Consensus-based joint target tracking and sensor localization

Lin Gao[*] Giorgio Battistelli[†] Luigi Chisci[‡] Ping Wei[§]

August 22, 2021

Abstract

In this paper, consensus-based Kalman filtering is extended to deal with the problem of joint target tracking and sensor self-localization in a distributed wireless sensor network. The average weighted Kullback-Leibler divergence, which is a function of the unknown drift parameters, is employed as the cost to measure the discrepancy between the fused posterior distribution and the local distribution at each sensor. Further, a reasonable approximation of the cost is proposed and an online technique is introduced to minimize the approximated cost function with respect to the drift parameters stored in each node. The remarkable features of the proposed algorithm are that it needs no additional data exchanges, slightly increased memory space and computational load comparable to the standard consensus-based Kalman filter. Finally, the effectiveness of the proposed algorithm is demonstrated through simulation experiments on both a tree network and a network with cycles as well as for both linear and nonlinear sensors.

[*] School of Electronic Engineering, Center for Cyber Security, University of Electronic Science and Technology of China. Email: gaolin_uestc@126.com.

[†] Dipartimento di Ingegneria dell'Informazione (DINFO), Università degli Studi di Firenze. Email: giorgio.battistelli@unifi.it.

[‡] Dipartimento di Ingegneria dell'Informazione (DINFO), Università degli Studi di Firenze. Email: luigi.chisciunifi.it.

[§] School of Electronic Engineering, Center for Cyber Security, University of Electronic Science and Technology of China. Email: pwei@uestc.edu.cn.

1 Introduction

Distributed target tracking (DTT) has recently attracted great attention due to the fast development of wireless sensor networks (WSNs). The remarkable feature of a WSN is that there is no fusion center, and the nearby nodes can freely exchange data. The goal of DTT is to achieve scalability and comparable performance with respect to the centralised algorithms[1]. In the past, most work on DTT assumed that all the sensors have been correctly localized with respect to a global coordinate system, or at least that each sensor knows the positions of neighbours with respect to its own local coordinate system[2, 3, 4, 5, 6]. In many practical scenarios, however, sensor self-localization is also an important task to be jointly faced along with target tracking. Most of the existing work on sensor self-localization relied on two approaches. In the first approach, called *cooperative localization*, each sensor is provided with direct measurements relative to the positions of its neighbors [7, 8, 9, 10, 11, 12, 13, 14, 15]. The second approach is based on exploiting some reference nodes of known positions (also called anchors) in the global coordinate system [16, 17, 18, 19, 20]. The locations of anchors are assumed known a priori or can be obtained by using global localization technology such as, e.g., GPS (Global Positioning System). Undoubtedly both approaches have their limitations. The former requires additional devices for measuring the positions of the neighboring nodes, and also that such devices have the capability of distinguishing signals from different nodes, or otherwise a data association problem has to be solved. The latter can only be used in some specific scenarios where either prior knowledge of the surveillance area is available or signals from the global localization equipment can be received. In this paper, the interest is for a technique that neither needs sensing the positions of neighbors nor the presence of reference nodes.

In this respect, two interesting techniques for *joint target tracking and sensor self-localization* (JTTSL) have been presented in [21] and [22]. In particular, [21] exploits online distributed *maximum likelihood* (ML) and *expectation-maximization* (EM) methods. The nodes iteratively exchange the local likelihoods based on the message passing (belief propagation) technique. This approach, however, suffers from three major drawbacks. First, each node must store the data of all its neighbors and thus needs much extra memory space. Secondly, at each sampling interval several iterations must be carried out in order to exchange the data through the network. The third and most important drawback is that the employed message passing method is well suited for networks with tree topology but suffers from the problem of double counting in networks with cycles.

Conversely, [22] follows a Bayesian approach in order to compute in each node the posterior distribution of the drift parameters (i.e., the relative positions of the neighbors). Specifically, a Monte Carlo method is adopted to represent such a distribution. The disadvantage of [22] is, therefore, that it needs a large amount of particles to approximate the drift parameter distribution, thus implying a heavy computational load which may be unsuitable for sensor nodes with limited computing capabilities.

In this paper, we propose to solve the JTTSL problem exploiting consensus [23, 24, 25, 26, 27, 28]. An advantage of a consensus approach is that any individual node needs only to store the integrated information resulting from the fusion with all its neighbors, and thus does not require significant extra memory space. Furthermore, consensus-based algorithms avoid double counting of information for any kind of network, also involving cycles. In the proposed algorithm, the weighted Kullback-Leibler (WKL) divergence [29] is employed as optimal cost function to measure the discrepancy between estimated and true drift parameters. An online optimization method is then used to update the drift parameters at each sampling time. The advantages of the proposed algorithm are that it needs no additional data exchanges, slight additional memory space and computational load compared to the standard consensus Kalman filter (CSKF), which means that the proposed algorithm can be easily implemented on low-cost sensor nodes and, thus, have wide potential application. The proposed algorithm, named *Joint Target Tracking and Sensor Localization CSKF* (JTTSL-CSKF), can also be regarded as a generalized version of the CSKF.

The rest of the paper is organized as follows. Section II deals with the JTTSL problem formulation. Section III reviews the CSKF approach to target tracking whenever the locations of neighboring sensor nodes are known. Then, section IV presents a novel consensus-based approach to JTTSL. Section V provides simulation examples to demonstrate the effectiveness of the proposed approach. Section VI ends the paper with concluding remarks as well as perspectives for future work.

2 Problem Formulation

This paper addresses the problem of target tracking by means of a distributed sensor network wherein each sensor node gets measurements of the target relative to its local coordinate system. The sensor network is denoted as $(\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of sensor nodes and $\mathcal{A} \subseteq \mathcal{N} \times \mathcal{N}$ the set of connections such that $(i,j) \in \mathcal{A}$ if node j can receive data from node i. Let $|\mathcal{N}|$ denotes the cardinality of $\mathcal{N}$. Further, for each node $i \subseteq \mathcal{N}$, $\mathcal{N}^i$ will denote the set of its in-neighbor nodes (including itself). It is further assumed that all sensors are time-synchronized.

The state vector of the target, in the local coordinates of node i, is denoted as $\mathrm{x}_t^i = [\xi_t^i, \dot{\xi}_t^i, \eta_t^i, \dot{\eta}_t^i]^T$ where (ξ_t^i, η_t^i) and $(\dot{\xi}_t^i, \dot{\eta}_t^i)$ denote the Cartesian coordinates of position and velocity at time t, respectively. The motion of the target is modelled by

$$\mathrm{x}_{t+1}^i = f_t\left(\mathrm{x}_t^i\right) + w_t. \tag{1}$$

Further, at each sensor i, measurements of the target are obtained according to

$$\mathrm{y}_t^i = h_t^i\left(\mathrm{x}_t^i\right) + v_t^i, i \in \mathcal{N}. \tag{2}$$

It is assumed that $w_t, v_t^1, v_t^2, \ldots$ are mutually uncorrelated zero-mean white noises with covariances $Q_t = E[w_t w_t^T] > 0$ and $R_t^i = E[v_t^i (v_t^i)^T] > 0$. It is also convenient to define the noise information matrices $W_t \triangleq Q_t^{-1}$ and $V_t^i \triangleq (R_t^i)^{-1}$. Without loss of generality, it is supposed that each node is located at the origin of its local coordinate system. Let $\left(\xi^{i,j}, \eta^{i,j}\right)$ denote the position of node j in the local coordinates of node i and $\theta^{i,j} \triangleq [\xi^{i,j}, 0, \eta^{i,j}, 0]'$ the drift parameter of sensor j with respect to i. Then, assuming that the sensor nodes are motionless, the target states x_t^i and x_t^j in nodes i and j are related by

$$\mathrm{x}_t^i = \mathrm{x}_t^j + \theta^{i,j} \tag{3}$$

Notice that $\theta^{i,j} = -\theta^{j,i}$ and $\theta^{i,i} = 0$. Due to the fact that each sensor can only communicate with its neighbours, for self-localization purposes only $\theta^{i,j}$ for $j \in \mathcal{N}^i \backslash \{i\}$ are needed by node i. Let Θ denote the vector of all drift parameters, i.e. $\Theta = col\left(\theta^{i,j}, (i,j) \in \mathcal{A}\right)$. Hence, sensor i is only interested to the sub-vector $\Theta^i = col\left(\theta^{i,j}, j \in \mathcal{N}^i \backslash \{i\}\right)$. The objective of distributed JTTSL is therefore, for each sensor node i, to jointly estimate the target state x_t^i (in the local coordinates of node i) and the drift parameter vector Θ^i.

