

Time-Critical Cooperative Control for Multiple Autonomous Vehicles

Robust decentralized strategies for path-following control and
time-coordination over dynamic communications networks

Enric Xargay, Vladimir Dobrokhodov, Isaac Kaminer,
António M. Pascoal, Naira Hovakimyan, and Chengyu Cao

Worldwide, there has been growing interest in the use of autonomous vehicles to execute missions of increasing complexity without constant supervision of human operators. A key enabling element for the execution of such missions is the availability of advanced systems for motion control of autonomous vehicles. Usually, the problems of motion control for a single autonomous vehicle are roughly classified into three groups. Namely, *point stabilization*, where the goal is to stabilize a vehicle about a given target point with a desired orientation; *trajectory tracking*, where the vehicle is required to track a time parameterized reference; and *path following*, where the objective is to make the vehicle converge to and follow a desired geometric path, without an explicit timing law assigned to it. Current research goes well beyond single vehicle control. In fact, challenging mission scenarios and the advent of powerful embedded systems, sensors, and communications networks have spawned widespread interest in the problem of cooperative motion control of multiple autonomous vehicles.

The types of applications considered are numerous and include aircraft and spacecraft formation control [1–13], coordinated control of land robots [14–17], control of multiple surface and underwater vehicles [18–22], and networked control of robotic systems [23–29].

In aerospace, for instance, unmanned systems have become ubiquitous in both military and civilian applications. Today, for example, in a given constrained airspace volume, unmanned air vehicles (UAVs) must execute military reconnaissance and strike operations, border patrol missions, forest fire detection, police surveillance, and recovery operations, to name a few. Representative applications include sequential auto-landing and coordinated ground-target suppression using multiple UAVs. The first application refers to the situation in which a fleet of UAVs must break up and arrive at the assigned glideslope separated by prespecified safe-guarding time intervals. For the case of ground-target suppression, a formation of UAVs must also break up and execute a coordinated maneuver to arrive at a predefined position over the target at the same time. In both cases, only relative –rather than absolute– temporal constraints are specified *a priori*, a critical point that needs to be emphasized. Furthermore, the vehicles must execute maneuvers in close proximity to each other. Thus, the key requirement is that all maneuvers must be collision-free. In addition, as pointed out in [30, 31], the flow of information among vehicles may be severely restricted, either for security reasons or because of tight bandwidth limitations. It is natural, under these circumstances, that no vehicle is able to communicate with the entire formation; furthermore, the amount of information that can be exchanged may be severely limited. It is therefore imperative to develop cooperative motion control strategies that can yield robust performance in the presence of time-varying communications networks arising from temporary loss of communication links and switching communication topologies.

Decoupling Space and Time

Motivated by the multi-vehicle mission scenarios mentioned above, in this article we present a solution to the problem of cooperative control of multiple heterogeneous autonomous vehicles that must operate under strict spatial and temporal constraints, while ensuring collision-free maneuvers. The theoretical framework adopted borrows from various disciplines, and integrates algorithms for trajectory generation, path following, time-critical coordination, and \mathcal{L}_1 adaptive control theory for fast and robust adaptation. Together, these techniques yield a control architecture that allows meeting strict performance requirements in the presence of complex vehicle dynamics, communication constraints, and partial vehicle failures.

The methodology developed, which is based on the key idea of *decoupling space and time* in the problem formulation, can be summarized in three basic steps. First, given a multiple vehicle mission, a set of feasible spatial paths together with a set of feasible speed profiles is generated for all the vehicles involved in the mission. This step relies on optimization methods that take explicitly into account initial and final boundary conditions, a general performance criterion to be optimized, simplified vehicle dynamics, and safety rules for collision avoidance. At this stage, the decoupling of spatial and temporal assignments in path generation ensures that the computational complexity of the trajectory-generation algorithm increases only linearly with the number of vehicles. This feature is critical for real-time implementation of the trajectory-generation algorithm. The second step consists of making each vehicle follow its assigned path, regardless of what the desired speed profile is, as long as the latter is physically feasible. This approach also takes advantage of the separation in space and time introduced during trajectory generation, and leaves the speed profile of the vehicle as an extra degree of freedom to be

modified at the time-coordination level. In this sense, the path-following approach adopted is in contrast to trajectory tracking, for which it is proven in [32] that, in the presence of unstable zero dynamics, there exist fundamental performance limitations that cannot be overcome by any controller structure. Finally, in the third step, the speed profile of each vehicle is adjusted about its desired speed profile obtained from the trajectory-generation algorithm, to enforce the temporal constraints that must be met in real time to coordinate the entire fleet of vehicles. This last step relies on the underlying time-varying communications network as a means to exchange information among the vehicles.

Another key feature of the framework presented in this article is that it exhibits a multiloop control structure in which an inner-loop controller stabilizes the vehicle dynamics, while a guidance outer-loop controller is designed to control the vehicle kinematics, providing path-following and time-coordination capabilities. To make these ideas more precise, we notice that a typical autonomous vehicle can be modeled as a cascade system consisting of the kinematic and dynamic equations of the vehicle. Following standard notation, the kinematics \mathcal{G}_e of the vehicle can be represented as

$$\dot{x}(t) = f(x(t)) + g(x(t))y(t), \quad (1)$$

where $x(t)$ denotes the kinematic state of the vehicle, which usually includes the vehicle's position and attitude, $y(t)$ represents the vector of variables driving the vehicle kinematics, such as vehicle angular and linear velocities, and $f(\cdot)$ and $g(\cdot)$ are known nonlinear functions. The dynamics \mathcal{G}_p of the vehicle can be expressed as

$$\dot{z}(t) = h(z(t), u(t), t), \quad (2)$$

$$y(t) = h_o(z(t), u(t), t), \quad (3)$$

where $z(t)$ denotes the dynamic state of the vehicle, $u(t)$ represents the control signal that drives the vehicle dynamics, and $h(\cdot)$ and $h_o(\cdot)$ are partially known nonlinear functions. The model adopted is sufficiently general to capture six-degree-of-freedom dynamics, together with plant uncertainty. In the cooperative control algorithms presented in this article, the path-following and time-critical coordination control laws are derived at the kinematic level for the system \mathcal{G}_e in (1) and are viewed as guidance outer-loop controllers providing reference commands to an inner-loop controller. The latter is designed to stabilize the dynamics \mathcal{G}_p in (2)-(3) and to ensure that the vehicle tracks the outer-loop commands. This inner-outer loop approach simplifies the design process and affords the designer a systematic approach to seamlessly tailor the algorithms for a very general class of vehicles that come equipped with inner-loop commercial autopilots. Moreover, in order to meet strict performance requirements in the presence of modeling uncertainty and significant environmental disturbances, the framework relies on the design of \mathcal{L}_1 adaptive control loops to augment the inner-loop controller. In fact, employing \mathcal{L}_1 adaptation allows us to make fairly general assumptions on the vehicle dynamics.

In this article, tools from real-time optimization, Lyapunov-based stability analysis, robust control, graph theory, and \mathcal{L}_1 adaptation are brought together for the development of cooperative control algorithms yielding robust performance of a fleet of autonomous vehicles executing various time-critical cooperative missions. In particular, since typical autopilots are normally designed to provide only guidance loops for waypoint navigation, the framework described in this article broadens the range of possible applications and mission scenarios of autonomous vehicles. The conceptual architecture of the complete solution is shown in Figure 1. In the subsequent sections, we describe each of the functional blocks in the figure. We start by formulating the problem of generation of feasible collision-free trajectories, and refer to specific techniques for

real-time implementation of trajectory-generation algorithms. Next, we present a nonlinear path-following controller, derived at the kinematic level, that ensures that each vehicle follows its desired path independently of its temporal assignments. Afterwards, we describe a strategy for time-coordinated control of multiple vehicles that relies on the adjustment of the speed profile of each vehicle. Then, in order to enhance the safety and success of the time-critical cooperative mission, we consider the implementation of \mathcal{L}_1 adaptive architectures so as to ensure consistent performance in the event of failures, vehicle damage, or in the presence of adverse environmental disturbances. Finally, we present experimental results of a cooperative road-search mission that exploits the multi-vehicle cooperative control framework discussed in this article.

Cooperative Trajectory Generation for Multiple Autonomous Vehicles

Real-time trajectory generation that explicitly accounts for given boundary conditions and vehicle dynamic constraints is a critical requirement for the autonomous vehicles engaged in executing the missions described in the introduction of this article. Surveys on trajectory-planning algorithms can be found in [33, 34]. Next, we formulate and describe a solution to the problem of cooperative trajectory generation to compute feasible spatial paths and speed profiles for multiple autonomous vehicles that satisfy collision-avoidance constraints.

For the cooperative missions of interest involving n vehicles, the cooperative *trajectory-generation problem* can be formulated as finding a set of n 3D time trajectories $\Phi_{d,i}(t_d) : \mathbb{R} \rightarrow \mathbb{R}^3$, conveniently parameterized by a single time variable $t_d \in [0, t_d^*]$, that together minimize a cost function $J(\cdot)$, satisfy desired boundary conditions, do not violate the dynamic constraints of each vehicle, and ensure that the vehicles maintain a predefined spatial

clearance. In this formulation, the variable t_d represents a *desired mission time*, which is used during the trajectory-generation phase and is distinct from the actual mission time that evolves as the mission unfolds, while t_d^* is the *desired mission duration*. For a given t_d , $\Phi_{d,i}(t_d)$ defines thus the desired position of the i th UAV t_d seconds after the initiation of the cooperative mission. From these time trajectories, *spatial paths* $p_{d,i}(\tau_{\ell,i}) : \mathbb{R} \rightarrow \mathbb{R}^3$ and the corresponding *desired speed profiles* $v_{d,i}(t_d) : \mathbb{R} \rightarrow \mathbb{R}$ can be easily derived for all the UAVs. For convenience, we parameterize each spatial path by its path length $\tau_{\ell,i} \in [0, \ell_{fi}]$, where ℓ_{fi} denotes the total length of the i th path, whereas the desired speed profiles are parameterized by the desired mission time t_d .

Feasible trajectory generation for a single vehicle

Before formulating the cooperative trajectory-generation problem for multiple vehicles, we first address the problem of generating a feasible trajectory for a single vehicle. We start by considering a desired spatial path to be followed by the i th vehicle and characterized by the three-dimensional curve $p_{d,i}(\tau_{\ell,i})$, conveniently parameterized by its path length $\tau_{\ell,i} \in [0, \ell_{fi}]$. A desired speed profile $v_{d,i}(t_d)$ can then be generated by relating the path length $\tau_{\ell,i}$ to mission time t_d through a dynamic relation of the form $\frac{d\tau_{\ell,i}}{dt_d} = \theta_i(\tau_{\ell,i})$, where $\theta_i(\cdot)$ is a positive function, smooth in its argument. The mission time t_d can thus be computed from the path length $\tau_{\ell,i}$ as

$$t_d = \int_0^{\tau_{\ell,i}} \frac{1}{\theta_i(\sigma_\tau)} d\sigma_\tau.$$

This notation allows us to express the vehicle's speed and acceleration as well as the curvature of the path, its torsion, and the flight path angle in terms of $p_{d,i}(\tau_{\ell,i})$ and its first, second, and third partial derivatives with respect to $\tau_{\ell,i}$, denoted by $p'_{d,i}(\tau_{\ell,i})$, $p''_{d,i}(\tau_{\ell,i})$, and

$p_{d,i}'''(\tau_{\ell,i})$ respectively, as

$$\begin{aligned}
v_{d,i}(t_d(\tau_{\ell,i})) &= \|p'_{d,i}(\tau_{\ell,i})\| \theta_i(\tau_{\ell,i}), \\
a_{d,i}(t_d(\tau_{\ell,i})) &= \|p''_{d,i}(\tau_{\ell,i})\theta_i(\tau_{\ell,i}) + p'_{d,i}(\tau_{\ell,i})\theta'_i(\tau_{\ell,i})\| \theta_i(\tau_{\ell,i}), \\
\kappa_{d,i}(\tau_{\ell,i}) &= \frac{\|p'_{d,i}(\tau_{\ell,i}) \times p''_{d,i}(\tau_{\ell,i})\|}{\|p'_{d,i}(\tau_{\ell,i})\|^3}, \\
\tau_{d,i}(\tau_{\ell,i}) &= \frac{(p'_{d,i}(\tau_{\ell,i}) \times p''_{d,i}(\tau_{\ell,i})) \cdot p'''_{d,i}(\tau_{\ell,i})}{\|p'_{d,i}(\tau_{\ell,i}) \times p''_{d,i}(\tau_{\ell,i})\|}, \\
\gamma_{d,i}(\tau_{\ell,i}) &= \arctan \left(\frac{p'_{d,i}(\tau_{\ell,i}) \cdot \vec{e}_{I3}}{\left((p'_{d,i}(\tau_{\ell,i}) \cdot \vec{e}_{I1})^2 + (p'_{d,i}(\tau_{\ell,i}) \cdot \vec{e}_{I2})^2 \right)^{\frac{1}{2}}} \right),
\end{aligned}$$

where the orthonormal vectors $\{\vec{e}_{I1}, \vec{e}_{I2}, \vec{e}_{I3}\}$ characterize an inertial reference frame \mathcal{I} . The unit vectors \vec{e}_{I1} and \vec{e}_{I2} lie in the horizontal plane, while the unit vector \vec{e}_{I3} points up vertically in the opposite direction of gravity. Moreover, the desired mission duration t_d^* can be written as

$$t_d^* = \int_0^{\ell_{fi}} \frac{1}{\theta_i(\tau_{\ell,i})} d\tau_{\ell,i}.$$

With the above formulation, we define a *feasible trajectory* as the one that satisfies maximum curvature, torsion, and flight-path-angle bounds, and it can be followed by a vehicle without having it exceed prespecified bounds on the vehicle speed $v_{d,i}(t_d)$ and acceleration $a_{d,i}(t_d)$. Letting v_{\min} , v_{\max} , a_{\max} , κ_{\max} , τ_{\max} , γ_{\min} , and γ_{\max} denote predefined bounds on the vehicle's velocity, acceleration, path curvature, torsion, and flight path angle, the trajectory $\Phi_{d,i}(t_d)$ is said to be feasible if the conditions

$$0 < v_{\min} \leq v_{d,i}(t_d(\tau_{\ell,i})) \leq v_{\max}, \quad |a_{d,i}(t_d(\tau_{\ell,i}))| \leq a_{\max}, \quad (4)$$

$$|\kappa_{d,i}(\tau_{\ell,i})| \leq \kappa_{\max}, \quad |\tau_{d,i}(\tau_{\ell,i})| \leq \tau_{\max}, \quad \gamma_{\min} \leq \gamma_{d,i}(\tau_{\ell,i}) \leq \gamma_{\max}, \quad (5)$$

are met for all $\tau_{\ell,i} \in [0, \ell_{fi}]$.

A feasible trajectory for the i th vehicle can thus be obtained by solving, for example, the optimization problem

$$\min_{\Xi_i} J(\cdot)$$

subject to initial and final boundary conditions as well as the feasibility conditions in (4)-(5). In the problem above, $J(\cdot)$ is a given cost function, and Ξ_i represents the vector of optimization parameters for the i th vehicle, which might include the total path length ℓ_{fi} , a set of parameters characterizing the curve $p_{d,i}(\cdot)$, and a set of parameters characterizing the timing function $\theta_i(\cdot)$. In the trajectory-generation problem above, the cost function $J(\cdot)$ may include terms related to mission-specific goals, while additional constraints can also be added to account for vehicle-to-ground communications limitations, sensory capabilities, and collision avoidance with obstacles.

Real-time collision-free trajectory generation for multiple vehicles

We now formulate the problem of cooperative trajectory generation for multiple vehicles. In particular, the time-critical missions described in this article require that each vehicle follow a collision-free trajectory, and that all vehicles arrive at their respective destinations at the same time, or at different times so as to meet a desired inter-vehicle schedule. Without loss of generality, we consider the problem of simultaneous arrival. For these missions, the generation of collision-free trajectories can be addressed using two complementary approaches. The first one, referred to as *collision avoidance in space*, ensures that no feasible paths intersect. Alternatively, the second approach –*collision avoidance in time*– implies that no two vehicles are at the same place at the same time. The first approach may be particularly useful in military applications, where jamming prevents vehicles from communicating with each other, and is preferable to the

current practice of separating vehicles by altitude. On the other hand, the second approach relies heavily on inter-vehicle communications to properly coordinate the vehicle motions and is thus a function of the quality of service (QoS) of the underlying network. Formally, these two strategies lead to two alternative constraints. For collision avoidance in space, the paths for the n vehicles need to satisfy the constraint

$$\min_{j,k=1,\dots,n,j \neq k} \|p_{d,j}(\tau_{\ell,j}) - p_{d,k}(\tau_{\ell,k})\|^2 \geq E^2 \quad \text{for all } (\tau_{\ell,j}, \tau_{\ell,k}) \in [0, \ell_{fj}] \times [0, \ell_{fk}],$$

whereas, for collision avoidance in time, the paths and speed profiles need to verify that

$$\min_{j,k=1,\dots,n,j \neq k} \|p_{d,j}(\tau_{\ell,j}(t_d)) - p_{d,k}(\tau_{\ell,k}(t_d))\|^2 \geq E^2 \quad \text{for all } t_d \in [0, t_d^*],$$

where E is the distance for spatial clearance. See Figures 2 and 3 for the illustration of these two approaches.

