Delayed-State Information Filter for Cooperative Decentralized Tracking

J. Capitán, L. Merino, F. Caballero and A. Ollero

***Abstract*— This paper presents a decentralized data fusion approach to perform cooperative perception with data gathered from heterogeneous sensors, which can be static or carried by robots. Particularly, a Decentralized Delayed-State Extended Information Filter (DDSEIF) is described, where full state trajectories are considered to fuse the information. This permits to obtain an estimation equal to that obtained by a centralized system, and allows delays and latency in the communications. The sparseness of the information matrix maintains the communications overhead at a reasonable level. The method is applied to cooperative tracking and some results in disaster management scenarios are shown. In this kind of scenarios the target might move in both open field and indoor areas, so fusion of data provided by heterogeneous sensors is beneficial.**

I. INTRODUCTION

Robotic application scenarios have evolved in the last decades from very simple and controlled environments to real-world dynamic applications. In this sense, the cooperation among robots and heterogeneous sensors embedded in the environment for different tasks, like surveillance in urban scenarios [1] or disaster management [2], holds as a very important issue. Real scenarios involve dynamic environments and varying conditions for perception. The robustness and reliability of autonomous perception in these scenarios are critical. In most cases, a single autonomous entity (i.e. a robot or a static surveillance camera) is not able to acquire all the information required for the application because of the characteristic of the particular task or the harmful conditions (i.e. loss of visibility), and thus, the cooperation of several of these entities is relevant.

Therefore, the goal would be to develop a data fusion framework that allows to combine information provided by a wide variety of heterogeneous sensors. The approach should be scalable, robust to communication failures and delays, under limited bandwidth. Decentralized approaches can cope with these requirements better than centralized ones [3]. In them, each node of the network employs only local information and shares its estimations with its neighbors, without knowledge of the full sensor network topology (which will change dynamically if mobile robots are considered) or broadcast facilities. They eliminate the need for a central node, and, as only local communications are performed, they are scalable. They allow the different agents of a multi-robot platform to work more independently without the need of maintaining in communication range with a central node continuously.

The main issues and problems with decentralized information fusion can be traced back to the work [4], where the Information Filter (IF, dual of the Kalman Filter) is used as the main tool for data fusion for process plant monitoring. The IF has very nice characteristics for decentralization, and for instance it has been used for decentralized mapping with aerial vehicles in [5], [6]. These works demonstrate that, for the case of static states (for instance, in mapping applications), the decentralized implementation of the IF allows to obtain locally a final estimation that is the same as that obtained by a centralized node with access to all the information.

In the case of dynamic states, for instance in tracking applications, it was noticed in [7] that if only information about the current estimation is exchanged, information will be missed with respect to a centralized estimation. The problem is due to the fact that there are some information not taken into account when performing the prediction steps in each fusing node.

This paper considers the use of a delayed-state IF to solve the decentralized cooperative perception problem with optimal information gain. As it will be described, the filter considers the trajectory of the state, that is, it maintains information about past states. The main contribution is that using this full state trajectory (not just the latest state) the nodes can recover the same information as in a centralized version, at the cost of higher message sizes. The sparse structure of the information matrix is used in order to keep the communication requirements bounded.

Other advantages of our proposal are the possibility to cope with latency in the network, as past information can be fused. Also, information about the state trajectory becomes quite important for the multi-target case, in which data association turns out to be a key issue. Since information from the past is maintained, this technique would allow to cope with previous wrong associations. Once the wrong association is detected in a past time, the trajectory could be recalculated forward from them.

Decentralized information fusion raises the problem of rumor propagation (or double counting of common information), which can lead to non-consistent estimations (due to the lost of independence in the sources) [4], [5]. Gaussian filters provide analytical solutions for fusion under unknown

J. Capitán, F. Caballero and A. Ollero are with the University of Seville, Camino de los Descubrimientos s/n, 41092, Sevilla (Spain). [jescap, caba, aollero]@cartuja.us.es. Their work is partially supported by the AWARE project (IST-2006-33579), funded by the European Commission

L. Merino is with Pablo de Olavide University, Carretera Utrera km1, 41013, Sevilla (Spain). lmercab@upo.es. His work is partially supported by the URUS project (IST-2006-045062), funded by the European Commission

common information by using the *covariance intersection* (CI) algorithm [8]. Furthermore, the use of delayed states allows to avoid common information due to common prediction functions, which is not considered by the canonical IF. In what is probably the closest work to this one up the knowledge of the authors, Bailey [9] shows how using delayed information can be even used to overcome the problems of rumor propagation in decentralized systems without the need of CI.