3 Distributed state estimation based on consensus

Suppose preliminarily that all drift parameters $\theta^{i,j}, (i,j) \in \mathcal{A}$, are a priori known. Such an assumption will be relaxed in the following section. Suppose further that in each node i of the network a probability density function (PDF) p^i is available representing the information at node i on the state vector x

expressed in local coordinates. If no information on the correlation between the PDFs of different sensors is available, when fusing the information of different sensors care must be taken in order to avoid the data incest phenomenon, that is the double counting of common information. As well known, fusion rules that are robust to double counting are Covariance Intersection (CI) and Exponential Mixture Density (the generalization of CI for arbitrary densities). Specifically, if the PDFs of all nodes are Gaussian with mean x^{i} and covariance P^i, then the fused PDF in the local coordinates of node i is also Gaussian with mean $\overline{\mathrm{x}}^i$ and covariance $\overline{P}^i$ given by

$$\begin{aligned} \left(\overline{P}^i\right)^{-1} &= \sum_{j\in\mathcal{N}} \pi^{i,j}\left(P^j\right)^{-1} \\ \left(\overline{P}^i\right)^{-1}\overline{\mathrm{x}}^i &= \sum_{j\in\mathcal{N}} \pi^{i,j}\left(P^j\right)^{-1}(\mathrm{x}^j+\theta^{i,j}) \end{aligned} \tag{4}$$

where, for any i, the weights $\pi^{i,j}, j \in \mathcal{N}$, are such that $\sum_{j\in\mathcal{N}}\pi^{i,j} = 1$ and $\pi^{i,j} > 0$. From an information theoretic point of view, such a fusion rule admits a meaningful interpretation in terms of average with respect to the Kullback-Leibler (KL) divergence [29]. In fact, the PDF $\overline{p}^i(\cdot)$ is the one minimizing the weighted Kullback-Leibler (WKL) divergence in that [29]

$$\overline{p}^i = \arg\min_p \sum_{j\in\mathcal{N}} \pi^{i,j} D_{KL}\left(p \| \, p^{i,j}\right) \tag{5}$$

where $p^{i,j}$ denotes the PDF representing the information available at node j expressed in the local coordinates of node i and $D_{KL}(p\| \, q)$ denotes the KL divergence of q from p defined as

$$D_{KL}\left(p\| \, q\right) = \int p(\mathrm{x}) \log\frac{p(\mathrm{x})}{q(\mathrm{x})} d\mathrm{x}\,. \tag{6}$$

While (4) and (5) require access to the PDFs of all the sensors, it was shown in [29] that actually the average PDF can be computed in a distributed way (i.e. exchanging only information with the neighbors) by using consensus. Combining this idea with local Kalman filters running in each network node, it is possible to derive a consensus Kalman filter with guaranteed stability properties.

Specifically, let us denote the local target estimated state vector and covariance at node i and time t as $\hat{\mathrm{x}}^i_{t|t}$ and $P^i_{t|t}$. The local information matrices are denoted as

$$\Omega^i_{t|t-1} \triangleq (P^i_{t|t-1})^{-1}, \quad \Omega^i_{t|t} \triangleq (P^i_{t|t})^{-1} \tag{7}$$

and the local information vectors

$$q^i_{t|t-1} \triangleq \Omega^i_{t|t-1}\hat{\mathrm{x}}^i_{t|t-1}, \quad q^i_{t|t} \triangleq \Omega^i_{t|t}\hat{\mathrm{x}}^i_{t|t} \tag{8}$$

In the context of distributed target tracking, each node uses the received measurements to first update the local information, and then exchanges data with the neighbors so that information fusion is carried out in order to improve the tracking performance of each node. In the information fusion step, the consensus algorithm can be used to minimize (5). The CSKF is summarized in Table 1. Note that the consensus weights $\pi^{i,j}$ should satisfy $\pi^{i,j} > 0$ and $\sum_{j\in\mathcal{N}^i}\pi^{i,j} = 1$. For the choice of the consensus weights $\pi^{i,j}$, which is outside the scope of this paper, please refer to [28].

In terms of KL average, each consensus step admits the following interpretation. Let $p^{i,j}_t(\mathrm{x},\ell)$ denote the PDF available at node j at consensus step ℓ of the t-th sample interval, expressed in the local

Table 1: Consensus Kalman Filter (node i, time t)

Correction
$C_t^i = \frac{\partial h_t^i}{\partial \mathrm{x}_t}\Big\rvert_{\mathrm{x}_t = \hat{\mathrm{x}}_{t\lvert t-1}^i}$
$\Omega_{t\lvert t}^i(0) = \Omega_{t\lvert t-1}^i + (C_t^i)^T V_t^i C_t^i$
Sample the local measurement y_t^i, then
$\bar{\mathrm{y}}_t^i = \mathrm{y}_t^i - h_t\left(\hat{\mathrm{x}}_{t\lvert t-1}^i\right) + C_t^i \hat{\mathrm{x}}_{t\lvert t-1}^i$
$q_{t\lvert t}^i(0) = q_{t\lvert t-1}^i + (C_t^i)^T V_t^i \bar{\mathrm{y}}_t^i$
Consensus
For $\ell = 0, 1, \ldots, L-1$ do
Exchange the information with its neighbors and fuse the quantities $q_t^i(\ell)$ and $\Omega_t^i(\ell)$ according to
$q_t^i(\ell+1) = \sum_{j \in \mathcal{N}^i} \pi^{i,j}[q_{t\lvert t}^j(\ell) + \Omega_{t\lvert t}^j(\ell)\theta^{i,j}]$
$\Omega_t^i(\ell+1) = \sum_{j \in \mathcal{N}^i} \pi^{i,j}\Omega_{t\lvert t}^j(\ell)$
End for
$\hat{\mathrm{x}}_{t\lvert t}^i = [\Omega_{t\lvert t}^i(L)]^{-1} q_t^i(L), \ \ \Omega_{t\lvert t}^i = \Omega_t^i(L)$
Prediction
$\hat{\mathrm{x}}_{t+1\lvert t}^i = f_t\left(\hat{\mathrm{x}}_{t\lvert t}^i\right), \ A_t = \frac{\partial f_t}{\partial \mathrm{x}_t}\Big\rvert_{\mathrm{x}_t = \hat{\mathrm{x}}_{t\lvert t}^i}$
$\Omega_{t+1\lvert t}^i = W_t - W_t A_t \left(\Omega_{t\lvert t}^i + A_t^T W_t A_t\right)^{-1} A_t^T W_t$
$q_{t+1\lvert t}^i = \Omega_{t+1\lvert t}^i \hat{\mathrm{x}}_{t+1\lvert t}^i$

coordinates of node i. Then, assuming Gaussian PDFs[1], the new PDF $p_t^{i,i}(\mathrm{x}, \ell+1)$ at node i is the one minimizing the weighted KL divergence with respect to the neighbors in that

$$p_t^{i,i}(\mathrm{x}, \ell+1) = \arg\min_p \sum_{j \in \mathcal{N}^i} \pi^{i,j} D_{KL}\left(p \| \, p_t^{i,j}(\cdot, \ell)\right). \tag{9}$$

4 Joint state estimation and online drift parameters calibration

In real applications, the drift parameters $\theta^{i,j}, (i,j) \in \mathcal{A}$, are not always a priori known and need, therefore, be estimated together with the state vector x_t. Notice preliminarily that a possible solution would amount to considering the distributed estimation problem for a global state X_t consisting of the target kinematic state x_t and the overall drift parameter vector Θ containing the drifts parameters of all links in the network. However, such an approach would not be scalable since the dimension of the global state $X_t = \left[\mathrm{x}_t^T, \Theta^T\right]^T$ would increase with the number of network links and further would require the knowledge of the network topology. Hence, in this paper we follow a different approach in which each node only estimates the drift parameters Θ^i pertaining to its neighbors $j \in \mathcal{N}^i \backslash \{i\}$. This is a suboptimal approach which, however, has the advantage of being scalable and does not require

[1] Clearly, the PDFs are Gaussian only when both the motion and measurement models are linear and both process and measurement noises are Gaussian. Otherwise this amounts to approximating the true local PDF with a Gaussian one having same mean and covariance.

any global information on the network topology. Further, in order to keep the communication load as low as possible, we propose an approach that does not require any additional data exchange with respect to the Consensus Kalman Filter of Table I.

The idea is to exploit the information theoretic interpretation of the consensus step (9) in order to define a suitable loss function which can be used for estimation of the drift parameters. In fact, since the transformation from the local coordinates of node j to those of node i is a function of the drift parameter $\theta^{i,j}$, it is possible to define the loss function

$$J_t^i(\Theta^i,\ell) = \min_p \sum_{j\in\mathcal{N}^i} \pi^{i,j} D_{KL}\left(p \| \, p_t^{i,j}(\cdot,\ell)\right) \tag{10}$$

where the vector Θ^i contains all drift parameters relative to the neighbors of node i. The rationale for such a choice is that, when all the local filters perform well, then all the local PDFs should provide a reasonably accurate estimate of the target state in local coordinates. In this case, the discrepancy between the PDFs in two neighboring nodes i and j is mainly due to the different coordinates. Hence, it is reasonable to take as estimate of the drift parameter $\theta^{i,j}$ the value which minimizes such a discrepancy. In turn, the estimates of the drift parameters are used for the distributed state estimator.

