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

Distributed Data Association for Multi-Target Tracking in Sensor Networks

Nils F. Sandell and Reza Olfati-Saber
Dartmouth College
Hanover, NH 03755
{nils.f.sandell, olfati}@dartmouth.edu

Abstract— **In this paper, we explore the problem of tracking multiple targets through a field of sensors. Each sensor node is capable of making noisy measurements of the targets' positions, performing on-board computation, and wirelessly transmitting information to neighboring nodes. The problem of Multi-target tracking (MTT) can be decomposed into two main fusion problems : estimation and data association. Using Kalman-Consensus Filtering (KCF), introduced by Olfati-Saber, the authors have recently addressed distributed estimation in tracking for a single target. Data association techniques for multi-target tracking are categorized by how many time-indexed sets of measurements are made before the associations are considered "fixed". Most multi-target tracking algorithms perform the data association at a central processing node through a multiple-scan method such as Multiple Hypothesis Tracking (MHT), or single-scan techniques such as Joint Probabilistic Data Association (JPDA), Markov Chain Monte Carlo methods (MCMC), or optimal graph matching. Here, the main contribution is to introduce data association algorithms for "distributed" multi-target tracking. A formulation of Joint Probabilistic Data Association for Kalman-Consensus Filtering is formally derived. Simulations are provided to demonstrate the effectiveness of our distributed multi-target tracking algorithm for tracking multiple maneuvering targets in the sensing environment of a sensor network with 25 nodes.**

I. Introduction

Sensor networks have broad applications in detection and monitoring of events occurring in an environment in a variety of home, health, scientific modeling, security, and military applications. The availability of low-cost sensors have made it possible to create large-scale sensing that enable acquisition of massive data from spatially-distributed sources of information. Solving large-scale information processing problems for sensor networks—namely, estimation and tracking—requires development of novel algorithms that are scalable and resilient to node/link failures. Multi-target tracking (MTT) is one of the primary applications of sensor networks which can be used in a variety of scenarios. For example, flocks of UAVs can cooperatively track ground targets [14] [12], aerially-dropped sensor arrays can form ad-hoc wireless tracking networks, and a wireless sensing-based border security system can maintain the tracks of border-crossings by people and vehicles.

Multi-target tracking is the combination of data association and estimation. During the past fifty years, several filtering algorithms were proposed for estimation of the state of dynamic processes from noisy and incomplete measurements. This includes the standard Kalman Filter, the Extended Kalman Filter [1], and other Kalman filtering variants such as the Unscented Kalman Filter [21] and the Interacting Multiple Models (IMM) filter [2]. For solving estimation problems over networks, decentralized versions of several of these filters have been derived [7]. Recently, a class of consensus-based estimation algorithms began to emerge with a peer-to-peer architecture [20], [18], [4] that rely on consensus theory [16], [15] and dynamic averaging. Since the performance of these distributed estimation algorithms were not satisfactory, a novel class of distributed Kalman filtering algorithms were introduced in [13] which are called the *Kalman-Consensus Filter (KCF)*—due to the structure of its state estimator. This filter enables the nodes of a sensor network to cooperatively reach a consensus on the state estimates without any need for distributed averaging. Our proposed distributed multi-target tracking framework heavily relies on the use of the Kalman-Consensus filter.

There are several algorithms available for data association including the multiple-scan algorithms such as Reid's Multiple Hypothesis Tracking (MHT) [19]) and single-scan algorithms such as Nearest Neighbors (NN), Strongest Neighbors (SN), Probabilistic Data Association (PDA), and Joint Probabilistic Data Association (JPDA) [3], [2].

In this paper, we address distributed multi-target tracking for sensor networks with a connected topology. Each sensor node runs an identical, peer-to-peer algorithm that stores no information related to the network topology, maximizing robustness to dynamic communication topologies. We use JPDA for data association as it has an extension for IMM and it allows for a distributed algorithm with less communication overhead than a multiple scan approach. The resulting algorithm is termed JPDA-KCF, and is the main contribution of this paper.

The distributed peer-to-peer nature of our proposed fusion architecture distinguishes it from the past research on multi-target tracking. A prevalent feature of the past research on MTT is that while measurements may have been made at remote nodes, little to no processing is done at the nodes themselves. Many assume the data is simply available at a central processing node, while others deal specifically with data-driven routing schemes that allow for a centralized architecture to be realized.

