Information Fusion 99 (2023) 101857

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

Perception-latency aware distributed target tracking☆

Rodrigo Aldana-López, Rosario Aragüés, Carlos Sagüés *

Departamento de Informatica e Ingenieria de Sistemas (DIIS), Universidad de Zaragoza, Zaragoza 50018, Spain
Instituto de Investigacion en Ingenieria de Aragon (I3A), Universidad de Zaragoza, Zaragoza 50018, Spain

ARTICLE INFO

Keywords:
Perception-latency
Target tracking
Mobile robot
Estimation fusion

ABSTRACT

This work is devoted to the problem of distributed target tracking when a team of robots detect the target through a variable perception-latency mechanism. A reference for the robots to track is constructed in terms of a desired formation around the estimation of the target position. However, it is noted that due to the perception-latency, classical estimation techniques have smoothness issues which prevent asymptotic stability for the formation control. We propose a near-optimal smooth-output estimator which circumvents this issue. Moreover, local estimations are fused using novel dynamic consensus techniques. The advantages of the proposal as well as a comparison with a non-smooth optimal alternative are discussed through simulation examples.

1. Introduction

Formation control of mobile robots for target tracking is a problem of great interest nowadays with a wide range of applications such as automated surveillance [1,2], aerial filming using drones [3], intelligent transportation [4], among other examples. In the most general setting, the goal is to coordinate a team of robots in order to maintain a formation around a moving target. This can be performed by detecting the target locally at each robot, fuse estimations in a decentralized fashion, and then use the resulting target estimation as a reference to achieve the formation.

Target localization is usually performed by the robots using a Visual Target Detector (VTD). Popular recent examples of VTD algorithms include [5–9], which mostly rely on Deep Learning (DL) models. For these type of algorithms, the term perception-latency is used to denote the computing time employed starting from the sampling instant for the vision sensors to the moment in which the vision algorithms produce an output [10]. Note that the quality of the perception output can be improved by increasing the image resolution, the number of features, using more complex feature descriptors or using different DL models depending on the vision algorithm. However, these improvements come at the expense of increasing the perception-latency as well. Such perception-latency and detection quality trade-off has been characterized for some methods as reported in [11].

In general, resource-constrained robots will trade-off between accuracy and speed for target detection. Some approaches allow to choose the perception latency as requested online for resource economy or to improve the accumulated estimation error. This can be performed either by using a bank of perception methods [10,12,13], or by using Anytime Neural Networks (ANN) as in [14,15]. In some cases, the perception latency may be too large when compared with the actual dynamics of the robot. Therefore, state estimation and prediction must be performed locally in order to obtain a target estimate in-between measurements. However, due to the hybrid nature of the problem, if standard techniques such as a Kalman filter are used, the resulting estimation will be discontinuous whenever a new measurement is used for correction. As we show later in this manuscript, if such estimation is used as a reference for the robot to track, the discontinuities in the trajectory may prevent asymptotic stability for the formation control task.

A similar issue occurs in the multi-robot setting. Several strategies have been proposed in the literature in order to fuse local Kalman filter estimations either using a fusion center as in [16,17] or in a decentralized fashion as is of interest in this work. In this context, in [18] local Kalman filter estimations are combined using discrete-time static consensus protocols, ignoring cross-correlations between agents. However, these only achieve exact convergence when an infinite number of iterations for the consensus protocol are performed at each sampling step.

☆ This work was supported via projects PID2021-124137OB-I00 and TED2021-130224B-I00 funded by MCIN/AEI/10.13039/501100011033, by ERDF A way of making Europe and by the European Union NextGenerationEU/PRTR, by the Gobierno de Aragón, Spain under Project DGA T45_23R, by the Universidad de Zaragoza and Banco Santander, by the Consejo Nacional de Ciencia y Tecnología (CONACYT-Mexico) with grant number 739841.

* Correspondence to: Departamento de Informatica e Ingenieria de Sistemas (DIIS). Instituto de Investigacion en Ingenieria de Aragon (I3A). Universidad de Zaragoza, Calle María de Luna, 1, 50018 Zaragoza, spain.

E-mail addresses: rodrigo.aldana.lopez@gmail.com (R. Aldana-López), raragues@unizar.es (R. Aragüés), csagues@unizar.es (C. Sagüés).

https://doi.org/10.1016/j.inffus.2023.101857
Received 8 July 2022; Received in revised form 20 February 2023; Accepted 26 May 2023
Available online 1 June 2023

In contrast, the works [19,20] use linear dynamic consensus filters [21] which do not have the previously mentioned issue. Despite this, as discussed in [22], linear dynamic consensus protocols cannot exhibit exact convergence with persistently varying references. Other recent fusion strategies have been proposed which improve the fusion quality by using the covariance intersection method [23–25]. The issue with all of the previously mentioned fusion methods is that they mainly work in discrete time. This means that synchronous measurements and updates might be assumed, which is incompatible with the perception-latency setting. On the other hand, the discontinuity issue in the estimations prevails in all these methods, since a continuous time-prediction must be computed in-between filter updates as well.

Motivated by this discussion we contribute with a hybrid fusion framework, which obtains near-optimal smooth estimations for the target, regardless of the perception-latency mechanism. This means that the control design is decoupled from the perception configuration decisions, avoiding any stability issues arising from the discontinuous estimations. Moreover, we leverage the recently developed dynamic consensus protocol [26] which, in contrast to other linear dynamic consensus protocols, achieves exact convergence under mild assumptions. The advantages of the proposal as well as a comparison with a non-smooth alternative are discussed through simulation examples.

1.1. Notation

Let $\binom{\mu}{\nu}$ denote the binomial coefficient. Let $\text{sign}(x) = 1$ if $x > 0$, $\text{sign}(x) = -1$ if $x < 0$ and $\text{sign}(0) = 0$. Moreover, if $x \in \mathbb{R}$, let $\lceil x \rfloor^\alpha := |x|^\alpha \text{sign}(x)$ for $\alpha > 0$ and $\lceil x \rfloor^0 := \text{sign}(x)$. When $\mathbf{x} = [x_1, \dots, x_n]^T \in \mathbb{R}^n$, then $\lceil \mathbf{x} \rfloor^\alpha := \left[\lceil x_1 \rfloor^\alpha, \dots, \lceil x_n \rfloor^\alpha\right]^T$ for $\alpha \geq 0$ and similarly for $\lceil \mathbf{A} \rfloor^\alpha, \mathbf{A} \in \mathbb{R}^{n\times n}$. Let $\mathbb{E}\{\bullet\}, \text{cov}\{\bullet\}$ denote expectation and covariance operators respectively. For any function $u(t)$, let $u^{(\mu)}(t)$ for $\mu \in \mathbb{N}$ denote its μth derivative when it exists.

2. Problem statement

Consider a team of N mobile robots located at positions $\mathbf{p}_i(t) \in \mathbb{R}^n, i \in \{1, \dots, \mathsf{N}\}$ where n is the dimension of the workspace of the robots. For simplicity, each robot is modeled to be holonomic with mth order integrator dynamics

$$\dot{\mathbf{x}}_i(t) = \mathbf{A}\mathbf{x}_i(t) + \mathbf{B}\mathbf{u}_i(t), \tag{1}$$

where $\mathbf{x}_i(t) = [(\mathbf{p}_i^{(0)}(t))^\top, \dots, (\mathbf{p}_i^{(m-1)}(t))^\top]^\top$, $\mathbf{u}_i(t) \in \mathbb{R}^n$ is a local control input and $\mathbf{A}, \mathbf{B}$ are given in Appendix A. We assume that each robot is able to measure its own state $\mathbf{x}_i(t)$ using local sensors. The robots are able to share information between them according to a communication network modeled by an undirected graph $\mathcal{G}$. In addition, a target of interest with position $\mathbf{p}(t) \in \mathbb{R}^n$ is assumed to have dynamics with similar integrator dynamics as in (1). Moreover, since the input at the target is unknown, we model it in a stochastic fashion as

$$d\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t)dt + \mathbf{B}d\mathbf{u}(t) \tag{2}$$

where $\mathbf{p}(t) = \mathbf{C}\mathbf{x}(t)$ with $\mathbf{C}$ defined in Appendix A and $\mathbf{u}(t) \in \mathbb{R}^n$ is a n-dimensional Wiener processes with covariance $\text{cov}\{\mathbf{u}(s), \mathbf{u}(r)\} = \mathbf{W}\min(s, r)$ [27, Page 63]. As usual, the process $\mathbf{u}(t)$ models disturbances, unknown inputs at the target, and non-modeled dynamics.

Perception mechanism: The robot i uses available sensors such as vision, range, etc, to produce *raw measurements* of the environment. For each raw measurement, a *processed measurement* for the position of the target is obtained through a detection process at, perhaps non-uniform, processing instants $\boldsymbol{\tau}_i = \{\tau_{i,k}\}_{k=0}^\infty$, $\tau_{i,0} = 0$. Processed measurements are detections of $\mathbf{p}(\tau_{i,k}) = \mathbf{C}\mathbf{x}(\tau_{i,k})$. According to the current processing and energy budget of robot i, a perception latency $\Delta_{i,k} \in [\Delta_{\min}, \Delta_{\max}]$ with $\Delta_{\min}, \Delta_{\max} > 0$ is chosen for the detection process at $t = \tau_{i,k}$ and the processed measurement $\mathbf{z}_i(\tau_{i,k}) = \mathbf{C}\mathbf{x}(\tau_{i,k}) + \mathbf{v}_i(\tau_{i,k})$ is available at $t = \tau_{i,k} + \Delta_{i,k}$. Here, $\mathbf{v}_i(\tau_{i,k})$ is a Gaussian noise modeling the accuracy of the perception method. In general, $\mathbf{R}(\Delta_{i,k}) = \text{cov}\{\mathbf{v}_i(\tau_{i,k})\}$ decreases in magnitude as more processing time $\Delta_{i,k}$ is employed.

Remark 1. In order to focus in the estimation and information fusion aspects of this work, we consider that the process for which the perception latencies $\{\Delta_{i,k}\}_{k=0}^\infty$ are chosen at robot i is already given. For instance, some ideas from [10,28] can be applied where only D available perception methods can be used, and the perception-latency schedule $\Delta_{i,k} \in \{\Delta^j\}_{j=1}^D$ is chosen to minimize some local performance function.