Delayed-state filters have been increasingly used by the SLAM community, as in [10], [11], but mainly due to the sparseness characteristics of the IF for Markov processes with high dimensional states. Here, the easy decentralization of the filter is exploited, and the sparseness characteristics are used to limit the communication overhead. In [12], the authors take into account both advantages at the same time for decentralized mapping of sensor nodes based on signal strength.

The paper is organized as follows; Section II describes the overall decentralized data fusion framework and details the use of state trajectory in the fusion process. Section III is devoted to present some experimental results. Finally, section IV gives some conclusions and future work.

II. DECENTRALIZED DATA FUSION APPROACH

A. State Trajectory Filter and Decentralized Data Fusion

The objective is to estimate the environment (defined by the state $\mathbf{X}$) by using all the measurements gathered by the sensors on the M robots of a fleet[1], $\mathbf{z}^t = [\mathbf{z}_1^{tT}, \ldots, \mathbf{z}_M^{tT}]^T$. Assuming that the data gathered by the different robots at any time instant t are *conditionally independent* given the state at that instant $\mathbf{X}_t$, and the usual Markovian assumptions, the Bayes filter to compute the belief state $bel(\mathbf{X}^t)$ for the state trajectory (from time 0 up to time t) is given by:

$$p(\mathbf{X}^t|\mathbf{z}^t) = \eta^{'} p(\mathbf{X}_0) \prod_{\tau=1}^{\tau=t} \Big[\prod_{j=1}^{M(\tau)} p(\mathbf{z}_{j,\tau}|\mathbf{X}_\tau) \Big] p(\mathbf{X}_\tau|\mathbf{X}_{\tau-1}) \tag{1}$$

with $p(\mathbf{X}_0)$ the prior, $M(\tau)$ the number of observations obtained at time τ, and $\eta^{'}$ a normalization constant. In this filter, accessing to all the information provided by the team at any moment is required. In a decentralized approach, however, each robot employs only its local data $\mathbf{z}_i^t$ and then *shares* its belief with its neighbors. The received information from other teammates is locally fused in order to improve the local perception of the world. The belief state over the full trajectory $bel_i(\mathbf{X}^t)$ for robot i is:

$$bel_i(\mathbf{X}^t) = \eta^{''} p(\mathbf{X}_0) \prod_{\tau=1}^{\tau=t} p(\mathbf{z}_{i,\tau}|\mathbf{X}_\tau) p(\mathbf{X}_\tau|\mathbf{X}_{\tau-1}) \tag{2}$$

Comparing this expression to eq. (1), it is possible to obtain the global belief from the local ones:

$$bel(\mathbf{X}^t) = \eta p(\mathbf{X}_0^t) \prod_{i=1}^{M} \frac{bel_i(\mathbf{X}^t)}{p(\mathbf{X}_0^t)} \tag{3}$$

where $p(\mathbf{X}_0^t) = p(\mathbf{X}_0) \prod_{\tau=1}^{\tau=t} p(\mathbf{X}_\tau|\mathbf{X}_{\tau-1})$. Then, if a node of the network receives all the beliefs from the other nodes, the fusion operation consists of combining all the local beliefs after removing the common information they share (the prior over the trajectory $p(\mathbf{X}_0^t)$). Applying this equation, the centralized belief can be exactly recovered.

In the case of a decentralized system, not only does each robot receives from its neighbors, but also sends information to them. In this case, the fusion equation is slightly different. If robot i received information from j, its belief would be updated as it follows:

$$bel_i(\mathbf{X}^t) \leftarrow \eta \frac{bel_i(\mathbf{X}^t) bel_j(\mathbf{X}^t)}{bel_{ij}(\mathbf{X}^t)} \tag{4}$$

where $bel_{ij}(\mathbf{X}^t)$ represents the common information between the robots (i.e., the common prior mentioned above but also information previously exchanged between the robots). This common information can be maintained by a separate filter called channel filter [13]. If there are loops in the information channels, the problem of double counting should be taken into account as well.

However, considering just marginal distributions at time t, it can be seen that it is not possible to obtain exactly the centralized marginal $bel(\mathbf{X}_t)$ from the local beliefs $bel_i(\mathbf{X}_t)$ unless these local beliefs are sent every time new information is gathered by a particular robot (or the state is static) [7], [13], [14]. The problem is due to the fact that there are some information not taken into account when performing the local prediction steps in each node. This difference will increase mainly with the number of predictions steps carried out in the local nodes between communication steps [13].

