sensors

MDPI

Article

Decentralized Mesh-Based Model Predictive Control for Swarms of UAVs

Salvatore Rosario Bassolillo [1], Egidio D'Amato [2,*], Immacolata Notaro [1], Luciano Blasi [1] and Massimiliano Mattei [1]

[1] Department of Engineering, University of Campania Luigi Vanvitelli, 81031 Aversa (CE), Italy; salvatorerosario.bassolillo@unicampania.it (S.R.B.); immacolata.notaro@unicampania.it (I.N.); luciano.blasi@unicampania.it (L.B.); massimiliano.mattei@unicampania.it (M.M.)

[2] Department of Science and Technology, University of Naples Parthenope, 80143 Naples, Italy

* Correspondence: egidio.damato@uniparthenope.it

Received: 29 June 2020; Accepted: 28 July 2020; Published: 3 August 2020

Abstract: This paper deals with the design of a decentralized guidance and control strategy for a swarm of unmanned aerial vehicles (UAVs), with the objective of maintaining a given connection topology with assigned mutual distances while flying to a target area. In the absence of obstacles, the assigned topology, based on an extended Delaunay triangulation concept, implements regular and connected formation shapes. In the presence of obstacles, this technique is combined with a model predictive control (MPC) that allows forming independent sub-swarms optimizing the formation spreading to avoid obstacles and collisions between neighboring vehicles. A custom numerical simulator was developed in a Matlab/Simulink environment to prove the effectiveness of the proposed guidance and control scheme in several 2D operational scenarios with obstacles of different sizes and increasing number of aircraft.

Keywords: UAV swarm; formation flight; decentralized control; anti-collision; obstacle avoidance

1. Introduction

Research on UAV has attracted increasing attention for at least two decades, exploring several topics, such as design, development and deployment. Military and civil applications are exponentially growing thanks to the great versatility and adaptability of these kind of platforms to a great number of missions. Furthermore, costs are decreasing more and more due to the technological developments driven by a greater attention and use.

However, it is not possible for a mission to always be completed effectively by a single UAV, as it may require the cooperation of several UAVs to perform complex tasks. This need may also derive from the choice to use smaller and lighter vehicles to reduce costs and increase reliability. In fact, costs of a single aircraft exponentially increase with the take-off weight [1–4], whereas the use of several smaller UAVs can also be effective in reducing the production and even operating costs [5,6]. The reduction of the mission time can be another important reason for the use of more than one aircraft whenever the limited endurance of a single UAV forces it to take off and land several times for refueling. Furthermore, the robustness of the overall mission can be enhanced by creating redundancy and reducing the payload weight per aircraft [7–9]. Finally, by involving several smaller vehicles having an improved mobility in confined spaces, it is possible to survey larger areas with respect to a bigger single aircraft, even if equipped with higher-quality sensors.

In the literature, several examples can be found regarding swarms of UAVs and related applications. An extended research field focuses on the swarming behavior, called *flocking*, known in nature as the collective behavior of a large number of agents able to interact and share a common objective [10–19]. Flocking techniques deal with the ability of a group of autonomous agents (being aircraft or robots) to move like a swarm of birds or insects.

Tracking a static or moving target has been studied in [20], where multiple UAVs searched for a source of pollution while keeping a safe distance from each other. An example of group of aircraft looking for targets using random walk paths can be found in [21], where an ant colony-like heuristic algorithm was considered. By depositing a virtual pheromone, the probability that other aircraft pass the same path was increased. Cooperative surveillance is another investigated task involving a swarm of UAVs: in [22], an evolutionary-based optimization technique was used to find the optimal position of UAVs in the airspace that maximizes the coverage of the area of interest, whereas in [23] a distributed algorithm was considered, taking into account communication constraints. In [24], a cognitive adaptive optimization algorithm was presented, having the goal of maximizing the surveilled area with a team of aircraft. In [25], a behavior-based algorithm leading a swarm of UAVs during a surveillance mission was presented.

However, the key point for an effective use of a swarm is the capability of the aircraft to properly collaborate in order to accomplish the mission objectives by coordinating their tasks.

An example of a collaboration concept consists of constrained flight between multiple UAVs, as in the case of several drones that work linked together to transport heavy loads [26–30].

