Journal of Intelligent & Robotic Systems (2021) 101: 65
https://doi.org/10.1007/s10846-021-01342-0

Hybrid Dec-POMDP/PID Guidance System for Formation Flight of Multiple UAVs

Bruno R. O. Floriano[1] · **Geovany A. Borges**[1] · **Henrique C. Ferreira**[1] · **João Y. Ishihara**[1]

Received: 2 October 2019 / Accepted: 8 February 2021 / Published online: 10 March 2021

Abstract

This paper proposes a guidance system with a new hybrid architecture for multiple fixed-wing unmanned aerial vehicles (UAVs) formation flight. The proposed architecture is hybrid in the sense of combining two control policies found in guidance systems: the Decentralized Partially Observable Markov Decision Process (Dec-POMDP) policy and the well-known PID controller. The Dec-POMDP policy part enables every UAV to avoid collision with any other UAV from the formation. The PID part is activated for regions far from the collision risk zone and is implemented in such a way that each UAV is completely controlled by only its own data, independent of its neighbors. The major novelty of the proposed system, compared to other hybrid approaches, is its decentralized nature which favors overcoming the usual problems that come with centralization such as the high dependence on a central decision node, a leader, or a ground station. Inherited from both parts, the overall system performance is robust in noisy and uncertain environments. Simulations were extensively performed to show the effectiveness of the proposed guidance system in distinct scenarios.

Keywords UAV · Aerial vehicles · Dec-POMDP · Hybrid · Guidance system · Formation flight · Cooperative control · Decision making · POMDP · PID

1 Introduction

The use of multiple unmanned aerial vehicles (UAVs) flying in formation has many advantages with respect to a single vehicle, such as the coverage of a larger area, the possibility of task differentiation and redundancy, which can then be applied in missions for visual recognition, mapping, target tracking, rescue missions, video, photography and package delivery [8, 19, 28, 58, 60, 62].

In terms of systems with multiple robot agents, recent works have developed several algorithmic solutions for problems such as guidance and tracking by using Partially Observable Markov Decision Process (POMDP) and its variants (see, e.g., [11, 17, 46, 47, 57]). One reason is that it takes into account noisy observations and uncertain transitions between states [21], being suitable for applications embedded in noisy environments.

For applications with UAV fleets, a POMDP variation called Multiagent POMDP (MPOMDP) has been applied in the works of Miller et al. [34] for target tracking, and of Wu et al. [57] for gathering information about the environment in emergency scenarios. MPOMDP models the system in a single POMDP policy, even though it is composed of multiple vehicles. However, this modeling leads to a dependence on a central planning node which results in high communication requirements and low redundancy capacity [12].

Nevertheless, there is a POMDP variation specifically to address the centralization problem, which was little used for UAV fleets, called Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [6]. Unlike the other POMDP variations, the Dec-POMDP allows for the construction of a decision policy that dictates the individual actions of an agent based on its own observations of the global states. The decentralization removes the necessity of a central unit such as a ground station or a leader agent, easing the communication requirements and the fleet's dynamics [3, 12, 39]. Therefore, due to the planning capability of the POMDP model in uncertain environments and its decentralized extension, the Dec-POMDP model fits well to a realistic cooperative UAV flight modeling.

✉ Bruno R. O. Floriano
brunofloriano3@gmail.com

[1] SG11 AT 23-11, UnB, Brasilia, DF, Brazil

The most noticeable work with Dec-POMDP for UAV fleets was done by Ragi et al. [47]. It showed the effectiveness of the cooperative and decentralized model compared to other simpler approaches in target tracking missions. The problem of its modeling is that a high number of states, actions and observations are necessary, which generates large decision trees (i.e. sets of rules that define actions based on sequences of observations) and therefore might spoil the computational effort.

To mitigate the computational problem, our previous work in [17] proposed to apply the Dec-POMDP policy in the outer loop of the control structure. As shown in Fig. 1, a common architecture for a vehicle inside a complex multi-UAV system involves multiple loops and subsystems [14–17]. In such architecture, the UAV communicates with its neighbors through the communication network, giving information for the guidance system to decide its trajectories and generate navigation commands (such as linear and angular velocities). Those commands are passed to the flight controller which converts them into actuators signals (such as aileron and elevator). Finally, sensors are used to create feedback to the subsystems which will be further detailed in Section 4. By applying the Markov-based model in the guidance system, it was possible to model higher-level states, actions and observations and, consequently, reduce the policy search set and the computational effort.

Moreover, the Dec-POMDP modeling in [17] also allowed for consistent collision avoidance during flight. Indeed, the avoidance achieved is shown to be better than methods such as simpler heuristics and PID controllers, as will be further shown. In terms of collision avoidance in multi-UAVs the literature is vast [24, 25, 29, 33, 36, 49, 52, 53, 56, 59, 60]. In particular for fixed-wing aircraft, a common approach is to use an artificial potential field in a bi-directional network structure [59, 60]. It relies on the use of an artificial repulsion force to drive the vehicles away from each other or from other obstacles. The problem of such a mechanism is the high optimization complexity and the arbitrary configuration for local minimum search [13, 27]. On the other hand, the collision avoidance of POMDP-based systems favors the simplicity in its modeling, as in [5, 10, 55]. Indeed, the model from [17] has such simplicity in collision avoidance while still bringing the decentralized nature of the Dec-POMDP. However, the fleet's time and error performance of Dec-POMDP from [17] was inferior compared to other simpler methods in the literature, as for example, those approaches based on PID controllers [1, 31].

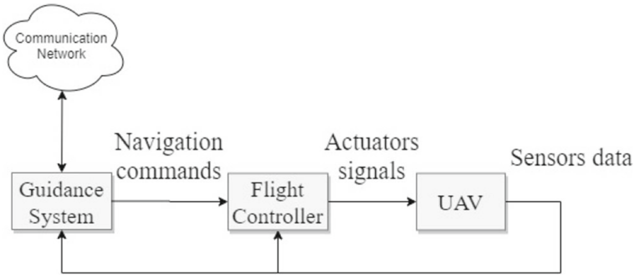


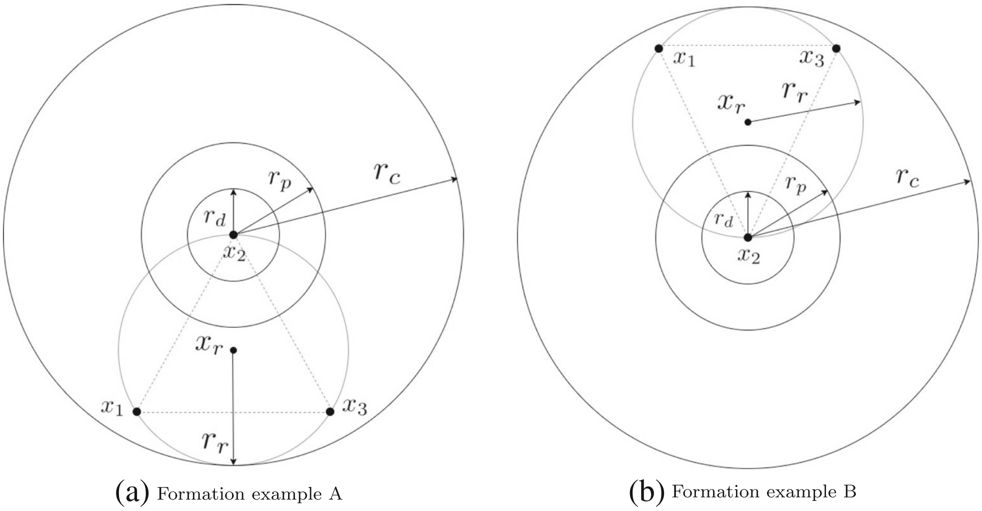
Fig. 1 System diagram of a general i-th UAV

An idea to attach the tracking performance of PID controllers with the low computational effort and collision avoidance of Dec-POMDP is to create a solution with both methods together. Such a combination would result in a hybrid system, i.e. an approach that simultaneously contains both continuous and discrete event-driven models (such as, but not limited to, PIDs and Dec-POMDP, respectively) [63].

In particular, due to its event-driven nature, the use of POMDP and its extensions at hybrid control settings has been developed in the literature (see e.g., [7, 37, 44, 45, 48, 51]). In [51], for example, a POMDP policy switches between distinct controllers of a biped platform, based on the observable terrain. Likewise, the work [44] uses the POMDP policy together with iterated linear quadratic Gaussian control to optimize robot trajectories. Another example is [48], in which the POMDP was combined with belief-desire-intention (BDI), a general framework for planning and execution of distinct tasks with multiple (and with different priority) goals, in order to control an autonomous agent by taking the advantage of each model (dealing with uncertainties of POMDP and goal management of BDI).

Specifically, in the multi-agent scenario, the hybrid POMDP-based approach was most notably used in [37]. In this work, Nair et al. derived the Role-based Multiagent Team Decision Problem (RMTDP), a distributed POMDP variation for role allocation. Nonetheless, at that time there were no effective optimization algorithms for models such as the Dec-POMDP resulting in a centralized approach. Therefore, a contribution of our paper is to develop a decentralized hybrid system based on Dec-POMDP, which was not yet verified in the literature (even outside the aerial vehicle scope).

Considering single agent, hybrid systems have already been applied for aerial vehicles (as seen in [4, 20, 32, 50, 54]). The work of Shen et al. [50], for example, uses a hybrid system combining backstepping with PID to control the position and attitude of an eight-rotor UAV. Also, the work of Jin et al. [20] models a switched system, a particular class of hybrid systems (see [63]), to control a set of wheeled nonholonomic mobile robots teleoperated by a UAV. However, all of those works were performed with a single UAV. Hence another contribution of this paper is to address the formation flight problem (which includes tracking and collision avoidance) by means of using a multi-agent hybrid system.



(a) Formation example A

(b) Formation example B

Fig. 2 Representation of distance parameters for formation flight

The contributions of this paper propose a novel hybrid guidance system to achieve a decentralized formation flight of multiple UAVs. It fills the literature gap that lacked of a decentralized hybrid system based on Dec-POMDP. Particularly for UAV fleets, it also provides novelty in expanding the hybrid application for multiple agents. In addition, by adopting the hybrid approach, the system improves the tracking performance of [17] by adding a controller such as a PID set, while it still has an efficient collision avoidance, provided by the Dec-POMDP policy. The new guidance architecture allows its modeling with fewer states, actions and observations compared to [47] which eases the computational effort. The proposed system's performance is verified through an extensive range of simulations with existing and realistic UAV models.

The paper is organized as follows: Section 2 states the problem that should be solved by the proposed method; Section 3 describes the Dec-POMDP model; Section 4 explains the hybrid guidance system designed based on the Dec-POMDP together with PIDs; Section 5 presents the results achieved and, finally, Section 6 states the final remarks of the work.

2 Problem Statement

It is considered a system of N fixed-wings UAVs flying in a three-dimensional space, with positions measured with respect to a inertial north-east-down (NED) reference frame. Each UAV is identified by an integer index $i \in \mathcal{N}$, in which $\mathcal{N} = \{1, \dots, N\}$ is the set of UAVs. The position of each vehicle, $x_i(t) \in \mathbb{R}^3$ is obtained by its own built-in GPS-based localization system and it is corrupted by a zero mean white Gaussian noise, $n_{x,i}(t)$, with variance σ^2. Therefore, $\hat{x}_i(t) = x_i(t) + n_{x,i}(t)$, in which $\hat{x}_i(t)$ is the position seen from the UAV. It also measures its own heading angle, $\psi_i(t)$, corrupted by the same type of noise, therefore, $\hat{\psi}_i(t) = \psi_i(t) + n_{\psi,i}(t)$, in respect to the North axis in the horizontal plane.

The main fleet reference is given by the virtual leader's position $x_r(t) \in \mathbb{R}^3$ and heading angle (in respect to the North axis) $\psi_r(t)$. Computationally, the main reference signals trajectory $x_r(t)$ and $\psi_r(t)$ are constructed following a model with the same dynamics as the UAVs and has as inputs its linear and angular velocities, $v_r(t)$ and $\omega_r(t)$, respectively. The fleet's desired formation is defined based on the difference between each vehicle's target position, $\bar{x}_i(t)$, and the reference, i.e. $f_i = \bar{x}_i(t) - x_r(t)$. In this work, it will be considered that the magnitude of such difference will be fixed and equal to $||f_i|| = r_r$ and that the target formation is at the same altitude as the reference.

Besides r_r, the constants r_d, r_p and r_c were defined as follows: r_d is the maximum distance between $\bar{x}_i$ and x_i for the formation to be considered as achieved (i.e. if $||\bar{x}_i - x_i|| < r_d$, for all $i \in \mathcal{N}$, then the formation is considered as achieved); r_p is the minimum safe distance between UAVs (i.e. if $||x_i - x_j|| < r_p$, then the i-th and j-th UAVs are considered to be in a collision danger zone); finally r_c is the maximum communication distance between UAVs in which they can communicate with each other (i.e. if $||x_i - x_j|| > r_c$, then the i-th and j-th UAVs can no longer communicate with each other). The neighborhood of the i-th UAV is the set of UAVs inside its communication range, $\mathcal{N}_i \subset \mathcal{N}$. When UAV $j \in \mathcal{N}_i$ then i and j exchange their position data. By default $r_d \leq r_p \leq r_r \leq r_c$.

