Design and SITL Performance of an online Distributed Target Estimation for UAV Swarm

Journal of Physics:
Conference Series

PAPER • OPEN ACCESS

Design and SITL Performance of an online Distributed Target Estimation for UAV Swarm

To cite this article: Fausto Francesco Lizzio *et al* 2023 *J. Phys.: Conf. Ser.* **2526** 012086

Design and SITL Performance of an online Distributed Target Estimation for UAV Swarm

Fausto Francesco Lizzio[1], Stefano Primatesta[1], Haoyu Guo[2], Giorgio Guglieri[3]

[1]Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Torino, Italy
[2]School of Automation Science and Electrical Engineering, Beihang University, Beijing, China
[3]Department of Mechanical and Aerospace Engineering, Politecnico di Torino, CNR-IEIIT, Torino, Italy

E-mail: `fausto.lizzio@polito.it`

Abstract. In this paper, an online distributed target estimation for a swarm of Unmanned Aerial Vehicles in formation flight is considered. In particular, a flocking algorithm is employed to handle the formation task, while a Distributed Kalman Filter in the Information form is used to perform the estimate of a mobile target state. Software-In-The-Loop simulations are carried on in the ROS/Gazebo environment to illustrate the coupling between the formation and the estimation tasks. Moreover, the influence of swarm connectivity in the estimation process is analyzed, showing the benefits of information fusion.

1. Introduction

Distributed control of Unmanned Aerial Vehicles (UAVs) has recently received a considerable amount of attention. This is due to the cost-effectiveness of the aerial platforms, the scalability of the control strategies, and the added value provided by multiple agents in terms of redundancy, greater coverage, and time efficiency.

Formation control [1] is one of the first issues to handle when dealing with a UAV swarm, as drones are often asked to attain a desired shape in space during the completion of a task. Distributed formation algorithms usually require the presence of a unique target to be followed by the agents to avoid the fragmentation of the swarm, as discussed in [2].

Most of the literature regarding distributed UAV formation [3, 4, 5] considers a collaborative target that shares its state information with the entire swarm or with a fraction of it. However, this is not always the case in real-world missions, as the precise state of the target may be unknown and must be estimated [6].

Thus, it is clear how formation control and target estimation are strictly related topics [7]. The aim of this work is to define a strategy for an online collaborative target estimation performed by a swarm of UAVs in formation flight.

For the formation task, we start from the flocking consensus algorithm introduced in [2], and, specifically, we consider a tailored version of this method presented by the authors in [8]. With respect to the standard algorithm, the proposed version is able to eliminate the steady-state errors arising in the inter-agent distances and obtain a smoother transient behavior. However, in our previous work, the exact position of the target chased by the UAVs was simply broadcast to all of them, so that the target acted as the leader of the swarm.

In this work, the formation algorithm developed in [8] is applied in cascade with a distributed target estimation one. In particular, a Distributed Kalman Filter in the Information form is employed. This choice is motivated by the ease of the information fusion step in distributed networks provided by this form of the filter, which reduces it to a trivial sum [9].

With this configuration, the swarm does not chase the exact position of the target, but rather the location provided by the collaborative target estimation process. This results in a leaderless scheme that further stresses the capability of the formation algorithm and brings the simulation closer to a real setup. Indeed, each UAV performs a noisy measurement of the target position through a range-bearing sensor. This information is first managed locally by each drone and, then, fused with the information collected by the other agents to obtain a collaborative estimate of the target's state.

In this paper, we employ the ROS/Gazebo environment in the Software-In-The-Loop (SITL) mode provided by the PX4 stack [10]. By doing so, it is possible to interface the autopilot as it would be done in an actual implementation and to use the navigation information that would be available on board on a real aerial platform.

Implementation details are reported to address the feasibility of the method. Several simulation rounds are carried on to investigate the coupling between the formation and the target estimation tasks. Metrics as the estimates' standard deviation will be provided to evaluate the influence of the swarm connectivity on the performance of the algorithm. This will help in addressing the impact that online distributed estimation may have in real flight scenarios such as patrolling, surveillance, and infrastructure inspection.