Another advantage of using delayed states is that the belief states can be received asynchronously. Each robot can accumulate evidence, and send it whenever it is possible. However, as the state grows over time, the size of the message needed to communicate its belief also does. For the normal operation of the robots, only the state trajectory over a time interval is needed, so these belief trajectories can be bounded. Note that the trajectories should be longer than the maximum expected delay in the network in order not to miss any measurements information.

B. Delayed-State Information Filter

In the particular case of Gaussian distributions, it can be seen that the overhead associated to maintaining (part of) the state trajectory can be controlled. In this case, the Information Filter is the natural way for decentralized estimation. The IF corresponds to the dual implementation of the Kalman Filter (KF). The constraints for the application of both filters are the same [15]: Markovian processes, linear prediction and measurement functions, Gaussian noises and initial Gaussian priors. While the KF represents the distribution using its first

[1]Capital letters indicate random quantities, and lower case letters realizations of these quantities. A subindex indicates information at time t, while a superindex indicate up to time t

$\boldsymbol{\mu}$ and second $\boldsymbol{\Sigma}$ order moments, the IF employs the so-called *canonical representation*. The fundamental elements are the *information vector* $\boldsymbol{\xi} = \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}$ and the *information matrix* $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$. Prediction and updating equations for the (standard) IF can also be derived from the usual KF. In the case of non-linear prediction or measurement, first order linearization leads to the Extended Information Filter (EIF). For more details, see [15], [14].

The IF presents some advantages and drawbacks when compared to the KF. One of the advantages of the canonical representation for the IF is that it can consider complete uncertainty seamlessly in the filter, by setting $\boldsymbol{\Omega}_t = \mathbf{0}$. The prediction and updating steps are dual in the KF and IF, in the sense that the prediction is more complicated in the IF than in the KF, but, on the other hand, the update steps are easier. Moreover, the additive nature of its updating step is what makes the IF interesting for multiple-system applications.

The information form also presents an interesting property when the full state trajectory $bel(\mathbf{X}^t)$ is considered. If the assumptions for the IF hold, it can be seen that the joint distribution over the full state is also Gaussian. The EIF considering delayed states can be derived from the general equation (2). The following system is considered:

$$\mathbf{X}_t = \mathbf{f}_t(\mathbf{X}_{t-1}) + \boldsymbol{\nu}_t \tag{5}$$

$$\mathbf{Z}_t = \mathbf{g}_t(\mathbf{X}_t) + \boldsymbol{\varepsilon}_t \tag{6}$$

where $\boldsymbol{\nu}_t$ and $\boldsymbol{\varepsilon}_t$ are additive noises. In a general case, $\mathbf{f}_t$ and $\mathbf{g}_t$ could be non-linear functions, so a linearization would be required. Defining the matrices $\mathbf{A}_t$ and $\mathbf{M}_t$ as $\mathbf{A}_t = \nabla \mathbf{f}_t(\boldsymbol{\mu}_{t-1})$ and $\mathbf{M}_t = \nabla \mathbf{g}_t(\bar{\boldsymbol{\mu}}_t)$, and knowing the information matrix and vector up to time $t-1$, $\boldsymbol{\Omega}^{t-1}$ and $\boldsymbol{\xi}^{t-1}$, the prediction steps are:

$$\begin{aligned}\bar{\boldsymbol{\Omega}}^t = &\begin{pmatrix} \mathbf{0} & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0} & \boldsymbol{\Omega}_{(t-1)(t-1)} & \cdots \\ \mathbf{0} & \vdots & \ddots \end{pmatrix} + \\ &+ \begin{pmatrix} \mathbf{R}_t^{-1} & -\mathbf{R}_t^{-1}\mathbf{A}_t & \mathbf{0}^T \\ -\mathbf{A}_t^T\mathbf{R}_t^{-1} & \mathbf{A}_t^T\mathbf{R}_t^{-1}\mathbf{A}_t & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix}\end{aligned} \tag{7}$$

$$\bar{\boldsymbol{\xi}}^t = \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\xi}_{t-1} \\ \boldsymbol{\xi}^{t-2} \end{pmatrix} + \begin{pmatrix} \mathbf{R}_t^{-1}(\mathbf{f}_t(\boldsymbol{\mu}_{t-1}) - \mathbf{A}_t\boldsymbol{\mu}_{t-1}) \\ -\mathbf{A}_t^T\mathbf{R}_t^{-1}(\mathbf{f}_t(\boldsymbol{\mu}_{t-1}) - \mathbf{A}_t\boldsymbol{\mu}_{t-1}) \\ \mathbf{0} \end{pmatrix} \tag{8}$$