For illustration, consider a 3-UAV fleet in a triangular shape formation. Figure 2 provides a visual representation of the distance parameters, depicting two examples of triangular formation within the circle of radius r_r and reference at its center. For clarity, the set of circles that represents the other constants were showed only for the UAV $i = 2$. The alternative arrangement of Fig. 2b has the

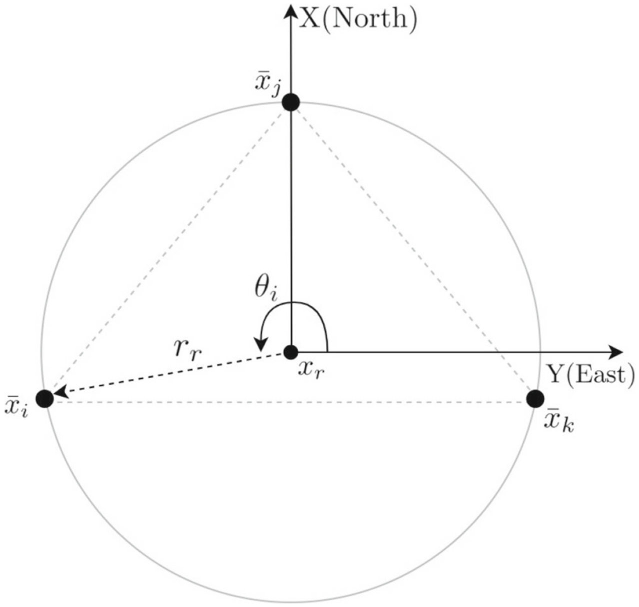


Fig. 3 Representation of angular parameter θ_i

same magnitude and altitude as the formation example of Fig. 2a, differing only by their angular positions, θ_i. As seen in Fig. 3, θ_i is the angular position of the i-th UAV in respect to the East axis at the horizontal plane. It distinguishes each UAV's target position within the perimeter of the circle.

The constant r_s defines the minimum distance between UAVs for which collision still does not happen. Its value is directly related to the UAV's dimensions, as depicted in Fig. 4. Indeed, it has to be the value of the largest length, l, of a UAV.

The problem considered in this paper is to design a guidance system for a formation flight with collision avoidance amongst the UAVs. More specifically, the objective is to design a guidance system with the tracking characteristics of the PID-based solutions without their drawback of not addressing collisions. The proposed solution is intuitively described as to combine, in unique architecture, the "collision avoidance aware" Dec-POMDP policy in emergency situations with the "accurate" PID policy in safe zones. Also, one should note that in order to minimize the collision risk, the control system should prioritize collision avoidance at the expense of reference tracking.

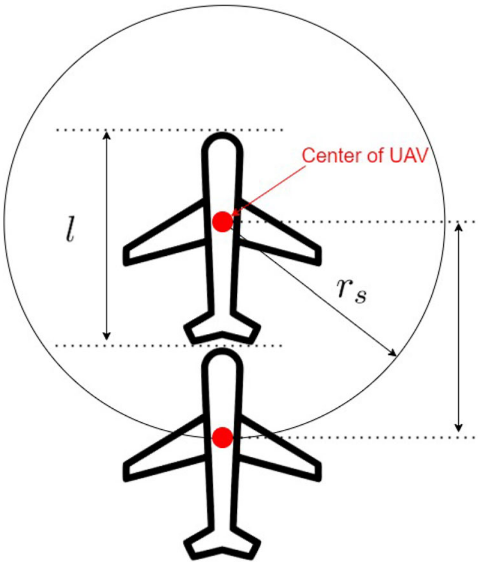


Fig. 4 Representation of collision margin r_s

3 Dec-POMDP Policy

The Dec-POMDP is an model for multi-agent decision-making problems (e.g. formation control and task allocation), designed such that the observation-action policy obtained is decentralized, which avoids the necessity of adding a central processing unit on which the vehicles might depend on. To model a problem in a Dec-POMDP frame, it should be described as global states and rewards, individual actions and observations and stochastic transition and observation functions [3].

It can be formally defined by the tuple $\{\mathcal{N}, S, \{A_i\}, T, R, \{Z_i\}, O, h, \gamma\}$, in which [2, 3]:

- $\mathcal{N}$ is the set of agents, from 1 to N;
- S is the set of global states with initial distribution b_0;
- A_i is the set of individual actions, a_i, for the i-th agent ($i \in \mathcal{N}$). In addition, $A = A_1 \times \cdots \times A_N$ is the joint action of the group, in which × is the Cartesian product operator;
- T is a function that furnishes the probability of transition from state $s \in S$ to state $s' \in S$ by taking the action $a \in A$, i.e. $T(s, a, s') = P(s'|a, s)$;
- R is the global reward function of being in state $s \in S$ and taking the joint action $a \in A$, i.e. $R : S \times A \to \mathbb{R}$;
- Z_i is the set of individual observations, z_i, of the i-th agent ($i \in \mathcal{N}$). In addition, $Z = Z_1 \times \cdots \times Z_N$ is the joint observation of the group;
- O is a function which defines the probability of observing $z \in Z$ after taking the action $a \in A$ that results in state $s' \in S$, i.e. $O(z, a, s') = P(z|a, s')$;
- h is the horizon of the problem, i.e. the number of time steps until termination.
- γ is the discount factor that weights the relevance of future rewards in comparison to the present.

Let the individual policy, π_i be a function that maps the previous and current observations made by the i-th agent to a given action, a_i. Then, the global policy of the group as a whole can be defined as $\pi = \pi_1 \times \cdots \times \pi_N$. The main objective of the Dec-POMDP model is to find the optimized global policy, π^*, i.e. the function that results in the maximum expected cumulative reward. If t is the discrete time of the group execution, then mathematically, the optimization problem is

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{h-1} \gamma^t R(s_t, a_t) | s, \pi\right]. \tag{1}$$

This optimization problem is formulated to be resolved in an offline planning phase by a search algorithm [2]. After the optimized policy is achieved and, consequently, the individual policies are found, then they can be added to the real system to be executed in an online phase. Although the optimization is done by considering a global reward, the decentralization is still attained once each agent receives its own policy, thus it should act based only on its own observation. In this work, the Dec-POMDP model is applied for UAV fleets, therefore, the terminology "agent" (consistent in the literature of multi-robot POMDP variations) stands for the same as "UAV".

4 Hybrid Dec-POMDP/PID Guidance System

As described in Section 1 and shown in Fig. 1, a UAV control architecture is composed by subsystems such as the guidance system and the flight controller, as well as a connection to the communication network. The approach used in this work is focused on the guidance system which will be detailed in this section. It is built with a hybrid architecture in which the Dec-POMDP policy, optimized in the planning phase, works in parallel with the continuous PID algorithm. A navigation system was built in order to appropriately switch between them at appropriate times. The proposed guidance system can be visualized as a block diagram in Fig. 5.

The navigation system takes the reference and the neighbors' data (including its own) and sends the error information for the PID algorithm, the individual observation, z_i for the Dec-POMDP policy and a selecting signal, s_s, for a multiplexer.

The flight control of the aircraft takes the velocities (v and ω) information as references and sends the corresponding responses for the ailerons, throttle and elevator commands (δ_a, δ_t and δ_e, respectively) which will then change the plane's position and heading angle to be measured by the sensors.

4.1 Dec-POMDP Modeling

4.1.1 States, Actions and Observations

During the flight, the fleet shape (described in terms of the UAV's positions with respect to the distance circles) can reach several configurations while the target formation is not achieved. Due to the discrete nature of the Dec-POMDP approach, such configurations have to be discretized in terms of Dec-POMDP states. As an example, Fig. 2 shows a possible discretization in two pointing directions (North, the desired pointing, and South).

The states s were set accordingly to Table 1 which also gives a brief description of each state with its mathematical condition and its corresponding reward. For all the conditions in the table $i, j = \{1, \ldots, N\}$; $i \neq j$. The conditions for different pointing directions are the same, only replacing $\bar{x}_i$ with $\bar{x}_i'$ which is the i-th UAV's position if the desired formation was pointing South. The reward function to be used in Eq. 1, $R(s, a)$, was built considering only the states, i.e. $R(s, a) = R(s)$ once the flying objectives rely only on their definition.

The joint action $a = \begin{bmatrix} a_1 \ldots a_N \end{bmatrix}^T$ combines the contribution of all individual actions a_i. Each individual action was designed as a vector $a_i = \begin{bmatrix} \alpha_v^i \ \alpha_\omega^i \end{bmatrix}^T$, in which

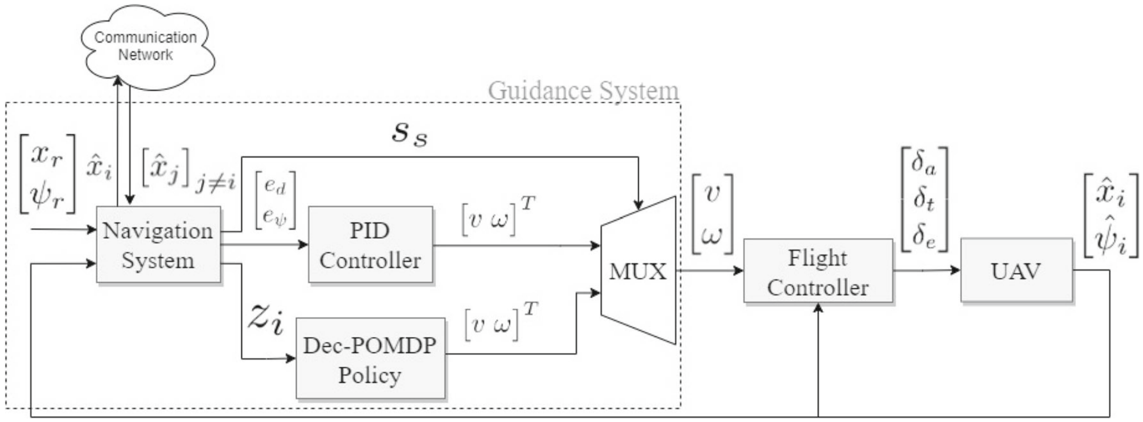

Communication Network
Guidance System
s_s
$\begin{bmatrix} x_r \\ \psi_r \end{bmatrix}$
$\hat{x}_i$
$[\hat{x}_j]_{j\neq i}$
$\begin{bmatrix} e_d \\ e_\psi \end{bmatrix}$
Navigation System
PID Controller
$[v \ \omega]^T$
z_i
Dec-POMDP Policy
$[v \ \omega]^T$
MUX
$\begin{bmatrix} v \\ \omega \end{bmatrix}$
Flight Controller
$\begin{bmatrix} \delta_a \\ \delta_t \\ \delta_e \end{bmatrix}$
UAV
$\begin{bmatrix} \hat{x}_i \\ \hat{\psi}_i \end{bmatrix}$

Fig. 5 Block diagram of the i-th UAV hybrid system

Table 1 States description

State s	Description	Pointing	Conditions	Reward $R(s)$
0	Collision danger	Any	$\exists\ j : \Vert x_i - x_j \Vert \leq r_p$	–100
1	Fleet inside reference circle	North	$\Vert x_i - x_r \Vert < r_r$ & $\Vert x_i - \bar{x}_i \Vert < \Vert x_i - \bar{x}'_i \Vert$	50
2	Fleet inside reference circle	South	$\Vert x_i - x_r \Vert < r_r$ & $\Vert x_i - \bar{x}_i \Vert > \Vert x_i - \bar{x}'_i \Vert$	20
3	Fleet at reference circle perimeter (target)	North	$\Vert x_i - \bar{x}_i \Vert \leq r_d$	100
4	Fleet at reference circle perimeter	South	$\Vert x_i - \bar{x}'_i \Vert \leq r_d\ \forall i$	40
5	Fleet outside reference circle	North	$\Vert x_i - x_r \Vert \geq r_r$ & $\Vert x_i - \bar{x}_i \Vert < \Vert x_i - \bar{x}'_i \Vert$	80
6	Fleet outside reference circle	South	$\Vert x_i - x_r \Vert \geq r_r$ & $\Vert x_i - \bar{x}_i \Vert > \Vert x_i - \bar{x}'_i \Vert$	20
7	Communication loss	Any	$\exists\ j : \Vert x_i - x_j \Vert \geq r_c$	–80

α_v^i and α_ω^i are, respectively, the linear forward acceleration of the aircraft and the angular acceleration sampled at the i-th sampling time and kept constant (i.e, $\boldsymbol{a}(t) = a_i = cte$) during each time step i. The vector a_i can be determined by

$$a_i = \left[\beta_v^i \alpha_{vb} \quad \beta_\omega^i \alpha_{\omega b}\right]^T, \tag{2}$$

in which α_{vb} and $\alpha_{\omega b}$ are base values of linear and angular acceleration, respectively. In addition, $\beta_v^i, \beta_\omega^i \in \{-1, 0, 1\}$ are discrete weighting parameters for each acceleration. Considering (2), the policy π^*, described in Section 3 and optimized in an offline phase, defines the values of β_v^i and β_ω^i based on the observed state, z_i at each time step. To match the input of the flight controller, composed by continuous linear $v(t)$ and angular velocity $\omega(t)$, the acceleration $\boldsymbol{a}(t)$ is integrated before being forwarded to the multiplexer.