The goal is for the robots to achieve a given formation provided known displacements $\mathbf{d}_1, \dots, \mathbf{d}_\mathsf{N} \in \mathbb{R}^n$ around the target. However, given that target localization is imperfect, an approximate scheme must be adopted. The strategy is to obtain local estimates $\hat{\mathbf{x}}_i(t)$ of the target state at each robot given the local perception-latency schedule $\{\Delta_{i,k}\}_{k=0}^\infty$, and then combine them into a single global estimate $\bar{\mathbf{p}}_\mathsf{G}(t)$ for the trajectory $\mathbf{p}(t)$ in a distributed and decentralized fashion. Then, achieve a formation around $\bar{\mathbf{p}}_\mathsf{G}(t)$ instead. This is, given a reference $\mathbf{p}_i^\mathsf{r}(t) = \bar{\mathbf{p}}_\mathsf{G}(t) + \mathbf{d}_i$, reach

$$\textbf{Goal}: \lim_{t\to\infty} \|\mathbf{p}_i(t) - \mathbf{p}_i^\mathsf{r}(t)\| = 0, \forall i \in \{1, \dots, \mathsf{N}\} \tag{3}$$

The reference should be smooth enough for $(\mathbf{p}_i^\mathsf{r})^{(m)}(t), \forall t \geq 0$ to exist and achieve trajectory tracking at each robot. Nonetheless, this is incompatible with existing optimal estimation techniques taking into account the hybrid nature of the problem.

2.1. Solution outline

To solve the problem, the proposal contains the following ingredients:

- **Smooth-output estimation**: We propose an alternative to classical filtering obtaining a smooth trajectory of the estimate $\hat{\mathbf{x}}_i(t)$ of $\mathbf{x}(t)$ locally at robot i.
- **Estimation fusion**: A distributed and decentralized consensus filter is used to fuse the local information of $\hat{\mathbf{x}}_i(t)$ into the single estimate $\bar{\mathbf{p}}_\mathsf{G}(t)$ of $\mathbf{p}(t)$.
- **Formation control:** A local controller $\mathbf{u}_i(t)$ is designed for the robot i to achieve the formation goal in (3) using the estimation framework.

Fig. 1 presents a high-level outline of the processing flow for our proposal. In particular, we assume that the target detection process is already given, taking raw measurements of the environment and producing detections for the target as $\mathbf{z}_i(\tau_{i,k})$. The Smooth-Output Estimation (SOE) block is described in detail in Section 3, which takes the output of the detection process, and computes smooth information vector and matrix $\hat{\mathbf{y}}_i(t), \hat{\mathbf{Q}}_i(t)$ associated with the estimation $\hat{\mathbf{x}}_i(t)$ and its covariance $\hat{\mathbf{P}}_i(t)$ for the state of the target. The fusion algorithm is described in detail in Section 4, and is composed of two modules. First, a consensus stage computes a fused version of the information vector and matrix $\mathbf{y}_{i,0}(t), \mathbf{Q}_{i,0}(t)$ for the target state as well as its derivatives $\{\mathbf{y}_{i,\mu}(t), \mathbf{Q}_{i,\mu}(t)\}_{\mu=1}^m$. Then, the fusion output stage uses them to compute the actual joint estimation $\bar{\mathbf{p}}_\mathsf{G}(t)$ for the target position $\mathbf{p}(t)$ and its first m derivatives. These estimations are stored in the local variables $\{\mathbf{p}_{i,\mu}(t)\}_{\mu=0}^m$. Finally, these signals are used to compute a local trajectory tracking control for each robot, which is described in Section 5.

The main advantage of the architecture depicted in Fig. 1, is that the smooth estimation block allows the fusion and control blocks to be designed independently from the detection procedure, providing more versatility than in prior literature. In addition, the smooth output estimation and fusion greatly reduces the estimation and tracking errors as shown in the experiments provided in Section 6.

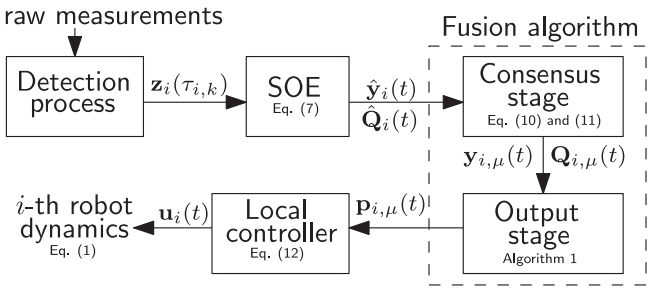

Fig. 1. High-level of the processing flow for our proposal as described in Section 2.1.

3. Smooth-output estimation

In this section, we focus on the local target estimation at each robot. To simplify the presentation, we drop the index i when there is no ambiguity given that all variables are assumed to be local. First, given a sequence of local sampling instants $\boldsymbol{\tau} = \{\tau_k\}_{k=0}^{\infty}$, adopt the notation $\mathbf{x}[k] := \mathbf{x}(\tau_k)$ and note that a sampled-data version of the target dynamics (2) for $\mathbf{x}(t)$ is obtained similarly as in [29, Section 4.5.2]:

$$\mathbf{x}(t) = \mathbf{A}_{\mathrm{d}}(t-\tau_k)\mathbf{x}[k] + \mathbf{u}_{\mathrm{d}}(t), \quad t \in [\tau_k, \tau_{k+1}] \tag{4}$$

where $\mathbf{u}_{\mathrm{d}}(t)$ is a normal random variable of zero mean and $\mathrm{cov}\{\mathbf{u}_{\mathrm{d}}(t)\} = \mathbf{W}_{\mathrm{d}}(t-\tau_k) = \int_0^t \mathbf{A}^{\mathrm{d}}(\tau)\mathbf{B}\mathbf{W}\mathbf{B}^\top\mathbf{A}^{\mathrm{d}}(\tau)^\top \mathrm{d}\tau$ with $\mathbf{A}^{\mathrm{d}}(\tau) = \exp(\mathbf{A}\tau)$.

Every robot can use its observations of the target $\mathbf{z}[0], \ldots, \mathbf{z}[k-1]$ and compute a causal optimal filter for (4) at the sampling instants $t \in \boldsymbol{\tau}$ of the form $\hat{\mathbf{x}}^*[k] = \mathbb{E}\{\mathbf{x}[k]|\mathbf{z}[0], \ldots, \mathbf{z}[k-1]\}$ which does not take into account $\mathbf{z}[k]$ due to the perception latency. By applying [27, Page 228 - Theorem 4.1] to system (4) evaluated at $t = \tau_{k+1}$, the recursive filter structure to obtain $\hat{\mathbf{x}}^*[k]$ and $\hat{\mathbf{P}}^*[k] = \mathrm{cov}\{\mathbf{x}[k] - \hat{\mathbf{x}}^*[k]\}$ has the following form:

$$\begin{aligned}
\mathbf{L}[k] &= \mathbf{A}_{\mathrm{d}}(\Delta_k)\hat{\mathbf{P}}^*[k]\mathbf{C}^\top\left(\mathbf{C}\hat{\mathbf{P}}^*[k]\mathbf{C}^\top + \mathbf{R}(\Delta_k)\right)^{-1} \\
\hat{\mathbf{x}}^*[k+1] &= \mathbf{A}_{\mathrm{d}}(\Delta_k)\hat{\mathbf{x}}^*[k] + \mathbf{L}[k]\left(\mathbf{z}[k] - \mathbf{C}\hat{\mathbf{x}}^*[k]\right) \\
\hat{\mathbf{P}}^*[k+1] &= \left(\mathbf{A}_{\mathrm{d}}(\Delta_k) - \mathbf{L}[k]\mathbf{C}\right)\hat{\mathbf{P}}^*[k]\left(\mathbf{A}_{\mathrm{d}}(\Delta_k) - \mathbf{L}[k]\mathbf{C}\right)^\top \\
&\quad + \mathbf{L}[k]\mathbf{R}(\Delta_k)\mathbf{L}[k]^\top + \mathbf{W}_{\mathrm{d}}(\Delta_k)
\end{aligned} \tag{5}$$

assuming knowledge of some initial values for $\hat{\mathbf{x}}^*[0], \hat{\mathbf{P}}^*[0]$. On the other hand, an optimal estimation $\hat{\mathbf{x}}^*(t|k) := \mathbb{E}\{\mathbf{x}(t)|\mathbf{z}[0], \ldots, \mathbf{z}[k-1]\}$ for $t \in (\tau_k, \tau_{k+1})$ can be obtained by using a model based prediction of (4) as:

$$\begin{aligned}
\hat{\mathbf{x}}^*(t|k) &= \mathbf{A}_{\mathrm{d}}(t-\tau_k)\hat{\mathbf{x}}^*[k] \\
\hat{\mathbf{P}}^*(t|k) &= \mathrm{cov}\{\mathbf{x}(t) - \hat{\mathbf{x}}^*(t|k)\} \\
&= \mathbf{A}_{\mathrm{d}}(t-\tau_k)\hat{\mathbf{P}}^*[k]\mathbf{A}_{\mathrm{d}}(t-\tau_k)^T + \mathbf{W}_{\mathrm{d}}(t-\tau_k)
\end{aligned} \tag{6}$$

Note that the expressions in (6) can be defined for all $t \geq 0$. However, we use $\hat{\mathbf{x}}^*(t)$ to refer to the whole optimal causal estimated trajectory constructed piecewise as $\hat{\mathbf{x}}^*(t) := \hat{\mathbf{x}}^*(t|k)$ and $\hat{\mathbf{P}}^*(t) := \hat{\mathbf{P}}^*(t|k)$ for $t \in [\tau_k, \tau_{k+1})$. Note also that (5) is a Kalman filter [27, Chapter 7]. Hence, the standard discrete-time Kalman filter coincides with the DOE at $t = \tau_k$.

Remark 2. Note that $\hat{\mathbf{x}}^*(t)$ is a smooth function of time for all $t \notin \boldsymbol{\tau}$. However, $\hat{\mathbf{x}}^*(t)$ may be discontinuous at $t \in \boldsymbol{\tau}$ due to the measurement corrections performed in (5). Hence we refer to $\hat{\mathbf{x}}^*(t)$ as a Discontinuous Optimal Estimation (DOE). If $\hat{\mathbf{x}}^*(t)$ is used to construct a reference for the robot, such discontinuities can cause persistent transients in the closed loop behavior of the robot compromising asymptotic stability. An example of this is shown in Section 6.1.

We propose a near-optimal alternative estimation $\hat{\mathbf{x}}(t)$ which does not have the discontinuity issue at $t \in \boldsymbol{\tau}$. The idea is to combine the current optimal estimate $\hat{\mathbf{x}}^*(t|k)$ for $t \in [\tau_k, \tau_{k+1})$ with the continued prediction from the previous estimate $\hat{\mathbf{x}}^*(t|k-1)$ for $t \geq \tau_k$, using a smooth transition between both.

