad (17)\\ \mathbf{W}_c^+ &= \lim_{k\to\infty}N\mathbf{V}_i^k \quad (18)\end{aligned}$$

From (17) and (18) we can see that as $k\to\infty$, the state estimate and information matrix at each node converges to the optimal centralized MAP estimate. The DMAP framework is summarized in Algorithm 1.

Algorithm 1 Distributed Maximum A Posteriori (DMAP) at C_i

Input: Prior state estimate $\mathbf{x}_i^-$, information matrix $\mathbf{W}_i^-$, observation matrix $\mathbf{H}_i$, measurement $\mathbf{z}_i$, measurement information matrix $\mathbf{B}_i$, consensus rate parameter ϵ and total consensus iterations K.

1) Compute initial information matrix and vector

$$\mathbf{V}_i^0 \leftarrow \frac{1}{N}\mathbf{W}_i^- + \mathbf{H}_i^T\mathbf{B}_i\mathbf{H}_i \quad (19)$$

$$\mathbf{v}_i^0 \leftarrow \frac{1}{N}\mathbf{W}_i^-\mathbf{x}_i^- + \mathbf{H}_i^T\mathbf{B}_i\mathbf{z}_i \quad (20)$$

2) Perform average consensus on $\mathbf{V}_i^0$ and $\mathbf{v}_i^0$ independently

for k = 1 to K **do**

a) Send $\mathbf{V}_i^{k-1}$ and $\mathbf{v}_i^{k-1}$ to all neighbors $j \in \mathcal{N}_i$

b) Receive $\mathbf{V}_j^{k-1}$ and $\mathbf{v}_j^{k-1}$ from all neighbors $j \in \mathcal{N}_i$

c) Update:

$$\mathbf{V}_i^k \leftarrow \mathbf{V}_i^{k-1} + \epsilon\sum_{j\in\mathcal{N}_i}\left(\mathbf{V}_j^{k-1} - \mathbf{V}_i^{k-1}\right) \quad (21)$$

$$\mathbf{v}_i^k \leftarrow \mathbf{v}_i^{k-1} + \epsilon\sum_{j\in\mathcal{N}_i}\left(\mathbf{v}_j^{k-1} - \mathbf{v}_i^{k-1}\right) \quad (22)$$

end for

3) Compute MAP estimate and Information matrix

$$\mathbf{x}_i^+ \leftarrow (\mathbf{V}_i^K)^{-1}\mathbf{v}_i^K \quad (23)$$

$$\mathbf{W}_i^+ \leftarrow N\mathbf{V}_i^K \quad (24)$$

Output: MAP estimate $\mathbf{x}_i^+$ and information matrix $\mathbf{W}_i^+$.

IV. Information-Weighted Consensus Filter (ICF)

In the previous section, we derived a distributed MAP estimator for the case where each node has the prior information that equals to the prior in the centralized framework. In this section, we will extend the DMAP algorithm considering state dynamics.

The state evolution is modeled using the following linear dynamical model,

$$\mathbf{x}(t+1) = \mathbf{\Phi}\mathbf{x}(t) + \boldsymbol{\gamma}(t). \tag{25}$$

Here $\mathbf{\Phi}$ is the state propagation matrix and process noise $\boldsymbol{\gamma}(t) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$. For the centralized case, where the state estimate at time t is $\mathbf{x}_c^+(t)$ with information matrix $\mathbf{W}_c^+(t)$, we have the Kalman filter [12] state propagation equations as,

$$\mathbf{W}_c^-(t+1) = \left(\mathbf{\Phi}\mathbf{W}_c^+(t)^{-1}\mathbf{\Phi}^T + \mathbf{Q}\right)^{-1}, \tag{26}$$

$$\mathbf{x}_c^-(t+1) = \mathbf{\Phi}\mathbf{x}_c^+(t). \tag{27}$$

Combining this with our distributed MAP estimator in Sec III-B, we get the information-weighted consensus filter (ICF). The approach is summarized in Algorithm 2. At each time step, if $k \to \infty$, the DMAP estimator in Algorithm 1 guarantees that the priors for the next time step at each node will be equal to the optimal centralized one. This in turn sets the optimality condition for the next time step. This guarantees the optimality of Algorithm 2 with $k \to \infty$ at each time step.

In reality, reaching true convergence may not be possible due to limited number of consensus iterations. The number of iterations needed to reach a reasonable convergence depends on the network size and number and position of naive nodes in the network graph. In Sec V, we will show experimentally that in case of only one or a few iterations, ICF is robust to small discrepancies between the state estimates across the nodes and achieves near optimal performance.

