Safe Coordinated Maneuvering of Teams of Multirotor UAVs

A cooperative control framework for multi-vehicle time-critical missions

Venanzio Cichella, Ronald Choe, S. Bilal Mehdi, Enric Xargay,
Naira Hovakimyan, Vladimir Dobrokhodov, Isaac Kaminer,
António M. Pascoal, and A. Pedro Aguiar — January 6, 2016

In the last two decades, significant improvements in materials, computing power of microcontrollers, miniaturization of sensors, versatility of communication, and high-density energy storage have revolutionized the design and capabilities of Unmanned Aerial Vehicles (UAVs). These multiple fundamental advances have broadened the range of missions involving a single UAV, and have truly enabled the execution of cooperative multi-UAV operations.

Besides advancing multiple aerodynamic designs of traditional fixed-wing and helicopter aircraft, a new class of vehicle appeared — the multirotor UAV. Multirotor UAVs feature multiple rotating propellers located at the end of fixed arms and are capable of independent control of speed of each rotor-propeller combination. Simplicity of the mechanical design that avoids the complexity of swash plates and the associated control challenges, resulted in a vertical take-off

and landing class UAV. Its capability of hovering still in the air, its small size, low cost, and high agility, make this aircraft especially interesting as a research and development platform.

Autonomous flight of multirotor vehicles is an extremely challenging topic from a theoretical and practical standpoint, with far reaching implications in scientific and commercial mission scenarios. For this reason, in the past years the topic has been the subject of considerable research and development effort, mainly for what concerns state estimation and sensor fusion [1]–[3], vehicle's design [4], [5], and motion planning and control [6]–[12]. The aim of this work is to present an integrated framework that solves the cooperative mission planning, coordinated motion control, and collision avoidance problems. In the remainder of this article, it is assumed that the vehicles are equipped with solutions that provide knowledge of the vehicles' states, as well as perception of external objects within a given detection range.

Research on autonomous operation of teams of cooperating multirotors is particularly extensive [7], [13]–[15]. The literature is mainly divided into two categories: *centralized* and *decentralized* cooperative strategies. The former refers to the case in which all of the vehicles are driven by a central computer. The latter, instead, assumes that each UAV runs its own planning and control algorithms, and is thus able to autonomously react to other vehicles' behavior and/or unforeseen events to safely achieve a common goal. For representative work in this area the reader is referred to [13], [16], [17] and references therein. When dealing with small multirotor UAVs, often equipped with CPUs with limited capabilities, a *hybrid* strategy, that is a combination of centralized and decentralized solutions, can be applied. This article presents a hybrid setup, where a central unit is responsible for the mission planning, and communicates with the vehicles before the beginning of the mission. Then, decentralized controllers embedded onboard the vehicles

ensure that the mission is accomplished in a safe manner by exchanging information with each other. In this sense, the hybrid controller described here is more robust than purely decentralized solutions, as the centralized planning phase is not affected by faults in the communication network; moreover, the mission execution is safer with respect to purely centralized solutions, since the decentralized control solution allows each vehicle to directly react in a timely fashion to other vehicles' failures and potentially hazardous maneuvers, without having to communicate with a central station.

When dealing with decentralized and hybrid cooperative controllers, the control designer needs to keep in mind that the performance of the overall system largely depends on the ability of the fleet to exchange information in a timely and reliable manner. Therefore, the quality of service of the supporting communication network plays a crucial role. In many scenarios the flow of information among vehicles may be severely restricted by the nature of the supporting communication network. As a consequence, no vehicle may be able to communicate with the entire fleet, and the amount of information that can be exchanged may be limited.

Another issue that must be taken into consideration in multiple UAVs missions is safety, which in the context of this article is regarded as the capability of each UAV to be able to avoid unpredicted static and dynamic obstacles in the airspace, and at the same time guarantee a minimum separation distance to the other vehicles involved in the cooperative mission. When a dynamic obstacle, for example an uncooperative UAV, enters the airspace of operation without its presence having been taken explicitly into account during the the trajectory-generation phase, sense and avoid technologies are necessary in order to detect and avert potential collisions. The problem of collision avoidance has been widely explored in the robotics and controls community.

A significant amount of this research is covered by velocity-obstacle (also known as collision-cone) [18]–[20] and artificial potential field based approaches [21]–[23]. A key advantage of these methods is their simplicity. However, they often ignore vehicle dynamics, and are hard to extend to multi-vehicle coordinated missions, where apart from collision avoidance, time-coordination between vehicles may be critical. Lately, rapidly exploring random trees [24], [25] based methods have been used for trajectory planning (or replanning), such that vehicle dynamics are also satisfied. In multi-vehicle operations, while performing escaping maneuvers to avoid obstacles, the presence of other vehicles in the fleet must also be taken into account. Namely, collision avoidance maneuvers must not interfere with the trajectories of the other UAVs in the team. Motivated by the challenges described above, this work addresses the problem of safely coordinating a fleet of multirotor UAVs in the presence of communication constraints and static and dynamic obstacles.

Executing Safe Cooperative Path-Following Maneuvers

The cooperative missions considered in this article require that each vehicle follow a feasible collision-free path, and that all vehicles arrive at their respective final destinations at the same time, or at different times so as to meet a desired inter-vehicle schedule, while avoiding unknown moving (or static) objects. Successful completion of the above mentioned mission can be achieved by combining strategies for *trajectory generation*, *cooperative path following*, and *collision avoidance*. The general framework presented is depicted in Figure 1. In the adopted framework, the integration of these three motion-planning and control modules can be summarized as follows: first, a (real-time) trajectory-generation algorithm produces a set of spatial

paths and speed profiles that account for initial and final conditions, vehicle dynamics, spatial and temporal coordination constraints, and inter-vehicle safety requirements; second, a cooperative path-following algorithm enables the UAVs to follow the desired paths while adjusting the speed of the vehicles to ensure coordination; and third, a collision-avoidance algorithm modifies —on the fly— the spatial paths and/or the speed profiles to avert a possible collision when an obstacle is detected.

The methodology developed is based on three key ideas. First, the trajectory-generation problem is solved by adopting a geometric approach. In particular, the use of a special class of curves, namely *Pythagorean Hodograph Bézier curves*, allows for the efficient computation of trajectories that are deconflicted either in space or in time. Moreover, this class of curves is particularly useful to solve the collision-avoidance problem, as they allow for fast online reshaping of trajectories. Second, the use of path-following control techniques —in contrast to trajectory-tracking methodologies— allows the spatial and temporal assignments of the mission to be solved independently. This flexibility achieved by reparameterizing the trajectories with a suitably defined variable, referred in this article to as *virtual time*. With this formulation, the rate of progression of virtual time can be used as an extra degree-of-freedom to achieve vehicle coordination. Third, the control architecture exhibits a multi-loop structure in which the multirotors are assumed to be equipped with an inner-loop controller autopilot which stabilizes the vehicle dynamics. Then, guidance outer-loop controllers are designed to control the vehicle kinematics, providing cooperative path-following and collision-avoidance capabilities.

The article at hand is strongly rooted in past research results [26]–[32]. Three methods for mission planning, execution, and collision avoidance, already published in conference and

journal articles, are joined in a single work, and presented as an integrated solution to the safe cooperative motion control problem. The purpose of this article is to describe the integration of these methods, how they intertwine with each other, and the benefits that their combination offers to a broad family of applications. In what follows, the overall control architecture is discussed with the help of a specific example, with the understanding that the general framework can be employed in a wider class of cooperative missions. Additional illustrative examples and case studies can be found in [27], [29], [30], [33]

A motivational example: coordinated road search

The example at hand is depicted in Figure 2. Three multirotor UAVs, denoted as UAV_1, UAV_2, and UAV_3, equipped with cameras, are tasked to inspect the road illustrated in Figure 2b from point A to point B. Initially, the UAVs are hovering at their initial positions. First, the trajectory-generation algorithm computes three *transition paths*, which start at the vehicles' initial positions, and end at specific desired locations. Additionally, the algorithm generates *road search paths*, which follow the road to allow the vehicles to inspect the desired area. The transition and road search paths need to be deconflicted and have to satisfy the dynamic constraints of the vehicles. Further, the position and speed of each UAV at the end of the transition paths need to coincide with the position and speed at the beginning of the road search paths, respectively, to allow for a continuous progression of the mission. Second, the cooperative path-following algorithm enables the UAVs to follow the paths, and at the same time enforces the mission's temporal constraints. Coordination along the transition paths ensures that the vehicles arrive at the final destination at the same time with desired speed profiles, and ensures inter-vehicle collision avoidance. Coordination along the road search paths guarantees overlapping of the

fields of view of the three cameras, as emphasized by Figure 2a. While the mission unfolds, the collision-avoidance algorithm allows the vehicles to avert possible collisions with obstacles that interfere with the trajectories of the UAVs.

Finally, it is possible that new points of interest appear that need to be inspected. This is the case for UAV_3, which is required to deviate and inspect a secondary road. After inspection, the vehicle re-converges to the original road search path synchronizing with the rest of the fleet. This last part highlights the importance of the use of multiple cooperative vehicles in such mission scenarios. Moreover, it brings to the reader's attention the benefits of employing cooperative control algorithms that —like the one presented in this article— do not necessarily lead to swarming behaviors.

Cooperative Trajectory Generation

A key enabling element for the realization of these cooperative missions is the availability of efficient cooperative planning strategies that can be implemented onboard the vehicles. A planning algorithm has to work within a complex set of constraints, and (near) real-time generation of trajectories is desired to allow the vehicles to replan their trajectories, if necessary. Typically, multirotor UAVs carry low power-consumption processors with limited memory in order to save weight for maximum payload and flight endurance capabilities. Therefore, the planning algorithm has to be computationally efficient. It is in this spirit that this article presents a solution that allows for efficiently generating trajectories for multiple cooperating vehicles, based on the following key ideas: (i) the desired trajectory is generated for the entire mission without discretizing the trajectory temporally or spatially; (ii) the desired trajectory is decomposed into

a geometric and a temporal element; and (iii) a specific family of polynomials is used that have favorable geometric and mathematical properties.

A stringent requirement, especially for multi-vehicle missions, is that the trajectories are collision-free with each other and, of course, with (static and dynamic) obstacles. With the adopted approach, inter-vehicle safety distances are guaranteed during the planning phase, through either *temporal* or *spatial* separation. Moreover, the desired trajectories that are generated within this framework, also allow for an elegant way to avoid potential collisions with obstacles during the execution of the mission, as shown in the discussion on collision avoidance.

Before going into details of the adopted framework, a formulation of the three-dimensional trajectory-generation problem for a general setting is given first. The objective here is to generate N trajectories $\mathbf{p}_{d,i}(t_d)$

$$\mathbf{p}_{d,i} : [0, t^{\mathrm{f}}_{d,i}] \to \mathbb{R}^3 \qquad\qquad i = 1, 2, \ldots, N\,, \tag{1}$$

where $t_d \in [0, T_d\,]$, with $T_d := \max\{t^{\mathrm{f}}_{d,1}, \ldots, t^{\mathrm{f}}_{d,N}\}$, is the time variable (used during the trajectory-generation phase and distinct from the actual mission time t that evolves as the mission unfolds), $t^{\mathrm{f}}_{d,i} \in \mathbb{R}^+$ are the individual final mission times of the vehicles obtained during the planning phase, and N is the number of vehicles. Note that the mission for vehicle i has terminated for all $t_d > t^{\mathrm{f}}_{d,i}$. For simplicity, this article only considers missions where simultaneous time-of-arrival of the vehicles is required, and therefore, $t^{\mathrm{f}}_{d,1} = \ldots = t^{\mathrm{f}}_{d,N} = T_d$. Together, the trajectories $\mathbf{p}_{d,i}(t_d)$ minimize a global cost function $J(\cdot)$, are collision-free, and satisfy boundary conditions, spatial constraints, and temporal constraints. Examples of commonly used cost functions include total path length, total mission time, and speed deviation from initial speed, to mention but a few. Spatial and temporal constraints can be mission-specific, such as

simultaneous arrival of the vehicles or endurance constraints, but can also be related to the dynamics of the vehicles, for example adhering to the maximum speed and acceleration of the multirotors.

Instead of generating the trajectories explicitly as a function of time, as given by Equation (1), each trajectory is decomposed into a *spatial path*, a geometric element with no temporal specifications, and a *timing law* associated with this path, which captures the temporal assignments of the trajectory. In fact, the timing law allows us to independently adjust the speeds of the vehicles, without altering the spatial paths along which these vehicles travel. In other words, different speed profiles $v_{d,i}(\cdot)$ can be assigned to a given spatial path $\mathbf{p}_{d,i}(\cdot)$ by changing the parameters of the timing law. In terms of the optimization problem presented subsequently, this feature translates to increasing the number of optimization variables, *without* increasing the degree of the polynomials that describe the *spatial paths*. The concept of decoupling the geometric curve and the timing law in order to adjust the spatial path and the speed profile independently was first described in [34], and later applied in [35].

The first step in decomposing the trajectory is to parameterize the spatial paths $\mathbf{p}_{d,i}(\cdot)$ by a dimensionless variable $\zeta_i \in [0, 1]$. In general, these spatial paths $\mathbf{p}_{d,i}(\zeta_i)$ are described by polynomials since evaluating polynomials only requires addition and multiplication, which are arithmetic operations that can be performed efficiently by digital computers. In the approach presented, the spatial paths are described by *Pythagorean Hodograph (PH) Bézier curves*. The main motivation for using PH Bézier curves instead of polynomials expressed in the monomial (or power) basis is driven by: (i) the existence of many computationally efficient algorithms designed for Bézier curves, such as algorithms to efficiently compute the minimum distance

between two Bézier curves [36]; and (ii) the existence of a closed-form solution for the arc lengths of the paths. The use of Bézier and PH Bézier curves for path-planning and trajectory generation has been reported in several works [37]–[39].

Hence, the spatial paths are described by *quintic* PH Bézier curves, that is, for $i = 1, \ldots, N$

$$\mathbf{p}_{d,i}(\zeta_i) = \sum_{k=0}^{5} \bar{\mathbf{p}}_{i,k}\, b_k^5(\zeta_i)\,, \tag{2}$$

where $\bar{\mathbf{p}}_{i,k} \in \mathbb{R}^3$ are the control points of the spatial path $\mathbf{p}_{d,i}(\zeta_i)$, and $b_k^5(\cdot)$ are the (up to degree 5) Bernstein basis polynomials. The choice for working with PH Bézier curves of degree 5 may seem restrictive, as it limits the class of trajectories that can be generated. It is, however, justified based on several considerations. First, increasing the degree of the spatial PH Bézier curves offers more flexibility in satisfying the constraints, but can potentially result in an increased computational cost. Second, PH Bézier curves of higher degree may exhibit unknown undesirable behavior and characteristics, since they have not been extensively explored yet. On the contrary, the properties and behavior of the quintic PH Bézier curves are extensively studied in, for example, [40], [41]. Finally, the quintic PH Bézier curves are the simplest PH Bézier curves that allow constructing smooth curves with given endpoints and derivatives [40], where the latter are boundary conditions on interest common to any general trajectory-generation problem.