In order to perform such transition, consider the following auxiliary function:

Definition 1 (*Transition Function*). $\eta(\bullet;\alpha) : [0,1] \to \mathbb{R}$ is a transition function if it complies that $\forall \mu \in \{1, \ldots, m\}$:

- **(i)** $\eta^{(\mu)}(t;\alpha)$ exists $\forall t \in [0,1]$.
- **(ii)** $\eta^{(\mu)}(0;\alpha) = \eta^{(\mu)}(1;\alpha) = 0$.
- **(iii)** $\eta(0;\alpha) = 0$, $\eta(1;\alpha) = 1$.
- **(iv)** $\lim_{\alpha\to 0}\eta(t,\alpha) = 1, \forall t \in (0,1]$

An example of a transition function is provided in Appendix B which can be assumed to be fixed as such through the rest of the manuscript. Hence, given $\alpha > 0$, a transition function, and by defining the information matrix $\hat{\mathbf{Q}}^*(t|k) = \hat{\mathbf{P}}^*(t|k)^{-1}$ and information vector $\hat{\mathbf{y}}^*(t|k) = \hat{\mathbf{Q}}^*(t|k)\hat{\mathbf{x}}^*(t|k)$, let the Smooth Output Estimation (SOE):

$$\begin{aligned}
\lambda_1(t|k) &= 1 - \eta((t-\tau_k)/\Delta_k;\alpha) \\
\lambda_2(t|k) &= \eta((t-\tau_k)/\Delta_k;\alpha) \\
\hat{\mathbf{Q}}(t) &= \lambda_1(t|k)\hat{\mathbf{Q}}^*(t|k-1) + \lambda_2(t|k)\hat{\mathbf{Q}}^*(t|k) \\
\hat{\mathbf{y}}(t) &= \lambda_1(t|k)\hat{\mathbf{y}}^*(t|k-1) + \lambda_2(t|k)\hat{\mathbf{y}}^*(t|k) \\
\hat{\mathbf{P}}(t) &= \hat{\mathbf{Q}}(t)^{-1} \\
\hat{\mathbf{x}}(t) &= \hat{\mathbf{Q}}(t)^{-1}\hat{\mathbf{y}}(t)
\end{aligned} \tag{7}$$

for $t \in [\tau_k, \tau_{k+1})$. Similarly as with $\hat{\mathbf{x}}^*(t)$, we drop the dependence on k in the notation for the SOE estimation $\hat{\mathbf{x}}(t)$ in order to refer to the whole sub-optimal trajectory. Now, we establish some important properties of (7).

Theorem 1. *Given $\alpha > 0$, the SOE in (7) complies with the following:*

- **(i) (Smoothness)** *The information vector and matrix $\hat{\mathbf{y}}(t), \hat{\mathbf{Q}}(t)$ are m-times differentiable $\forall t \geq 0$. As a consequence, $\hat{\mathbf{x}}(t)$ is m-times differentiable $\forall t \geq 0$.*
- **(ii) (Unbiasedness)** $\mathbb{E}\{\mathbf{x}(t) - \hat{\mathbf{x}}(t)\} = 0, \forall t \geq 0$.
- **(iii) (Tight consistency)** $\hat{\mathbf{P}}^*(t) \preceq cov\{\mathbf{x}(t) - \hat{\mathbf{x}}(t)\} \preceq \hat{\mathbf{P}}(t), \forall t \geq 0$ *with equality for all $t \notin \boldsymbol{\tau}$ as $\alpha \to 0$.*

Proof. The proof can be found in Appendix D.1.

Remark 3. Different from a discrete-time Kalman filter or the DOE, the SOE is ensured to have a sufficiently smooth output for any $t \geq 0$, even under large perception latency. There are two basic smooth-output alternatives to the SOE proposed in (7): interpolation techniques and low-pass filtering. In the case of interpolation, an mth order spline may be used in a moving horizon fashion to fill in the spaces between samples $\{\hat{\mathbf{y}}^*[k], \hat{\mathbf{Q}}^*[k]\}_{k=0}^{\infty}$ for all $t \notin \boldsymbol{\tau}$. On the other hand, an mth order low-pass filter can be used on top of (6) to obtain smooth versions of $\hat{\mathbf{y}}^*(t), \hat{\mathbf{Q}}^*(t)$ as well. Therefore, an analogous to property **(i)** of Theorem 1 is obtained for both of these alternatives. However, neither unbiasedness nor consistency can be ensured for such types of estimations. In addition, contrary to the alternatives, the tight near-optimal quality of the estimation of our proposal can be adjusted appropriately by modifying the parameter $\alpha > 0$ as shown in item **(iii)** of Theorem 1.

Remark 4. In practice, VTDs such as the ones in [5,7,8] are prone to data association problems when false positives or miss-detections occur, and under false negatives and occlusions where the procedure cannot detect the target. However, a VTD in conjunction with target tracking using state-based predictions provide a more robust alternative since predicted targets can be matched with detections. For instance, [30] provides some examples in which state-based predictions are used to improve the accuracy of the target detector. In this context, the SOE can be used as a state-based prediction by evaluating (7) at any time until a new sample is available, which can then be used for data association.

4. Estimation fusion

Using the SOE in (7), each robot i is able to compute a near-optimal local estimation $\hat{\mathbf{x}}_i(t)$ and a consistent covariance matrix $\hat{\mathbf{P}}_i(t)$ with their corresponding smooth information vector and matrix $\hat{\mathbf{y}}_i(t), \hat{\mathbf{Q}}_i(t)$ as a result of Theorem 1. Hence, we turn our attention to the task of fusing this information across all robots in the communication network. Note that even when processed measurements are uncorrelated, the estimations $\hat{\mathbf{x}}_i(t)$ are correlated since both depend on the same noise process $\mathbf{u}(t)$ driving system (2). However, keeping track of the cross-correlations between estimates at different robots is not scalable with respect to the network size. Therefore, we ignore cross-correlations and fuse estimation information according to:

$$\begin{aligned} \bar{\mathbf{Q}}_{\mathsf{G}}(t) &= \frac{1}{\mathsf{N}}\sum_{i=1}^{\mathsf{N}} \hat{\mathbf{Q}}_i(t), \qquad & \bar{\mathbf{y}}_{\mathsf{G}}(t) &= \frac{1}{\mathsf{N}}\sum_{i=1}^{\mathsf{N}} \hat{\mathbf{y}}_i(t) \\ \bar{\mathbf{x}}_{\mathsf{G}}(t) &= \bar{\mathbf{Q}}_{\mathsf{G}}(t)^{-1}\bar{\mathbf{y}}_{\mathsf{G}}(t), \qquad & \bar{\mathbf{p}}_{\mathsf{G}}(t) &= \mathbf{C}\bar{\mathbf{x}}_{\mathsf{G}}(t) \end{aligned} \tag{8}$$

Note that the matrix $\bar{\mathbf{P}}_{\mathsf{G}}(t) = \bar{\mathbf{Q}}_{\mathsf{G}}(t)^{-1}$ is consistent in the sense that $\mathrm{cov}\{\mathbf{x}(t) - \bar{\mathbf{x}}_{\mathsf{G}}(t)\} \preceq \bar{\mathbf{P}}_{\mathsf{G}}(t)$ since $\bar{\mathbf{Q}}_{\mathsf{G}}(t)$ is a convex combination of the information matrices $\{\hat{\mathbf{Q}}_i(t)\}_{i=1}^{\mathsf{N}}$ and by the covariance-intersection principle from [31, Section 2.1]. The same idea of computing a consistent inverse-covariance fusion estimate as in (8), ignoring cross-correlations, has shown to be very successfully in different state estimation contexts such as [32].

If every robot had access to the global quantity $\bar{\mathbf{p}}_{\mathsf{G}}(t)$ from (8), hence a local trajectory $\mathbf{p}_i^{\mathsf{r}}(t) = \bar{\mathbf{p}}_{\mathsf{G}}(t) + \mathbf{d}_i$ can be constructed for each agent to follow. However, in order to construct a controller $\mathbf{u}_i(t)$ to achieve trajectory tracking of (1) towards $\mathbf{p}_i^{\mathsf{r}}(t)$, knowledge of the derivatives $(\mathbf{p}_i^{\mathsf{r}})^{(\mu)}(t) = \bar{\mathbf{p}}_{\mathsf{G}}^{(\mu)}(t), \mu \in \{1, \dots, m\}$ is required as we discuss in Section 5. While the expressions in (8) constitute a transformation between information and state representations, the following result provides an equivalent transformation between the derivatives of $\bar{\mathbf{p}}_{\mathsf{G}}(t)$ and the ones of $\bar{\mathbf{Q}}_{\mathsf{G}}(t), \bar{\mathbf{y}}_{\mathsf{G}}(t)$.

Lemma 1. *Let $\bar{\mathbf{p}}_G(t).\bar{\mathbf{Q}}_G(t), \bar{\mathbf{y}}_G(t)$ defined in (8) and $\bar{\mathbf{P}}_G(t) = \bar{\mathbf{Q}}_G(t)^{-1}$. Then, for any $\mu \in \{1, \dots, m\}$:*

(i) $\bar{\mathbf{P}}_G^{(\mu)}(t) = -\bar{\mathbf{P}}_G(t)\sum_{\nu=0}^{\mu-1}\binom{\mu}{\nu}\bar{\mathbf{Q}}_G^{(\mu-\nu)}(t)\bar{\mathbf{P}}_G^{(\nu)}(t)$.

(ii) $\bar{\mathbf{p}}_G^{(\mu)}(t) = \mathbf{C}\sum_{\nu=0}^{\mu}\binom{\mu}{\nu}\bar{\mathbf{P}}_G^{(\nu)}(t)\bar{\mathbf{y}}_G^{(\mu-\nu)}(t)$.

Proof. The proof can be found in Appendix D.2.

Computing (8) as well as the derivatives in Lemma 1-**(ii)** in a decentralized fashion under time-varying $\hat{\mathbf{y}}_i(t), \hat{\mathbf{Q}}_i(t)$ is not trivial. In the following we provide an algorithm to compute such quantities asymptotically without steady state error using exact dynamic consensus tools, even under persistently varying $\hat{\mathbf{y}}_i(t), \hat{\mathbf{Q}}_i(t)$. The fusion algorithm is composed by two sequential stages. First, we use a *consensus stage* where information fusion is performed to compute $\bar{\mathbf{y}}_{\mathsf{G}}(t), \bar{\mathbf{Q}}_{\mathsf{G}}(t)$ and its first m derivatives using local information $\{\hat{\mathbf{y}}_i(t), \hat{\mathbf{Q}}_i(t)\}_{i=1}^{\mathsf{N}}$ in a decentralized fashion. Second, we use an *output stage* where the outputs of the consensus stage are organized in order to compute $\bar{\mathbf{p}}_{\mathsf{G}}(t)$ and its first m derivatives based on the transformations given in (8) and Lemma 1.