In addition to collision avoidance, the simultaneous time-of-arrival requirement adds an additional constraint on trajectory-generation problem. Let $\delta t_{d,i}^* \triangleq [t_{d\min,i}^*, t_{d\max,i}^*]$ be the *arrival-time window* for the i th vehicle, where $t_{d\min,i}^*$ and $t_{d\max,i}^*$ represent the minimal and maximal possible durations of the mission for the i th vehicle, defined as

$$t_{d\min,i}^* \triangleq \frac{\ell_{fi}}{v_{\max}}, \quad t_{d\max,i}^* \triangleq \frac{\ell_{fi}}{v_{\min}}.$$

Then, the simultaneous arrival problem has a solution if and only if the intersection of the arrival-time windows is nonempty, that is, $\delta t_{d,i}^* \cap \delta t_{d,j}^* \neq \emptyset$ for all $i, j \in \{1, \dots, n\}$, $i \neq j$. In particular, if we define the *arrival margin* δT^* as

$$\delta T^* \triangleq \min_i t_{d\max,i}^* - \max_i t_{d\min,i}^*,$$

then non-emptiness of the intersection of arrival-time windows is implied by enforcing a positive arrival margin; see Figure 4. Moreover, enlarging the arrival margin adds robustness to the mission

operation at the coordination level.

Then, letting T_d^* be a predefined upper bound on the final time for the mission to be completed, and defining a cost function $J(\cdot)$ to be minimized, the cooperative *trajectory-generation problem* can be formulated as two alternative optimization problems. The first optimization problem addresses collision avoidance in space and is formulated as follows:

$$\min_{\Xi_1 \times \dots \times \Xi_n} J(\cdot) \quad (6)$$

subject to initial and final boundary conditions and the feasibility conditions in (4)-(5) for all vehicles $i \in \{1, \dots, n\}$, as well as the constraints

$$\begin{aligned} \min_{j,k=1,\dots,n,j \neq k} \|p_{d,j}(\tau_{\ell,j}) - p_{d,k}(\tau_{\ell,k})\|^2 &\geq E^2 \quad \text{for all } (\tau_{\ell,j}, \tau_{\ell,k}) \in [0, \ell_{fj}] \times [0, \ell_{fk}], \\ \delta T^* &\geq \delta T_0^* > 0, \\ t_d^* &= \int_0^{\ell_{fj}} \frac{1}{\theta_j(\tau_{\ell,j})} d\tau_{\ell,j} = \int_0^{\ell_{fk}} \frac{1}{\theta_i(\tau_{\ell,k})} d\tau_{\ell,k}, \quad \text{for all } j, k = 1, \dots, n, \ j \neq k, \\ t_d^* &\leq T_d^*, \end{aligned}$$

where Ξ_i is the set of optimization parameters for the i th vehicle including the total path length ℓ_{fi} , the set of parameters characterizing the curve $p_{d,i}(\cdot)$, and the set of parameters characterizing the timing function $\theta_i(\cdot)$. In the above, E is the minimal allowable separation distance between the paths, which must be selected based on the prespecified path-following controller performance. Finally, $\delta T^* \geq \delta T_0^* > 0$ imposes a bounded away from zero arrival margin requirement.

The second optimization problem accounts for the collision avoidance in time and is

posed as:

$$\min_{\Xi_1 \times \dots \times \Xi_n} J(\cdot) \quad (7)$$

subject to initial and final boundary conditions and the feasibility conditions in (4)-(5) for all vehicles $i \in \{1, \dots, n\}$, as well as the constraints

$$\min_{j,k=1,\dots,n,j \neq k} \|p_{d,j}(\tau_{\ell,j}(t_d)) - p_{d,k}(\tau_{\ell,k}(t_d))\|^2 \geq E^2 \quad \text{for all } t_d \in [0, t_d^*],$$

$$\delta T^* \geq \delta T_0^* > 0,$$

$$t_d^* = \int_0^{\ell_{fj}} \frac{1}{\theta_j(\tau_{\ell,j})} d\tau_{\ell,j} = \int_0^{\ell_{fk}} \frac{1}{\theta_k(\tau_{\ell,k})} d\tau_{\ell,k}, \quad \text{for all } j, k = 1, \dots, n, \quad j \neq k,$$

$$t_d^* \leq T_d^*,$$

where Ξ_i is the set of optimization parameters for the i th vehicle including the total path length ℓ_{fi} , the set of parameters characterizing the curve $p_{d,i}(\cdot)$, and the set of parameters characterizing the timing function $\theta_i(\cdot)$. In the above, E represents again the minimal allowable separation distance between the paths, which in this case is to be selected based not only on the prespecified path-following controller performance, but also of the coordination controller performance, and the QoS of the communications network, characterized later in the article.

In the cooperative trajectory-generation problems above, the cost function $J(\cdot)$ includes terms related to mission-specific goals and cooperative performance criteria, while additional constraints can also be added to account, for instance, for inter-vehicle and vehicle-to-ground communications limitations, sensory capabilities, and collision avoidance with obstacles. The formulation of these problems can also address the problem of mission planning and task allocation under resource constraints; see [35, 36] and references therein.

The optimization problems (6) and (7) are in general complex, and may be NP hard and nonconvex. Nevertheless, these problems can in principle be tackled using methods that include multiple shooting [37, 38], pseudospectral Legendre methods [39, 40], the *nonlinear trajectory generation method* [41, 42], randomized techniques such as rapidly-exploring random trees [43, 44] and probabilistic roadmap algorithms [45], the *direct method for rapid trajectory prototyping* in [46], the generalized Benders decomposition [47], branch and bound approaches [48], outer approximations [49], and the generalized cross decomposition [50], to name a few.

The outcome of the optimization problems (6) and (7) is a set of n feasible spatial paths $p_{d,i}(\tau_{\ell,i})$ and corresponding desired speed profiles $v_{d,i}(t_d)$ such that, if each agent follows its assigned path and speed profile, the time-critical mission is expected to be executed in the ideal case. However, the presence of disturbances, modeling uncertainty, and failures in the communications network, require the synthesis of robust feedback laws to ensure that the mission can be accomplished with a high degree of confidence. In the remaining sections of this article, we present a general framework to synthesize path-following and coordination control laws that can address the performance of the overall time-critical mission in the presence of uncertainty and a faulty time-varying communications network.

3D Path Following for a Single Vehicle

In this section, we describe an outer-loop 3D path-following control algorithm that ensures, at a kinematic level, that the i th vehicle converges to and follows the path $p_{d,i}(\cdot)$ for an arbitrary speed profile, subjected only to feasibility conditions. In particular, the path-following algorithm described in this article relies on the insight that a vehicle can follow a

given path using only its attitude, thus leaving its speed as an extra degree of freedom to be used at the coordination level. The key idea of the algorithm is to use the vehicle’s attitude control effectors to follow a *virtual target vehicle* running along the path. To this effect, we introduce a frame attached to this virtual target and define a generalized error vector between this moving coordinate system and a frame attached to the actual vehicle. With this setup, the path-following control problem is reduced to driving this generalized error vector to zero by using only vehicle’s attitude control effectors, while following an arbitrary feasible speed profile. This approach, which is presented in this article for the case of 3D spatial paths, is motivated by the work on 2D path-following control reported in [51]. A brief overview of different approaches used for the derivation of path-following algorithms can be found in “Path-Following Control”.

Next, we characterize the dynamics of the kinematic errors between the i th vehicle and its virtual target, and present an outer-loop 3D path-following control algorithm that solves the path-following problem at the kinematic level. In the description below, the notation $\{v\}_F$ is used to denote the vector v resolved in frame \mathcal{F} ; $\{\vec{e}\}_F$ represents the versor \vec{e} resolved in frame \mathcal{F} ; $\omega_{F1/F2}$ denotes the angular velocity of frame $\mathcal{F}1$ with respect to frame $\mathcal{F}2$; the rotation matrix from frame $\mathcal{F}1$ to frame $\mathcal{F}2$ is represented by R_{F1}^{F2} ; and $\dot{v}]_F$ indicates that the time-derivative of vector v is taken in frame \mathcal{F} . For notational simplicity, we drop the subscript i used to denote a particular vehicle.

Following a virtual target vehicle

Figure 5 captures the geometry of the problem at hand. Let $p_d(\cdot)$ be the desired path assigned to one of the UAVs, and let ℓ_f be its total length. Let \mathcal{I} denote an inertial reference

frame $\{\vec{e}_{I1}, \vec{e}_{I2}, \vec{e}_{I3}\}$, and let $p_I(t)$ be the position of the center of mass Q of the UAV in this inertial frame. Further, let P be an arbitrary point on the desired path that plays the role of the virtual target, and let $p_d(\ell)$ denote its position in the inertial frame. Here $\ell \in [0, \ell_f]$ is a free length variable that defines the position of the virtual target vehicle along the path. In the setup adopted, the total rate of progression of the virtual target along the path is an extra design parameter. This approach is in striking contrast with the strategy used in the path-following algorithm introduced in [52], where P is defined as the point on the path that is closest to the vehicle. Endowing the point P with an extra degree of freedom is the key to the path-following algorithm presented in [51] and its extension to the 3D case described in this article.

For our purposes, it is convenient to define a *parallel transport frame* \mathcal{F} [53, 54] attached to the point P on the path and characterized by the orthonormal vectors $\{\vec{t}(\ell), \vec{n}_1(\ell), \vec{n}_2(\ell)\}$. The vectors $\{\vec{t}, \vec{n}_1, \vec{n}_2\}$ define an orthonormal basis for \mathcal{F} , in which the unit vector $\vec{t}(\ell)$ defines the tangent direction to the path at the point determined by ℓ , while $\vec{n}_1(\ell)$ and $\vec{n}_2(\ell)$ define the normal plane perpendicular to $\vec{t}(\ell)$. Moreover, let $p_F(t)$ be the position of the vehicle's center of mass Q in the parallel transport frame, and let $x_F(t)$, $y_F(t)$, and $z_F(t)$ be the components of the vector $p_F(t)$ with respect to the basis $\{\vec{t}, \vec{n}_1, \vec{n}_2\}$.

Let \mathcal{W} denote a vehicle-carried *velocity frame* $\{\vec{w}_1, \vec{w}_2, \vec{w}_3\}$ with its origin at the UAV center of mass Q and its x -axis aligned with the velocity vector of the UAV. The z -axis is chosen to lie in the plane of symmetry of the UAV, and the y -axis is determined by completing the right-hand system. In this article, $q(t)$ and $r(t)$ are the y -axis and z -axis components, respectively, of the vehicle's rotational velocity resolved in the \mathcal{W} frame. With a slight abuse of notation, $q(t)$ and $r(t)$ are referred to as *pitch rate* and *yaw rate*, respectively, in the \mathcal{W} frame.

We also introduce an auxiliary frame $\mathcal{D} \{\vec{b}_{1D}, \vec{b}_{2D}, \vec{b}_{3D}\}$, which is used to shape the approach attitude to the path as a function of the “cross-track” error components y_F and z_F . The frame \mathcal{D} has its origin at the UAV center of mass and the vectors $\vec{b}_{1D}(t)$, $\vec{b}_{2D}(t)$, and $\vec{b}_{3D}(t)$ are defined as

$$\vec{b}_{1D} \triangleq \frac{d\vec{t} - y_F \vec{n}_1 - z_F \vec{n}_2}{(d^2 + y_F^2 + z_F^2)^{\frac{1}{2}}}, \quad \vec{b}_{2D} \triangleq \frac{y_F \vec{t} + d \vec{n}_1}{(d^2 + y_F^2)^{\frac{1}{2}}}, \quad \vec{b}_{3D} \triangleq \vec{b}_{1D} \times \vec{b}_{2D}, \quad (8)$$

with $d > 0$ being a constant characterizing distance. The basis vector $\vec{b}_{1D}(t)$ defines the desired direction of the UAV’s velocity vector. As illustrated in Figure 6, when the vehicle is far from the desired path, the vector $\vec{b}_{1D}(t)$ becomes perpendicular to $\vec{t}(\ell)$. As the vehicle comes closer to the path and the cross-track error becomes smaller, then $\vec{b}_{1D}(t)$ tends to $\vec{t}(\ell)$.

Finally, let $\tilde{R}(t)$ be the rotation matrix from \mathcal{W} to \mathcal{D} , that is,

$$\tilde{R} \triangleq R_W^D = R_F^D R_W^F = (R_D^F)^\top R_W^F,$$

and define the real-valued error function

$$\Psi(\tilde{R}) \triangleq \frac{1}{2} \text{tr} \left[(\mathbb{I}_3 - \Pi_R^\top \Pi_R) (\mathbb{I}_3 - \tilde{R}) \right], \quad (9)$$

where Π_R is defined as $\Pi_R \triangleq \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$. The function $\Psi(\tilde{R})$ in (9) can be expressed in terms of the entries of $\tilde{R}(t)$ as

$$\Psi(\tilde{R}) = \frac{1}{2} (1 - \tilde{R}_{11}),$$

where $\tilde{R}_{11}(t)$ denotes the $(1, 1)$ entry of $\tilde{R}(t)$. Therefore, $\Psi(\tilde{R})$ is positive-definite about $\tilde{R}_{11} = 1$.

We note that $\tilde{R}_{11} = 1$ corresponds to the situation where the velocity vector of the UAV is aligned with the basis vector $\vec{b}_{1D}(t)$, which defines the desired attitude of the UAV.

With the above notation, as shown in [55], the path-following kinematic-error dynamics \mathcal{G}_e between the vehicle and its virtual target vehicle can be written as

$$\dot{p}_F]_F = -\dot{\ell}\vec{t} - \omega_{F/I} \times p_F + v\vec{w}_1, \quad (10)$$

$$\dot{\Psi}(\tilde{R}) = e_{\tilde{R}} \cdot \left(\begin{bmatrix} q \\ r \end{bmatrix} - \Pi_R \tilde{R}^\top (R_F^D \{\omega_{F/I}\}_F + \{\omega_{D/F}\}_D) \right), \quad (11)$$

where $\cdot]_F$ is used to indicate that the derivative is taken in the parallel transport frame, $v(t)$ denotes the magnitude of the UAV's ground velocity vector, and $e_{\tilde{R}}(t)$ is the attitude kinematic error vector defined as

$$e_{\tilde{R}} \triangleq \frac{1}{2} \Pi_R \left((\mathbb{I}_3 - \Pi_R^\top \Pi_R) \tilde{R} - \tilde{R}^\top (\mathbb{I}_3 - \Pi_R^\top \Pi_R) \right)^\vee,$$

where $(\cdot)^\vee : \text{SO}(3) \rightarrow \mathbb{R}^3$ denotes the *vee map* [56]. In the kinematic-error model (10)-(11), $q(t)$ and $r(t)$ play the role of control inputs, while the rate of progression $\dot{\ell}(t)$ of the point P along the path becomes an extra variable that can be manipulated at will. At this point, it is convenient to formally define the path-following generalized error vector $x_{\text{pf}}(t)$ as

$$x_{\text{pf}} \triangleq \begin{bmatrix} p_F^\top & e_{\tilde{R}}^\top \end{bmatrix}^\top.$$

Notice that, within the region where $\Psi(\tilde{R}) < 1$, if $x_{\text{pf}} = 0$, then both the path-following position error and the path-following attitude error are equal to zero, that is, $p_F = 0$ and $\tilde{R}_{11} = 0$.

Path-following control law

Using the above formulation, and given a feasible spatially defined path $p_d(\cdot)$, we define the *path-following problem* as that of determining feedback control laws for $q(t)$, $r(t)$, and $\dot{\ell}(t)$ such that all closed-loop signals are bounded and the path-following generalized error

vector $x_{\text{pf}}(t)$ converges to a neighborhood of the origin, regardless of what the temporal assignment of the mission is –as long as it is physically feasible.

To solve the path-following problem described above, we first let the rate of progression of point P along the path be governed by

$$\dot{\ell} = (v \vec{w}_1 + K_\ell p_F) \cdot \vec{t}, \quad (12)$$

where K_ℓ is a positive constant gain. Then, the rate commands $q_c(t)$ and $r_c(t)$ given by

$$\begin{bmatrix} q_c \\ r_c \end{bmatrix} \triangleq \Pi_R \tilde{R}^\top \left(R_F^D \{ \omega_{F/I} \}_F + \{ \omega_{D/F} \}_D \right) - 2K_{\tilde{R}} e_{\tilde{R}}, \quad (13)$$

where $K_{\tilde{R}}$ is also a positive gain, drive the path-following generalized error vector $x_{\text{pf}}(t)$ to zero with a guaranteed rate of convergence. More precisely, it can be shown that, if the speed of the vehicle satisfies $0 < v_{\min} \leq v(t) \leq v_{\max}$, then the origin of the kinematic error equations in (10)-(11) with the controllers $q_c(t)$ and $r_c(t)$ defined in (13) is locally exponentially stable. A formal statement of this result can be found in [57], while a detailed proof is provided in [55].

The path-following control law above relies on the use of the Special Orthogonal group $\text{SO}(3)$ in the formulation of the attitude control problem. This formulation avoids the geometric singularities and complexities that appear when dealing with local parameterizations of the vehicles attitude, and leads thus to a singularity-free path-following control law.

Also, we note that the choice of the characterizing distance d in the definition of the auxiliary frame \mathcal{D} can be used to adjust the rate of convergence for the path-following closed-loop system. A large parameter d reduces the penalty for cross-track position errors, which results in a slow rate of convergence of the UAV to the path. On the other hand, small values of d allow for a high rate of convergence, which however might result in oscillatory path-following

behavior. Figure 7 illustrates this point. Further insights into this path-following control algorithm can be found in [57].

The solution to the path-following problem for the i th vehicle presented above is independent of the desired speed profile $v_{d,i}(\cdot)$, and uses only local measurements for feedback. The path-following control laws $q_c(t)$ and $r_c(t)$ represent outer-loop guidance commands that are to be tracked by the vehicle to ensure the safety and success of the cooperative mission. In this sense, this solution for path-following control departs from standard backstepping techniques in that the final path-following control law makes explicit use of the existing autopilot and retains its stabilizing properties and tracking capabilities.