Next, in order to reconstruct the spatial trajectory $\mathbf{p}_{d,i}(t_d)$, the dimensionless parameter ζ_i needs to be related to time t_d. This relationship is provided by the timing law $\theta_i(\cdot)$. The timing law dictates how the variable ζ_i for the ith vehicle evolves with the time variable t_d and, as such, affects the desired rate at which the vehicle travels along the planned path. Hence, the timing law offers a means to meet the temporal requirements of the mission. Let the timing law $\theta_i(\cdot)$

be given through a dynamic relation of the form

$$\theta_i(t_d) = \frac{\mathrm{d}\zeta_i}{\mathrm{d}t_d}\,, \tag{3}$$

where $\theta_i(t_d)$ is a smooth positive polynomial function. In order to preserve the computationally attractive properties within the trajectory-generation framework, the timing laws $\theta_i(t_d)$ are defined by *quadratic* Bézier curves:

$$\theta_i(t_d) = \sum_{k=0}^{2} \bar{\theta}_{i,k}\, b_k^2(t_d)\,, \tag{4}$$

where $\bar{\theta}_{i,k} \in \mathbb{R}$ are the control points of the timing law $\theta_i(t_d)$. The parameter for any Bézier curve is defined on the interval $[0, 1]$ and, therefore, the time variable for the timing law needs to be properly scaled by T_d. This concept is explained in more detail in [32]. However, for the sake of brevity, the parameter of the timing law is denoted by t_d throughout the rest of the article.

Note that it is highly desirable that an analytical expression for the function $\zeta_i(t_d)$ exists, as the map $\zeta_i(t_d)$ permits a one-to-one correspondence between the time variable t_d and the parameter ζ_i. As such, the desired position $\mathbf{p}_{d,i}(\zeta_i(t_d))$ of the ith vehicle at time t_d can be obtained through Equation (2). With the timing law $\theta_i(t_d)$ defined as in Equation (3), the map $\zeta_i(t_d)$ is given by the integral

$$\zeta_i(t_d) = \int_0^{t_d} \theta_i(\tau)\mathrm{d}\tau\,, \tag{5}$$

which is also a Bézier polynomial and of degree 3.

Given the problem formulation above, the trajectory-generation framework can be cast into a constrained optimization problem where a set of desired trajectories are obtained by

minimizing the cost function $J(\cdot)$. The optimization problem can be formulated as follows:

$$\begin{aligned} &\min_{\boldsymbol{\Xi}_i \times \ldots \times \boldsymbol{\Xi}_N} J(\cdot) \\ &\text{subject to} \quad \text{boundary conditions,} \\ &\qquad\qquad\quad \text{dynamic constraints of the vehicles,} \\ &\qquad\qquad\quad \text{mission-specific constraints,} \\ &\qquad\qquad\quad \text{minimum separation constraints,} \end{aligned} \tag{6}$$

where $\boldsymbol{\Xi}_i$ represents the vector of optimization parameters for the ith vehicle, and $J(\cdot)$ is a given cost function that may include terms related to mission-specific goals. The optimization parameters are variables that uniquely determine the control points $\bar{\mathbf{p}}_{i,k}$ and $\bar{\theta}_{i,k}$, which characterize the spatial paths $\mathbf{p}_{d,i}(\zeta_i)$ and timing laws $\theta_i(t_d)$, given in Equations (2) and (4), respectively. The expressions that are used to compute the control points from a given vector of optimization variables $\boldsymbol{\Xi}_i$ are presented in [32]. An example of a cost function $J(\cdot)$ could be the sum of the arc lengths. In this case, it can be verified that $J(\cdot)$ is a function of the optimization variables that determine the control points $\bar{\mathbf{p}}_{i,k}$.

Boundary conditions and feasibility constraints

This section presents the set of constraints that a given pair of spatial path $\mathbf{p}_{d,i}(\zeta_i)$ and timing law $\theta_i(t_d)$ must satisfy in order to be considered a feasible trajectory for vehicle i. First, in the problem of trajectory generation for autonomous vehicles, typically the initial and final conditions of the trajectory, here referred to as boundary conditions, are prespecified. Hence, for

the ith vehicle the following boundary conditions are given:

$$\begin{aligned}
&\mathbf{p}_{d,i}(\zeta_i = 0) = \mathbf{p}_i^{\mathrm{i}}\,, \quad \gamma_i(\zeta_i = 0) = \gamma_i^{\mathrm{i}}\,, \quad \psi_i(\zeta_i = 0) = \psi_i^{\mathrm{i}}\,, \quad v_{d,i}(t_d = 0) = v_i^{\mathrm{i}}\,, \\
&\mathbf{p}_{d,i}(\zeta_i = 1) = \mathbf{p}_i^{\mathrm{f}}\,, \quad \gamma_i(\zeta_i = 1) = \gamma_i^{\mathrm{f}}\,, \quad \psi_i(\zeta_i = 1) = \psi_i^{\mathrm{f}}\,, \quad v_{d,i}(t_d = T_d) = v_i^{\mathrm{f}}\,,
\end{aligned} \tag{7}$$

where $\mathbf{p}_i^{\mathrm{i}}$, γ_i^{i}, ψ_i^{i}, and v_i^{i} are the initial position, flight-path angle, course, and speed, respectively, while $\mathbf{p}_i^{\mathrm{f}}$, γ_i^{f}, ψ_i^{f}, and v_i^{f} are the specified quantities at the final endpoint of the trajectory.

Flyable trajectories

Flyable trajectories are trajectories that comply with the dynamic constraints of the vehicles and, therefore, can be closely followed if the vehicles executing the mission are equipped with flight control systems that enable accurate tracking of the reference commands. Trajectories that are not flyable inevitably result in path-following errors and may jeopardize the completion of the mission or, in the worst case, lead to a loss of a vehicle.

It is of common practice to relate dynamic constraints of the vehicles to geometric properties of the spatial path. However, this approach has several severe drawbacks, of which the most detrimental one is that straight lines and curves with inflection points, for which the *torsion* τ along these curves is unbounded, are ruled out as candidate spatial paths [32]. The approach that is adopted here is to derive expressions for the kinematics of the vehicles along a given trajectory, characterized by $\mathbf{p}_{d,i}(\zeta_i)$ and $\theta_i(t_d)$, and subject these expressions to bounds that can be obtained from the dynamic constraints of the vehicles. It is not surprising, and in fact desirable, that these equations can be formulated in Bézier form.

Multirotor UAVs are extremely agile aerial vehicles, and their dynamic constraints are solely determined by the available thrust provided by the rotors. For example, multirotors are

capable of hovering, and can change their attitude practically at any rate by tilting the total thrust vector. Therefore, the only dynamic constraints of a multirotor are given by the maximum speed $v_{\max}$ and maximum total acceleration $a_{\max}$, which can be determined from the maximum available thrust throughout the flight envelope. Hence, the dynamic constraints of multirotors can be specified as

$$v_{d,i}(t_d) \leq v_{d,\max} < v_{\max}\,, \quad a_{d,i}(t_d) \leq a_{d,\max} < a_{\max}\,. \tag{8}$$

Note that during the trajectory-generation phase (in particular, in the optimization problem of Equation (6)), the more restrictive bounds $v_{d,\max}$ and $a_{d,\max}$ are used so as to allow the time-coordination and collision avoidance algorithms to adjust the speeds of the vehicles, if necessary, in order to maintain coordination between the multirotors or to avoid potential collisions with obstacles, respectively. The planning algorithm must ensure that the generated trajectories do not demand maneuvers that exceed the bounds given in Equation (8) on the speed and acceleration profiles. Hence, it is necessary to derive the speed and acceleration profiles along a given trajectory. Given the timing law $\theta_i(t_d)$, the desired speed profile $v_{d,i}(t_d)$ can be found by differentiating the spatial path $\mathbf{p}_{d,i}(\zeta_i)$ with respect to time t_d:

$$v_{d,i}(t_d) = \|\mathbf{p}'_{d,i}(\zeta_i(t_d))\|\theta_i(t_d)\,, \tag{9}$$

where $\mathbf{p}'_{d,i}(\zeta_i(t_d))$ is the first derivative of $\mathbf{p}_{d,i}(\zeta_i(t_d))$ with respect to ζ_i. It can be verified that $v_{d,i}(t_d)$ is a Bézier polynomial [32], due to the use of PH Bézier curves.

In general, the equation for the acceleration profile $a_{d,i}(t_d)$ does not permit a Bézier polynomial form. This fact becomes apparent when taking a closer look at the expression for the total acceleration $a_{d,i}(t_d)$

$$a_{d,i}(t_d) = \left\| \frac{\mathrm{d}^2\mathbf{p}_i(\zeta_i(t_d))}{\mathrm{d}t_d^2} \right\| = \left\| \mathbf{p}'_{d,i}(\zeta_i(t_d))\frac{\mathrm{d}\theta_i(t_d)}{\mathrm{d}t_d} + \mathbf{p}''_{d,i}(\zeta_i(t_d))\theta_i^2(t_d) \right\|\,, \tag{10}$$

where $\mathbf{p}''_{d,i}(\zeta_i(t_d))$ is the second derivative of $\mathbf{p}_{d,i}(\zeta_i(t_d))$ with respect to ζ_i. It is clear, that the right-hand side of Equation (10) does not result in a polynomial as it involves a square root arising from the 2-norm. To preserve the Bézier polynomial structure in the framework, the expression for $a^2_{d,i}(t_d)$ is derived and evaluated against a^2_{max}. Summarized, in order for the generated trajectories to meet the dynamic constraints of the vehicles, Equations (9) and (10) are required to satisfy the following bounds, which are equivalent to the dynamic constraints given in Equation (8):

$$v_{d,i}(t_d) \leq v_{d,\text{max}} < v_{\text{max}}\,, \quad a^2_{d,i}(t_d) \leq a^2_{d,\text{max}} < a^2_{\text{max}}\,. \tag{11}$$

Note that scaling by T_d is omitted in Equations (9) and (10). The reader is referred to [32] for the complete equations.

Feasible trajectories

Feasible trajectories are flyable trajectories that are also spatially deconflicted in order to avoid inter-vehicle collision and, thus, ensure safe simultaneous operation in a common airspace. Deconfliction between trajectories can be guaranteed through *spatial separation* (the spatial paths are separated in space) or *temporal separation* (the vehicles are separated in time). While temporal separation is more computationally efficient, it relies heavily on the performance of a time-coordination algorithm, which, in turn, depends on the quality and robustness of the communication network over which the vehicles exchange information with each other. On the other hand, spatial separation can be employed in situations where the communication network is faulty or jammed, and coordination cannot be guaranteed. The approach presented in this article offers the mission planner the flexibility to generate trajectories that are either spatially

or temporally deconflicted. A more in-depth discussion on both strategies can be found in [32].

It is here where the presented trajectory-generation framework distinguishes itself from other novel approaches to path planning that discretize the trajectories either in space or in time. When discretizing the trajectories in one way or another, spatial separation between the trajectories can be ensured at the discretization nodes, but unfortunately, deconfliction is not guaranteed in between the nodes. To avoid violation of the minimum separation requirement in between the nodes, the number of nodes can be increased. However, increasing the number of nodes considerably affects the spatial and temporal scalability of the method. For example, if the number of nodes increases proportionally, then the number of deconfliction constraints alone increases quadratically. The authors of [42] and [43] state this issue as one of the limitations of their approaches, where the trajectories in both methodologies are discretized in time.

In order to ascertain whether two paths are sufficiently deconflicted, it is essential to compute the minimum distance between the two curves. As discussed earlier, Bézier curves are completely determined by a finite number of control points and are contained within the convex hull of these control points. These properties make Bézier curves extremely suitable to describe spatial paths, for the minimum distance between two Bézier curves can be computed efficiently [36], without resorting to any kind of discretization of the curve. Therefore, not only are the trajectories guaranteed to be collision-free by formulating the trajectory-generation framework in terms of Bézier polynomials; moreover, trajectory generation is achieved at an extremely high computational efficiency. In this article, generation of collision-free trajectories is demonstrated by imposing temporal separation for the example of a cooperative mission, as described in the previous section. For illustrative examples of deconfliction through spatial

separation, the reader is referred to [26], [29], [31], [32].

A minimum spatial clearance E has to be ensured between all pairs of generated trajectories for the set of trajectories to be collision-free. In the case when temporal separation is preferred over spatial separation, deconfliction is ensured if, for any time t_d, the minimum distance between the ith and jth vehicle is greater than or equal to this minimum spatial clearance E. Therefore, the path $\mathbf{p}_{d,i}(\zeta_i)$ has to be re-parameterized by the time variable t_d. This task can be performed efficiently since the trajectories are represented by Bézier polynomials, and the composition of functions is a Bézier polynomial on its own. It can be easily verified that $\mathbf{p}_{d,i}(t_d)$ is a degree 15 spatial PH Bézier curve. Hence, the *trajectory* for the ith vehicle is described by

$$\mathbf{p}_{d,i}(t_d) = \sum_{k=0}^{15} \tilde{\mathbf{p}}_{i,k} b_k^{15}(t_d)\,,$$

where $\tilde{\mathbf{p}}_{i,k}$ are the control points, which can be determined using a recursive algorithm for computing the control points of the composition of two Bernstein polynomials. Then, the temporal separation between the ith and jth vehicle is defined as

$$\min_{\substack{i,j=1,\ldots,N \\ i\neq j}} \left\| \sum_{k=0}^{15} \left(\tilde{\mathbf{p}}_{i,k} - \tilde{\mathbf{p}}_{j,k}\right) b_k^{15}(t_d) \right\|^2 \geq E_d^2\,, \qquad \text{for all} \quad t_d \in [0, T_d] \tag{12}$$

where E_d, that is used during the trajectory-generation phase, is larger than the actual required minimum distance E, so as to allow for path-following errors or deviations from the desired paths necessary to prevent a potential collision. Notice that the lefthand side of Equation (12) is equivalent to finding the minimum distance between a Bézier curve and the origin $[0, 0, 0]^\top$ and, therefore, Equation (12) can be efficiently evaluated by using the minimum distance algorithm [36]. The importance of the algorithm to efficiently evaluate Equation (12) cannot be overemphasized here.

The constrained optimization problem presented in Equation (6) has now been re-formulated as a framework that specifically uses quintic PH Bézier curves to represent the spatial paths, and quadratic Bézier polynomials for the timing laws. The constrained optimization problem belongs to the class of semi-infinite optimization problems [44]. Due to the non-convex formulation, the solutions are in general suboptimal.

Illustrative example

What follows is a demonstration of the efficacy of the developed trajectory-generation framework. Desired trajectories are generated for the multirotors executing the transition phase of the mission, described earlier as an example of a typical multi-vehicle cooperative mission. During the planning phase, a desired trajectory for each UAV is generated so as to cooperatively execute the transition from vehicle launch to the start of the road search mission. The numerical values for the boundary conditions for this part of the mission are given in Table I, along with the dynamic constraints for each individual multirotor. The cooperative trajectory-generation framework is implemented in Matlab® on a desktop computer with Intel® Core™ i5-3470 CPU 3.20GHz, 8GB of RAM and running 64-bit Windows 7. The number of optimization variables is $4N + 1$, while the number of inequality constraints is $\frac{1}{2}N(N-1) + 2N$. The latter includes the evaluation of the minimum distance between $\frac{1}{2}N(N-1)$ pairs of trajectories. The computation time taken to generate the set of trajectories for this particular example, where $N = 3$, is 12.4 s.

Figure 3a shows the flight paths for transition phase. The separation between the vehicles is presented in Figure 3b and the separations between the paths are given in Figures 3c-3e. Recall, that deconfliction is enforced in this mission by temporal separation. Therefore, although the

minimum spatial separations between the paths are less than the required minimum distance $E_d = 2$ m during the planning phase, as shown in Figures 3c-3e, the temporal separation requirement is not violated, and the algorithm ensures that the vehicles are sufficiently separated from each other at any point in time t_d (Figure 3b). Lastly, from Figure 4 it is clear that the generated desired trajectories do not violate the maximum permissible speed and total acceleration.

Cooperative Path Following of Teams of Multirotor UAVs

This section presents the cooperative control framework that allows a fleet of vehicles to execute coordinated, collision-free maneuvers. In the adopted framework, a path-following algorithm —implemented onboard the UAVs— is responsible for making each vehicle follow a *virtual target* vehicle running along the spatial path, while a distributed coordination control law adjusts the rate of progression of this virtual vehicle so as to coordinate the entire fleet. One of the main benefits of this framework lies in the fact that the speed of the vehicles is adjusted online to synchronize the UAVs, as opposed to the coordinated trajectory-tracking approach where the coordination task is solved offline, and thus the control algorithm cannot adapt to external disturbances or vehicles' tracking errors.