Another example of a collaboration concept is the use of fleets of fixed-wing aircraft to improve the aerodynamic efficiency during flight [31]. This concept falls into the more general problem of managing the aircrafts' relative positions to optimize performance in terms of fuel consumption or time spent to achieve mission goals [32–37].

Generally speaking, the output of a swarm control algorithm consists in the computation of the appropriate control signals to maintain the shape of the swarm, satisfying the assigned performance constraints [38]. For example, flocking research focuses on consensus-based formation control [15], which uses the relative position of the vehicles to realize a certain shape of the flight formation. Formation control algorithms focus mainly on stability and performance constraints. The classic flocking control model is based on three heuristic rules [39]: cohesion, separation and alignment. The concept of using flexible swarm shapes has been developed only in recent years.

Another important research field is cooperative motion planning, which is based on collaboration among vehicles with different point of views. Taking into account mission goals (usually in terms of waypoints) and environmental constraints, such as obstacles or no-fly zones, aircraft are driven along optimal trajectories, while benefiting from coordination [40]. Cooperative motion planning algorithms mainly focus on safety distance from obstacles, computational time, swarm shape and trajectory smoothness [41,42].

To maintain the given shape of the flight formation, several control architectures have been proposed: leader–follower, virtual structure and behavior-based.

The leader–follower approach considers a leader working as a reference aircraft in the swarm with full information about the other vehicles. Some papers introduce the idea of a virtual leader in order to overcome the issues of centralization [43–45].

A virtual structure architecture was proposed in [46]. According to this approach, the vehicles keep a rigid geometric relationship to each other in order to maintain a reference shape [47]. This concept considers the formation shape as a rigid body, maintained by minimizing the position error between the virtual structure and the actual aircraft position.

Behavior-based formation control was first proposed in [48]. It deals with the formation control problem by using a hybrid vector-weighted control function, which is able to compute the command action based on several kinds of formation missions. Applications of this approach to large-scale robot formations can be found in [49–52].

Within the flocking research field, one of the most popular approaches to control flocking behavior is the use of artificial potential fields and consensus [38,53,54]. Flock aggregation and collision avoidance are often dealt with using a potential field, whose design is still an active field of research [55].

Another common approach is based on the use of the nonlinear dynamic inversion technique (NDI). Through the use of nonlinear functions, the NDI control technique attempts to cancel out the inherent dynamics of a plant and enforce the dynamics of a reference model. The most important benefit is the ability to linearize systems. The resulting model can be controlled by any technique, such as sliding model control (SMC) [56,57] and pole placement [58,59], $\mathcal{H}_\infty$ [60]. The authors of [61] dealt with a decentralized algorithm for a swarm of UAVs that uses the potential field approach combined with a sliding mode control technique. Optimal control is another option that can be used to compute the control command to achieve swarm objectives while preserving performance and environmental constraints. A stochastic optimal control approach was presented in [62], whereas linear and nonlinear MPCs. were presented in [63–65].

In several papers, authors used a machine learning heuristics to tackle the problem of controlling a swarm. For example, [66] described a task assignment algorithm using a self-reinforcement learning model to specialize the vehicles for the different task types, balancing the work load; another example is the use of the ant colony algorithm to organize task assignment: in [36] a swarming technique was developed to perform distributed target recognition using UAVs. In [67], multiple UAVs were involved in looking for targets that are displaying evading behavior in order to avoid detection. In [68], a genetic algorithm was used to optimize the tuning of the consensus control parameters of a swarm of quadrotors. In this work, UAVs try to find an efficient geometric flight configuration in order to optimize their sensing capabilities. Nature-inspired optimization algorithms were also used in reconfiguration problems [69,70].

Scalability and decentralization represent two other important issues in swarm management strategy [61]. They are strictly related to the swarm architecture and control approach used to maintain the formation. While several leader–follower architectures are centralized on the leader, computing the desired positions for each aircraft, flocking techniques are usually decentralized, in order to mimic the natural behavior.

In this paper, the design of a decentralized flocking strategy is dealt with for a fleet of UAVs flying at a given altitude to reach a target position while maintaining an assigned connection topology with desired mutual distances.