Based on the REDCHO algorithm [26] given in Appendix C with parameters $\{k_\mu, \gamma_\mu\}_{\mu=0}^m$ and θ, the consensus stage is composed by consensus protocols for the information vector

$$\begin{aligned} \mathbf{y}_{i,\mu}(t) &= \hat{\mathbf{y}}_i^{(\mu)}(t) - \textstyle\sum_{\nu=0}^m G_{\mu+1,\nu+1}\mathbf{v}_{i,\nu}(t) \\ \dot{\mathbf{v}}_{i,\mu}(t) &= k_\mu\theta^{\mu+1}\textstyle\sum_{j=1}^{\mathsf{N}} a_{ij}\left\lceil \mathbf{y}_{i,0}(t) - \mathbf{y}_{j,0}(t)\right\rfloor^{\frac{m-\mu}{m+1}} + \mathbf{v}_{i,\mu+1}(t) - \gamma_\mu\mathbf{v}_{i,\mu}(t), \\ &\quad \text{for } 0 \le \mu < m \\ \dot{\mathbf{v}}_{i,m}(t) &= k_m\theta^{m+1}\textstyle\sum_{j=1}^{\mathsf{N}} a_{ij}\left\lceil \mathbf{y}_{i,0}(t) - \mathbf{y}_{j,0}(t)\right\rfloor^{0} - \gamma_m\mathbf{v}_{i,m}(t) \end{aligned} \tag{9}$$

and for the information matrix

$$\begin{aligned} \mathbf{Q}_{i,\mu}(t) &= \hat{\mathbf{Q}}_i^{(\mu)}(t) - \textstyle\sum_{\nu=0}^m G_{\mu+1,\nu+1}\mathbf{V}_{i,\nu}(t) \\ \dot{\mathbf{V}}_{i,\mu}(t) &= k_\mu\theta^{\mu+1}\textstyle\sum_{j=1}^{\mathsf{N}} a_{ij}\left\lceil \mathbf{Q}_{i,0}(t) - \mathbf{Q}_{j,0}(t)\right\rfloor^{\frac{m-\mu}{m+1}} + \mathbf{V}_{i,\mu+1}(t) - \gamma_\mu\mathbf{V}_{i,\mu}(t), \\ &\quad \text{for } 0 \le \mu < m \\ \dot{\mathbf{V}}_{i,m}(t) &= k_m\theta^{m+1}\textstyle\sum_{j=1}^{\mathsf{N}} a_{ij}\left\lceil \mathbf{Q}_{i,0}(t) - \mathbf{Q}_{j,0}(t)\right\rfloor^{0} - \gamma_m\mathbf{V}_{i,m}(t) \end{aligned} \tag{10}$$

where a_{ij} are the components of the adjacency matrix of $\mathcal{G}$, $G_{\mu+1,\nu+1}$ with $\mu, \nu \in \{0, \dots, m\}$ are the components of $\mathbf{G}$ defined in Appendix A.

The protocols in (9) and (10) take as an input the local information $\hat{\mathbf{y}}_i(t), \hat{\mathbf{Q}}_i(t)$, and have outputs $\{\mathbf{y}_{i,\mu}(t), \mathbf{Q}_{i,\mu}(t)\}_{\mu=0}^m$ computed through the internal variables $\{\mathbf{v}_{i,\mu}(t), \mathbf{V}_{i,\mu}(t)\}_{\mu=0}^m$. The purpose of (9) is that $\mathbf{y}_{i,0}(t)$, $\dots, \mathbf{y}_{i,m}(t)$ will converge towards $\bar{\mathbf{y}}_{\mathsf{G}}(t)$ and its first m derivatives for each robot $i \in \{1, \dots, \mathsf{N}\}$. Similarly, $\mathbf{Q}_{i,0}(t), \dots, \mathbf{Q}_{i,m}(t)$ will converge towards $\bar{\mathbf{Q}}_{\mathsf{G}}(t)$ and its first m derivatives. In addition, the structure of the protocol allows each agent to communicate only $\mathbf{y}_{i,0}(t)$ and $\mathbf{Q}_{i,0}(t)$ to its neighbors. It is clear that each robot requires to compute $\{\hat{\mathbf{y}}_i^{(\mu)}(t), \hat{\mathbf{Q}}_i^{(\mu)}(t)\}_{\mu=0}^m$ locally as observed in equations for $\mathbf{y}_{i,\mu}(t), \mathbf{Q}_{i,\mu}(t)$ in (9) and (10). These can be computed explicitly from (7) since the expression for the transition function $\eta(\bullet;\alpha)$ and the model based predictions in (6) are known explicitly as well.

Moreover, note that each component of the equations in (9) and (10) is an independent instance of the REDCHO protocol in (C.1) from Appendix C. As a result, (9) and (10) can be alternatively implemented using $\mathsf{nm}+nm(nm+1)/2$ REDCHO instances, one for each non-repeated component of $\mathbf{y}_{i,m}(t) \in \mathbb{R}^{nm}, \mathbf{Q}_{i,m}(t) \in \mathbb{R}^{nm\times nm}$ provided that $\mathbf{Q}_{i,m}(t)$ is symmetric.

In the output stage, $m+1$ outputs are obtained from $\{\mathbf{y}_{i,\mu}(t), \mathbf{Q}_{i,\mu}(t)\}_{\mu=0}^m$ as given in Algorithm 1 which is based on the transformation of Lemma 1. In fact, the structure in Algorithm 1 is equivalent to the one in Lemma 1 assuming that $\{\mathbf{y}_{i,\mu}(t), \mathbf{Q}_{i,\mu}(t)\}_{\mu=0}^m$ have already converged towards $\bar{\mathbf{y}}_{\mathsf{G}}(t), \bar{\mathbf{Q}}_{\mathsf{G}}(t)$ and their first m derivatives. Hence, in order to ensure convergence we adopt the following assumption.

Algorithm 1 Estimation fusion output stage

```
inputs: {y_{i,μ}(t), Q_{i,μ}(t)}_{μ=0}^m  computed from (9) and (10)
P_{i,0}(t) ← Q_{i,0}(t)^{-1}
p_{i,0}(t) ← C P_{i,0}(t) y_{i,0}(t)
for μ ∈ {1, ..., m} do
    P_{i,μ}(t) ← −P_{i,0}(t) Σ_{ν=0}^{μ−1} (μ choose ν) Q_{i,μ−ν}(t) P_{i,ν}(t)
    p_{i,μ}(t) ← C Σ_{ν=0}^{μ} (μ choose ν) P_{i,ν}(t) y_{i,μ−ν}(t)
end for
return {p_{i,μ}(t)}_{μ=0}^m
```

Assumption 1. Let gains $\gamma_0, \dots, \gamma_m > 0$ and $\hat{s}_i(t)$ be an arbitrary component of $\hat{\mathbf{y}}_i(t)$ or $\hat{\mathbf{Q}}_i(t)$. Hence, $\{\hat{s}_i(t)\}_{i=1}^{\mathsf{N}}$ satisfy Assumption 2 in Appendix C for some $L > 0$.

Remark 5. Due to item **(i)** of Theorem 1, Assumption 1 is complied for t on any compact interval and sufficiently big $L > 0$. However, this assumption can be complied for any $t \ge 0$ as well by assuming that the motion of the target has bounded derivatives, which is mild in practical scenarios.

Theorem 2. *Let Assumption 1 hold and $\mathcal{G}$ be a connected network. Moreover, let the initial conditions and gains for each REDCHO instance configured to satisfy the conditions of Proposition 3 in Appendix C. Then, there exists an m-times differentiable consensus signal $\tilde{\mathbf{p}}(t) \in \mathbb{R}^n$ and $T > 0$ such that the outputs of Algorithm 1 satisfy that for all $\mu \in \{0, \dots, m\}$:*

$$\tilde{\mathbf{p}}^{(\mu)}(t) = \mathbf{p}_{1,\mu}(t) = \cdots = \mathbf{p}_{N,\mu}(t), \forall t \ge T$$

In addition, $\lim_{t\to\infty}\left|\tilde{\mathbf{p}}^{(\mu)}(t)-\bar{\mathbf{p}}_G^{(\mu)}(t)\right|=0$

Proof. The proof can be found in Appendix D.3.

Remark 6. The interpretation of the convergence result in Theorem 2 is that the outputs of Algorithm 1 for all robots will converge to some arbitrary consensus signal $\tilde{\mathbf{p}}(t)$ in finite time, which is not necessarily the average $\bar{\mathbf{p}}_{\mathsf{G}}(t)$. Achieving consensus in finite time is interesting from the point of view of formation control since it allows to the observer and the controller to be designed independently without compromising the stability of the overall system as discussed in Section 5. Moreover, all robots will place themselves around the same formation center $\tilde{\mathbf{p}}(t)$ in finite time, which will converge towards $\bar{\mathbf{p}}_{\mathsf{G}}(t)$ according to Theorem 2.

5. Formation control

Equipping all agents with the previously presented tools for distributed estimation we construct a local controller of the form:

$$\mathbf{u}_i(t)=\mathbf{p}_{i,m}(t)-\kappa_0(\mathbf{p}_i(t)-\mathbf{p}_{i,0}(t)-\mathbf{d}_i)-\sum_{\mu=1}^{m-1}\kappa_\mu(\mathbf{p}_i^{(\mu)}(t)-\mathbf{p}_{i,\mu}(t)) \tag{11}$$

where the roots of the polynomial $\lambda^m+\sum_{\mu=0}^{m-1}\kappa_\mu\lambda^\mu$ have negative real part.

Proposition 1. *Let the conditions of Theorem 2 hold. Define,* $\mathbf{p}_i^{\mathsf{r}}(t):=\tilde{\mathbf{p}}_G(t)+\mathbf{d}_i$ *with* $\tilde{\mathbf{p}}_G(t)$ *in (8). Then, the closed loop system (1) under controller (11) satisfies the formation goal in (3) regardless of* $\mathbf{x}_i(0)$.

Proof. First, note that due to Theorem 2, then there exists $T>0$ such that $\mathbf{p}_{i,\mu}(t)\equiv\tilde{\mathbf{p}}^{(\mu)}(t)$. Moreover, the outputs $\mathbf{p}_{i,\mu}(t)$ remain bounded for $t\in[0,T]$. The closed loop system (1) under (11) is input to state stable with respect to $\mathbf{u}_i(t)$. Thus, there are no finite-time escapes of $\mathbf{x}_i(t)$ for any $t\in[0,T]$. For $t\geq T$, (11) takes the form

$$\mathbf{u}_i(t)=\tilde{\mathbf{p}}^{(m)}(t)-\kappa_0(\mathbf{p}_i(t)-\tilde{\mathbf{p}}(t)-\mathbf{d}_i)-\sum_{\mu=1}^{m-1}\kappa_\mu(\mathbf{p}_i^{(\mu)}(t)-\tilde{\mathbf{p}}^{(\mu)}(t)) \tag{12}$$

Now, define $\mathbf{e}(t)=\mathbf{p}_i(t)-\tilde{\mathbf{p}}(t)-\mathbf{d}_i$ to obtain $\mathbf{e}^{(m)}(t)=-\sum_{\mu=0}^{m-1}\kappa_\mu\mathbf{e}^{(\mu)}(t)$ which is asymptotically stable towards the origin. Hence, $\mathbf{p}_i(t)$ converge asymptotically towards $\tilde{\mathbf{p}}(t)+\mathbf{d}_i$ which converge towards $\bar{\mathbf{p}}_{\mathsf{G}}(t)+\mathbf{d}_i$ due to Theorem 2, implying (3).