Each agent makes an individual observation z_i of the state s, i.e. the observation space is equal to the state space. Accordingly, each agent perceives a noisy observation of the state based on its own sensors and the communication received. Consequently, they follow the same numbering and descriptions as seen in Table 1.

4.1.2 Transition Function

To set the transition function $T(s, a, s')$, the probability square matrix T_a represents the possible outcome of the joint action a. Each of its lines represents the current state s, each column, all possible next states s' and its entry $p_a^{s,s'}$, the probability of transition between those states, with such action. However, the effect of a is given by the combination of the individual actions α_v^i and α_ω^i. Therefore the contribution of each one should be determined separately in the first moment. Accordingly, the matrices T_v^i and T_ω^i are analogous to T_a, but to consider the individual effects of the linear and angular movements, respectively, in the state transitions. Their entries are $p_{\alpha v}^{s,s'}$ and $p_{\alpha\omega}^{s,s'}$, respectively.

To illustrate how to properly set each line of those matrices, let us consider the example of 3 UAVs in a triangular formation as depicted in Fig. 6. In this example, the fleet is at state $s = 5$, i.e. the agents are outside the circle of radius r_r and closer to the main formation. In this case, one can analyze what would happen if agent $i = 1$ decides to reduce its linear velocity, i.e. $\alpha_v^1 = -\alpha_{vb}$ (or $\beta_v^1 = -1$), i.e. to distance from the circle. It is clear that, by taking this action, it is unlikely that the system reaches the circle and achieves states $s' \in \{0, 1, 2, 3, 4\}$. It is also unlikely that the agent goes forward of the current position and achieves state $s' = 6$, so they all receive a chance of $p_{\alpha v}^{5,s'} = 0$. So, the only possibilities are that the system remains in the current state, or they leave to the state $s' = 7$ of communication loss (it goes to far from the others). Since the latter is the most likeable it receives the probability of $p_{\alpha v}^{5,7} = 0.75$ while to maintain the current state it is given $p_{\alpha v}^{5,5} = 0.2$. The only exception lies on including $p_{\alpha v}^{5,0} = 0.05$, chance of getting in state $s' = 0$, due to the possible collision with an agent that is also outside the circle (e.g. if agent $i = 3$ gets nearby). This example is the 6-th line of the $T_v^1(-1)$ matrix in the Appendix A.1. All of those values were manually chosen to represent the high or low probability of each situation as

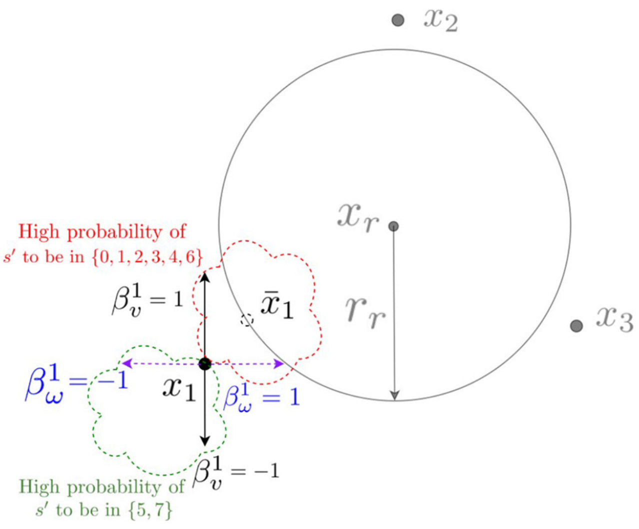


Fig. 6 Example of the actions' effect on the states

exemplified, with the constraint of each line summing up to one.

Hence, this analysis is performed until the contribution of all actions, for all agents, are determined individually. For the numerical example in Section 5, the final matrices T_v^i and T_ω^i obtained following the above procedure can be visualized in the Appendix A.1.

The next step is to attach the contributions of T_v^i and T_ω^i in a single matrix, $T_{a_i}^i$, for the i-th agent individual action. Let $p_{a_i}^{s,s'}$ be its s-s' entry. Then it can be determined by multiplying the corresponding entries of the T_v^i and T_ω^i, due to the combination of both action in the final a_i. Therefore,

$$p_{a_i}^{s,s'} = p_{\alpha v}^{s,s'} p_{\alpha\omega}^{s,s'}. \tag{3}$$

Moreover, each line of $T_{a_i}^i$ obtained in Eq. 3 must be normalized in order that they all sum up to 1 as probabilities.

Finally, in order to get the global transition matrix T_a, considering the actions of all agents together, another product should be computed and normalized analogously to the previous step as

$$p_a^{s,s'} = \prod_{i=1}^{N} p_{a_i}^{s,s'}. \tag{4}$$

4.1.3 Observation Function

Similar to the previous case, the observation function $O(z, a, s')$ was configured as a matrix O_z. Since the action does not affect the observation of the system, the dependence on a was dropped. Let $q_z^{z,s'}$ be the entry of matrix O_z (i.e. the joint observation z corresponds to its rows and state s' to its columns). Let $\zeta(z) = \{z_1, \dots, z_N\}$ be the set of individual observations perceived and $\zeta_i(z) = \{z_i \in \zeta | z_i = s'\}$ the set of observations that are equal to s'. Finally, let $N_{\zeta_i}(z)$ be the number of elements in the set $\zeta_i(z)$ (i.e. the number of agents in which $z_i = s'$).

Larger values of $N_{\zeta_i}(z)$ means that there are more agents in which $z_i = s$. As the agents become far from each other (therefore, as the state gets higher), the communication between vehicles becomes weaker. So it is natural to infer that as communication becomes weaker, it is less likely that the agents will observe the right state. Therefore, if z results in a large $N_{\zeta_i}(z)$, $q_z^{z,s'}$ is inversely proportional to s' and vice-versa. The only exception is for $s' = 7$ (state of communication loss) because instead of receiving a wrong message the agent loses all communication, so it is more likely that it observes $z_i = 7$ (i.e. it does not get any external message).

Knowing this relation between $N_{\zeta_i}(z)$, $q_z^{z,s'}$ and s', Table 2 was built to address the values of $q_z^{z,s'}$ in each situation. Although this paper works with $N = 3$, the extension for different values of N is straightforward, as Table 2 will have N lines ($N_{\zeta_i}(z)$ ranges from 1 to N) and the probability distributions within its columns still follows the pattern described earlier. The variable D_j is the number of combinations of the set $\zeta_i(z)$, in which $N_{\zeta_i}(z) = j$, therefore D_j is given by

Table 2 Observation matrix entry $q_z^{z,s'}$

$N_{\zeta_i}(z)$	$s' = 0$	$1 \leq s' \leq 4$	$5 \leq s' \leq 6$	$s' = 7$
1	$0.01/D_1$	$0.05/D_1$	$0.1/D_1$	$0.05/D_1$
2	$0.09/D_2$	$0.15/D_2$	$0.2/D_2$	$0.1/D_2$
3	0.9	0.8	0.7	0.85

$$D_j = \frac{N!}{j!(N-j)!}(N_{z_i} - 1)^{N-j}, \tag{5}$$

in which, N_{z_i} is the total number of individual observations. The proof of Eq. 5 is in the Appendix A.2. The division for D_j has to be performed because if $N_{\zeta_i}(z) < N$ there is more than one combination, so the probability $q_z^{z,s'}$ must be equally divided among them.

4.2 PID Controller

The PID controller part of the hybrid system shown in Fig. 5 relates the input $u(t) = \begin{bmatrix} e_d & e_\psi \end{bmatrix}^T$ and the output $y(t) = \begin{bmatrix} v & \omega \end{bmatrix}^T$ as follows:

$$y(t) = K_P u(t) + K_I \int_0^t u(\tau)\,\mathrm{d}\tau + K_D \frac{\mathrm{d}}{\mathrm{d}t} u(t), \tag{6}$$

in which $K_P = \operatorname{diag}(K_{Pv}, K_{P\omega})$, $K_I = \operatorname{diag}(K_{Iv}, K_{I\omega})$ and $K_D = \operatorname{diag}(K_{Dv}, K_{D\omega})$ are the proportional, derivative and integral matrices, respectively, and the constants that compose the diagonals in each of these matrices are to control the linear velocity and the angular velocity, respectively.

4.3 Navigation System

As shown in Fig. 5, the proposed navigation system is the block in the controller scheme that processes the internal and external data in order to define the errors and the observations, as well as the responsible for deciding which of the two previous systems should act. Each agent's navigation system has the knowledge of the virtual leader information (x_r, ψ_r) and receives information $\hat{x}_j$ of its neighbor agents $j \in \mathcal{N}_i$. The setting is general and several different particular implementations of the navigation system can be used. In order to illustrate one possibility, consider the navigation system implemented by the pseudo-code of Algorithm 1.

In the input of Algorithm 1, the i-th agent receives the information determined offline from the virtual leader $(x_r$

Algorithm 1 Navigation system pseudo-code.

```
Input: $(x_r, \psi_r, \hat{x}_i, \hat{\psi}_i, \hat{x}_j)$
 1: $f_d = x_r + f_i - \hat{x}_i$
 2: $f'_d = x_r + f'_i - \hat{x}_i$
 3: $d_r = ||\hat{x}_i - x_r||$
 4: if $\mathcal{N}_i = \emptyset$ then
 5:     $z_i = 7$                        ▷ No communication received
 6: else if $\min_{j \in \mathcal{N}_i} \left(||\hat{x}_i - \hat{x}_j||\right) \leq r_p$ then
 7:     $z_i = 0$
 8: else if $||f_d|| \leq r_d$ then
 9:     $z_i = 3$
10: else if $||f'_d|| \leq r_d$ then
11:     $z_i = 4$
12: else if $(d_r > r_r)\&(||f_d|| \leq ||f'_d||)$ then
13:     $z_i = 5$
14: else if $(d_r > r_r)\&(||f_d|| > ||f'_d||)$ then
15:     $z_i = 6$
16: else if $(d_r < r_r)\&(||f_d|| \leq ||f'_d||)$ then
17:     $z_i = 1$
18: else if $(d_r < r_r)\&(||f_d|| > ||f'_d||)$ then
19:     $z_i = 2$
20: else
21:     $z_i = 0$
22: end if
23: $\psi_e = \text{atan2}(f_{dy}, f_{dx})$
24: $e_d = ||f_d|| \cos(\psi_e - \hat{\psi}_i)$
25: if $e_d \leq 0$ then
26:     $e_\psi = (\psi_r - \hat{\psi}_i)$
27: else
28:     $e_\psi = 0.25\psi_e + \psi_r - \hat{\psi}_i$
29: end if
30: if $z_i \leq 2$ then
31:     $s_s = -1$
32: else
33:     $s_s = 1$
34: end if
35: return $s_s, z_i, e_d, e_\psi$
```

and ψ_r) which state the desired trajectory of the flight, from its own internal states ($\hat{x}_i$ and $\hat{\psi}_i$) and from the neighbors through the communication network. Then, the algorithm computes the relative distance to its own position in the formation f_d (cf. Figure 7), the distance to the alternative formation f'_d and the distance to the group reference (virtual leader) d_r. Finally, the output signals s_s, z_i, e_d and e_ψ are calculated.

In Algorithm 1, determining the Dec-POMDP observation, z_i, is a series of conditional statements (lines 4 to 22) accordingly with the definitions presented so far. Preference is given to the states of no communication and collision avoidance, as they represent situations that need fast and prioritized responses. Except for those states, the odd observations represent formations that are nearer to the main desired formation, while the even states are for when the agent gets near the alternative formation, as shown in Fig. 2. For the linear velocity control, the error e_d (computed in line 24) is based on the distance to the desired point, i.e. f_d and the angular component $\cos(\psi_e - \hat{\psi}_i)$ to account for the direction of the forward velocity of the aircraft, in which ψ_e is the angle of the distance vector (in respect to the North axis) as seen in Fig. 7. This is done in order to avoid the agent to accelerate in the wrong direction (the maximum acceleration happens when $\psi_e = \hat{\psi}_i$, i.e. when the agent is heading to the desired point). The angular error, e_ψ (computed in lines 25 to 29), was built such that the agent heading direction ψ_i is able to track both the distance angle ψ_e, in order to point in the right direction, and the reference's heading angle ψ_r, such that the path to follow in maintained. To avoid each UAV to spin when the reference is behind it (when $e_d \leq 0$) an if statement withdraws the term that tracks ψ_e (line 26). Returning to the guidance system, the errors e_d and e_ψ are the inputs for the PID controllers. The observation z_i is the input of the Dec-POMDP policy and the selecting signal s_s is defined as equal to -1 when the Dec-POMDP policy should define the final action and equal to 1 when the PID must take control.

