Journal of Intelligent & Robotic Systems (2022) 106: 11
https://doi.org/10.1007/s10846-022-01715-z

REGULAR PAPER

Autonomous 4D Trajectory Planning for Dynamic and Flexible Air Traffic Management

Christian Vitale[1,2] · Savvas Papaioannou[1,2] · Panayiotis Kolios[1,2] · Georgios Ellinas[1,2]

Received: 6 August 2021 / Accepted: 26 July 2022 / Published online: 22 August 2022

Abstract
With an ever increasing number of unmanned aerial vehicles (UAVs) in flight, there is a pressing need for scalable and dynamic air traffic management solutions that ensure efficient use of the airspace while maintaining safety and avoiding mid-air collisions. To address this need, a novel framework is developed for computing optimized 4D trajectories for UAVs that ensure dynamic and flexible use of the airspace, while maximizing the available capacity through the minimization of the aggregate traveling times. Specifically, a network manager (NM) is utilized that considers UAV requests (including start/target locations) and addresses inherent mobility uncertainties using a linear-Gaussian system, to compute efficient and safe trajectories. Through the proposed framework, a family of mathematical programming problems is derived to compute control profiles for both distributed and centralized implementations. Extensive simulation results are presented to demonstrate the applicability of the proposed framework to maximize air traffic throughput under probabilistic collision avoidance guarantees.

Keywords Unmanned autonomous vehicles · Motion planning · Cooperative autonomous driving

1 Introduction

Current air traffic management (ATM) architectures have strictly-constrained configuration options that limit their ability to efficiently manage the traffic demand. In doing so, the true capacity of the network remains unexplored. This problem is further exacerbated by the increasing introduction of UAVs which impose significantly more fluctuating demands in the utilization of the airspace. Hence, a radically different solution for ATM is needed to fully optimize the network flows and exploit the total capacity of the available airspace.

To address this challenge a fully dynamic ATM concept is developed that takes into consideration all capacity/demand aspects and mobility constraints to deliver a unified mathematical framework for computing optimized 4D, i.e., in space and time, trajectories. Notably, various alternative optimization strategies exist for collision avoidance among UAVs, broadly categorized in: (i) centralized or distributed 4D UAV trajectory computation exploiting deterministic motion models to ensure minimum distance between UAVs, e.g., [4] and [1], and (ii) sequential algorithms accounting for motion model uncertainties, e.g., [12, 26]. Nevertheless, techniques tackling cooperative objectives, such as the airspace capacity maximization in the presence of collision avoidance guarantees have not still been adequately addressed [20].

Hence, in this work a novel modeling framework is presented that maximizes the capacity of a flying environment accounting for approximated UAV mobility states and creating collision-free areas for UAVs to use. To do so, future locations of the UAVs are estimated considering the

This work was supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant 739551 (KIOS CoE - TEAMING) and Grant 101003439 (C-AVOID), and by the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy.

✉ Christian Vitale
vitale.christian@ucy.ac.cy

Savvas Papaioannou
papaioannou.savvas@ucy.ac.cy

Panayiotis Kolios
pkolios@ucy.ac.cy

Georgios Ellinas
gellinas@ucy.ac.cy

[1] KIOS Research and Innovation Center of Excellence, University of Cyprus, 1678 Nicosia, Cyprus

[2] Department of Electrical and Computer Engineering, University of Cyprus, 1678 Nicosia, Cyprus

propagated location uncertainty, as a function of the possible controls applied. Due to this characterization, future control profiles can be selected for each vehicle so that collision-free areas (characterized by ellipsoids as elaborated later), never intersect with each other. Considering a linear-Gaussian system for each UAV's motion model, this safety constraint is easily transformed into a time-dependent minimum distance between the ellipsoid barycenters (centroids). The resulting feasibility region (area outside the ellipsoids) is non-convex, leading to hard optimization problems. Thus, relaxations are made by introducing worst-case piecewise linear, hence convex, constraints to reduce complexity.

Overall, two alternative mobility control strategies suffice: (i) the `Distributed Optimum`: UAVs are ordered randomly at the moment of the optimization and controls are decided sequentially; (ii) the `Centralized Optimum`: the controls of all UAVs in the flying environment are decided jointly, i.e., in a coordinated manner at a central entity, the NM. In both cases, an elegant optimization problem is presented to choose among all control profiles satisfying the aforementioned safety constraint, aiming to minimize the *aggregated* flying time. This metric was chosen to maximize the capacity of the flying environment, since it equivalently maximizes the UAVs' average speed, hence avoiding the selection of controls that allow UAVs to wait for the flying environment to become free before moving. In addition to the optimal formulations, a family of sub-optimal Mixed Integer Quadratically Constrained Programs (MIQCPs), with the integer variables being binary variables, denoted as `Distributed` and `Centralized Dodecahedron`, are derived to address the aforementioned complexity of the concavity of the feasible region. Following this relaxation, valid upper bounds on the performance of the `Distributed` and `Centralized Optimum` are also derived. Summarizing, the contributions of this paper are:

- a novel framework for safe airspace capacity maximization, accounting for UAVs motion model uncertainty;
- the transformation of collision probability constraints into safe distance between UAVs' barycenters (centroids). For the case of the linear-Gaussian UAVs motion model, such transformation allows selecting jointly the UAVs controls, for the first time avoiding the typical assumption of known-a-priori UAV trajectories;
- a real-time solution for UAV flying time minimization, i.e., our `Distributed Dodecahedron` approach;
- the first optimization jointly selecting UAV controls to maximize cooperatively, in small scenarios, the airspace capacity (`Centralized Dodecahedron`).

The rest of the paper is organized as follows. Section 2 reviews the state-of-the-art, Section 3 presents the system model, while Section 4 showcases the optimization framework underlying the `Centralized` and `Distributed Optimum` solutions. Section 5 introduces the `Centralized` and `Distributed Dodecahedron` heuristics and a valid upper-bound is derived. Section 6 unveils the potential of the introduced `Centralized` and `Distributed Dodecahedron` optimizations, including also a comparison with the upper-bound on the optimal solution in several instances. Finally, Section 7 concludes the work and provides avenues for future research.

2 Related Work

A recent detailed survey on the topic can be found in [17]. In [23], the authors propose a method to design provably safe conflict resolution maneuvers between two aircraft by modeling the aircraft and the maneuvers as a hybrid control system and in [10], a protocol-based 2D conflict resolution method for multiple aircraft is presented. Although this approach exhibits an analytical solution, it assumes that aircraft can change headings instantaneously and requires synchronous maneuvers (i.e., all aircraft to change headings at the same time), while [22] extends this approach for 3D.

In a similar vein, the authors in [15] propose a decentralized collision avoidance policy for guiding multiple vehicles towards their assigned goal regions. Although this approach is scalable and completely decentralized, it does not account for random disturbances and unexpected events. Moreover, the authors in [4] pose the multi-UAV conflict resolution problem as an optimal control problem that minimizes a certain objective function, while maintaining safe distances among the UAVs. This method, however, is purely deterministic and does not consider the stochasticity of the model dynamics. Similarly, the methodology proposed in [1] uses model predictive control (MPC) to generate collision-free trajectories between networked vehicles with deterministic dynamics. The authors use Lagrangian relaxation to convert their non-convex centralized formulation into a convex semidefinite program which can be solved more efficiently. Additionally, the work in [5] presents a collision-free formation control method based on MPC, in which a collision avoidance cost function is added to the UAV's objective function as a penalty term to prevent it from colliding with nearby UAVs. In [7], the problem of multi-agent safe trajectory planning with collision avoidance constraints has been formulated as a sequential decision making problem and solved using deep reinforcement learning, while [14] proposes a Markov Decision Process (MDP)-based short-term conflict avoidance algorithm for an automated unmanned ATM system. The approach in [14] generates advisories for each aircraft to follow, and is based on decomposing a large multi-agent Markov decision process and fusing the solutions. More recently, the work in [2] also proposes an MDP-based

trajectory planner for the problem of multi-agent distributed trajectory planning in the presence of cooperative and non-cooperative actors in a high-density free flight airspace. Similarly, the work in [25] develops a message-based decentralized guidance algorithm which allows multiple aircraft to be safely guided to their destinations, while avoiding potential conflicts among them.

For non-deterministic UAVs motion models, the more relevant works include [13], in which a stochastic optimal control method is proposed for designing conflict-free 3D trajectories by considering the uncertainties during flight. Specifically, a probabilistic conflict detection algorithm is proposed based on a generalized polynomial chaos method and the resulting nonlinear programming problem is solved using sequential quadratic programming. More recently, in [26], a chance-constraint nonlinear MPC problem is presented for the distributed generation multi-UAV collision-free trajectories to minimize single-UAV controls, and in [12], multi-robot motion planning based on distributed model predictive control minimizes the UAVs' energy consumption. Apart from using different objectives, [12, 26] also introduce a potential field cost to violate, if required, the safe distance, without ensuring a bound on the maximum collision probability between UAVs at all times, as is done in our work.

While all these aspects are interesting, in the presence of model uncertainty, neither collective objectives, such as maximizing the airspace capacity (or equivalently minimizing the aggregate flying time), nor a solution with a guaranteed bound on collision probability have been considered.