In a practical implementation scenario, at system startup or for the first few iterations in a naive node, $\mathbf{V}_i^K$ in (32) can be $\mathbf{0}$ (thus, not invertible), if there is no prior or measurement information available in the local neighborhood. At that situation, a node will not perform step 4 and 5 in Algorithm 2 until it receives non-zero information from its neighbors (through step 3) or gets a measurement itself (through step 1) yielding $\mathbf{V}_i^K$ to be non-zero.

Algorithm 2 ICF at node C_i at time step t

Input: Prior state estimate $\mathbf{x}_i^-(t)$, prior information matrix $\mathbf{W}_i^-(t)$, observation matrix $\mathbf{H}_i$, consensus rate parameter ϵ, total consensus iterations K, state transition matrix $\mathbf{\Phi}$ and process covariance $\mathbf{Q}$.

1) Get measurement $\mathbf{z}_i$ and measurement information matrix $\mathbf{B}_i$

2) Compute initial information matrix and vector

$$\mathbf{V}_i^0 \leftarrow \frac{1}{N}\mathbf{W}_i^-(t) + \mathbf{H}_i^T\mathbf{B}_i\mathbf{H}_i \tag{28}$$

$$\mathbf{v}_i^0 \leftarrow \frac{1}{N}\mathbf{W}_i^-(t)\mathbf{x}_i^-(t) + \mathbf{H}_i^T\mathbf{B}_i\mathbf{z}_i \tag{29}$$

3) Perform average consensus on $\mathbf{V}_i^0$ and $\mathbf{v}_i^0$ independently

for k = 1 to K **do**

a) Send $\mathbf{V}_i^{k-1}$ and $\mathbf{v}_i^{k-1}$ to all neighbors $j \in \mathcal{N}_i$

b) Receive $\mathbf{V}_j^{k-1}$ and $\mathbf{v}_j^{k-1}$ from all neighbors $j \in \mathcal{N}_i$

c) Update:

$$\mathbf{V}_i^k \leftarrow \mathbf{V}_i^{k-1} + \epsilon\sum_{j\in\mathcal{N}_i}\left(\mathbf{V}_j^{k-1} - \mathbf{V}_i^{k-1}\right) \tag{30}$$

$$\mathbf{v}_i^k \leftarrow \mathbf{v}_i^{k-1} + \epsilon\sum_{j\in\mathcal{N}_i}\left(\mathbf{v}_j^{k-1} - \mathbf{v}_i^{k-1}\right) \tag{31}$$

end for

4) Compute a posteriori state estimate and information matrix for time t

$$\mathbf{x}_i^+(t) \leftarrow (\mathbf{V}_i^K)^{-1}\mathbf{v}_i^K \tag{32}$$

$$\mathbf{W}_i^+(t) \leftarrow N\mathbf{V}_i^K \tag{33}$$

5) Predict for next time step $(t+1)$

$$\mathbf{W}_i^-(t+1) \leftarrow \left(\mathbf{\Phi}(\mathbf{W}_i^+(t))^{-1}\mathbf{\Phi}^T + \mathbf{Q}\right)^{-1} \tag{34}$$

$$\mathbf{x}_i^-(t+1) \leftarrow \mathbf{\Phi}\mathbf{x}_i^+(t) \tag{35}$$

Output: State estimate $\mathbf{x}_i^+(t)$ and information matrix $\mathbf{W}_i^+(t)$.

V. Experimental Evaluation

In this section, we evaluate the performance of the proposed ICF algorithm in a simulated environment and compare it with other methods: the Centralized Kalman Filter (CKF) [12], the Kalman Consensus Filter (KCF) [2] and the Generalized Kalman Consensus Filter (GKCF) [3].

We simulate a camera network in this experiment, where the state estimation algorithms are used for tracking a target roaming within a 500×500 space. The target's initial state vector is random. The target's state vector is a $4D$ vector, with the $2D$ position and $2D$ velocity components. The initial speed is uniformly picked from $10-20$ units per time step, with a random direction uniformly chosen from 0 to 2π. The targets evolve for 40 time steps using the target dynamical model of (25). The state transition matrix $\mathbf{\Phi}$ and process covariance $\mathbf{Q}$ are chosen as the following

$$\mathbf{\Phi} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} 10 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The target randomly changes its direction and is reflected back when it reaches the grid boundary.