The starting point of the proposed approach is the triangle formation algorithm (TFA), previously described in [71,72]. Such a technique represents an interactive control algorithm used to organize the swarm into an equilateral triangle-based formation. However, TFA does not assign a precise topology to the formation and may have some issues in the presence of obstacles, allowing the formation of sub-swarms that may not rendezvous with the main formation.

This is an important open question concerning the spreading of the formation in the presence of obstacles. Several approaches use a virtual leader, allowing the followers to spread between obstacles without creating sub-swarms with a prescribed topology [73–75].

In this paper, we propose a new strategy, based on an extension of the Delaunay triangulation algorithm: in the absence of obstacles, the assigned topology guarantees a single regular-meshed flight formation, whereas in the presence of obstacles, it allows the formation to split into independent sub-formations to avoid collisions, forcing them to rendezvous after obstacles.

The application of the Delaunay triangulation in the presence of obstacles represents a novel aspect requiring the definition of a visibility property. Furthermore, while the idea of triangular mesh-based formation has already been exploited [72], the new feature consists in the capability to generate sub-swarms without the definition of any physical or virtual leader.

Another original point concerns trajectory tracking and collision avoidance: TFA computes the desired position of the aircraft to accomplish the triangle-based formation. This is tracked by a proportional navigation-based control algorithm that cannot take into account performance and environmental constraints. In [72], an artificial potential field was also used to take obstacles into account. Our approach is to use a potential field [76,77] to compute the aircraft's desired heading angle and speed to keep the desired mutual distances between aircraft, whereas an MPC-based control algorithm performs the trajectory tracking and the obstacle collision avoidance.

In our previous paper [78], we presented an MPC strategy in the absence of obstacles with prescribed rules between aircraft. Based on this work, we propose an original reactive MPC aimed at improving the performance of the anti-collision scheme using practical geometrical constraints leading to a computationally tractable algorithm.

The paper is organized as follows:

- In Section 2, the problem is precisely stated and the general scheme of the proposed strategy is illustrated;
- Section 3 focuses on the swarm topology based on the Delaunay triangulation;
- Section 4 describes the potential based guidance algorithm driving the swarm on a regular-meshed flight formation;
- Section 5 focuses on the MPC-based trajectory tracking algorithm with collision avoidance;
- Finally, in Section 6, numerical simulation results are shown for several operational scenarios.

2. Problem Statement and General Architecture

Consider a swarm of N vehicles flying at fixed speed and altitude to simulate a cruise or surveillance mission phase.

As the focus of this paper is mainly on swarm guidance and collision avoidance, we neglected any atmospheric disturbance as well as any change in aircraft properties (i.e., mass variation due to fuel consumption).

We then assume that each aircraft is equipped with its own flight control system (FCS), so the i-th vehicle's closed loop dynamics can be described well by the following simple kinematic equations:

$$\begin{cases} \dot{x}_i(t) & = & V_i(t)\cos\psi_i(t) \\ \dot{y}_i(t) & = & V_i(t)\sin\psi_i(t) \\ \dot{V}_i(t) & = & u_{V_i}(t) \\ \dot{\psi}_i(t) & = & u_{\psi_i}(t) \end{cases} \tag{1}$$

where $\mathbf{P}_i(t) = [x_i(t), y_i(t)]^T$ is the position at time t, $V_i(t)$ is the speed, $\psi_i(t)$ is the heading angle $u_{\psi_i}(t)$ and $u_{V_i}(t)$ are the input control signals in terms of turning rate and acceleration along the flight trajectory, respectively.

The goal of the fleet is to reach a given target position, namely $\mathbf{P}^f = [x^f, y^f]^T$, preserving the mutual distance between pairs of aircraft.

Figure 1, shows the architecture of the guidance algorithm implemented on each aircraft. This is based on the following components:

- The so-called swarm awareness algorithm (SAA) aims to compute the connection topology $(\mathcal{P}, \overline{\mathcal{Q}}_i)$ using an extended Delaunay triangulation technique that takes into account the presence of obstacles.
- The swarm guidance algorithm (SGA) is based on a potential field technique to compute the desired heading angle and speed $(\tilde{\psi}_i, \tilde{V}_i)$ so as to maintain the mutual distances between aircraft and reach the target position $\mathbf{P}^f$.
- The trajectory tracking and collision avoidance algorithm (TTCAA) is based on a model predictive controller and provides the reference signals u_{ψ_i} and u_{V_i} to the FCS, taking into account environmental constraints (obstacles, no-fly zones, other aircraft) and possible constraint on aircraft state and input (e.g., minimum/maximum speed, maximum turn rate).