Remark 7. Note that a linear controller was proposed in (11), whose design is decoupled from the sampling instants $\boldsymbol{\tau}_i$ arising from the perception mechanism. As evident from the proof of Proposition 1 and since the outputs of the estimation fusion technique from Algorithm 1 converge all $m+1$ derivatives of $\bar{\mathbf{p}}_{\mathsf{G}}(t)$, then these ideas can be extended directly to other trajectory tracking controllers, without requiring a co-design between the perception mechanism and the controller in order to ensure stability.

6. Simulation examples

For simplicity in the presentation, we assume that each of the n coordinates is uncorrelated both in the detection and in the target model (2). Hence, it suffices to analyze each coordinate by separate. Equivalently, we consider $n=1$. Now, let $m=2$ for (1) and (2) to represent second-order integrators. All REDCHO instances are configured with $m=2$ and gains $k_0=6,k_1=11,k_2=6,\gamma_0=\gamma_1=\gamma_2=1,\theta=40$ obtained using the tuning rules from [22,26]. Similarly, all robots use the controller (11) with $\kappa_0=1,\kappa_1=2$.

As characterized by [11], it is expected that a standard target detector based on, e.g., convolutional neural networks, improve its performance when more computing power is employed, effectively increasing its perception latency. Hence, to illustrate the performance of our proposal under this perception latency and detection quality trade-off, the perception mechanism is configured with two possible perception methods with latencies $\Delta^1=1,\Delta^2=0.5$. These correspond to a computation-intensive detection method and a lighter one, respectively. As a result, for illustration purposes, we choose covariance matrices $\mathbf{R}^1=0.01,\mathbf{R}^2=0.1$ respectively to simulate that the first method performs better than the second one at the expense of large computing time.

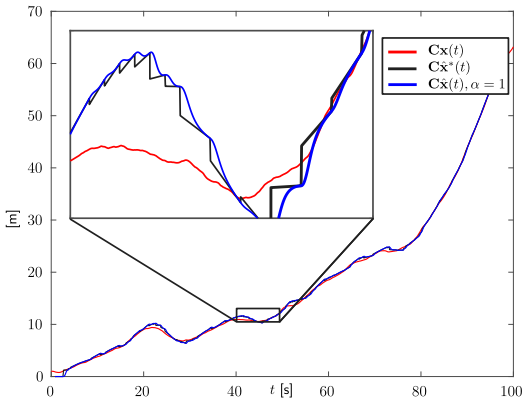

Fig. 2. A realization of the target position $\mathbf{Cx}(t)$ as well as its corresponding estimations $\mathbf{C}\hat{\mathbf{x}}(t)$ and $\mathbf{C}\hat{\mathbf{x}}^*(t)$ for the SOE and the DOE respectively. Note that the SOE is always a smooth estimation whereas the DOE is discontinuous at the sampling instants.

Furthermore, let $\mathbf{W}=1$ for the noise process in (2). The stochastic system (2) was simulated using Euler–Maruyama with time step $\Delta t=10^{-6}$ whereas the REDCHO instances in (C.1) from Appendix C were simulated using explicit Euler method with the same time step.

6.1. Single robot

In this section, we consider $\mathsf{N}=1$ in order to evaluate the performance of the SOE from Section 3. Consider a randomly generated sequence of perception latencies $\{\Delta_{1,k}\}_{k=0}^{\infty}$ with $\Delta_{1,k}\in\{1,0.5\}$ leading to a sequence of sampling instants $\boldsymbol{\tau}_1=\{\tau_{1,k}\}_{k=1}^{\infty}$. Fig. 2 shows a realization $\mathbf{x}(t)$ of (2), and the SOE $\hat{\mathbf{x}}_1(t)$ from (7) for $\alpha=1$. Similarly, we show the output $\hat{\mathbf{x}}_1^*(t)$ for the DOE in (6) for comparison which coincides with the Kalman filter at $t\in\boldsymbol{\tau}_1$. We construct a control input analogous to (11) as:

$$\mathbf{u}_i(t)=\ddot{\mathbf{p}}_1^{\mathsf{r}}(t)-\kappa_0(\mathbf{p}_1(t)-\mathbf{p}_1^{\mathsf{r}}(t))-\kappa_1(\dot{\mathbf{p}}_1(t)-\dot{\mathbf{p}}_1^{\mathsf{r}}(t)) \tag{13}$$

for all $t\notin\boldsymbol{\tau}_1$, where $\mathbf{p}_1^{\mathsf{r}}(t)=\mathbf{C}\hat{\mathbf{x}}_1(t)+\mathbf{d}_1$ for the SOE and $\mathbf{p}_1^{\mathsf{r}}(t)=\mathbf{C}\hat{\mathbf{x}}_1^*(t)+\mathbf{d}_1$ for the DOE. The expressions for the derivatives $\ddot{\mathbf{p}}_1^{\mathsf{r}}(t),\dot{\mathbf{p}}_1^{\mathsf{r}}(t)$ can be obtained explicitly from (6) and (7) and are omitted here for brevity. Moreover, note that $\ddot{\mathbf{p}}_1^{\mathsf{r}}(t),\dot{\mathbf{p}}_1^{\mathsf{r}}(t),\forall t\in\boldsymbol{\tau}_1$ does not exist when using the DOE. Fig. 3 shows the trajectory tracking performance of the robot (1) using the controller (13) for each case. It can be observed that the tracking error converges to zero when using the SOE. On the other hand, persistent transients are observed when using the DOE due to the discontinuities in the reference at $t\in\boldsymbol{\tau}_1$.

6.2. Multi-robot

Now, consider a communication network of $\mathsf{N}=10$ robots connected in a ring topology. The estimation fusion protocol was implemented by separate for the X and Y components of a two dimensional target trajectory $\mathbf{p}(t)$. Fig. 4 shows the convergence of the first components of $\mathbf{y}_{i,0}(t),\mathbf{y}_{i,1}(t),\mathbf{y}_{i,2}(t)$ for the estimation of the X coordinate, where it can be observed that all agents converge to a common signal in finite time, and then converge asymptotically to the first component of the centralized signal $\bar{\mathbf{y}}_{\mathsf{G}}(t)$ and its derivatives. The convergence

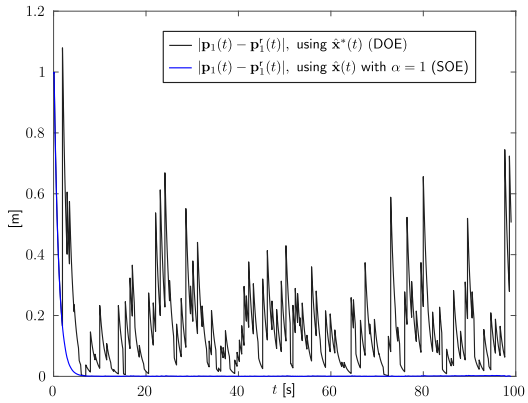

Fig. 3. Trajectory tracking performance of a single robot (1) under the control input (13). Note that the reference is always smooth when using the SOE, resulting in asymptotic convergence for the tracking error. However, the reference is discontinuous when using the DOE, which leads to persistent transients in the robot performance, preventing asymptotic convergence of the tracking error.

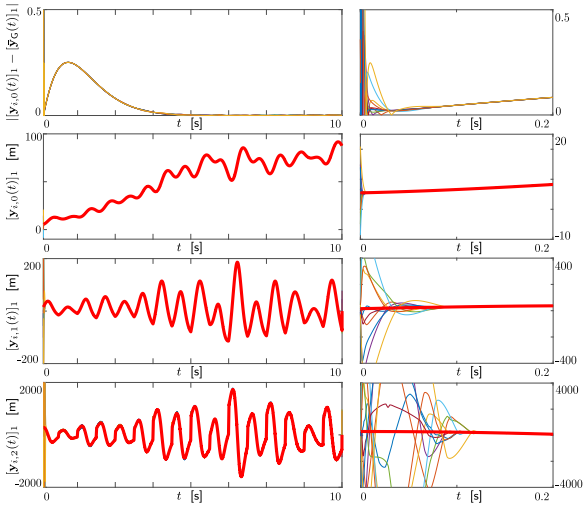

Fig. 4. Trajectories for the first components of $\mathbf{y}_{i,0}(t), \mathbf{y}_{i,1}(t), \mathbf{y}_{i,2}(t)$ denoted as $[\mathbf{y}_{i,0}(t)]_1, [\mathbf{y}_{i,1}(t)]_1, [\mathbf{y}_{i,2}(t)]_1$, shown to converge to $[\bar{\mathbf{y}}_{\mathsf{G}}(t)]_1, [\dot{\bar{\mathbf{y}}}_{\mathsf{G}}(t)]_1, [\ddot{\bar{\mathbf{y}}}_{\mathsf{G}}(t)]_1$ which appear in solid red color. Figures on the left show trajectories in the interval $t \in [0, 10]$ to depict the asymptotic convergence behavior of the algorithm towards the global signals. On the other hand, figures on the right show convergence towards consensus which occurs in the interval $t \in [0, 0.2]$.

of the rest of the components of $\mathbf{y}_{i,0}(t), \mathbf{y}_{i,1}(t), \mathbf{y}_{i,2}(t)$ as well as for $\mathbf{Q}_{i,0}(t), \mathbf{Q}_{i,1}(t), \mathbf{Q}_{i,2}(t)$ is similar and is omitted here for the sake of brevity. In addition, we use the Root Mean Squared (RMS) error performance index in this experiment to measure the impact of using the fusion block. For a single experiment of duration T, the RMS value for an arbitrary scalar signal $x(t)$ is computed as

$$\mathsf{RMS}\{x(t)\} := \sqrt{\frac{1}{T}\int_0^T x(t)^2 \mathrm{d}t}$$

Furthermore, we compute the average RMS value for arbitrary scalar signals $\{x_i(t)\}_{i=1}^{\mathsf{N}}$ as $\mathsf{RMS}_{\mathsf{avg}}\{x_i(t)\}_{i=0}^{\mathsf{N}} := \frac{1}{\mathsf{N}}\sum_{i=1}^{\mathsf{N}} \mathsf{RMS}\{x_i(t)\}$. The estimation error after the output stage of Algorithm 1 is shown in Fig. 5 where it can be observed that the overall global estimate reduces the RMS error in a factor of 3.5 with respect to the average RMS of the individual local estimations. In addition, Fig. 6 shows the actual formation behavior of the robots on the plane. Here, the robots start at random positions and converge to a circular formation around the target estimation $\bar{\mathbf{p}}_{\mathsf{G}}(t)$.