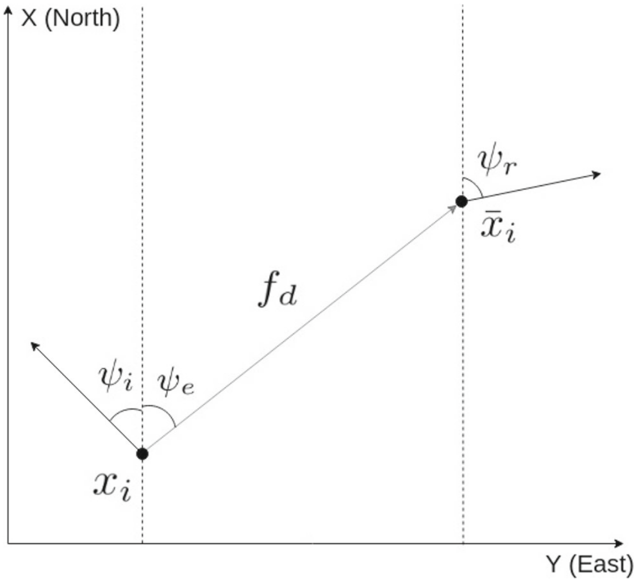


Fig. 7 Top view representation of the vehicle i tracking its reference $\bar{x}_i$

5 Simulation Experiments

The proposed hybrid system was validated in realistic scenarios by performing an extensive range of simulations with different references, formations, noisy measurement conditions, horizons, among other parameters. The tracking effectiveness of the model was tested in simulations with oscillating trajectories in Section 5.2. Performance parameters such as steady-state error, mean squared error and settling time were used to quantify the tracking capability. Likewise, as presented in Section 5.3, collision avoidance was valued in stressed situations in which the vehicle trajectories were configured to collide. The avoidance was broadly analyzed through the minimum distance achieved between UAVs, the time spent within the danger zone and the effective collision occurrence. In Sections 5.3 and 5.4, the proposed method was also compared to other methods in the literature such as formation track with loitering [61], Dec-POMDP only [17], PID only [1, 31] and simple heuristics. Finally, in Section 5.5 the performance of the system with different horizon values are tested. The simulations were performed with a fixed-step solver using the ode4 method (Runge-Kutta) and time step equal to 0.01s. Two distinct UAV models already validated in the literature were used: the WVU YF-22 aircraft developed by Campa et. al. [9] (also used in works such as [14]) and the nonlinear model of Zhang et. al. [61].

The policy was optimized in an Intel Core i5 with 2.3 GHz and 8 GB of RAM, using the MADP Toolbox developed by Oliehoek et al. [42]. This toolbox is a C++ program and library to solve multi-agent problems related to decision-making, such as the Dec-POMDP, with distinct solving methods. The solver chosen was JESP (Joint Equilibrium-based Search for Policies) [18]. Section 5.1 presents the computational analysis of the optimization phase performed offline.

5.1 Computational Analysis

Due to the processing limitations of the computing resources used, a horizon of $h = 3$ was chosen. Note that this choice increases the one used in [17] which was $h = 2$. The matrices used in the observations and transition functions, as well as the states rewards, were in accordance to the description given in Section 4.1 and can be referenced in the Appendix A.1. The initial distribution was set such that the system starts at state $s = 0$, i.e. $b_0(s = 0) = 1$ and zero otherwise. The discount factor, as described in Eq. 1, was set as $\gamma = 0.9$.

The limitation of a maximum horizon of $h = 3$ is due to the problem's complexity. In order to compare it with other methods, the number of different joint policies, N_π, of the model will be used. It is given by (see [40]):

$$N_\pi = O\left(|A_*|^{N_z}\right), \tag{7}$$

in which, $N_z = \frac{N(|Z_*|^h-1)}{|Z_*|-1}$ and $|A_*|$ and $|Z_*|$ are the largest number of individual actions and observations, respectively. This quantity, N_π, is directly related with its computational effort and feasibility.

Table 3 was built by applying (7) in different methods. The number of joint policies of the presented hybrid system was compared with the method from [47] and from the Dec-Tiger [38]. Dec-Tiger is one of the benchmark problems in Dec-POMDP literature and it models a simple scenario with 2 agents searching for the right door between 2 options [41]. As seen in the table, the Dec-Tiger allows for less policies, due to its simplicity ($|A_*| = 3$ and $|Z_*| = 2$). Consequently, it is feasible to solve for horizons up to $h = 5$ with the available computational resources. However, as the system's complexity increases, it becomes computationally harder to solve larger horizons. After all, complex systems (as in multiple UAVs applications) require larger sets to represent the environment with higher fidelity. Indeed the hybrid system method at $h = 3$ has more joint policies than the Dec-Tiger at $h = 5$. Blank spaces in Table 3 represent values larger than $1.8 \cdot 10^{308}$, which is the maximum representation of the IEEE Standard 754 for double-precision floating-point values [22].

Nonetheless, one of the main objectives of this work is to reduce the optimization effort in the context of hybrid Dec-POMDP/PID controllers by decreasing the dimensionality through the remodeling of the formation flight problem. By reducing the number of actions and observations in the proposed method ($|A_*| = 9$ and $|Z_*| = 8$) it was possible to accomplish such reduction, resulting in less joint policies, and consequently less optimization effort when compared to [47].

Solving the problem described resulted in the optimized value of 0 with a total processing time of 0.17 s (for $h = 2$) and 2.04 s (for $h = 3$). The resulting policies have many ramifications (for $h = 2$, visualization of these ramifications is given in Fig. 19) whose number grows exponentially with respect to the horizon. Therefore, the visualization of the policy for $h \geq 3$ is impaired.

For each i-th agent there are 3 time-steps (since $h = 3$). At each time t, the agent takes an action. An action is always followed by an observation that leads to the next time-step

Table 3 Number of different joint policies, N_π, for each Dec-POMDP methods

Horizon	Dec-Tiger	Hybrid system	Method [47]
$h = 1$	9	729	256
$h = 2$	729	$5.8 \cdot 10^{25}$	$8.7 \cdot 10^{40}$
$h = 3$	$4.8 \cdot 10^{6}$	$9.5 \cdot 10^{208}$	–
$h = 4$	$2 \cdot 10^{14}$	–	–
$h = 5$	$3.8 \cdot 10^{29}$	–	–

Table 4 Action equivalences in term of β_v^i and β_ω^i

Action a_i	β_v^i	β_ω^i
0	–1	–1
1	0	–1
2	1	–1
3	–1	0
4	0	0
5	1	0
6	–1	1
7	0	1
8	1	1

until the process is finished. Each action a_i of the policy is a number from 0 to 8 that corresponds to a specific pair of β_v^i and β_ω^i. Such equivalency is described in Table 4.

5.2 Reference Tracking Analysis

This section seeks to analyze the tracking performance of the proposed hybrid system, which means evaluating error and time parameters while tracking a given reference.

To obtain the parameters K_P, K_I and K_D of Eq. 6, an optimization scan was performed, by choosing the set of values that provided the smaller steady-state error (SSE) and mean square error (MSE). For the linear velocity control, the values that better suited the model and the control objectives were $K_{Pv} = 0.03$, $K_{Iv} = 0$ and $K_{Dv} = 1$. As for the angular velocity, the values achieved were $K_{P\omega} = 5$, $K_{I\omega} = 0$ and $K_{D\omega} = 3$. Therefore, a PD control was find to be already suited for such model. The base values were set to $\alpha_v = 1$ m/s^2, $\alpha_\omega = \frac{2\pi}{100}$ rad/s^2.

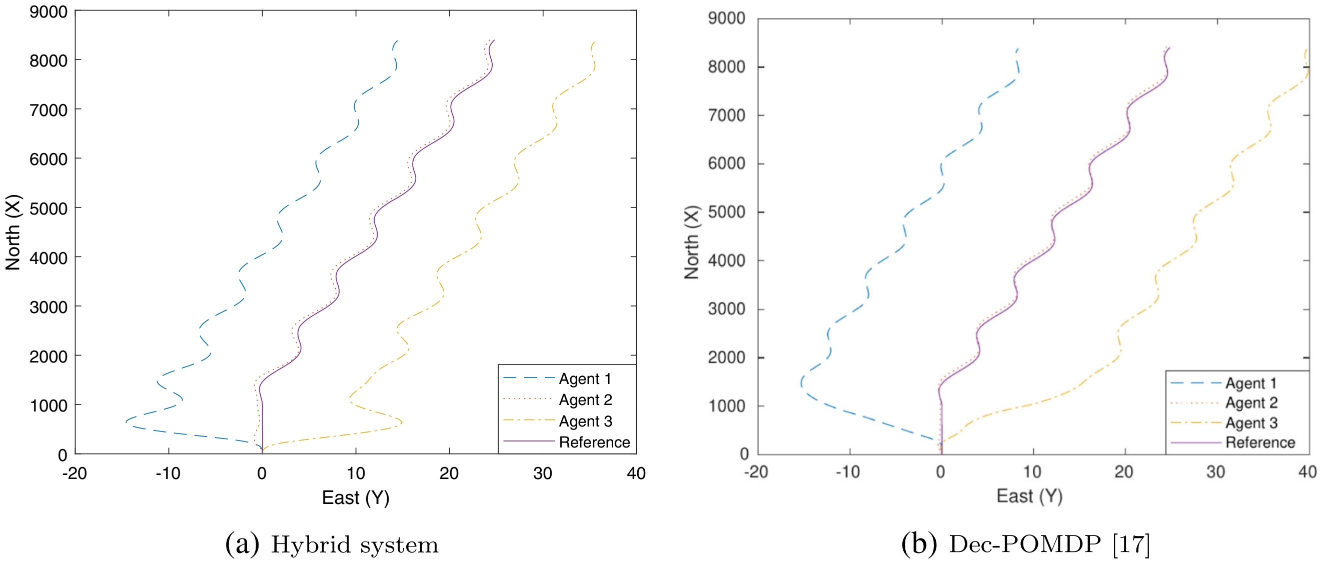
The same reference and formation from [17] were used, i.e. the fleet is guided by a virtual leader with sine wave angular velocity reference of $\omega_r = 0.05\sin(0.23t)$ rad/s if $t > 25$ s and 0 otherwise, and constant linear velocity ($v_r = 42$ m/s). Considering the model of Campa et al. [9], $r_s = 3$ m. The formation was kept constant with $r_r = 15$ m, $r_d = 7$, $r_p = 5$ m, $r_c = 150$ m and $\theta_1 = 5\pi/4$ rad, $\theta_2 = \pi/2$ rad and $\theta_3 = -\pi/4$ rad.

The vehicles' trajectories in the horizontal plane and their errors evolution with time are depicted in Figs. 8a and 9a, respectively. The analogous results achieved in [17] are also depicted for comparison in Figs. 8b and 9b. An initial comparison of the two graphs of Fig. 8 can already perceive that the trajectories are similar yet not the same. However, in the graphs of Fig. 9, in which the largest steady-state error is highlighted, shows that indeed the hybrid system has a lower steady-state error. The counterpoint lies on the time evolution since it appears to be slower in the hybrid system.

Table 5 presents the steady-state error and the settling time of both systems for each agent. Indeed the time parameter of the hybrid system is significantly longer than the Dec-POMDP alone. However, the steady-state error is smaller in the first, for all agents, which shows a compromise between these parameters. In such a trade-off between accuracy and speed, it was preferred that the proposed hybrid system should give preference to being more accurate than the Dec-POMDP alone, instead of being faster.

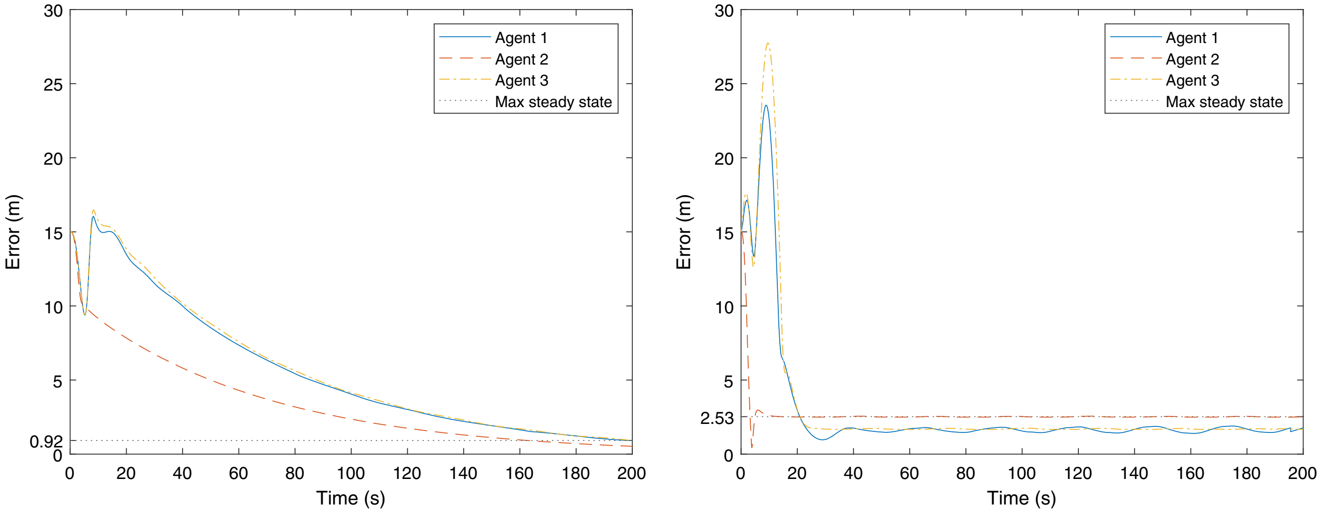
5.2.1 Noise Evaluation in Reference Tracking

To provide a more realistic view of the proposed system, this scenario was also simulated multiple times with the



(a) Hybrid system (b) Dec-POMDP [17]

Fig. 8 UAVs trajectories in North-East (horizontal) plane with oscillating reference



(a) Hybrid system

(b) Dec-POMDP [17]

Fig. 9 UAVs error evolution with oscillating reference. Hybrid system error declines slower but has a smaller steady-state error (SSE)

inclusion of the noise. The standard deviation σ was swept and collected, in each one, the results of the mean-squared error (MSE) and steady-state error (SSE). The results are shown in Fig. 10. It is consistent that the errors increase as the noise raises, but the order of magnitude of such change is not significant as the vehicles remain within the target region of r_d. Therefore the noise increase does not affect significantly the reference tracking performance of the hybrid system.