And, if one measurement is received, the updating equations are:

$$\boldsymbol{\Omega}^t = \bar{\boldsymbol{\Omega}}^t + \begin{pmatrix} \mathbf{M}_t^T\mathbf{S}_t^{-1}\mathbf{M}_t & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{9}$$

$$\boldsymbol{\xi}^t = \bar{\boldsymbol{\xi}}^t + \begin{pmatrix} \mathbf{M}_t^T\mathbf{S}_t^{-1}(\mathbf{z}_t - \mathbf{g}_t(\bar{\boldsymbol{\mu}}_t) + \mathbf{M}_t\bar{\boldsymbol{\mu}}_t) \\ \mathbf{0} \end{pmatrix} \tag{10}$$

where $\mathbf{R}_t$, $\mathbf{S}_t$ are the corresponding covariances of the additive noises for the prediction and measurement models (5) and (6) respectively. Further details can be seen in [14]. The delayed-state IF is summarized in Algorithm 1, where **Add_M** adds a block row and a block column to the previous information matrix and **Add_V** adds a block row to the previous information vector.

Algorithm 1 $(\boldsymbol{\xi}^t, \boldsymbol{\Omega}^t) \leftarrow$Information Filter$(\boldsymbol{\xi}^{t-1}, \boldsymbol{\Omega}^{t-1}, \mathbf{z}_t)$

1: $\bar{\boldsymbol{\Omega}}^t = \textbf{Add\_M}(\boldsymbol{\Omega}^{t-1}) + \begin{pmatrix} \begin{pmatrix} \mathbf{I} \\ -\mathbf{A}_t^T \end{pmatrix} \mathbf{R}_t^{-1} \begin{pmatrix} \mathbf{I} & -\mathbf{A}_t \end{pmatrix} & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$

2: $\bar{\boldsymbol{\xi}}^t = \textbf{Add\_V}(\boldsymbol{\xi}^{t-1}) + \begin{pmatrix} \mathbf{R}_t^{-1}(\mathbf{f}_t(\boldsymbol{\mu}_{t-1}) - \mathbf{A}_t\boldsymbol{\mu}_{t-1}) \\ -\mathbf{A}_t^T\mathbf{R}_t^{-1}(\mathbf{f}_t(\boldsymbol{\mu}_{t-1}) - \mathbf{A}_t\boldsymbol{\mu}_{t-1}) \\ \mathbf{0} \end{pmatrix}$

3: $\boldsymbol{\Omega}^t = \bar{\boldsymbol{\Omega}}^t + \begin{pmatrix} \mathbf{M}_t^T\mathbf{S}_t^{-1}\mathbf{M}_t & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$

4: $\boldsymbol{\xi}^t = \bar{\boldsymbol{\xi}}_t + \begin{pmatrix} \mathbf{M}_t^T\mathbf{S}_t^{-1}(\mathbf{z}_t - \mathbf{g}_t(\bar{\boldsymbol{\mu}}_t) + \mathbf{M}_t\bar{\boldsymbol{\mu}}_t) \\ \mathbf{0} \end{pmatrix}$

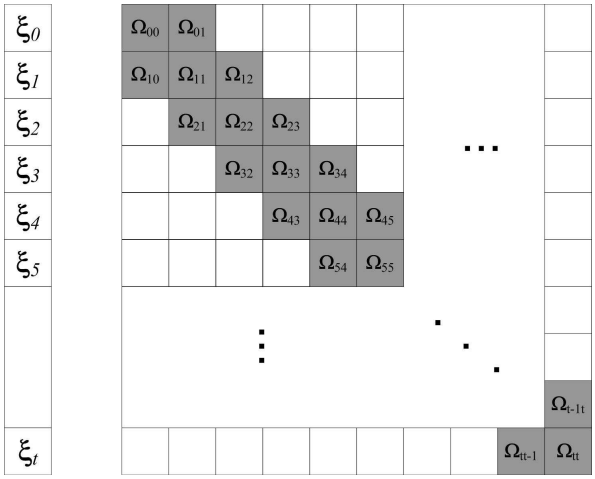

Fig. 1. Structure of the information matrix for the full trajectory. The information matrix is a block tridiagonal symmetric matrix, due to the Markov structure of the process.

Evidently, the state grows along time. In the general case of an information matrix, for a N-dimensional state, the storage required is $O(N^2)$. However, in this case, as it can be seen from the prediction and updating equations, the matrix structure is block tridiagonal and symmetric (see Fig. 1) at any time, and thus the storage required is $O(N)$ (where N is the number of time steps). Also, the computational complexity of the algorithm itself is $O(1)$, as the prediction and updating computations at each time instant only involve the previous block.