Time-Critical Cooperative Path Following for Multiple Vehicles

The previous section presents a control algorithm that solves the path-following problem for a single vehicle and an arbitrary speed profile by using a control strategy in which the vehicle's attitude control effectors are used to follow a virtual target running along the path. We now address the problem of time-critical cooperative path-following control of multiple vehicles. For this purpose, the speeds of the vehicles are adjusted based on coordination information exchanged among the vehicles over a time-varying network. In particular, the outer-loop coordination control law is intended to provide a correction to the desired speed profile $v_{d,i}(\cdot)$ obtained in the trajectory-generation step, and to generate a total speed command $v_{c,i}(t)$. This speed command is then to be tracked by the i th vehicle to achieve coordination in time.

To solve this coordination problem, we first formulate it as a *consensus problem*, in which the objective of the fleet of vehicles is to reach an agreement on some distributed variables of

interest. An overview of consensus algorithms and their application to cooperative control of networked multi-agent systems can be found in [58].

Limited communication among vehicles

To enforce the temporal constraints that must be met in real time to coordinate the entire fleet of vehicles, an appropriate coordination variable needs to be defined for each vehicle that captures the objective of the cooperative mission, in our case, simultaneous arrival of all the vehicles at their final destinations.

For this purpose, we start by defining $\ell'_{d,i}(t_d)$ as the desired normalized curvilinear abscissa of the i th UAV along its corresponding path at the desired mission time t_d , which is given by

$$\ell'_{d,i}(t_d) \triangleq \frac{1}{\ell_{fi}} \int_0^{t_d} v_{d,i}(\sigma_t) d\sigma_t, \quad (14)$$

with ℓ_{fi} and $v_{d,i}(\cdot)$ being, respectively, the length of the path and the desired speed profile corresponding to the i th UAV. The trajectory-generation algorithm ensures that the desired speed profiles $v_{d,i}(\cdot)$ satisfy the feasibility conditions

$$0 < v_{\min} \leq v_{d,i}(\cdot) \leq v_{\max}, \quad i = 1, \dots, n. \quad (15)$$

Hence, from the definition of $\ell'_{d,i}(t_d)$ and the bounds in (15), it follows that $\ell'_{d,i}(t_d)$ is a strictly increasing continuous function of t_d mapping $[0, t_d^*]$ into $[0, 1]$, and satisfying $\ell'_{d,i}(0) = 0$ and $\ell'_{d,i}(t_d^*) = 1$. We also define $\eta_i : [0, 1] \rightarrow [0, t_d^*]$ to be the inverse function of $\ell'_{d,i}(t_d)$, $t_d \in [0, t_d^*]$. Clearly, $\eta_i(\cdot)$ is also a strictly increasing continuous function of its argument. Then, letting $\ell'_i(t)$ be the normalized curvilinear abscissa at time t of the i th virtual target vehicle running along

its path, defined as

$$\ell'_i(t) \triangleq \frac{\ell_i(t)}{\ell_{fi}},$$

we define the time-dependent variables

$$\xi_i(t) \triangleq \eta_i(\ell'_i(t)), \quad i = 1, \dots, n. \quad (16)$$

From this definition, it follows that $\xi(t) \in [0, t_d^*]$, and therefore this variable can be seen as a *virtual time* that characterizes the status of the mission for the i th UAV at time t in terms of the desired mission time t_d .

We note that, for any two vehicles i and j , if $\xi_i(t) = \xi_j(t) = t'_d$ at a given time t , then $\ell'_i(t) = \ell'_{d,i}(t'_d)$ and $\ell'_j(t) = \ell'_{d,j}(t'_d)$, which implies that at time t the target vehicles corresponding to UAVs i and j have the desired relative position along the path at the desired mission time t'_d . Clearly, if $\xi_i(t) = \xi_j(t)$ for all $t \geq 0$, then the i th and j th virtual target vehicles maintain the desired relative position along the path at all times and, in particular, these two target vehicles arrive at their final destinations at the same time, which does not necessarily correspond to the desired mission duration t_d^* . Also, in the case of collision avoidance in time, if $\xi_i(t) = \xi_j(t)$ for all $t \geq 0$, then the solution to the path-generation problem ensures that the virtual targets i and j do not collide. Moreover, if the i th virtual target travels at the desired speed for all times in the interval $[0, t]$, that is, $\dot{\ell}_i(\tau) = v_{d,i}(\tau)$ for all $\tau \in [0, t]$, then we have that $\ell_i(\tau) = \ell_{d,i}(\tau)$ for all $\tau \in [0, t]$, which implies that $\xi_i(\tau) = \tau$ (or equivalently, that $\dot{\xi}_i(\tau) = 1$) for all $\tau \in [0, t]$. This set of properties makes the variables $\xi_i(t)$ an appropriate metric for vehicle coordination, and therefore we refer to them as *coordination states*. We notice that the use of these specific coordination variables is motivated by the work in [59].

To meet the desired temporal assignments of the cooperative mission, the coordination

states are to be exchanged among the UAVs over the supporting communications network. Next, we use tools and facts from *algebraic graph theory* to model the information exchange over the time-varying network as well as the constraints imposed by the communication topology. Key concepts and details on algebraic graph theory can be found in [60].

First, in order to account for the communication constraints imposed by the inter-vehicle network, we assume that the i th UAV can only exchange information with a neighboring set of vehicles, denoted here by G_i . We also assume that the communication between two UAVs is bidirectional and that the information is transmitted continuously with no delays. Moreover, since the flow of information among vehicles may be severely restricted, either for security reasons or because of tight bandwidth limitations, we impose the constraint that each vehicle only exchanges its coordination state $\xi_i(t)$ with its neighbors. Finally, we assume that the connectivity of the communications graph $\Gamma(t)$ that captures the underlying bidirectional communications network topology of the fleet at time t satisfies the persistency of excitation (PE)-like condition

$$\frac{1}{n} \frac{1}{T} \int_t^{t+T} Q L(\tau) Q^\top d\tau \geq \mu \mathbb{I}_{n-1}, \quad \text{for all } t \geq 0, \quad (17)$$

where $L(t)$ is the Laplacian of the graph $\Gamma(t)$, and Q is an $(n-1) \times n$ matrix such that $Q1_n = 0$ and $QQ^\top = \mathbb{I}_{n-1}$, with 1_n being the vector in \mathbb{R}^n whose components are all 1. The parameters $T, \mu > 0$ characterize the QoS of the communications network, which in the context of this article represents a measure of the level of connectivity of the communications graph.

The PE-like condition (17) requires only the communications graph $\Gamma(t)$ to be connected in an integral sense, not pointwise in time. In fact, the graph may be disconnected during some interval of time or may even fail to be connected for the entire duration of the mission. Similar type of conditions can be found, for example, in [61] and [62].

Coordination control law

Using the above formulation, and given a fleet of n vehicles supported by an inter-vehicle communications network and a set of desired 3D time trajectories $\Phi_{d,i}(t_d)$, the problem of *time-critical cooperative path following* can be formulated as that of designing feedback control laws for pitch rate $q(t)$, yaw rate $r(t)$, and speed $v(t)$ such that all closed-loop signals are bounded, for each vehicle i , $i \in \{1, \dots, n\}$, the path-following generalized error vector $x_{pf,i}(t)$ converges to a neighborhood of the origin, and for each pair of vehicles i and j , $i, j \in \{1, \dots, n\}$, the coordination error $|\xi_i(t) - \xi_j(t)|$ converges to a neighborhood of the origin, guaranteeing quasi-simultaneous time of arrival and ensuring collision-free maneuvers.

We start by noting that the evolution of the i th coordination state is given by [55]

$$\dot{\xi}_i(t) = \frac{\dot{\ell}_i(t)}{v_{d,i}(\xi_i(t))}.$$

Recalling from the solution to the path-following problem that the evolution of the i th virtual target vehicle along the path is given by

$$\dot{\ell}_i = (v_i \vec{w}_{1,i} + K_\ell p_{F,i}) \cdot \vec{t}_i,$$

where for simplicity we keep K_ℓ without indexing and drop the dependency of the various variables on t , the dynamics of the i th coordination state can be rewritten as

$$\dot{\xi}_i = \frac{(v_i \vec{w}_{1,i} + K_\ell p_{F,i}) \cdot \vec{t}_i}{v_{d,i}(\xi_i)}. \quad (18)$$

Then, to solve the time-coordination problem, we use dynamic inversion and define the speed command for the i th vehicle as

$$v_{c,i} \triangleq \frac{u_{\text{coord},i} v_{d,i}(\xi_i) - K_\ell p_{F,i} \cdot \vec{t}_i}{\vec{w}_{1,i} \cdot \vec{t}_i}, \quad (19)$$

where $u_{\text{coord},i}(t)$ is a coordination control law, yet to be defined. With this speed command, the partially closed-loop coordination dynamics for the i th target vehicle can be rewritten as

$$\dot{\xi}_i = u_{\text{coord},i} + \frac{e_{v,i}}{v_{d,i}(\xi_i)} \vec{w}_{1,i} \cdot \vec{t}_i, \quad (20)$$

where $e_{v,i}(t) \triangleq v_i(t) - v_{c,i}(t)$ denotes the velocity tracking error for the i th vehicle.

Recall now that each vehicle is allowed to exchange only its coordination parameter $\xi_i(t)$ with its neighbors G_i , which are defined by the possibly time-varying communications topology. To observe this constraint, we propose the decentralized coordination law

$$u_{\text{coord},1}(t) = -a \sum_{j \in G_1} (\xi_1(t) - \xi_j(t)) + 1, \quad (21)$$

$$u_{\text{coord},i}(t) = -a \sum_{j \in G_i} (\xi_i(t) - \xi_j(t)) + \chi_{I,i}(t), \quad i = 2, \dots, n, \quad (22)$$

$$\dot{\chi}_{I,i}(t) = -b \sum_{j \in G_i} (\xi_i(t) - \xi_j(t)), \quad \chi_{I,i}(0) = 1, \quad i = 2, \dots, n, \quad (23)$$

where vehicle 1 is elected as the formation leader, and a and b are positive adjustable coordination control gains. Note that the coordination control law has a proportional-integral structure, which provides disturbance rejection capabilities. Moreover, we note that the vehicles exchange information only about the corresponding virtual targets, rather than exchanging their own state information. The importance of this observation can hardly be overemphasized. The benefits of using “virtual information” in consensus problems are illustrated in [63].

Convergence properties of the combined cooperative path-following control laws

Figure 8 shows the complete time-critical cooperative path-following closed-loop control architecture for the i th vehicle, including the nonlinear path-following algorithm and the

decentralized coordination control law. With this approach, the overall cooperative control architecture presented in this article exhibits a multiloop control structure in which an inner-loop controller stabilizes the vehicle dynamics, while guidance outer-loop controllers are designed to control the vehicle kinematics, providing path-following and time-critical cooperative capabilities.

At a kinematic level, it is proven in [55] that, if the connectivity of the communications graph verifies the PE-like condition (17) and the initial conditions are within a given domain of attraction, then there exist control gains for the path-following control law (12)-(13) and the coordination control law (19)-(23) that ensure, first, that the path-following generalized error vector $x_{\text{pf},i}(t)$ of each vehicle converges exponentially fast to zero; second, that for each pair of vehicles i and j , $i, j \in \{1 \dots, n\}$, the coordination error $|\xi_i(t) - \xi_j(t)|$ also converges to zero exponentially fast; and third, that the speed of each vehicle satisfies $v_{\min} \leq v_i(t) \leq v_{\max}$ for all $t \geq 0$.

Additionally, the results in [55] also present explicit tracking performance bounds for the inner-loop controller that ensure stability of the overall time-critical cooperative path-following control system. In particular, for the case of non-ideal inner-loop tracking, and provided that the inner-loop performance bounds are satisfied, the path-following generalized error vectors and the coordination errors can be proven to converge exponentially fast to a neighborhood of zero and to be *uniformly ultimately bounded*. Furthermore, the ultimate bounds are proportional to the inner-loop angular-rate and speed tracking performance bounds; see [55, Theorem 1].

Lemma 3 in [55] also demonstrates that the QoS of the network, characterized by the parameters T and μ , limits the achievable guaranteed rate of convergence for the coordination control loop. These results also imply that, as the parameter T goes to zero and the communica-

tions graph becomes thus connected pointwise in time, the convergence rate can be set arbitrarily fast by increasing the coordination control gains. This fact is consistent with results obtained in previous work on time-critical cooperative path-following control; see [64, Lemma 2].

Finally, we note that successful execution of the mission requires that the design of the overall cooperative path-following control algorithm with the inner-loop autopilot provide the level of performance considered for trajectory generation; see optimization problems (6) and (7). In the context of this article, the performance of the path-following and coordination controllers can be characterized, for example, in terms of the inner-loop tracking performance bounds or the ultimate bounds for path-following and coordination errors.

\mathcal{L}_1 Adaptive Control for Autopilot Augmentation

As shown in [55, Theorem 1], safety and success of the cooperative time-critical mission relies on the fact that each vehicle can track precisely the angular-rate and speed commands provided by the outer-loop path-following and coordination algorithms. For the missions of interest, typical off-the-shelf autopilots are capable of providing uniform performance across the flight envelope of small UAVs while operating in nominal conditions. However, these commercial autopilots may fail to provide adequate performance across the operational envelope in the event of actuator failures, vehicle damage, or in the presence of adverse environmental disturbances. Under these unfavorable circumstances, adaptive augmentation loops are seen as an appealing technology that can improve vehicle performance.

In this section, we propose the implementation of \mathcal{L}_1 adaptive controllers for control augmentation of onboard commercial autopilots. The theory of \mathcal{L}_1 adaptive control enables the

design of robust adaptive control architectures using fast adaptation schemes, which results in predictable, repeatable, verifiable, and safe adaptive flight control algorithms. The key feature of \mathcal{L}_1 adaptive control is the decoupling of the adaptation loop from the control loop, which enables fast adaptation without sacrificing robustness. Fast adaptation allows for compensation of the undesirable effects of rapidly varying uncertainties and significant changes in the system dynamics, and is also critical to achieve a predictable and consistent response of the closed-loop adaptive system. The performance and robustness properties of \mathcal{L}_1 adaptive control are described in detail in [65], while insights into the application of \mathcal{L}_1 adaptive control to safety-critical flight control are presented in [66].

In [67, 68], for example, we present an \mathcal{L}_1 adaptive control architecture for autopilot augmentation that retains the properties of the onboard commercial autopilot, and adjusts the autopilot commands only when the tracking performance degrades or the mission effectiveness is reduced. Figure 9 shows the inner-loop control architecture considered in [67, 68], with the adaptive augmentation loop wrapped around the autopilot. In this setup, the adaptive controller uses angular-rate and speed measurements to modify the commands generated by the outer-loop algorithms, which are then sent to the autopilot as references to be tracked. This structure for autopilot augmentation does not require any modifications to the autopilot itself, and at the same time it does not use internal states of the autopilot for control design purposes. In particular, the control architecture illustrated in Figure 9 is the one used in the cooperative road-search mission scenario discussed later in the next section.

Cooperative Road Search with Multiple Unmanned Aerial Vehicles

In this section we discuss flight test results for a cooperative road-search mission that show the efficacy of the multi-vehicle cooperative control framework presented in this article. Cooperative path-following missions involving multiple UAVs were flown for the first time at Camp Roberts, CA, in November 2009, and then demonstrated four more times at the same location in February, May, July, and November of 2010. The flight tests were performed during the quarterly run Tactical Network Topology field experiments conducted through the Field Experimentation Cooperative Program, which is being led by the U.S. Special Operations Command and the Naval Postgraduate School (NPS) [69, 70]. These results verify the main theoretical claims of the cooperative control algorithm presented in this article and demonstrate the feasibility of the onboard implementation of the algorithms and the validity of the approach.

Mission description

Today's operational environments face a growing need for up-to-date satellite-like imagery, with enough resolution to detect humans, weapons, and other potential threats. While accurate high-resolution imagery is traditionally provided by satellites and high-end aerial intelligence surveillance and reconnaissance platforms, these assets are not always available to the end-user due to time-of-day, visibility, or mission priority. In such cases, the use of small tactical UAVs outfitted with the ability to capture actionable, high-resolution, geo-referenced imagery and full motion video, represents an economical and expeditious alternative. Moreover, the fact that the UAVs can deliver the information to the end-user in seconds or minutes, rather than hours or days, can potentially revolutionize the way we operate and save lives.

One of the applications that motivates the use of multiple cooperative UAVs and poses several challenges to systems engineers, both from a theoretical and practical standpoint, is automatic road search for improvised explosive device detection; see Figure 10. The mission is initiated by a minimally trained user who scribbles a path on a digital map, generating a precise continuous ground-track for the airborne sensors to follow. This ground-track is then transmitted over the network to a fleet of small tactical UAVs equipped with complementary visual sensors. Decentralized optimization algorithms autonomously generate feasible flight trajectories that maximize road coverage and account for sensor capabilities –field of view, resolution, and gimbal constraints– as well as inter-vehicle and ground-to-air communications limitations. The fleet of UAVs then starts the cooperative road search. During this phase, the information obtained from the sensors mounted onboard the UAVs is shared over the network and retrieved by remote users in near real time. The explosive device detection can thus be done remotely on the ground, based on in-situ imagery data delivered over the network.

In this particular mission scenario, a robust cooperative control algorithm for the fleet of UAVs can improve mission performance and provide reliable target discrimination, by effectively combining the capabilities of the onboard sensors [71]. In fact, flying in a coordinated fashion is what allows, for example, to maximize the overlap of the fields of view (FOVs) of multiple sensors and to take full advantage of complementary sensors.