This work is organized as follows. In Section 2, the flocking task is addressed, and the tailored flocking algorithm previously developed by the authors is summarized. In Section 3, the distributed estimation algorithm is introduced. Section 4 provides the simulation details and illustrates the numerical results. Finally, Section 5 provides some concluding remarks.

2. Tailored Flocking Algorithm

In this section, we address the formation task. Specifically, we start from the well-known flocking algorithm found in [2] which we will refer to as the *standard protocol*. Through this protocol, the agents of a swarm are able to stay close to the centroid of the swarm, to avoid collision with other members and to travel with the same velocity, as firstly explained in [11].

Consider N double-integrator agents

$$\dot{\mathbf{q}}_i = \mathbf{p}_i, \quad \dot{\mathbf{p}}_i = \mathbf{u}_i, \quad \text{for } i = 1, ..., N, \tag{1}$$

with $\mathbf{q}_i = [x_i \; y_i]^T \in \mathbb{R}^2$ and $\mathbf{p}_i = [\dot{x}_i \; \dot{y}_i]^T \in \mathbb{R}^2$ representing the planar position and velocity of agent i. The control input of the *standard protocol*

$$\mathbf{u}_i = \underbrace{\mathbf{u}_{i,d}}_{\textit{distance regulator}} + \underbrace{\mathbf{u}_{i,v}}_{\textit{velocity matching}} + \underbrace{\mathbf{u}_{i,t}}_{\textit{target following}} \tag{2}$$

is given by the sum of three terms. The first one $\mathbf{u}_{i,d}$ is a distance regulator term, which is able to drive the relative distance between agents to a predefined safe value d. It is generated by a pairwise potential field whose minimum is in d, and whose gradient provides

$$\mathbf{u}_{i,d} = -K_d \sum_{j \in \mathcal{N}_i} a_{ij} \cdot (\mathbf{q}_i - \mathbf{q}_j) \cdot \frac{\phi(\|\mathbf{q}_i - \mathbf{q}_j\|_\sigma - \|d\|_\sigma)}{1 + \epsilon \|\mathbf{q}_i - \mathbf{q}_j\|_\sigma}, \tag{3}$$

where K_d is a positive gain, $\mathcal{N}_i$ is the set of neighbors of agent i, function $\phi(z) = \frac{z}{\sqrt{1+z^2}}$, and $\| \cdot \|_\sigma$ indicates a map $\mathbb{R}^m \to \mathbb{R}_0^+$ differentiable everywhere (also in $z = 0$) given by

$\|z\|_\sigma = \frac{1}{\epsilon}[\sqrt{1+\epsilon\|z\|^2} - 1]$, with $\epsilon \in (0,1)$. Finally, denoting the communication range of each agent as r_{comm}, a_{ij} is an interaction parameter depending on the relative distance among agents, so that $a_{ij} = a_{ij}(\mathbf{q}_i, \mathbf{q}_j) = a_{ij}(\frac{\|\mathbf{q}_i - \mathbf{q}_j\|_\sigma}{\|r_{\text{comm}}\|_\sigma})$ and

$$a_{ij}(z) = \begin{cases} 1, & \textit{if } z \in [0,h) \\ \frac{1}{2}[1+\cos(\pi\frac{z-h}{1-h})], & \textit{if } z \in [h,1] \\ 0, & \textit{otherwise} \end{cases} \tag{4}$$

such that the repulsive force is stronger than the attractive one, and contributions from far-away agents are discarded. The second term $\mathbf{u}_{i,v}$ is a consensus regulation

$$\mathbf{u}_{i,v} = -K_v \sum_{j\in\mathcal{N}_i} a_{ij} \cdot (\mathbf{p}_i - \mathbf{p}_j) \tag{5}$$

driving the velocities of the agents to a common value, being K_v a positive gain.