Some of the weaknesses of centralized fusion architectures can be summarized as follows. First, the central processing node must run all the association and estimation routines. Second, a massive amount of data needs to be transferred to the central node via routing. This multi-hop routing process

978-1-4244-3124-3/08/$25.00 ©2008 IEEE

can be relatively expensive to perform in a reliable fashion requiring too many packet acknowledgments. Third, measurements may be arriving in an arbitrary order which further complicates the association performed by the central node. Finally, routing paths must be maintained to the central node in order to perform association and estimation. This limits the robustness of the method to not only node failures, but also node mobility and other forms of variable communication topology. A hierarchical approach to multi-target tracking is described in [10] where the authors propose a Markov Chain Monte Carlo method for data association.

The outline of the paper is as follows. The problem of MTT is introduced in Section II where a basic solution form is presented. In Section III, a review of the JPDA algorithm and its notation is given. In Section IV, the JPDA-KCF algorithm is introduced. Finally, simulation results are presented in Section V.

II. Multi-Target Tracking

In its most basic formulation, Multi-Target Tracking (MTT) is tracking of a number of targets through an area monitored by multiple sensor nodes with embedded computing capabilities. Every sensor takes noisy measurements of each target, and the node computes estimates of each target's state. Each target estimate is referred to as a *track*.

This estimation process can be broken into two tasks. The first task is establishing an association between the measurements and the targets. We focus on the so-called *single-scan* MTT, meaning this measurement-to-track assignment are computed only once for a set of measurements and considered fixed thereafter. Illustration of a measurement-to-track data association for a single tracking node is given in Fig. 2 (a). Once the associations have been made, each estimate must be updated based on its assigned measurement using an estimation algorithm such as a Kalman filter. Fig. 1 (a) illustrates a flow diagram of this process.

The assignment of measurements to tracks is an example of the 2D assignment (or graph matching) problem. In other words, pairings are made between measurements and tracks so that the total assignment cost is minimal among all possible assignments. In this case, cost can be defined by the likelihood that a measurement was emitted from a target which will be formulated in Section II-C. Polynomial-time solutions to the 2D assignment problem are well studied [9]. After performing data association, a Kalman filter can be used to update the tracks.

In this paper, we address the *distributed multi-target tracking* problem which is far less explored than the standard centralized MTT. Consider a network of sensor nodes with a communication topology $\mathcal{G} = (V, E)$, where $|V| = m$. Each node i communicates with its neighbors N_i on graph $\mathcal{G}$. Every node receives independent position measurements of a common set of targets.

Each node could simply run a centralized MTT process, but cooperation and communication among neighboring sensors will produced higher quality tracks than could be maintained by those from individual nodes. Here, we seek a distributed MTT algorithm that is scalable, self-similar [5], and topology independent such as the one shown in Fig. 1 (b).

In distributed MTT, each node obtains measurements of the targets and computes the optimal measurement-to-track assignment, as shown in Fig. 2 (b). Each node then broadcasts its set of local tracks, along with their associated measurements. As a result, each node has its own set of tracks as well as the track sets of their neighbors. However, the nodes do not necessarily know which of their neighbors' tracks correspond to their own tracks. Therefore, another round of data association must be performed, creating an assignment between local tracks and neighbors' tracks. Once this is complete, a distributed estimation algorithm can be used to update the tracks. Here, this algorithm is referred to as MTT-KCF and is formalized in Section II-C.

A. Target Dynamics and Sensor Model

Every target t is a linear process with the state x_t that evolves according to following model

$$x_t[k+1] = A[k]x_t[k] + B[k]w_t[k]$$

where $w_t[k]$ is a zero-mean Gaussian noise with covariance matrix Q_i. All sensors make measurements of the targets' state based on a linear sensing model

$$z_i^t[k] = H_i x_t[k] + v_i^t[k] \quad (1)$$

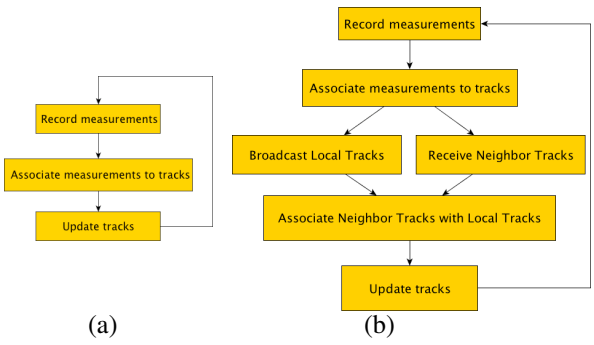

Fig. 1. (a) Flow diagram for a single node performing MTT (b) Flow diagram for networked nodes performing MTT