Assuming Gaussian PDFs, the loss function (10) turns out to be a quadratic function of Θ^i. In fact, for two Gaussian distributions $p^i(\cdot), p^j(\cdot)$ with means $\mathrm{x}^i, \mathrm{x}^j \in \mathbb{R}^d$ and covariance matrices P^i, P^j, their KL divergence in the coordinates of node i can be written as

$$D_{KL}(p_i \| \, p_j) = \tfrac{1}{2}\left[\left(\mathrm{x}^i - \left(\mathrm{x}^j + \theta^{i,j}\right)\right)^T \left(P^j\right)^{-1} \left(\mathrm{x}^i - \left(\mathrm{x}^j + \theta^{i,j}\right)\right) - d + tr\left(\left(P^j\right)^{-1} P^i\right) + \ln\left(\tfrac{\det P^j}{\det P^i}\right)\right]. \tag{11}$$

Then, the PDF minimizing the WKL in (10) is again Gaussian with mean

$$\tilde{\mathrm{x}}_t^i(\Theta^i,\ell+1) = [\Omega_t^i(\ell+1)]^{-1} \sum_{j\in\mathcal{N}^i} \pi^{i,j} \Omega_t^j(\ell)[\hat{\mathrm{x}}_t^j(\ell) + \theta^{i,j}], \tag{12}$$

$\hat{\mathrm{x}}_t^j(\ell) = [\Omega_t^j(\ell)]^{-1} q_t^j(\ell)$, and inverse covariance $\Omega_t^i(\ell+1)$ independent of Θ^i.

In view of (11) and (12), it is an easy matter to see that the loss function $J_t^i(\Theta^i,\ell)$ can be written as the quadratic form

$$J_t^i(\Theta^i,\ell) = (\Theta^i)^T \Phi_t^i(\ell) \Theta^i + 2(\Theta^i)^T \varphi_t^i(\ell) + c_t^i(\ell) \tag{13}$$

for suitable $\Phi_t^i(\ell), \varphi_t^i(\ell), c_t^i(\ell)$ independent of Θ^i. In particular, it can be checked that

$$\Phi_t^i(\ell) = \frac{1}{2}\left[\Psi_t^i(\ell) \, E^i \left(\Omega_t^i(\ell+1)\right)^{-1} \left(E^i\right)^T \Psi_t^i(\ell) - \Psi_t^i(\ell)\right] \tag{14}$$

$$\varphi_t^i(\ell) = \Phi_t^i(\ell) \left[\mathbf{x}_t^i(\ell) - E^i \hat{\mathrm{x}}_t^i(\ell)\right] \tag{15}$$

where

$$\Psi_t^i(\ell) = block - diag\left(\pi^{i,j} \Omega_t^j(\ell), j \in \mathcal{N}^i \backslash \{i\}\right) \tag{16}$$

$$E^i = col\,\underbrace{(I_4, \cdots, I_4)}_{|\mathcal{N}^i|-1 \; times} \tag{17}$$

$$\mathbf{x}_t^i(\ell) = col\left(\hat{\mathrm{x}}_t^j(\ell), j \in \mathcal{N}^i \backslash \{i\}\right) \tag{18}$$

and $block - diag$ denotes the block diagonal matrix obtained from its arguments.

Proof: According to the definitions of (16) to (18), we have

$$\begin{aligned}
\Omega_t^i\left(\ell+1\right) &= \left(E^i\right)^T \Psi_t^i\left(\ell\right) E^i + \pi^{i,i}\Omega_t^i\left(\ell\right) \\
\tilde{\mathrm{x}}_t^i\left(\Theta^i, \ell+1\right) &= \left(\Omega_t^i\left(\ell+1\right)\right)^{-1}\left[\left(E^i\right)^T \Psi_t^i\left(\ell\right)\left(\mathbf{x}_t^i\left(\ell\right)+\Theta^i\right) + \pi^{i,i}\Omega_t^i\left(\ell\right)\hat{\mathrm{x}}_t^i\left(\ell\right)\right]
\end{aligned}$$

Then the cost $J_t^i(\Theta^i, \ell)$ is computed as

$$\begin{aligned}
&J_t^i(\Theta^i, \ell) \\
&= \sum_{j\in\mathcal{N}^i} \pi^{i,j} D_{KL}\left(\tilde{p}_t^i\left(\cdot,\ell+1\right)\middle\|\, p_t^{i,j}\left(\cdot,\ell\right)\right) \\
&= \frac{1}{2}\sum_{j\in\mathcal{N}^i}\left[\left(\tilde{\mathrm{x}}_t^i\left(\Theta^i,\ell+1\right) - \left(\hat{\mathrm{x}}_t^j\left(\ell\right)+\theta^{i,j}\right)\right)^T \pi^{i,j}\Omega_t^j\left(\ell\right)\left(\tilde{\mathrm{x}}_t^i\left(\Theta^i,\ell+1\right) - \left(\hat{\mathrm{x}}_t^j\left(\ell\right)+\theta^{i,j}\right)\right)\right] \\
&= \frac{1}{2}\left\{\left(\tilde{\mathrm{x}}_t^i\left(\Theta^i,\ell+1\right)\right)^T \Omega_t^i\left(\ell+1\right)\left(\tilde{\mathrm{x}}_t^i\left(\Theta^i,\ell+1\right)\right)\right. \\
&\quad \left. - \sum_{j\in\mathcal{N}^i\backslash\{i\}}\left(\left(\hat{\mathrm{x}}_t^j\left(\ell\right)+\theta^{i,j}\right)\right)^T \pi^{i,j}\Omega_t^j\left(\ell\right)\left(\hat{\mathrm{x}}_t^j\left(\ell\right)+\theta^{i,j}\right) + \left(\hat{\mathrm{x}}_t^i\left(\ell\right)\right)^T \pi^{i,i}\Omega_t^i\left(\ell\right)\hat{\mathrm{x}}_t^i\left(\ell\right)\right\} + C \\
&= \frac{1}{2}\left\{\left[\left(E^i\right)^T \Psi_t^i\left(\ell\right)\left(\mathbf{x}_t^i\left(\ell\right)+\Theta^i\right) + \pi^{i,i}\Omega_t^i\left(\ell\right)\hat{\mathrm{x}}_t^i\left(\ell\right)\right]^T \left(\Omega_t^i\left(\ell+1\right)\right)^{-1}\right. \\
&\quad \times\left[\left(E^i\right)^T \Psi_t^i\left(\ell\right)\left(\mathbf{x}_t^i\left(\ell\right)+\Theta^i\right) + \pi^{i,i}\Omega_t^i\left(\ell\right)\hat{\mathrm{x}}_t^i\left(\ell\right)\right] - \left[\left(\mathbf{x}_t^i\left(\ell\right)+\Theta^i\right)\right]^T \Psi_t^i\left[\left(\mathbf{x}_t^i\left(\ell\right)+\Theta^i\right)\right] \\
&\quad \left. - \left(\hat{\mathrm{x}}_t^i\left(\ell\right)\right)^T \pi^{i,i}\Omega_t^i\left(\ell\right)\hat{\mathrm{x}}_t^i\left(\ell\right)\right\} + C \\
&= \left(\Theta^i + \mathbf{x}_t^i\left(\ell\right)\right)^T \Phi_t^i\left(\ell\right)\left(\Theta^i + \mathbf{x}_t^i\left(\ell\right)\right) + 2\left(\Theta^i + \mathbf{x}_t^i\left(\ell\right)\right)^T \frac{1}{2}\Psi_t^i\left(\ell\right) E^i\left(\Omega_t^i\left(\ell+1\right)\right)^{-1}\pi^{i,i}\Omega_t^i\left(\ell\right)\hat{\mathrm{x}}_t^i\left(\ell\right) + C \\
&= \left[\Theta^i + \mathbf{x}_t^i\left(\ell\right) + \frac{1}{2}\left(\Phi_t^i\left(\ell\right)\right)^{-1}\Psi_t^i\left(\ell\right) E^i\left(\Omega_t^i\left(\ell+1\right)\right)^{-1}\pi^{i,i}\Omega_t^i\left(\ell\right)\hat{\mathrm{x}}_t^i\left(\ell\right)\right]^T \Phi_t^i\left(\ell\right) \\
&\quad \times\left[\Theta^i + \mathbf{x}_t^i\left(\ell\right) + \frac{1}{2}\left(\Phi_t^i\left(\ell\right)\right)^{-1}\Psi_t^i\left(\ell\right) E^i\left(\Omega_t^i\left(\ell+1\right)\right)^{-1}\pi^{i,i}\Omega_t^i\left(\ell\right)\hat{\mathrm{x}}_t^i\left(\ell\right)\right] + C \qquad (19)
\end{aligned}$$

According to the *Matrix Inversion Lemma*, we have

$$\begin{aligned}
\left(\Phi_t^i\left(\ell\right)\right)^{-1} &= 2\left[\Psi_t^i\left(\ell\right) E^i\left(\Omega_t^i\left(\ell+1\right)\right)^{-1}\left(E^i\right)^T \Psi_t^i\left(\ell\right) - \Psi_t^i\left(\ell\right)\right]^{-1} \\
&= -2\left\{\left(\Psi_t^i\left(\ell\right)\right)^{-1} + E^i\left[\Omega_t^i\left(\ell+1\right) - \left(E^i\right)^T \Psi_t^i\left(\ell\right) E^i\right]\left(E^i\right)^T\right\} \\
&= -2\left\{\left(\Psi_t^i\left(\ell\right)\right)^{-1} + E^i\left(\pi^{i,i}\Omega_t^i\left(\ell\right)\right)^{-1}\left(E^i\right)^T\right\} \qquad (20)
\end{aligned}$$