C. State Reduction

In certain situations, the length of the trajectory estimated should be limited, for instance due to storage or bandwidth restrictions. Therefore, a method for reducing the state whenever the size of the trajectory grows over a given threshold is required.

In order to do this, the removed part of the trajectory should be marginalized out. The marginal of a multivariate

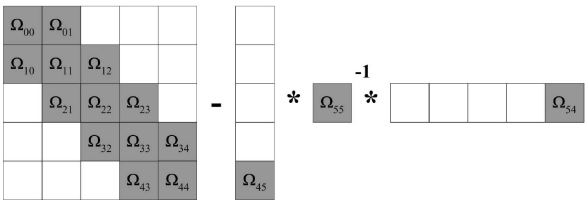

Fig. 2. Marginalization of the removed point of the trajectory. Due to the structure of the information matrix, the marginalization only involves local block operations.

Gaussian in the canonical form can be computed in closed form [11]. Moreover, due to the structure of the information matrix for this case, the computations required only involve local block matrix operations (see Fig. 2). In addition, this marginalization operation maintains the block tridiagonal structure of the matrix. In general, if the information at time t is eliminated, the only blocks affected are those linked to it (that is, $t-1$ and $t+1$), following:

$$\begin{aligned}
\boldsymbol{\Omega}_{t-1t-1} &\leftarrow \boldsymbol{\Omega}_{t-1t-1} - \boldsymbol{\Omega}_{tt-1}^T \boldsymbol{\Omega}_{tt}^{-1} \boldsymbol{\Omega}_{tt-1} \\
\boldsymbol{\xi}_{t-1} &\leftarrow \boldsymbol{\xi}_{t-1} - \boldsymbol{\Omega}_{tt-1}^T \boldsymbol{\Omega}_{tt}^{-1} \boldsymbol{\xi}_t \\
\boldsymbol{\Omega}_{t+1t+1} &\leftarrow \boldsymbol{\Omega}_{t+1t+1} - \boldsymbol{\Omega}_{t+1t} \boldsymbol{\Omega}_{tt}^{-1} \boldsymbol{\Omega}_{t+1t}^T \\
\boldsymbol{\xi}_{t+1} &= \boldsymbol{\xi}_{t+1} - \boldsymbol{\Omega}_{t+1t}^T \boldsymbol{\Omega}_{tt}^{-1} \boldsymbol{\xi}_t \\
\boldsymbol{\Omega}_{t+1t-1} &\leftarrow -\boldsymbol{\Omega}_{t+1t} \boldsymbol{\Omega}_{tt}^{-1} \boldsymbol{\Omega}_{tt-1}
\end{aligned} \tag{11}$$

D. Decentralized Information Filter

The proposed IF can be easily extended to the multi-robot case, considering a decentralized approach. In this case, each robot will run locally Algorithm 1, updating its full trajectory state with the information obtained from its sensors. When a robot i is within communication range with other robot j, they can share their beliefs, represented by their information vectors $\boldsymbol{\xi}^{i,t}$ and $\boldsymbol{\xi}^{j,t}$, and matrices $\boldsymbol{\Omega}^{i,t}$ and $\boldsymbol{\Omega}^{j,t}$. For Gaussian distributions, equation (4) leads to a quite simple fusion rule:

$$\boldsymbol{\Omega}^{i,t} \leftarrow \boldsymbol{\Omega}^{i,t} + \boldsymbol{\Omega}^{j,t} - \boldsymbol{\Omega}^{ij,t} \tag{12}$$

$$\boldsymbol{\xi}^{i,t} \leftarrow \boldsymbol{\xi}^{i,t} + \boldsymbol{\xi}^{j,t} - \boldsymbol{\xi}^{ij,t} \tag{13}$$

which only requires using a separate EIF to maintain $\boldsymbol{\Omega}^{ij,t}$ and $\boldsymbol{\xi}^{ij,t}$ (which represent the common information exchanged between i and j in the past). It is important to remark that, using this fusion equation and considering delayed states, the local estimator can obtain an estimation that is equal to that obtained by a centralized system. This common information can be locally estimated assuming a tree-shaped network topology (no cycles or duplicated paths of information). However, this fixed network topology is a constraint too strong on the potential communication links among the (mobile) robots. If there is no assumptions about the network topology, prior to combining the beliefs, unknown common information should be removed. If not, non-consistent estimations could be obtained due to the fact of adding several times the same information. Another option is to employ a conservative fusion rule, which ensures that the system does not become overconfident even in presence of duplicated information. As commented above, for the case of the IF, there is an analytic solution for this, given by the covariance intersection algorithm [8]. Therefore, the conservative rule to combine the local belief of a robot i with that received from another robot j is given by:

$$\boldsymbol{\Omega}^{i,t} \leftarrow \omega \boldsymbol{\Omega}^{i,t} + (1-\omega) \boldsymbol{\Omega}^{j,t} \tag{14}$$

$$\boldsymbol{\xi}^{i,t} \leftarrow \omega \boldsymbol{\xi}^{i,t} + (1-\omega) \boldsymbol{\xi}^{j,t} \tag{15}$$

for $\omega \in [0\,1]$. It can be seen that the estimation is consistent (in the sense that no overconfident estimations are done) for any ω. The value of ω can be selected following some criteria, such as maximizing the obtained determinant of $\boldsymbol{\Omega}^{i,t}$ (minimizing the entropy of the final distribution). The option chosen by the authors is to use ω as a fixed weight that shows the system confidence in its own estimation and the neighbor's ones.

Although employing the CI formula avoids the need to maintain an estimation of the common information transmitted to the neighbor systems, as these fusion rules are conservative, some information is lost with respect to the purely centralized case.

Synchronization of the trajectories: Special care has to be taken considering synchronization issues when combining trajectories. The trajectory is represented at discrete time intervals. The combination formula will work provided that the differences in these intervals are bounded. Therefore, trajectories should be adjusted so that the state space is the same in both cases. Fig. 3 depicts an example of the method.

In our implementation, first, newest time steps are predicted (eq. 7 and 8), and oldest ones are marginalized out (eq. 11) until trajectories are adjusted. Thus, in the example, T_0' must be removed and T_4' predicted. Then, each time step is matched with the closest one of the other trajectory. Furthermore, matchings are just allowed if the time difference is lower than a certain threshold. No matched time steps must be also marginalized out before fusing (T_2 in the example).

Finally, notice that the algorithm cannot allow crossed matchings such as the one labeled as *WRONG* in Fig. 3 This kind of wrong matchings could result in fatal errors in the estimations.

III. EXPERIMENTAL RESULTS

In order to test the decentralized perception scheme presented above, a tracking application is considered in this Section. Experimental results obtained during real field experiments integrating three sources of information (two cameras and a wireless sensor network) will be presented. The information provided by these sensors have been used to track the position of a person moving into the experiments area by means of the decentralized data fusion approach.

The cameras were used to detect the person into the field of view, providing bearing-only information of the position. Both cameras were fixed, with known intrinsic and extrinsic

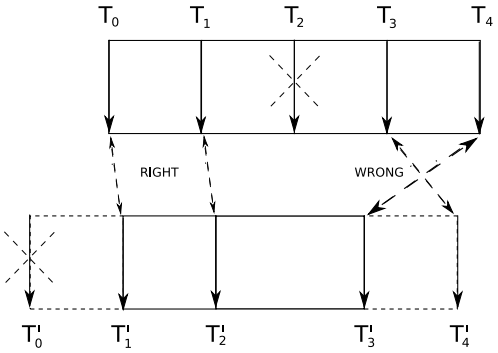

Fig. 3. Example of the method to synchronize two trajectories.

calibration parameters. In addition, the person carried a wireless sensor node that was used by the sensor network to provide positioning information based on the received signal strength information (RSSI) by means of an approach similar to [12]. This information allows to initialize the track.

A C++ implementation of the decentralized data fusion scheme called Perception Subsystem (PSS) has been used to locally process the data gathered by each sensor. This process will incorporate the local information obtained by its sensors as well as the information provided by other PSSs (neighbours' beliefs), cooperating among them in order to achieve common objectives. Then, three PSSs were launched during the experiment: camera 1, camera 2 and wireless sensor network.

The state estimated and shared between PSSs consisted of the 3D position and velocity of the person to track, both in the global coordinate system:

$$\mathbf{X}_t = \begin{pmatrix} X & Y & Z & V_x & V_y & V_z \end{pmatrix}^T \tag{16}$$

The results of the proposed algorithm are compared with the results obtained by a centralized implementation described in [16]. In that version, all the measurements were processed offline by a centralized EKF without considering communication issues or delays. The centralized filter has access to all the information provided by all the sensors instantly, a very important advantage with respect decentralized approach. Unfortunately, only the X and Y estimations are shown in the paper due to space constraints.

Thus, Fig. 4 shows the estimated X and Y position of the target provided by the software instance attached to camera 1. It can be seen how the error with respect the centralized estimation is, in mean, about one meter. In addition, the estimated standard deviation from the filter is coherent with the errors and always inside the 3σ confident interval.