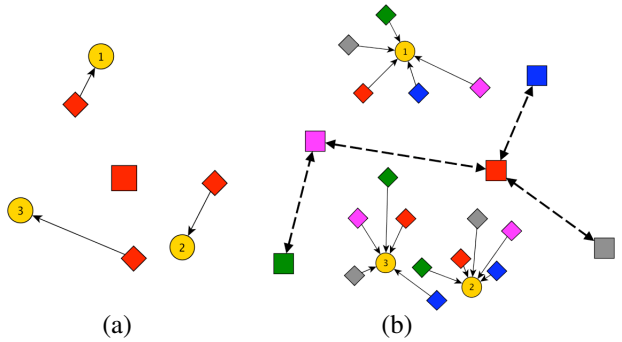

Fig. 2. Measurement-to-track associations in (a) single node environment (b) network of nodes environment. Each square represents a sensor node, each circle a target. Each diamond is a measurement made by the sensor node of the corresponding color. The dashed lines indicate communication linkage between sensor nodes.

where i denotes the sensor node recording the measurement and $v_i^t[k]$ is a zero-mean Gaussian measurement noise with covariance matrix R_i.

B. Kalman-Consensus Filter (KCF)

The Kalman-Consensus Filter is a distributed estimation algorithm with a peer-to-peer architecture [13]. The KCF algorithm uses consensus on estimates obtained by local Kalman filters, rather than constructing fused measurements and covariance information of a central Kalman Filter [11] using distributed average-consensus. Recently, the authors have applied this algorithm to tracking a single maneuvering target in sensor networks with limited sensing range [17].

Consider a network of sensors with linear sensing model in (1) so that node i has an output matrix H_i and the covariance matrix of the sensor noise R_i. Every node can broadcast packets of information to its neighbors N_i.

One iteration of the KCF is described in Algorithm 1 for tracking a single target. Note that whenever dealing with a single target, the subscript t indicating the index of a target is dropped from $x_t[k]$. Matrices P_i and M_i in Algorithm 1 are covariance of the estimation and prediction errors defined by

$$P_i[k] = E\left[\left(\bar{x}_i[k] - x[k]\right)\left(\bar{x}_i[k] - x[k]\right)^T\right],$$
$$M_i[k] = E\left[\left(\hat{x}_i[k] - x_t[k]\right)\left(\hat{x}_i[k] - x[k]\right)^T\right],$$

and $\bar{x}_i$ and $\hat{x}_i$ are state prior estimate and estimate, respectively. In addition, $\|\cdot\|$ denotes the Frobenius norm $\|X\| = \mathrm{tr}(X^TX)^{\frac{1}{2}}$.

C. Multi-Target Tracking with Kalman-Consensus Filter

Let us refer to the basic MTT algorithm described in Section II couple with the KCF estimator as the MTT-KCF algorithm. Node i starts iteration k by taking a set of measurements

$$Z_i[k] = \{z_{i1}[k], z_{i2}[k], \ldots\},$$

where each measurement $z_{ij}[k]$ corresponds to a target. The next step is for node i to assign its measurement set $Z_i[k]$ to its set of tracks $\mathcal{X} = \{\hat{x}_1, \hat{x}_2, \ldots\}$. There are a number of choices for solving the assignment problem (graph matching) in combinatorial optimization by reducing the problem to a network flow problem [8] or using Munkres Algorithm [9]. The result of any algorithm is the optimal (minimum cost) matching $f_i : Z_i[k] \to \mathcal{X}$. The matching weights w_{ij}^t for assigning measurement z_{ij} to track t are

$$w_{ij}^t = (z_{ij} - H_i\bar{x}_i^t)^T(H_iP_i^tH_i^T + R_i)^{-1}(z_{ij} - H_i\bar{x}_i^t),$$

corresponding to the number of standard deviations between z_{ij} and the expected measurement.