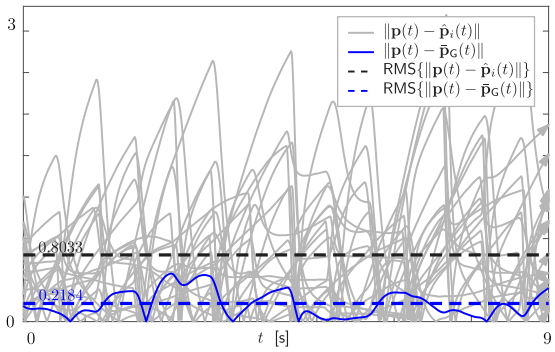

Fig. 5. Error comparison between the actual realization of the target position $\mathbf{p}(t)$, local SOE estimates $\hat{\mathbf{p}}_i(t)$ (grey) and the collaborative estimation $\bar{\mathbf{p}}_{\mathsf{G}}(t)$ (blue). Moreover, average Root Mean Squared (RMS) values are shown in each case, where an improvement of a factor of 3.5 is observed when comparing the collaborative estimate with respect to the local ones.

6.3. Ablation and parameter analysis

In this section, we study the influence of the parameter α in the SOE as well as the fusion block. For this purpose, we perform $\mathsf{N}_{\mathsf{MC}} = 100$ Monte-Carlo runs, for each of the following configurations. We use a similar setting as in Section 6.2 with $\mathsf{N} = 10$ robots connected in a ring topology. We test the SOE with different values of the parameter $\alpha \in \{0.1, 1, 10\}$ as well as the DOE. In addition, for each value of α, we test the performance of the system when using the fusion block or not for computation of the reference signal $\bar{\mathbf{p}}_{\mathsf{G}}(t)$. In the case of the DOE, we do not include the fusion block, since the output of such estimator does not satisfy Assumption 1 due to the discontinuities in the output, regardless of the motion of the target. When the fusion block is not used, we compute $\hat{\mathbf{p}}_i(t) = \mathbf{C}\hat{\mathbf{x}}_i(t)$ from the SOE (7) or the DOE (6) depending on the configuration. Otherwise, we compute $\hat{\mathbf{p}}_i(t) = \hat{\mathbf{p}}_{i,0}(t)$ as the output of the estimation fusion output stage from Algorithm 1.

For each experiment, a different trajectory of the target (2) was generated in the interval $t \in [0, T]$ with $T = 100$, as well aas different initial conditions for the robots in (1). As performance indicators, use the value of the estimation error $\mathsf{RMS}_{\mathsf{avg}}\{\|\hat{\mathbf{p}}_i(t) - \mathbf{p}(t)\|\}_{i=1}^{\mathsf{N}}$ and the tracking error $\mathsf{RMS}_{\mathsf{avg}}\{\|\bar{\mathbf{p}}_{\mathsf{G}}(t) + \mathbf{d}_i - \mathbf{p}_i(t)\|\}_{i=1}^{\mathsf{N}}$ where recall that $\mathbf{p}_i(t)$ is the position of the ith robot. In addition, we measure the control effort by means of $\mathsf{RMS}_{\mathsf{avg}}\{\|\mathbf{u}_i(t)\|\}_{i=1}^{\mathsf{N}}$ and

$$\mathsf{PEAK}\{\mathbf{u}_i(t)\}_{i=1}^{\mathsf{N}} := \max_{i\in\{1,\ldots,\mathsf{N}\}} \sup_{t\in[0,T]} \|\mathbf{u}_i(t)\|$$

The results of these simulations are summarized in Table 1. The performance indicators in the rows of Table 1 were averaged over all 100 experiments for each column of the table. It can be observed that the DOE performs the best in terms of the estimation error among the configurations which do not use fusion. However, the SOE in conjunction with the fusion method is able to outperform the DOE in all cases, being the best configuration with $\alpha = 0.1$ due to the tight consistency of the estimations ensured by Theorem 1. The disadvantages of the DOE are more evident when looking at the tracking error, where it performs the worst up to 1 order of magnitude. The reason is that the discontinuities in the DOE cause the persistent transients illustrated in Fig. 3 whereas the SOE does not have this problem in any configuration.

On the other hand, the bad tracking performance of the DOE is balanced by the resulting small control effort both in RMS and peak values, outperforming the SOE in all cases. The reason is that the DOE ignores the values of the derivatives of the estimation at the discontinuities since they are undefined for those instants. However, the SOE is able to compute these derivatives for all time. Due to the fact that as $\alpha \to 0$ one recovers a discontinuous behavior, the derivatives can

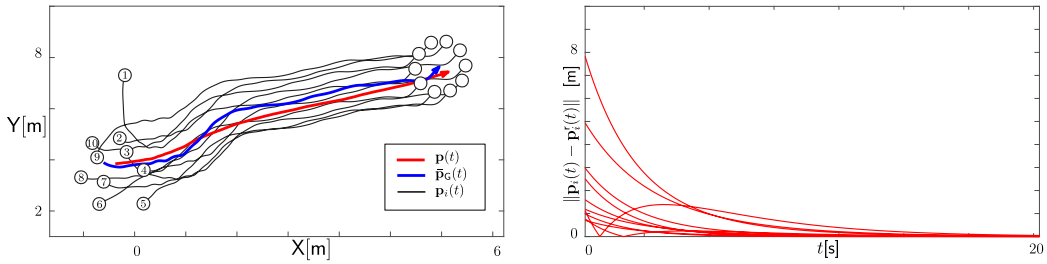

Fig. 6. (Left) Actual target trajectory $\mathbf{p}(t)$ on the plane as well as the global estimation $\bar{\mathbf{p}}_G(t)$ and individual robot trajectories $\mathbf{p}_i(t)$ which converge to the circular formation around $\bar{\mathbf{p}}_G(t)$. (Right) Formation error between the robot position $\mathbf{p}_i(t)$ and the reference $\mathbf{p}_i^r(t) = \bar{\mathbf{p}}_G(t) + \mathbf{d}_i$.

Table 1
Ablation and parameter analysis for the proposal. The DOE is compared with our proposal using different configurations. Moreover, different performance indicators are depicted, where the best value for each row is marked in bold font. In particular the first two rows represent the estimation and tracking RMS errors measured in [m]. The last two rows depict the control effort in terms of RMS and peak values, both measured in [m/s^2].

Estimator	DOE	SOE $\alpha = 0.1$		SOE $\alpha = 1$		SOE $\alpha = 10$	
Fusion	✗	✗	✓	✗	✓	✗	✓
$\text{RMS}_{\text{avg}}\{\|\hat{\mathbf{p}}_i(t) - \mathbf{p}(t)\|\}_{i=1}^{\text{N}}$	0.74	0.79	**0.19**	0.85	0.23	0.92	0.34
$\text{RMS}_{\text{avg}}\{\|\bar{\mathbf{p}}_G(t) + \mathbf{d}_i - \mathbf{p}_i(t)\|\}_{i=1}^{\text{N}}$	56.4	2.84	**2.81**	2.82	2.84	2.83	2.84
$\text{RMS}_{\text{avg}}\{\|\mathbf{u}_i(t)\|\}_{i=1}^{\text{N}}$	**6.23**	18.43	22.74	10.75	9.24	15.22	16.81
$\text{PEAK}\{\mathbf{u}_i(t)\}_{i=1}^{\text{N}}$	**8.65**	70.83	75.98	30.21	34.45	65.01	63.78

grow unbounded at the sampling instants as α is decreased. Since these derivatives are used explicitly in the control law (11), hence the control effort is impacted by the value of α in a similar fashion. Moreover, an equivalent behavior happens when α increases, requiring a compromise for this parameter in terms of control effort. For example, the value of $\alpha = 1$ obtains good results for the control effort when compared to $\alpha = 0.1, 10$.

7. Conclusion

A combination of a smooth-output estimator and an estimation fusion stage was proposed for distributed target estimation. It was shown that, in contrast to the non-smooth optimal alternative, the formation control is able to achieve asymptotic convergence to the formation goal. This allows the control design to be decoupled from the perception-latency decisions. The advantages of the proposal where discussed through simulation examples, when compared to a non-smooth optimal alternative. The proposal is yet to be validated in real-world platforms, which imposes an interesting but highly non-trivial challenge to be explored in a future work.

CRediT authorship contribution statement

Rodrigo Aldana-López: Conceptualization, Software, Validation, Formal analysis, Investigation, Data curation, Writing – original draft. **Rosario Aragüés:** Formal analysis, Resources, Investigation, Funding acquisition, Writing – review & editing. **Carlos Sagüés:** Formal analysis, Resources, Supervision, Project administration, Funding acquisition, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix A. Auxiliary matrices

Given $m, n \in \mathbb{N}$, let

$$\mathbf{A}_0 = \begin{bmatrix} \mathbf{0}_m & \mathbf{I}_m \\ 0 & \mathbf{0}_m^\top \end{bmatrix}$$

$\mathbf{B}_0 = [\mathbf{0}_m, 1]^\top$, $\mathbf{C}_0 = [1, \mathbf{0}_m^\top]$, $\mathbf{I}_m$ is the identity matrix of $\mathbb{R}^{m\times m}$ and $\mathbf{0}_m$ is the zero vector in $\mathbb{R}^m$. Furthermore, $\mathbf{A} = \mathbf{A}_0 \otimes \mathbf{I}_n, \mathbf{B} = \mathbf{B}_0 \otimes \mathbf{I}_n, \mathbf{C} = \mathbf{C}_0 \otimes \mathbf{I}_n$ where $\otimes$ is the Kronecker product. Moreover, given $\gamma_0, \dots, \gamma_m \in \mathbb{R}$, let $\mathbf{\Gamma} = \mathbf{A}_0 - \text{diag}(\gamma_0, \dots, \gamma_m)$, $\mathbf{G} = [(\mathbf{C}_0)^\top, (\mathbf{C}_0\mathbf{\Gamma})^\top, \cdots, (\mathbf{C}_0\mathbf{\Gamma}^m)^\top]^\top$.

Appendix B. Example of a transition function

Proposition 2. *Let $\alpha > 0$ and $\eta(\tau;\alpha) = \dfrac{\Phi(\tau)}{\Phi(\tau) + \Phi(\alpha(1-\tau))}$ with $\Phi(\tau) = \tau^{m+1}$. Then, $\eta(\tau;\alpha)$ satisfy Definition 1.*

Proof. Item (i) follows from smoothness of $\Phi(\tau)$ leading to smoothness of $\eta(\tau,\alpha)$ for any $\tau \in [0,1]$. Note $\Phi^{(\mu)}(0) = 0, \forall \mu \in \{1, \dots, m\}$ and hence $\eta^{(\mu)}(\tau,\alpha) = 0$ as well from the product rule. Moreover, $\eta(\tau;\alpha) = 1 - \eta(\alpha(1-\tau);\alpha^{-1})$ so that $\eta^{(\mu)}(1;\alpha) = -\eta^{(\mu)}(0;\alpha) = 0, \forall \mu \in \{1, \dots, m\}$ showing item (ii). Item (iii) of Definition 1 is complied by evaluating $\eta(\tau;\alpha) = 0, \eta(\tau;\alpha) = 1$. Finally, note that for fixed $\tau \in (0,1)$, it follows that $\lim_{\alpha\to 0} \Phi(\alpha(1-\tau)) = 0$ point-wise, implying item (iv).