The cooperative path-following control problem is solved in three main steps: the first step consists of implementing a reference vehicle, a *virtual target*, moving along the path computed by the trajectory-generation algorithm described earlier. This objective is achieved by introducing a new parameter, *virtual time*, denoted here as γ_i, and letting the desired trajectory to be followed by the multirotor UAVs be $\mathbf{p}_{d,i}(\gamma_i(t))$, where the subscript i refers to the ith UAV involved in the cooperative mission; the second step consists of making each UAV follow the *virtual target*.

This step, referred to as path following, reduces to driving a suitably defined error vector to zero by using the vehicle's control inputs, for example, angular rates and total thrust [28], or angles and vertical speed [27]; finally, to enforce the temporal constraints of the mission, a consensus problem is formulated, in which the objective of the fleet of vehicles is to reach agreement on some distributed variables of interest that capture the objective of the time-coordination control problem.

In the remainder of this section, the three steps mentioned above are described in detail, and a solution to the cooperative control problem is given. An overview of different approaches used for the derivation of cooperative path-following algorithms can be found in "Cooperative Path Following".

Following a virtual target

Given the trajectory $\mathbf{p}_{d,i}(t_d)$ produced by the trajectory-generation algorithm described in the previous section, and letting the *virtual time* $\gamma_i(t)$ be a function (yet to be defined) that relates actual (clock) time t to mission planning time t_d:

$$\gamma_i : \mathbb{R}^+ \to [0, T_d] \,, \quad \text{for all} \quad i = 1 \,, \, \dots \,, \, N \,, \tag{13}$$

then the *virtual target*'s position, velocity, and acceleration, which are the *commands* to be followed by the ith UAV at time t, are defined as

$$\mathbf{p}_{c,i}(t) = \mathbf{p}_{d,i}(\gamma_i(t)) \,, \quad \mathbf{v}_{c,i}(t) = \dot{\mathbf{p}}_{d,i}(\gamma_i(t), \dot{\gamma}_i(t)) \,, \quad \mathbf{a}_{c,i}(t) = \ddot{\mathbf{p}}_{d,i}(\gamma_i(t), \dot{\gamma}_i(t), \ddot{\gamma}_i(t)) \,. \tag{14}$$

With the above formulation, if $\dot{\gamma}_i(t) \equiv 1$, then the speed profile of the virtual target is equal to the *desired* speed profile chosen at the trajectory-generation level. More precisely, assume that

$\dot{\gamma}_i(t) = 1\,, \quad$ for all $\quad t \in [0, T_d]$, with $\gamma_i(0) = 0$. This assumption implies that $\gamma_i(t) = t_d$ for all t. In turn, the following equalities hold

$$\mathbf{p}_{c,i}(t) = \mathbf{p}_{d,i}(\gamma_i(t)) = \mathbf{p}_{d,i}(t) = \mathbf{p}_{d,i}(t_d)\,,$$

which, in other words, means that the desired and commanded trajectories coincide for all time, and the speed profile of the virtual target coincides with the desired speed profile chosen at trajectory-generation level. On the other hand, $\dot{\gamma}_i > 1$ ($\dot{\gamma}_i < 1$) implies a faster (slower) execution of the mission. This statement becomes evident when expressing the speed of the virtual target in terms of the derivative of the desired trajectory $\mathbf{p}_{d,i}(t_d)$ as follows:

$$\left\|\dot{\mathbf{p}}_{c,i}(t)\right\| = \|\dot{\mathbf{p}}_{d,i}(\gamma_i(t), \dot{\gamma}_i(t))\| = \left\| \frac{d\,\mathbf{p}_{d,i}(\gamma_i(t))}{d\gamma_i(t)} \frac{d\gamma_i(t)}{dt} \right\| = \left\| \frac{d\,\mathbf{p}_{d,i}(\gamma_i(t))}{d\gamma_i(t)} \dot{\gamma}_i(t) \right\|\,. \tag{15}$$

As discussed later in this article, the dynamics of $\gamma_i(t)$ (actually its second derivative $\ddot{\gamma}_i(t)$) can be explicitly controlled and used as an extra degree-of-freedom to achieve time coordination. Therefore, since $\ddot{\gamma}_i(t)$ is governed by some control law, yet to be defined, the dynamic constraints on the speed and acceleration of the vehicle, as well as the bounds given in (8) must be considered in order to derive feasibility limits on $\dot{\gamma}_i(t)$ and $\ddot{\gamma}_i(t)$. These limits can be determined using Equation (15), where $\dot{\mathbf{p}}_{d,i}(\gamma_i(t), \dot{\gamma}_i(t)) = d(\mathbf{p}_{d,i}(\gamma_i(t)))/dt$ denotes the velocity of the virtual target, corresponding to the commanded velocity vector to be tracked by the UAV at time t. Hence, $\|\dot{\mathbf{p}}_{c,i}(t)\|$ is limited to the physical speed constraints of the vehicle, that is,

$$\|\dot{\mathbf{p}}_{c,i}(t)\| \leq v_{\text{max}}\,. \tag{16}$$

Using (15), these speed constraints result in the following inequalities:

$$\dot{\gamma}_{\text{max}}\, v_{d,\text{max}} \leq v_{\text{max}}\,. \tag{17}$$

Equations (16) and (17) relate the limits of the commanded speed profile $\|\dot{\mathbf{p}}_{c,i}(t)\|$ to the limits of $\dot{\gamma}_i(t)$. Similar limits can be derived for the commanded acceleration profile $\|\ddot{\mathbf{p}}_{c,i}(t)\|$. In fact,

by differentiating Equation (15), and imposing the following upper bound on the commanded acceleration

$$\|\ddot{\mathbf{p}}_{c,i}(t)\| \leq a_{\max}\,, \tag{18}$$

similar inequalities as (17) can be found for the acceleration and $\ddot{\gamma}_i(t)$:

$$\ddot{\gamma}_{\max}\, v_{d,\max} + \dot{\gamma}_{\max}^2\, a_{d,\max} \leq a_{\max}\,. \tag{19}$$

Equations (18) and (19) relate the limits of the commanded acceleration profiles $\|\ddot{\mathbf{p}}_{c,i}(t)\|$ to the limits of $\dot{\gamma}_i(t)$ and $\ddot{\gamma}_i(t)$. In other words, Equations (17) and (19) give an explicit relationship between the dynamic constraints imposed at the trajectory-generation level, namely $v_{d,\max}$ and $a_{d,\max}$ in Equation (8) (which, in turn, determine the geometric path and timing law given in Equations (2) and (4), respectively), to the saturation limits of the dynamics of virtual time, namely $\dot{\gamma}_{\max}$ and $\ddot{\gamma}_{\max}$.

With this setup, the objective is to derive a path-following controller to ensure that the vehicles converge to their respective virtual targets.

To formally state the path-following problem, let $\mathbf{p}_i(t) \in \mathbb{R}^3$ be the position of the center of mass of the ith multirotor. Since $\mathbf{p}_{c,i}(t)$ describes the commanded position to be followed by the ith vehicle at time t, the position and velocity error vectors are defined as

$$\mathbf{e}_{p,i}(t) = \mathbf{p}_{c,i}(t) - \mathbf{p}_i(t) \in \mathbb{R}^3\,, \quad \mathbf{e}_{v,i}(t) = \mathbf{v}_{c,i}(t) - \dot{\mathbf{p}}_i(t) \in \mathbb{R}^3\,. \tag{20}$$

Then, the path-following control objective reduces to that of regulating the path-following errors defined in Equation (20) to a neighborhood of zero. Notice that this approach leaves the yaw angle of the vehicles as an extra-degree of freedom, which can be adjusted without affecting the control framework.

In [27] and [28] the authors present two solutions to the path-following problem. In particular, in [28] it is assumed that the vehicle is equipped with an autopilot capable of tracking angular rates and total thrust commands. Then, it is shown that the path-following controller drives the path-following error to a neighborhood of zero even in the case of non-ideal tracking performance of the autopilot. A similar result is obtained in [27], where the authors present a path-following control law for AR.Drone UAVs, equipped with control systems for Euler-angle and vertical-speed command tracking.

Coordination among multiple vehicles

This section addresses the time-coordination problem of a fleet of N multirotor UAVs. In what follows, first, the objective of time coordination is defined; second, a set of assumptions for the supporting communication network is formulated that ensures coordination; finally, the time-coordination error vector is presented, and a formal statement of the problem at hand is given.

Definition of the coordination objective

Recall that the position of the virtual target assigned to the ith vehicle at time t is given by $\mathbf{p}_{c,i}(t) = \mathbf{p}_{d,i}(\gamma_i(t))$, where $\mathbf{p}_{d,i}(\cdot)$ is the geometric path obtained from the trajectory-generation algorithm, and the path parameter $\gamma_i(t)$ is the virtual time defined in (13). The virtual time and its first time derivative play a crucial role in the time-coordination problem. In fact, because the commanded path assigned to each vehicle is parameterized by $\gamma_i(t)$, then if

$$\gamma_i(t) - \gamma_j(t) = 0\,, \quad \text{for all} \quad i, j \in \{1, \ldots, N\}\,, \quad i \neq j\,, \tag{21}$$

at time t, all the virtual targets are coordinated. In addition, if

$$\dot{\gamma}_i(t) - \dot{\gamma}_d(t) = 0\,, \quad \text{for all} \quad i \in \{1, \ldots, N\}\,, \tag{22}$$

then the virtual targets run along the paths at the desired rate of progression $\dot{\gamma}_d \in \mathbb{R}$. The variable $\dot{\gamma}_d(t)$, shared among the vehicles, can be used to adjust the rate of progression of the mission. For example, if $\dot{\gamma}_d(t) \equiv 1$, and Equations (21) and (22) are satisfied for all the vehicles, then the mission is executed at the pace originally planned in the trajectory-generation phase. On the other hand, $\dot{\gamma}_d(t) > 1$ (or $\dot{\gamma}_d(t) < 1$), results in a faster (or slower) execution of the mission. As discussed in the next section, the use of the term $\dot{\gamma}_d(t)$ becomes important when external unforeseen events demand adjustments to the nominal execution of the mission, such as moving objects obstructing the flying zone for a limited time frame, requiring the vehicles to modify their speed profiles to avoid collision. However, in this section only the time-coordination problem is discussed, and $\dot{\gamma}_d(t)$ is regarded as a reference command, rather than a control input. Finally, Equations (21) and (22) capture the objective of time-coordination, and a control law for $\ddot{\gamma}_i(t)$ must be formulated to ensure convergence to this equilibrium.

Communication losses, dropouts, and switching topologies

To achieve the time-coordination objective, information must be exchanged among the vehicles over a supporting communication network. The information flow as well as the constraints imposed by the communication topology can be modeled using tools from algebraic graph theory. The reader is referred to [45] for key concepts and details on this topic. To account for the communication constraints imposed by the communication network, it is assumed that the individual vehicle i exchanges information with only a subset of all vehicles, denoted as

$\mathcal{N}_i(t)$. It is also assumed that the information between vehicles is transmitted bidirectionally, continuously, and without time delays. Moreover, to minimize the amount of information that must be interchanged between the vehicles, the UAVs are allowed to exchange only its own virtual time variable, $\gamma_i(t)$, with each other. Finally, it is assumed that the connectivity of the communication graph $\Gamma(t)$ that captures the communication network topology of the UAVs satisfies the following condition [46]

$$\frac{1}{NT}\int_{t}^{t+T} \mathbf{Q}\mathbf{L}(\tau)\mathbf{Q}^{\top} d\tau \geq \mu \mathbf{I}_{N-1}\,, \quad \text{for all } t \geq 0\,, \tag{23}$$

where $\mathbf{L}(t) \in \mathbb{R}^{N\times N}$ is the Laplacian of the graph $\Gamma(t)$, and $\mathbf{Q} \in \mathbb{R}^{(N-1)\times N}$ is a matrix such that $\mathbf{Q}\mathbf{1}_N = 0$ and $\mathbf{Q}(\mathbf{Q})^{\top} = \mathbf{I}_{N-1}$, with $\mathbf{1}_N$ being a vector in $\mathbb{R}^N$ whose components are all 1. In Equation (23), the parameters $T > 0$ and $\mu \in (0, 1]$ represent a measure of the level of connectivity of the communication graph. Note that condition (23), hereinafter referred to as *persistency of excitation* (PE)-like condition, requires the communication graph $\Gamma(t)$ to be connected only in an integral sense, not pointwise in time. As a matter of fact, the graph may be disconnected during some interval of time or may even fail to be connected at all times. In this sense, it is general enough to capture packet dropouts, loss of communication, and switching topologies.

Time-coordination control

With the above notation, the coordination control problem can be summarized as follows: consider a set of N multirotor UAVs equipped with autopilots and path-following algorithms that enable the vehicles to follow a set of N commanded trajectories $\mathbf{p}_{c,i}(t)$. Let the vehicles communicate with each other over a network satisfying the PE-like assumption given in Equation

(23). Then, the objective of the cooperative control problem is to design feedback control laws for $\ddot{\gamma}_i(t)$ for all the vehicles such that the states defined in Equations (21) and (22) converge to (a neighborhood of) zero, and such that inequalities (17) and (19) are not violated.

Coordination control law

To solve this problem, let the evolution of $\gamma_i(t)$ be given by

$$\ddot{\gamma}_i = \ddot{\gamma}_d - b(\dot{\gamma}_i - \dot{\gamma}_d) - a \sum_{j \in \mathcal{N}_i} (\gamma_i - \gamma_j) - \bar{\alpha}_i(\mathbf{e}_{p,i}) \,, \tag{24}$$

$$\gamma_i(0) = \gamma_d(0) = 0 \,, \quad \dot{\gamma}_i(0) = \dot{\gamma}_d(0) = 1 \,, \tag{25}$$

where a and b are positive coordination control gains, while $\bar{\alpha}_i(\mathbf{e}_{p,i})$ is defined as

$$\bar{\alpha}_i(\mathbf{e}_{p,i}) = \frac{\mathbf{v}_{c,i}^{\top} \mathbf{e}_{p,i}}{\|\mathbf{v}_{c,i}\| + \epsilon} \,,$$

with ϵ being a positive design parameter, and $\mathbf{e}_{p,i}$ the position error vector defined in Equation (20). The coordination control law given in Equation (24) comprises four terms. The feed-forward term, $\ddot{\gamma}_d$, allows the virtual target to follow the acceleration profile of γ_d (which, as discussed later, is adjusted by the collision-avoidance module). The second term, $-b(\dot{\gamma}_i - \dot{\gamma}_d)$, forces the virtual target to track the speed profile imposed by $\dot{\gamma}_d$, which corresponds to the control objective given in Equation (22). If $\dot{\gamma}_d$ is set to 1, then the virtual target converges to the desired speed profile determined by the trajectory-generation algorithm. The third term, $-a \sum_{j \in \mathcal{N}_i} (\gamma_i - \gamma_j)$, ensures that the virtual target associated with a given UAV coordinates its position along the path with its neighbors, as specified by the coordination requirement in Equation (21). Finally, the fourth term, $-\bar{\alpha}_i(\mathbf{e}_{p,i})$, is a correction term that accounts for along-track path-following errors. By virtue of this path-following dependent term if, for example, one

vehicle is far behind the desired position, $||\mathrm{e}_{p,i}|| \neq 0$, then its own virtual target modifies its speed and waits for it, thus making all the other vehicles involved in the cooperative mission slow down to maintain coordination. The contribution of the various terms of the coordination control law is illustrated in Figure 5, which shows UAV_1, UAV_2, and UAV_3 coordinating along the transition paths of the coordinated road search mission. For details on the path-following controller implemented to perform this simulation, the reader is referred to [28]. At time $t = 0$, UAV_1 and UAV_2 overlap with their respective virtual targets, while UAV_3 is positioned far away from the desired trajectory. When the mission starts, the virtual target associated with UAV_3 slows down ($\dot{\gamma}_3 < 1$) and waits for UAV_3 to catch up. Thanks to the coordination algorithm, the other two virtual targets also slow down, followed by their respective UAVs. Figure 5c shows the desired speed profiles, the speed profiles of the virtual targets, and the speed profiles of the actual vehicles. It is important to notice that, once the UAVs reach coordination, and UAV_3 converges to its virtual target, then the speed profiles of the UAVs converge to the desired speed profiles, and approach the final positions of the transition paths with speed equal to 4 m/s, as imposed by the trajectory-generation algorithm. Figure 5 also shows that the initial deceleration due to the path-following error of UAV_3 causes a delay in the mission, which terminates approximately 0.4 seconds after the mission time computed at the trajectory-generation level. Figures 5a and 5b depict the positions of the UAVs at times $t = 13$ s and $t = 18.3$ s, showing that the vehicles arrive at their final destination at the same time. The UAVs can then proceed along the road search paths and keep coordinating along them to ensure safety and success of the overall cooperative road search mission.