3 System Model

This work assumes that a total of $\mathcal{M}$ UAVs communicate with a NM, asking permission to travel in a bounded 3D environment. A UAV's motion model follows the discrete-time linear dynamics below [12]:

$$x_t^j = \boldsymbol{\Phi} x_{t-1}^j + \Gamma u_{t-1}^j + G\alpha_{t-1}^j,\ j \in [1, ..., \mathcal{M}] \tag{1}$$

where $x_t^j = [\mathrm{x}^j, \dot{\mathrm{x}}^j]_t^\top \in \mathbb{R}^6$ denotes the state of the j^{th} UAV at time t which consists of position $\mathrm{x}_t^j = [p_x, p_y, p_z]_t^j \in \mathbb{R}^3$ and velocity $\dot{\mathrm{x}}_t^j = [v_x, v_y, v_z]_t^j \in \mathbb{R}^3$ components in 3D Cartesian coordinates. Each UAV j is controllable through $u_t^j = [\mathrm{u}_x^j, \mathrm{u}_y^j, \mathrm{u}_z^j]_t^\top \in \mathbb{R}^3$ which denotes the applied force (control) vector at time t, and $\alpha_t^j = [a_x^j, a_y^j, a_z^j]_t^\top \in \mathbb{R}^3 \sim \mathcal{N}(0, \Sigma_\alpha^j)$ denotes the perturbing acceleration noise, due to uncontrolled forces acting on the UAV, e.g., unwanted accelerations due to wind, drawn from a zero mean multivariate normal distribution with covariance matrix Σ_α^j [11]. The matrices $\boldsymbol{\Phi}$, Γ, and G are further given by:

$$\boldsymbol{\Phi} = \begin{bmatrix} \mathrm{I}_3 & \delta T \cdot \mathrm{I}_3 \\ 0_3 & \phi \cdot \mathrm{I}_3 \end{bmatrix}, \Gamma = \begin{bmatrix} 0_3 \\ \gamma \cdot \mathrm{I}_3 \end{bmatrix}, G = \begin{bmatrix} 0.5\delta T^2 \cdot \mathrm{I}_3 \\ \delta T \cdot \mathrm{I}_3 \end{bmatrix} \tag{2}$$

where δT denotes the sampling interval, I_3 and 0_3 are the identity matrix and zero matrix of dimension 3×3, respectively, and parameters ϕ and γ are given by $\phi = (1 - \eta)$ and $\gamma = \frac{\delta T}{m^j}$, where η is used to model the air resistance and m^j denotes the mass of UAV j. As shown in Eq. 1, the UAV dynamics obey the Markov property i.e., the state of a UAV at the next time step depends only upon its current state and control input. That said, given a known initial state x_0 and a sequence of control inputs $u_{[0:T-1]}$ over the planning horizon of T time-steps, the state $x_t, t \in [1, ..., T]$ of the UAV can be computed by recursive application of Eq. 1 as:

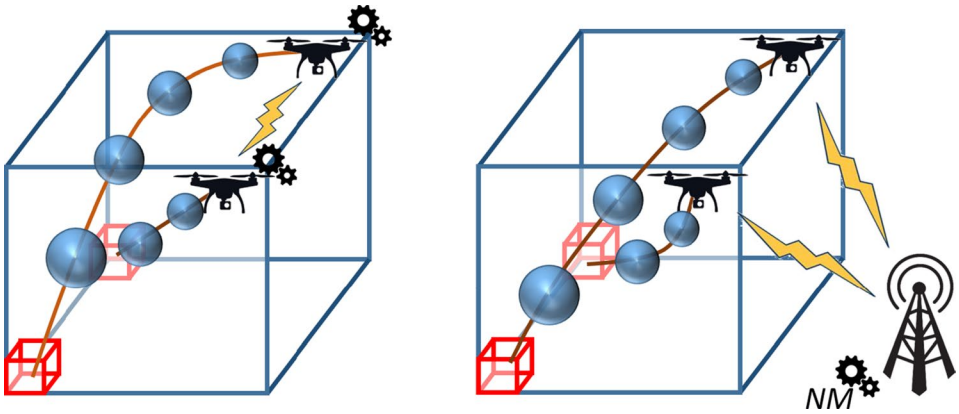
$$x_t = \boldsymbol{\Phi}^t x_0 + \sum_{\tau=0}^{t-1} \boldsymbol{\Phi}^\tau \left[\Gamma u_{t-\tau-1} + G\alpha_{t-\tau-1}\right], \forall t \tag{3}$$

where the agent (UAV) index j is omitted for notational clarity. Subsequently, the UAV trajectory $X_T = \{x_t\}, t \in [1, ..., T]$ over the planning horizon is a stochastic process, where each future state x_t is distributed according to $x_t \sim \mathcal{N}(\mu_t, \Xi_t)$ with $\mu_t = [\boldsymbol{\mu}, \dot{\boldsymbol{\mu}}]_t^\top$ and Ξ_t obtained as [8]:

$$\mu_t = \boldsymbol{\Phi}^t x_0 + \sum_{\tau=0}^{t-1} \boldsymbol{\Phi}^\tau \Gamma u_{t-\tau-1},\ \Xi_t = \sum_{\tau=0}^{t-1} \boldsymbol{\Phi}^\tau Q (\boldsymbol{\Phi}^\top)^\tau \tag{4}$$

where $Q = G\Sigma_\alpha G^\top$ is the covariance of the zero mean Gaussian disturbance acting on the system. Interestingly, the covariance matrix Ξ_t does not depend on the applied controls $u_{[0:T-1]}$, and can be easily pre-computed. As a consequence, selecting the controls for a UAV only modifies the average μ_t of the predicted system state, not its distribution. Furthermore, computing recursively μ_t, the average of the predicted system state only depends linearly on the applied controls.

Figure 1 summarizes the approaches used in this work to select the controls of the UAVs in the flying environment, i.e., the `Distributed Optimum` and the `Centralized Optimum` techniques. The adopted distributed control approach is based on the work in [19, 24], which is applied in a certain class of control problems that exhibit dynamically decoupled sub-systems with coupled constraints. In summary, this large planning optimization problem is divided into smaller sub-problems and the sub-problems are solved in sequence, e.g., following the order of the UAVs' entry time in the flying zone, by the distributed controllers located at the UAVs. Optimization plans are then communicated between the distributed controllers in order to guarantee the satisfaction of the coupled constraints. This procedure is usually referred to in the literature as sequential distributed control (or non-cooperative distributed control) and requires a form of coordination (i.e., ordering) to decide the UAVs' plans [6]. In our implementation, the NM



(a) Distributed Approach. (b) Centralized Approach.

Fig. 1 Trajectory planning proposed: (**a**) Distributed approach; UAVs decide their trajectory autonomously, following a sequential order and (**b**) Centralized approach; the NM receives UAVs' system state estimations and decides jointly future trajectories in the flying environment. As a result, UAVs leave enough space between them to maneuver, thus increasing the overall capacity of the flying environment. As explained in the following, spheres represent the areas containing the barycenters of UAVs with fixed probability. As derived in Eq. 4, perturbation noise is such that the uncertainty grows with time

acts as a coordinator, selecting the sequence used to decide the UAVs' controls, as well as forwarding messages among UAVs, in case of possible obstructions in the flying environment. For what concerns the adopted centralized control approach, the NM sets simultaneously the control profiles of all UAVs in the 3D flying zone. Choosing at the same time all controls $\mathbf{u}^{j}_{[0:T-1]}$, $\forall j$, even though it increases the computational complexity, allows to greatly improve the optimization of collaborative objective functions, as in this work. Indeed, in both the proposed optimizations, the objective is to minimize the UAVs' aggregate flying time between starting and target locations. Collaboratively, UAVs can synchronize their trajectories, leaving enough space between them to maneuver, thus increasing the overall capacity of the flying environment.

4 An Optimization Framework for UAV Collision-free Trajectory Planning

In this section, our first contribution is presented. Specifically, a novel framework for safe airspace capacity maximization that accounts also for UAVs' motion model uncertainty is showcased. Initially, a method to transform motion model uncertainty into collision probabilities and subsequently into a minimum safe distance among UAVs is presented (Section 4.1). Then, the two possible optimizations aiming at minimizing the UAVs' aggregate flying time whilst determining safe flying trajectories are introduced: (i) the `Distributed Optimum` (Section 4.2), and (ii) the `Centralized Optimum` (Section 4.3). Implicitly, minimizing the UAVs' flying time maximizes the capacity of the flying environment, since, in the obtained solutions, UAVs reach their target locations as quickly as possible, without waiting for the flying environment to become free before moving.

4.1 Transforming Target Collision Probability into a Safe Distance Among UAVs

To account for possible collisions during the UAV's path planning, the 3D-area containing the barycenter of a UAV with probability $1-\epsilon$, with ϵ arbitrarily small, is modeled as an ellipsoid based on the UAV's location distribution. The underlying idea is that, if, at any time t, the minimum distance between two points lying on the ellipsoids containing the barycenters of UAVs i and j is at least $d_{min} > 0$, then the probability of collision between the two UAVs is bounded. Specifically, the two UAVs *may* collide only if at least one of them is outside the ellipsoid. Hence, the maximum probability of collision P_c between i and j is:

$$P_c = 1-(1-\epsilon)^2 = 2\epsilon - \epsilon^2 \leq 2\epsilon \tag{5}$$

Distance d_{min} accounts for the dimensions of the UAVs and inter-UAV safe distance, i.e., $d_{min} = d_i + d_j + s$, where d_x represents the maximum distance between the barycenter and the chassis of UAV x, while s is the minimum safe distance between UAVs. The description of such ellipsoids follows well-known statistical results, which take into account the numerical integration of the distribution of a UAV's location [18]. Specifically, the 3×3 sub-matrix of Ξ_t, denoted hereinafter as M_t, is considered, describing the multi-variate Gaussian distribution $\boldsymbol{\mu}_t$ of the UAV's barycenter location.