5.2.2 Evaluation of Different Formations

Another parameter that can be analyzed is the system's performance to different formation parameters. In particular, such performance was tested to the variation of the term r_r, which determines the agents' distances to the reference. The result can be seen in Fig. 11. The results show that the errors increase with the formation radius but still remain inside the allowable range. The steady-state error, for example, is always lower than the distance margin ($r_d = 7$ m) which means the reference tracking was kept with larger formations.

Table 5 Error parameters in reference tracking simulation with different methods

Method	Steady-state	Settling Time (s)
	Error (m)	
	0.92	85.53
Hybrid system	0.53	50.01
	0.91	87.77
	1.77	21.41
Dec-POMDP [17]	2.53	15.01
	1.68	26.46

For each method, each line is for an agent, respectively

5.3 Collision Avoidance with Colliding Trajectories

With the same UAV model, a time-varying formation was tested in order to verify the collision avoidance capacity in extreme situations. Indeed in this scenario, the UAVs $i = 1$ and $i = 3$ are set to head for a front-to-front collision. The formation set is equal to the previous case, except that after 150 s, agents 1 and 3 exchange their positions in the formation. The reference is kept constant ($\omega_r = 0$ rad/s) with the same linear velocity.

The results achieved can be visualized and compared with [17] in Fig. 12 (the trajectory of the formation in the North-East plane) and Fig. 13 (distance between each pair of UAVs through time). It can be observed from Fig. 12 that the hybrid system provides a flight with fewer oscillations compared with the Dec-POMDP alone. In addition, Fig. 13 shows that, although it was necessary a larger deviation from the path, the collision avoidance was not only maintained but actually increased by 25.2% during the position exchange (from a minimum distance of 3.19 m of [17] to 3.99 m). Figure 14 shows in which moment each of the controllers acts. There are 3 moments after the formation change in which the Dec-POMDP policies are activated. However, they are short in time, with a maximum of 0.43 s. This shows the fast response that takes the UAVs away from the danger zone.

A comparison is also made with a simple heuristics method and with a PID alone in the same situation. In the

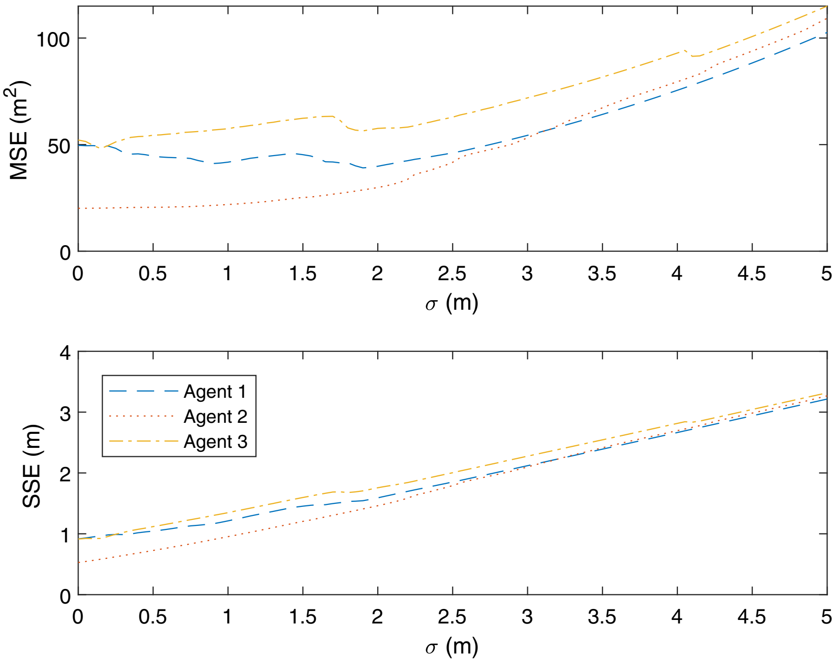


Fig. 10 Error parameters in reference tracking simulations with increasing noise standard deviation. Errors increase but the system remains within the desired target region

first case, when the vehicles achieve the states $s \leq 2$, instead of activating the Dec-POMDP policy, it uses a heuristic approach to avoid a collision, i.e. one agent turns left, and the other turns right with an angular velocity of $\omega_i = 0.9$ rad/s. Also, they reduce their linear velocity by 10 m/s. In the second case, all of the fleet's performance is made with the PIDs, without activating any collision avoidance algorithm at any time.

Table 6 gives better detail on this comparison by analyzing parameters such as the minimum distance between agents (MDBA) and the maximum time in which an agent remains inside the danger zone (i.e. when the MDBA is below r_p). In addition, the table provides the information if it was verified collision among any UAV pair, i.e. when their distance is below r_s. From these results, it can be seen that indeed the hybrid system is the best method in avoiding collision among the ones tested, providing a safe distance of at least 3.99 m between agents. Alongside the Dec-POMDP only of [17], the hybrid system achieves collision avoidance, which was not accomplished with PID only and simple heuristics methods. In addition, the hybrid system is able to maintain the UAVs in the danger zone by

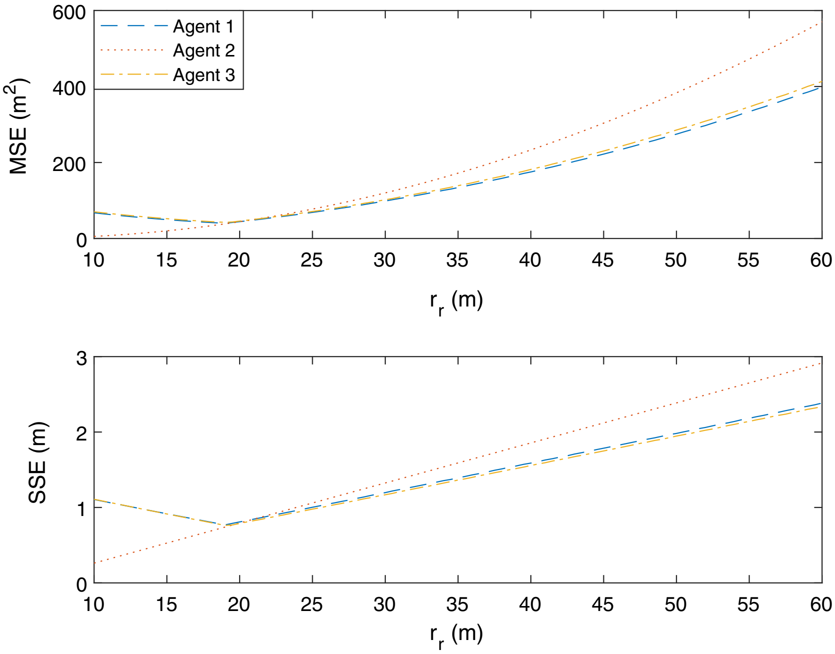
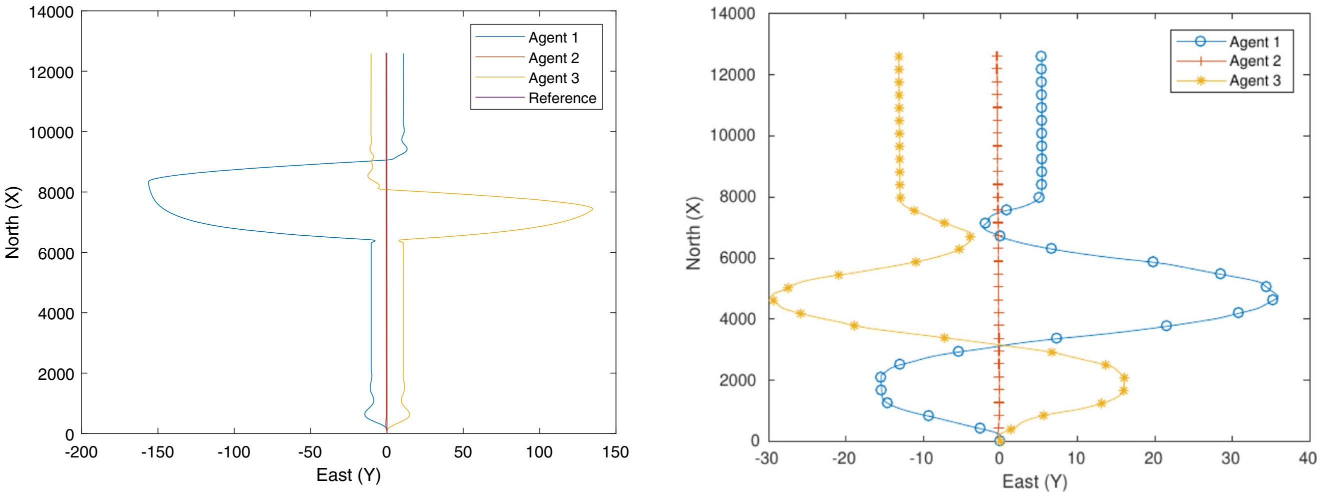


Fig. 11 Error parameters in reference tracking simulations with incresing r_r. Such increase does not affect significantly the errors' parameters as all the vehicles remain within the target region



(a) Hybrid system

(b) Dec-POMDP [17]

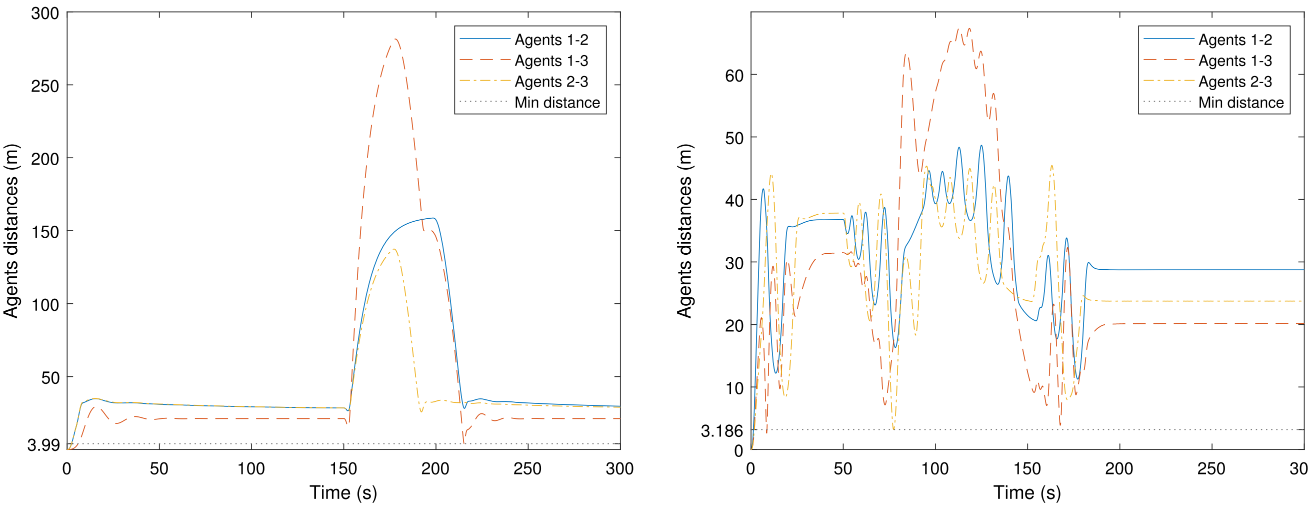
Fig. 12 UAV trajectories in North-East (horizontal) plane with colliding formation

the least amount of time, 0.43 s. Therefore, it not only avoids collision but gets a fast response when in danger.

5.3.1 Noise Evaluation in Collision Avoidance

Similar to the previous section, the collision avoidance scenario was also extensively tested in scenarios with noise. Once more, the noise standard deviation σ was swept and checked the parameters SSE and minimum distance for each simulation. The result can be seen in Fig. 15. It can be seen that the collision is maintained for the most part of the simulations. Indeed, the collision is only verified when $1.60 \text{ m} \leq \sigma \leq 1.65 \text{ m}$, in which the MDBA gets below $r_s = 3$ m. In addition, the steady-state error (SSE) increases with the noise, which is expected as the agents receive corrupted data. However, the SSE parameters remained within the allowable margin of $r_d = 7$ m. Beyond that point ($\sigma > 1.65$ m), the noise becomes significantly higher which results in the loss of the reference track.