And

$$
\begin{aligned}
&\left(\Phi_t^i\left(\ell\right)\right)^{-1}\Psi_t^i\left(\ell\right)E^i\left(\Omega_t^i\left(\ell+1\right)\right)^{-1}\pi^{i,i}\Omega_t^i\left(\ell\right)\hat{\mathrm{x}}_t^i\left(\ell\right)\\
&=-2\left\{\left(\Psi_t^i\left(\ell\right)\right)^{-1}+E^i\left(\pi^{i,i}\Omega_t^i\left(\ell\right)\right)^{-1}\left(E^i\right)^T\right\}\Psi_t^i\left(\ell\right)E^i\left(\Omega_t^i\left(\ell+1\right)\right)^{-1}\pi^{i,i}\Omega_t^i\left(\ell\right)\hat{\mathrm{x}}_t^i\left(\ell\right)\\
&=-2\left\{E^i\left(\Omega_t^i\left(\ell+1\right)\right)^{-1}\pi^{i,i}\Omega_t^i\left(\ell\right)\hat{\mathrm{x}}_t^i\left(\ell\right)\right.\\
&\quad\left.+\,E^i\left(\pi^{i,i}\Omega_t^i\left(\ell\right)\right)^{-1}\left[\left(\Omega_t^i\left(\ell+1\right)\right)-\pi^{i,i}\Omega_t^i\left(\ell\right)\right]\left(\Omega_t^i\left(\ell+1\right)\right)^{-1}\pi^{i,i}\Omega_t^i\left(\ell\right)\hat{\mathrm{x}}_t^i\left(\ell\right)\right\}\\
&=-2E^i\hat{\mathrm{x}}_t^i\left(\ell\right)
\end{aligned}
\tag{21}
$$

Consequently, (19) can be rewritten as

$$
\begin{aligned}
J_t^i\left(\Theta^i,\ell\right)&=\left[\Theta^i+\mathbf{x}_t^i\left(\ell\right)-E^i\hat{\mathrm{x}}_t^i\left(\ell\right)\right]^T\Phi_t^i\left(\ell\right)\left[\Theta^i+\mathbf{x}_t^i\left(\ell\right)-E^i\hat{\mathrm{x}}_t^i\left(\ell\right)\right]\\
&=(\Theta^i)^T\Phi_t^i(\ell)\Theta^i+2(\Theta^i)^T\varphi_t^i(\ell)+c_t^i(\ell)
\end{aligned}
\tag{22}
$$

As the result, (14) and (15) are proved.

□

In order to obtain a reliable estimate of the drift parameters, we follow the strategy usually adopted in recursive parameter estimation and consider a total loss function up to the current time instant/consensus step defined recursively as

$$
\bar{J}_t^i(\Theta^i,\ell+1)=\lambda\bar{J}_t^i(\Theta^i,\ell)+J_t^i(\Theta^i,\ell)
$$

where, in accordance with the algorithm of Table I, we have $\bar{J}_{t+1}^i(\Theta^i,0)=\bar{J}_t^i(\Theta^i,L)$ and the scalar $\lambda\in(0,1)$ is a forgetting factor used to discount the past (since the quality of the local estimate of the state vector improves with time, it is reasonable to weight less the past loss functions). As it can be easily checked, recursive minimization of $\bar{J}_t^i(\Theta^i,\ell)$ gives rise to a standard recursive least squares (RLS) algorithm of the form

$$
\begin{array}{rcl}
\hat{\Theta}_t^i(\ell+1) & = & \hat{\Theta}_t^i(\ell)-[\bar{\Phi}_t^i(\ell+1)]^{-1}[\varphi_t^i(\ell)+\Phi_t^i(\ell)\hat{\Theta}_t^i(\ell)]\\
\bar{\Phi}_t^i(\ell+1) & = & \lambda\bar{\Phi}_t^i(\ell)+\Phi_t^i(\ell)\,.
\end{array}
\tag{23}
$$

Combining the recursive drift parameter estimation algorithm (23) with the consensus Kalman filter of Table I, we get the JTTSL-CSKF algorithm reported in Table 2.

Remark 4.1 *When the computational capabilities of each single sensor are low, several simplifications of the proposed solution are possible. For instance, instead of using an RLS algorithm for updating the estimate which requires the storage of a possibly large matrix, we can simply use an online optimization technique of the form*

$$
\hat{\Theta}_t^i(\ell+1)=\hat{\Theta}_t^i(\ell)-\gamma_t(\ell)\ \nabla_{\Theta^i}J_t^i(\Theta^i,\ell)\big|_{\Theta^i=\hat{\Theta}_t^i(\ell)}
\tag{24}
$$

where $\gamma_t(\ell)$ is a suitable stepsize. Another possible simplification in order to alleviate the computational burden amounts to updating the estimates of the drift parameters only once at each sampling interval, for instance ofter the last consensus step L. This approximated solution has been employed in the simulation results of the following section.

Table 2: JTTSL-CSKF (node i, time t)

```
Correction
  Perform the correction step as in table I
Consensus and calibration
  Set $\hat{\Theta}_t^i(0) = \hat{\Theta}_{t-1}^i(L)$
  Set $\bar{\Phi}_t^i(0) = \bar{\Phi}_{t-1}^i(L)$
  For $\ell = 0, 1, \ldots, L-1$ do
    Exchange the information with its neighbors and
    compute $\hat{\Theta}_t^i(\ell+1)$ and $\bar{\Phi}_t^i(\ell+1)$ as in (23)
    Fuse the quantities $q_t^i(\ell)$ and $\Omega_t^i(\ell)$ according to
      $q_t^i(\ell+1) = \sum_{j \in \mathcal{N}^i} \pi^{i,j} [q_{t|t}^j(\ell) + \Omega_{t|t}^j(\ell)\, \hat{\theta}_t^{i,j}(\ell+1)]$
      $\Omega_t^i(\ell+1) = \sum_{j \in \mathcal{N}^i} \pi^{i,j} \Omega_{t|t}^j(\ell)$
  End for
  $\hat{\mathrm{x}}_{t|t}^i = [\Omega_{t|t}^i(L)]^{-1} q_t^i(L), \;\; \Omega_{t|t}^i = \Omega_t^i(L)$
Prediction
  Perform the prediction step as in table I
```

5 Simulation examples

In this section, the performance of the proposed JTTSL-CSKF algorithm is evaluated by means of simulation experiments. The target motion obeys the constant velocity model, which means that $f_t\left(\mathrm{x}_t^i\right) = A\mathrm{x}_t^i$ in (1) and matrices A, Q are given by

$$A = \begin{pmatrix} 1 & T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & T \\ 0 & 0 & 0 & 1 \end{pmatrix}, \; Q = \sigma_x^2 I_4 \tag{25}$$

where $\sigma_x = 10\,[m]$, $T = 1\,[s]$, and $\mathrm{x}_\mathrm{t}^\mathrm{i}$ represents the target state with respect to a coordinate system with origin at the location of node i. The algorithm is tested on both linear and nonlinear sensors with measurement functions in (2) given as follows

Linear sensor: $h_t^i\left(\mathrm{x}_t^i\right) = C^i \mathrm{x}_t^i$ where

$$C^i = \alpha^i \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, R^i = \sigma_y^2 I_2 \tag{26}$$

Nonlinear sensor:

$$h_t^i\left(\mathrm{x}_t^i\right) = \begin{pmatrix} \sqrt{\left(\xi_t^i\right)^2 + \left(\eta_t^i\right)^2} \\ \mathrm{atan}(\eta_t^i / \xi_t^i) \end{pmatrix}, \; R^i = \mathrm{diag}(\sigma_r^2, \sigma_\beta^2) \tag{27}$$

where α^i are randomly selected for each node within the interval $[0.75, 1.25]$, $\sigma_y = 30\,[m], \sigma_r = 15\,[m]$, and $\sigma_\beta = 0.5\,[^o]$. The proposed algorithm is also tested on two different types of networks: one with a tree topology and the other with cycles, both consisting of 9 nodes as shown in Fig. 1. The initial drift parameter estimates are set to $\hat{\theta}_0^{i,j} = 0, (i,j) \in \mathcal{A}$, in both networks. The stepsize is set as $\gamma_t = 1.2$ for the linear sensor case and, respectively, $\gamma_t = 20$ for the nonlinear sensor case.