A set of $N = 5$ camera sensors monitor the area. The observations are generated using (3). The observation matrix $\mathbf{H}_i$ and the communication adjacency matrix $\mathbf{A}$ are set as the following

$$\mathbf{H}_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

If a camera has a measurement, measurement information matrix $\mathbf{B}_i = 0.01\mathbf{I}_2$ is used. Otherwise, $\mathbf{B}_i$ is set to $0\mathbf{I}_2$. The consensus rate parameter ϵ is set to $0.65/\Delta_{max}$ where $\Delta_{max} = 2$.

For the first experiment, the initial prior state $\mathbf{x}_i^-(1)$ and prior covariance $\mathbf{P}_i^-(1)$ is set equal at each node. A diagonal

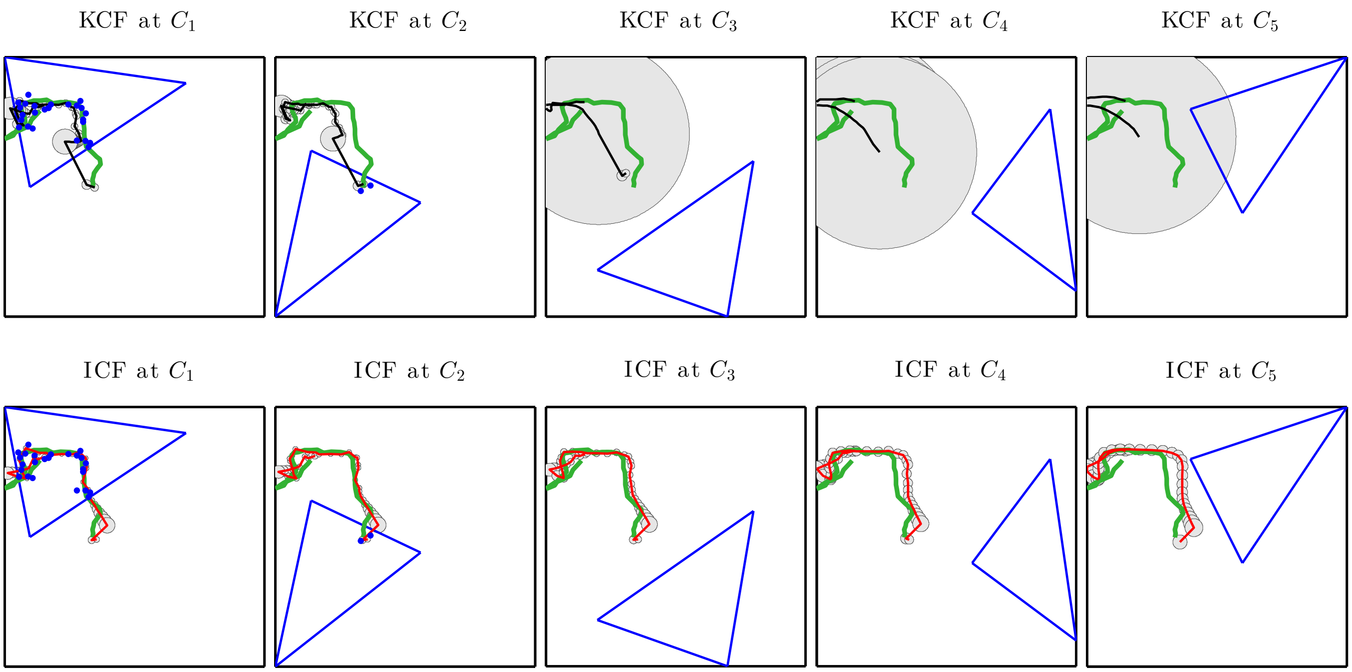

Fig. 1: Here the simulation framework is shown along with the tracking results for KCF and ICF for $K = 2$. The rectangular boxes represent the same simulation area from each camera's perspective. Within each rectangle, the blue triangles represent a camera's field of view (FOV). The green line represents the ground truth track and the blue dots represent the observations at the individual cameras. The state estimates of KCF and ICF are shown in black and red lines respectively. The gray ellipses depict the covariances of the estimates. It shows that even for $K = 2$ and the presence of naivety, ICF performs significantly better than KCF.

matrix is used for $\mathbf{P}_i^-(1)$ with the main diagonal elements as $\{100, 100, 10, 10\}$. The initial prior state $\mathbf{x}_i^-(1)$ is generated by adding zero-mean Gaussian noise of covariance $\mathbf{P}_i^-(1)$ to the ground truth state. The tracking results for KCF and ICF is shown in Fig. 1. In this experiment, the total number of consensus iterations K is set to 2. The cameras C_3, C_4 and C_5 is naive about the target's state for most of the time steps. Compared to the state estimates of KCF, in all the cameras, especially in the naive nodes, the state estimates of ICF are much closer to the ground truth. As a measure of performance, we compute the estimation error e, defined as the Euclidean distance between the ground truth position and the estimated posterior position. The mean error $\bar{e}$ is computed by averaging the errors over all cameras and time steps. In this experiment, the mean error $\bar{e}$ for KCF is 69.0376 and for ICF is 23.251.