The three semi-axes of the ellipsoid containing the location of the UAV's barycenter with probability $1-\epsilon$ are then:

$$a_{t,1} = \sqrt{K_\epsilon \lambda_{t,1}}; \quad a_{t,2} = \sqrt{K_\epsilon \lambda_{t,2}}; \quad a_{t,3} = \sqrt{K_\epsilon \lambda_{t,3}};$$

where $\lambda_{t,1}$, $\lambda_{t,2}$, and $\lambda_{t,3}$ are the eigenvalues of M_t, and K_ϵ is the inverse of the cumulative density function (CDF) of the chi-squared distribution having three degrees of freedom computed at $1-\epsilon$ [18]. In order for any two points of the ellipsoids to respect the minimum distance d_{min} at any time t, a sufficient distance $d_t^{i,j}$ between the expected UAVs' barycenters is imposed. Then, selecting control profiles such that the expected barycenter locations $\boldsymbol{\mu}_t^i$ and $\boldsymbol{\mu}_t^j$ respect $d_t^{i,j}$ ensures obtaining UAV trajectories with a bounded collision probability. For this task, the geometry space is exploited; initially, in order to simplify the computations, each ellipsoid is first approximated with a sphere with radius r_t equal to

$$r_t = \max(a_{t,1}, a_{t,2}, a_{t,3}). \tag{6}$$

Thus, any point in the ellipsoid is also in the sphere, and the probability of having the barycenter of the UAV within the sphere is larger than $1-\epsilon$. Note that, if all elements of Σ_α assume the same value and Σ_α is diagonal, then the ellipsoid and sphere descriptions correspond exactly. At this stage, the full description of the spheres describing each UAV's location is not available, since their center location $\boldsymbol{\mu}_t$ depends on the unknown controls. Nevertheless, imposing

$$d_t^{i,j} = r_t^i + r_t^j + d_{min}, \quad \forall t \tag{7}$$

ensures, as required, that any two points on the two spheres, hence on the ellipsoids as well, are at least d_{min} away. The computation of $d_t^{i,j}$ thus enables the transformation of a constraint on maximum collision probability into a minimum safe distance between the spheres' barycenters.

4.2 Minimizing Flying Time with a Distributed Approach

This section describes how control profiles for the UAVs are selected sequentially following a random order, i.e., the `Distributed Optimum`, with the objective of minimizing the UAVs' flying time. In this optimization, UAVs for which controls have been already selected, i.e., UAVs $\{1, ..., j-1\}$, broadcast their expected future locations $\boldsymbol{\mu}_t$. Thanks to such information, UAV j can safely decide its own controls. For UAV j, the present and future locations of other UAVs act as obstructions, whose locations are known at any time t. Such approach can be easily adapted to a situation where UAVs enter and leave the flying environment at different times, assuming that the trajectories of the UAVs are selected following a temporal arrival order.

For notational clarity, the flying zone is a cube of edge size l, centered at [0, 0, 0]. The target area for UAV j is represented by a cube of edge size l_G^j, centered around the target location $[p_{Gx}, p_{Gy}, p_{Gz}]^j$. An auxiliary binary variable b_t^j is 0 when UAV j is inside its target cube, and 1 otherwise. Hence, the overall objective of minimizing the flying time is presented as the minimization of the auxiliary variables b_t^j, $\forall t$, in the 3D flying zone.

Problem `Distributed Optimum` :

$$\min_{\mathrm{u}^j_{[0:T-1]}} \quad \sum_t b_t^j \tag{8a}$$

subject to:

$$\boldsymbol{\mu}_t^j \text{ as in (4)} \qquad \forall t \tag{8b}$$

$$|[\boldsymbol{\mu}_k]_t^j - [p_{Gk}]^j| \le l_G^j/2 + b_t^j M \qquad k \in \{x, y, z\}, \forall t \tag{8c}$$

$$b_t^j \in \{0, 1\}, b_T^j = 0 \qquad \forall t \tag{8d}$$

$$|[\boldsymbol{\mu}_x]_t^j|, |[\boldsymbol{\mu}_y]_t^j|, |[\boldsymbol{\mu}_z]_t^j| \le l/2 \qquad \forall t \tag{8e}$$

$$|\mathrm{u}_t^j| \le \mathrm{u}_{MAX} \qquad \forall t \tag{8f}$$

$$|\mathrm{u}_t^j - \mathrm{u}_{t-1}^j| \le \Delta \mathrm{u} \qquad \forall t \tag{8g}$$

$$|\dot{\boldsymbol{\mu}}_t^j| \le v_{MAX} \qquad \forall t \tag{8h}$$

$$|\boldsymbol{\mu}_t^j - \boldsymbol{\mu}_t^i| \ge d_t^{i,j} \qquad \forall i < j, \forall t \tag{8i}$$

In the `Distributed Optimum`, the absolute values present in Eqs. 8c-8d can be easily transformed into a series of additional linear constraints [21], without impairing the computational complexity of the proposed approach. Hence, Eqs. 8c-8d are linear constraints used to set the value of the auxiliary binary variables b_t^j. In case UAV j is not at the target location at time t, in order for the constraints to be satisfied, the binary variable b_t^j, multiplied by a large constant M, is forced to 1. On the contrary, as soon as UAV j is at the target location, the constraints are satisfied without the help of M and the binary variable b_t^j may turn to 0. In fact, as the objective of the `Distributed Optimum` is to minimize the summation of the auxiliary binary variables (i.e., a linear function), as soon as UAV j is at the target location, the binary variables actually do turn to 0. That is to say, the objective function (Eq. 8a) represents the overall time spent by UAV j outside of its target zone. Furthermore, b_T^j is equal to 0, ensuring that UAV j has reached its target location within the optimization horizon T. Also, Eq. 8b imposes

that system state predictions respect the discrete-time linear dynamics as in Eq. 4. Furthermore: (i) Eq. 8e can be easily transformed in a set of linear constraints [21] that forces UAV j within the 3D flying area over the planning horizon; (ii) Eq. 8f is a convex quadratic constraint in the control variables of UAV j that limits the module of applied forces to u_{MAX}; (iii) Eq. 8g is a convex quadratic constraint that, in order to achieve feasible control sequences, imposes a maximum difference $\Delta\mathrm{u}$ between consecutive controls; (iv) Eq. 8h is a convex quadratic constraint that limits the module of the expected UAVs' speed to v_{MAX}. Finally, Eq. 8i keeps the probability of collision between UAVs i and j under a given threshold, imposing that the expected distance between their barycenters is *larger* than $d_t^{i,j}$, as defined in Eq. 7. Substituting Eq. 4 into Eq. 7, such quadratic constraint can be easily transformed into the form:

$$\frac{1}{2}\underline{u}^T L \underline{u} + c^T \underline{u} \geq (d_t^{i,j})^2 \tag{9}$$

with $\underline{u}$ being the set of future controls for UAVs i and j, and with L and c constants allowing to express Eq. 7 in matrix form [16]. Nevertheless, given the *larger or equal* sign of the constraint on the UAV barycenters' distance, the optimization has an overall concave feasibility region in the unknown variables u_t^j. Given the presented formulation, the `Distributed Optimum` is a non-convex MIQCP, where the integer variables are binary variables, that cannot be solved optimally in real time.

4.3 Minimizing Flying Time with a Centralized Approach

In this section, the `Centralized Optimum` is presented. In this version of the optimization, the NM is the one selecting the controls for the UAVs with the objective of minimizing the UAVs' aggregate flying time.

Problem `Centralized Optimum` :

$$\min_{\mathrm{u}^j_{[0:T-1]},\forall j} \quad \sum_j \sum_t b_t^j \tag{10a}$$

subject to:

$$\mu_t^j \text{ as in (4)} \qquad \forall j, \forall t \tag{10b}$$

$$|[\boldsymbol{\mu}_k]_t^j - [p_{Gk}]^j| \leq l_G^j/2 + b_t^j M \qquad k \in \{x,y,z\}, \forall j, t \tag{10c}$$

$$b_t^j \in \{0,1\}, b_T^j = 0 \qquad \forall j, \forall t \tag{10d}$$

$$|[\boldsymbol{\mu}_x]_t^j|, |[\boldsymbol{\mu}_y]_t^j|, |[\boldsymbol{\mu}_z]_t^j| \leq l/2 \qquad \forall j, \forall t \tag{10e}$$

$$|\mathrm{u}_t^j| \leq \mathrm{u}_{MAX} \qquad \forall j, \forall t \tag{10f}$$

$$|\mathrm{u}_t^j - \mathrm{u}_{t-1}^j| \leq \Delta\mathrm{u} \qquad \forall j, \forall t \tag{10g}$$

$$|\dot{\boldsymbol{\mu}}_t^j| \leq v_{MAX} \qquad \forall j, \forall t \tag{10h}$$

$$|\boldsymbol{\mu}_t^j - \boldsymbol{\mu}_t^i| \geq d_t^{i,j} \qquad \forall i,j \in \mathcal{M}, \forall t \tag{10i}$$

The structure and the constraints of `Centralized Optimum` are very similar to `Distributed Optimum`. The main difference is that all UAVs' controls are now optimization variables (all UAVs' locations are expressed as a linear function of the controls as in in Eq. 10b). UAVs do not act any more as moving obstacles, but their trajectories are selected jointly to minimize a collaborative objective function, i.e., the aggregate flying time (Eq. 10a). This is obtained through the use of the auxiliary binary variables $b_t^j, \forall j$, i.e., the linear constraints derived by Eqs. 10c-10d, which, thanks to the same reasoning presented for the `Distributed Optimum`, are equal to 1 if UAV j is outside the target area at time t and 0 otherwise. Again, each of the feasible solutions ensures that all UAVs reached their destination within the optimization horizon T (Eq. 10d), ensuring that the $\mathcal{M}$ requests reside in the capacity region of the flying environment. As in the `Distributed Optimum`, bounds are defined on the forces applied to the UAVs and to the UAVs' speed, i.e., Eqs. 10f-10h. Also, UAVs' expected locations respect the boundaries of the 3D flying zone (Eq. 10e) and the minimum safe distance $d_t^{i,j}, \forall i,j \in \mathcal{M}$ (Eq. 10i). This constraint can be transformed, as expressed in Eq. 9, into a quadratic constraint in the control variables of UAVs i and j, $\forall i,j \in \mathcal{M}$ (i.e., in a set of linear coupled constraints). Hence, transforming the target collision probability into a safe minimum distance among UAVs allows to jointly select the trajectory of all UAVs in the system simultaneously, even in presence of UAVs' system state uncertainty. Given the presented formulation, due to the concave nature of the feasibility region, the `Centralized Optimum` is a non-convex MIQCP, where the integer variables are binary variables, that cannot be solved optimally in real time.