Airborne system architecture

The small tactical UAVs employed in this particular mission are two SIG Rascals 110 operated by NPS; see Figure 11. The two UAVs have the same avionics and the same

instrumentation onboard, the only difference being the vision sensors. The first UAV has a bank-compensated high-resolution 12-MPx imagery camera, while the second UAV has a full-motion video camera suspended on a pan-tilt gimbal. Due to payload constraints, each UAV is allowed to carry only one camera at a time, and therefore the two cameras need to be mounted on different platforms. The rest of the onboard avionics, common to both platforms, includes two PC-104 industrial embedded computers [72] assembled in a stack, a wireless Mobile Ad-hoc Network (MANET) link [73], and the Piccolo Plus autopilot [74] with its dedicated 900-MHz command and control channel. Details of the complete airborne network-centric architecture are presented in Figure 12.

The first PC-104 computer acts as a secondary autopilot controller, running the cooperative-control algorithms in hard real time at 100 Hz and directly communicating with the Piccolo Plus autopilot at 50 Hz over a dedicated serial link. This connection efficiently eliminates communication delays between the outer-loop control algorithms and the autopilot. The second PC-104 is a mission management computer that implements a set of non-real-time routines enabling onboard preprocessing and retrieval of the sensory data –high-resolution imagery or video– in near real time over the network. Integration of the MANET link allows for robust transparent inter-vehicle and ground communication, which is needed for both the coordination algorithms and the expedited sensory data delivery to a remote mission operator. In fact, the MANET link provides “any-to-any” connectivity capability, allowing every node –vehicle or ground station– to communicate directly with every other node. Moreover, information about the connectivity of the entire network can be retrieved in near real time. Details on the flight test architecture, the supporting network infrastructure, and the management of the communication bandwidth for coordination control and data dissemination can be found in [73, 75].

Flight Test Results

We now present flight test results for a cooperative road-search mission executed by the two SIG Rascals. The objective of the mission is to detect a target moving along a given road and, if detection occurs, to collect information about the target. This information is then to be shared over a MANET link so that it can be retrieved by remote mission operators in near real time. Success of the mission relies on the ability to overlap the footprint of the FOVs of the two cameras along the road, which increases the probability of target detection [71]. Next, we provide details about the execution of this coordinated road-search mission, which we divide in four consecutive phases, namely, initialization, transition, road search, and vision-based target tracking. The description is supported by one of the flight tests results performed during a Tactical Network Testbed field experiment at Camp Roberts, CA; see figures 13-16.

In the *initialization phase*, an operator specifies on a digital map the road of interest. Then, a centralized optimization algorithm generates road-search suboptimal paths and desired speed profiles for the two UAVs that explicitly account for UAV dynamic constraints, collision-avoidance constraints, and mission-specific constraints such as inter-vehicle and vehicle-to-ground communications limitations as well as sensory capabilities. In particular, the trajectory-generation algorithm is designed to maximize the overlap of the footprints of the FOVs of the high-resolution camera and the full-motion video during the road search. In addition to the road-search paths and the corresponding desired speed profiles, the outcome of the trajectory-generation algorithm includes a *sensor trajectory* on the ground to be followed by the vision sensors. The two road-search paths and the sensor path, along with the three corresponding speed profiles, are then transmitted to the UAVs over the MANET link.

In the *transition phase*, the two UAVs fly from their standby starting positions to the initial points of the respective road-search paths. For this purpose, decentralized optimization algorithms generate feasible collision-free 3D trajectories to ensure that the two UAVs arrive at the initial points of the road-search paths at the same time. Once these transition trajectories are generated, the two vehicles start operating in cooperative path-following mode. From that moment on, the UAVs follow the transition paths while adjusting their speeds based on coordination information exchanged over the MANET link in order to achieve simultaneous arrival at the starting point of the road-search paths. The transition and road-search spatially-deconflicted paths obtained for this particular mission scenario, together with the corresponding desired speed profiles and the path separations, are shown in Figure 13. Figure 14 illustrates the performance of the coordination control algorithm during the transition phase of the mission.

The third phase addresses the *cooperative road-search mission* itself, in which the two UAVs follow the road-search paths generated in the initialization phase while adjusting their speeds to ensure the required overlap of the FOV footprints of the cameras. In this phase, a target vehicle running along the sensor path is virtually implemented on one of the UAVs. For this road-search mission, a natural choice for this sensor path is the road itself, and this virtual vehicle determines thus the spot of the road being observed by the vision sensors mounted onboard the UAVs at a given time. This virtual vehicle is indeed used as a leader in the coordination algorithm, and its speed is also adjusted, based on the coordination states of the two UAVs. The coordination state of this virtual vehicle is also transmitted over the tactical network and used in the coordination control laws of the two “real” vehicles. The performance of the cooperative path-following control algorithm is illustrated in Figure 15. For this particular mission scenario, the coordination errors remain below 7% during the entire duration of the road search, while

the path-following cross-track errors converge to a 3 m tube around the desired spatial paths.

Finally, when a target is detected on the road, the two UAVs immediately switch to *cooperative vision-based tracking* mode. In this phase, the UAVs track the target by means of guidance loops that use visual information for feedback, while simultaneously providing in-situ imagery for precise geo-location of the point of interest. During this target-tracking phase, a coordination algorithm ensures that the two UAVs keep a predefined phase separation of $\frac{\pi}{2}$ rad while “orbiting” around the target. This coordination algorithm uses the coordination control law described in previous sections to adjust the orbiting speed of the UAVs, with the main difference that *phase on orbit* is now used as a coordination state, rather than virtual time. Besides collision avoidance, cooperation through phase-on-orbit coordination allows for several additional benefits, including reduced sensitivity to target escape maneuvers [76] and possible extraction of 3D information from 2D images [77]. The performance of the cooperative path-following control algorithm is illustrated in Figure 16, which shows the trajectories of the two UAVs while tracking the target as well as the phase-coordination error between the UAVs. Details about the vision-based guidance loop used in this phase can be found in [78].

Flight test summary and accessory mission outcomes

The results presented above illustrate the benefits of using cooperative control based on the algorithms described in this article when dealing with missions involving multiple vehicles. Such cooperative strategies ensure collision-free maneuvers, and efficiently combine heterogeneous information provided by complementary sensors.

To visually illustrate the effect of time-critical cooperation among the UAVs, Figure 17

presents a mosaic of four consecutive high-resolution images taken during a flight experiment. In this experiment, the road-search paths are intentionally separated by altitude and optimized such that, if the coordination algorithm adequately adjusts the speed of the two UAVs, then the UAV flying at lower altitude is expected to be continuously present in the FOV of the camera flying at higher altitude. The figure schematically represents the progression of the lines of sight connecting the two cameras with the virtual target vehicle running along the sensor path. Time-coordination ensures that cameras observe the same spot on the road and thus maximize the overlap of the footprints of their FOVs, which is critical to provide reliable target discrimination.

Also, in order to illustrate possible accessory mission outcomes, Figure 18 presents examples of imagery data utilization. In Figure 18a, for example, the 3D geo-referenced model of the operational environment is built from 2D high-resolution frames using proprietary technology [79]. In Figure 18b, a geo-referenced mosaic is obtained in near real time from high-resolution frames sent by one of the UAVs through the MANET link while in mission [77].

In summary, the results presented above demonstrate the benefits of the onboard integration of the nonlinear path-following and coordination algorithms as well as \mathcal{L}_1 adaptation. During the flight experiments, the required control commands never exceeded the limits defined for the UAV in traditional waypoint navigation mode. At the same time, the achieved functionality of the UAV following 3D curves in an inertial space outperforms the conventional waypoint navigation method typically implemented on off-the-shelf commercial autopilots. These results provide also a roadmap for further development and onboard implementation of advanced cooperative algorithms, opening new frontiers for UAV operations.

Conclusions

In this article, we describe an approach to cooperative control of multiple autonomous systems for time-critical missions. The approach presented applies to teams of heterogeneous systems and does not necessarily lead to swarming behavior, which is unsuitable for many of the mission scenarios envisioned in this article. The methodology proposed unfolds in three basic steps. Initially, each vehicle is assigned a feasible path with a desired speed profile that together satisfy the mission requirements and the vehicle dynamic constraints, while ensuring collision-free maneuvers. Then, a path-following algorithm ensures that every vehicle follows its own path independently of the temporal assignments of the mission. Finally, the vehicles coordinate their position along the path with the remaining vehicles engaged in the mission by exchanging coordination information over the communications network. These three steps are accomplished by judiciously decoupling space and time in the formulation of the trajectory-generation, path-following, and time-coordination problems, and by relying on the existing inner-loop controllers for nominal control of the autonomous systems. These inner-loop controllers are augmented with \mathcal{L}_1 adaptive loops, which ensure robust performance in the event of failures, vehicle damage, or in the presence of adverse environmental disturbances. As a result, the described work yields a systematic framework for integration of various tools and concepts from a broad spectrum of disciplines, leading to a streamlined design procedure for cooperative path-following control. The benefits of this approach have been demonstrated in a cooperative road-search mission scenario involving multiple unmanned aerial vehicles. The framework presented has also been tested on cooperative missions involving multiple heterogeneous autonomous marine vehicles operating in uncertain environments; see [80, 81] and references therein for details about these experiments.

References

- [1] D. J. Stilwell and B. E. Bishop, "Platoons of underwater vehicles," *IEEE Control Systems Magazine*, vol. 20, no. 6, pp. 45–52, 2000.
- [2] F. Giulietti, L. Pollini, and M. Innocenti, "Autonomous formation flight," *IEEE Control Systems Magazine*, vol. 20, no. 6, pp. 34–44, 2000.
- [3] R. W. Beard, J. Lawton, and F. Y. Hadaegh, "A coordination architecture for spacecraft formation control," *IEEE Transactions on Control System Thechnology*, vol. 9, no. 6, pp. 777–790, 2001.
- [4] 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, 2001.
- [5] M. Pachter, J. J. D'Azzo, and A. W. Proud, "Tight formation flight control," *Journal of Guidance, Control and Dynamics*, vol. 24, no. 2, pp. 246–254, 2001.
- [6] J. R. T. Lawton, R. W. Beard, and B. J. Young, "A decentralized approach to formation maneuvers," vol. 19, no. 6, pp. 933–941, 2000.
- [7] 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, 2005, pp. 2983–2988.
- [8] 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, 2004.

- [9] F. Zhang and P. S. Krishnaprasad, "Coordinated orbit transfer for satellite clusters," *Annals of the New York Academy of Sciences*, vol. 1017, pp. 112–137, 2004.
- [10] A. Rahmani, M. Mesbahi, and F. Y. Hadaegh, "Optimal balanced-energy formation flying maneuvers," *Journal of Guidance, Control and Dynamics*, vol. 29, no. 6, pp. 1395–1403, 2006.
- [11] 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, 2006, pp. 4906–4913.
- [12] I. Kaminer, O. A. Yakimenko, V. Dobrokhodov, A. M. Pascoal, N. Hovakimyan, V. V. Patel, C. Cao, and A. Young, "Coordinated path following for time-critical missions of multiple UAVs via \mathcal{L}_1 adaptive output feedback controllers," in: *AIAA Guidance, Navigation and Control Conference*, AIAA-2007-6409, Hilton Head, SC, 2007.
- [13] A. P. Aguiar, I. Kaminer, R. Ghabcheloo, A. M. Pascoal, E. Xargay, N. Hovakimyan, C. Cao, and V. Dobrokhodov, "Coordinated path following of multiple UAVs for time-critical missions in the presence of time-varying communication topologies," in: *IFAC World Congress*, Seoul, South Korea, 2008.
- [14] M. Egerstedt and X. Hu, "Formation constrained multi-agent control," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 6, pp. 947–951, 2001.
- [15] W. Ren and N. Sorensen, "Distributed coordination architecture for multi-robot formation control," *Robotics and Autonomous Systems*, vol. 56, no. 6, pp. 324–333, 2008.
- [16] R. Ghabcheloo, A. M. Pascoal, C. Silvestre, and I. Kaminer, "Non-linear co-ordinated path following control of multiple wheeled robots with bidirectional communication

- constraints,” *International Journal of Adaptive Control and Signal Processing*, vol. 21, no. 2–3, pp. 133–157, 2007.
- [17] 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, 2002, pp. 1699–1704.
 - [18] 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*, vol. 1, Orlando, FL, 2001, pp. 964–969.
 - [19] I.-A. F. Ihle, “Coordinated control of marine craft,” PhD thesis, Trondheim, Norway: Norwegian University of Science and Technology, 2006.
 - [20] E. Kyrkjebø, “Motion coordination of mechanical systems: Leader-follower synchronization of Euler-Lagrange systems using output feedback control,” PhD thesis, Trondheim, Norway: Norwegian University of Science and Technology, 2007.
 - [21] F. Zhang, D. M. Fratantoni, D. A. Paley, J. M. Lund, and N. E. Leonard, “Control of coordinated patterns for ocean sampling,” *International Journal of Control*, vol. 80, no. 7, pp. 1186–1199, 2007.
 - [22] T. B. Curtin, J. G. Bellingham, J. Catipovic, and D. Webb, “Autonomous oceanographic sampling networks,” *Oceanography*, vol. 6, no. 3, pp. 86–94, 1993.
 - [23] W. B. Dunbar and R. M. Murray, “Distributed receding horizon control for multi-vehicle formation stabilization,” *Automatica*, vol. 42, no. 4, pp. 549–558, 2006.
 - [24] D. V. Dimarogonas, S. G. Loizou, K. J. Kyriakopoulos, and M. M. Zavlanos, “A feedback stabilization and collision avoidance scheme for multiple independent non-point agents,” *Automatica*, vol. 42, no. 2, pp. 229–243, 2006.

- [25] D. J. Klein, C. Matlack, and K. A. Morgansen, “Cooperative target tracking using oscillator models in three dimensions,” in: *American Control Conference*, New York, NY, 2007, pp. 2569–2575.
- [26] N. Moshtagh and A. Jadbabaie, “Distributed geodesic control laws for flocking of non-holonomic agents,” *IEEE Transactions on Automatic Control*, vol. 52, no. 4, pp. 681–686, 2007.
- [27] Z. Lin, “Coupled dynamic systems: From structure towards stability and stabilizability,” PhD thesis, Toronto, Canada: University of Toronto, 2006.
- [28] S. Nair and N. E. Leonard, “State agreement for continuous-time coupled nonlinear systems,” *SIAM Journal on Control and Optimization*, vol. 47, no. 4, pp. 661–683, 2008.
- [29] Q.-C. Pham and J.-J. E. Slotine, “Stable concurrent synchronization in dynamic system networks,” *Neural Networks*, vol. 20, no. 1, pp. 62–77, 2007.
- [30] I. Kaminer, A. M. Pascoal, E. Hallberg, and C. Silvestro, “Trajectory tracking for autonomous vehicles: An integrated approach to guidance and control,” *Journal of Guidance, Control and Dynamics*, vol. 21, no. 1, pp. 29–38, 1998.
- [31] Y. Kim and M. Mesbahi, “On maximizing the second smallest eigenvalue of state-dependent graph Laplacian,” *IEEE Transactions on Automatic Control*, vol. 51, no. 1, pp. 116–120, 2006.
- [32] A. P. Aguiar, J. P. Hespanha, and P. V. Kokotović, “Performance limitations in reference tracking and path following for nonlinear systems,” *Automatica*, vol. 44, no. 3, pp. 598–610, 2008.
- [33] J. T. Betts, “Survey of numerical methods for trajectory optimization,” *Journal of Guidance, Control and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

- [34] C. Goerzen, Z. Kong, and B. Mettler, “A survey of motion planning algorithms from the perspective of autonomous UAV guidance,” *Journal of Intelligent and Robotic Systems*, vol. 57, no. 1, pp. 65–100, 2010.
- [35] T. W. McLain and R. W. Beard, “Coordination variables, coordination functions, and cooperative timing missions,” *Journal of Guidance, Control and Dynamics*, vol. 28, no. 1, pp. 150–161, 2005.
- [36] H.-L. Choi, A. K. Whitten, and J. P. How, “Decentralized task allocation for heterogeneous teams with cooperative constraints,” in: *American Control Conference*, Baltimore, 2010, pp. 3057–3062.
- [37] H. J. Pesch, “Real-time computation of feedback controls for constrained optimal control problems. Part 1: Neighboring extremals,” *Optimal Control Applications and Methods*, vol. 10, no. 2, pp. 129–145, 1989.
- [38] H. J. Pesch, “Real-time computation of feedback controls for constrained optimal control problems. Part 2: A correction method based on neighboring extremals,” *Optimal Control Applications and Methods*, vol. 10, no. 2, pp. 147–171, 1989.
- [39] G. Elnagar, M. A. Kazemi, and M. Razzaghi, “The pseudospectral Legendre method for discretizing optimal control problems,” *IEEE Transactions on Automatic Control*, vol. 40, no. 10, pp. 1793–1796, 1995.
- [40] I. M. Ross and F. Fahroo, “Legendre pseudospectral approximations of optimal control problems,” in: vol. 295, *Lecture Notes in Control and Information Sciences*, New York, NY: Springer-Verlag New York, Inc., 2003, pp. 327–342.