Next, we compare the performance of KCF, GKCF and ICF with CKF after convergence. The simulation is run

Method	Mean error	Standard deviation of error
KCF (K=100)	23.8389	8.5846
GKCF (K=100)	15.2794	3.5737
ICF (K=1)	13.6672	2.5332
ICF (K=4)	12.2227	2.2702
ICF (K=10)	11.6551	2.2055
ICF (K=100)	11.5361	2.1931
CKF	11.5361	2.1931

TABLE I: Mean and standard deviation of the errors of different methods for different total number of consensus iterations K.

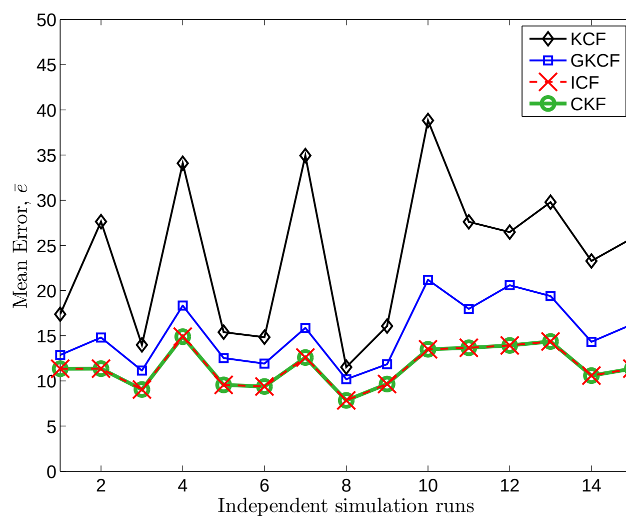

Fig. 2: Mean error of the converged estimates of different algorithms at multiple independent simulation runs. It supports the theory that with high number of consensus iterations (e.g., $K = 100$), ICF approaches the optimal centralized performance.

15 times with different randomly generated tracks. The convergence was assumed after 100 consensus iterations. The results of this experiment are shown in Fig 2. It is apparent from this figure that ICF performs better than KCF and GKCF and achieves optimal centralized performance (with high number of consensus iterations). In Table I, the mean and standard deviation of the errors for each method in this experiment are shown.

Finally, to show the robustness of ICF, we conduct an experiment by relaxing the optimality condition where the initial prior states and covariances are different at different nodes. The prior states are initialized by adding Gaussian noises (generated using the corresponding prior covariance matrices) to the initial ground truth states. The initial prior error across different cameras are correlated with correlation coefficient $\rho = 0.5$. The total number of consensus iterations K is varied from 1 to 20 with increments of 1. A total of 15 cameras are used and the camera locations, orientations, network topologies and ground truth tracks are generated randomly.

The results of this experiment is shown in Fig 3. The simulation results are averaged over 400 independent simulation runs. The mean error (solid lines) and the standard deviation ($\pm 0.2\sigma$ with dotted lines) for different methods are shown using different colors. The results show that ICF achieves near-optimal performance even when the optimality conditions are not met. This is because ICF is a consensus based approach and irrespective of the initial condition, after several time steps or consensus iterations, the states reach a reasonable converge. ICF was proved to be optimal with converged prior states. Thus, after a few time steps it achieves near-optimal performance as the system approaches the optimality conditions.

Comparing Fig 2 and 3 we can see that the performance of KCF deteriorated in the latter but the performance of GKCF and ICF was not affected much. As the number of cameras was increased from 5 to 15 in Fig. 3 (with the same number of neighbors per node), the number of naive nodes increased. This shows that unlike KCF, ICF handles the issue of naivety well. The issue of naivety in distributed frameworks was one of the main motivations for the derivation of the ICF approach.

ICF requires low communication bandwidth which is half the required bandwidth of GKCF and comparable with that of KCF. The information sent from each node to a neighbor at each iteration for various methods is shown in Table II.