5 A Near-Optimal Approximation for the Distributed and Centralized Optimums

In this section, the formulation of the `Centralized` and `Distributed Optimum` is approximated to reduce complexity, while obtaining near-optimal solutions. First, an approach is presented that convexifies the feasibility region

of these optimizations (Section 5.1) and then, this approach is applied to obtain: (i) a real-time solution for UAV flying time minimization, i.e., the `Distributed Dodecahedron` approach (Section 5.2); and (ii) a novel optimization that jointly selects UAV controls to maximize cooperatively, in small scenarios, the airspace capacity, i.e., the `Centralized Dodecahedron` (Section 5.3). Finally, in order to compare the obtained results with the corresponding original formulations, the same approach is used to obtain a valid upper-bound to the `Centralized` and `Distributed Optimum` (Section 5.4).

5.1 A Higher Dimensional Space to Convexify the `Centralized` and `Distributed Optimum` Feasibility Regions

The complexity of the original optimization problems is due to the concave safety constraint (Eqs. 8i, 10i), that ensures a collision probability less than or equal to 2ϵ, for each pair of UAVs $i, j \in \mathcal{M}$, and for each time slot t in the optimization horizon of UAVs i and j. Such constraint, i.e., the 2-norm of the difference between UAVs' barycenters, can be depicted as a sphere of radius $d_t^{i,j}$, centered at one of the two UAVs' barycenters, that the other UAV cannot enter. Depending on the previously presented optimizations, the center of such a sphere is: (i) pre-computed, as in the `Distributed Optimum`, or (ii) unknown at the time of optimization, i.e., a linear combination of the UAV controls (Eq. 4), as in the `Centralized Optimum`.

To help reduce computational complexity, in both cases, the 2-norm constraint can be approximated by a piecewise linear region representing a regular polyhedron circumscribing the sphere. By definition, the new set of linear constraints originating by the circumscribed polyhedron always ensure that the distance between UAVs' barycenters is at least $d_t^{i,j}$. Considering that all the remaining constraints are not modified, this tighter constraint ensures that the obtained feasibility region is completely included in the feasibility region of the optimum formulations, hence leading to a sub-optimal (but easier to achieve) solution.

In the following, a regular *dodecahedron* circumscribing the sphere originating from the constraints in Eqs. 8i and 10i is constructed for each pair of UAVs i and j and for each time t, but any other polyhedron could also be used. Specifically, the inradius of the dodecahedron, i.e., the radius of the inscribed sphere that is tangent to each of the dodecahedron's faces, is equal to $d_t^{i,j}$. A dodecahedron centered at $\boldsymbol{\mu}_t^i$ exhibits 12 pentagonal faces and 20 vertices (for both the case where $\boldsymbol{\mu}_t^i$ is known at the moment of the optimization, as in the `Distributed Optimum`, and the case where $\boldsymbol{\mu}_t^i$ is a linear function of the UAV's i controls, as in the `Centralized Optimum`). These 20 vertices are formed as the union of the vertices of a cube and the vertices of 3 rectangles on the yz, xz, and xy planes, all appropriately scaled and oriented, with Cartesian coordinates given by:

$$\text{Cube}: \quad ([\boldsymbol{\mu}_x]_t^i \pm v_1, [\boldsymbol{\mu}_y]_t^i \pm v_1, [\boldsymbol{\mu}_z]_t^i \pm v_1) \tag{11}$$

$$\text{Rectangle on } yz: \quad ([\boldsymbol{\mu}_x]_t^i, [\boldsymbol{\mu}_y]_t^i \pm v_2, [\boldsymbol{\mu}_z]_t^i \pm v_3) \tag{12}$$

$$\text{Rectangle on } xz: \quad ([\boldsymbol{\mu}_x]_t^i \pm v_3, [\boldsymbol{\mu}_y]_t^i, [\boldsymbol{\mu}_z]_t^i \pm v_2) \tag{13}$$

$$\text{Rectangle on } xy: \quad ([\boldsymbol{\mu}_x]_t^i \pm v_2, [\boldsymbol{\mu}_y]_t^i \pm v_3, [\boldsymbol{\mu}_z]_t^i) \tag{14}$$

where $v_1 = \frac{\sqrt{3-\varphi}}{\varphi} d_t^{i,j}$, $v_2 = (\sqrt{3-\varphi}) d_t^{i,j}$, and $v_3 = \frac{\sqrt{3-\varphi}}{\varphi^2} d_t^{i,j}$, with the constant φ being the golden ratio, equal to $\varphi = \frac{1+\sqrt{5}}{2}$.

An illustrative example of how the circumscribed dodecahedron is built is shown in Fig. 2. In Fig. 2(c), the difference between the original constraint, the sphere, and its approximation, i.e., the dodecahedron, represents also the difference between the original feasibility region and the feasibility region obtained with the introduced approximation. That said, if the estimated location of UAV j is outside the dodecahedron at time t, then it is also outside the original sphere, respecting constraints in Eqs. 8i) and 10i. In order for UAV j to reside outside the dodecahedron, it is sufficient that the estimated location of UAV's j barycenter is outside of at least one of the planes including a face of the dodecahedron. To obtain such a result, a big-M approach, similar to the one that is used in [3] to avoid static polyhedron obstacles, is introduced. First, for each of the half-spaces originating by the faces of the dodecahedron where the UAV can reside, a linear constraint is presented:

$$\begin{aligned}
&a_{11}([\boldsymbol{\mu}_x]_t^j - [\boldsymbol{\mu}_x]_t^i) + a_{12}([\boldsymbol{\mu}_y]_t^j - [\boldsymbol{\mu}_y]_t^i) + a_{13}([\boldsymbol{\mu}_z]_t^j - [\boldsymbol{\mu}_z]_t^i) \le c_1 \\
&a_{21}([\boldsymbol{\mu}_x]_t^j - [\boldsymbol{\mu}_x]_t^i) + a_{22}([\boldsymbol{\mu}_y]_t^j - [\boldsymbol{\mu}_y]_t^i) + a_{23}([\boldsymbol{\mu}_z]_t^j - [\boldsymbol{\mu}_z]_t^i) \le c_2 \\
&\qquad\qquad \vdots \\
&a_{n1}([\boldsymbol{\mu}_x]_t^j - [\boldsymbol{\mu}_x]_t^i) + a_{n2}([\boldsymbol{\mu}_y]_t^j - [\boldsymbol{\mu}_y]_t^i) + a_{n3}([\boldsymbol{\mu}_z]_t^j - [\boldsymbol{\mu}_z]_t^i) \le c_n
\end{aligned} \tag{15}$$

where $\boldsymbol{\alpha_k} = (a_{k1}, a_{k2}, a_{k3})$ is the outward normal vector to the k^{th} face of the dodecahedron, c_k is the dot product of the outward normal with a known point on the k^{th} face, and $n = 12$ is the total number of faces of the dodecahedron. Equation 15 can also be written more compactly as $A_t^{i,j}(\boldsymbol{\mu}_t^j - \boldsymbol{\mu}_t^i) \le C_t^{i,j}$, where $A_t^{i,j}$ is a $n \times 3$ matrix, and $C_t^{i,j}$ is a $n \times 1$ vector.

Then, a big-M method ensures that at least one of the introduced constraints is respected. Specifically, let $\boldsymbol{b}_t^{i,j}$ be a binary vector where each of the elements, i.e., $b_t^{i,j}(p)$ with $p \in \{1, ..., 12\}$, is summed at the right-hand side of one of the linear constraints in Eq. 15 after being multiplied by a large constant M, i.e.,:

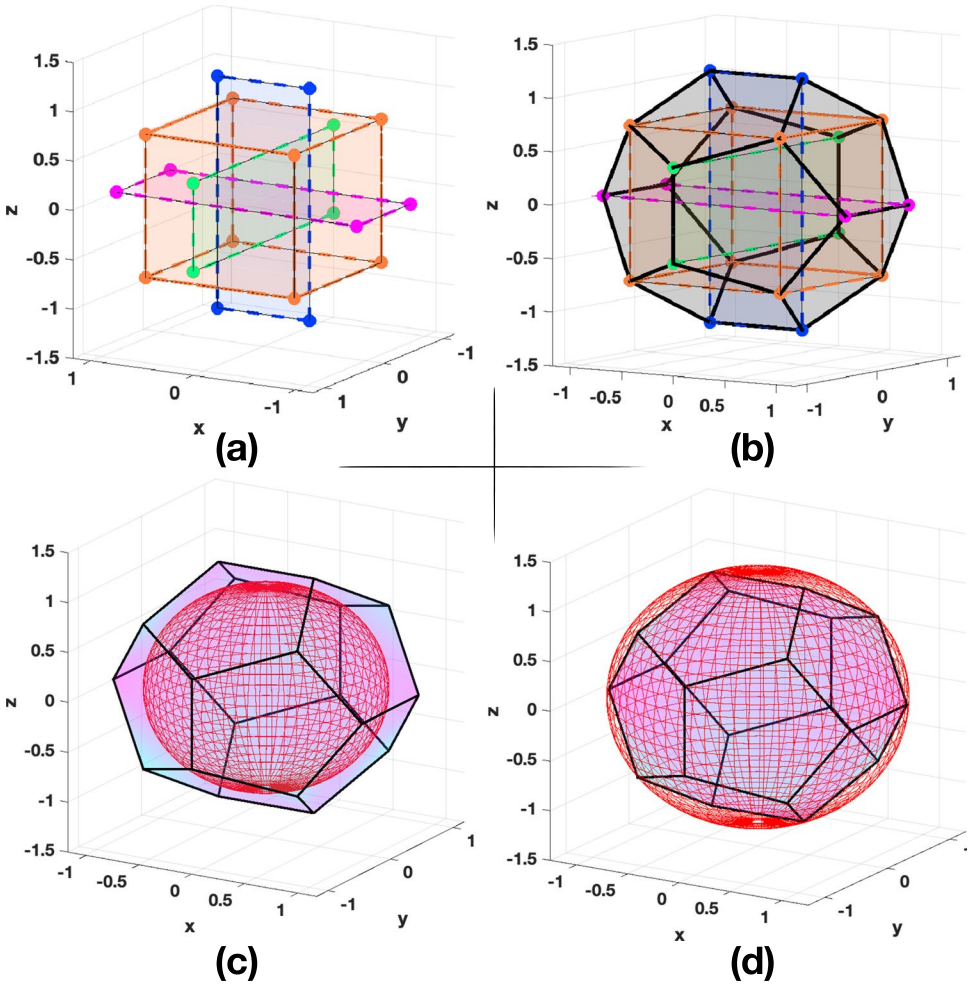


Fig. 2 (**a**)-(**b**) Dodecahedron at the origin constructed by connecting the vertices of a cube (orange) with the vertices of 3 rectangles in the yz (green), xz (blue) and xy (pink) planes, all properly scaled and oriented, (**c**)-(**d**), respectively, circumscribed and inscribed dodecahedron about the sphere with radius $d_t^{i,j}$

$$A_t^{i,j}(\boldsymbol{\mu}_t^j - \boldsymbol{\mu}_t^i) \leq C_t^{i,j} + \boldsymbol{b}_t^{i,j} M. \tag{16}$$