Let $z_i^t[k]$ be the measurement assigned to target t so that $f_i(z_i^t[k]) = x_t$. The information vector and matrix for the track of target t on node i are

$$\begin{aligned} u_i^t[k] &= H_i^TR_i^{-1}z_i^t[k] \\ U_i^t[k] &= H_i^TR_i^{-1}H_i. \end{aligned} \tag{2}$$

Algorithm 1 Iteration of KCF on node i

Given P_i, $\bar{x}_i$, parameter ε

1: Obtain measurement $z_i = H_ix + v_i$, $v_i \sim \mathcal{N}(0, R_i)$

2: Compute information vector and matrix

$$u_i = H_i^TR_i^{-1}z_i$$
$$U_i = H_i^TR_i^{-1}H_i$$

3: Broadcast message $m_i = (u_i, U_i, \bar{x}_i)$ to neighbors in N_i.

4: Receive messages $m_j = (u_j, U_j, \bar{x}_j)$ from neighbors.

5: Aggregate the information matrices and vectors of the inclusive neighbors $J_i = N_i \cup i$ of node i

$$y_i = \sum_{j\in J_i} u_j, S_i = \sum_{j\in J_i} U_j$$

6: Compute the Kalman-Consensus Estimate

$$M_i = \left(P_i^{-1} + S_i\right)^{-1}$$
$$\hat{x}_i = \bar{x}_i + M_i(y_i - S_i\bar{x}_i) + \varepsilon\frac{M_i}{1 + \|M_i\|}\sum_{j\in N_i}(\bar{x}_j - \bar{x}_i)$$

7: Update filter state

$$P_i \leftarrow AM_iA^T + BQ_iB^T$$
$$\bar{x}_i \leftarrow A\hat{x}_i$$

Every node broadcasts a message $(u_i^t, U_i^t, \bar{x}_i^t)$, $\forall t$. Over the same period, every node receives a message of the same form from its neighboring nodes $l \in N_i$. After this communication period is complete, every node has a set of information from each of its neighbors about the targets. However, each node orders its set of tracks differently, and must match its own set of tracks with the set of tracks of all of its neighbors. Again we use an assignment algorithm to form a set of optimal matchings $g_{il} : \mathcal{X} \to \mathcal{X}$, where g_{il} matches the tracks of node i with the tracks of node l. We define the cost of assigning track t_1 of node i to track t_2 of node l as

$$(\bar{x}_i^{t_1} - \bar{x}_l^{t_2})^T(P_i^{t_1} + P_l^{t_2})^{-1}(\bar{x}_i^{t_1} - \bar{x}_l^{t_2}).$$

The assignment functions g_{il} allow the KCF filter updates to be completed. Information vectors and matrices for track t on node i are formed as follows

$$y_i^t[k] = \sum_{l\in J_i} u_l^{g_{il}(t)}[k], S_i^t = \sum_{l\in J_i} U_l^{g_{il}(t)}[k] \tag{3}$$

III. Joint Probabilistic Data Association

Real-world sensors such as radar and sonar make measurements that are *cluttered*. Meaning that, in addition to the data originating from the targets, a set of measurements are recorded that correspond to no targets. Let $Z_i[k] = \{z_{i1}[k], z_{i2}[k], \ldots\}$ denote the set of $m_i[k]$ measurements obtained by node i at time k and define $Z_i^k = \{Z_i[1], Z_i[2], \ldots, Z_i[k]\}$.

In the presence of clutter, there might be more measurements in $Z_i[k]$ than there are actual targets. Direct use of

a matching algorithm that assigns tracks to measurements is an option that leads to poor performance under a dense clutter scenario, as the possibility of a false assignment is not considered in the estimation. Furthermore, the probability of detecting a target during a measurement scan is not one. A proper multi-target tracking algorithm should thus take into account the possibility that the target was not detected.

Joint Probabilistic Data Association (JPDA) is a suboptimal method of data association introduced in [6]. This method is specifically designed for cluttered measurement models. The idea in JPDA is to compute an expected state estimate over the various possibilities of measurement associations. Let $\hat{x}_i^t$ and $\bar{x}_i^t$ denote the estimate and prior estimate of the state of target t by node i, respectively. Note that time indices will be dropped wherever there is no ambiguity. The state estimate of target t computed by node i is

$$\begin{aligned}\hat{x}_i^t &= E\left[x_t|Z_i\right] \\ &= \sum_{j=1}^{m_i} E\left[x_t|\chi_{ij}^t, Z_i^k\right] P\left[\chi_{ij}^t|Z_i^k\right]\end{aligned} \tag{4}$$

where χ_{ij}^t *denotes the event that the measurement* j *on node* i *originated from target* t.