Convergence properties and results

The control law given by Equation (24) ensures that the time-coordination errors characterized by Equations (21) and (22) converge exponentially to zero in the presence of an autopilot which exhibits ideal performance [27], [29], [30]. Moreover, it is demonstrated that the maximum guaranteed exponential convergence rate is given by the sum of the converge rate of the path-following error and the following term

$$\frac{a}{b}\frac{N\mu}{T(1+\frac{a}{b}NT)^2}\,,$$

which depends on the control gains a and b, the number of vehicles N, and the quality of service of the communication network, characterized by the parameters T and μ. In other words, given N vehicles and fixed control gains, the performance of the time-coordination algorithm depends on the amount of information that the UAVs exchange with each other over time. This result is supported by Figure 5d, which shows the time history of the coordination error, computed as

$$\sqrt{(\gamma_1-\gamma_2)^2+(\gamma_1-\gamma_3)^2+(\gamma_2-\gamma_3)^2}\,, \tag{26}$$

when UAV_1, UAV_2, and UAV_3 coordinate along the transition paths. Notice that, for illustration purposes, the previous equation does not include the errors $\dot{\gamma}_i-\dot{\gamma}_d$. However, it has been argued previously that the speed profiles of the virtual targets converge to the desired speed profiles (which is, $\dot{\gamma}_1,\dot{\gamma}_2,\dot{\gamma}_3$ converge to $\dot{\gamma}_d=1$), which in turn implies that the speed profiles of the vehicles converge to the desired speed profiles. In Figure 5d the error given by Equation (26) is plotted in two different cases: first, in the case of all-to-all communication between the three vehicles, and second, in the case where UAV_1 communicates with only UAV_2, UAV_2 communicates with both vehicles, and UAV_3 communicates with UAV_2 only. As expected, the

algorithm exhibits better performance in the first case. Additionally, by properly scaling the control gains a and b with the number of vehicles N, the guaranteed rate of convergence reduces to $\mu/(4T^2)$, which indeed implies that the guaranteed performance of the algorithm does not depend on the number of vehicles involved in the cooperative mission [30]. Finally, it can be shown that the presented solution ensures stability of the multi-vehicle system in the case of non-ideal tracking performance of the autopilot [27], [30]. Also in this case it is shown that the guaranteed performance of the overall control architecture depends on the quality of service of the communication network and the performance of the autopilot.

In light of these results, it is important to notice that if the desired trajectories $\mathbf{p}_{d,i}(t_d)$ satisfy the temporal separation requirement given by Equation (12), then the solution to the cooperative path-following control problem ensures inter-vehicle safety (which is, the cooperative vehicles maintain a minimum separation even without knowing each others' positions). Figure 5b gives a convenient perspective of the mission at hand, and supports this statement. As highlighted in the figure, the paths of UAV_1 and UAV_2 intersect with each other. However, the minimum spatial clearance at the intersection is maintained in the presence of disturbances, by the coordination algorithm. As a last comment on this argument, the relationship between autopilot performance, quality of the service of the communication network, and performance of the cooperative path-following algorithm, can be judiciously used by the control designer to guarantee inter-vehicle safety throughout the mission. In fact, based on the knowledge of (i) the quality of service of the communication network, and (ii) the performance of the onboard autopilot, E_d in Equation (12) can be chosen large enough to guarantee that the vehicles never collide throughout the mission. The efficacy of the described cooperative control framework has been demonstrated in real flight test indoor scenarios. In particular, the coordinated path-following

algorithm is implemented, and two AR.Drones are programmed to autonomously perform various cooperative tasks (for example, synchronized exchange of position, phase-to-phase coordination on a circular trajectory, dance tango). Videos of the flight tests and simulations can be found in [47], where the reader can find further details on the experiments.

Collision Avoidance Strategies for Safe Operations

In a real mission scenario, unexpected obstacles could interfere with the UAVs and obstruct the airspace during a certain time interval. In this case, the vehicles are required to execute a collision avoidance maneuver to ensure safe operation in the airspace. To be effective and reliable, a collision avoidance algorithm must take into consideration many constraints: first, the algorithm must be fast to react to sudden unpredicted hazards; second, collision avoidance must be guaranteed with minimum exchange of information; third, a collision avoidance maneuver must not prevent the UAVs from satisfying other mission requirements such as time-coordination and final position and course; finally, when one vehicle is avoiding an external obstacle, it must do it in such a way as to also ensure inter-vehicle safety. The collision avoidance algorithm is based on three key steps, namely *detection*, *analysis*, and *avoidance*. Here, the detection algorithm continuously determines whether there is a possible collision based on information about the obstacle. In case a possible collision is predicted, the analysis step decides whether it must be avoided through speed adjustment or trajectory replanning. Lastly, depending on this analysis, either the speed adjustment block modifies the progression of the mission, or the trajectory replanning block redesigns the trajectory to avoid the obstacle. In what follows, these steps are described in detail.

Detection

Upon detection of an obstacle, it is assumed that a prediction of its trajectory is available. The obstacle's trajectory can be predicted using measurements of current position and velocity of the object, which can be obtained from onboard sensors (see, for example, [48]). This predicted trajectory can be updated at any time according to newly available information about the obstacle. Since this detection algorithm is run repeatedly during the mission, it automatically re-adjusts based on the updated predicted trajectory.

At any time during the mission, for a given predicted trajectory of the obstacle, the objective of the detection algorithm is to verify whether the obstacle's trajectory interferes with the assigned trajectories causing a potential collision. To this end, assume that within a detection range d_{s}, that is for

$$||\mathbf{p}_c(t) - \mathbf{p}_o(t)|| = ||\mathbf{p}_d(\gamma(t)) - \mathbf{p}_o(t)|| \leq d_{\text{s}}\,, \tag{27}$$

the UAV is able to detect the presence of an obstacle positioned at $\mathbf{p}_o(t)$. Notice that, for the sake of simplicity, the index i has been dropped from the position vector of the ith UAV. In fact, for detection and analysis parts of collision avoidance, only one UAV is considered, with the understanding that the solution is intended to be adopted for all vehicles (independently) in multi-vehicle missions.

The vehicle is considered to be in collision with the obstacle if there exist t_{col} and γ_{col} such that

$$||\mathbf{p}_d(\gamma_{\text{col}}) - \mathbf{p}_o(t_{\text{col}})|| = \min_{t,\gamma\geq 0} ||\mathbf{p}_d(\gamma) - \mathbf{p}_o(t)|| \leq d_{\text{safe}}\,,$$

where d_{safe} is the minimum safety distance required between the vehicle and the obstacle, and γ_{col}

and t_{col} denote the parameters of the UAV and obstacle trajectories, respectively. For simplicity, assume that the obstacle always intersects the desired trajectory of the UAV. Therefore, there exists $\mathbf{p}_{\text{col}}$ such that $\mathbf{p}_{\text{col}} = \mathbf{p}_d(\gamma_{\text{col}}) = \mathbf{p}_o(t_{\text{col}})$. Finally, define

$$\Delta t^{\text{v}} = |(\gamma_{\text{col}} - \gamma(t)) - (t_{\text{col}} - t)|$$

as the *virtual time separation* between the vehicle and the obstacle with respect to $\mathbf{p}_{\text{col}}$. To better understand the meaning of this *virtual time separation*, notice that $(t_{\text{col}} - t)$ is the mission-time needed for the obstacle to arrive at $\mathbf{p}_{\text{col}}$, while $(\gamma_{\text{col}} - \gamma(t))$ is the time needed for the UAV to arrive at this same location, if the rate of progression along the path is given by $\dot{\gamma}(t) = 1$ (recall that $\gamma(t)$ was defined as *virtual time*, and $\dot{\gamma}(t) = 1$ implies that $\gamma(t)$ evolves like time). Consider the following inequality:

$$\Delta t^{\text{v}} < t_{\text{sep}}\,, \tag{28}$$

where t_{sep} is a constant design parameter defined as *desired minimum temporal separation*, and which is discussed in more detail later. If condition (28) is not satisfied, the vehicle and the obstacle at $\mathbf{p}_{\text{col}}$ have a *virtual time separation* of at least t_{sep} seconds. Therefore, if the vehicle progresses along its trajectory with rate of progression $\dot{\gamma} = 1$, for an appropriately chosen t_{sep}, it is guaranteed that the UAV and the obstacle are sufficiently separated in space at $\mathbf{p}_{\text{col}}$, and hence the UAV does not need to take any action. On the other hand, if (28) is satisfied at time t, then requiring $\dot{\gamma} = 1$ possibly causes a collision, since both the vehicle and the obstacle reach $\mathbf{p}_{\text{col}}$ with an insufficient time separation. Obviously, the parameter t_{sep} must be chosen with particular attention, while taking into consideration the dynamics of the vehicle and the obstacle, as well as their dimensions. With this observations in mind, it can certainly be stated that *if at time t conditions* (27) *and* (28) *are satisfied, then the UAV must execute a collision avoidance maneuver*.

Analysis

Once a collision is predicted, there are many possible ways a UAV can avoid collision, such as through speed adjustment, change of course, or trajectory replanning. In multi-vehicle missions, it is usually preferred to only adjust the speed of the mission so that the UAV remains on the desired path for all times. However, depending on the speed and direction of the obstacle, adjusting the mission's speed is not always *preferable*. Moreover, dynamic speed and acceleration limits must be taken into consideration, to verify if such a maneuver is *feasible*. In the overall collision-avoidance framework, feasibility and preferability of the speed-adjustment methodology are first determined. The speed adjustment method is employed if it is found to be both feasible and preferable. Otherwise, the trajectory replanning is employed in order to avoid collisions.

Here, the speed-adjustment method is deemed preferable if

$$t_{\text{sep}} \geq \Delta t^{\text{v}}_{\text{pref}} \,, \tag{29}$$

where $\Delta t^{\text{v}}_{\text{pref}}$ represents the time change in mission time required to avoid the collision. On the other hand, the speed-adjustment method is considered to be feasible only if the dynamics of the vehicles involved in the mission allow for such a maneuver. The reader is referred to [29] for the exact formulation of these criteria.

Avoidance

Depending on the type of collision that the vehicle must avoid, and according to the analysis described before, the UAV can start an avoidance maneuver either by adjusting its speed without leaving the desired path, or by deviating from it. In what follows, both solutions

are described in detail.

Collision avoidance through speed adjustment: this method for collision avoidance relies on the fact that the speed of the mission can be easily adjusted using the parameter $\dot{\gamma}_d(t)$ given in the description of the time-coordination problem in Equation (22). For this reason, the collision avoidance algorithm presented in this section fits in well with the cooperative path-following algorithm described previously. By slowing down or speeding up the overall mission, not only collision avoidance with external objects is achieved, but also inter-vehicle safety is automatically guaranteed. Moreover, as long as only relative temporal constraints are enforced, the vehicles can safely proceed along their paths without violating the mission requirements. On the other hand, note that in the case where more than one collision avoidance maneuver is required at the same time, the UAVs must agree and adopt the same solution. This extension can be implemented by combining the collision avoidance algorithm with some simple logic. Moreover, in the case where two or more UAVs have conflicting constraints (for example, one UAV can only accelerate while another can only decelerate), then collision avoidance through speed adjustment is not possible, and the vehicles must adopt an alternative solution (such as trajectory replanning).

Then, the control law for $\ddot{\gamma}_d(t)$, used in Equation (24), is given as follows:

$$\ddot{\gamma}_d(t) = k_1(1-\bar{s})(1-\dot{\gamma}_d(t)) - \ddot{\gamma}_{\max}\bar{s}\ \text{sign}((\gamma_{\text{col}} - \gamma(t)) - (t_{\text{col}} - t))\,, \tag{30}$$

where k_1 is a positive control gain, $\bar{s} = 1$ whenever speed-adjustment method is found to be feasible and preferable and spatial collision conditions given in Equations (27), (28) and (29) are satisfied. Otherwise, $\bar{s} = 0$.

Then, as shown in [29], the UAV avoids collision with the obstacle, thus ensuring that the temporal separation between the UAV and the obstacle at the collision point is always

greater than the desired minimum time separation t_{sep}. Moreover, in [29] the authors also show that the collision avoidance solution accounts for the presence of time-coordination errors which arise from the use of autopilots with non-ideal tracking performance. Notice that, if $(\gamma_{\text{col}} - \gamma(t)) > (t_{\text{col}} - t)$, then the obstacle is expected to arrive at the collision point before the vehicle does. Accordingly, $\ddot{\gamma}_d(t) = -\ddot{\gamma}_{\text{max}}$; therefore, the UAV slows down rather than speeding up. Analogous arguments can be made for the case where $(\gamma_{\text{col}} - \gamma(t)) < (t_{\text{col}} - t)$, and $\ddot{\gamma}_d(t) = \ddot{\gamma}_{\text{max}}$.

It is important to observe that while conditions such as (27) and (29) are only useful to decide if the vehicle must start a collision avoidance maneuver, condition (28) determines how long this maneuver lasts. In fact, when $|(\gamma_{\text{col}} - \gamma(t)) - (t_{\text{col}} - t)| > t_{\text{sep}}$, which implies (28) is not satisfied anymore, then according to (30) the vehicle can proceed with rate of progression $\dot{\gamma} = 1$, while satisfying the time-separation requirement.

Collision avoidance through trajectory replanning: if the vehicle under consideration predicts a collision by verifying that conditions (27) and (28) are met, but collision through speed-adjustment is not feasible or preferable, then the UAV is required to replan its trajectory. Unlike collision avoidance through speed adjustment, this method of avoidance only affects the mission for the UAV at a collision course; the rest of the UAVs continue their mission without any change. This feature ensures that extra communication costs required for the exchange of trajectory information is not incurred. Despite this lack of information exchange, however, it is ensured that the trajectories are still deconflicted.