Subsequently, thanks to the introduced big-M method, in case $b_t^{i,j}(p) = 1$, the corresponding linear constraint is not active, since it is always respected. When $b_t^{i,j}(p) = 0$, instead, the constraint is active and the UAV is forced in one of the half-spaces outside of the dodecahedron. Hence, if for each pair of UAVs i, j and for each t, the summation of the binary variables in $\boldsymbol{b}_t^{i,j}$ is less than the number of the faces of the polyhedron used to approximate the 2-norm constraint of the original formulation, the obtained trajectories reside outside the dodecahedron and the original safety constraint is respected.

5.2 The `Distributed Dodecahedron` Optimization

The presented approximation of the safety constraint is initially applied explicitly to the `Distributed Optimum`, resulting in the `Distributed Dodecahedron`.

Problem `Distributed Dodecahedron` :

$$\min_{\mathrm{u}^j_{[0:T-1]}} \quad \sum_t b_t^j \tag{17a}$$

subject to:

$$\boldsymbol{\mu}_t^j \text{ as in Eqn. (4)} \qquad \forall t \tag{17b}$$

$$|[\boldsymbol{\mu}_k]_t^j - [p_{Gk}]^j| \leq l_G^j/2 + b_t^j M \qquad k \in \{x, y, z\}, \forall t \tag{17c}$$

$$b_t^j \in \{0, 1\}, b_T^j = 0 \qquad \forall t \tag{17d}$$

$$|[\boldsymbol{\mu}_x]_t^j|, |[\boldsymbol{\mu}_y]_t^j|, |[\boldsymbol{\mu}_z]_t^j| \leq l/2 \qquad \forall t \tag{17e}$$

$$|\mathrm{u}_t^j| \leq \mathrm{u}_{MAX}; |\mathrm{u}_t^j - \mathrm{u}_{t-1}^j| \leq \Delta \mathrm{u} \qquad \forall t \tag{17f}$$

$$|\dot{\boldsymbol{\mu}}_t^j| \leq v_{MAX} \qquad \forall t \tag{17g}$$

$$A_t^{i,j}(\boldsymbol{\mu}_t^j - \boldsymbol{\mu}_t^i) \leq C_t^{i,j} + \boldsymbol{b}_t^{i,j} M \qquad \forall i < j, \forall t \quad (17h)$$

$$b_t^{i,j}(p) \in \{0,1\} \qquad b_t^{i,j}(p) \in \boldsymbol{b}_t^{i,j}, \forall i < j, \forall t \quad (17i)$$

$$\sum_{p=1}^{12} b_t^{i,j}(p) < 12 \qquad \forall i < j, \forall t \quad (17j)$$

In the `Distributed Dodecahedron`, Eqs. 17b-17g are the same as Eqs. 8b-8h in the `Distributed Optimum`, representing (i) the flying time computation via binary auxiliary variables, (ii) the linear relationship between controls and UAV system state; (iii) limits on controls, and (iv) limits on UAV j's speed. Eqs. 17h-17j represent the set of linear constraints introduced by the approximation described in Section 5.1 to substitute the quadratic constraint in `Distributed Optimum`, i.e., Eq. 8i. Specifically, Eq. 17h represents the linear constraints originated by the 12 faces of the dodecahedron used to approximate the 2-norm between UAVs j and i, with $i < j$. Note that at the time of the optimization, UAV j's location is unknown, but it can be expressed as a linear function of the controls (Eq. 17b). Given the presented formulation, the `Distributed Dodecahedron` is a convex MIQCP, where the integer variables are binary variables, with a number of unknowns (i.e., the control variables $\mathrm{u}^j_{[0:T-1]}$ and the indirectly tunable binary variables $\boldsymbol{b}_t^{i,j}$) and constraints that grow linearly with the planning window T and with the number of UAVs in the flying environment.

The key aspect of the proposed approach is a reduced complexity compared to `Distributed Optimum`, at the expense of a sub-optimal result, since the feasibility region is only approximated. The approach is modular, and performance can be traded with computational complexity, if required. For example, the number of planes used to approximate the safety quadratic constraint can be reduced, while the objective function computation could be easily parallelized for fairly short optimization horizon T. Indeed, it should be noted that the possible combinations of values assumed by the binary variables in the objective functions are very limited, i.e., 1 when out of the target area, 0 in the target area. Therefore, a set of optimizations, one for each possible flying time for UAV j, could be run in parallel, leading to an achievable real-time solution.

5.3 The `Centralized Dodecahedron` Optimization

Finally, the presented approximation is applied to the `Centralized Optimum` problem and the following simplified `Centralized Dodecahedron` optimization is obtained.

Problem `Centralized Dodecahedron` :

$$\min_{\mathrm{u}^j_{[0:T-1]}, \forall j} \sum_j \sum_t b_t^j \quad (18a)$$

subject to:

$$\boldsymbol{\mu}_t^j \text{ as in Eq. (4)} \qquad \forall j, \forall t \quad (18b)$$

$$|[\boldsymbol{\mu}_k]_t^j - [p_{Gk}]^j| \leq l_G^j/2 + b_t^j M \qquad k \in \{x,y,z\}, \forall j, t \quad (18c)$$

$$b_t^j \in \{0,1\}, b_T^j = 0 \qquad \forall j, \forall t \quad (18d)$$

$$|[\boldsymbol{\mu}_x]_t^j|, |[\boldsymbol{\mu}_y]_t^j|, |[\boldsymbol{\mu}_z]_t^j| \leq l/2 \qquad \forall j, \forall t \quad (18e)$$

$$|\mathrm{u}_t^j| \leq \mathrm{u}_{MAX} \qquad \forall j, \forall t \quad (18f)$$

$$|\mathrm{u}_t^j - \mathrm{u}_{t-1}^j| \leq \Delta \mathrm{u} \qquad \forall j, \forall t \quad (18g)$$

$$|\dot{\boldsymbol{\mu}}_t^j| \leq v_{MAX} \qquad \forall j, \forall t \quad (18h)$$

$$A_t^{i,j}(\boldsymbol{\mu}_t^j - \boldsymbol{\mu}_t^i) \leq C_t^{i,j} + \boldsymbol{b}_t^{i,j} M \qquad \forall i,j \in \mathcal{M}, \forall t \quad (18i)$$

$$b_t^{i,j}(p) \in \{0,1\} \qquad b_t^{i,j}(p) \in \boldsymbol{b}_t^{i,j}, \forall i,j \in \mathcal{M}, \forall t \quad (18j)$$

$$\sum_{p=1}^{12} b_t^{i,j}(p) < 12 \qquad \forall i,j \in \mathcal{M}, \forall t \quad (18k)$$

The `Centralized Dodecahedron` problem presents the same flying time computation, limits on controls, UAV's speed, and relationship between controls and UAV system state that are present in the `Centralized Optimum` problem (see Eqs. 18b-18h and Eqs. 10b-10h, respectively). The main difference with the `Distributed Dodecahedron` approximation is that the positions of all UAVs are unknown at the moment of the optimization in `Centralized Dodecahedron` (even though they can be expressed as a linear function of the controls). Hence, the set of linear constraints originated by the approximation of the 2-norm with the dodecahedron, i.e., Eqs. 18i-18k as explained in Section 5.1, substitutes the safety constraint in Eq. 10i for each pair of UAVs.

Given the presented formulation, the `Centralized Dodecahedron` is a convex MIQCP, where the integer variables are binary variables. Contrary to

the Distributed Dodecahedron, the number of unknowns and constraints grows linearly with the planning window T and combinatorially with the number of UAVs in the flying environment, ultimately limiting the applicability of the approach to fairly small scenarios. Nevertheless, the improvement in airspace capacity shown by the Centralized Dodecahedron is large, as it is shown below, motivating the introduction of the proposed solution. Possible scalability improvements, which will be the focus of future research efforts, could be achieved by splitting systematically large flying environments, as suggested in [25], or by applying iterative consensus-based approaches.