In the Joint Probabilistic Data Association Filter (JPDAF), the Kalman filter is used to find the estimate $E[x_t|\chi_{jt}, Z_i]$. Using the notation $\beta_{ij}^t = P\left[\chi_{ij}^t|Z_i\right]$ and $\beta_{i0}^t = 1 - \sum_{j=1}^{m_i}\beta_{ij}^t$ with $j = 0$ denoting the probability that no measurement is associated with target t. The JPDAF state estimate is

$$\begin{aligned}\hat{x}_i^t &= \beta_{i0}^t\bar{x}_i^t + \sum_{j=1}^{m_i}\beta_{ij}^t(\bar{x}_i^t + K_i^t(z_{ij} - H_i\bar{x}_i^t)) \\ &= \bar{x}_i^t + K_i^t\left(\sum_{j=1}^{m_i}\beta_{ij}^t z_{ij} - (1-\beta_{i0}^t)H_i\bar{x}_i^t\right) \\ &= \bar{x}_i^t + K_i^t(z_i^t - (1-\beta_{i0}^t)H_i\bar{x}_i^t)\end{aligned} \tag{5}$$

where

$$z_i^t = \sum_{j=1}^{m_i}\beta_{ij}^t z_{ij} \tag{6}$$

$$\begin{aligned}K_i^t &= P_i^t H_i^T(H_i P_i^t H_i^T + R_i)^{-1} \\ &= P_i^t H_i^T (W_i^t)^{-1}\end{aligned}$$

After taking into account the uncertainty in the origin of the measurements, the error covariance of the estimate is given by

$$M_i^t = P_i^t - (1-\beta_{i0}^t)K_i^t W_i^t (K_i^t)^T + K_i^t\tilde{P}_i^t(K_i^t)^T \tag{7}$$

where

$$\begin{aligned}\tilde{P}_i^t = &\sum_{j=1}^{m_i}\beta_{ij}^t(z_{ij} - H_i\bar{x}_i^t)(z_{ij} - H_i\bar{x}_i^t)^T \\ &- (z_i^t - H_i^t\bar{x}_i^t)(z_i^t - H_i^t\bar{x}_i^t)^T\end{aligned}$$

as shown in [6]. The process of computing β_{ij}^t is discussed at the end of Section IV and can be found in [6] as well.

IV. Joint Probabilistic Data Association with Kalman Consensus Filter (JPDA-KCF)

The architecture of the multi-target tracking algorithm in this case is the same as MTT-KCF with a difference that JPDA is used to perform local measurement-to-track associations. Therefore, the measurement assigned to track t of node i takes the form in (6).

Since these associations are not deterministic, we must reformulate the KCF algorithm to take into account the properties of JPDA.

First, we address the estimate updates. The JPDA state update equation (5) is of the same form as the standard Kalman Filter update, and we can convert this to its analogous information form using $u_i^t = H_i^T R_i^{-1} z_i^t$ and $U_i^t = H_i^T R_i^{-1} H_i$. We get

$$\hat{x}_i^t = \bar{x}_i^t + \left((P_i^t)^{-1} + U_i^t\right)^{-1}\left(u_i^t - (1-\beta_{i0}^t)U_i^t\bar{x}_i^t\right). \tag{8}$$

The information form allows updates from multiple independent measurement sources be expressed as a sum of information. Given node i and track-to-track association functions g_{il}, the state update takes the form

$$\hat{x}_i^t = \bar{x}_i^t + \left((P_i^t)^{-1} + S_i^t\right)^{-1}\left(y_i^t - S_{i0}^t\bar{x}_i^t\right) \tag{9}$$

with

$$\begin{aligned}S_i^t = \sum_{l\in J_i} U_l^{g_{il}(t)}, y_i^t = \sum_{l\in J_i} u_l^{g_{il}(t)} \\ S_{i0}^t = (1-\beta_{i0})S_i^t\end{aligned}$$

and $J_i = N_i \cup i$.