In order to take into account a digital implementation, the algorithm works with a fixed sampling period Δt. In the following the discrete time $t_k = t_{k-1} + \Delta t$ is referred to as k.

The proposed strategy is based on the assumption that each vehicle can measure or estimate its current position, speed and heading. Moreover, it is equipped with an ADS-B or a similar device to share these estimates with the other vehicles.

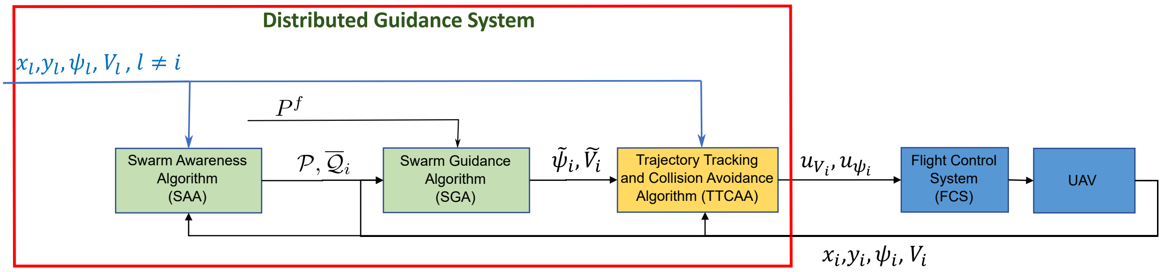

Distributed Guidance System
$x_l, y_l, \psi_l, V_l, l \neq i$
P^f
Swarm Awareness Algorithm (SAA)
$\mathcal{P}, \overline{\mathcal{Q}}_i$
Swarm Guidance Algorithm (SGA)
$\tilde{\psi}_i, \tilde{V}_i$
Trajectory Tracking and Collision Avoidance Algorithm (TTCAA)
u_{V_i}, u_{ψ_i}
Flight Control System (FCS)
UAV
x_i, y_i, ψ_i, V_i

Figure 1. On-board guidance and control algorithm architecture (time information has been omitted).

3. Swarm Awareness Algorithm

The SAA is based on the Delaunay triangulation [79] with the addition of a visibility concept.

Definition 1. *Given the set $\mathcal{P}(k) = \{\mathbf{P}_l(k), l = 1, 2, \ldots, N\}$ of positions of the swarm vehicles, the Delaunay triangulation, $\mathcal{D}(\mathcal{P}(k)) = \{\mathfrak{T}_{jml} : \forall\ \mathbf{P}_j(k), \mathbf{P}_m(k), \mathbf{P}_l(k) \in \mathcal{P}(k)\}$, is a triangular partition of the swarm such that no position $\mathbf{P}_i(k) \in \mathcal{P}(k)$ is inside the circumcircle of any triangle $\mathfrak{T}_{jml}(k) \in \mathcal{D}(\mathcal{P})$, $(j, m, l \neq i)$.*

Definition 2. *Given a Delaunay triangulation, the associated Delaunay triangulation graph (DTG) $\mathcal{G}(k) = \{\mathcal{V}(k), \mathcal{A}(k)\}$ is composed of the nodes set $\mathcal{V}(k) = \mathcal{P}(k)$ and the arc set $\mathcal{A}(k)$ containing all the pairs $< \mathbf{P}_m(k)\mathbf{P}_l(k) >$ such that the segment $\overline{\mathbf{P}_m, \mathbf{P}_l}$ is an edge of a Delaunay triangle.*

Definition 3. *A vehicle m is a neighbor of the vehicle l, at time k, if the pair $< \mathbf{P}_m(k), \mathbf{P}_l(k) >\in \mathcal{A}(k)$. The set of neighboring aircraft of the j-th vehicle is:*

$$\mathcal{Q}_j(k) = \{m : \quad < \mathbf{P}_m(k), \mathbf{P}_j(k) >\in \mathcal{A}(k)\} \tag{2}$$

In the presence of obstacles, which are modeled as flat geometrical figures, in order to implement an obstacle-avoidance strategy based on the possible splitting of the swarm into independent sub-swarms, a visibility concept is also introduced.