An important aspect in decentralized approaches is to verify that the estimation carried out by the different software instances converge to a single solution. This is shown in Fig. 5, where the estimated XY trajectory provided by camera 1, camera 2 and wireless sensor network are plot together with the centralized estimation. It can be seen how both estimations converge to the same solution with errors in the order of one meter.

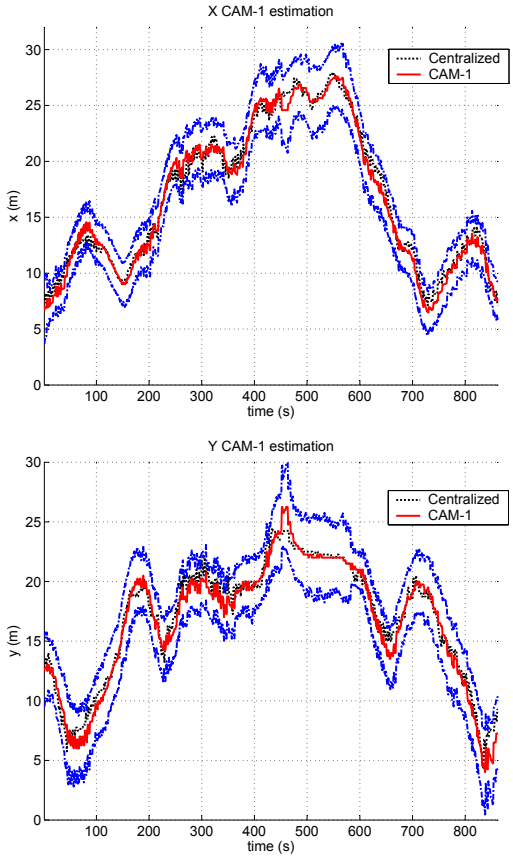

Fig. 4. Position estimation of the person using the decentralized approach presented in this paper (red solid line) in camera 1. The estimation provided by a centralized filter is also presented (black dotted line). It can be seen how the estimation is always inside the 3σ confident interval (blue dashed line)

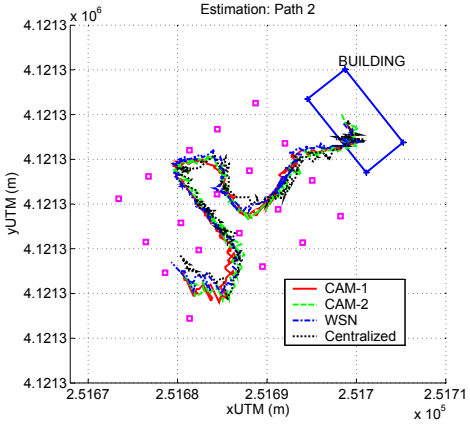

Fig. 5. XY estimation provided by the two cameras and the wireless sensor network, and centralized estimation. A sensor network was deployed into the experiments area, pink squares denote the position of each sensor node.

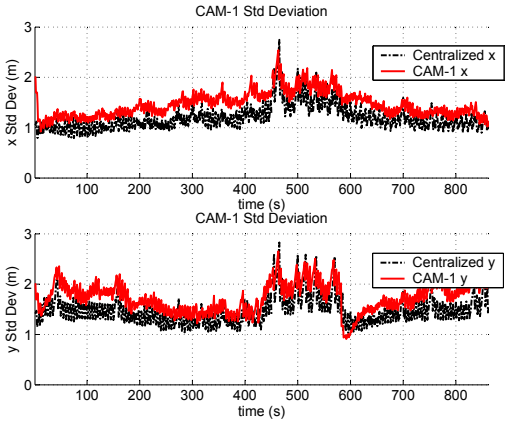

Fig. 6. Estimated standard deviation using the decentralized approach (red solid line) versus standard deviation computed by the centralized filter (black dashed line)

Fig. 6 presents the estimated standard deviation computed by the decentralized approach and the estimated by the centralized filter. As expected, the decentralized approach presents more conservative estimations than the centralized filter produced by communications issues, the *covariance intersection* algorithm, and the fact that the decentralized approach cannot access to all the information at the same time as the centralized filter does. However, it is worth to mention the closeness of both estimations which differs in no more than half a meter. This fact remarks the consistency and benefits of the proposed approach.

Thus, the experiments showed that the proposed decentralized approach is able to provide estimations with small errors (about one meter) with respect centralized filters and very similar standard deviations estimation (about half a meter difference), but with the advantage of processing the information in a fully decentralized manner, which basically improves the fault tolerance and scalability of the system.