Calling $\mathbf{q}_t = [x_t \; y_t] \in \mathbb{R}^2$ and $\mathbf{p}_t = [\dot{x}_t \; \dot{y}_t] \in \mathbb{R}^2$ the planar position and velocity of a moving target, the last term

$$\mathbf{u}_{i,t} = -K_{d,t} \cdot (\mathbf{q}_i - \mathbf{q}_t) - K_{v,t} \cdot (\mathbf{p}_i - \mathbf{p}_t) \tag{6}$$

is proportional to the position and velocity error of agent i with respect to the target, being $K_{d,t}$ and $K_{v,t}$ positive gains.

However, when applied to agents with nonlinear dynamics, the *standard protocol* yields a steady-state offset on the relative distance between agents, as pointed out in [12]. Thus, the authors introduced in [8] a tailored version of the *standard protocol*. In particular, considering the error in the relative distance between agents

$$\mathbf{e}_{i,int} = \sum_{j\in\mathcal{N}_i} (\mathbf{q}_j - \mathbf{q}_i) \cdot \frac{\phi(\|\mathbf{q}_i - \mathbf{q}_j\|_\sigma - \|d\|_\sigma)}{1+\epsilon\|\mathbf{q}_i - \mathbf{q}_j\|_\sigma} \tag{7}$$

the following integral action

$$\mathbf{u}_{i,int} = K_{int} \int \mathbf{e}_{i,int} dt \tag{8}$$

was added to eliminate the steady state offset, being K_{int} a positive gain. This came with the price of an increased overshoot and considerable oscillations during the transient phase. Hence, the following integral action on the velocity error between the agents and the target

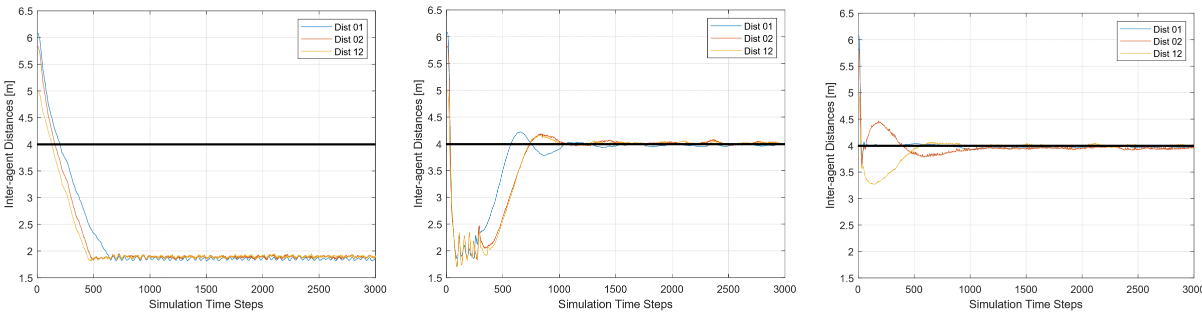
$$\mathbf{u}_{i,v\_int} = -K_{v\_int} \int (\mathbf{p}_i - \mathbf{p}_t) dt \tag{9}$$

was added to dampen the initial transient, being K_{v_int} a positive gain. Finally, a dynamic gain in the position error between the agents and the target

$$K_{d,t}(\mathbf{q}_i, \mathbf{q}_t) = \arctan(\frac{\|\mathbf{q}_i - \mathbf{q}_t\|}{D_{d,t}}) \tag{10}$$

was employed to reduce the attractive force of the target in its proximity, being $D_{d,t}$ a positive parameter modulating this adjustment.

With such modifications, the control input was able to attain zero steady-state offset while yielding a satisfactory transient phase. Figures 1a, 1b and 1c show the inter-agent distances between 3 UAVs under the consensus protocols described above. The drones are asked to reach a $d = 4m$ relative distance starting from random initial positions. Figure 1a clearly displays the steady-state offset in the inter-agent distances produced by the standard protocol. Figure 1b shows how the integral action on the agents' relative distances cancels the offset out, while yielding overshoot and oscillations during the transient phase. Finally, Figure 1c depicts the satisfactory performance provided by the tailored version of the algorithm.