Definition 4. *At time k, aircraft j is visible from aircraft i if segment $\overline{\mathbf{P}_i(k)\mathbf{P}_j(k)}$ does not intersect any obstacle. The set of visible neighbors of the aircraft i is*

$$\overline{\mathcal{Q}}_i(k) = \{j \in \mathcal{Q}_i : \quad \Phi_{ij}(k) = 1\} \tag{3}$$

where $\Phi_{ij}(k)$ is the visibility operator

$$\Phi_{ij}(k) = \Phi_{ji}(k) = \begin{cases} 1 \textit{ if i is visible from j} \\ \quad 0 \textit{ otherwise.} \end{cases} \tag{4}$$

According to the visibility concept, in the presence of obstacles, the DTG is modified by considering only arcs present in $\bigcup_{i=1}^{N} \overline{\mathcal{Q}}_i(k)$. For this reason, DTG may become not connected and several subgraphs without common nodes may arise. However, once the obstacles have been overcome, the DTG becomes connected again, resulting in a rendezvous of the formation.

On each vehicle i, the SAA calculates the DTG and $\overline{\mathcal{Q}}_i$ at each time step k.

4. The Swarm Guidance Algorithm

The SGA computes the guidance law to guarantee a correct position of the aircraft in the swarm while flying toward the target point. It is based on a local interaction between visible neighboring aircraft and a potential field approach [80,81] forcing the swarm vehicles to take place on the nodes of a regular triangular mesh with a given mutual distance $\bar{d}$.

At each discrete time step, the SGA computes a desired heading $\tilde{\psi}_i(k)$ and speed $\tilde{V}_i(k)$, which are then passed to the trajectory tracking algorithm described in the next section. In other terms, the SGA assures that the vehicle i moves toward a target direction $\mathbf{\Gamma}_i(k) = [\cos\tilde{\psi}_i(k), \sin\tilde{\psi}_i(k)]^T$ with a given speed.

As shown in Figure 2, let $d_{i,j}(k) = ||\mathbf{P}_j(k) - \mathbf{P}_i(k)||$ be the distance between two aircraft, i and j, visible to each other. An isotropic artificial potential field forces aircraft i to stay on a circumference with radius $\bar{d}$ centered on $\mathbf{P}_j(k)$, for all visible aircraft j. To achieve this goal, the desired direction $\boldsymbol{\gamma_{ij}}(k) = [x_{\gamma_{ij}}(k), y_{\gamma_{ij}}(k)]^T$ derives from the attractive–repulsive action defined as:

$$\begin{aligned}
x_{\gamma_{ij}}(k) &= \begin{cases} k_1 \cdot (\bar{d}/d_{i,j}(k) - 1) \cdot \cos\alpha_{ij}(k) & \text{if } d_{i,j}(k) \leq \bar{d} \\ k_2 \cdot (d_{i,j}(k)/\bar{d} - 1) \cdot \cos(\alpha_{ij}(k) + \pi) & \text{if } d_{i,j}(k) > \bar{d} \end{cases} \\
y_{\gamma_{ij}}(k) &= \begin{cases} k_1 \cdot (\bar{d}/d_{i,j}(k) - 1) \cdot \sin\alpha_{ij}(k) & \text{if } d_{i,j}(k) \leq \bar{d} \\ k_2 \cdot (d_{i,j}(k)/\bar{d} - 1) \cdot \sin(\alpha_{ij}(k) + \pi) & \text{if } d_{i,j}(k) > \bar{d} \end{cases}
\end{aligned} \tag{5}$$

where k_1 and k_2 are positive coefficients used to tune the repulsive/attractive component in the overall potential field.