Appendix C. The REDCHO protocol

Consider N agents connected in a communication network modeled by a graph $\mathcal{G}$. Each agent i has access to a local signal $\hat{s}_i(t)$ and has $m+1$ internal scalar variables $v_{i,0}(t), \dots, v_{i,m}(t)$ and outputs $s_{i,0}(t), \dots, s_{i,m}(t)$ for which $s_{i,0}(t)$ is communicated to its neighbors. The REDCHO protocol [26] of mth order achieves robust Exact Dynamic Consensus (EDC), i.e. the outputs $s_{i,\mu}(t)$ converge to the signal $\bar{s}^{(\mu)}(t) = (1/\text{N}) \sum_{i=1}^{\text{N}} \hat{s}_i^{(\mu)}(t), \quad \mu \in \{0, \dots, m\}$, regardless of spontaneous connection or disconnection of agents as long as the network topology remains connected. Given positive parameters $\{k_\mu, \gamma_\mu\}_{\mu=0}^m, \theta$, the structure of

REDCHO is:

$$\begin{aligned} s_{i,\mu}(t) &= \hat{s}_i^{(\mu)}(t) - \textstyle\sum_{\nu=0}^{m} G_{\mu+1,\nu+1} v_{i,\nu}(t) \\ \dot{v}_{i,\mu}(t) &= k_\mu \theta^{\mu+1} \textstyle\sum_{j=1}^{\mathsf{N}} a_{ij} \left\lceil s_{i,0}(t) - s_{j,0}(t) \right\rfloor^{\frac{m-\mu}{m+1}} + v_{i,\mu+1}(t) - \gamma_\mu v_{i,\mu}(t), \\ & \text{for } 0 \le \mu < m \\ \dot{v}_{i,m}(t) &= k_m \theta^{m+1} \textstyle\sum_{j=1}^{\mathsf{N}} a_{ij} \left\lceil s_{i,0}(t) - s_{j,0}(t) \right\rfloor^{0} - \gamma_m v_{i,m}(t) \end{aligned} \tag{C.1}$$

where a_{ij} are the components of the adjacency matrix of $\mathcal{G}$, $G_{\mu+1,\nu+1}$ with $\mu, \nu \in \{0, \dots, m\}$ are the components of the matrix $\mathbf{G}$ defined in Appendix A. The REDCHO algorithm requires the following assumption.

Assumption 2. Let

$$\tilde{s}_i(t) = \left(\bar{s}^{(m+1)}(t) - \hat{s}_i^{(m+1)}(t)\right) + \sum_{\mu=0}^{m} l_\mu \left(\bar{s}^{(\mu)}(t) - \hat{s}_i^{(\mu)}(t)\right)$$

where $l_0, \dots, l_m$ are the coefficients of the polynomial $(\lambda+\gamma_0)\cdots(\lambda+\gamma_m) = \lambda^{(m+1)} + \sum_{\mu=0}^{m} l_\mu \lambda^\mu$. Thus, $|\tilde{s}_i(t)| \le L, \forall t \ge 0$ for fixed $\gamma_0, \dots, \gamma_m$ and known $L > 0$.

Proposition 3 (*[26, Adapted from Theorem 10]*). *Let $\mathcal{G}$ be a connected graph and let Assumption 2 hold for given L. Then, for fixed $\gamma_0, \dots, \gamma_m > 0$, there exists $k_0, \dots, k_m, \theta$ sufficiently big, $T > 0$, compact sets $\mathcal{R}_0, \dots, \mathcal{R}_m \subset \mathbb{R}^N$ for the initial conditions $\{s_{i,\mu}(0)\}_{\mu=0}^{m}$ to lie and m-times differentiable signal $\bar{s}(t)$ such that (C.1) satisfies for all $\mu \in \{0, \dots, m\}$:*

- **(i)** $\bar{s}^{(\mu)}(t) = s_{1,\mu}(t) = \cdots = s_{N,\mu}(t), \forall t \ge T$.
- **(ii)** $\lim_{t\to\infty} |\bar{s}(t) - \bar{s}(t)| = 0$.
- **(iii)** $\mathcal{R}_\mu$ *can be made arbitrarily large by increasing* θ.

Appendix D. Proofs

D.1. Proof of Theorem 1

For item **(i)**, note that $\hat{\mathbf{y}}(t), \hat{\mathbf{Q}}(t)$ are m-times differentiable for any $t \notin \boldsymbol{\tau}$ from the expressions in (6) and (7) and item **(i)** of Definition 1. For $t \in \boldsymbol{\tau}$, take an arbitrary $\tau_k \in \boldsymbol{\tau}$ and compute the μth derivative of $\hat{\mathbf{y}}(t)$ with $\mu \in \{0, \dots, m\}$ as $t \to \tau_k^+$ as:

$$\begin{aligned} \lim_{t\to\tau_k^+} \hat{\mathbf{y}}^{(\mu)}(t) = \lim_{t\to\tau_k^+} \sum_{\nu=0}^{\mu} \binom{\mu}{\nu} \lambda_1^{(\nu)}(t|k) (\hat{\mathbf{y}}^*)^{(\mu-\nu)}(t|k-1) \\ + \binom{\mu}{\nu} \lambda_2^{(\nu)}(t|k) (\hat{\mathbf{y}}^*)^{(\mu-\nu)}(t|k) = (\hat{\mathbf{y}}^*)^{(\mu)}(\tau_k|k-1) \end{aligned}$$

where the μth derivative product rule was used as well as by noting that $\binom{\mu}{0} = 1$ and

$$\lim_{t\to\tau_k^+} \lambda_1(t|k) = 1 - \eta(0;\alpha) = 1$$
$$\lim_{t\to\tau_k^+} \lambda_2(t|k) = \eta(0;\alpha) = 0$$
$$\lim_{t\to\tau_k^+} \lambda_1^{(\nu)}(t|k) = -\eta^{(\nu)}(0;\alpha) = 0$$
$$\lim_{t\to\tau_k^+} \lambda_2^{(\nu)}(t|k) = \eta^{(\nu)}(0;\alpha) = 0$$

for any $\nu \in \{1, \dots, m\}$ by items **(ii)** and **(iii)** of Definition 1. Now, for $t \to \tau_k^-$ consider $\hat{\mathbf{y}}(t) = \lambda_1(t|k-1)\hat{\mathbf{y}}^*(t|k-2) + \lambda_2(t|k-1)\hat{\mathbf{y}}^*(t|k-1)$ for $t \in [\tau_{k-1}, \tau_k)$ from (7). Similarly as before,

$$\begin{aligned} \lim_{t\to\tau_k^-} \hat{\mathbf{y}}^{(\mu)}(t) = \lim_{t\to\tau_k^-} \sum_{\nu=0}^{\mu} \binom{\mu}{\nu} \lambda_1^{(\nu)}(t|k-1) (\hat{\mathbf{y}}^*)^{(\mu-\nu)}(t|k-2) \\ + \binom{\mu}{\nu} \lambda_2^{(\nu)}(t|k-1) (\hat{\mathbf{y}}^*)^{(\mu-\nu)}(t|k-1) = (\hat{\mathbf{y}}^*)^{(\mu)}(\tau_k|k-1) \end{aligned} \tag{D.1}$$

where the following identities were used:

$$\lim_{t\to\tau_k^-} \lambda_1(t|k-1) = 1 - \eta((\tau_k - \tau_{k-1})/\Delta_{k-1};\alpha) = 0$$
$$\lim_{t\to\tau_k^-} \lambda_2(t|k-1) = \eta(1;\alpha) = 1$$
$$\lim_{t\to\tau_k^-} \lambda_1^{(\nu)}(t|k-1) = -\eta^{(\nu)}(1;\alpha) = 0$$
$$\lim_{t\to\tau_k^-} \lambda_2^{(\nu)}(t|k-1) = \eta^{(\nu)}(1;\alpha) = 0$$

due to $(\tau_k - \tau_{k-1}) = \Delta_{k-1}$ and Definition 1. Hence, the two-sided limit $\lim_{t\to\tau_k} \hat{\mathbf{y}}^{(\mu)}(t) = (\hat{\mathbf{y}}^*)^{(\mu)}(\tau_k|k-1)$ exists for arbitrary $\tau_k \in \boldsymbol{\tau}$. The proof is the same for $\hat{\mathbf{Q}}(t)$. For item **(ii)** note that $\hat{\mathbf{x}}(t|k)$ is unbiased since it comes from the Kalman filter structure of (5) and (6). Hence, $\hat{\mathbf{x}}(t)$ is unbiased by linearity of $\mathbb{E}\{\bullet\}$ and by the definition of $\hat{\mathbf{x}}(t)$ from (7) as a convex combination of unbiased estimates since $\lambda_1(t|k)+\lambda_2(t|k) = 1$. For item **(iii)** note that $\hat{\mathbf{P}}^*(t|k), \hat{\mathbf{P}}^*(t|k-1)$ for $t \in [\tau_k, \tau_{k+1})$ are covariances for $\hat{\mathbf{x}}^*(t|k), \hat{\mathbf{x}}^*(t|k-1)$ but that $\operatorname{cov}\{\mathbf{x}(t) - \hat{\mathbf{x}}^*(t|k), \mathbf{x}(t) - \hat{\mathbf{x}}^*(t|k-1)\} \neq 0$ since both estimations are correlated by their dependence of the noise process $\{\mathbf{u}(t')|t' \le t\}$ in (2). However, note that the expression of $\hat{\mathbf{Q}}(t)$ is a convex combination of $\hat{\mathbf{Q}}^*(t|k), \hat{\mathbf{Q}}^*(t|k-1)$. Hence, the covariance-intersection principle from [31, Section 2.1] ensures consistency of $\hat{\mathbf{P}}(t)$ i.e., $\hat{\mathbf{P}}^*(t) \preceq \hat{\mathbf{P}}(t)$. Finally, note that by letting $\alpha \to 0$ we have $\lambda_1(t|k) = 0, \lambda_1(t|k) = 1$ for all $t \in (\tau_k, \tau_{k+1})$ using item **(iv)** from Definition 1. Hence, $\lim_{\alpha\to 0} \hat{\mathbf{P}}(t) = \hat{\mathbf{P}}^*(t|k) = \hat{\mathbf{P}}^*(t), \forall t \notin \boldsymbol{\tau}$.