(a) Standard Flocking Protocol. (b) Integral Action on relative distances. (c) Tailored Flocking Algorithm.

Figure 1: Inter-agent distances.

3. Distributed Target Estimation

In this section, we address the estimation task. Specifically, we employ a distributed version of the Extended Kalman Filter found in [9], in which the Information form of the filter is discussed. Instead of the covariance matrix $\mathbf{P}(l|m)$ and state estimate $\boldsymbol{\xi}(l|m)$ of a dynamic process, this filter formulation deals with the information matrix $\mathbf{Y}(l|m)$ and information vector $\mathbf{y}(l|m)$ defined as

$$\mathbf{Y}(l|m) = \mathbf{P}^{-1}(l|m), \quad \mathbf{y}(l|m) = \mathbf{P}^{-1}(l|m) \cdot \boldsymbol{\xi}(l|m), \tag{11}$$

where l and m are two generic time instants, [13].

Although being algebraically equivalent to the classical filter, the Information form has many advantages when it comes to decentralize the estimation process. Indeed, the maximum dimension of a matrix to be inverted is equal to the state estimate dimension, rather than the observation one. In a multi-sensor network, the former is usually much smaller than the latter. Moreover, the information fusion step reduces to a sum. Finally, initialization is trivial because zero information about the process can be assumed at the beginning of the estimation task.

In this work, we focus on a target's planar position and velocity, assuming a linear behavior such as

$$\boldsymbol{\xi}(k+1) = \mathbf{F} \cdot \boldsymbol{\xi}(k) + \mathbf{w}(k), \tag{12}$$

where $\boldsymbol{\xi}(k) = [\hat{x}_t(k)\ \hat{y}_t(k)\ \hat{\dot{x}}_t(k)\ \hat{\dot{y}}_t(k)]^T \in \mathbb{R}^4$ is the state estimate, $\mathbf{w}(k) \in \mathbb{R}^4$ is a zero mean white noise whose covariance matrix is $\mathbf{Q}(k) \in \mathbb{R}^{4\times 4}$, while

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta T & 0 \\ 0 & 1 & 0 & \Delta T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{13}$$

being ΔT the estimation time step.

The observations are performed by $i = 1, \ldots, N$ sensors whose measurement model is

$$\mathbf{z}_i(k) = \mathbf{h}_i(\boldsymbol{\xi}(k)) + \mathbf{v}_i(k), \tag{14}$$

with

$$\mathbf{h}_i(\boldsymbol{\xi}(k)) = \begin{bmatrix} \sqrt{(\hat{x}_t(k) - x_i(k))^2 + (\hat{y}_t(k) - y_i(k))^2} \\ \arctan(\frac{\hat{y}_t(k) - y_i(k)}{\hat{x}_t(k) - x_i(k)}) \end{bmatrix} \tag{15}$$

i.e. a range-bearing sensor affected by an additive zero mean white noise $\mathbf{v}_i(k) \in \mathbb{R}^2$ whose covariance matrix is $\mathbf{R}_i(k) \in \mathbb{R}^{2\times 2}$.

Given the previous considerations, the prediction step of the filter is given by:

$$\begin{cases} \mathbf{Y}_i(k|k-1) = (\mathbf{F} \cdot \mathbf{Y}_i(k-1|k-1) \cdot \mathbf{F}^T + \mathbf{Q}(k))^{-1} \\ \mathbf{y}_i(k|k-1) = \mathbf{Y}_i(k|k-1)(\mathbf{F} \cdot \mathbf{Y}_i(k-1|k-1) \cdot \mathbf{y}_i(k-1|k-1)) \end{cases} \tag{16}$$

where both $\mathbf{Y}_i(k-1|k-1)$ and $\mathbf{y}_i(k-1|k-1)$ can be initialized to zero.