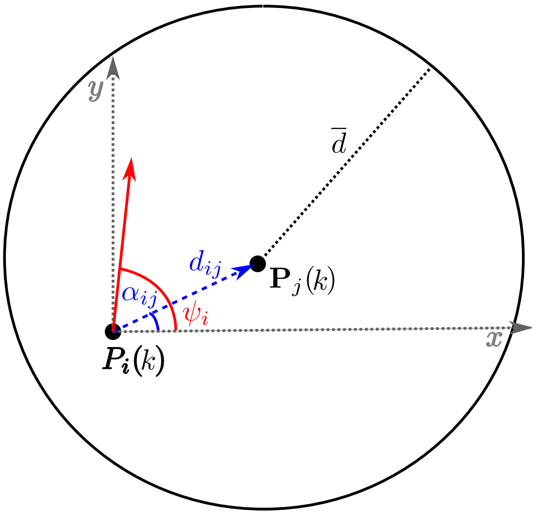


Figure 2. Reference angles for potential field computation. Red arrow represents the *i*-th vehicle's flight direction. The repulsive force driven by *j* forces vehicle *i* to stay on the circumference with a radius equal to the desired distance $\bar{d}$.

In order to establish a hierarchy in the propagation of the potential field action, a *relative position* operator $\Xi_{ij}(k)$ is then defined establishing whether aircraft j is flying ahead ($\Xi_{ij}(k) = 1$) or behind ($\Xi_{ij}(k) = 0$) aircraft i, where:

$$\Xi_{ij}(k) = \begin{cases} 1 & \text{if } |\Delta\alpha_{ij}(k)| \leq \pi/2 \\ 0 & \text{if } |\Delta\alpha_{ij}(k)| > \pi/2 \end{cases} \tag{6}$$

and

- $\Delta\alpha_{ij}(k) = \lfloor \alpha_{ij}(k) - \psi_i(k) \rceil_{-\pi}^{\pi}$,
- the operator $\lfloor \cdot \rceil_{-\pi}^{\pi}$ wraps the angle in the interval $[-\pi, \pi]$,

- $\alpha_{ij}(k) = \tan^{-1}\left\lfloor \frac{y_j(k)-y_i(k)}{x_j(k)-x_i(k)} \right\rfloor_0^{2\pi}$ is the angle between the vector $\mathbf{P}_j(k) - \mathbf{P}_i(k)$ and the horizontal axis, and
- $\psi_i(k)$ is the heading of the i-th aircraft.

The final desired direction $\mathbf{\Gamma}_i(k) = [x_{\Gamma_i}(k), y_{\Gamma_i}(k)],^T$ of the i-th aircraft is computed as a weighted averaged action from the visible neighboring aircraft, combined with a potential action driving to the target point:

$$\mathbf{\Gamma}_{\boldsymbol{i}}(k) = \frac{\sum_{j\in\overline{\mathcal{Q}}_i(k)} \Xi_{ij}(k)\gamma_{ij}(k)}{\mathrm{card}(\overline{\mathcal{Q}}_i(k))} + k_{Target}\frac{\mathbf{P}^f - \mathbf{P}_i(k)}{\left\|\mathbf{P}^f - \mathbf{P}_i(k)\right\|} \tag{7}$$

where $\mathrm{card}(\overline{\mathcal{Q}}_i(k))$ represents the cardinality of the set $\overline{\mathcal{Q}}_i(k)$. The k_{Target} positive constant is chosen to balance the attraction to the target with all other actions.

The resulting heading angle is finally computed as follows:

$$\tilde{\psi}_i(k) = \left\lfloor \tan^{-1}\left(\frac{y_{\Gamma_i}(k)}{x_{\Gamma_i}(k)}\right)\right\rfloor_0^{2\pi} \tag{8}$$

Additionally, the desired speed $\tilde{V}_i(k)$ is a combination of the reference cruise speed V_c and a repulsive or attractive action from neighboring vehicles, $k_V > 0$ being a tuning parameter:

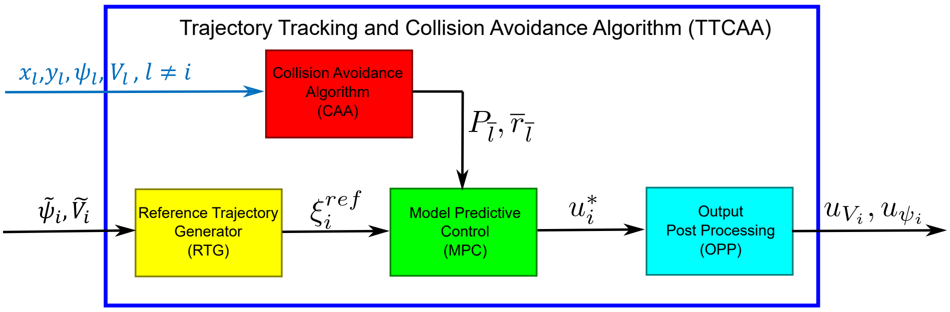
$$\tilde{V}_i(k) = V_c + k_V \sum_{j\in\overline{\mathcal{Q}}_i(k)} (2\Xi_{ij}(k) - 1)\frac{d_{i,j}(k) - \overline{d}}{|d_{i,j}(k) - \overline{d}|}||\gamma_{ij}(k)|| \tag{9}$$