Finally, the same noise evaluation for the PID only and heuristic methods was performed. The results are shown in Fig. 16. The graph shows that indeed, non of those methods were able to avoid a collision, being the hybrid system the only one which maintained a collision avoidance for the most part of the simulations.



(a) Hybrid system

(b) Dec-POMDP [17]

Fig. 13 UAV inter-distances evolution with time with colliding formation. Hybrid system provides better collision avoidance as the minimum distance between agents is larger

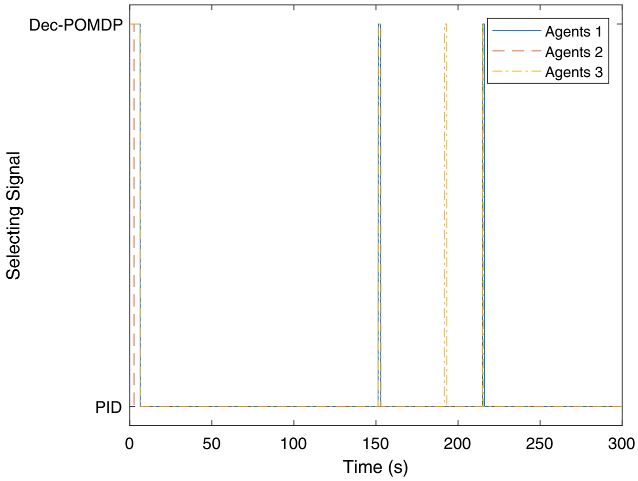


Fig. 14 UAVs selecting signal in collision avoidance

5.4 Hybrid System in Different UAV Models

Let us consider the guidance solution based on the sliding mode given by [61]. For a fair comparison, the references and formations were taken from [61], as well as the UAV model and inner loop controls. Since the models were different from the ones used in the previous case, another PID tuning was required. Considering the same method for tuning, the parameters found were $K_{Pv} = 0.02$, $K_{Iv} = 0.00015$ and $K_{Dv} = 0$ in the linear velocity branch and $K_{P\omega} = 0.2$, $K_{I\omega} = 1$ and $K_{D\omega} = 0$ for the angular velocity. Then, unlike the previous system, this model, requires a PI control, without a derivative term.

The reference follows a path described by an initial heading angle of $\pi/4$ rad that suffers from a change (ramp) between 250 and 267 seconds of -0.1 rad/s. In its turn, the linear velocity is defined as starting at 2 m/s, the reference changes linearly to 10 m/s between 350 and 370 s; later on it changes again to 12 m/s between times 450 and 460 s, keeping this value until 800 s.

The formation desired is a equilateral triangle within a circle of radius $r_r = 150$ m and the reference at its center. In addition the other constants were set to $r_d = 60$ m, $r_p = 5$ m and $r_c = 300$ m. Agent 2 is set to be right in front of the reference ($\theta_2 = \pi/2$ rad) while agents 1 and 3 are positioned at $\theta_1 = 7\pi/6$ rad and $\theta_3 = -\pi/6$ rad, respectively.

The results achieved can be visualized and compared to [61] in Fig. 17 (UAVs' trajectories) and in Table 7 (MSE). Figure 17 shows that the trajectories followed by the

Table 6 Control parameters in collision avoidance simulation with different methods

Method	MDBA (m)	Collision occurrence	Time in danger zone (s)
PID only	0.31	Yes	0.93
Simple heuristics	1.09	Yes	2.32
Dec-POMDP [17]	3.186	No	1.64
Hybrid system	3.99	No	0.43

Only Dec-POMDP and hybrid system avoids collision. The later spends the least time within danger zone

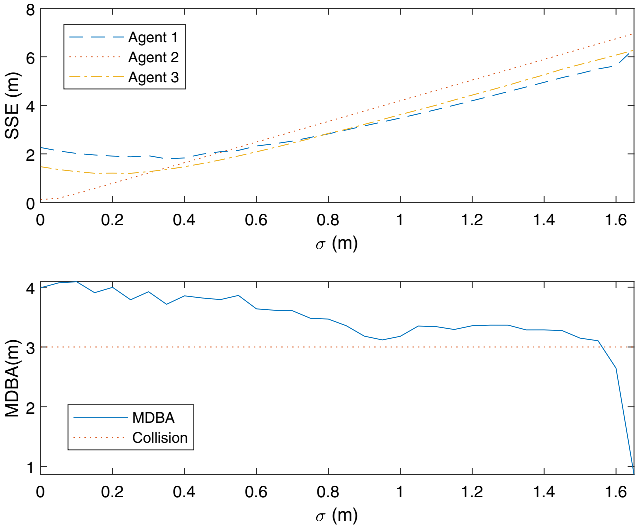


Fig. 15 Error and distance parameters in collision avoidance simulations with swept noise standard deviation. The errors increase with the noise, but remain within the allowable margin, while the collision not avoided only for $\sigma > 1.6$ m

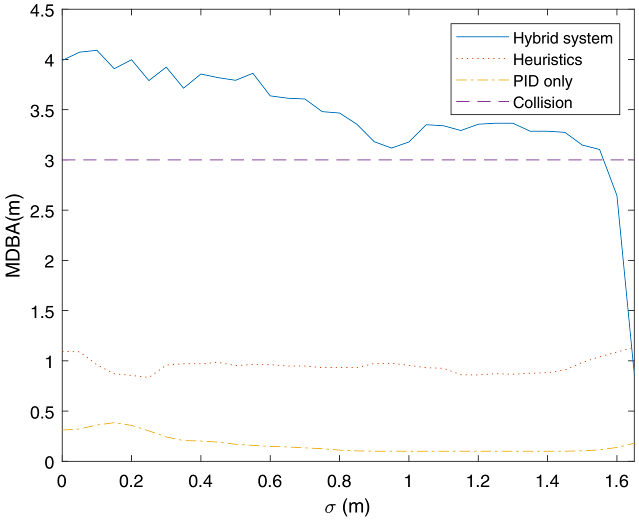
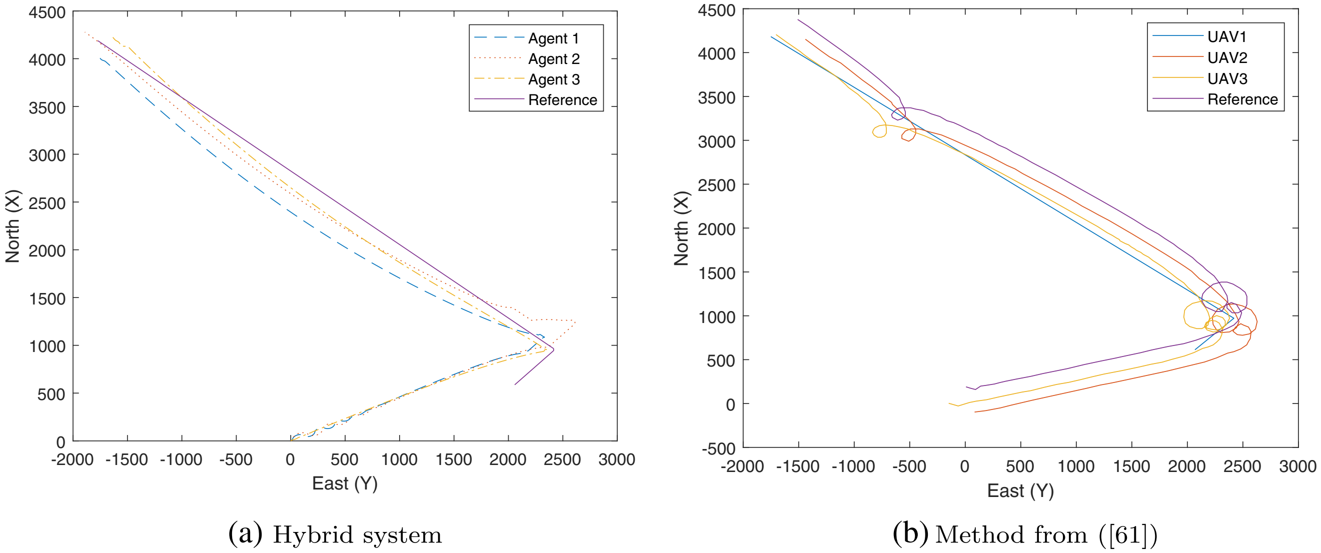


Fig. 16 Minimum distance between agents in collision avoidance simulations with increasing noise standard deviation. Hybrid system provides the better collision avoidance even in noisy environments



(a) Hybrid system

(b) Method from ([61])

Fig. 17 UAV trajectories in North-East (horizontal) plane with reference and UAV model based on [61]. The trajectories are similar and comparable, but the hybrid system does not use loitering

fleet were quite similar to the path achieved in [61], while maintaining the formation.

In addition, Table 7 shows the mean squared error as measured for each agent. Although the results point the hybrid system to have a better mean performance is important to highlight the fact that the method proposed in [61] involves loitering which is not the case in the hybrid method and therefore might be the reason for that great difference. However, the comparison is still valid in order to confirm that the proposed hybrid system is indeed comparable to other existing methods.

5.4.1 Noise Evaluation with Different UAV Model

As in previous sections, noise evaluation of the hybrid system with the model from [61] (which did not account for noisy measurements). The same scan, changing σ, were performed. The steady-state error (SSE) achieved is shown in Fig. 18. The SSE evolution follows the same pattern as observed in Section 5.2.1, increasing along with the noise. However, as before, the UAVs remained within the target margin of r_d.

Table 7 Error parameters in reference tracking simulation with different methods

Method	MSE (m^2)
Hybrid system ($\sigma = 0$)	$3.28 \cdot 10^5$ $2.34 \cdot 10^5$ $4.39 \cdot 10^5$
Method from [61]	$4.16 \cdot 10^5$ $4.40 \cdot 10^5$ $4.55 \cdot 10^5$

For each method, each line is for an agent, respectively

5.5 Horizon Analysis

The optimization of this problem was also performed with a horizon of $h = 2$, resulting in the policy showed in Fig. 19. The description of the action's values is the same given in Table 4.

Table 8 shows the comparison of the system's performance with $h = 2$ and $h = 3$ in both reference tracking

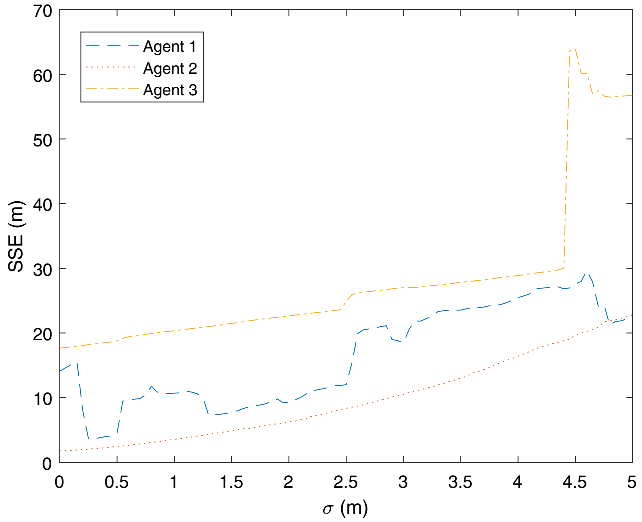


Fig. 18 Steady-state error with increasing noise standard deviation and UAV model from [61]. SSE increases but remains within target margin

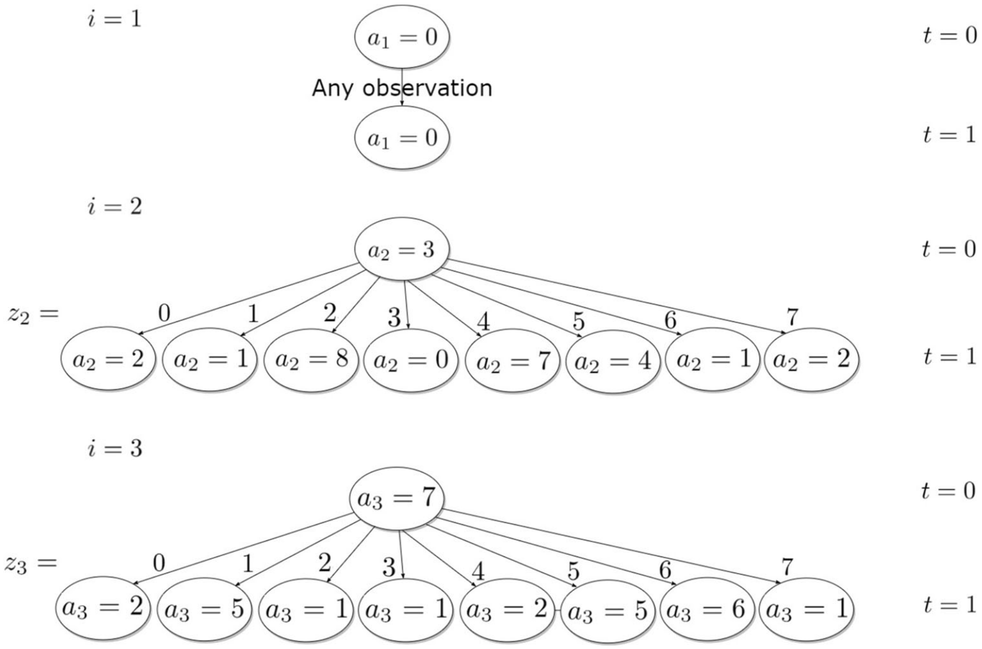


Fig. 19 Resulting policy of optimization with $h = 2$

and collision avoidance simulations. In all of the parameters compared, the larger horizon achieved a better result. In reference tracking, the system with $h = 3$ achieves smaller errors (both SSE and MSE) and faster response (smaller settling time). In the collision avoidance part, the system with a larger horizon spent the least time in the danger zone, its MDBA was larger and it actually prevented the collision from happening.