5.4 Upper Bound on the Distributed/Centralized Optimum

In order to compare the results obtained approximating the 2-norm between the UAVs' estimated locations as a circumscribed polyhedron in reasonably large instances, an upper bound on the performance of the Centralized and Distributed Optimum is obtained. The approach follows the same philosophy as in Section 5.1 and is denoted as Centralized and Distributed Bounds. To obtain such an upper bound on the Optimum formulations, the 2-norm between the UAVs' estimated locations, i.e., the sphere representing the minimum distance to respect between any two UAVs, is now approximated as an *inscribed* regular polyhedron. Considering, as in Section 5.1, that all the remaining constraints are not modified, this leads to a looser set of constraints, where the feasibility region of the Bounds formulations is completely included in the feasibility region of the Centralized and Distributed Optimums. An example of such an approach is reported in Fig. 2(d). In this case, the inscribed dodecahedron (i.e., a dodecahedron touched by the sphere at all vertices) is built starting from the union of the vertices of a cube and the vertices of 3 rectangles on the yz, xz, and xy planes as in Eq. 11, where $v_1 = \frac{1}{\sqrt{3}} d_t^{i,j}$, $v_2 = \frac{\varphi}{\sqrt{3}} d_t^{i,j}$, and $v_3 = \frac{1}{\sqrt{3}\varphi} d_t^{i,j}$. Substituting the set of linear-binary constraints originated by the approximation of the 2-norm with the dodecahedron approximation mentioned above, the formulations of the Centralized and Distributed Bound problems could be easily obtained as in Sections 5.2 and 5.3.

6 Performance Evaluation

This section evaluates the performance of the Centralized and Distributed Optimum approximations. First, the reference scenario is illustrated (Section 6.1). Then, Section 6.2 presents a detailed explanation, in a toy scenario, of how choosing cooperatively the controls of the UAVs changes the chosen trajectories, compared to a distributed approach. Next, in Section 6.3, the performance of the Centralized and Distributed Bound is compared with the performance of the approaches proposed in this work.

6.1 Simulation Set-Up

To showcase the performance of the proposed approaches, the solutions have been obtained using the GUROBI solver [9] and a computing platform with 8 GB RAM and 2.6 GHz CPU frequency. Furthermore, the input set-up used in the simulations, if not differently stated, is as follows. A cube of side equal to 30m is used (i.e., $l = 30$m) as the 3D flying environment. The size of the chosen flying environment is small, as suggested by [25]. Such choice keeps the number of simultaneously admitted UAVs fairly limited, allowing to compare the performance achieved by Distributed and Centralized Dodecahedron. The sampling time is equal to 1s and $T = 20$ (hence the optimization horizon is 20s). Starting and target locations are selected at random from the same pool of possible locations, i.e., the corners and the center of each face of the cube representing the flying zone, forcing UAVs to cross, in general, the entire flying area. Any other scenario could also be adopted, including also scenarios where UAVs enter the flying environments at different times. Nevertheless, the chosen scenario increases the probability of having possibly colliding trajectories, hence better showcasing the capabilities of the proposed approaches. Specifically, taking as reference the side of the flying environment located at the positive half-space of the abscissa, the starting/target locations are: $[\frac{l}{2}, \frac{l}{2}, \frac{l}{2}]$, $[\frac{l}{2}, \frac{l}{2}, 0]$, $[\frac{l}{2}, \frac{l}{2}, -\frac{l}{2}]$, $[\frac{l}{2}, 0, \frac{l}{2}]$, $[\frac{l}{2}, 0, 0]$, $[\frac{l}{2}, 0, -\frac{l}{2}]$, $[\frac{l}{2}, -\frac{l}{2}, \frac{l}{2}]$, $[\frac{l}{2}, -\frac{l}{2}, 0]$, $[\frac{l}{2}, -\frac{l}{2}, -\frac{l}{2}]$. Similarly, starting/target locations are individuated in all six faces of the flying environment. Further, while the starting location corresponds to the aforementioned locations, the selected 3D target area is cubic, with a side equal to $l_G^j = 2$m. Also, UAVs select, with uniform distribution, the module of their initial speed in the interval [0, 5]m/s, with their directions towards the center of the flying zone. The initial UAVs' system states, constituted by the initial location and speed on the 3D Cartesian coordinates as described above, are known with no uncertainty. Limits on speed and controls are $\Delta u = 1$N, $u_{max} = 10$N, and $v_{max} = 14$m/s (~ 50km/h), while the probability of having the barycenter of the UAV within the ellipsoid described in Section 4.1 is $1 - \epsilon = 0.99999$. Furthermore, the motion model is characterized by: (i) the air resistance coefficient $\mu = 0.8$, and (ii) the UAV mass $m^j = 3$kg. Finally, the uncertainty of the system state prediction due to the perturbing acceleration noise is $\Sigma_\alpha^j = [0.1, 0.1, 0.1]$m/s^2.

6.2 A Toy Example

To better understand how the simultaneous selection of the controls of UAVs changes their trajectories, an illustrative example is presented, for a flying environment with now $l = 60$m. The starting points of four UAVs are $[\pm\frac{l}{2}, \pm\frac{l}{2}, 0]$, i.e., the center of the faces perpendicular to the $x-y$ plane of the 3D flying environment. Each UAV traverses the flying environment and arrives at the center of the face opposite to its starting location (which is also the starting location of another UAV).

In order to select the controls within T, the `Centralized` and the `Distributed Dodecahedron` enforce, at any time t, at least $d_{min} = 2$m between the ellipsoids containing the barycenters of the UAVs. In the `Centralized Dodecahedron` controls are selected simultaneously for all UAVs and in the `Distributed Dodecahedron`, UAVs are ordered randomly and controls are selected sequentially.

Figure 3 summarizes the trajectories obtained for the UAVs (dots illustrate the UAVs' expected locations at each time slot and dot size increases as time passes). Initially, Figs. 3(a) and (d) show the trajectories of 2 UAVs for the `Distributed` and `Centralized Dodecahedron`, respectively. It is shown that with the `Distributed Dodecahedron`, UAV 2 simply avoids UAV 1, which has previously selected the fastest route to the destination while with the `Centralized Dodecahedron`, the two UAVs modify (symmetrically) their trajectories so as to have a small deviation from the fastest route to the destination, effectively reducing the aggregate flying time by 1s.

Similarly, Figs. 3(b) and (e) show the trajectories of 3 UAVs. In both the `Distributed` and `Centralized Dodecahedron` UAV 3 avoids UAVs 1 and 2 by passing below their expected trajectories. However, in the `Centralized Dodecahedron` UAVs 1 and 2 perform small arcs (sufficiently separated), allowing UAV 3 to reduce its flying time, thus reducing the aggregate flying time by 2s compared to `Distributed Dodecahedron`.

Finally, Figs. 3(c) and (f) show the trajectories of 4 UAVs. In `Distributed Dodecahedron`, UAV 4 cannot fly over or below the rest of the UAVs (as other UAVs already chose those trajectories), thus it moves towards the center of the 3D environment, waiting for UAV 1 to pass, before crossing towards its destination area. In `Centralized Dodecahedron`, instead, UAVs cooperatively adjust their trajectories, reducing the aggregate flying time by 3s as compared to `Distributed Dodecahedron` (over a total of 62s aggregate flying time).

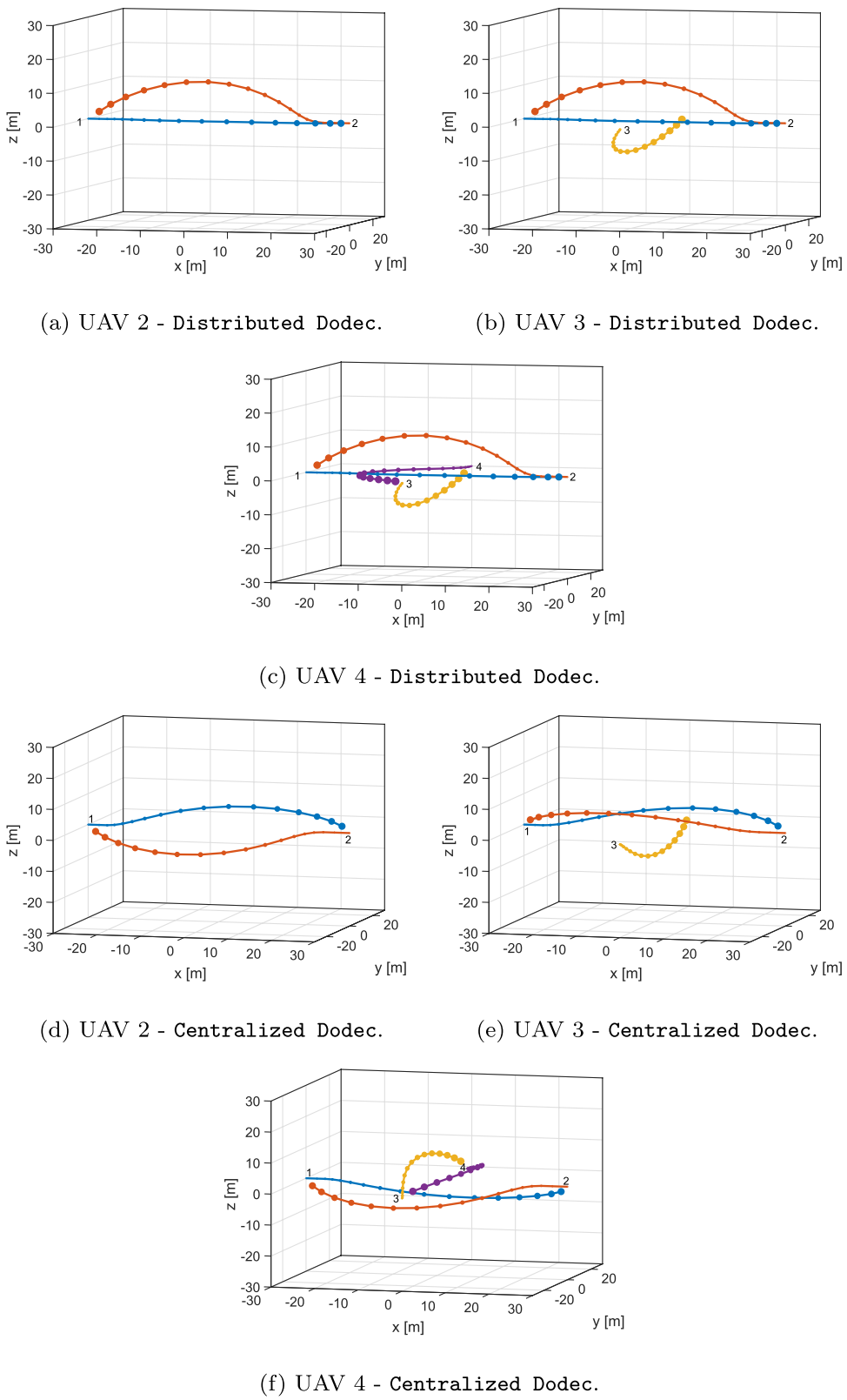
It is noted that the chosen trajectories respect Eqs. 18i-18k, ensuring that the pair-wise collision probability is less than 2ϵ. To showcase such result, the `Centralized Dodecahedron` with 4 UAVs is used (Fig. 3(f). Comparing at each time slot the trajectories obtained from the `Centralized Dodecahedron` optimization, Table 1 reports the expected minimum distance between each pair of UAVs, showing that each UAV passes at $\sim$ 14m from two of the three other UAVs (with the pairs of UAVs 1-3 and 2-4 passing even further away from each other, at $\sim$ 18m).