In this way, vehicle i is attracted to vehicle j if the mutual distance $d_{i,j}(k) > \overline{d}$ and vice versa.

5. Trajectory Tracking and Collision Avoidance Algorithm

The TTCAA runs on each aicraft i and is composed of the following parts (see Figure 3):

- A Reference Trajectory Generation (RTG): on the basis of the inputs from the SGA, it computes the reference trajectory for the MPC;
- A Collision Avoidance Algorithm (CAA): it adds constraints to the MPC problem, whenever a potential collision with other vehicles or obstacles is detected.
- Model Predictive Control (MPC): it computes the acceleration vector needed to follow the reference trajectory and avoid any potential collision;
- MPC Output Post Processing (OPP): it converts acceleration into control signal for the FCS, namely acceleration along the flight trajectory u_{V_i} and turn rate u_{ψ_i}.



Trajectory Tracking and Collision Avoidance Algorithm (TTCAA)
$x_l, y_l, \psi_l, V_l, l \neq i$
Collision Avoidance Algorithm (CAA)
$P_{\overline{l}}, \overline{r}_{\overline{l}}$
$\tilde{\psi}_i, \tilde{V}_i$
Reference Trajectory Generator (RTG)
ξ_i^{ref}
Model Predictive Control (MPC)
u_i^*
Output Post Processing (OPP)
u_{V_i}, u_{ψ_i}

Figure 3. Trajectory Tracking and Collision Avoidance Algorithm.

According to the MPC strategy, at each time step k, the algorithm computes the optimal control action $\mathbf{u}_{\mathbf{i}}^*(k|k)$ as the first element of the optimal sequence $\mathbf{U}_i^*(k) = [\mathbf{u}_{\mathbf{i}}^{*T}(k|k), \mathbf{u}_{\mathbf{i}}^{*T}(k+1|k), ..., \mathbf{u}_{\mathbf{i}}^{*T}(k+n_c|k)]^T$, obtained by solving an optimal control problem over a finite prediction horizon $[k, k+n_p]$ [82].

Generally speaking the MPC requires the solution of the following problem.

Problem 1. *Constrained discrete time control optimization problem. Find the optimal sequence* $\mathbf{U}_i^*(k)$ *composed of* n_c *control moves, with* $n_c \leq n_p$*, such that:*

$$\mathbf{U}_i^*(k) = \underset{\mathbf{U}_i}{\operatorname{argmin}} J\left(\boldsymbol{\xi}_i\left(k|k\right), \mathbf{U}_i\left(k\right), k\right) \tag{10}$$

subject to

$$\boldsymbol{\eta}\left(\boldsymbol{\xi}_i\left(k+p|k\right), \mathbf{u}_i\left(k+p|k\right), k\right) \leq \mathbf{0} \quad \forall p \in [0, ..., n_p] \tag{11}$$

$$\boldsymbol{\xi}_i(k+p+1|k) = \mathbf{f}(\boldsymbol{\xi}_i(k+p|k), \mathbf{u}_i(k+p|k), k) \quad \forall p \in [0, ..., n_p] \tag{12}$$

$$\boldsymbol{\xi}_i\left(k|k\right) = \boldsymbol{\xi}_i\left(k\right)$$

where:

- $J(\cdot,\cdot,\cdot)$ *is the cost function to be minimized;*
- $\boldsymbol{\xi}_i\left(k+p+1|k\right)$ *is the predicted state at the time step* $k+p+1, \forall p \in [0, ..., n_p]$*. This is computed using (12) which represents the dynamics of the aircraft, comprehensive of the initial condition* $\boldsymbol{\xi}(k)$ *which is measured or estimated;*
- $\mathbf{u}_i(k+p|k)$ *is the control signal at the time instant* $k+p, \forall p \in [0, ..., n_p]$*;*
- *Equation (11) defines static constraints involving states and inputs.*