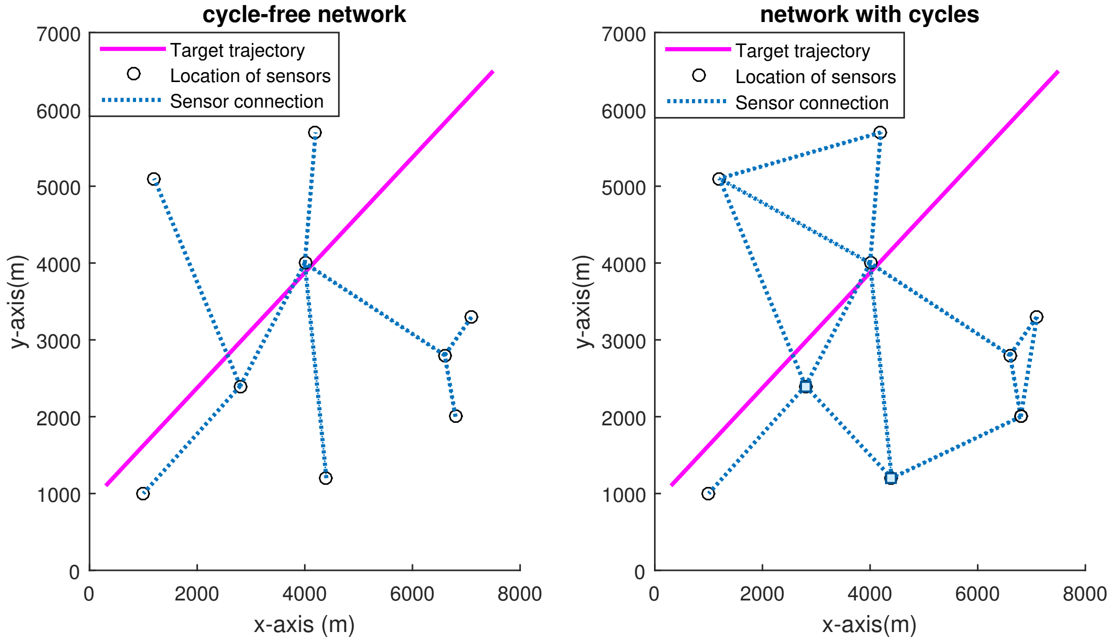

Figure 1: Tree network and network with cycles.

Figs. 2-5 analyse the self-localization performance by displaying the time evolution, in single typical realizations, of the drift parameter estimation errors $\tilde{\theta}_t^{i,j} \triangleq \theta^{i,j} - \hat{\theta}_t^{i,j} = \left[\tilde{\xi}_t^{i,j}, 0, \tilde{\eta}_t^{i,j}, 0\right]^T$ for the two considered networks of Fig. 1 and the two types of sensors in (26) and (27). It can be seen that in all cases the estimated drift parameters converge to their true values. Further, the effect of the stepsize γ_t on the speed of convergence is shown in Fig. 6 in the case of tree network and linear sensors for $L = 1$. It can be seen that the speed of self-localization of JTTSL-CSKF can be controlled by tuning γ_t, thus providing robustness in real applications.

Next, Figs. 7 and 8 analyse the target tracking performance by showing the time evolution of the target state Root Mean Square Error (RMSE), averaged over 200 independent Monte Carlo trials, in the two cases of linear (26) and, respectively, nonlinear sensors (27). In both figures, the RMSEs of the centralized (Extended) KF and the single-sensor (Extended) KF are also reported as reference lower and upper bounds for the performance of the JTTSL-CSKF algorithm. Further, performance of JTTSL-CSKF is also reported for the two network scenarios of Fig. 1 as well as for a single ($L = 1$) and multiple ($L = 10$) consensus steps. It can be seen that, at convergence, the performance of the proposed method stays in between the centralized and single sensor (E)KF for any L. Further, as expected, the RMSE of the proposed method is getting closer to the centralized (E)KF when the number L of consensus steps is increased. It can also be seen that the proposed method has wide potential applications, being immune to the type of network.

A final consideration is important. By looking at the simulation results, it is evident that during the initial learning phase the JTTSL-CSKF performs worse than the single-sensor filter. This is due to the fact that, at the beginning, the estimates of the drift parameters are not accurate enough and hence the fusion is not reliable. This drawback can be easily avoided with a slight modification of the proposed approach in which the fusion is performed at each time instant in order to update the estimates of the drift parameters, but the local estimates of the target state are replaced with the

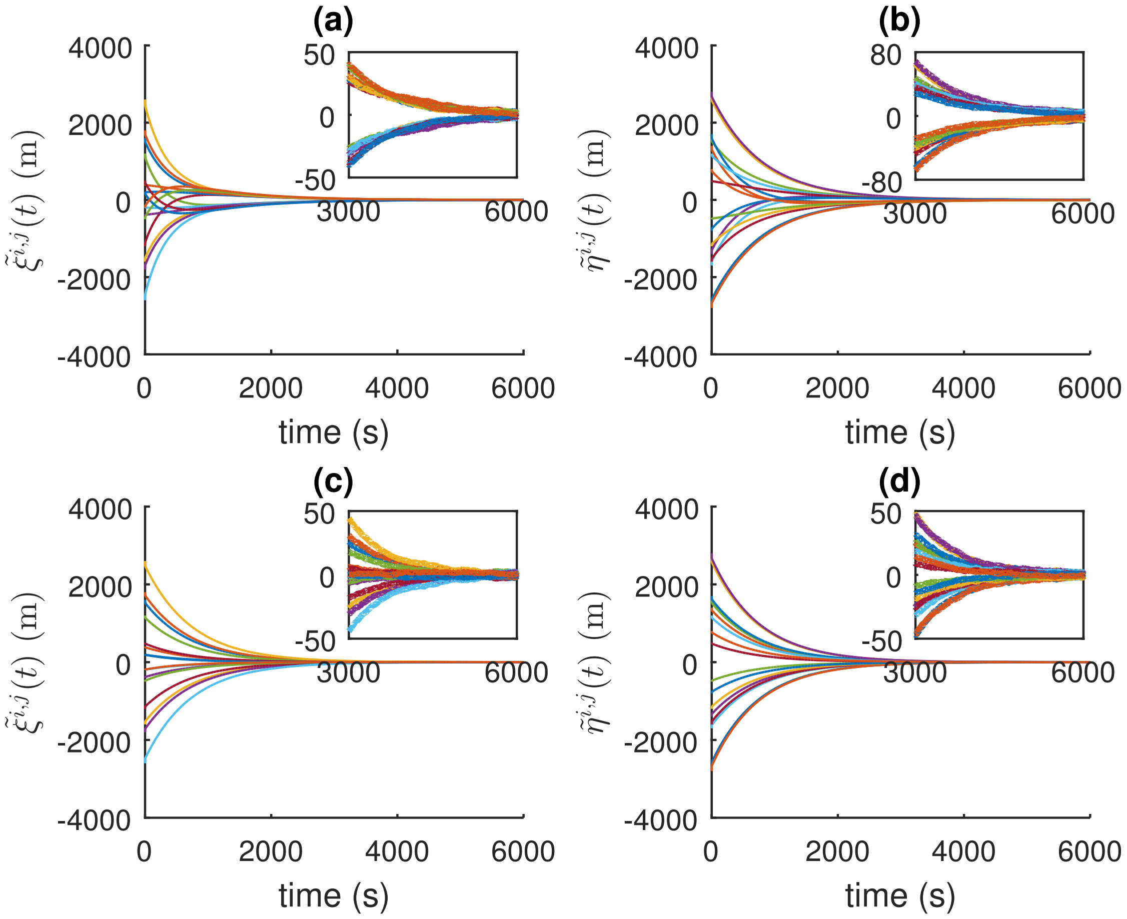

Figure 2: Time-behavior of $\tilde{\theta}^{i,j}(t), (i,j) \in \mathcal{A}$, in a tree network with linear sensors, for $L = 1$ (a), (b) and $L = 10$ (c), (d) consensus steps.

fused ones only when the drift parameter estimates are reliable enough (for example, when the loss function $J_t^i(\Theta^i, \ell)$ is below a given threshold).

6 Conclusion

The paper has addressed the important practical issue of fusing, in a distributed way, target tracking information over a sensor network with sensor nodes that get target measurements in their own local coordinate system and do not know their relative positions. The problem, referred to as *Joint Target Tracking and Sensor self-Localization* (JTTSL), has been solved by exploiting a novel computationally efficient consensus Kalman filtering approach by which each node is able to self-localize with respect to neighbors and thus correctly fuse target-state information. The effectiveness of the proposed approach has been successfully tested via simulation experiments concerning both networks with tree topology and networks with cycles as well as both linear and nonlinear sensors. Future developments of this work will possibly regard a theoretical convergence analysis as well as extensions in several directions like, e.g., multitarget scenario, mobile sensors, the case of unknown relative orientations among sensors and cooperative multirobot simultaneous localization and mapping.