The procedure for replanning is outlined in the following. The key idea behind the algorithm is to exploit the geometric properties of the Bézier curves to add a detour to the

original trajectory thereby obtaining a new trajectory which (i) avoids the obstacle, (ii) does not violate the dynamic constraints of the vehicle, mission requirements, and minimum inter-vehicle separation, and (iii) ensures continuity of the trajectory and its derivatives. The detailed step-by-step procedure for collision avoidance algorithm through trajectory replanning can be found in [49]. Here, these steps are listed as follows:

First, the trajectory replanning control block approximates the obstacle's predicted trajectory with Bézier curves. Second, assuming that the mission progression rate has converged to $\dot{\gamma} = 1$, the algorithm then finds the Bézier curves that describe the separation vector between the vehicle and the obstacle as a function of time,

$$\mathbf{d}(t_d) = \mathbf{p_d}(t_d) - \mathbf{p}_o(t_d). \tag{31}$$

For illustrative purposes, a two-dimensional example of separation curve is given in Figure 6, where the red circle represents the collision area, the radius of which is the minimum safety distance, d_{safe}. Third, the collision avoidance algorithm finds a new trajectory $\mathbf{p}_{d,\text{new}}(t_d)$ by adding a detour to $\mathbf{p_d}(t_d)$ such that the separation curve moves out of the collision area. For illustration purposes, the reader is referred to the following video `www.youtube.com/watch?v=ALZQcEK8f_Q`.

The detour is found in such a way that the new trajectory $\mathbf{p}_{d,\text{new}}$, with velocity vector $\mathbf{v}_{d,\text{new}}$ and acceleration vector $\mathbf{a}_{d,\text{new}}$, avoids the collision while keeping the current position, velocity, and acceleration unchanged, satisfies the boundary conditions while meeting the following requirements

$$||\mathbf{p}_{d,\text{new}}(t_d) - \mathbf{p_d}(t_d)|| < \Delta_p, \quad ||\mathbf{v}_{d,\text{new}}(t_d) - \mathbf{v_d}(t_d)|| < \Delta_v, \quad ||\mathbf{a}_{d,\text{new}}(t_d) - \mathbf{a_d}(t_d)|| < \Delta_a,$$

where Δ_p, Δ_v and Δ_a can be computed a *p*riori given some assumptions on the obstacle size

and speed.

Notice that the bounds Δ_p, Δ_v and Δ_a characterize the deviation in position, velocity, and acceleration of the new trajectory from the original one. Through a priori computation of these bounds, inter-vehicle safety can be guaranteed. For example, if the vehicles are required to maintain a safety distance of E from each other, then the inter-vehicle distance can be set to be $E_d \geq E + \Delta_p$ during the initial trajectory generation. The vehicles are then guaranteed to maintain a distance of at least E from each other even during collision avoidance. A similar argument applies for satisfaction of vehicle dynamic constraints. The reader is referred to [31], [49] for detailed analysis and proofs.

Simulation Results

In this section, the multi-vehicle road search motivational example is implemented to illustrate the performance of the cooperative framework presented in this article. The example is used to demonstrate that with each of the components described previously, the general framework allows for the execution of the cooperative mission. To briefly summarize, the objective is to inspect an area by three cooperating multirotors. The multirotors have to transit from their holding area to the area of interest, which is the *transition phase* and, subsequently, commence the inspection of a designated road. It is important that the multirotors simultaneously arrive at their destinations during the transition phase, as the search is executed in a *parallel sweep search pattern* along the road. Needless to say, that the roach search has to be executed in a coordinated way in order to maintain the overlap of fields of view of their onboard cameras. Therefore, coordination between the vehicles is not only critical from a safety point of view, it

is also essential to accomplish the mission at hand. During the road search phase, the vehicles encounter obstacles. However, it is shown that the mission objectives are not compromised due to the online collision avoidance algorithm.

The mission scenario is implemented in Matlab/Simulink®. As mentioned earlier, the emphasis of this article is not on the performance of an inner-loop controller, but rather it is considered as given with certain performance specifications. However, for the sake of providing a realistic simulation scenario, 6DOF models representing the dynamics of the multirotors have been implemented; each of the multirotor is equipped with a simple inner-loop autopilot for angular rate command tracking; additionally, measurement noise and transmission delays have been added.

Figure 7 shows the complete execution of the mission, from transition phase to the completion of the actual road search. The three-dimensional flight paths are presented in Figure 7a, while the separations between each pair of vehicles are shown in Figure 7b. It can be seen that the vehicles maintain a minimum distance of $E = 1$ m at all times. The errors between the commanded path and the actual flight path of the multirotors are given in Figure 7c. The implemented path-following algorithm allows close tracking of the commanded paths, while the dynamic constraints of the vehicles (see Figures 7d and 7e) are not exceeded throughout the entire mission. The values for $v_{i,\max}$ and $a_{i,\max}$ are given in Table I.

As explained previously, coordination between the vehicles is achieved if the errors between the coordination variables $\gamma_i(t)$ are zero. This coordination error is plotted in Figure 8a and it can be concluded that the vehicles are coordinated throughout the entire mission. Lastly, Figure 8b presents the time histories of the rates of progression $\dot{\gamma}_i(t)$ and the commanded rate of

progression $\dot{\gamma}_d(t)$. It can be seen that the $\dot{\gamma}_i(t)$ track the command closely. Note that the desired rate of progression is equal to 1 throughout the mission, except in one occasion where speed adjustment of the vehicles is required to avoid a moving obstacle. This case is discussed in more detail below. Three distinct events during the mission occur that require a separate presentation of the results: the *transition phase*, *avoidance of a moving obstacle*, and *avoidance of a static obstacle*.

Transition phase: the results regarding the transition phase have already been discussed in the section on time-coordination and, hence, they are not repeated here. However, note that in the simulation shown in this section, the initial errors between the commanded path and the actual position of the vehicles are of smaller value. Therefore, the path-following and time-coordination errors are less pronounced but more realistic.

Moving obstacle: as discussed earlier, for coordinated missions it is preferable to avoid a collision by changing the mission progression rate. To demonstrate this capability, a moving obstacle in a collision course with UAV_2 in introduced. To avoid this obstacle, the mission progression rate is increased by the collision avoidance algorithm. This avoidance maneuver is illustrated in Figure 9. Figure 9a shows the distance of the obstacle from all three UAVs with respect to time. It can be seen that the vehicles maintain a distance of more than 1 m from the obstacle. In the same figure, the distance of UAV_2 from obstacle is plotted in the case where the collision avoidance maneuver is not executed, thus illustrating where the collision would happen. To demonstrate regular time coordination between vehicles during the avoidance maneuver, the time-coordination error is presented in Figure 9b. Finally, the increase in the mission progression rate for the avoidance maneuver is shown in Figure 9c, where $\dot{\gamma}_i(t)$ for all vehicles along with $\dot{\gamma}_d(t)$ take values greater than 1.

Static obstacle: Next, consider a static obstacle in the path of UAV_3. Since the obstacle is static, collision avoidance through speed adjustment is infeasible and, therefore, collision avoidance through trajectory replanning is used. Figure 10a shows the distance of the vehicle from the obstacle with and without collision avoidance. Note that the obstacle is not detected by UAV_1 and UAV_2. It can be seen that the vehicle avoids the collision by maintaining a distance of more than 1 m. The replanned trajectory can be seen against the original one in Figure 10b where the small detour added to the trajectory is visible. Lastly, the coordination error and the mission progression rate $\dot{\gamma}_i(t)$ for the UAVs are plotted in Figures 10c and 10d, respectively, thus demonstrating that the time-coordination between the vehicles remains unaffected.

Conclusion

In this article a safe multi-vehicle control framework has been presented, which allows a fleet of multirotor UAVs to follow deconflicted desired trajectories, coordinate along them in order to satisfy relative temporal constraints, while avoiding collision with moving and static obstacles. The described methodology is based on three key results. First, a centralized cooperative trajectory-generation algorithm produces a set of spatially deconflicted paths together with a set of desired speed profiles. These trajectories guarantee inter-vehicle safety, while satisfying specific temporal mission requirements, as well as dynamic constraints of the vehicles. Then, a distributed cooperative path-following controller ensures that the vehicles follow the trajectories while coordinating along them in order to arrive at the final destination at the same time, or with a predefined temporal separation, according to the mission requirements. The cooperative control architecture relies on the presence of an inner-loop autopilot and an outer-loop path-following

controller which guarantee that the distance between the vehicle and its desired position along the path remains bounded throughout the mission. Then, the speed of each vehicle is indirectly adjusted in order to satisfy the temporal constraints and achieve coordination. Finally, collision avoidance control laws are formulated to guarantee safety of the overall fleet of vehicles even in the presence of unpredicted obstacles. The described solution uses the speed of the vehicles to control the progression along the path in order to avert a possible collision with a moving obstacle. Moreover, a fast onboard trajectory replanning solution is presented to allow the vehicles to deviate from the original trajectory whenever collision avoidance through speed adjustment is not possible. The described approach borrows concepts and tools from a broad spectrum of disciplines, leading to a simple design procedure based on self-contained control blocks. A specific example, consisting of a cooperative road search mission, was discussed throughout the article to describe the developed framework. The benefits of this approach can be extended to cooperative control of multiple heterogeneous robots, such as autonomous marine vehicles and fixed-wing UAVs, or ground robots and multirotor UAVs, which is the subject of ongoing research.

References

[1] A. Bachrach, S. Prentice, R. He, and N. Roy, “RANGE–robust autonomous navigation in GPS-denied environments,” *Journal of Field Robotics*, vol. 28, no. 5, pp. 644–666, 2011.

[2] S. Shen, N. Michael, and V. Kumar, “Autonomous multi-floor indoor navigation with a computationally constrained MAV,” in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, People’s Republic of China, May 2011, pp. 20–25.

[3] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys, “Pixhawk: A system for autonomous flight using onboard computer vision,” in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, People’s Republic of China, May 2011, pp. 2992–2997.

[4] S. K. Phang, C. Cai, B. M. Chen, and T. H. Lee, “Design and mathematical modeling of a 4-standard-propeller (4SP) quadrotor,” in *2012 10th World Congress on Intelligent Control and Automation (WCICA)*, Beijing, People’s Republic of China, July 2012, pp. 3270–3275.

[5] S. Bouabdallah, P. Murrieri, and R. Siegwart, “Design and control of an indoor micro quadrotor,” in *2004 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 5, Barcelona, Spain, April 2004, pp. 4393–4398.

[6] M. Mesbahi and F. Y. Hadaegh, “Formation flying control of multiple spacecraft via graphs, matrix inequalities, and switching,” *Journal of Guidance, Control and Dynamics*, vol. 24, no. 2, pp. 369–377, March–April 2001.

[7] M. Saska, T. Krajník, V. Vonásek, Z. Kasl, V. Spurnỳ, and L. Přeučil, “Fault-tolerant formation driving mechanism designed for heterogeneous MAVs-UGVs groups,” *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1, pp. 603–622, 2014.

[8] Y. D. Song, Y. Li, and X. H. Liao, “Orthogonal transformation based robust adaptive close formation control of multi-UAVs,” in *American Control Conference*, vol. 5, Portland, OR, June 2005, pp. 2983–2988.

[9] D. M. Stipanović, G. İnalhan, R. Teo, and C. J. Tomlin, “Decentralized overlapping control of a formation of unmanned aerial vehicles,” *Automatica*, vol. 40, no. 8, pp. 1285–1296, August 2004.

[10] R. Ghabcheloo, A. M. Pascoal, C. Silvestre, and I. Kaminer, “Coordinated path following control of multiple wheeled robots using linearization techniques,” *International Journal of Systems Science*, vol. 37, no. 6, pp. 399–414, May 2006.

[11] B. P. Gerkey and M. J. Matarić, “A formal analysis and taxonomy of task allocation in multi-robot systems,” *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[12] M. Turpin, N. Michael, and V. Kumar, “Concurrent assignment and planning of trajectories for large teams of interchangeable robots,” in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013, pp. 842–848.

[13] N. Michael, J. Fink, and V. Kumar, “Cooperative manipulation and transportation with aerial robots,” *Autonomous Robots*, vol. 30, no. 1, pp. 73–86, 2011.

[14] U. Gurcuoglu, G. A. Puerto-Souza, F. Morbidi, and G. L. Mariottini, “Hierarchical control of a team of quadrotors for cooperative active target tracking,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, November 2013, pp. 5730–5735.

[15] J. Capitan, M. T. Spaan, L. Merino, and A. Ollero, “Decentralized multi-robot cooperation with auctioned POMDPs,” *The International Journal of Robotics Research*, vol. 32, no. 6,

pp. 650–671, 2013.

[16] V. Gazi, “Swarm aggregations using artificial potentials and sliding-mode control,” *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1208–1214, 2005.

[17] H. Tanner, A. Jadbabaie, and G. Pappas, “Flocking in fixed and switching networks,” *IEEE Transactions on Automatic Control*, vol. 52, no. 5, pp. 863–868, 2007.

[18] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.

[19] A. Chakravarthy and D. Ghose, “Generalization of the collision cone approach for motion safety in 3-D environments,” *Autonomous Robots*, vol. 32, no. 3, pp. 243–266, April 2012.

[20] J. van den Berg, M. Lin, and D. Manocha, “Reciprocal velocity obstacles for real-time multi-agent navigation,” in *2008 IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, CA, May 2008, pp. 1928–1935.

[21] K. Listmann, M. Masalawala, and J. Adamy, “Consensus for formation control of nonholonomic mobile robots,” in *2009 IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009, pp. 3886–3891.

[22] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal, “Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields,” in *8th IEEE-RAS International Conference on Humanoid Robots*, Daejeon, South Korea, December 2008, pp. 91–98.

[23] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *1985 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, St. Louis, MO, March 1985, pp. 500–505.

[24] V. Desaraju and J. How, “Decentralized path planning for multi-agent teams in complex environments using rapidly-exploring random trees,” in *2011 IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, People’s Republic of China, May 2011, pp. 4956–4961.

[25] S. M. Lavalle, “Rapidly-exploring random trees: A new tool for path planning,” Computer Science Dept., Iowa State University, Tech. Rep. TR 98-11, October 1998.

[26] R. Choe, V. Cichella, E. Xargay, N. Hovakimyan, A. C. Trujillo, and I. Kaminer, “A Trajectory-Generation Framework for Time-Critical Cooperative Missions,” in *AIAA Infotech@Aerospace*, Boston, MA, August 2013, AIAA 2013-4582.

[27] V. Cichella, I. Kaminer, E. Xargay, V. Dobrokhodov, N. Hovakimyan, A. Aguiar, and A. António M. Pascoal, “A Lyapunov-based approach for Time-Coordinated 3D Path-Following of multiple quadrotors,” in *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, Maui, HI, 2012, pp. 1776–1781.

[28] V. Cichella, R. Choe, S. B. Mehdi, E. Xargay, N. Hovakimyan, I. Kaminer, and V. Dobrokhodov, “A 3D Path-Following Approach for a Multirotor UAV on SO(3),” in *2nd IFAC Workshop on Research, Education and Development of Unmanned Aerial Systems (2013)*, Compiegne, France, November 2013.

[29] V. Cichella, R. Choe, B. S. Mehdi, E. Xargay, N. Hovakimyan, V. Dobrokhodov, and I. Kaminer, “Trajectory generation and collision avoidance for safe operation of cooperating UAVs,” in *AIAA Guidance, Navigation and Control Conference*, National Harbor, MD, January 2014, AIAA 2014-0972.

[30] V. Cichella, I. Kaminer, V. Dobrokhodov, E. Xargay, R. Choe, N. Hovakimyan, A. P. Aguiar, and A. M. Pascoal, “Cooperative Path-Following of Multiple Multirotors over Time-Varying

Networks," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 945–957, July 2015.

[31] S. B. Mehdi, R. Choe, V. Cichella, and N. Hovakimyan, "Collision avoidance through path replanning using Bézier curves," in *AIAA Guidance, Navigation and Control Conference*, Kissimmee, FL, January 2015.

[32] R. Choe, J. Puig-Navarro, V. Cichella, E. Xargay, and N. Hovakimyan, "Trajectory generation using spatial Pythagorean Hodograph Bézier curves," in *AIAA Guidance, Navigation and Control Conference*, Kissimmee, FL, January 2015.