- [41] M. B. Milam, K. Mushambi, and R. M. Murray, “A new computational approach to real-time trajectory generation for constrained mechanical systems,” in: *IEEE Conference on Decision and Control*, vol. 1, Sydney, Australia, 2000, pp. 845–851.
- [42] M. B. Milam, “Real-time optimal trajectory generation for constrained dynamical systems,” PhD thesis, Pasadena, CA: California Institute of Technology, 2003.
- [43] S. M. LaValle, *Rapidly-exploring random trees: A new tool for path planning*. Tech. rep. 98-11, Ames, IA: Iowa State University, 1998.
- [44] S. M. LaValle, “Planning algorithms,” in: Cambridge, UK: Cambridge University Press, 2006, chap. 5: Sampling-Based Motion Planning.
- [45] L. E. Kavraki, P. Švestka, J.-C. Latombe, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configurations spaces,” *Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [46] 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, 2000.
- [47] A. M. Geoffrion, “Generalized Benders decomposition,” *Journal of Optimization Theory and Applications*, vol. 10, no. 4, pp. 237–260, 1972.
- [48] B. Borchers and J. E. Mitchell, “An improved branch and bound algorithm for mixed integer nonlinear programming,” *Computers and Operations Research*, vol. 21, no. 4, pp. 359–367, 1994.
- [49] M. A. Duran and I. E. Grossmann, “An outer-approximation algorithm for a class of mixed-integer nonlinear programs,” *Mathematical Programming*, vol. 36, no. 3, pp. 307–339, 1986.

- [50] T. J. V. Roy, “Cross decomposition for mixed integer programming,” *Mathematical Programming*, vol. 25, no. 1, pp. 46–63, 1983.
- [51] 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, 2003, pp. 1387–1392.
- [52] A. Micaelli and C. Samson, *Trajectory tracking for unicycle-type and two-steering-wheels mobile robot*, tech. rep. 2097, Sophia-Antipolis, France: INRIA, 1993.
- [53] R. L. Bishop, “There is more than one way to frame a curve,” *The American Mathematical Monthly*, vol. 82, no. 3, pp. 246–251, 1975.
- [54] A. J. Hanson and H. Ma, *Parallel transport approach to curve framing*, tech. rep., Indiana University Compute Science Department, 1995.
- [55] E. Xargay, I. Kaminer, A. M. Pascoal, N. Hovakimyan, V. Dobrokhodov, V. Cichella, A. P. Aguiar, and R. Ghabcheloo, “Time-critical cooperative path following of multiple UAVs over time-varying networks,” *Journal of Guidance, Control and Dynamics* Submitted in 2011.
- [56] T. Lee, M. Leok, and N. H. McClamroch, “Control of complex maneuvers for a quadrotor uav using geometric methods on $se(3)$,” *IEEE Transactions on Automatic Control*, Submitted. Available online: `arXiv:1003.2005v3`, 2010.
- [57] V. Cichella, E. Xargay, V. Dobrokhodov, I. Kaminer, A. M. Pascoal, and N. Hovakimyan, “Geometric 3D path-following control for a fixed-wing UAV on $SO(3)$,” in: *AIAA Guidance, Navigation and Control Conference*, AIAA-2011-6415, Portland, OR, 2011.
- [58] R. Olfati-Saber, J. A. Fax, and R. M. Murray, “Consensus and cooperation in networked multi-agent systems,” *In Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

- [59] R. Ghabcheloo, I. Kaminer, A. P. Aguiar, and A. M. Pascoal, “A general framework for multiple vehicle time-coordinated path following control,” in: *American Control Conference*, St. Louis, MO, 2009, pp. 3071–3076.
- [60] N. Biggs, *Algebraic Graph Theory*, New York, NY: Cambridge University Press, 1993.
- [61] M. Arcak, “Passivity as a design tool for group coordination,” *IEEE Transactions on Automatic Control*, vol. 52, no. 8, pp. 1380–1390, 2007.
- [62] Z. Lin, B. A. Francis, and M. Maggiore, “State agreement for continuous-time coupled nonlinear systems,” *SIAM Journal on Control and Optimization*, vol. 46, no. 1, pp. 288–307, 2007.
- [63] E. Kharisov, X. Wang, and N. Hovakimyan, “Distributed event-triggered consensus algorithm for uncertain multi-agent systems,” in: *AIAA Guidance, Navigation and Control Conference*, AIAA-2010-8325, Toronto, Canada, 2010.
- [64] A. P. Aguiar, I. Kaminer, R. Ghabcheloo, A. M. Pascoal, E. Xargay, N. Hovakimyan, C. Cao, and V. Dobrokhodov, “Time-coordinated path following of multiple UAVs over time-varying networks using \mathcal{L}_1 adaptation,” in: *AIAA Guidance, Navigation and Control Conference*, AIAA-2008-7131, Honolulu, HI, 2008.
- [65] N. Hovakimyan and C. Cao, *\mathcal{L}_1 Adaptive Control Theory*, Philadelphia, PA: Society for Industrial and Applied Mathematics, 2010.
- [66] N. Hovakimyan, C. Cao, E. Kharisov, E. Xargay, and I. M. Gregory, “ \mathcal{L}_1 adaptive control for safety-critical systems. Guaranteed robustness with arbitrarily fast adaptation,” *IEEE Control Systems Magazine* 2011.
- [67] I. Kaminer, A. Pascoal, E. Xargay, N. Hovakimyan, C. Cao, and V. Dobrokhodov, “Path following for unmanned aerial vehicles using \mathcal{L}_1 adaptive augmentation of commercial

- autopilots,” *Journal of Guidance, Control and Dynamics*, vol. 33, no. 2, pp. 550–564, 2010.
- [68] V. Dobrokhodov, I. Kaminer, I. Kitsios, E. Xargay, N. Hovakimyan, C. Cao, I. M. Gregory, and L. Valavani, “Experimental validation of \mathcal{L}_1 adaptive control: rohrcounterexample in flight,” *Journal of Guidance, Control and Dynamics*, vol. 34, no. 5, pp. 1311–1328, 2011.
- [69] R. Buettner, *USSOCOM-NPS field experimentation cooperative capabilities based experimentation*, <http://faculty.nps.edu/rrbuetttn/TNT.html>, 2010.
- [70] D. Netzer and A. Bordetsky, “TNT testbed for self-organizing tactical networking and collaboration,” in: *International Command and Control Research and Technology Symposium*, Washington, DC, 2009.
- [71] M. T. Flynn, R. Juergens, and T. L. Cantrell, “Employing ISR: SOF best practices,” *Joint Force Quarterly*, vol. 50, pp. 56–61, 2008.
- [72] *PC104-Plus Embedded Single Board Computers: AMD Geode LX800 CPU 500MHz - MSM800SEV/SEL/BEV*, Advanced Digital Logic.
- [73] *Persistent Systems: Wave Relay*, <http://www.persistentsystems.com>, Persistent Systems.
- [74] J. Burl, *Piccolo/Piccolo Plus Autopilots - Highly Integrated Autopilots for Small UAVs*, <http://cloudcaptech.com>, Cloud Cap Technology, Inc.
- [75] M. R. Clement, E. Bourakov, K. D. Jones, and V. Dobrokhodov, “Exploring network-centric information architectures for unmanned systems control and data dissemination,” in: *AIAA Infotech@Aerospace*, AIAA-2009-1999, Seattle, WA, 2009.

- [76] S. Morris and E. W. Frew, *Cooperative tracking of moving targets by teams of autonomous unmanned air vehicles*, tech. rep. FA9550-04-C-0107, MLB Company and University of Colorado, 2004.
- [77] *2d3, Applications*, <http://www.2d3.com/application>, 2d3, 2010.
- [78] V. Dobrokhodov, I. Kaminer, K. D. Jones, and R. Ghabcheloo, “Vision-based tracking and motion estimation for moving targets using small UAVs,” *Journal of Guidance, Control and Dynamics*, vol. 31, no. 4, pp. 907–917, 2008.
- [79] *Urban robotics*, <http://www.urbanrobots.com/home.php>, Urban Robotics, Inc., 2010.
- [80] A. P. Aguiar, J. Almeida, M. Bayat, B. M. Carneira, R. Cunha, A. J. Häusler, P. Maurya, A. Oliveira, A. M. Pascoal, A. Pereira, M. Rufino, L. Sebastião, C. Silvestre, and F. Vanni, “Cooperative autonomous marine vehicle motion control in the scope of the EU GREX Project: Theory and practice,” in: *OCEANS*, Bremen, Germany, 2009, pp. 1–10.
- [81] A. P. Aguiar, J. Almeida, M. Bayat, B. M. Carneira, R. Cunha, A. J. Häusler, P. Maurya, A. Oliveira, A. M. Pascoal, A. Pereira, M. Rufino, L. Sebastião, C. Silvestre, and F. Vanni, “Cooperative control of multiple marine vehicles: Theoretical challenges and practical issues,” in: *IFAC International Conference on Manoeuvring and Control of Marine Craft*, Guarujá, Brazil, 2009.

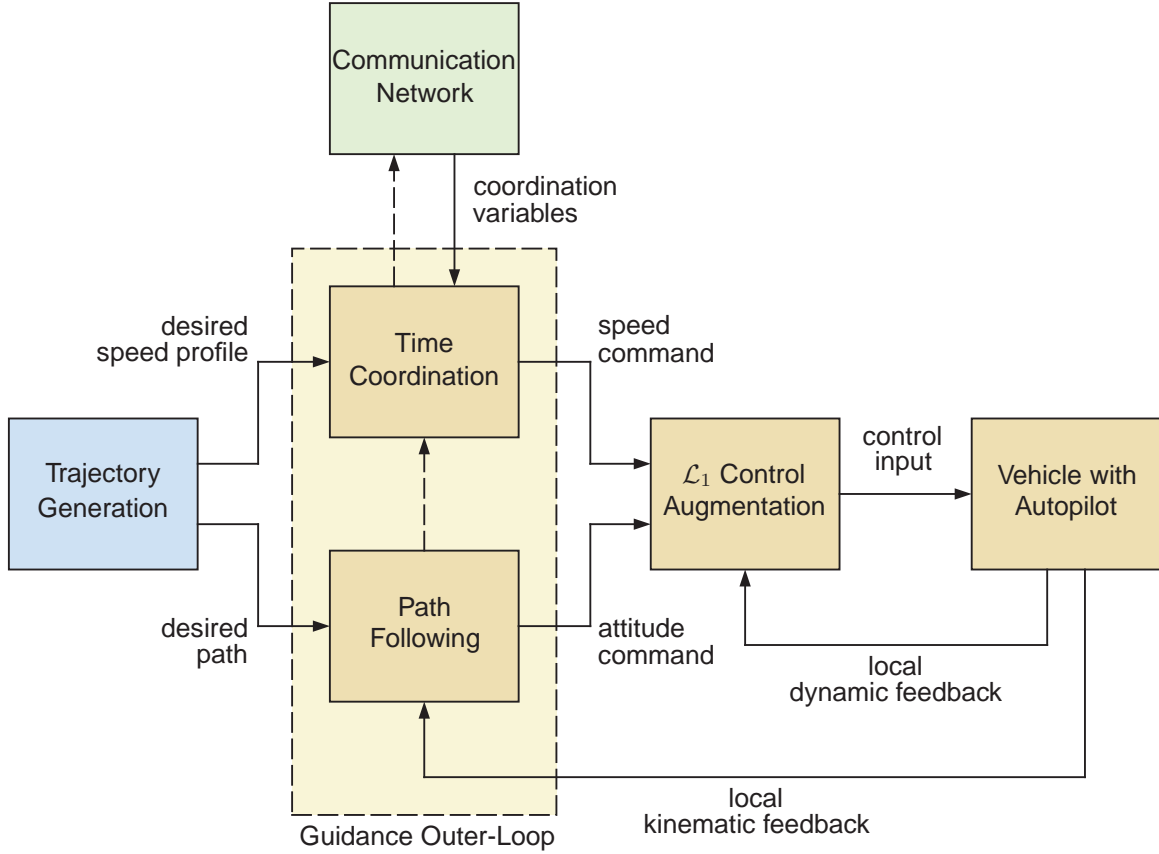
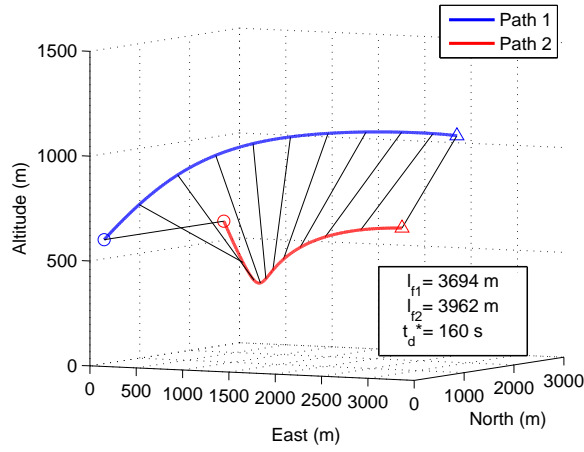
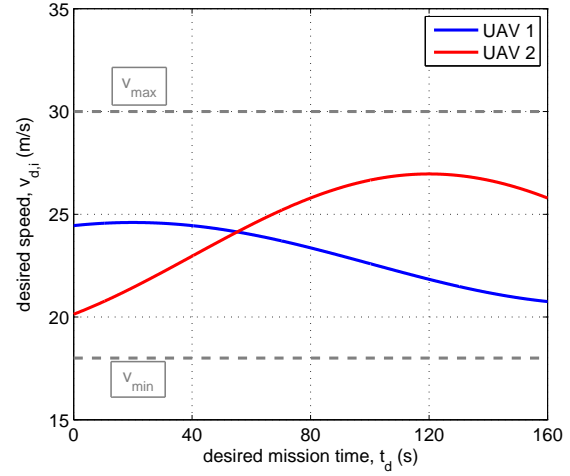


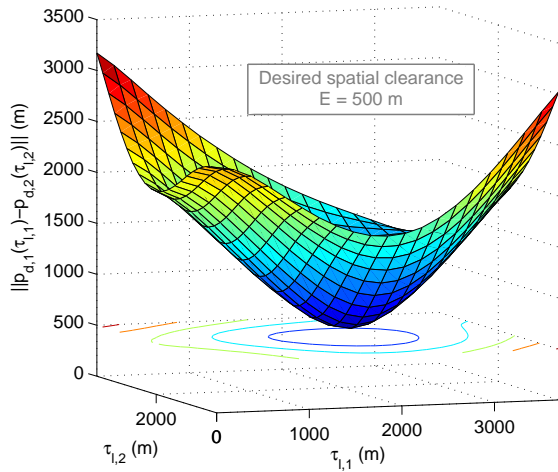
Figure 1. Conceptual architecture of the cooperative control framework adopted. Decoupling space and time in the problem formulation allows for the problems of path following and time-coordination to be solved independently. On one hand, a path-following algorithm ensures that every vehicle follows its own path independently of the temporal assignments of the mission. On the other hand, the speed profile of each vehicle is adjusted about a desired speed profile so as to enforce the temporal constraints that must be met in real time to coordinate the entire fleet of vehicles.



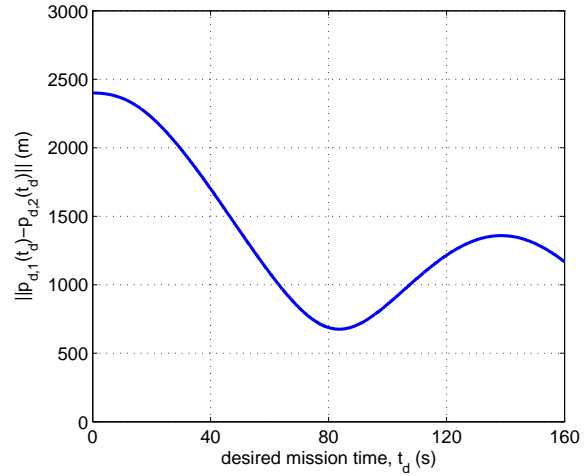
(a) 3D spatial paths. (\circ = initial condition; \triangle = final condition)



(b) Speed profiles.

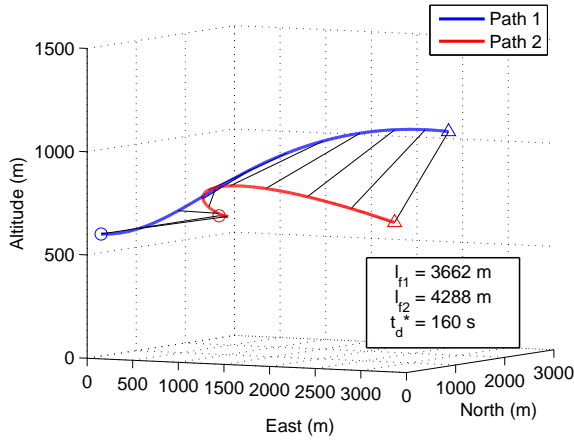


(c) Path separation.

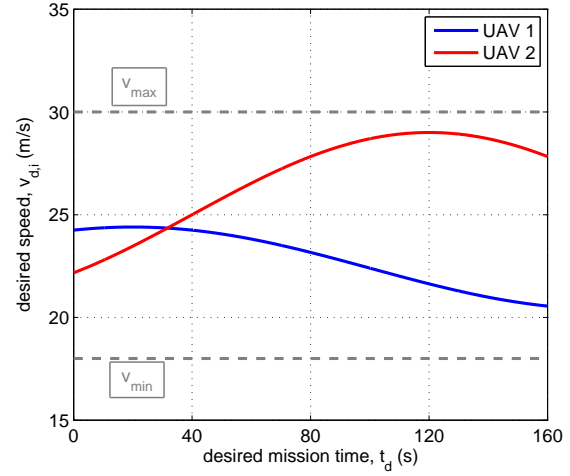


(d) Vehicle separation.