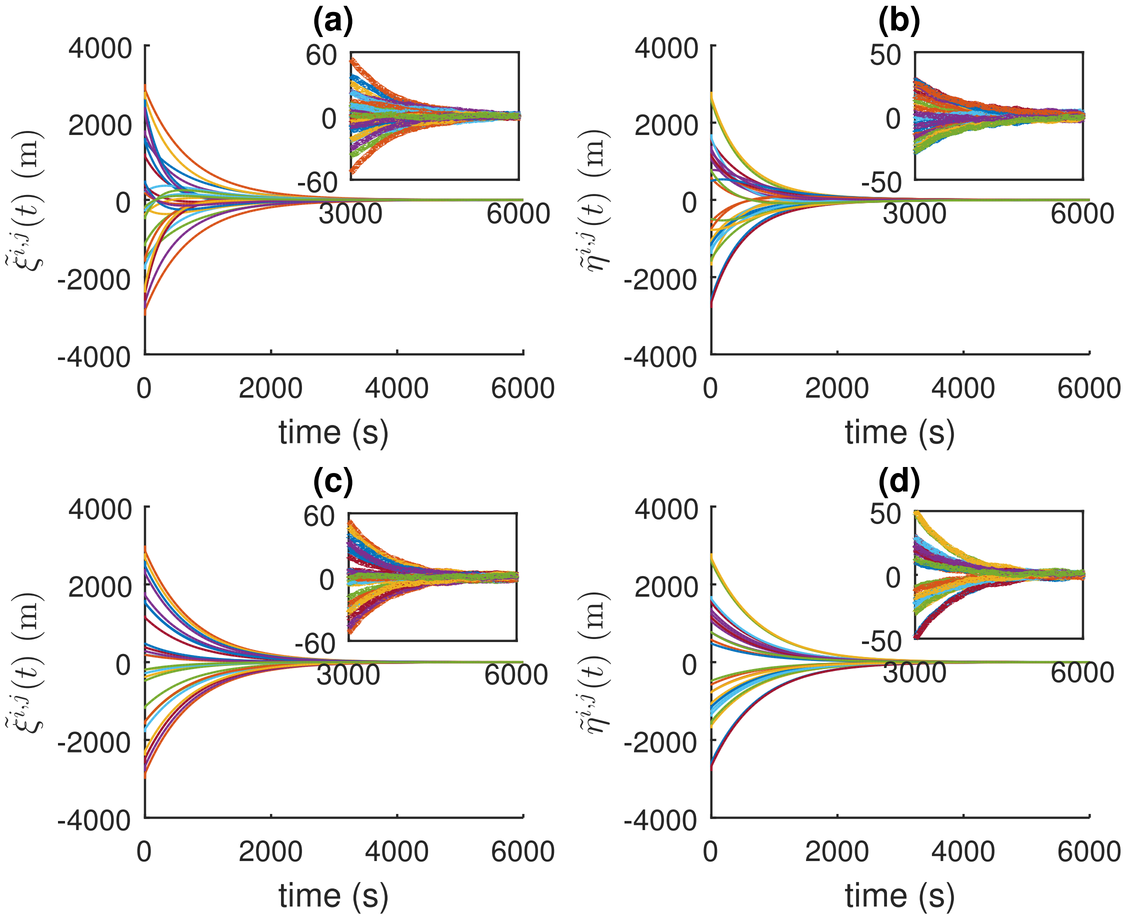

Figure 3: Time-behavior of $\tilde{\theta}^{i,j}(t), (i,j) \in \mathcal{A}$, in a network with cycles and linear sensors, for $L = 1$ (a), (b) and $L = 10$ (c), (d) consensus steps.

References

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

[2] A. T. Kamal, J. A. Farrell, and A. K. Roy-Chowdhury, "Information weighted consensus filters and their application in distributed camera networks", *IEEE Trans. on Automatic Control*, vol. 58, no. 12, pp. 3112–3125, 2013.

[3] S. Das and J. M. Moura, "Distributed Kalman filtering with dynamic observations consensus", *IEEE Trans. on Signal Processing*, vol. 63, no. 17, pp. 4458–4473, 2015.

[4] G. Battistelli, L. Chisci, and C. Fantacci, "Parallel consensus on likelihoods and priors for networked nonlinear filtering", *IEEE Signal Processing Letters*, vol. 21, no. 7, pp. 787–791, 2014.

[5] F. S. Cattivelli and A. H. Sayed, "Diffusion strategies for distributed Kalman filtering and smoothing", *IEEE Trans. on Automatic Control*, vol. 55, no. 9, pp. 2069–2084, 2010.

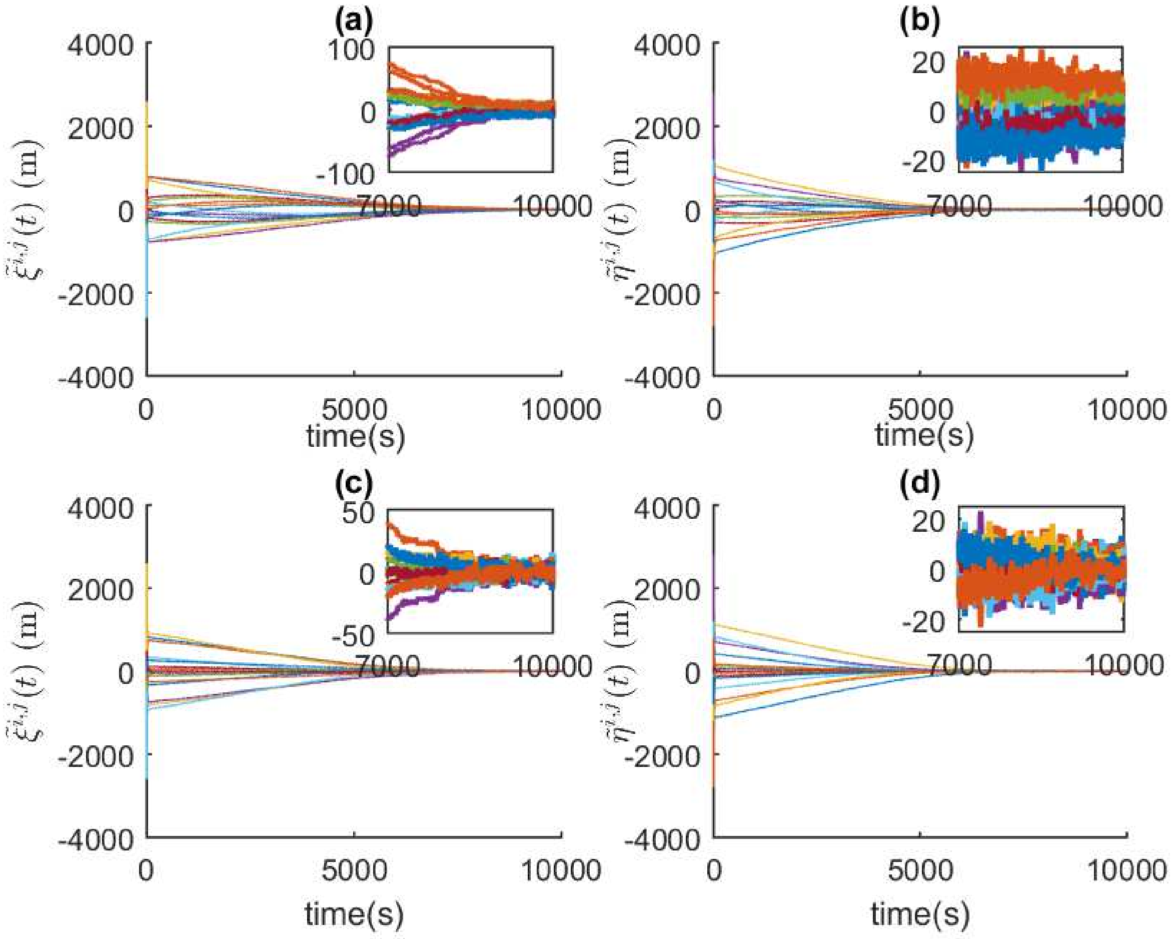

Figure 4: Time-behaviour of $\tilde{\theta}^{i,j}(t), (i,j) \in \mathcal{A}$, in a tree network with nonlinear sensors, for $L = 1$ (a), (b) and $L = 10$ (c), (d) consensus steps.

[6] ——, “Distributed nonlinear Kalman filtering with applications to wireless localization”, in *Proc. 35th IEEE Int. Conf. on Acoustics Speech and Signal Processing (ICASSP)*, pp. 3522–3525, Dallas, TX, USA, 2010.

[7] S. Čapkun, M. Hamdi, and J.-P. Hubaux, “GPS-free positioning in mobile ad hoc networks”, *Cluster Computing*, vol. 5, no. 2, pp. 157–167, 2002.

[8] G. Mao, B. Fidan, and B. D. Anderson, “Wireless sensor network localization techniques”, *Computer Networks*, vol. 51, no. 10, pp. 2529–2553, 2007.

[9] K. D. Frampton, “Acoustic self-localization in a distributed sensor network”, *IEEE Sensors Journal*, vol. 6, no. 1, pp. 166–172, 2006.

[10] F. Meyer, E. Riegler, O. Hlinka, and F. Hlawatsch, “Simultaneous distributed sensor self-localization and target tracking using belief propagation and likelihood consensus,” in *Proc. 46th Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, pp. 1212–1216, Pacific Grove, CA, USA, 2012.

[11] M. Sun, L. Yang, and K. Ho, “Accurate sequential self-localization of sensor nodes in closed-form”, *Signal Processing*, vol. 92, no. 12, pp. 2940–2951, 2012.