The control profiles for the UAVs in the `Centralized Dodecahedron` optimization are then applied (in the presence of the perturbing acceleration noise) to evaluate the actual trajectories of the UAVs. Considering a large set (i.e., 100k) of simulations, Fig. 4 presents the empirical PDF of the minimum distance between each pair of simulated UAVs. As expected, the results show the presence of two main peaks at around 14 and 18 meters, i.e., the minimum distance among UAVs when the expected trajectories are considered. Interestingly, the minimum distance between UAVs in all simulations is 7.04m, i.e., larger than d_{min} in all cases. These results show that the computed empirical approximation of the collision probability (equal to 0 in our simulations) respects its theoretical bound, i.e., 2ϵ as computed in Eq. 5 for the proposed heuristics.

6.3 Performance Results of the Heuristics

This section presents the flying environment's capacity achieved by the different approaches, with the simulation parameters of Section 6.1. Two different metrics are used to assess the performance of the introduced approaches: (i) the empirical cumulative density function (CDF) of the number of UAVs admitted before obtaining an infeasible solution, when the UAVs are introduced to the 3D flying environment sequentially; (ii) the average and the $20-80^{th}$ percentile of the time needed by the UAVs to reach their destinations, depending on the number of UAVs in the 3D flying environment. For a fair comparison amongst the approaches, the same initial conditions and the same starting/target locations are used.

Table 2 presents the number of UAVs admitted to the 3D environment, before obtaining an infeasible solution from the solver. Up to 8 UAVs request access to the flying area from the NM, sequentially, in 100 simulations with different starting/target locations. As demonstrated in Table 2, the `Distributed Dodecahedron` struggles to achieve a large number of admitted UAVs (hence the limit of 8 UAVs used), since the control decisions are not taken cooperatively, but rather independently for each UAV. Selecting cooperatively the controls of the UAVs improves performance, as it is clear from the results achieved by `Centralized Dodecahedron` and by `Centralized Bound`. Indeed, `Centralized Dodecahedron` admitted on average 7.79 UAVs, i.e., 31.81% more compared to `Distributed Dodecahedron` (which admitted 5.91 UAVs



(a) UAV 2 - `Distributed Dodec.` (b) UAV 3 - `Distributed Dodec.`

(c) UAV 4 - `Distributed Dodec.`

(d) UAV 2 - `Centralized Dodec.` (e) UAV 3 - `Centralized Dodec.`

(f) UAV 4 - `Centralized Dodec.`

Fig. 3 Trajectories obtained with `Distributed` and `Centralized Dodecahedron` for a toy example with 2,3, and 4 UAVs

on average). Furthermore, `Centralized Dodecahedron` and `Centralized Bound` perform similarly for the scenario under consideration.

Figure 5 presents the average and the 20-80th percentile of the UAVs flying time distribution. As expected, increasing

Table 1 Minimum inter-UAV distance - `Centralized Dodecahedron`

UAV Pair	Min. Dist. (m)	UAV Pair	Min. Dist. (m)
1-2	13.1	2-3	14.73
1-3	18.4	2-4	18.65
1-4	13.96	3-4	15.05

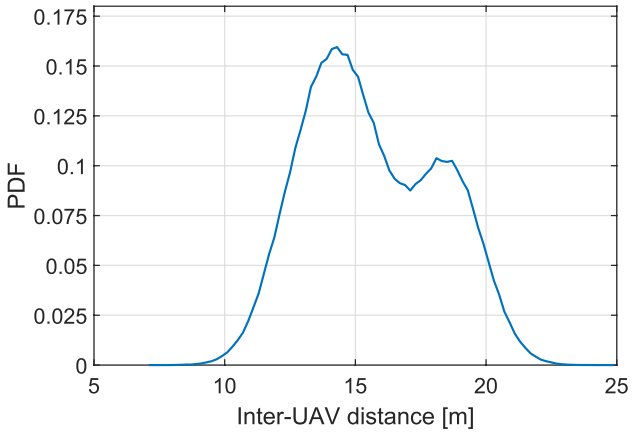


Fig. 4 Empirical PDF of the minimum inter-UAV distance (toy example)

the number of UAVs in the system increases the contention in the flying environment and consequently the average flying time of the UAVs. Nevertheless, such increase is less evident with `Centralized Dodecahedron`, since the UAV trajectories adapt cooperatively to accommodate the new incoming UAVs. Such behavior is also evident by looking at the 80^{th} percentile of the distribution. While the 80^{th} percentile of the `Distributed Dodecahedron`'s flying time increases considerably, thus increasing the number of UAVs in the environment (the last incoming UAVs need to avoid several moving obstacles, hence increasing their flying times), the 80^{th} percentile of the `Centralized Dodecahedron` approach is unaffected, as UAVs find cooperatively new trajectories for accommodating all requests without increasing, when possible, their flying times. Again, for this metric, the difference between `Centralized Dodecahedron` and `Centralized Bound` is small, demonstrating that the presented approach is nearly optimal.

Finally, in order to show how the `Distributed Dodecahedron` handles scenarios with a higher number of UAVs, Fig. 6 presents the average and the 20-80^{th} percentile of the flying time distribution in two larger flying environments. Specifically, the two selected flying environments have sides $l = 70$ m and $l = 100$ m, respectively, with starting and target positions also at the center of each cube

Table 2 CDF of the number of UAVs admitted in the 3D flying environment before obtaining an infeasible solution

Num. UAV	1	2	3	4	5	6	7	8
`Distributed Dodec.`	0.04	0.1	0.13	0.19	0.28	0.52	0.83	1
`Centralized Dodec.`	0	0	0	0	0	0.05	0.16	1
`Centralized Bound`	0	0	0	0	0	0	0.03	1

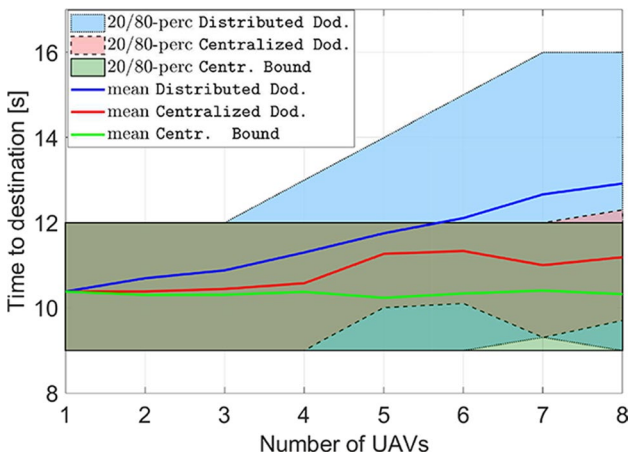


Fig. 5 Average and 20-80^{th} percentile of the time required by UAVs to reach their intended destinations

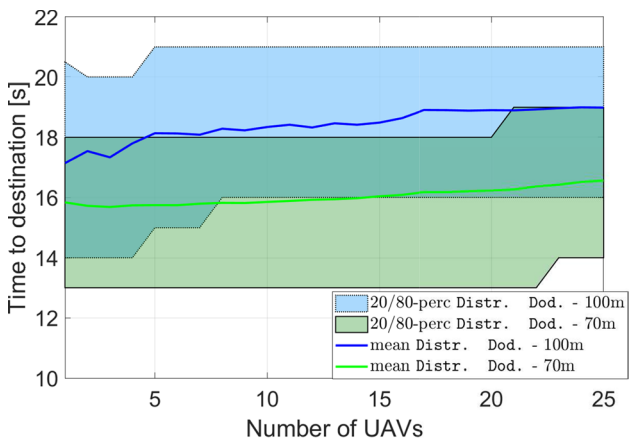


Fig. 6 Larger simulation set-up. Average and 20-80^{th} percentile of the time required by UAVs to reach their intended destinations with `Distributed Dodecahedron` Optimization

side (25 total maximum flying UAVs). Additionally, to cope with larger distances to travel, the optimization window is now equal to $N = 25$. Because of the larger flying environment (hence a larger capacity also for the `Distributed Dodecahedron`), in all 100 simulations all UAVs (all 25 of them) found a possible trajectory to their target locations. To further demonstrate that the flying environment can accommodate all the UAVs (due to its larger size), the average and the 20-80th percentiles do not increase significantly while increasing the number of admitted UAVs, contrarily to what happens in the smaller simulation set-up (Fig. 5). As expected, between the two new simulation set-ups, the average flying time is larger when a larger flying environment is used, i.e., with $l = 100$ m.

7 Conclusions

This work presented a novel framework for UAV 4D trajectory planning for dynamic ATM that takes into consideration the motion prediction uncertainty. With linear-Gaussian motion models, an ellipsoid around each UAV's barycenter characterizes the area in the flying environment that contains (in the present and in the future), with a fixed probability, the UAV. Then, if the selected trajectories are such that ellipsoids never intersect, a bound on the collision probability among UAVs can be guaranteed. Initially, a concave optimization able to plan the trajectory for each UAV in the system, i.e., the `Optimum` formulations, is presented. In order for the ellipsoids to not intersect, the proposed optimizations ensure a minimum dynamic distance (which depends on the uncertainty) between the barycenters of each pair of UAVs in the system.