[33] I. Kaminer, E. Xargay, V. Cichella, N. Hovakimyan, A. M. Pascoal, A. P. Aguiar, V. Dobrokhodov, and R. Ghabcheloo, "Time-critical cooperative path following of multiple UAVs: Case studies," in *Advances in Estimation, Navigation, and Spacecraft Control*. Springer, 2015, pp. 209–233.

[34] V. T. Taranenko, *Experience on Application of Ritz's, Poincare's, and Lyapunov's Methods for Solving Flight Dyanmics Problems*. Moscow: Air Force Engineering Academy Press, 1968, (in Russian).

[35] O. A. Yakimenko, "Direct method for rapid prototyping of near-optimal aircraft trajectories," *Journal of Guidance, Control and Dynamics*, vol. 23, no. 5, pp. 865–875, September–October 2000.

[36] J.-W. Chang, Y.-K. Choi, M.-S. Kim, and W. Wang, "Computation of the minimum distance between two Bézier curves/surfaces," *Computers & Graphics*, vol. 35, no. 3, pp. 677–684, June 2011.

[37] M. A. Shah and N. Aouf, "3D cooperative Pythagorean Hodograph path planning and obstacle avoidance for multiple UAVs," in *IEEE International Conference on Cybernetic*

Intelligence Systems, Reading, UK, September 2010.

[38] A. Tsourdos, B. White, and M. Shanmugavel, *Cooperative Path Planning of Unmanned Aerial Vehicles*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc, 2011.

[39] M. I. Lizarraga and G. H. Elkaim, "Spatially deconflicted path generation for multiple UAVs in a bounded airspace," in *IEEE/ION Position, Location and Navigation Symposium*, Monterey, CA, May 2008, pp. 1213–1218.

[40] R. T. Farouki, *Pythagorean-Hodograph Curves*. Berlin Heidelberg: Springer-Verlag, 2008.

[41] R. T. Farouki and T. Sakkalis, "Pythagorean hodographs," *IBM Journal of Research and Development*, vol. 34, no. 5, pp. 736–752, September 1990.

[42] F. Augugliaro, A. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal, October 2012, pp. 1917–1922.

[43] Y. Chen, M. Cutler, and J. P. How, "Decoupled multiagent path planning via incremental sequential convex programming," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, May 2015, pp. 5954–5961.

[44] E. Polak, *Optimization: Algorithms and Consistent Approximations*. New York, NY: Springer-Verlag, 1997.

[45] N. Biggs, *Algebraic Graph Theory*. New York, NY: Cambridge University Press, 1993.

[46] M. Arcak, "Passivity as a design tool for group coordination," *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1380–1390, August 2007.

[47] Advanced Controls Research Laboratory at the University of Illinois at Urbana Champaign,

"Quadrotor UAVs," naira.mechse.illinois.edu/quadrotor-uavs/, .

[48] P. Conroy, D. Bareiss, M. Beall, and J. v. d. Berg, "3-D reciprocal collision avoidance on physical quadrotor helicopters with on-board sensing for relative positioning," *arXiv preprint arXiv:1411.3794*, 2014.

[49] S. B. Mehdi, R. Choe, and N. Hovakimyan, "Collision avoidance through path replanning using Bézier curves," in *IEEE Conference on Decision and Control*, Osaka, Japan, December 2015, accepted.

[50] P. Encarnação and A. M. Pascoal, "Combined trajectory tracking and path following: An application to the coordinated control of autonomous marine craft," in *IEEE Conference on Decision and Control*, Orlando, FL, December 2001, pp. 964–969.

[51] L. Lapierre, D. Soetanto, and A. M. Pascoal, "Coordinated motion control of marine robots," in *IFAC Conference on Manoeuvering and Control of Marine Craft*, Girona, Spain, September 2003.

[52] R. Skjetne, I.-A. F. Ihle, and T. I. Fossen, "Formation control by synchronizing multiple maneuvering systems," in *IFAC Conference on Manoeuvering and Control of Marine Craft*, Girona, Spain, September 2003.

[53] R. Skjetne, S. Moi, and T. I. Fossen, "Nonlinear formation control of marine craft," in *IEEE Conference on Decision and Control*, vol. 2, Las Vegas, NV, December 2002, pp. 1699–1704.

[54] R. Hindman and J. Hauser, "Maneuver modified trajectory tracking," in *International Symposium on the Mathematical Theory of Networks and Systems*, St. Louis, MO, June 1996.

[55] D. Soetanto, L. Lapierre, and A. M. Pascoal, "Adaptive, non-singular path-following control

of dynamic wheeled robots," in *International Conference on Advanced Robotics*, Coimbra, Portugal, June–July 2003, pp. 1387–1392.

[56] R. Skjetne, A. R. Teel, and P. V. Kokotović, "Stabilization of sets parameterized by a single variable: Application to ship maneuvering," in *Proceedings of the 15th International Symposium on Mathematical Theory of Networks and Systems*, Notre Dame, IN, August 2002.

[57] I.-A. F. Ihle, "Coordinated control of marine craft," Ph.D. dissertation, Norwegian University of Science and Technology, Trondheim, Norway, September 2006.

[58] R. Ghabcheloo, A. P. Aguiar, A. M. Pascoal, C. Silvestre, I. Kaminer, and J. P. Hespanha, "Coordinated path-following in the presence of communication losses and delays," *SIAM Journal on Control and Optimization*, vol. 48, no. 1, pp. 234–265, 2009.

[59] R. Ghabcheloo, "Coordinated path following of multiple autonomous vehicles," Ph.D. dissertation, Instituto Superior Técnico, Lisbon, Portugal, May 2007.

[60] I. Kaminer, O. A. Yakimenko, A. M. Pascoal, and R. Ghabcheloo, "Path generation, path following and coordinated control for time-critical missions of multiple UAVs," in *American Control Conference*, Mineapolis, MN, June 2006, pp. 4906–4913.

[61] E. Xargay, "Time-critical cooperative path-following control of multiple unmanned aerial vehicles," Ph.D. dissertation, University of Illinois at Urbana-Champaign, Urbana, IL, United States, May 2013.

[62] A. P. Aguiar and A. M. Pascoal, "Coordinated path-following control for nonlinear systems with logic-based communication," in *IEEE Conference on Decision and Control*, New Orleans, LA, December 2007, pp. 1473–1479.

[63] Y. Xu and J. P. Hespanha, "Communication logic design and analysis for networked control

systems," in *Current Trends in Nonlinear Systems and Control*, ser. Systems and Control: Foundations & Applications, L. Menini, L. Zaccarian, and C. T. Abdallah, Eds. Cambridge, MA: Birkhäuser Boston, 2006, pp. 495–514.

[64] J. K. Yook, D. M. Tilbury, and N. R. Soparkar, "Trading computation and bandwidth: Reducing communication in distributed control systems using state estimators," *IEEE Transactions on Control System Technology*, vol. 10, no. 4, pp. 503–518, July 2002.

[65] E. Kyrkjebø and K. Y. Pettersen, "Ship replenishment using synchronization control," in *IFAC Conference on Manoeuvering and Control of Marine Craft*, Girona, Spain, September 2003.

[66] E. Kyrkjebø, M. Wondergem, K. Y. Pettersen, and H. Nijmeijer, "Experimental results on synchronization control of ship rendezvous operations," in *IFAC Conference on Control Applications in Marine Systems*, Ancona, Italy, July 2004, pp. 453–458.

[67] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 947–951, December 2001.

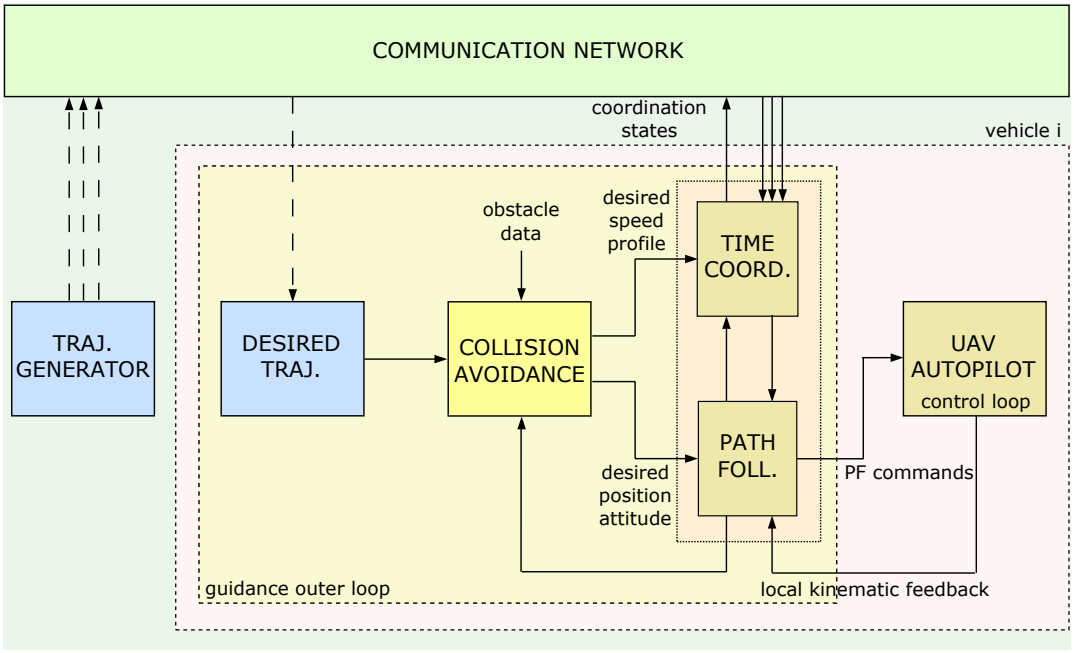
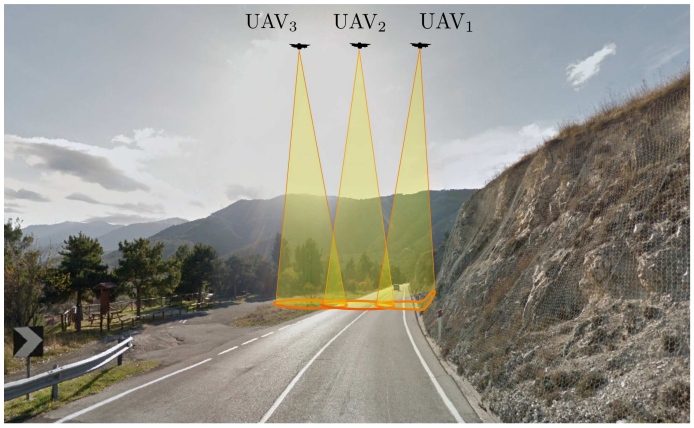
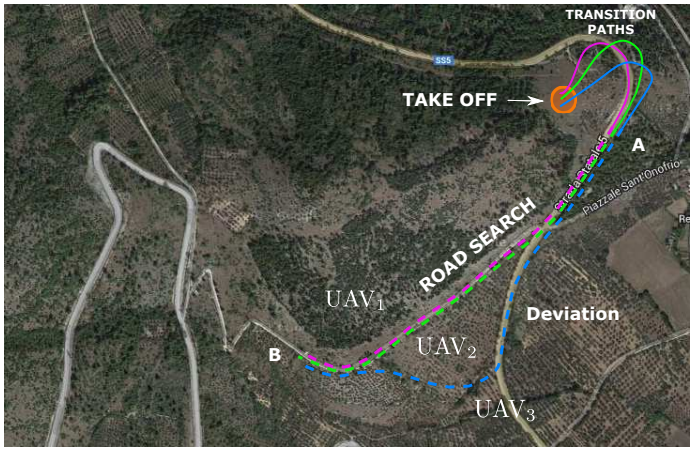


Figure 1: General framework: the trajectory-generation, cooperative path-following, and collision avoidance blocks interact to ensure safe execution of cooperative missions. The trajectory-generation algorithm computes a geometric path and a speed profile which are sent to the collision avoidance module. The collision avoidance block checks if an imminent collision with a detected obstacle is going to take place. In this case, it modifies the path or the speed profile, and sends them to the cooperative path-following algorithm. Otherwise, the desired trajectory is left unchanged and sent to the cooperative control module. The cooperative path-following control block, comprised of path-following and time-coordination algorithms, allows the UAV to execute the cooperative mission.

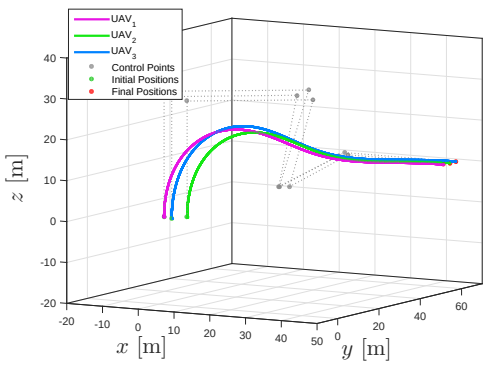


(a) *Google Maps*. 3D view of the mission scenario.

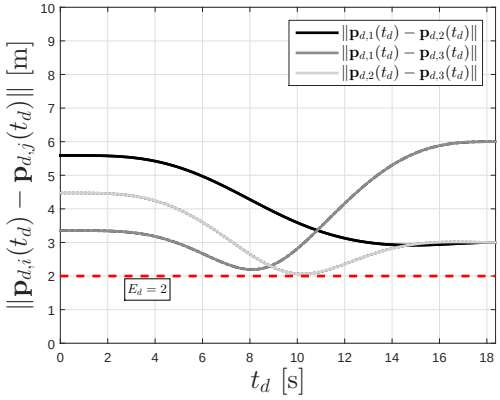


(b) *Google Maps*. 2D projection of the mission scenario.

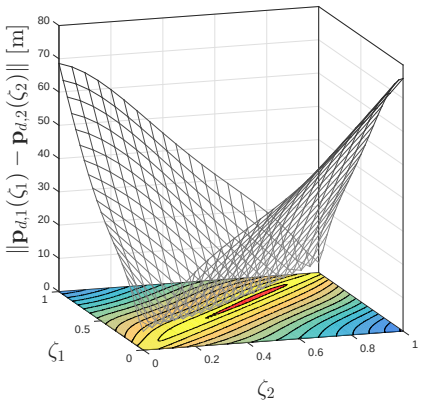
Figure 2: Cooperative road search. The figures illustrate a scenario in which cooperation among the multirotors is required to accomplish the task at hand. The UAVs, starting from random initial positions, follow the transition paths, depicted as solid lines, and arrive at point A. Then, they proceed along the road search paths, represented by solid lines, while coordinating with each other to accomplish the cooperative road search mission. Cooperation along the road search paths guarantees non-zero intersection between the fields of view of the cameras.



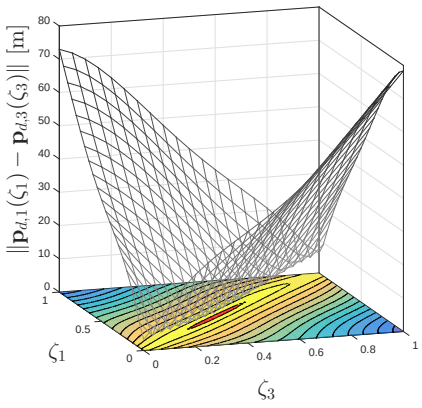
(a) Three-dimensional flight paths.



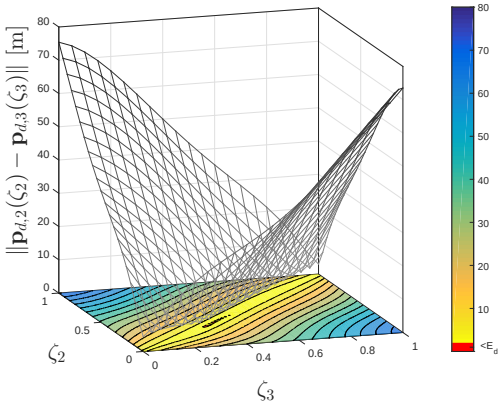
(b) Vehicle separation.