The measurements performed by the N sensors allow to compute

$$\begin{cases} \mathbf{I}_i(k) &= \nabla \mathbf{h}_i^T(\boldsymbol{\xi}(k|k-1)) \cdot \mathbf{R}_i^{-1}(k) \cdot \nabla \mathbf{h}_i(\boldsymbol{\xi}(k|k-1)) \\ \mathbf{i}_i(k) &= \nabla \mathbf{h}_i^T(\boldsymbol{\xi}(k|k-1)) \cdot \mathbf{R}_i^{-1}(k) \cdot (\mathbf{z}_i(k) - \mathbf{h}_i(\boldsymbol{\xi}(k|k-1)) + \nabla \mathbf{h}_i(\boldsymbol{\xi}(k|k-1)) \cdot \boldsymbol{\xi}(k|k-1)) \end{cases} \tag{17}$$

where $\nabla \mathbf{h}_i^T(\boldsymbol{\xi}(k|k-1))$ denotes the gradient operator.

Finally, the update step is given by

$$\begin{cases} \mathbf{Y}_i(k|k) &= \mathbf{Y}_i(k|k-1) + \mathbf{I}_i(k) + \sum_{j \in \mathcal{N}_i} \mathbf{I}_j(k) \\ \mathbf{y}_i(k|k) &= \mathbf{y}_i(k|k-1) + \mathbf{i}_i(k) + \sum_{j \in \mathcal{N}_i} \mathbf{i}_j(k) \end{cases} \tag{18}$$

which can be seen as the sum of the information coming from the prediction and from the measurement performed by agent i and by all of its neighbors.

Through $\mathbf{Y}_i(k|k)$ and $\mathbf{y}_i(k|k)$ and the inverse of (11), it is possible to obtain a local estimate $\boldsymbol{\xi}(k|k)$ of the target state which can be used by the target following term (6) of the flocking algorithm. In this way, each agent tracks the estimate of the position and velocity of the target, rather than its actual state.

4. Numerical Examples

In this section, we provide numerical examples of the described application. We employ the Software-In-The-Loop (SITL) simulation mode provided by the PX4 autopilot in the ROS/Gazebo environment. We simulate a swarm of 3 Iris quadcopters which are asked to generate a flocking behavior with an inter-agent distance $d = 4\,\mathrm{m}$ and a communication range of $r_{\mathrm{comm}} = 4.8\,\mathrm{m}$. Through the Distributed Kalman Filter described in the previous section, the UAVs perform a collaborative estimate of the state of a target quadcopter moving along a sinusoidal path. The estimated position and velocity of the target UAV are employed in the target following part of the flocking algorithm. In order to provide a coherent information fusion, the agents need to share a global reference system.

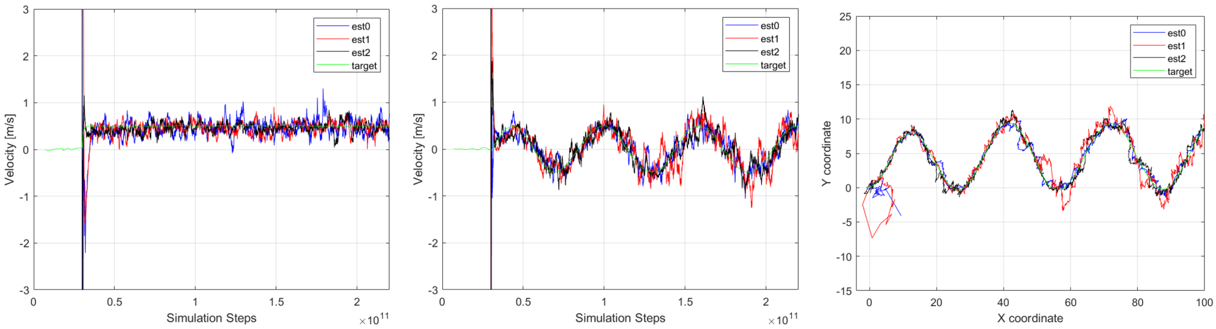
The process noise covariance matrix is assumed constant, and is given by