5.5.1 Noise Evaluation with Different Horizon

In addition, the noise response of the system with $h = 2$ was also tested. Sweeping σ resulted in Fig. 20. Results show that the horizon $h = 2$ fails in avoiding collision in many distinct places. While $h = 3$ failed in avoiding collision at only two points ($\sigma > 1.6$ m), the smaller horizon failed at 20 of the 101 tested points, including the ones in which $0 \text{ m} \le \sigma \le 0.7$ m, $1.1 \text{ m} \le \sigma \le 1.35$ m and $\sigma = 1.95$ m.

5.6 Implementation in Real Platforms

For practical implementation of the proposed hybrid Dec-POMDP/PID strategy, the requirements on the hardware system should essentially be the same as those for implementation of the Dec-POMDP policies in [12, 26, 43]. The main issues are related to sensors for state observation and communication channel reliability. It is expected that the most used sensor systems are sufficient for the proposed controller implementation. The sensor system can be based on cameras (as in [12, 35, 43]), Inertial Measurement Units (IMU) (as in [9, 23, 30]) and laser range finders ([26]). As a guideline for channel communication implementation, it is a good policy to keep the package loss rate up to 10% as recommended in [23] and a maximum delay of 5 ms as recommended in [30]. The processing of the control strategy should not be a problem in most situations due to the growing processing power of the modern central

Table 8 Horizons comparison

Horizon	Reference Tracking			Collision Avoidance		
	SSE (m)	MSE (m^2)	Settling Time (s)	Time in danger zone (s)	MDBA (m)	Collision occurrence
$h = 2$	1.85	180.70	132.78	0.75	1.65	Yes
	3.33	602.07	172.94			
	1.86	188.74	134.95			
$h = 3$	0.92	49.64	85.53	0.43	3.99	No
	0.53	20.13	50.01			
	0.91	52.16	87.77			

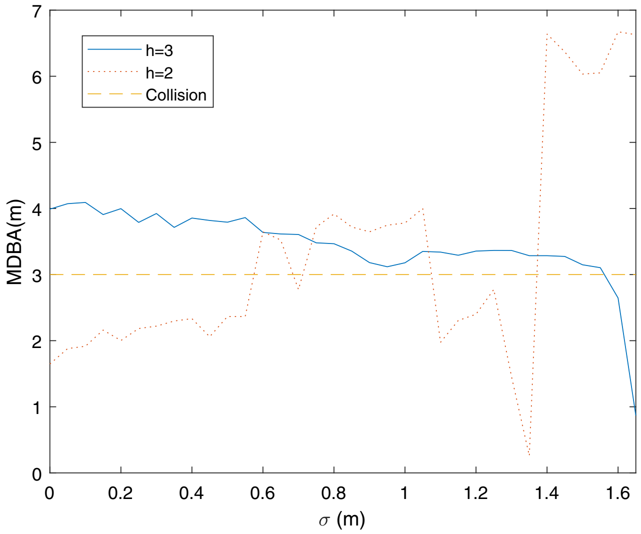


Fig. 20 Minimum distance between agents in collision avoidance simulations with swept noise standard deviation for $h = 2$ and $h = 3$. Collision avoidance is not achieved in many noisy points with $h = 2$

processing units (CPUs) and the fact that the optimization step is performed off-line.

6 Conclusions

The formation flight problem requires complex and dynamic objectives to be performed including reference and formation tracking, collision and obstacle avoidance, noise and communication robustness and many others. As systems with multiple loops, the fleet's control may require different types of commands. Therefore, a hybrid system, based in PID algorithms and Dec-POMDP models was proposed, in order to achieve an effective performance while still avoiding a collision.

The Dec-POMDP model showed to be a suitable model to be used in a hybrid method. Due to its decentralized approach, this model is proper to a flight scenario, in which the environment is very uncertain and a given UAV should not rely on a central decision-making unit.

Through an extensive range of simulations, it was possible to see that indeed the PID addition to the Dec-POMDP policies has several advantages as it is able to enhance the tracking capability of a given reference and formation while in emergency situations, as was the case for collisions, were avoided. Some parameters, such as the settling time, were worsen compared to previous works, but are compensated by the collision avoidance capability, as well as the steady-state error and mean-squared error.

Suggestions for future works include the refinement of the PID controllers, possibly replacing them with another continuous system; testing different actions to be performed by the Dec-POMDP policy for other applications such as quadrotors, helicopters and even non-aerial vehicles and systems; verifying the delay effects in the system and using an online policy that is able to compute the decisions in real-time in response to the environment. The implementation of this system in real UAVs is also desired as well the test for its scalability by evaluating a scenario with more than 3 agents.

Appendix A

A.1 Transition Matrices

The transition probability matrices used in Section 5, T_v^i and T_ω^i, are described as follows (the number in parentheses is the index β_v and β_ω that multiplies the base values α_{vb} or $\alpha_{\omega b}$ respectively, of each matrix):

$$T_v^1(-1) = \begin{bmatrix} 0.2 & 0.7 & 0 & 0.05 & 0 & 0.05 & 0 & 0 \\ 0.1 & 0.2 & 0 & 0.6 & 0 & 0.1 & 0 & 0 \\ 0.6 & 0.1 & 0.2 & 0 & 0 & 0 & 0.1 & 0 \\ 0.2 & 0 & 0 & 0.1 & 0 & 0.6 & 0 & 0.1 \\ 0.1 & 0.05 & 0.6 & 0 & 0.1 & 0 & 0.1 & 0.05 \\ 0.05 & 0 & 0 & 0 & 0 & 0.2 & 0 & 0.75 \\ 0.05 & 0 & 0.05 & 0 & 0.6 & 0.05 & 0.2 & 0.05 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0.3 & 0.4 \end{bmatrix}$$

$$T_v^1(0) = 0.8I + \frac{0.2}{7}(1_{8\times 8} - I),$$

in which I is the 8-th dimensional identity matrix and 1_{8x8} is the 8-th dimensional square matrix completely filled with 1.

$$T_v^1(1) = \begin{bmatrix} 0.2 & 0 & 0.7 & 0 & 0.05 & 0 & 0.05 & 0 \\ 0.6 & 0.2 & 0.1 & 0 & 0 & 0.1 & 0 & 0 \\ 0.1 & 0 & 0.2 & 0 & 0.6 & 0 & 0.1 & 0 \\ 0.1 & 0.6 & 0.05 & 0.1 & 0 & 0.1 & 0 & 0.05 \\ 0.2 & 0 & 0 & 0 & 0.1 & 0 & 0.6 & 0.1 \\ 0.05 & 0.05 & 0 & 0.6 & 0.05 & 0.2 & 0 & 0.05 \\ 0.05 & 0 & 0 & 0 & 0 & 0 & 0.2 & 0.75 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0.3 & 0.4 \end{bmatrix}$$

Due to the symmetry of the problem:

$$T_v^2(-1) = T_v^1(1)$$

$$T_v^2(0) = T_v^1(0)$$

$$T_v^2(1) = T_v^1(-1)$$

$$T_v^3 = T_v^1$$

$$T_\omega^1(-1) = \begin{bmatrix} 0.2 & 0.2 & 0.2 & 0.1 & 0.1 & 0.1 & 0.1 & 0 \\ 0 & 0.2 & 0.1 & 0.3 & 0 & 0.4 & 0 & 0 \\ 0 & 0.1 & 0.2 & 0 & 0.3 & 0 & 0.4 & 0 \\ 0 & 0 & 0 & 0.1 & 0 & 0.4 & 0.2 & 0.3 \\ 0 & 0 & 0 & 0 & 0.1 & 0.2 & 0.4 & 0.3 \\ 0 & 0 & 0 & 0 & 0 & 0.2 & 0.1 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 0.1 & 0.2 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0.3 & 0.4 \end{bmatrix}$$

$$T_\omega^1(0) = T_v^1(0)$$

$$T_\omega^1(1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.8 & 0.1 & 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0.8 & 0 & 0.1 & 0 & 0 & 0 & 0.1 & 0 \\ 0.5 & 0.3 & 0 & 0 & 0 & 0.2 & 0 & 0 \\ 0.5 & 0 & 0.3 & 0 & 0 & 0 & 0.2 & 0 \\ 0.1 & 0.5 & 0 & 0.1 & 0 & 0.2 & 0 & 0.1 \\ 0.1 & 0 & 0.5 & 0 & 0.1 & 0 & 0.2 & 0.1 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0.3 & 0.4 \end{bmatrix}$$

$$T_\omega^2(-1) = \begin{bmatrix} 0.1 & 0.45 & 0.45 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0.2 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.2 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 \\ 0 & 0 & 0 & 0 & 0 & 0.3 & 0.3 & 0.4 \end{bmatrix}$$

$$T_\omega^2(0) = T_\omega^1(0)$$
$$T_\omega^2(1) = T_\omega^2(-1)$$
$$T_\omega^3(-1) = T_\omega^1(1)$$
$$T_\omega^3(0) = T_\omega^1(0)$$
$$T_\omega^3(1) = T_\omega^1(-1)$$

A.2 Proof of Eq. 5

There is a total of N agents that get some observation z_i. However, a total of $N_{\zeta_i} = j$ of those agents that perceive $z_i = s'$ can be allocated. Since these agents were fixed, there are another $N - j$ agents that might observe any from the $N_{z_i} - 1$ observations in which $z_i \neq s'$. In this arrangement, the number of combinations is given by

$$N_{comb} = (N_{z_i} - 1)^{N-j}. \tag{8}$$

However, there are other arrangements for the j agents that observe $z_i = s'$. The number of possible permutations to fix those agents is given by

$$N_{arr} = \frac{N!}{j!(N-j)!}. \tag{9}$$

For each arrangement, there are N_{comb} combinations so the total number of combinations, D_j, is given by

$$D_j = N_{arr} N_{comb}. \tag{10}$$

Replacing (8) and (9) in (10) results in Eq. 5.

Acknowledgements The authors would like to thank the Brazilian National Council for the Improvement of Higher Education (CAPES) and the Brazilian National Council of Scientific and Technological Development (CNPq) for the support.

Author Contributions All authors contributed to the study conception, design and methodology. Bruno R. O. Floriano was also responsible for the investigation, software and writing. Geovany A. Borges, Henrique C. Ferreira and João Y. Ishihara were also responsible for writing and revision. All authors read and approved the final manuscript.

Funding This research was funded to the researchers Bruno R. O. Floriano and João Y. Ishihara, respectively, by the institutions:

- Brazilian National Council for the Improvement of Higher Education (CAPES)
- Brazilian National Council of Scientific and Technological Development (CNPq)

Availability of Data and Materials Custom data and materials are not available to the general public

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Albaker, B.M., Rahim, N.A.: Flight path PID controller for propeller-driven fixed-wing unmanned aerial vehicles. Int. J. Phys. Sci. **6**(8), 1947–1964 (2011)
2. Amato, C., Chowdhary, G., Geramifard, A., Ure, N.K., Kochenderfer, M.J.: Decentralized control of partially observable Markov decision processes. In: 52nd IEEE Conference on Decision and Control, pp. 2398–2405. IEEE (2013)
3. Amato, C., Konidaris, G., Anders, A., Cruz, G., How, J.P., Kaelbling, L.P.: Policy search for multi-robot coordination under uncertainty. Int. J. Robot. Res. **35**(14), 1760–1778 (2016)
4. Arifianto, O., Farhood, M.: Optimal control of a small fixed-wing UAV about concatenated trajectories. Control. Eng. Pract. **40**, 113–132 (2015)
5. Bai, H., Hsu, D., Kochenderfer, M.J., Lee, W.S.: Unmanned aircraft collision avoidance using continuous-state POMDPs. Robot. Sci. Syst. VII **1**, 1–8 (2012)
6. Bernstein, D.S., Givan, R., Immerman, N., Zilberstein, S.: The complexity of decentralized control of markov decision processes. Math. Oper. Res. **27**(4), 819–840 (2002)
7. Brunskill, E., Kaelbling, L.P., Lozano-Perez, T., Roy, N.: Continuous-state POMDPs with hybrid dynamics. In: ISAIM (2008)
8. Cai, Z., Wang, L., Zhao, J., Wu, K., Wang, Y.: Virtual target guidance-based distributed model predictive control for formation control of multiple UAVs. Chin. J. Aeronaut. **33**(3), 1037–1056 (2020)
9. Campa, G., Gu, Y., Seanor, B., Napolitano, M.R., Pollini, L., Fravolini, M.L.: Design and flight-testing of non-linear formation control laws. Control. Eng. Pract. **15**(9), 1077–1092 (2007)
10. Candido, S., Hutchinson, S.: Minimum uncertainty robot navigation using information-guided POMDP planning. In: 2011 IEEE International Conference on Robotics and Automation, pp. 6102–6108. IEEE (2011)