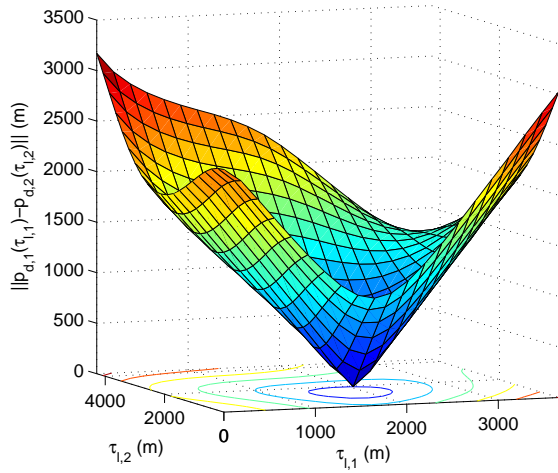
Figure 2. *Collision avoidance in space* ensures that no feasible paths intersect. This approach may be particularly useful in military applications, where jamming prevents vehicles from communicating with each other, and is preferable to the current practice of separating vehicles by altitude.



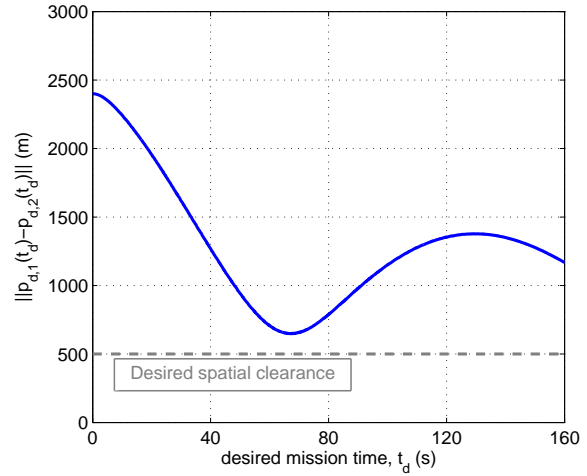
(a) 3D spatial paths. (\circ = initial condition; \triangle = final condition)



(b) Speed profiles.



(c) Path separation.



(d) Vehicle separation.

Figure 3. *Collision avoidance in time* implies that no two vehicles are at the same place at the same time. This approach relies heavily on inter-vehicle communications and is thus a function of the quality of service of the underlying network. The example above shows that, even if the paths intersect at some point –zero path separation–, the desired speed profiles ensure that the two vehicles maintain a prespecified inter-vehicle separation and do not collide.

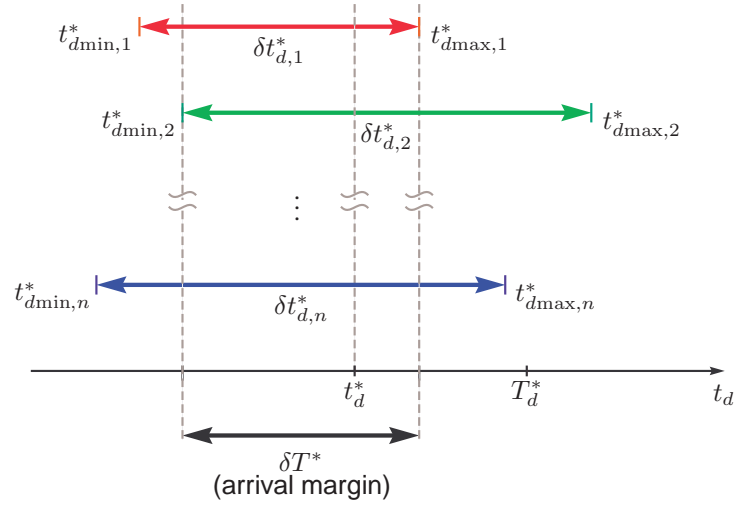


Figure 4. Arrival margin. The simultaneous arrival problem has a solution if and only if the intersection of all individual arrival-time windows is nonempty. A positive arrival margin ensures that this intersection is not empty. Moreover, the magnitude of the arrival margin can be used to characterize the robustness of the trajectory-generation solution at the coordination level.

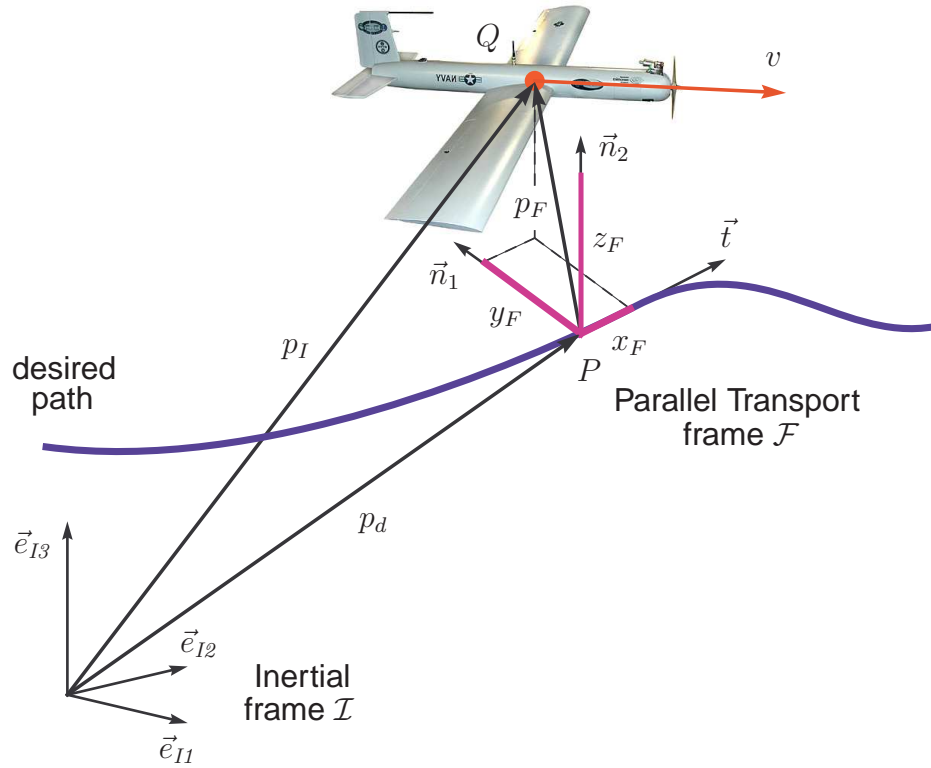


Figure 5. Following a virtual target vehicle. The path-following algorithm presented in this article uses the vehicle's attitude control effectors to follow a virtual target vehicle running along the desired path with a rate of progression that can be selected at will. With this approach, the speed of the vehicle remains as an extra degree of freedom to be used at the coordination level.

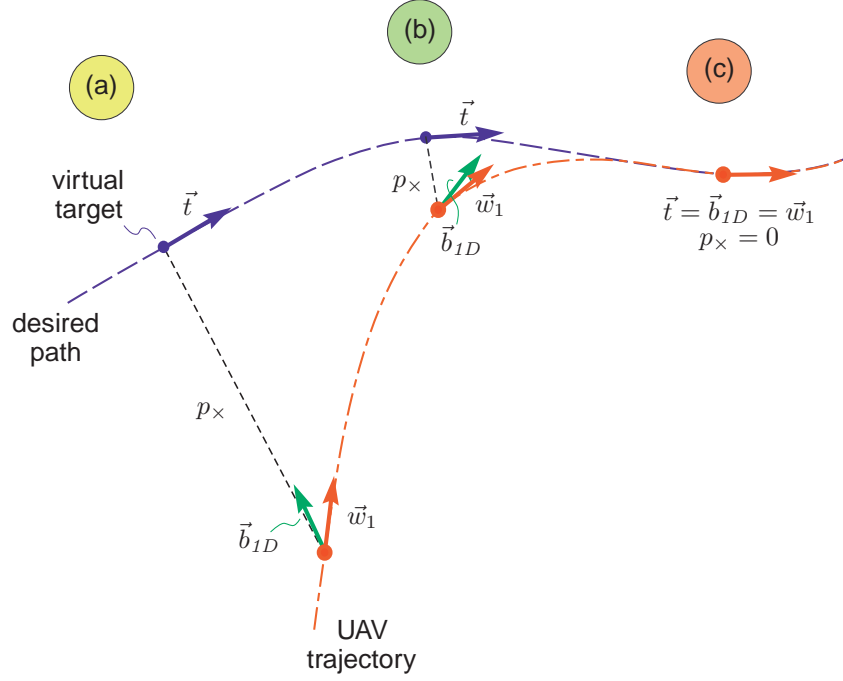


Figure 6. Shaping the approach to the path. An auxiliary frame \mathcal{D} is used to shape the approach attitude to the path as a function of the cross-track error p_\times . (a) When the vehicle is far from the desired path, the versor $\vec{b}_{1D}(t)$ becomes quasi-perpendicular to $\vec{t}(\ell)$. (b) As the vehicle comes closer to the path and the cross-track error becomes smaller, the orientation of $\vec{b}_{1D}(t)$ tends to $\vec{t}(\ell)$. (c) Finally, when the cross-track error becomes zero, $\vec{b}_{1D}(t)$ coincides with $\vec{t}(\ell)$. We notice that, for simplicity, the plot above assumes that the along-path error $x_F(t)$ is zero.

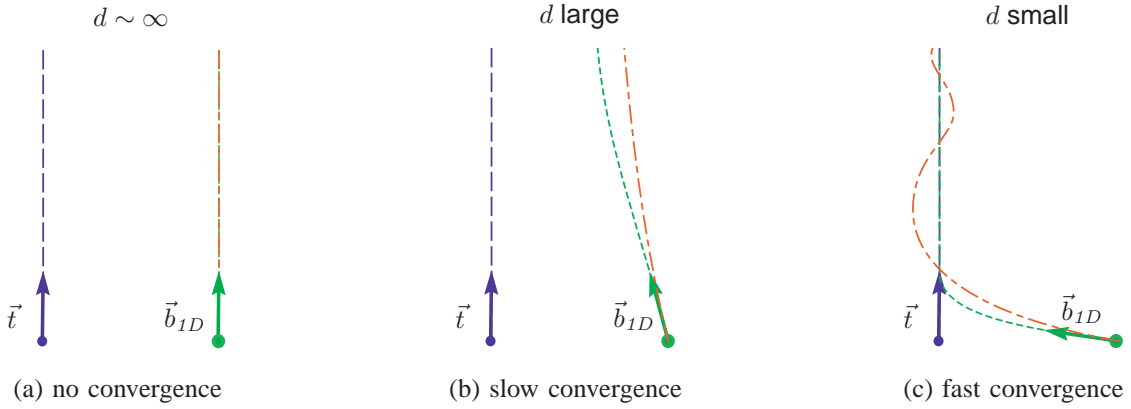


Figure 7. Effect of the characterizing distance d on the convergence of the vehicle to the path. The choice of the characterizing distance d in the definition of the auxiliary frame \mathcal{D} can be used to adjust the rate of convergence for the path-following closed-loop system. (a) When $d \sim \infty$, the vehicle never converges to the path, since $\omega_{D/F} = 0$. (b) For large values of d , the term $\omega_{D/F}$ introduces only small corrections to the “feedforward” term $\omega_{F/I}$, and therefore the rate of convergence of the vehicle to the desired path is slow. On the other hand, (c) small values of d allow for higher rates of convergence, which however might result in oscillatory path-following behavior. In these plots, the blue line is the desired path, the green line represents the desired approach curve, and the red line corresponds to the resulting vehicle trajectory.

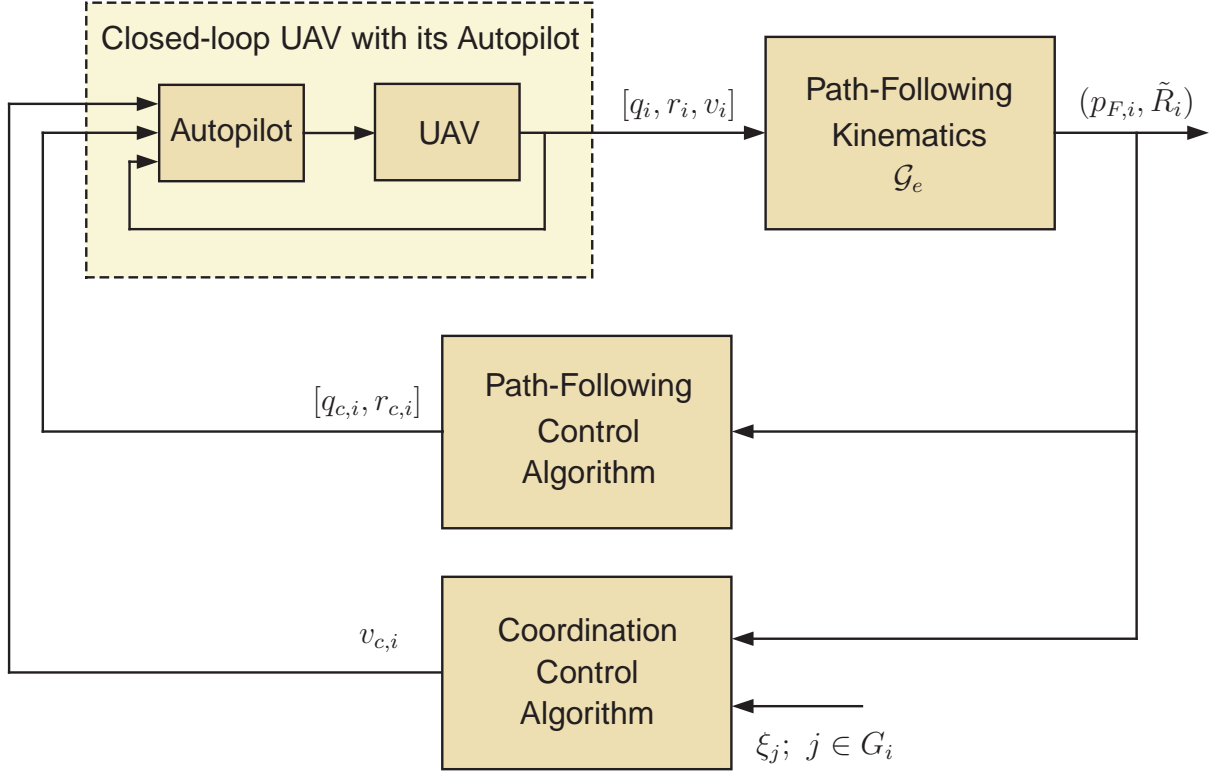


Figure 8. Coordinated path-following closed-loop for the i th vehicle. The cooperative control architecture presented in this article exhibits a multiloop control structure in which an inner-loop controller stabilizes the vehicle dynamics, while guidance outer-loop controllers are designed to control the vehicle kinematics, yielding path-following and time-coordination capabilities.

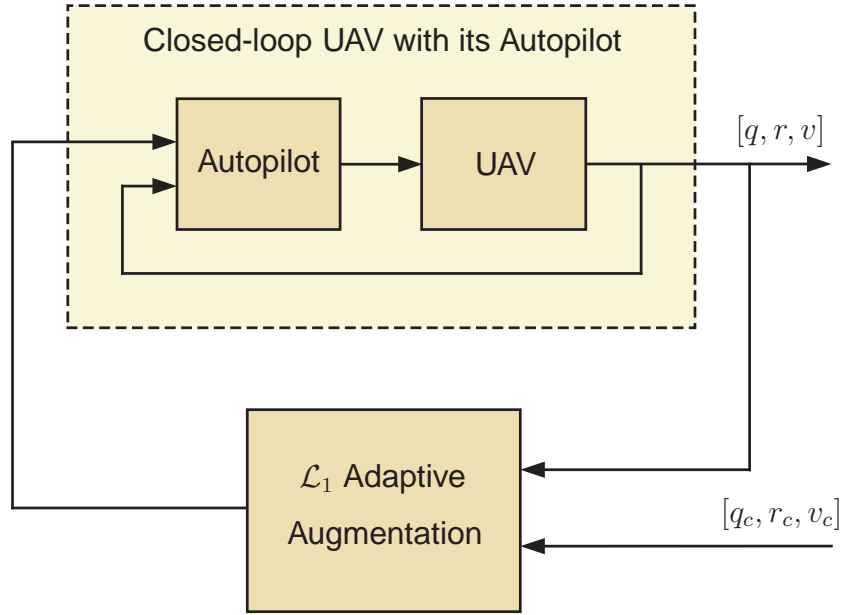


Figure 9. Inner-loop structure with the adaptive augmentation loop. The architecture considered for autopilot augmentation is an output-feedback architecture that uses angular-rate and speed measurements to modify the commands generated by the outer-loop algorithms, which are then sent to the autopilot as reference signals to be tracked. This structure for autopilot augmentation does not require any modifications to the autopilot itself, and at the same time does not use internal states of the autopilot for control design purposes.