$$\mathbf{Q} = \begin{bmatrix} 0.03^2 & 0 & 0 & 0 \\ 0 & 0.03^2 & 0 & 0 \\ 0 & 0 & 0.03^2 & 0 \\ 0 & 0 & 0 & 0.03^2 \end{bmatrix} \tag{19}$$

while the measurement noise covariance matrix is given by

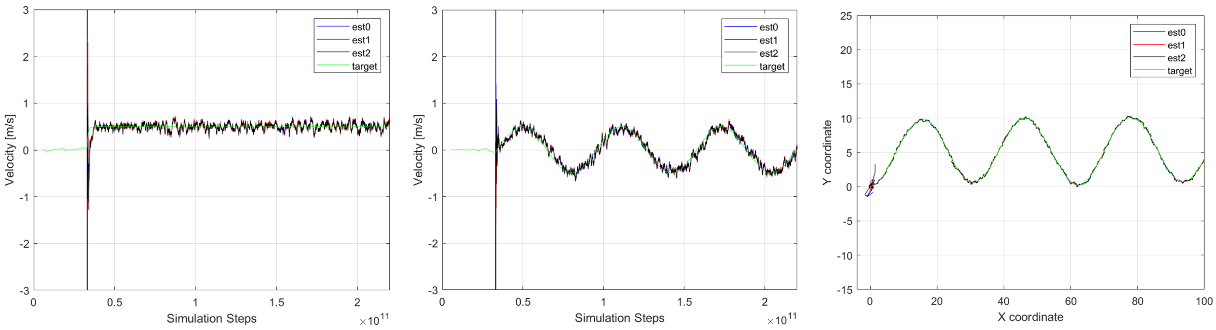
$$\mathbf{R}_i(k) = \Gamma_i^T \begin{bmatrix} 0.4^2 & 0 \\ 0 & \rho_i^2 0.1^2 \end{bmatrix} \Gamma_i \tag{20}$$

where Γ_i is a rotation matrix and $\rho_i = \sqrt{(x_t(k) - x_i(k))^2 + (y_t(k) - y_i(k))^2}$ is the distance between UAV i and the target. Thus, the measurement accuracy increases as the sensor gets closer to the target, as in [7].



(a) X axis velocity. (b) Y axis velocity. (c) Position.

Figure 2: Estimates of Target State with No Information fusion



(a) X axis velocity. (b) Y axis velocity. (c) Position.

Figure 3: Estimates of Target State with Full Information fusion

The estimation process is executed at 20 Hz, and the agents start to follow the target after 20 iterations to avoid being drifted away by the initial estimates.

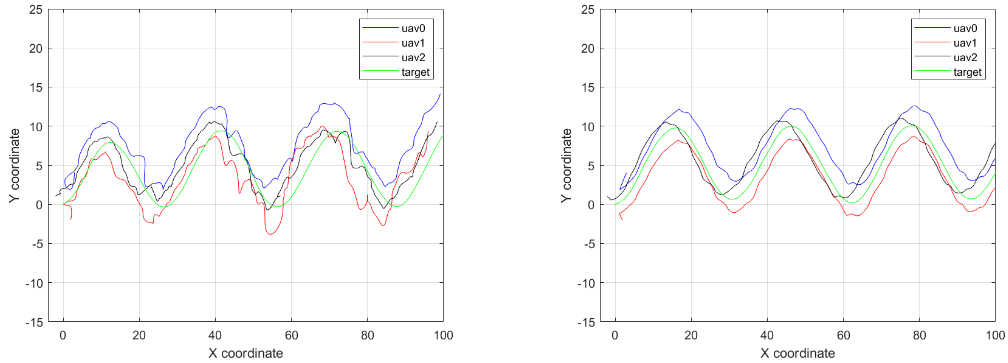
In this work, we conducted two rounds of simulation. In the first one, no information fusion is performed, so each agent carries on the estimation process by itself. In the second case, each drone shares with all of its neighbors the measurement information computed in (17). The reason behind this choice is to evaluate the advantages of full information fusion.