11. Capitan, J., Merino, L., Ollero, A.: Cooperative decision-making under uncertainties for multi-target surveillance with multiples UAVs. J. Intell. Robot. Syst. **84**(1-4), 371–386 (2016)
12. Capitan, J., Spaan, M.T., Merino, L., Ollero, A.: Decentralized multi-robot cooperation with auctioned POMDPs. Int. J. Robot. Res. **32**(6), 650–671 (2013)
13. Chen, Y.B., Luo, G.C., Mei, Y.S., Yu, J.Q., Su, X.L.: UAV Path planning using artificial potential field method updated by optimal control theory. Int. J. Syst. Sci. **47**(6), 1407–1420 (2016)
14. Cordeiro, T.F.K., Ferreira, H.C., Ishihara, J.Y.: Non linear controller and path planner algorithm for an autonomous variable shape formation flight. In: 2017 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 1493–1502. IEEE (2017)
15. D'Amato, E., Mattei, M., Notaro, I.: Distributed reactive model predictive control for collision avoidance of unmanned aerial vehicles in civil airspace. J. Intell. Robot. Syst. **97**(1), 185–203 (2019)
16. Dong, X., Zhou, Y., Ren, Z., Zhong, Y.: Time-varying formation control for unmanned aerial vehicles with switching interaction topologies. Control. Eng. Pract. **46**, 26–36 (2016)
17. Floriano, B., Borges, G.A., Ferreira, H.: Planning for decentralized formation flight of UAV fleets in uncertain environments with Dec-POMDP. In: 2019 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE (2019)
18. Gabel, T., Riedmiller, M.: Joint equilibrium policy search for multi-agent scheduling problems. In: German Conference on Multiagent System Technologies, pp. 61–72. Springer (2008)
19. He, L., Bai, P., Liang, X., Zhang, J., Wang, W.: Feedback formation control of UAV swarm with multiple implicit leaders. Aerosp. Sci. Technol. **72**, 327–334 (2018)
20. Jin, J., Ramirez, J.P., Wee, S., Lee, D., Kim, Y., Gans, N.: A switched-system approach to formation control and heading consensus for multi-robot systems. Intell. Serv. Robot. **11**(2), 207–224 (2018)
21. Kaelbling, L.P., Littman, M.L., Cassandra, A.R.: Planning and acting in partially observable stochastic domains. Artif. Intell. **101**(1-2), 99–134 (1998)
22. Kahan, W.: Ieee standard 754 for binary floating-point arithmetic. Lect. Notes Stat. IEEE **754**(94720-1776), 11 (1996)
23. Kartal, Y., Subbarao, K., Gans, N.R., Dogan, A., Lewis, F.: Distributed backstepping based control of multiple uav formation flight subject to time delays. IET Cont. Theor. Appl. **14**(12), 1628–1638 (2020)
24. Kuriki, Y., Namerikawa, T.: Formation control with collision avoidance for a multi-UAV system using decentralized mpc and consensus-based control. SICE J. Cont. Measure. Syst. Integrat. **8**(4), 285–294 (2015)
25. Lao, M., Tang, J.: Cooperative multi-UAV collision avoidance based on distributed dynamic optimization and causal analysis. Appl. Sci. **7**(1), 83 (2017)
26. Lauri, M., Heinänen, E., Frintrop, S.: Multi-robot active information gathering with periodic communication. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 851–856. IEEE (2017)
27. Li, X., Fang, Y., Fu, W.: Obstacle avoidance algorithm for Multi-UAV flocking based on artificial potential field and Dubins path planning. In: 2019 IEEE International Conference on Unmanned Systems (ICUS), pp. 593–598. IEEE (2019)
28. Liao, F., Teo, R., Wang, J.L., Dong, X., Lin, F., Peng, K.: Distributed formation and reconfiguration control of vtol UAVs. IEEE Trans. Control Syst. Technol. **25**(1), 270–277 (2017)
29. Lin, Z., Castano, L., Mortimer, E., Xu, H.: Fast 3d collision avoidance algorithm for fixed wing UAS. J. Intell. Robot. Syst. **97**(3), 577–604 (2020)
30. Liu, Y., Montenbruck, J.M., Zelazo, D., Odelga, M., Rajappa, S., Bülthoff, H.H., Allgöwer, F., Zell, A.: A distributed control approach to formation balancing and maneuvering of multiple multirotor uavs. IEEE Trans. Robot. **34**(4), 870–882 (2018)
31. Lwin, N., Tun, H.M.: Implementation of flight control system based on Kalman and PID controller for UAV. Int. J. Scientif. Technol. Res. **3**(4), 309–312 (2014)
32. Marcosig, E.P., Giribet, J.I., Castro, R.: Hybrid adaptive control for UAV data collection: A simulation-based design to trade-off resources between stability and communication. In: 2017 Winter Simulation Conference (WSC), pp. 1704–1715. IEEE (2017)
33. Meng, W., He, Z., Su, R., Yadav, P.K., Teo, R., Xie, L.: Decentralized multi-UAV flight autonomy for moving convoys search and track. IEEE Trans. Control Syst. Technol. **25**(4), 1480–1487 (2017)
34. Miller, S.A., Harris, Z.A., Chong, E.K.: A POMDP framework for coordinated guidance of autonomous UAVs for multitarget tracking EURASIP. J. Adv. Signal Process. **2009**(724597) (2009)
35. Montufar, D., Munoz, F., Espinoza, E., Garcia, O., Salazar, S.: Multi-Uav testbed for aerial manipulation applications. In: 2014 International Conference on Unmanned Aircraft Systems (ICUAS), pp. 830–835. IEEE (2014)
36. Mukherjee, S., Namuduri, K.: Formation control of UAVs for connectivity maintenance and collision avoidance. In: 2019 IEEE National Aerospace and Electronics Conference (NAECON), pp. 126–130 (2019)
37. Nair, R., Tambe, M.: Hybrid BDI-POMDP framework for multiagent teaming. J. Artif. Intell. Res. **23**, 367–420 (2005)
38. Nair, R., Tambe, M., Yokoo, M., Pynadath, D., Marsella, S.: Taming decentralized POMDPs: Towards efficient policy computation for multiagent settings. In: IJCAI, vol. 3, pp. 705–711 (2003)
39. Oliehoek, F.A., Amato, C., et al.: A concise introduction to decentralized POMDPs, vol. 1. Springer, New York (2016)
40. Oliehoek, F.A., Spaan, M.T., Vlassis, N.: Optimal and approximate q-value functions for decentralized pomdps. J. Artif. Intell. Res. **32**, 289–353 (2008)
41. Oliehoek, F.A., Spaan, M.T., Vlassis, N., et al.: Dec-POMDPs with delayed communication. In: The 2nd Workshop on Multi-Agent Sequential Decision-Making in Uncertain Domains (2007)
42. Oliehoek, F.A., Spaan, M.T.J., Terwijn, B., Robbel, P., ao, V., Messias, J.: The MADP toolbox: An open source library for planning and learning in (multi-)agent systems. J. Mach. Learn. Res. **18**(89), 1–5 (2017)
43. Omidshafiei, S., Agha-Mohammadi, A.A., Amato, C., Liu, S.Y., How, J.P., Vian, J.: Decentralized control of multi-robot partially observable markov decision processes using belief space macro-actions. Int. J. Robot. Res. **36**(2), 231–258 (2017)
44. Pajarinen, J., Kyrki, V., Koval, M., Srinivasa, S., Peters, J., Neumann, G.: Hybrid control trajectory optimization under uncertainty. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE (2017)
45. Paquet, S., Chaib-draa, B., Ross, S.: Hybrid POMDP algorithms. In: Proceedings of The Workshop on Multi-Agent Sequential Decision Making in Uncertain Domains (MSDM-06), pp. 133–147 (2006)
46. Ragi, S., Chong, E.K.P.: UAV Path planning in a dynamic environment via partially observable markov decision process. IEEE Trans. Aerosp. Electron. Syst. **49**(4), 2397–2412 (2013)
47. Ragi, S., Chong, E.K.P.: Decentralized guidance control of UAVs with explicit optimization of communication. J. Intell. Robot. Syst. **73**(1-4), 811–822 (2014)
48. Rens, G., Moodley, D.: A hybrid POMDP-BDI agent architecture with online stochastic planning and plan caching. Cogn. Syst. Res. **43**, 1–20 (2017)

49. Shames, I., Fidan, B., Anderson, B.D.: Close target reconnaissance with guaranteed collision avoidance. Int. J. Robust Nonlinear Control **21**(16), 1823–1840 (2011)
50. Shen, X., Fan, J., Wang, H.: Design and simulation of eight-rotor unmanned aerial vehicle based on hybrid control system. Int. J. Aerospace Eng. **2018** (2018)
51. Sreenath, K., Hill, C.R., Kumar, V.: A partially observable hybrid system model for bipedal locomotion for adapting to terrain variations. In: Proceedings of the 16th International Conference on Hybrid Systems: Computation and Control - HSCC'13. ACM Press, New York (2013)
52. Sun, J., Tang, J., Lao, S.: Collision avoidance for cooperative UAVs with optimized artificial potential field algorithm. IEEE Access **5**, 18,382–18,390 (2017)
53. Tang, J., Fan, L., Lao, S.: Collision avoidance for multi-UAV based on geometric optimization model in 3d airspace. Arab. J. Sci. Eng. **39**(11), 8409–8416 (2014)
54. Tanner, H.G.: Switched UAV-UGV cooperation scheme for target detection. In: Proceedings 2007 IEEE International Conference on Robotics and Automation, pp. 3457–3462. IEEE (2007)
55. Wang, C., Wang, J., Shen, Y., Zhang, X.: Autonomous navigation of UAVs in large-scale complex environments: A deep reinforcement learning approach. IEEE Trans. Veh. Technol. **68**(3), 2124–2136 (2019)
56. Wang, D., Fan, T., Han, T., Pan, J.: A two-stage reinforcement learning approach for multi-UAV collision avoidance under imperfect sensing. IEEE Robot. Autom. Lett. **5**(2), 3098–3105 (2020)
57. Wu, F., Ramchurn, S.D., Chen, X.: Coordinating human-UAV teams in disaster response. In: Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), pp. 524–530 (2016)
58. Zhang, D., Duan, H.: Switching topology approach for UAV formation based on binary-tree network. J. Franklin Inst. **356**(2), 835–859 (2019)
59. Zhang, J., Yan, J., Zhang, P.: Fixed-wing UAV formation control design with collision avoidance based on an improved artificial potential field. IEEE Access **6**, 78,342–78,351 (2018)
60. Zhang, J., Yan, J., Zhang, P., Kong, X.: Collision avoidance in fixed-wing UAV formation flight based on a consensus control algorithm. IEEE Access **6**, 43,672–43,682 (2018)
61. Zhang, M., Liu, H.H.T.: Cooperative tracking a moving target using multiple fixed-wing UAVs. J. Int. Robot. Syst. **81**(3-4), 505–529 (2016)
62. Zheng, Z., Qian, M., Li, P., Yi, H.: Distributed adaptive control for UAV formation with input saturation and actuator fault. IEEE Access **7**, 144,638–144,647 (2019)
63. Zhu, F., Antsaklis, P.J.: Optimal control of hybrid switched systems: A brief survey. Discrete Event Dynamic Syst. **25**(3), 345–364 (2015)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Bruno R. O. Floriano received a B.S. degree in Electrical Engineering in 2017 and an M.S. degree in Electronics and Automation Engineering in 2019, both from the University of Brasilia (UnB), Braslia, Brazil. He is currently a Ph.D. student in Electrical Engineering, at the same university. His research interests include cooperative control, robotics, aerospace systems and machine learning.

Geovany A. Borges received a PhD on robotics at Universit Montpellier II, France, in 2002. Since 2003 is associate professor at Electrical Eng. Dep. Of University of Brasilia, Brazil. In 2012, he spent a one semester sabatical with MERS-CSAIL research team at Massachussetts Instittue of Technology. He is coordinator of Lab. of Automation and Robotics at University of Brasilia. Since 2003, he has conducted dozens of private and public funded research and development projects, most of them in robotics, stochastic filtering, aerospacial systems and biomedical engineering.

Henrique C. Ferreira received a Ph.D. in Electric Engineering from the University of São Paulo, São Paulo, Brazil, in 2008. He is currently a professor at the electrical engineering department of the University of Brasilia (UnB), Braslia, Brazil, since 2009. His research interests include robust control, non-linear control, UAVs control and multi-agent systems control.

João Y. Ishihara received the Ph.D. degree in electrical engineering from the University of São Paulo, São Paulo, Brazil, in 1998. He is currently an Associate Professor with the University of Brasilia, Brasilia, Brazil. His research interests include robust filtering and control theory, singular systems, and robotics.