Next, we find the covariance updates for the MTT-KCF based on (7) from [6]. Let us define $\tilde{M}_i^t = \left((P_i^t)^{-1} + S_i^t\right)^{-1}$. The Kalman gain in information form is given by

$$K_i^t = \tilde{M}_i^t H_i^T R_i^{-1}.$$

Furthermore, using the matrix inversion lemma, one can show that $\tilde{M}_i^t = P_i^t - K_i^t W_i^t K_i^t$. Rewriting M_i^t in (7) gives

$$M_i^t = \beta_{i0}^t P_i^t + (1-\beta_{i0}^t)(P_i^t - K_i^t W_i^t K_i^t) + K_i^t\tilde{P}_i^t(K_i^t)^T$$

which can be simplified as

$$M_i^t = \beta_{i0}^t P_i^t + (1-\beta_{i0}^t)\left((P_i^t)^{-1} + S_i^t\right)^{-1} + K_i^t\tilde{P}_i^t(K_i^t)^T \tag{10}$$

Finally, the β_{ij}'s are calculated using the following formula [6]:

$$\begin{aligned}\beta_{ij}^t &= \frac{\exp(-(\tilde{z}_{ij}^t)^T\Lambda^{-1}\tilde{z}_{ij}^t/2)}{b + \sum_{j=1}^{m_i}\exp(-\tilde{z}_{ij}^T\Lambda^{-1}\tilde{z}_{ij}/2)} \\ \beta_{i0}^t &= \frac{b}{b + \sum_{j=1}^{m_i}\exp(-(\tilde{z}_{ij}^t)^T\Lambda^{-1}\tilde{z}_{ij}^t/2)} \\ b &= (2\pi)^{d/2}\lambda_f\det(\Lambda)^{1/2}(1-P_D P_G)/P_D\end{aligned} \tag{11}$$

where $\tilde{z}_{ij}^t = z_{ij} - H_i\bar{x}_i^t$, d is the dimension of the space in which the target is moving, P_D is the probability of detecting a target, P_G is the probability of a target lying in a gate used to eliminate events with negligible probabilities, and Λ is the covariance of the distribution of z_{ij}'s. The steps of JPDA-KCF are outlined in Algorithm 2.

Algorithm 2 JPDA-KCF on node i

Given P_i^t and $\bar{x}_i^t$ $\forall t$, parameter ε

1: Poll sensor for measurement set Z_i.
2: Use JPDA to compute the weights β_{ij}^t.
3: Compute information vector and matrix

$$u_i^t = H_i^T R_i^{-1} \sum_{j=1}^{m_i} \beta_{ij}^t z_{ij}$$

$$U_i^t = H_i^T R_i^{-1} H_i$$

4: Broadcast message $\mathcal{M}_i$ to neighbors containing
   1) $(u_i^1, u_i^2, \ldots, u_i^{m_i})$
   2) $(U_i^1, U_i^2, \ldots, u_i^{m_i})$
   3) $(\bar{x}_i^1, \bar{x}_i^2 \ldots, \bar{x}_i^{m_i})$
   4) P_i^t $\forall t$
   5) R_i, H_i
   6) β_{ij}^t $\forall j, t$
   7) z_{ij} $\forall j$ such that $\exists t$, $\beta_{ij}^t > 0$
5: Receive neighbors messages $\mathcal{M}_j$, $\forall j \in N_i$.
6: Compute track-to-track matchings g_{il}.
7: Fuse the information matrices and vectors of the inclusive neighbors $J_i = N_i \cup \{i\}$ of node i

$$y_i^t = \sum_{l \in J_i} u_l^{g_{il}(t)},$$

$$S_i^t = \sum_{l \in J_i} U_l^{g_{il}(t)},$$

$$S_{i0}^t = (1 - \beta_{i0}) S_i^t$$

8: Compute the Kalman-Consensus estimate

$$\hat{x}_i^t = \bar{x}_i^t + \left((P_i^t)^{-1} + S_i^t\right)^{-1} \left(y_i^t - S_{i0}^t \bar{x}_i^t\right) + \varepsilon M_i^t / (1 + \|M_i^t\|) \sum_{l \in J_i} (\bar{x}_l^{g_{il}(t)} - \bar{x}_i^t)$$

$$M_i^t = \beta_{i0}^t P_i^t + (1 - \beta_{i0}^t) \left((P_i^t)^{-1} + S_i^t\right)^{-1} + K_i^t \tilde{P}_i^t (K_i^t)^T$$