(c) UAV 1 and 2.

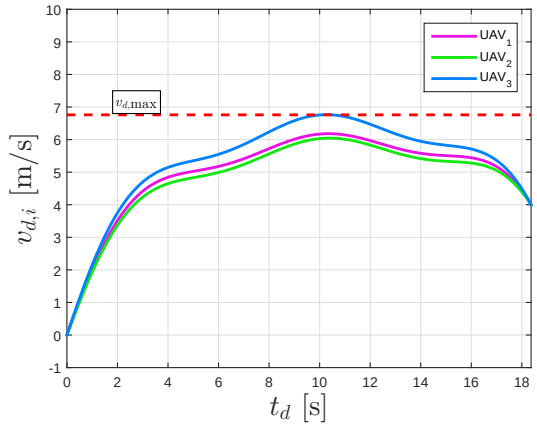


(d) UAV 1 and 3.

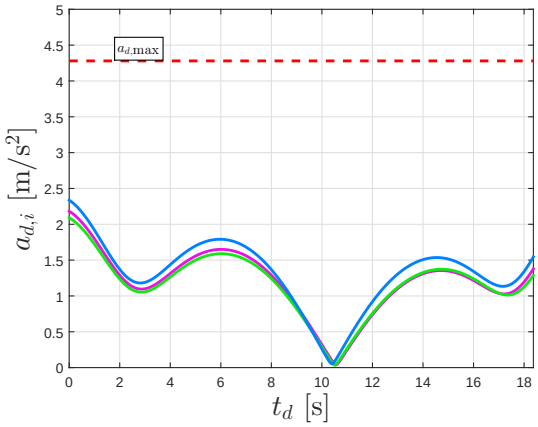


(e) UAV 2 and 3.

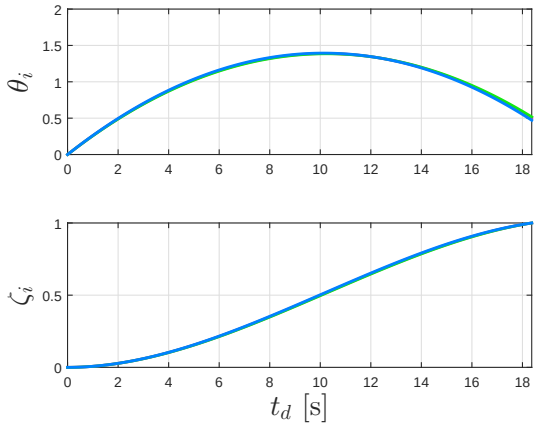
Figure 3: Three-dimensional trajectories for a team of three cooperating UAVs. Spatial deconfliction is ensured through temporal separation. The computation time is 12.4 s.



(a) Speed profile $v_{d,i}(t_d)$.

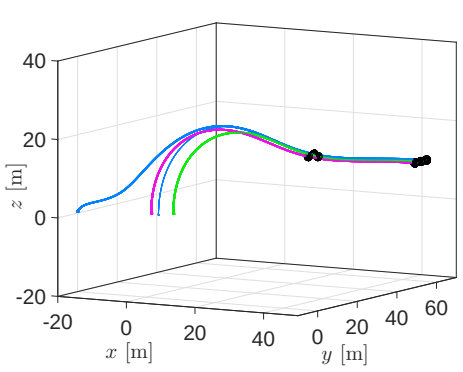


(b) Acceleration profile $a_{d,i}(t_d)$.

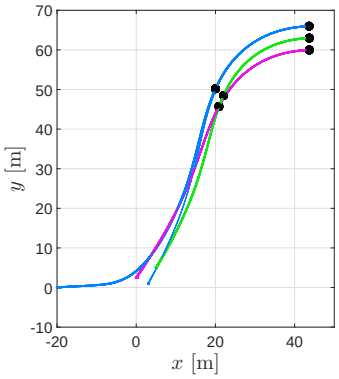


(c) Timing law $\theta_i(t_d)$ and the function $\zeta_i(t_d)$.

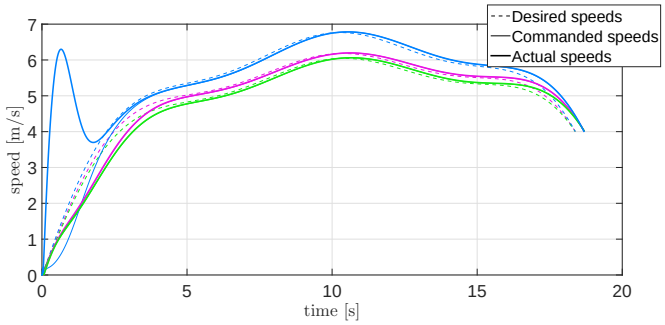
Figure 4: Dynamic constraints and timing laws for a team of three cooperating UAVs. Spatial deconfliction is ensured through temporal separation.



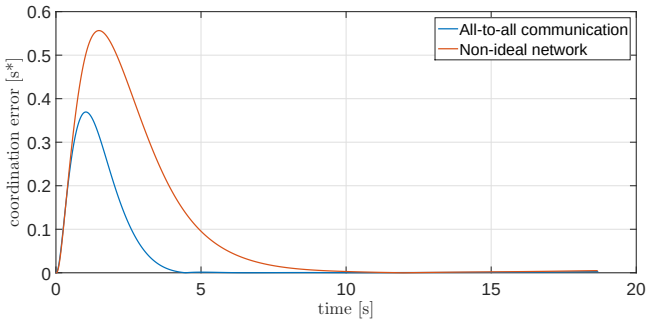
(a) 3D view.



(b) 2D projection.

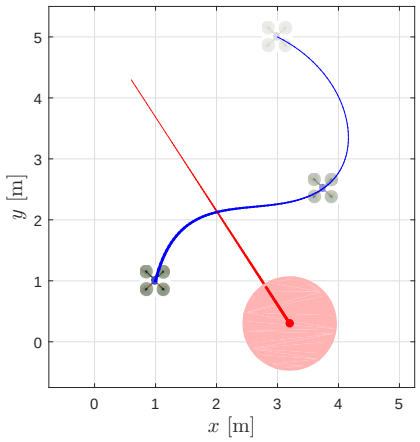


(c) Speed profiles.

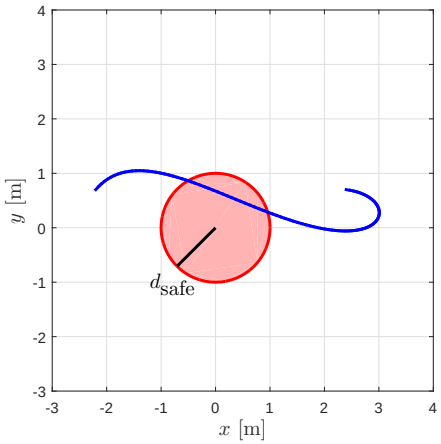


(d) Coordination error.

Figure 5: Time coordination: simulation results. UAV_1, UAV_2 and UAV_3 coordinate along the transition paths, and arrive at the final destination at the same time with desired speed profiles.

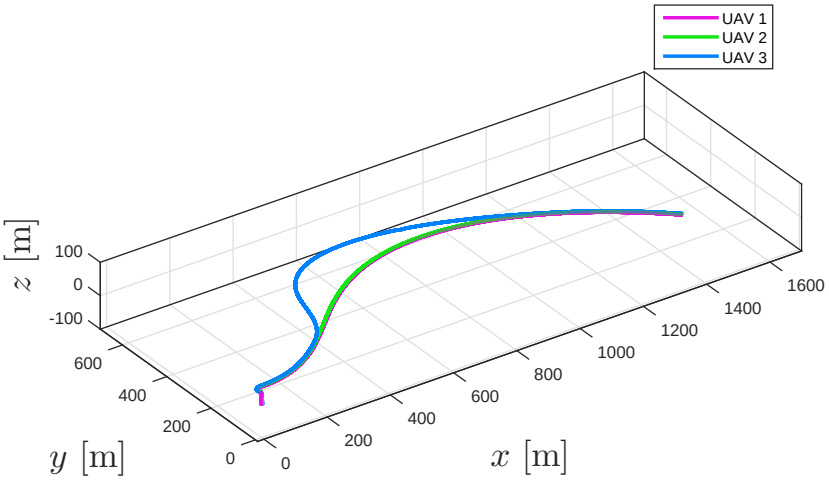


(a) Obstacle's trajectory $\mathbf{p}_o(t_d)$ (shown in red) and vehicle's trajectory $\mathbf{p}(t_d)$ (shown in blue).

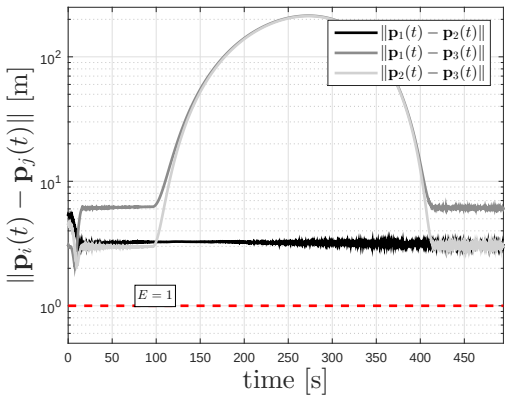


(b) Separation Bézier curve $\mathbf{d}(t_d)$ shown against a circle around origin.

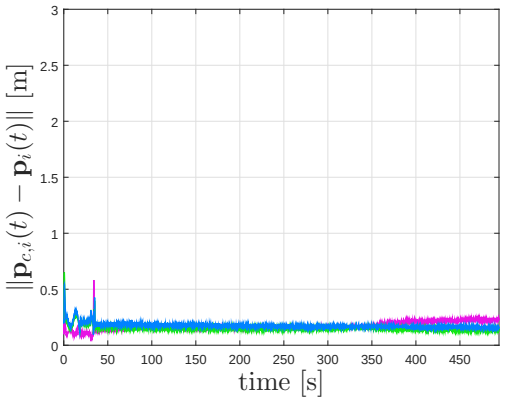
Figure 6: Demonstration of the separation curve: The top figure shows the trajectory of the vehicle and the quadrotor in blue and red, respectively. It is clear that their trajectories overlap, however, it is hard to tell if a collision occurs. In the bottom figure, the separation curve is shown which enters the circle with radius d_{safe}, thus it is clear that a collision does occur.



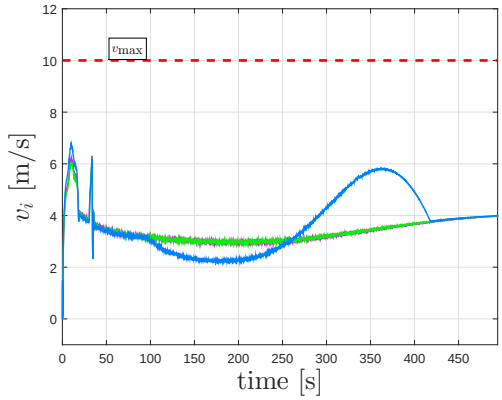
(a) Three-dimensional flight paths.



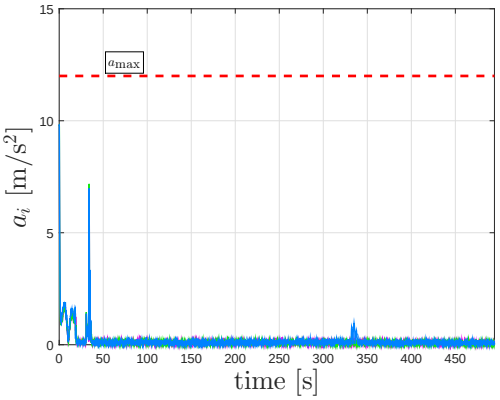
(b) Vehicle separation.



(c) Path-Following Error.

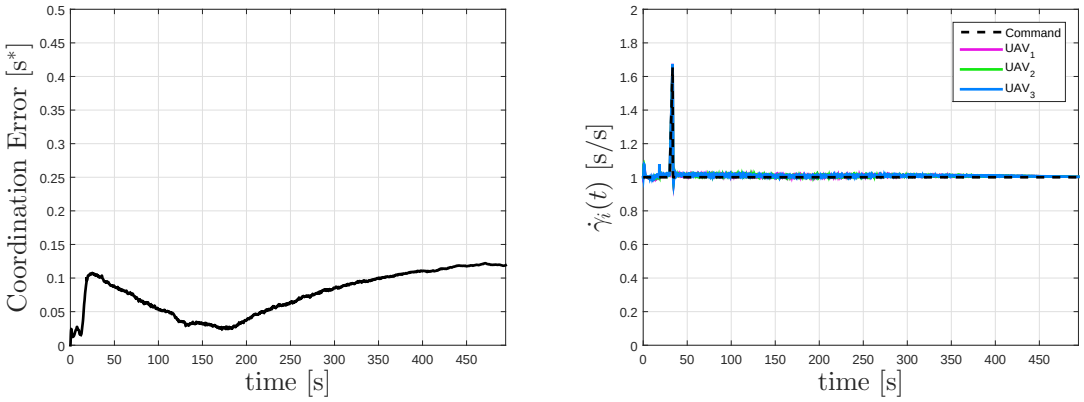


(d) Speed profiles.



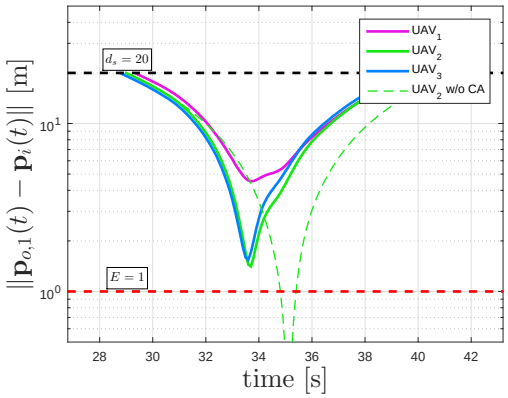
(e) Acceleration profiles.

Figure 7: Road search mission execution by team of three cooperating UAVs, including the transition phase from holding area to area of interest.

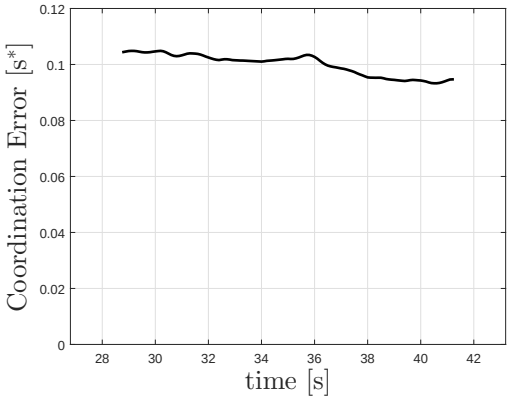


(a) Coordination Error. (b) $\dot{\gamma}_i(t)$.

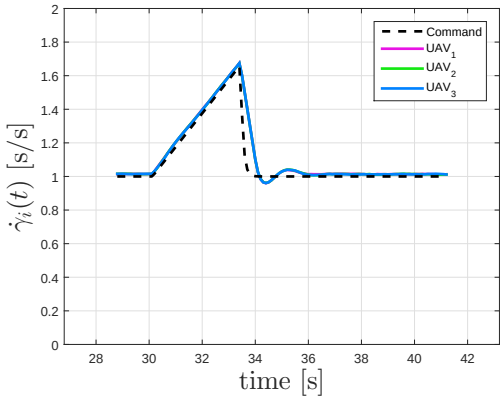
Figure 8: Time-coordination errors and rate of progression of the vehicles during the mission.



(a) Distance of all vehicles to the moving obstacle.