The estimates provided by the 3 UAVs of the target velocities on X and Y axis, and of its trajectory, are shown in Figures 2 and 3. As it is clear, the performance of the estimation filter is considerably better when full information fusion is performed.

Indeed, the standard deviations of the estimates are much higher in the case of no information fusion (IF), resulting in very noisy estimates. This is clear from Table 1, in which the mean over the 3 UAVs of the estimates' standard deviations are reported.

Table 1: Standard Deviations of the Estimates.

	$\sigma_{\hat{x}}$	$\sigma_{\hat{y}}$	$\sigma_{\hat{\dot{x}}}$	$\sigma_{\hat{\dot{y}}}$
No IF	0.2647	0.3903	0.1328	0.1285
Full IF	0.1333	0.1149	0.0762	0.0619



(a) No information fusion. (b) Full information fusion.

Figure 4: Trajectories of the 3 UAVs and the target.

Also, the flocking behavior is heavily affected by the information fusion. Figures 4a and 4b depicts the planar trajectories of the swarm and the target with no information fusion and full information fusion, respectively. This shows how the performance of the estimation and the flocking tasks are heavily coupled.

Notice also that, even though the target model used in the estimation process is linear, the actual path of the target is sinusoidal. However, full information fusion is able to compensate for this fact.

5. Conclusion

In this paper, the coupling between the flocking task and the distributed target estimation for a group of UAVs has been addressed. First, a well-known flocking consensus algorithm was presented, and a tailored version previously developed by the authors was discussed. Then, a Distributed Kalman Filter was analyzed, and the advantages of the Information Form of the filter were explained. The coupled application consisted of a group of 3 quadcopters generating a swarm and chasing a target whose position was collaboratively estimated through the distributed filter. The simulations performed in the ROS/Gazebo environment through the PX4 SITL mode showed the influence of swarm connectivity on the application performances. In particular, much more cohesive estimates and smoother trajectories were observed when full information fusion was employed, with respect to non-collaborative estimation. Future works will focus on nonlinear process models in the filter formulation, as well as on more accurate sensor models in the ROS/Gazebo environment.

References

[1] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.

[2] R. Olfati-Saber. Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Transactions on Automatic Control*, 51(3):401–420, 2006.

[3] W. Ren. Consensus strategies for cooperative control of vehicle formations. *Control Theory & Applications, IET*, 1:505 – 512, 04 2007.

[4] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.

[5] W. Ren and Randal Beard. Distributed consensus in multi-vehicle cooperative control: Theory and applications. *Communications and Control Engineering*, 01 2007.

[6] R. Olfati-Saber. Distributed kalman filter with embedded consensus filters. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 8179–8184, 2005.

[7] R. Olfati-Saber and P. Jalalkamali. Coupled distributed estimation and control for mobile sensor networks. *IEEE Transactions on Automatic Control*, 57(10):2609–2614, 2012.

[8] Fausto Francesco Lizzio, Elisa Capello, and Giorgio Guglieri. Implementation and performance evaluation of a consensus protocol for multi-uav formation with communication delay. In *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2022.

[9] Arthur G. O. Mutambara. *Decentralized Estimation and Control for Multisensor Systems.* CRC Press, Inc., USA, 1st edition, 1998.

[10] Lorenz Meier, Dominik Honegger, and Marc Pollefeys. Px4: A node-based multithreaded open source robotics framework for deeply embedded platforms. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6235–6240, 2015.

[11] Craig W. Reynolds. *Flocks, Herds and Schools: A Distributed Behavioral Model.* Association for Computing Machinery, New York, NY, USA, 1987.

[12] Osamah Saif, Isabelle Fantoni, and Arturo Zavala. Distributed integral control of multiple uavs, precise flocking and navigation. *IET Control Theory I& Applications*, 13, 09 2019.

[13] Giorgio Battistelli, Luigi Chisci, Stefano Morrocchi, and Francesco Papi. An information-theoretic approach to distributed state estimation. *IFAC Proceedings Volumes*, 44(1):12477–12482, 2011. 18th IFAC World Congress.