9: Update the covariance and prior estimate:

$$P_i^t \leftarrow A M_i^t A^T + B Q_i B^T$$

$$\bar{x}_i^t \leftarrow A \hat{x}_i^t$$

V. Simulations

To test the performance of our MTT-KCF algorithm, we consider maneuvering targets with a piece-wise linear model introduced in [17]. Intuitively, each target moves with a constant velocity along a noisy line until it hits a wall and softly reflects off the wall to stay inside a rectangular region. We use a square grid of 25 sensors evenly distributed in a region $[-a, a]^2$ with $a = 45$. The communication radius is set to 24 and the nearest neighbor distance is 22.5. The sensors make noisy position measurements according to (1) with an output matrix

$$H_i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$Q_i = I$, and $R_i = k_v^2 I_2$ where k_v^2 will vary in simulations. The sensors will be cluttered by a random number of false measurements which will be distributed as a Poisson process with the mean λ_f. Each false measurement will be placed randomly in the sensing region with uniform probability. The value chosen for λ_f will either be "no clutter" ($\lambda_f = .005$) or "cluttered" ($\lambda_f = 25$). Sample iterations from one run of the algorithm with clutter are shown in Fig. 3.

To properly assess the performance of the JPDA-KCF Algorithm, we compare it to the basic distributed multi-target tracking algorithm introduced in Section II-C. JPDA-KCF was introduced specifically to handle a cluttered sensor model, so it is expected that the two algorithms should perform comparably in the case where there is no clutter. To test this, we simulate both algorithms tracking a single target without clutter. A single target was chosen to eliminate any interference effects from multiple targets that would compound the clutter error. Fig. 4 plots the mean-squared position error of the target for both algorithms for different values of sensor error covariance k_v^2. As expected, for any value of k_v^2, the algorithms perform identically.

Next, we run the same test under "high clutter". In this case, the increasing value of k_v^2 effectively increases the clutter effect, as the average number of false measurements that appear to be reasonable given that the sensor error variance increases. It is expected that as k_v^2 increases, both methods will lose accuracy. However, it is believed that KCF-JPDA estimates will degrade in quality slower than those of MTT-KCF. This assertion is illustrated in Fig. 5.

Optimal assignment methods are often not used as they can lose their target in the clutter. This phenomenon is seen only in the tests with large sensor error covariance ($k_v^2 > 7$). During simulation, if the estimate error was greater than 15, the target was considered lost and the simulation was ended. Thus, this phenomenon makes only minor contribution to the estimate error for optimal assignment in Fig. 5. Simulation of JPDA-KCF was never terminated due to loss of target.

Intuitively, except the times when targets come close, the error for tracking multiple targets should be approximately the same as tracking a single target in clutter. This is verified

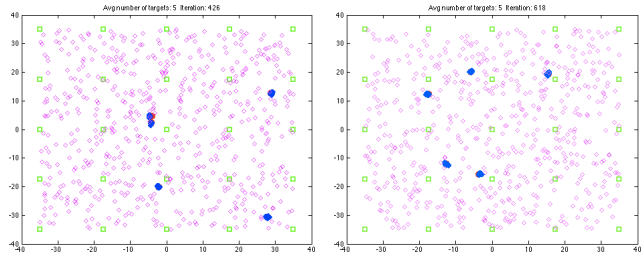

Fig. 3. Two iterations of the JPDA-KCF algorithm run on a field of 25 sensor nodes tracking 5 targets. The green squares represent sensors, the purple diamonds represent false measurements, the blue circles represent target estimates, and the red circles represent targets.

in Fig. 6. However, when two or more targets are close in both position and velocity, it is unavoidable that one target is mistaken for the other.

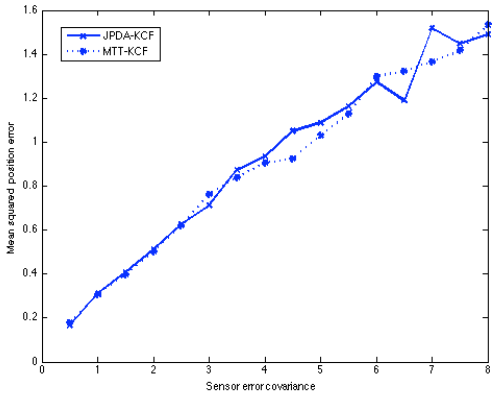

Fig. 4. Comparison of the performance of MTT-KCF and JPDA-KCF in the absence of clutter.