Nevertheless, due to the concave nature of its feasibility region, the `Optimum` formulations' complexity is such that no real-time solution is possible. To obtain a practical solution to the 4D trajectory planning problem, two heuristics are then proposed. In both cases, the quadratic constraint representing the minimum distance to respect between each pair of UAVs is approximated by a set of binary-linear constraints that helps to improve the tractability of the optimizations. First, a heuristic that can be used as a valid real-time performance benchmark is presented, i.e., the `Distributed Dodecahedron`, where UAVs' controls are selected sequentially. Then, to improve performance, at least on small scenarios, a more complex heuristic that coordinates simultaneously all UAVs in the flying environment, i.e., the `Centralized Dodecahedron`, is also showcased. In the performed simulations, both the `Distributed` and the `Centralized Dodecahedron` demonstrated their utility. Indeed, the `Distributed Dodecahedron` presents high scalability capabilities, and is able to plan, in the presence of location uncertainty, trajectories for UAVs in dense scenarios. Choosing simultaneously the controls of UAVs, as in the `Centralized Dodecahedron`, instead, increases the capacity of small flying environments, presenting nearly-optimal performance at the price of additional complexity.

Future research includes the consideration of more complex non-linear UAV motion models and novel approximations to improve the scalability of the `Centralized Dodecahedron` approach using techniques for partitioning the flying environment or consensus-based approaches.

Author Contributions All authors contributed to the study conception and design in equal manner. Material preparation, analysis and implementation was performed by Christian Vitale and Savvas Papaioannou.

Funding This work was supported by the European Union's Horizon 2020 Research and Innovation Programme under Grant 739551 (KIOS CoE - TEAMING) and Grant 101003439 (C-AVOID), and by the Republic of Cyprus through the Deputy Ministry of Research, Innovation and Digital Policy.

Data Availability Not applicable

Code Availability Not applicable

Declarations

Conflicts of Interest No Conflicts of interest

Ethics Approval Not applicable

Consent to Participate Not applicable

Consent for Publication Not applicable

References

1. Alrifaee, B., Kostyszyn, K., Abel, D.: Model predictive control for collision avoidance of networked vehicles using Lagrangian relaxation. IFAC-PapersOnLine **49**(3), 430–435 (2016)
2. Bertram, J., Wei, P.: Distributed computational guidance for high-density urban air mobility with cooperative and non-cooperative collision avoidance. In: AIAA Scitech 2020 Forum (2020)
3. Blackmore, L., Ono, M., Williams, B.: Chance-constrained optimal path planning with obstacles. IEEE Transactions on Robotics **27**(6), 1080–1094 (2011)
4. Borrelli, F., Subramanian, D., Raghunathan, A., Biegler, L.: MILP and NLP techniques for centralized trajectory planning of multiple unmanned air vehicles. In: Proc. American Control Conference (ACC), (2006) https://doi.org/10.1109/ACC.2006.1657644
5. Chao, Z., Zhou, S.L., Ming, L., Zhang, W.G.: UAV formation flight based on nonlinear model predictive control. Mathematical Problems in Engineering (2012)
6. Christofides, P.D., Scattolini, R., de la Pena, D.M., Liu, J.: Distributed model predictive control: A tutorial review and future

research directions. Computers & Chemical Engineering **51**, 21–41 (2013)
7. Everett, M., Chen, Y., How, J.: Motion planning among dynamic, decision-making agents with deep reinforcement learning. In: Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2018)
8. Grinstead, C.M., Snell, J.L.: Introduction to probability. American Mathematical Soc. (1997)
9. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual. https://www.gurobi.com (2022)
10. Hwang, I., Tomlin, C.: Protocol-based conflict resolution for finite information horizon. In: Proc. American Control Conference (ACC), https://doi.org/10.1109/ACC.2002.1024903 (2002)
11. Lennart, L.: System identification: Theory for the user. Prentice Hall (1999)
12. Luis, C.E., Vukosavljev, M., Schoellig, A.P.: Online trajectory generation with distributed model predictive control for multi-robot motion planning. IEEE Robotics and Automation Letters **5**(2), 604–611 (2020). https://doi.org/10.1109/LRA.2020.2964159
13. Matsuno, Y., Tsuchiya, T.: Stochastic 4D trajectory optimization for aircraft conflict resolution. In: Proc. IEEE Aerospace Conference, (2014) https://doi.org/10.1109/AERO.2014.6836275
14. Ong, H.Y., Kochenderfer, M.J.: Markov decision process-based distributed conflict resolution for drone air traffic management. Journal of Guidance, Control, and Dynamics **40**(1), 69–80 (2017)
15. Pallottino, L., Scordio, V., Bicchi, A., Frazzoli, E.: Decentralized cooperative policy for conflict resolution in multivehicle systems. IEEE Transactions on Robotics **23**(6), 1170–1183 (2007). https://doi.org/10.1109/TRO.2007.909810
16. Prajna, S., Papachristodoulou, A., Parrilo, P.: Introducing SOSTOOLS: A general purpose sum of squares programming solver. In: Proc. IEEE Conference on Decision and Control (CDC) (2002)
17. Rădulescu, R., Mannion, P., Roijers, D.M., Nowe, A.: Multi-objective multi-agent decision making: a utility-based analysis and survey. Autonomous Agents and Multi-Agent Systems **34**(1), 1–52 (2020)
18. Ribeiro, M.I .: (Portugal) Gaussian probability density functions: Properties and error characterization. In: Institute for Systems and Robotics (ISR/IST) (2004)
19. Richards, A., How, J.P.: Robust distributed model predictive control. International Journal of control **80**(9), 1517–1531 (2007)
20. SESAR Joint Undertaking. Supporting safe and secure drone operations in Europe. In: SESAR U-space Research and Innovation Results (2020)
21. Shanno, D.F., Weil, R.L.: "linear'' programming with absolute-value functionals. Operations Research **19**(1), 120–124 (1971)
22. Stanley, A.: Flight path deconfliction of autonomous UAVs. In: Infotech@Aerospace (2005)
23. Tomlin, C., Pappas, G.J., Sastry, S.: Conflict resolution for air traffic management: A study in multiagent hybrid systems. IEEE Transactions on Automatic Control **43**(4), 509–521 (1998). https://doi.org/10.1109/9.664154
24. Trodden, P.A., Richards, A.G.: Cooperative tube-based distributed mpc for linear uncertain systems coupled via constraints. In: Distributed Model Predictive Control Made Easy, Springer, pp 57–72 (2014)
25. Yang, X., Wei, P.: Scalable multi-agent computational guidance with separation assurance for autonomous urban air mobility. Journal of Guidance, Control, and Dynamics **43**, 1473–1486 (2020)
26. Zhu, H., Alonso-Mora, J.: Chance-constrained collision avoidance for MAVs in dynamic environments. IEEE Robotics and Automation Letters **4**(2), 776–783 (2019). https://doi.org/10.1109/LRA.2019.2893494

Christian Vitale holds a B.S. and a M.Sc., from the University of Pisa and a Ph.D. in Telecommunication Engineering from Universidad Carlos III de Madrid. He is a Research Associate at the KIOS Research and Innovation Center of Excellence at the University of Cyprus, where he received an Individual Widening Marie-Curie Fellowship. His interests mainly focus on analytical modeling of complex systems, e.g., wireless networks and intelligent transportation systems, design of mechanisms improving network efficiency, and algorithms for guaranteeing QoS to vertical applications in cellular environments.

Savvas Papaioannou received the B.S. degree in Electronic and Computer Engineering from the Technical University of Crete, Chania, Greece in 2011, the M.S. degree in Electrical Engineering from Yale University, New Haven, CT, USA, in 2013 and the Ph.D. degree in Computer Science from the University of Oxford, Oxford, U.K. in 2017. He is currently a Research Associate with the KIOS Research and Innovation Center of Excellence, University of Cyprus, Nicosia, Cyprus. His research interests include multi-agent and autonomous systems, state estimation and control, multi-target tracking, probabilistic inference, Bayesian reasoning, and intelligent unmanned-aircraft vehicle (UAV) systems and applications. Dr. Papaioannou is reviewer for various journals within the IEEE and ACM associations.

Panayiotis Kolios received the B.Eng. (2008) and Ph.D. (2011) degrees in Telecommunications Engineering from King's College London. He is a Research Assistant Professor with the KIOS Research and Innovation Center of Excellence, University of Cyprus. His interests focus on both basic and applied research on networked intelligent systems. Examples of such systems include intelligent transportation systems, autonomous unmanned aerial systems, and the plethora of cyber-physical systems that arise within IoT.

Georgios Ellinas holds B.Sc., M.Sc., M.Phil., and Ph.D. degrees in Electrical Engineering from Columbia University. He is a Professor at the Department of Electrical and Computer Engineering and a founding member of the KIOS Research and Innovation Center of Excellence at the University of Cyprus. Prior to joining the University of Cyprus, he also served as an Associate Professor of Electrical Engineering at City College of the City University of New York, as a Senior Network Architect at Tellium Inc., and as a Research Scientist/Senior Research Scientist in Telcordia Technologies' (formerly Bell Communications Research (Bellcore)) Optical Networking Research Group. Prof. Ellinas is a Fellow of the IET (2019), and a Senior Member of IEEE, OSA, and ACM. He has co-authored/co-edited four books on optical networks, more than 300 archived articles/conference papers/book chapters, and he is the holder of 30 patents on optical networking. His research interests are in the areas of optical networks, intelligent transportation systems, IoT, emergency response systems, and unmanned aerial systems.

Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”).
Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.
These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.
We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.
While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.
These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.
To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.
Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.
If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com