D.2. Proof of Lemma 1

For item **(i)**, take the μth derivative of the identity $\mathbf{I}_m = \bar{\mathbf{Q}}_{\mathsf{G}}(t)\bar{\mathbf{P}}_{\mathsf{G}}(t)$ yielding:

$$0 = \sum_{\nu=0}^{\mu} \binom{\mu}{\nu} \bar{\mathbf{Q}}_{\mathsf{G}}^{(\mu-\nu)}(t) \bar{\mathbf{P}}_{\mathsf{G}}^{(\nu)}(t) = \bar{\mathbf{Q}}_{\mathsf{G}}(t) \bar{\mathbf{P}}_{\mathsf{G}}^{(\mu)}(t) + \sum_{\nu=0}^{\mu-1} \binom{\mu}{\nu} \bar{\mathbf{Q}}_{\mathsf{G}}^{(\mu-\nu)}(t) \bar{\mathbf{P}}_{\mathsf{G}}^{(\nu)}(t)$$

where $\binom{\mu}{\mu} = 1$ was used and from which the expression in item **(i)** is obtained by solving for $\bar{\mathbf{P}}_{\mathsf{G}}^{(\mu)}(t)$. Finally, item **(ii)** is obtained by applying the μth derivative to the definition $\bar{\mathbf{p}}_{\mathsf{G}}(t) = \mathbf{C}\bar{\mathbf{P}}_{\mathsf{G}}(t)\bar{\mathbf{y}}_{\mathsf{G}}$, completing the proof.

D.3. Proof of Theorem 2

First, note that each component of the equations in (9) and (10) is a REDCHO block of the form (C.1). Each individual REDCHO block in (9) (resp. (10)) has scalar input $\hat{s}_i(t)$ given by one component of $\hat{\mathbf{y}}_i(t)$ (resp. $\hat{\mathbf{Q}}_i(t)$), internal scalar variables $\{v_{i,\mu}(t)\}_{\mu=0}^{m}$ given by one component of each $\{\mathbf{v}_{i,\mu}(t)\}_{\mu=0}^{m}$ (resp. $\{\mathbf{V}_{i,\mu}(t)\}_{\mu=0}^{m}$) and outputs $\{s_{i,\mu}(t)\}_{\mu=0}^{m}$ given by one component of each $\{\mathbf{y}_{i,\mu}(t)\}_{\mu=0}^{m}$ (resp. $\{\mathbf{Q}_{i,\mu}(t)\}_{\mu=0}^{m}$). Hence Assumption together with item **(i)** of Proposition 3 imply that there exists $T > 0$ such that

$$\tilde{\mathbf{y}}^{(\mu)}(t) = \mathbf{y}_{1,\mu}(t) = \cdots = \mathbf{y}_{\mathsf{N},\mu}(t)$$
$$\tilde{\mathbf{Q}}^{(\mu)}(t) = \mathbf{Q}_{1,\mu}(t) = \cdots = \mathbf{Q}_{\mathsf{N},\mu}(t)$$

for all $\mu \in \{0, \dots, m\}$ for some consensus signals $\tilde{\mathbf{y}}(t) \in \mathbb{R}^{nm}, \tilde{\mathbf{Q}}(t) \in \mathbb{R}^{nm\times nm}$. Moreover, define $\tilde{\mathbf{P}}(t) = \tilde{\mathbf{Q}}(t)^{-1}$ Hence, for all $t \ge T$ and $i \in \{1, \dots, N\}$ the expressions in Algorithm 1 reads:

$$\begin{aligned} \mathbf{p}_{i,0}(t) &\equiv \mathbf{C}\tilde{\mathbf{P}}(t)\tilde{\mathbf{y}}_0(t) \\ \mathbf{P}_{i,\mu}(t) &\equiv -\tilde{\mathbf{P}}(t) \sum_{\nu=0}^{\mu-1} \binom{\mu}{\nu} \tilde{\mathbf{Q}}^{(\mu-\nu)}(t) \mathbf{P}_{i,\nu}(t) \\ \mathbf{p}_{i,\mu}(t) &\equiv \mathbf{C} \sum_{\nu=0}^{\mu} \binom{\mu}{\nu} \mathbf{P}_{i,\nu}(t) \tilde{\mathbf{y}}^{(\mu-\nu)}(t) \end{aligned} \tag{D.2}$$

Now, item **(ii)** of Proposition 3 imply that $\tilde{\mathbf{y}}(t), \tilde{\mathbf{Q}}(t)$ converge to $\bar{\mathbf{y}}_{\mathsf{G}}(t)$, $\bar{\mathbf{Q}}_{\mathsf{G}}(t)$ as $t \to \infty$. Hence, the expressions in (D.2) converge to $\bar{\mathbf{p}}_{\mathsf{G}}(t)$, $\bar{\mathbf{P}}_{\mathsf{G}}^{(\mu)}(t), \bar{\mathbf{p}}_{\mathsf{G}}^{(\mu)}(t)$ respectively due to Lemma 1, completing the proof.

References

[1] Y. Lan, Z. Lin, M. Cao, G. Yan, A distributed reconfigurable control law for escorting and patrolling missions using teams of unicycles, in: 49th IEEE Conference on Decision and Control, CDC, 2010, pp. 5456–5461.

[2] F. Castanedo, J. García, M.A. Patricio, J.M. Molina, Data fusion to improve trajectory tracking in a cooperative surveillance multi-agent architecture, Inf. Fusion 11 (3) (2010) 243–255, agent-Based Information Fusion.

[3] L.-E. Caraballo, A. Montes-Romero, J.-M. Diaz-Banez, J. Capitan, A. Torres-Gonzalez, A. Ollero, Autonomous planning for multiple aerial cinematographers, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2020, pp. 1509–1515.

[4] N. Michael, J. Fink, V. Kumar, Cooperative manipulation and transportation with aerial robots, Auton. Robots 30 (1) (2011) 73–86.

[5] J. Redmon, S.K. Divvala, R.B. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 779–788.

[6] S. Beery, G. Wu, V. Rathod, R. Votel, J. Huang, Context r-cnn: Long term temporal context for per-camera object detection, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2020, pp. 13072–13082.

[7] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (06) (2015).

[8] J. Dai, Y. Li, K. He, J. Sun, R-fcn: Object detection via region-based fully convolutional networks, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, Curran Associates Inc. Red Hook, NY, USA, 2016, pp. 379–387.

[9] L. Wang, L. Zhang, Z. Yi, Trajectory predictor by using recurrent neural networks in visual tracking, IEEE Trans. Cybern. 47 (10) (2017) 3172–3183.

[10] R. Aldana-López, R. Aragüés, C. Sagüés, Attention vs. precision: latency scheduling for uncertainty resilient control systems, in: IEEE Conference on Decision and Control, CDC, 2020, pp. 5697–5702.

[11] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, K. Murphy, Speed/accuracy trade-offs for modern convolutional object detectors, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July (2017) 21-26, 2017, pp. 3296–3297.

[12] H. Luo, W. Xie, X. Wang, W. Zeng, Detect or track: Towards cost-effective video object detection/tracking, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, 2019, pp. 8803–8810.

[13] M. Guan, C. Wen, M. Shan, C.-L. Ng, Y. Zou, Real-time event-triggered object tracking in the presence of model drift and occlusion, IEEE Transactions on Industrial Electronics (2018) 05.

[14] S. Yao, Y. Hao, Y. Zhao, H. Shao, D. Liu, S. Liu, T. Wang, J. Li, T. Abdelzaher, Scheduling real-time deep learning services as imprecise computations, in: IEEE 26th International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA, 2020, pp. 1–10.

[15] H. Hu, D. Dey, M. Hebert, J. Bagnell, Learning anytime predictions in neural networks via adaptive loss balancing, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol.33, 2019, pp. 3812–3821.

[16] H. Jin, S. Sun, Distributed filtering for multi-sensor systems with missing data, Inf. Fusion (2022).

[17] S.-L. Sun, Multi-sensor optimal fusion fixed-interval kalman smoothers, Inf. Fusion 9 (2) (2008) 293–299.

[18] A.T. Kamal, J.A. Farrell, A.K. Roy-Chowdhury, Information weighted consensus filters and their application in distributed camera networks, IEEE Trans. Automat. Control 58 (12) (2013) 3112–3125.

[19] R. Olfati-Saber, Distributed kalman filter with embedded consensus filters, in: IEEE Conference on Decision and Control, 2005, pp. 8179–8184.

[20] R. Olfati-Saber, Distributed kalman filtering for sensor networks, in: IEEE Conference on Decision and Control, 2007, pp. 5492–5498.

[21] S.S. Kia, B. Van Scoy, J. Cortes, R.A. Freeman, K.M. Lynch, S. Martinez, Tutorial on dynamic average consensus: The problem, its applications, and the algorithms, IEEE Control Syst. Mag. 39 (3) (2019) 40–72.

[22] R. Aldana-López, R. Aragüés, C. Sagüés, EDCHO: High order exact dynamic consensus, Automatica 131 (2021) 109750.

[23] S. Wang, W. Ren, On the convergence conditions of distributed dynamic state estimation using sensor networks: A unified framework, IEEE Trans. Control Syst. Technol. 26 (4) (2018) 1300–1316.

[24] X. He, W. Xue, H. Fang, Consistent distributed state estimation with global observability over sensor network, Automatica 92 (2018) 162–172.

[25] E. Sebastian, E. Montijano, C. Sagues, All-in-one: Certifiable optimal distributed kalman filter under unknown correlations, in: IEEE Conference on Decision and Control, CDC, 2021, pp. 6578–6583.

[26] R. Aldana-López, R. Aragüés, C. Sagüés, REDCHO: Robust exact dynamic consensus of high order, Automatica 141 (2022) 110320.

[27] K. Åström, Introduction To Stochastic Control Theory, Mathematics in Science and Engineering, Academic Press, 1970.

[28] Y.V. Pant, H. Abbas, K. Mohta, R.A. Quaye, T.X. Nghiem, J. Devietti, R. Mangharam, Anytime computation and control for autonomous systems, IEEE Trans. Control Syst. Technol. 29 (2) (2021) 768–779.

[29] T. Soderstrom, Discrete-Time Stochastic Systems: Estimation and Control, second ed., Springer-Verlag, Berlin, Heidelberg, 2002.

[30] Y.-M. Song, K. Yoon, Y. Young Chul, K.-C. Yow, M. Jeon, Online multi-object tracking with gmphd filter and occlusion group management, IEEE Access (2019) 1.

[31] W. Niehsen, Information fusion based on fast covariance intersection filtering, in: International Conference on Information Fusion, Vol. 2, 2002, pp. 901–904.

[32] R. Aragues, C. Sagues, Y. Mezouar, Feature-based map merging with dynamic consensus on information increments, in: IEEE International Conference on Robotics and Automation, 2013, pp. 2725–2730.