IV. CONCLUSIONS AND FUTURE WORKS

The paper presented a decentralized data fusion scheme valid to perform cooperative perception tasks using a set of heterogeneous sensors. An extension of the usual EIF considering delayed states was proposed, which allows to obtain locally the same estimate than a centralized filter, and permits to overcome the usual delays and latency in inter-process communications.

In addition, methods to match trajectories from different agents and to fuse the information in a conservative way were explained. This is particularly important in decentralized architectures in order to face double counting information.

The decentralized data fusion approach has been implemented in C++ and tested with real information, three data sources has been integrated in those tests. The experiential results shown that the proposed approach is able to track the position of a moving object in a fully decentralized manner with small errors with respect a centralized filter, obtaining similar results in mean (about one meter error) and standard deviation (about half a meter difference).

Future works will consider exploiting the information provided by the trajectory. Techniques such as mutual information could be very useful in order to cope with the track-to-track association problem. Moreover, extending that work to the multi-target case, new algorithms could be developed in order to deal with wrong associations made in the past by using the trajectories. Finally, to demonstrate the scalability of the approach, we plan to apply it to a bigger, network involving several robots, a fixed camera network of around 20 cameras and a Wireless Sensor Network.

ACKNOWLEDGMENTS

The authors would like to thank Benjamin Grocholsky for fruitful discussions and comments about decentralized sensing.

REFERENCES

[1] A. Sanfeliu and J. Andrade-Cetto, “Ubiquitous networking robotics in urban settings,” in *Workshop on Network Robot Systems. Proceedings of the 2006 IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2006.

[2] AWARE, “Aware project web site,” 2006, http://www.aware-project.net.

[3] A. Makarenko, A. Brooks, S. Williams, H. Durrant-Whyte, and B. Grocholsky, “A decentralized architecture for active sensor networks,” in *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, vol. 2, 2004, pp. 1097–1102.

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

[5] S. Sukkarieh, E. Nettleton, J.-H. Kim, M. Ridley, A. Goktogan, and H. Durrant-Whyte, “The ANSER Project: Data Fusion Across Multiple Uninhabited Air Vehicles,” *The International Journal of Robotics Research*, vol. 22, no. 7-8, pp. 505–539, 2003.

[6] E. Nettleton, H. Durrant-Whyte, and S. Sukkarieh, “A Robust Architecture for Decentralised Data Fusion,” in *Proceedings of the International Conference on Advanced Robotics (ICAR)*, 2003.

[7] M. Rosencrantz, G. Gordon, and S. Thrun, “Decentralized sensor fusion with distributed particle filters,” in *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence*, 2003, pp. 493–500.

[8] S. Julier and J. Uhlmann, “A non-divergent estimation algorithm in the presence of unknown correlations,” in *Proceedings of the American Control Conference*, vol. 4, Jun. 1997, pp. 2369–2373.

[9] T. Bailey and H. Durrant-Whyte, “Decentralised data fusion with delayed states for consistent inference in mobile ad-hoc networks,” Australian Centre for Field Robotics, University of Sydney, Tech. Rep., 2007. [Online]. Available: http://www-personal.acfr.usyd.edu.au/tbailey/papers/delayedstateddf.pdf

[10] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte, “Simultaneous localization and mapping with sparse extended information filters,” *Internation Journal of Robotics Research*, vol. 23, no. 7–8, pp. 693–716, 2004.

[11] R. Eustice, H. Singh, and J. Leonard, “Exactly Sparse Delayed-State Filters for View-Based SLAM,” *IEEE Transactions on Robotics*, vol. 22, no. 6, pp. 1100–1114, 2006.

[12] F. Caballero, L. Merino, P. Gil, I. Maza, and A. Ollero, “A Probabilistic Framework for Entire WSN Localization Using a Mobile Robot,” *Journal of Robotics and Autonomous Systems*, vol. 56, no. 10, pp. 798–806, 2008.

[13] F. Bourgault and H. Durrant-Whyte, “Communication in general decentralized filters and the coordinated search strategy,” in *Proceedings of The 7th International Conference on Information Fusion*, 2004, pp. 723–730.

[14] L. Merino, “A cooperative perception system for multiple unmanned aerial vehicles. Application to the cooperative detection, localization and monitoring of forest fires,” Ph.D. dissertation, University of Seville, 2007. [Online]. Available: http://www.upo.es/isa/lmercab/tesis.pdf

[15] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.
[16] J. Capitan, D. Mantecon, P. Soriano, and A. Ollero, “Autonomous perception techniques for urban and industrial fire scenarios,” in *Proceedings of IEEE International Workshop on Safety, Security, and Rescue Robotics (SSRR)*, Rome, Italy, September 2007, pp. 1–6.