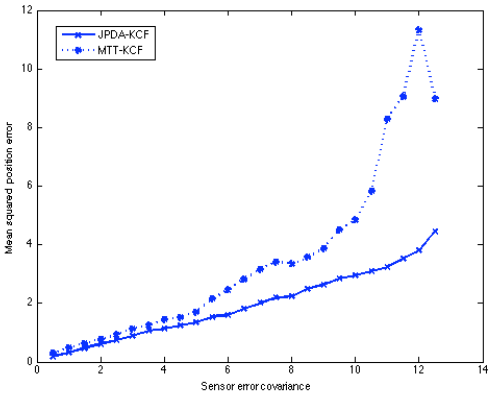

Fig. 5. Comparison of the performance of MTT-KCF and JPDA-KCF where each sensor detects an average of 25 false measurements per iteration.

VI. Conclusions

The problem of distributed data association for multi-target tracking in wireless sensor networks of is addressed based on extension of the pioneering work of Fortmann *et al.* [6] on JPDA. The resulting algorithm called JPDA-KCF (or Algorithm 2) uses the Kalman-Consensus Filter of Olfati-Saber as a distributed estimator for tracking the state the individual targets after data association. Simulation results were provided that verified Algorithm 2 mitigated the negative effect of clutter.

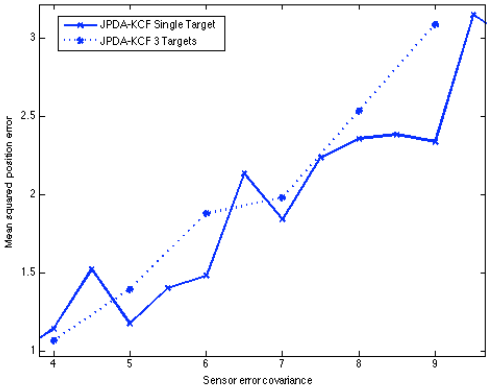

Fig. 6. Comparison of mean squared position error for JPDA-KCF algorithm when there is one target versus multiple targets

References

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

[2] Y. Bar-Shalom and W. D. Blair. *Multitarget-multisensor Tracking: Applications and Advances*. Artech House, Inc., Norwood, MA, 2000.

[3] Y. Bar-Shalom and L. Xiao-Rong. *Multitarget-multisensor tracking: Principles and Techniques*. Artech House, Inc., Norwood, MA, 1995.

[4] R. Carli, A. Chiuso, L. Shcenato, and S. Zampieri. Distributed kalman filtering using consensus strategies. *Proceedings of the 2007 IEEE Conference on Decision and Control*, pages 5486–5491, December 2007.

[5] M. Chandy and M. Charpetier. Self-similar algorithms for dynamic distributed systems. *Proceedings of the 27th International Conference on Distributed Compting*, June 2007.

[6] T. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, OE-8(3), July 1983.

[7] S. Grime and H. Durrant-Whyte. Data fusion in decentralized sensor networks. *J. Control Engineering Practice*, page 2, 1994.

[8] E.L. Lawler. *Combinatorial Optimization: Networks and Matroids*. Courier Dover Publications, 2001.

[9] J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Math*, 5(1), March 1957.

[10] S. Oh, S. Sastry, and L. Schenato. A hierarchical multiple-target tracking algorithm for sensor networks. *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, Apr. 2005.

[11] R. Olfati-Saber. Distributed Kalman filters with embedded consensus filters. *Proceedings of the 44th Conference on Decision and Control*, 2005.

[12] R. Olfati-Saber. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Transactions on Automatic Control*, pages 401–420, March 2006.

[13] R. Olfati-Saber. Distributed kalman filtering for sensor networks. *Proceedings of the 2007 Conference on Decision and Control*, 2007.

[14] R. Olfati-Saber. Distributed tracking for mobile sensor networks with information-driven mobility. *Proceedings of the 2007 American Control Conference*, July 2007.

[15] R. Olfati-Saber, J. A. Fax, and R. M. Murray. Consensus and Cooperation in Networked Multi-Agent Systems. *Proceedings of the IEEE*, 95(1):215–233, January 2007.

[16] 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.

[17] R. Olfati-Saber and N.F. Sandell. Distributed tracking in sensor networks with limited sensing range. *Proceedings of the 2008 American Control Conference*, 2008.

[18] R. Olfati-Saber and J. S. Shamma. Consensus filters for sensor networks and distributed sensor fusion. *Proceedings of the 44th Conference on Decision and Control*, 2005.

[19] D.B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, AC-24(6), December 1979.

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

[21] E.A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. *Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158, 2000.