Method	Message content
KCF	For 1^{st} consensus step: $\mathbf{u}_i \in \mathcal{R}^p$, $\mathbf{U}_i \in \mathcal{R}^{p\times p}$, $\mathbf{x}_i \in \mathcal{R}^p$ For additional consensus steps: $\mathbf{x}_i \in \mathcal{R}^p$
GKCF	$\mathbf{u}_i \in \mathcal{R}^p$, $\mathbf{U}_i \in \mathcal{R}^{p\times p}$, $\mathbf{x}_i \in \mathcal{R}^p$, $\mathbf{W}_i \in \mathcal{R}^{p\times p}$
ICF	$\mathbf{v}_i \in \mathcal{R}^p$, $\mathbf{V}_i \in \mathcal{R}^{p\times p}$

TABLE II: Information sent at each consensus step

VI. Conclusion

In this paper, we proposed information-weighted consensus algorithms, i.e., a distributed maximum a posteriori (DMAP) estimation framework for parameter estimation and its extension to a information-weighted consensus filter (ICF) for state estimation. We showed both theoretically and experimentally that ICF approaches the optimal centralized performance even in the presence of naive nodes as the total number of consensus iterations K increases. Simulation results showed that ICF is robust even when the optimality conditions were not met and has near-optimal performance while requiring low communication resources.

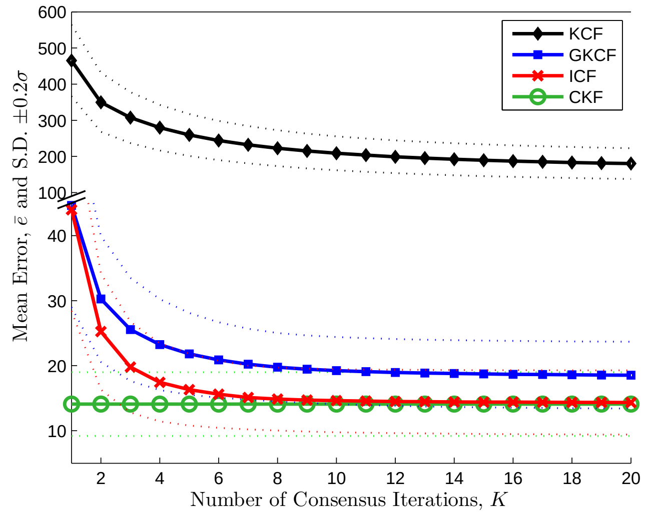

Fig. 3: Performance comparison of different approaches by varying the total number of consensus iterations, K. Each line represents the mean error $\bar{e}$ for each method. The dotted lines represent the standard deviation ($\pm 0.2\sigma$). The priors at $t = 1$ were set to be different and correlated with $\rho = 0.5$. This figure shows that ICF is robust and achieves near-optimal performance even when the optimality conditions are not met.

References

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

[2] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," in *IEEE Conference on Decision and Control*, Dec. 2009, pp. 7036 –7042.

[3] A. T. Kamal, C. Ding, B. Song, J. A. Farrell, and A. K. Roy-Chowdhury, "A generalized Kalman consensus filter for wide area video networks," in *IEEE Conference on Decision and Control*, Dec. 2011, pp. 7863 –7869.

[4] R. Tron and R. Vidal, "Distributed computer vision algorithms," *Signal Processing Magazine, IEEE*, vol. 28, no. 3, pp. 32 –45, May 2011.

[5] A. T. Kamal, J. A. Farrell, and A. Roy-Chowdhury, "Optimal distributed maximum a posteriori estimation," in *Intl. Conf. on Image Processing*, 2012.

[6] M. Taj and A. Cavallaro, "Distributed and decentralized multicamera tracking," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 46 –58, May 2011.

[7] B. Song, A. T. Kamal, C. Soto, C. Ding, J. A. Farrell, and A. K. Roy-Chowdhury, "Tracking and activity recognition through consensus in distributed camera networks," *IEEE Trans. on Image Processing*, vol. 19, no. 10, pp. 2564 –2579, Oct. 2010.

[8] A. T. Kamal, B. Song, and A. K. Roy-Chowdhury, "Belief consensus for distributed action recognition," in *Intl. Conf. on Image Processing*, Sept. 2011, pp. 141 –144.

[9] B. Song, C. Ding, A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Distributed camera networks: Integrated sensing and analysis for wide area scene understanding," *IEEE Signal Processing Magazine*, vol. 28, no. 3, pp. 20 –31, May 2011.

[10] R. Olfati-Saber and N. F. Sandell, "Distributed tracking in sensor networks with limited sensing range," *American Control Conference*, June 2008.

[11] S. M. Kay, *Fundamentals of statistical signal processing: estimation theory*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.

[12] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.