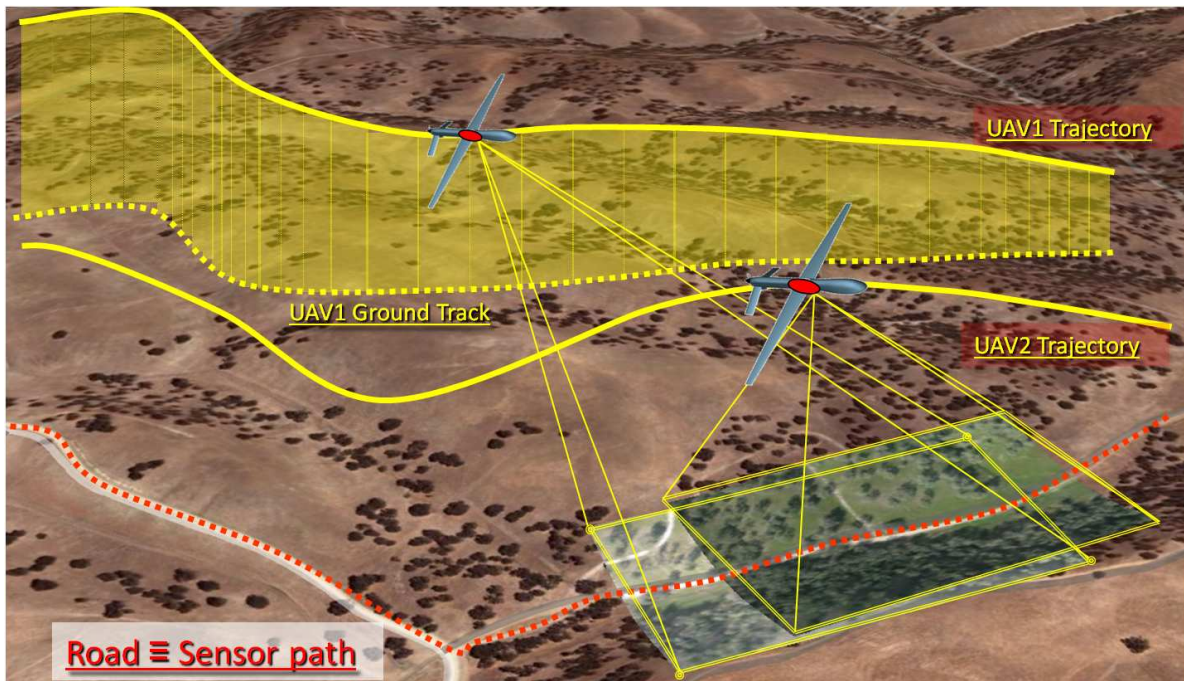
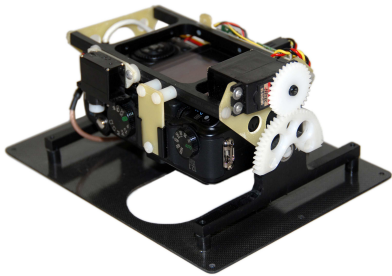


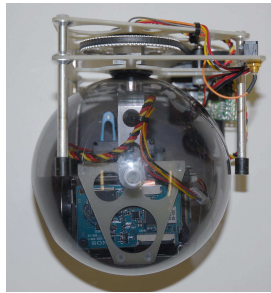
Figure 10. Coordinated road search using multiple unmanned aerial vehicles (UAVs). Two small tactical UAVs equipped with complementary vision sensors detect and follow an improvised explosive device along a road. Cooperative control can ensure a satisfactory overlap of the field-of-view footprints of the sensors along the road, thus increasing the probability of target detection.



(a) SIG Rascal 110 research aircraft.



(b) High-resolution camera.



(c) Full-motion video camera.

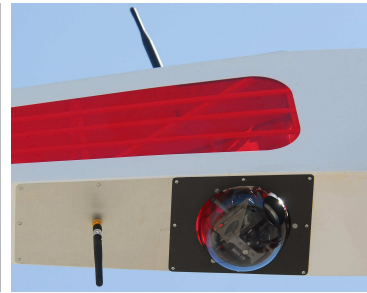


Figure 11. SIG Rascal UAV with two different onboard cameras. The SIG Rascal UAVs (a) used for cooperative path-following missions are equipped with complementary vision sensors. The first UAV has a bank-compensated high-resolution 12-MPx camera (b), while the second UAV has a full-motion video camera suspended on a pan-tilt gimbal enclosure (c).

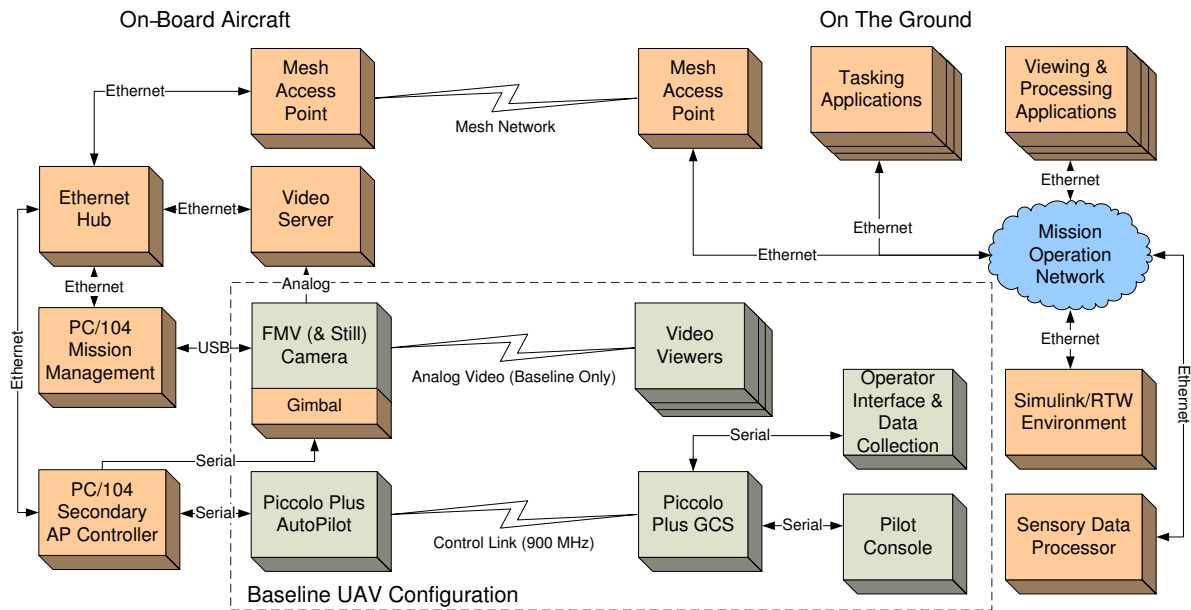
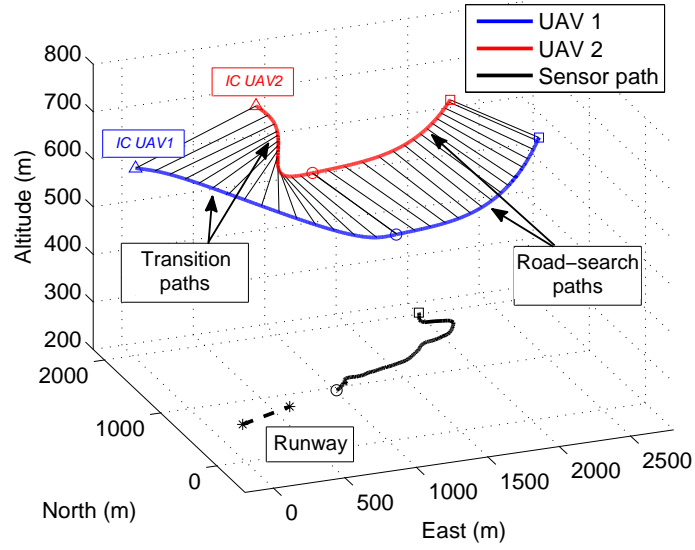
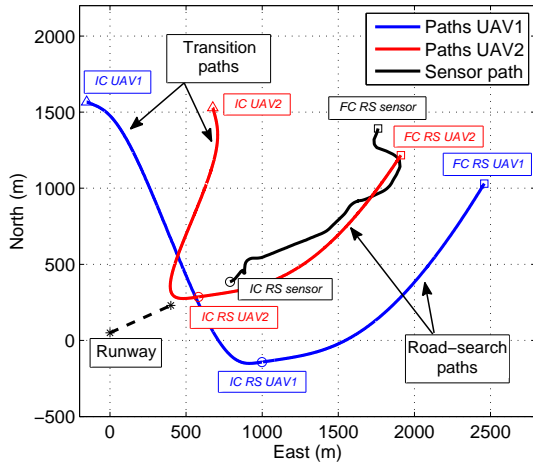


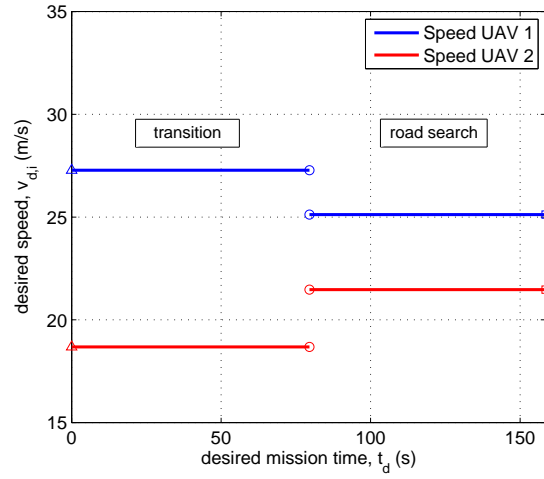
Figure 12. Network-Centric architecture of the airborne platform. The Rascal UAV avionics includes two PC-104 industrial embedded computers assembled in a stack, a wireless Mobile Ad-hoc Network (MANET) link, and the Piccolo Plus autopilot with its dedicated 900-MHz command and control channel. The PC-104 computers are used to run the cooperative control algorithms in hard real time as well as mission management routines enabling onboard preprocessing and retrieval of the sensory data.



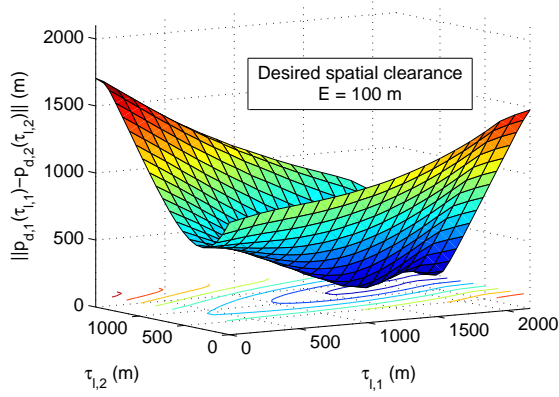
(a) Desired 3D spatial paths.



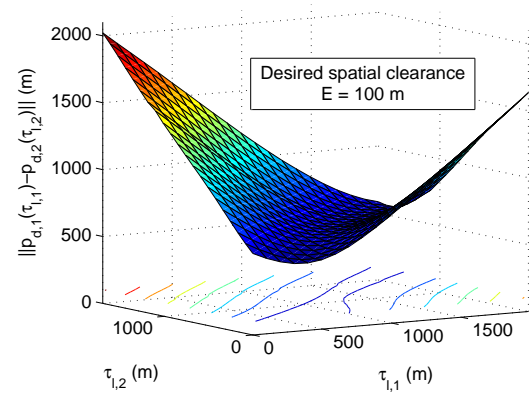
(b) 2D projections.



(c) Desired speed profiles.

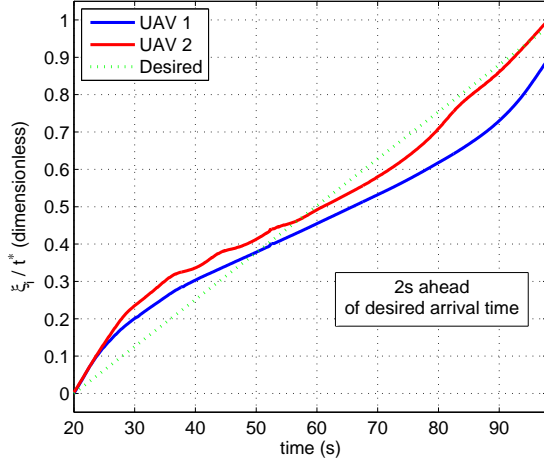


(a) Path separation for transition phase.

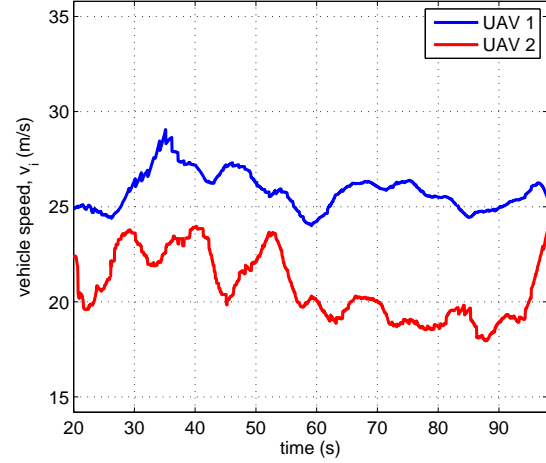


(b) Path separation for road-search phase.

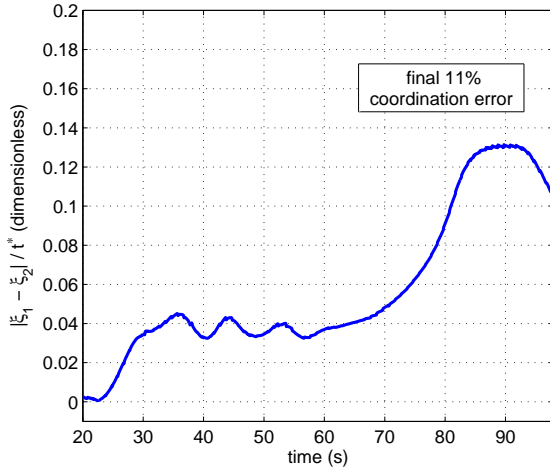
Figure 13. Coordinated road-search trajectory generation. In this mission scenario, two SIG Rascal UAVs cooperate to detect a target moving along a given road. The two UAVs fly from their standby locations (IC UAV1 and IC UAV2) in a coordinated fashion along transition paths so as to arrive at the starting points of the road-search paths (IC RS UAV1 and IC RS UAV2) at the same time. Then, the two UAVs follow the road-search paths while trying to detect a target moving along the sensor path. During the search, the UAVs cooperate and adjust their speeds to ensure the required overlap of the field-of-view footprints of the cameras. The plots show the desired spatial paths for both the transition and road-search phases (a)-(b), the corresponding desired speed profiles (c), and the separation between paths (d)-(e).



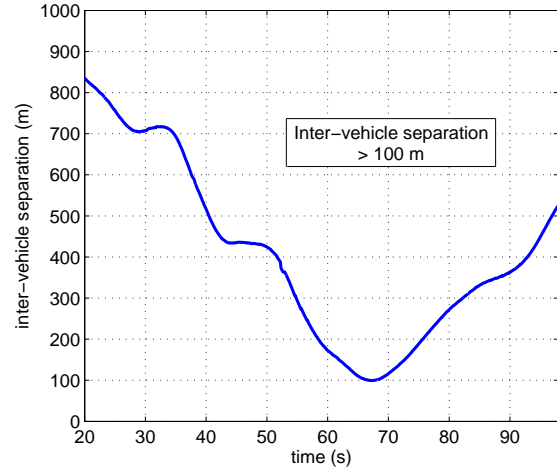
(a) Normalized coordination states.



(b) UAV speeds.

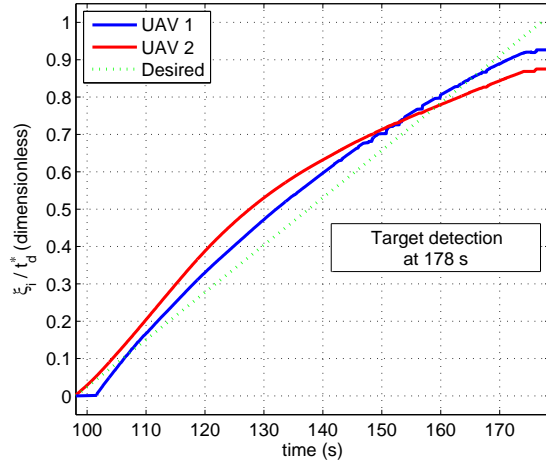


(c) Normalized coordination error.

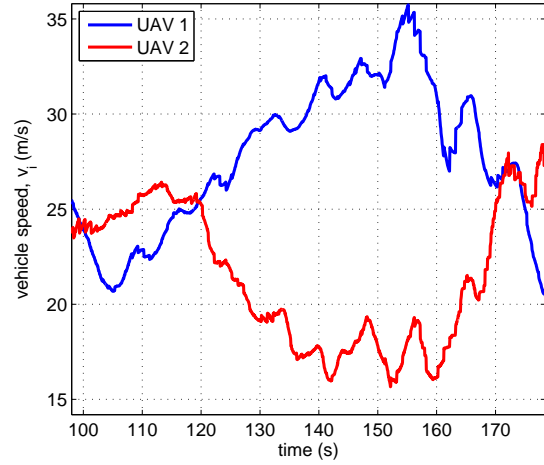


(d) Inter-vehicle separation.

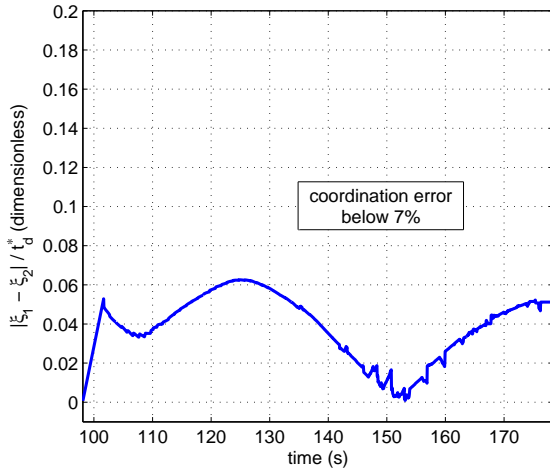
Figure 14. Time-coordination during the transition phase. These plots illustrate the performance of the coordination control algorithm during the transition phase of the mission. Although the transition paths for the two UAVs have significantly different lengths, decentralized coordination control laws adjust the speed profiles of the UAVs based on coordination information exchanged over the supporting communications network. The two UAVs arrive at the starting point of the road-search paths with an 11%-error difference.