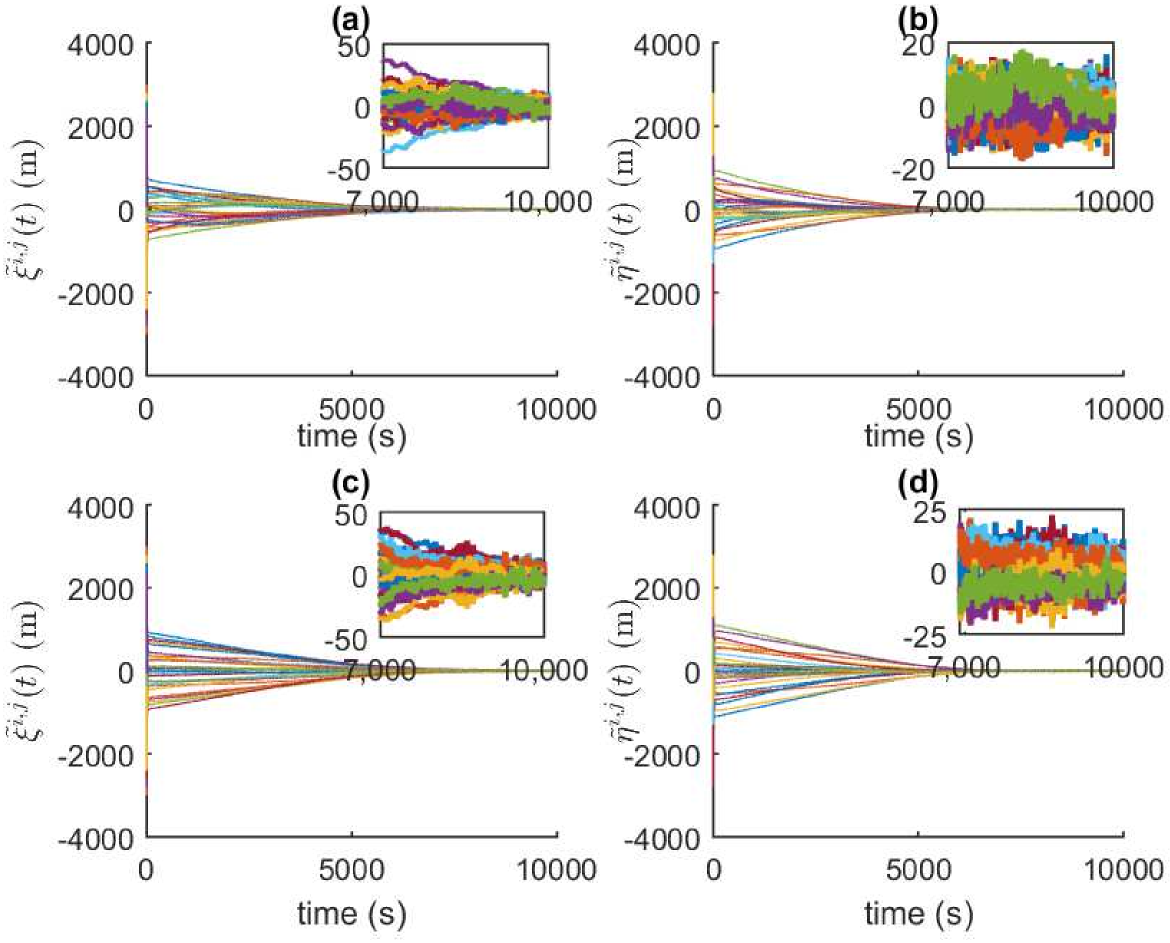

Figure 5: Time-behavior of $\tilde{\theta}^{i,j}(t), (i,j) \in \mathcal{A}$, in a network with cycles and nonlinear sensors, for $L = 1$ (a), (b) and $L = 10$ (c), (d) consensus steps.

[12] A. T. Ihler, J. W. Fisher, R. L. Moses, and A. S. Willsky, “Nonparametric belief propagation for self-localization of sensor networks”, *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 4, pp. 809–819, 2005.

[13] R. L. Moses, D. Krishnamurthy, and R. M. Patterson, “A self-localization method for wireless sensor networks”, *EURASIP Journal on Advances in Signal Processing*, vol. 2003, no. 4, pp. 839–843, 2003.

[14] N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal, “Locating the nodes: cooperative localization in wireless sensor networks”, *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 54–69, 2005.

[15] H.-J. Shao, X.-P. Zhang, and Z. Wang, “Efficient closed-form algorithms for AOA based self-localization of sensor nodes using auxiliary variables”, *IEEE Trans. on Signal Processing*, vol. 62, no. 10, pp. 2580–2594, 2014.

[16] U. A. Khan, S. Kar, and J. M. Moura, “Distributed sensor localization in random environments using minimal number of anchor nodes”, *IEEE Trans. on Signal Processing*, vol. 57, no. 5, pp. 2000–2016, 2009.

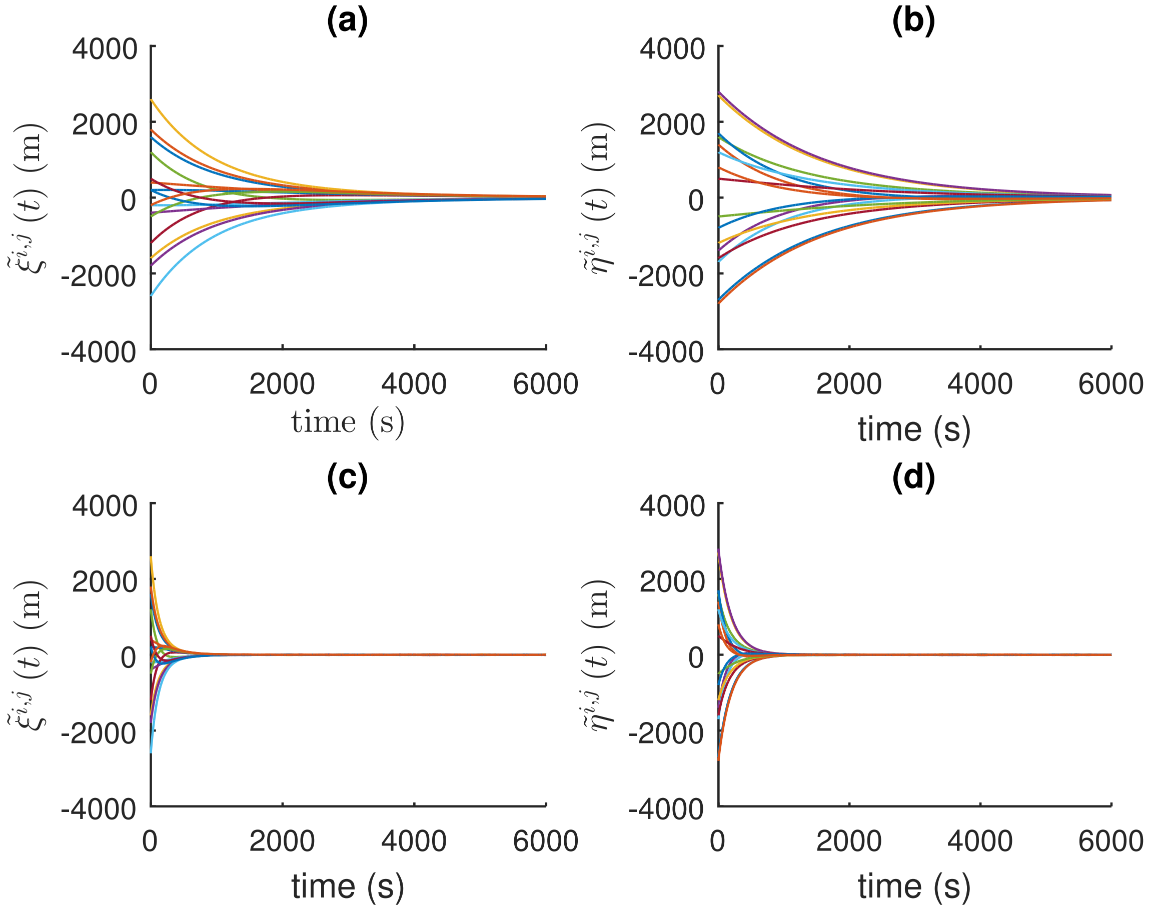

Figure 6: Stepsize affections of the speed of convergence, for $\gamma_t = 0.6$ (a),(b) and $\gamma_t = 4.8$ (c),(d).

[17] M. Vemula, M. F. Bugallo, and P. M. Djurić, "Sensor self-localization with beacon position uncertainty", *Signal Processing*, vol. 89, no. 6, pp. 1144–1154, 2009.

[18] V. Cevher and J. H. McClellan, "Acoustic node calibration using a moving source", *IEEE Trans. on Aerospace and Electronic Systems*, vol. 42, no. 2, pp. 585–600, 2006.

[19] X. Chen, A. Edelstein, Y. Li, M. Coates, M. Rabbat, and A. Men, "Sequential Monte Carlo for simultaneous passive device-free tracking and sensor localization using received signal strength measurements," in *Proc. 10th Int. Conf. on Information Processing in Sensor Networks (IPSN)*, pp. 342–353, Chicago, IL, USA, 2011.

[20] C. Pedersen, T. Pedersen, and B. H. Fleury, "A variational message passing algorithm for sensor self-localization in wireless networks", in *Proc. IEEE Int. Symposium on Information Theory (ISIT)*, pp. 2158–2162, Saint Petersburg, Russia, 2011.