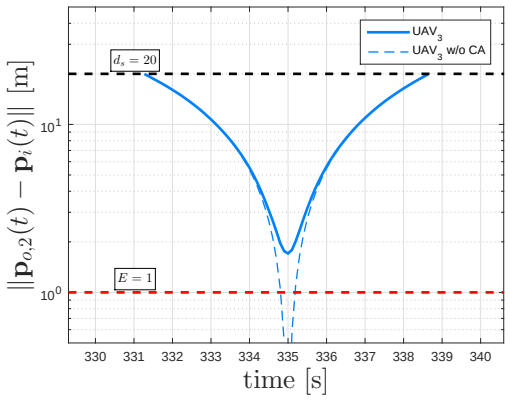


(b) Coordination Error.

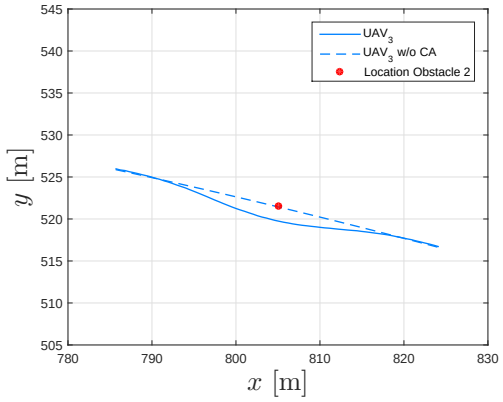


(c) Desired mission progression rate $\dot{\gamma}_d(t)$ along with mission progression rate for all UAVs $\dot{\gamma}_i(t) \quad i = \{1, 2, 3\}$. The three lines overlap.

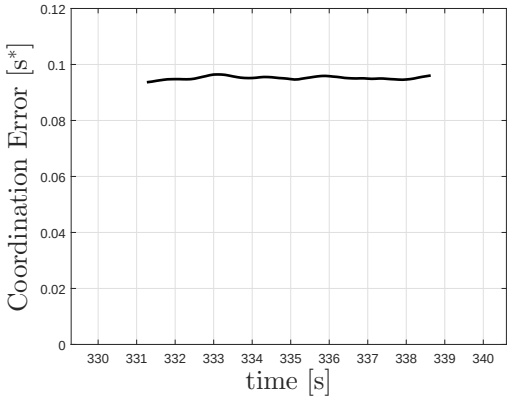
Figure 9: Demonstration of collision avoidance using speed adjustment method.



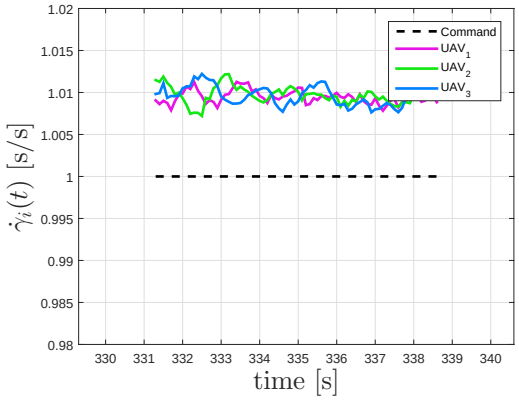
(a) Distance of UAV_3 to the static obstacle.



(b) Two-Dimensional Flight Path of UAV_3.



(c) Coordination Error.



(d) Desired mission progression rate $\dot{\gamma}_d(t)$ along with mission progression rate for all UAVs $\dot{\gamma}_i(t) \quad i = \{1, 2, 3\}$.

Figure 10: Demonstration of collision avoidance through trajectory replanning.

TABLE I: Flight conditions and dynamic constraints of the UAVs.

	UAV 1	UAV 2	UAV 3
$\mathbf{p}_d^{\mathrm{i}}$ [m]	$(0, 2.50, 1.00)$	$(5.00, 5.00, 1.00)$	$(3.00, 1.00, 1.00)$
v_d^{i} [m/s]	0	0	0
$\mathbf{p}_d^{\mathrm{f}}$ [m]	$(45.00, 60.00, 10.00)$	$(45.00, 63.00, 10.00)$	$(45.00, 66.00, 10.00)$
v_d^{f} [m/s]	4.00	4.00	4.00
$v_{d,\max}$ [m/s]	6.76	6.76	6.76
$a_{d,\max}$ [m/s^2]	4.28	4.28	4.28
$v_{\max}$ [m/s]	10.00	10.00	10.00
$a_{\max}$ [m/s^2]	12.00	12.00	12.00

Sidebar 1: Cooperative Path Following

The problem of cooperative path following amounts to making a fleet of vehicles converge to and follow a set of desired spatial paths, while meeting pre-specified spatial and temporal constraints. Over the last decade, there has been growing interest in the problem of coordinated path-following control of fleets of autonomous vehicles, mainly for the execution of cooperative marine missions involving multiple autonomous surface and underwater vehicles. Initial work in this topic can be found in [50]–[53]. The coordinated path-following control problem was implicit in the early work in [50], where the authors built on and extended the single-vehicle "manoeuvre regulation approach in [54], and presented a solution to the problem of coordinated operation of an autonomous surface vehicle and an autonomous underwater vehicle. The strategy adopted, however, requires the vehicles to exchange a large amount of information, and cannot be easily generalized to larger teams of vehicles. These drawbacks were later overcome in [51], which presented a leader-follower cooperative approach that (almost) decouples the temporal and spatial assignments of the mission. The solution adopted is rooted in the results on path-following control of a single vehicle presented in [55], and takes advantage of the fact that, with this path-following algorithm, the speed profile of each vehicle becomes an additional degree-of-freedom that can be exploited for vehicle coordination. Moreover, in this setup, the two vehicles only need to exchange the (scalar) "along-path positions of their virtual targets, which reduces drastically the amount of information to be exchanged among vehicles when compared to the solution developed in [50]. Interestingly, an approach similar to the one in [51] was presented at approximately the same time in the work in [52] and [53], where a nonlinear control design method was presented for formation control of a fleet of ships. The approach relies on the maneuvering

methodology developed in [56], which is then combined with a centralized guidance system that adjusts the speed profile of each vehicle so as to achieve and maintain the desired formation configuration. The maneuvering strategy in [56] was also exploited in [57], where a passivity framework is used to solve the problem of vehicle coordination and formation maneuvering. In [58], the authors extended the approach in [51] and addressed the problem of steering a group of vehicles along predefined spatial paths while holding a desired (possibly time-varying) formation pattern. Using results from nonlinear systems and algebraic graph theory, conditions were derived under which the control algorithm solves the coordinated path-following control problem in the presence of switching communication topologies and network link latencies. In particular, stability of the closed-loop system was analyzed under two scenarios: first, networks with brief connectivity losses; and second, uniformly jointly connected communication graphs. Stability of coordinated path-following strategies under communication limitations was also investigated in [59]. The approach in [51] was also extended in [60], where the authors addressed the problem of coordinated control of multiple UAVs. To enforce the temporal constraints of the mission, the coordination algorithm relies on a distributed control law with a proportional-integral structure, which ensures that each vehicle travels along its path at the desired constant speed, and also provides disturbance rejection capabilities against steady winds. The work in [60] was later extended in [61] to the case of arbitrary (feasible) desired speed profiles, multiple leaders, and low-connectivity scenarios. Related work can also be found in [62], which presents a multi-vehicle control architecture aimed at reducing the frequency at which information is exchanged among vehicles by incorporating logic-based communications. To this effect, the authors borrow from and expand some of the key ideas exposed in [63] and [64], where decentralized controllers for distributed systems are derived by using, for each system, its local state information together

with estimates of the states of the systems that it communicates with. Other relevant cooperative motion-control algorithms have been presented in the literature that address problems akin to that of coordinated path following. In [65] and [66], for example, synchronization techniques are used to develop control laws for ship rendezvous maneuvers. Also, the work in [67] presents a solution to the problem of coordinated path following for multi-agent formation control. In the setup adopted, a reference path is specified for a nonphysical point of the formation, which plays the role of a virtual leader, while a desired formation pattern is defined with respect to this nonphysical point. Control laws are then derived that ensure that the real vehicles converge to the desired reference points of the formation, while the virtual leader follows the reference path.

Author Information

Venanzio Cichella (cichell2@illinois.edu) received his M.S. in Automation Engineering in 2011 from the University of Bologna. Before that he spent 9 month at TU Delft as an Erasmus student, and 1 year at the Naval Postgraduate School as a visiting scholar and research assistant. In 2011 he started working on control of autonomous vehicles at the Naval Postgraduate School. In 2012 he moved to the University of Illinois at Urbana-Champaign, where he is currently a Ph.D. candidate in the Department of Mechanical Engineering. In 2015, he received the Ross J. Martin Memorial Award from the College of Engineering at UIUC for outstanding research achievement. His research interests include cooperative control of autonomous aerial and ground robots, nonlinear systems, robust and adaptive control, and human-centered robotic design.

Ronald Choe received his M.Sc. degree in Aerospace Engineering from the Delft University of Technology, The Netherlands in 2002. From 2005-2009 he worked at ST Aerospace and DSO National Laboratories, both in Singapore. Currently he is a Ph.D. candidate in the Department of Aerospace Engineering at the University of Illinois at Urbana-Champaign. His research interests are in the theory of nonlinear and robust adaptive control with applications in (autonomous) aerospace systems, distributed optimization, and cooperative trajectory generation and control of multi-vehicle systems.

Syed Bilal Mehdi received his MS degree in 2012 from the Mechanical Science and Engineering department at the University of Illinois at Urbana-Champaign (UIUC). He is currently a PhD candidate in the same department. In 2013, he was named a CSE fellow by the Computational Science and Engineering department (CSE) and was later awarded as a teaching fellow at

the Mechanical Science and Engineering department at UIUC. His research interests include cooperative multi-vehicle missions with a focus on collision avoidance and nonlinear robust adaptive control.

Enric Xargay received his Ph.D. in Aerospace Engineering from the University of Illinois at Urbana-Champaign (UIUC) in 2013. Before that, he earned an M.S. in Control Engineering from the Technical University of Catalonia and an M.S. in Aerospace Engineering from the Politecnico di Torino, both in 2007. Since 2013, he has been with the Department of Mechanical Science and Engineering at UIUC, where he is a postdoctoral research associate. In 2011, he received the Roger A. Strehlow Memorial Award from the Department of Aerospace Engineering at UIUC for outstanding research accomplishment. His professional and research interests include aircraft flight control; fault-tolerant and adaptive control; nonlinear systems; and cooperative motion planning and control of autonomous systems.

Naira Hovakimyan received her M.S. degree in Theoretical Mechanics and Applied Mathematics in 1988 from Yerevan State University in Armenia. She got her Ph.D. in Physics and Mathematics in 1992, in Moscow, from the Institute of Applied Mathematics of Russian Academy of Sciences, majoring in optimal control and differential games. In 1997 she has been awarded a governmental postdoctoral scholarship to work in INRIA, France. In 1998 she was invited to the School of Aerospace Engineering of Georgia Tech, where she worked as a research faculty member until 2003. In 2003 she joined the Department of Aerospace and Ocean Engineering of Virginia Tech, and in 2008 she moved to University of Illinois at Urbana-Champaign, where she is a professor, university scholar and Schaller faculty scholar of Mechanical Science and Engineering. She has co-authored a book and more than 300 refereed publications. She is the recipient of the SICE

International scholarship for the best paper of a young investigator in the VII ISDG Symposium (Japan, 1996), and also the 2011 recipient of AIAA Mechanics and Control of Flight award. In 2014 she was awarded the Humboldt prize for her lifetime achievements and was recognized as Hans Fischer senior fellow of Technical University of Munich. She is an associate fellow and life member of AIAA, a Senior Member of IEEE, and a member of SIAM, AMS and ISDG. Her research interests are in the theory of robust adaptive control and estimation, control in the presence of limited information, networks of autonomous systems, game theory and applications of those in safety-critical systems of aerospace, mechanical, electrical, petroleum and biomedical engineering.

Isaac Kaminer received his Ph.D. in Electrical Engineering from University of Michigan in 1992. Before that he spent four years working at Boeing Commercial first as a control engineer in 757/767/747-400 Flight Management Computer Group and then as an engineer in Flight Control Research Group. Since 1992 he has been with the Naval Postgraduate School first at the Aeronautics and Astronautics Department and currently at the Department of Mechanical and Aerospace Engineering where he is a Professor. He has a total of over 20 years of experience in development and flight testing of guidance, navigation and control algorithms for both manned and unmanned aircraft. His more recent efforts were focused on development of coordinated control strategies for multiple UAVs and vision based guidance laws for a single UAV. Professor Kaminer has co-authored more than a hundred refereed publications. Over the years his research has been supported by ONR, NASA, US Army, NAVAIR and USSOCOM.

Vladimir Dobrokhodov received his M.S. degrees in Astronautical Engineering in 1991 and in Operations Research in 1993 from the Moscow State Aviation Institute and the Air

Force Engineering Academy (AFEA), respectively. He earned his Ph.D. in Aeronautical and Astronautical Engineering in 1999, from the Air Force Engineering Academy. Upon finishing his Ph.D. he joined the Flight Dynamics chair at the AFEA as a research scientist. He joined the Naval Postgraduate School (NPS) in February 2001, first as a winner of the United States National Research Council Postdoctoral Fellowship award and then as an Associate Professor of the Mechanical and Aerospace engineering department at the NPS. He has authored/coauthored more than 90 refereed publications. His research interests include flight dynamics and control; online trajectory optimization; guidance, navigation and control of unmanned systems; cooperative control of multi-vehicle formations; real-time embedded control systems design; design of validation and verification flight experiments.

António M. Pascoal received the Licenciatura degree in electrical engineering from the Instituto Superior Técnico (IST), Lisbon, Portugal in 1974, the M.S. degree in electrical engineering from the University of Minnesota in 1983, and the Ph.D. degree in control science from the same school in 1987. From 1987-88 he was a Research Scientist with Integrated Systems Incorporated, Santa Clara, California, where he conducted research and development work in the areas of system modeling and identification and robust and adaptive control. Since 1998 he has been with the Department of Electrical Engineering of IST, where he is currently an Associate Professor of Control and Robotics and Member, Scientific Council of the Institute for Systems and Robotics, Lisbon. He is also an Adjunct Scientist with the National Institute of Oceanography (NIO), Goa India. He was the Elected Chair, IFAC Technical Committee Marine Systems, from 2008-2014. He has coordinated and participated in a large number of international projects with institutions in Europe, US, India, and Korea, that have led to the design, development, and field-testing of single and multiple autonomous marine and air vehicles. His research interests include

linear and nonlinear control theory, robust adaptive control, and networked cooperative motion planning, navigation, and control of multiple autonomous vehicles with applications to air, land, and underwater robots. His long-term goal is to contribute to the development of advanced robotic systems for ocean exploration and exploitation.

A. Pedro Aguiar received the Licenciatura, M.S. and Ph.D. in electrical and computer engineering from the Instituto Superior Tcnico (IST), Technical University of Lisbon, Portugal in 1994, 1998 and 2002, respectively. Currently, Dr. Aguiar holds an Associate Professor position with the Department of Electrical and Computer Engineering (DEEC), Faculty of Engineering, University of Porto (FEUP). From 2002 to 2005, he was a post-doctoral researcher at the Center for Control, Dynamical-Systems, and Computation at the University of California, Santa Barbara (UCSB). From 2005 to 2012, he was a senior researcher with the Institute for Systems and Robotics at IST (ISR/IST), and an invited assistant professor with the Department of Electrical and Computer Engineering, IST. His research interests include modeling, control, navigation, and guidance of autonomous robotic vehicles, nonlinear control, switched and hybrid systems, tracking, path-following, performance limitations, nonlinear observers, the integration of machine vision with feedback control, networked control, and coordinated/cooperative control of multiple autonomous robotic vehicles.