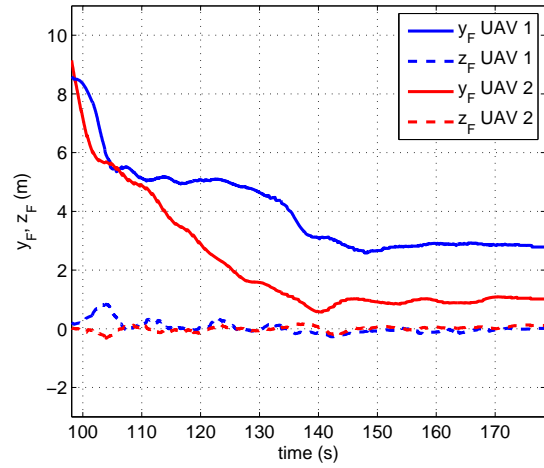
(a) Normalized coordination states.



(b) UAV speeds.

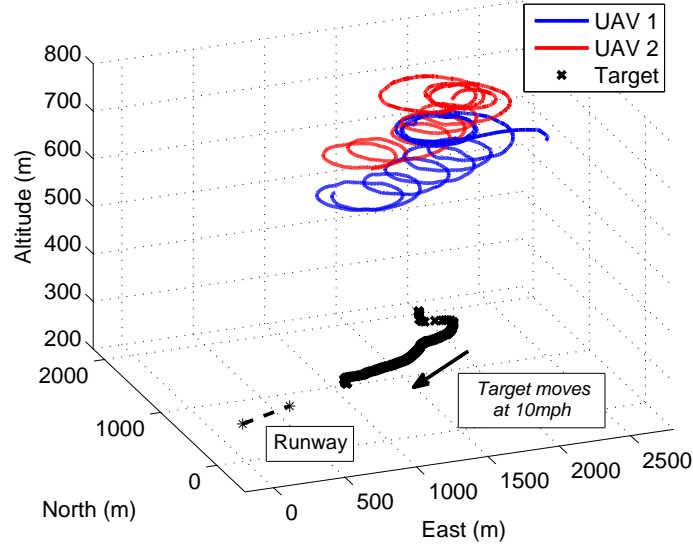


(c) Normalized coordination error.

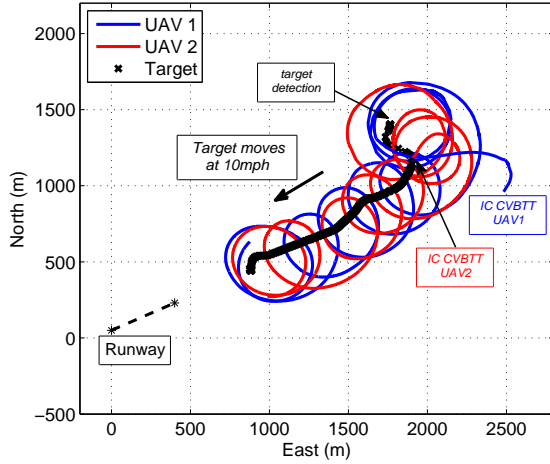


(d) Inter-vehicle separation.

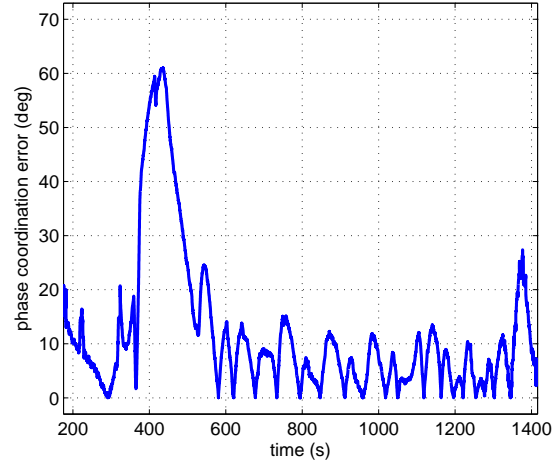
Figure 15. Cooperative path-following control during the road-search phase. These plots illustrate the performance of the cooperative path-following control algorithm during the road search. During this phase of the mission, the coordination errors remain below 7% during the entire duration of the road search, while the path-following cross-track errors converge to a 3 m tube around the desired spatial paths. Cooperation ensures a satisfactory overlap of the footprints of the fields of view of the two cameras. A target is detected on the road at time 178 s. Upon detection, the two UAVs switch to cooperative vision-based tracking mode.



(a) 3D trajectories.



(b) 2D projections.



(c) Phase coordination error.

Figure 16. Cooperative vision-based target tracking (CVBTT). Upon target detection, the two UAVs start tracking the target by means of guidance loops that use visual information for feedback, while simultaneously providing in-situ imagery for precise geo-location of the point of interest. During the target-tracking phase, a coordination algorithm ensures that the two UAVs keep a predefined phase separation of $\frac{\pi}{2}$ rad while “orbiting” around the target. The coordination algorithm uses the coordination control law described in this article to adjust the orbiting speed of the UAVs, with the main difference that the phase on orbit is now used as a coordination state, rather than virtual time.

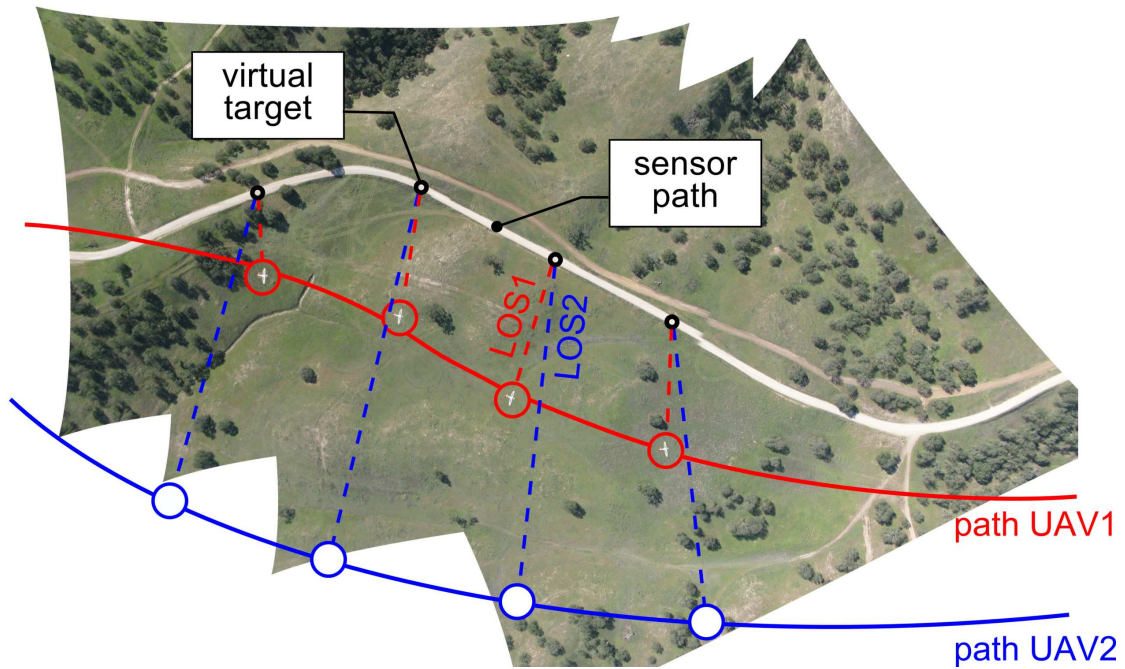
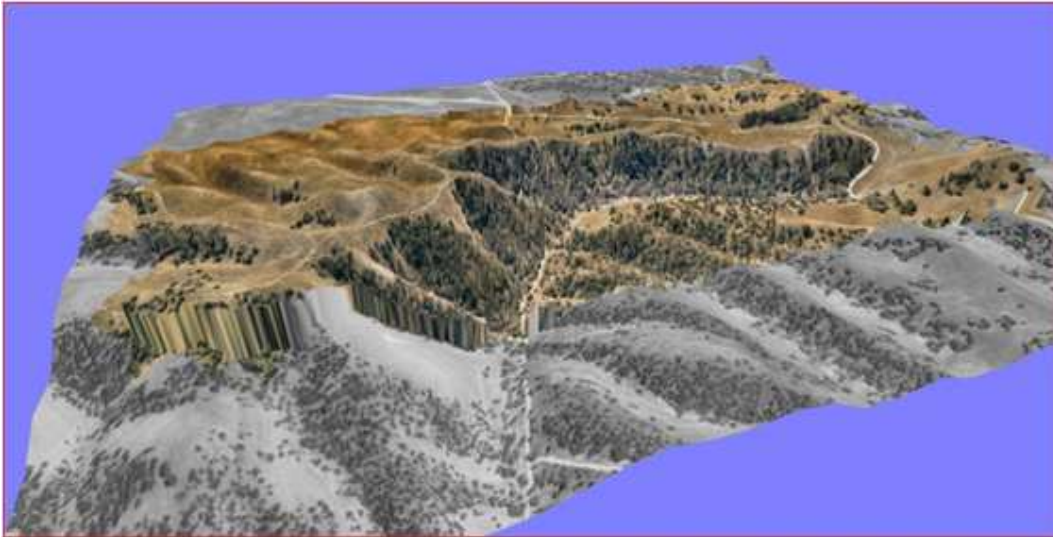


Figure 17. Time-critical cooperation in a road-search mission. In this experiment, the road-search paths are intentionally separated by altitude and optimized such that the UAV flying at lower altitude is continuously present in the field of view of the camera flying at higher altitude. A mosaic of four consecutive high-resolution images illustrate the progression of the lines of sight (LOSs) connecting the two onboard cameras with the virtual target vehicle running along the sensor path.



(a) Automated 3D terrain extraction from 2D high-resolution data. (Courtesy of Urban Robotics)



(b) Near-real-time geo-referenced map obtained from high-resolution data. (Courtesy of 2D3)

Figure 18. High-resolution image exploitation. The use cooperative algorithms in missions involving multiple UAVs can provide accessory mission outcomes, such as (a) 3D geo-referenced models of the operational environment, or (b) geo-referenced maps obtained in near real time from high-resolution imagery.

Sidebar: Path-Following Control

The problem of path following can be briefly described as that of making a vehicle converge to and follow a desired spatial path, while tracking a desired speed profile that may be path-dependent. The temporal and spatial assignments are therefore separated. Often, it is simply required that the speed of the vehicle be kept constant. Path-following control algorithms are pervasive in many robotic applications and are key to the operation of multiple vehicles undergoing cooperative missions.

There is a wealth of literature on path-following algorithms that defies a short summary. Pioneering work in the area can be found in [S1], where an elegant solution to the problem of path-following control is presented for a wheeled robot at the kinematic level. In the setup adopted, the kinematic model of the vehicle is derived with respect to a Frenet-Serret frame moving along the path, while playing the role of a virtual target vehicle to be tracked by the real vehicle. The origin of the Frenet-Serret is placed at the point on the path closest to the real vehicle.

The work in [S1] has spurred a great deal of activity in the literature addressing the path-following problem. A popular approach that has emerged out of this research effort is to solve a trajectory-tracking problem and then reparameterize the resulting feedback controller using an independent variable other than time. See, for example, the work in [S2–S4] and references therein. The approach proposed in [S1] is extended to unmanned aerial vehicles (UAVs) with full account of its dynamics in [S5], where the authors address the issue of path following of trimming trajectories and derive nonlinear path-following controllers that satisfy a linearization property.

Related results can be found in [S6] for autonomous underwater vehicles using a backstepping approach. A common feature of the latter papers is to reduce the path-following problem to that of driving the kinematic errors resolved in Frenet-Serret frame to zero. This approach ensures that path following is essentially done by proper choice of the vehicle’s attitude, a strategy that is akin to that used by pilots when they fly airplanes. The same property does not necessarily hold in the case of the strategies that emerge out of the work in [S2–S4].

The setup used in [S1] is reformulated in [S7], leading to a feedback control law that steers the dynamic model of a wheeled robot along a desired path and overcomes some of the constraints present in [S1]. The key to this algorithm is to explicitly control the rate of progression of the virtual target along the path. This effectively creates an extra degree of freedom that can be exploited to avoid the singularities that occur when the distance to the path is not well defined – this occurs for example when the vehicle is located exactly at the center of curvature of a circular path. Related strategies were exploited in the work of [S8, S9] on output maneuvering and also in the work of [S10]. The path-following algorithm described in this article is an extension of the algorithm presented in [S7] to the case of 3D spatial paths.

Other path-following methods have been presented in the literature that depart from the ideas and concepts of the algorithms described above. In [S11], lateral acceleration commands are used to make a UAV converge to and follow planar curved paths. A nonlinear path-following method that generates acceleration commands to steer a holonomic vehicle towards a given 3D path is presented in [S12]. Path-following algorithms based on the concept of vector fields can be found in [S13, S14]. Finally, the work reported in [S15, S16] presents an elegant approach to path following that uses Lagrange multipliers to derive path-following control laws for mechanical

systems subject to both holonomic and nonholonomic constraints.

References

- [S1] A. Micaelli and C. Samson, *Trajectory tracking for unicycle-type and two-steering-wheels mobile robot*, tech. rep. 2097, Sophia-Antipolis, France: INRIA, 1993.
- [S2] R. Hindman and J. Hauser, “Maneuver modified trajectory tracking,” in: *International Symposium on the Mathematical Theory of Networks and Systems*, St. Louis, MO, 1996.
- [S3] S. A. Al-Hiddabi and N. H. McClamroch, “Tracking and maneuver regulation control for nonlinear non-minimum phase systems,” *IEEE Transactions on Control System Thechnology*, vol. 10, no. 6, pp. 780–792, 2002.
- [S4] A. P. Aguiar and J. Hespanha, “Position tracking of underactuated vehicles,” in: *American Control Conference*, Denver, CO, 2003, pp. 1988–1993.
- [S5] I. Kaminer, A. M. Pascoal, E. Hallberg, and C. Silvestreo, “Trajectory tracking for autonomous vehicles: An integrated approach to guidance and control,” *Journal of Guidance, Control and Dynamics*, vol. 21, no. 1, pp. 29–38, 1998.
- [S6] P. Encarnação, “Nonlinear path following control systems for ocean vehicles,” PhD thesis, Lisbon, Portugal: Instituto Superior Técnico, 2002.
- [S7] 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, 2003, pp. 1387–1392.
- [S8] 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, 2002.

- [S9] R. Skjetne, T. I. Fossen, and P. V. Kokotović, “Robust output maneuvering for a class of nonlinear systems,” *Automatica*, vol. 40, no. 3, pp. 373–383, 2004.
- [S10] F. Díaz del Río, G. Jiménez, J. L. Sevillano, C. Amaya, and A. Civit Balcells, “A new method for tracking memorized paths: Application to unicycle robots,” in: *Mediterranean Conference on Control and Automation*, Lisbon, Portugal, 2009.
- [S11] S. Park, J. Deyst, and J. P. How, “Performance and Lyapunov stability of a nonlinear path-following guidance method,” *Journal of Guidance, Control and Dynamics*, vol. 30, no. 6, pp. 1718–1728, 2007.
- [S12] D. J. Gates, “Nonlinear path following method,” *Journal of Guidance, Control and Dynamics*, vol. 33, no. 2, pp. 321–332, 2010.
- [S13] D. R. Nelson, D. B. Barber, T. W. McLain, and R. W. Beard, “Vector field path following for miniature air vehicles,” *Transactions on Robotics*, vol. 23, no. 3, pp. 519–529, 2007.
- [S14] D. A. Lawrence, E. W. Frew, and W. J. Pisano, “Lyapunov vector fields for autonomous unmanned aircraft flight control,” *Journal of Guidance, Control and Dynamics*, vol. 31, no. 5, pp. 1220–1229, 2008.
- [S15] E. Peyami and T. I. Fossen, “A Lagrangian framework to incorporate positional and velocity constraints to achieve path-following control,” in: *IEEE Conference on Decision and Control*, Orlando, FL, 2011.
- [S16] E. Peyami and T. I. Fossen, “Motion control of marine craft using virtual positional and velocity constraints,” in: *International Conference on Control and Automation*, Santiago, Chile, 2011.

Author Information

Enric Xargay (xargay@illinois.edu) is a Ph.D. candidate in the Department of Aerospace Engineering at the University of Illinois at Urbana-Champaign. He received 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. His research interests include aircraft flight control, nonlinear systems, adaptive control, robust control, and cooperative control of autonomous systems.

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 (since 2004) as an Assistant Professor of the Mechanical and Aerospace engineering department at the NPS. He has authored/co-authored more than 60 refereed publications. His research interests include flight mechanics and trajectory optimization; guidance, navigation and control with an emphasis on unmanned systems; cooperative control and combat maneuvering of multi-vehicle formations; real-time embedded control systems design.

Isaac Kaminer received 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.

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 Vice-Director, Scientific Affairs of the Institute for Systems and Robotics (ISR). He has coordinated and participated in a large number of international projects 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 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.

Naira Hovakimyan received her M.S. in theoretical mechanics and applied mathematics in 1988 from Yerevan State University in Armenia. She received 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 the Department of Mechanical Science and Engineering. She is the 2011 recipient of the AIAA Mechanics and Control of Flight Award. She has coauthored one book and more than 250 refereed publications. Her research interests are in the theory of robust adaptive control and estimation with an emphasis on aerospace applications, control in the presence of limited information, networks of autonomous systems and game theory. She is an associate fellow and life member of AIAA, a Senior Member of IEEE, and a member of AMS and ISDG.

Chengyu Cao has been an assistant professor of mechanical engineering at the University of Connecticut since 2008. Prior to that, he was a research scientist in the Department of Aerospace and Ocean Engineering at the Virginia Tech. He received his Ph.D. in mechanical engineering from the Massachusetts Institute of Technology in 2004. He earned his B.S. degree in electronics and information engineering from Xi'an Jiaotong University, China, and M.S. in manufacturing engineering from Boston University in 1995 and 1999, respectively. His research interests are in the areas of dynamics and control, adaptive and intelligent systems, and mechatronics, with focus on unmanned systems and aerospace applications. He is the author of more than 100

publications.