[21] N. Kantas, S. S. Singh, and A. Doucet, "Distributed maximum likelihood for simultaneous self-localization and tracking in sensor networks", *IEEE Trans. on Signal Processing*, vol. 60, no. 10, pp. 5038–5047, 2012.

[22] M. Uney, B. Mulgrew, and D. E. Clark, "A cooperative approach to sensor localisation in distributed fusion networks., *IEEE Trans. on Signal Processing*, vol. 64, no. 5, pp. 1187–1199, 2016.

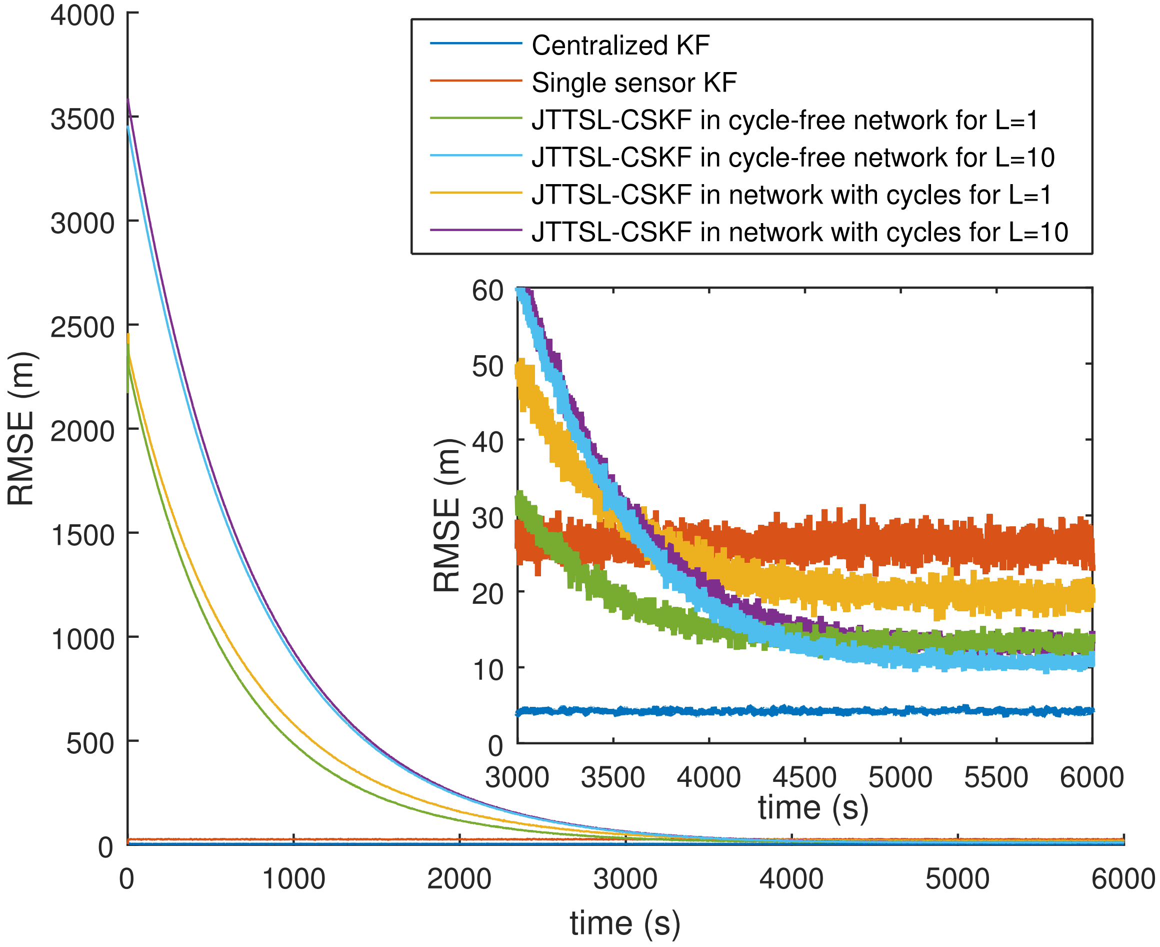

Figure 7: Time-behavior of the target state RMSE under linear observations.

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

[24] L. Xiao, S. Boyd, and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. 4th Int. Symposium on Information Processing in Sensor Networks (IPSN)*, pp. 1-9, Los Angeles, CA, USA, 2005.

[25] R. Carli, A. Chiuso, L. Schenato, and S. Zampieri, "Distributed Kalman filtering based on consensus strategies", *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 4, pp. 622-633, 2008.

[26] R. Olfati-Saber, "Kalman-consensus filter: Optimality, stability, and performance," in *Proc. 48th IEEE Conference on Decision and Control (CDC)* held jointly with the *28th Chinese Control Conference (CCC)*, pp. 7036–7042, Shanghai, China, 2009.

[27] S. S. Stanković, M. S. Stanković, and D. M. Stipanović, "Consensus based overlapping decentralized estimation with missing observations and communication faults", *Automatica*, vol. 45, no. 6, pp. 1397–1406, 2009.

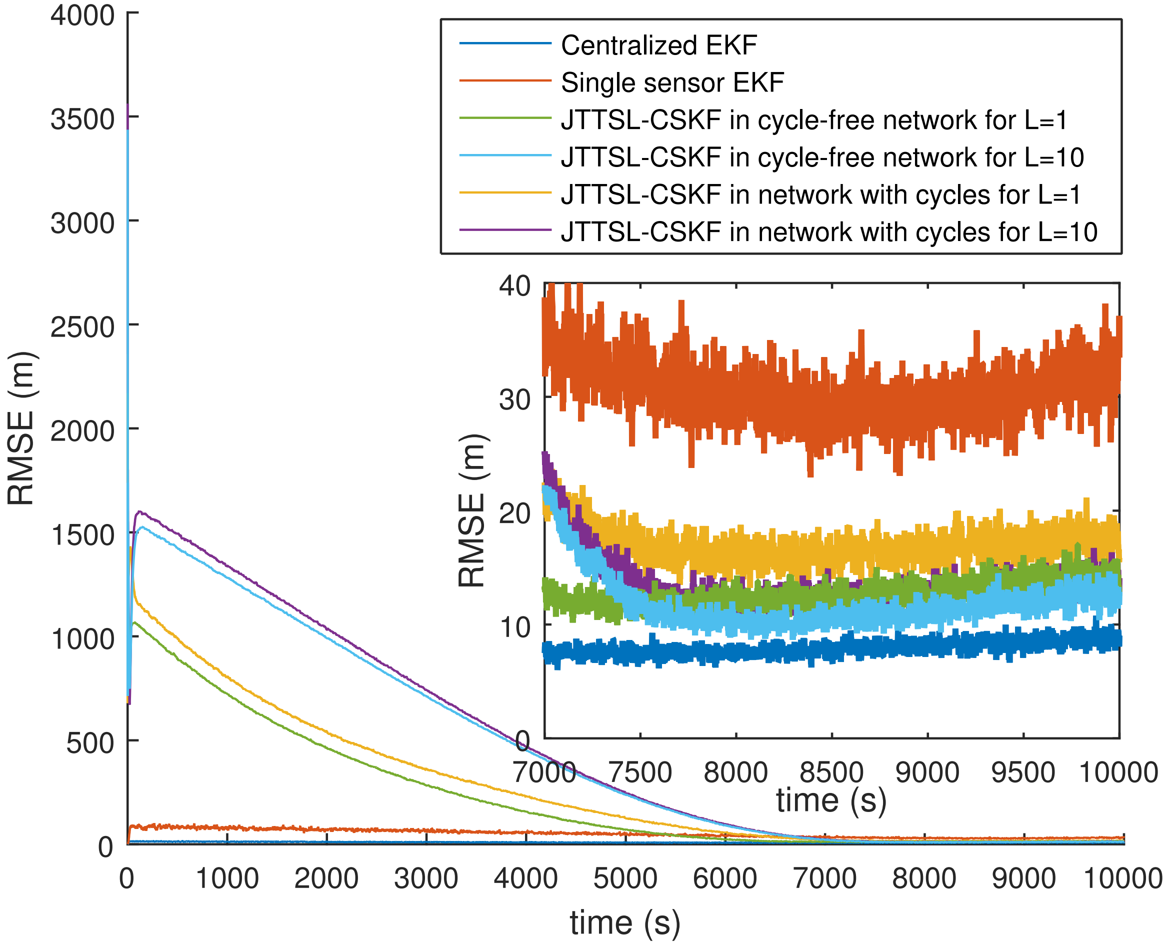

Figure 8: Time-behavior of the target state RMSE under nonlinear observations.

[28] G. Battistelli, L. Chisci, G. Mugnai, A. Farina, and A. Graziano, "Consensus-based linear and nonlinear filtering", *IEEE Trans. on Automatic Control*, vol. 60, no. 5, pp. 1410–1415, 2015.

[29] G. Battistelli and L. Chisci, "Kullback–Leibler average, consensus on probability densities, and distributed state estimation with guaranteed stability", *Automatica*, vol. 50, no. 3, pp. 707–718, 2014.