In particular, to predict the aircraft state trajectory a linear kinematic model is considered, whose state-space representation is:

$$\boldsymbol{\xi}_i(k+1) = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{\xi}_i(k) + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \mathbf{u}_i(k) = \boldsymbol{A}\boldsymbol{\xi}_i(k) + \mathbf{B}\mathbf{u}_i(k) \tag{13}$$

where the state vector $\boldsymbol{\xi}_i(k) = \left[x_i(k), y_i(k), v_{i,x}(k), v_{i,y}(k)\right]^T$ is composed of the position $\mathbf{P}_i(k) = [x_i(k), y_i(k)]^T$ and the velocity vector $\mathbf{V}_i(k) = [v_{i,x}(k), v_{i,y}(k)]^T$ of the i-th aircraft, and $\mathbf{u}_i(k) = \left[a_{i,x}(k), a_{i,y}(k)\right]^T$ is the acceleration vector that must be optimized in order to allow the aircraft to follow the reference trajectory $\boldsymbol{\xi}_i^{ref}(k) = [x_i^{ref}(k+p+1|k), y_i^{ref}(k+p+1|k), v_{i,x}^{ref}(k+p+1|k), v_{i,y}^{ref}(k+p+1|k)]^T$ ($p = 0, ..., n_p$). This trajectory can be computed by integrating the following equation

$$\begin{cases} x_i^{ref}(k+p+1|k) = x_i^{ref}(k+p|k) + v_{i,x}^{ref}(k+p|k)\Delta t \\ y_i^{ref}(k+p+1|k) = y_i^{ref}(k+p|k) + v_{i,y}^{ref}(k+p|k)\Delta t \\ v_{i,x}^{ref}(k+p+1|k) = \tilde{V}_i(k)\cos\tilde{\psi}_i(k) \\ v_{i,y}^{ref}(k+p+1|k) = \tilde{V}_i(k)\sin\tilde{\psi}_i(k) \end{cases} \tag{14}$$

over the prediction horizon $[k, k+n_p]$, starting from the initial condition $\boldsymbol{\xi}_i^{ref}(k|k) = [x_i(k), y_i(k), \tilde{V}_i(k)\cos\tilde{\psi}_i(k), \tilde{V}_i(k)\sin\tilde{\psi}_i(k)]^T$, where $[x_i(k), y_i(k)]$ represents the measured current position, $\tilde{V}_i(k)$ and $\tilde{\psi}_i(k)$ are provided by the SGA.

In order to obtain an online numerical solution to the MPC Problem 1 with Quadratic Programming (QP) techniques, the objective function is defined quadratic in the state and input variables as follows:

$$J(\boldsymbol{\xi}_i(k|k), \mathbf{U}_i, k) = \sum_{p=0}^{n_p} \mathbf{e}_{\xi_i}(k+p+1|k)^T \mathbf{Q}\mathbf{e}_{\xi_i}(k+p+1|k) + \\ + \sum_{p=0}^{n_c} \mathbf{u}_i(k+p|k)^T \mathbf{R}\mathbf{u}_i(k+p|k) \tag{15}$$

The first term of the objective function J accounts for the difference between the predicted and reference trajectory, $\mathbf{e}_{\xi_i}(\tau|k) = \boldsymbol{\xi}_i(\tau|k) - \boldsymbol{\xi}_i^{ref}(\tau|k)$, whereas the second one aims to limit the control effort. Weighting matrices $\mathbf{Q}$ and $\mathbf{R}$ are positive definite.

On the other hand, an effort has to be spent to formulate constraints as linear functions of the optimization variable.

5.1. Acceleration and Speed Limits

Control laws must be consistent with the aircraft acceleration and speed limits.

Speed is limited by the following nonlinear constraints:

$$V_{i,min} \leq V_i(k+p+1|k) \leq V_{i,max} \quad \forall p = 0, ..., n_p \tag{16}$$

where $V_{i,min}$ and $V_{i,max}$ are the minimum and maximum allowed speed and $V_i = \sqrt{v_{i,x}^2 + v_{i,y}^2}$.

Equation (16) defines a set of non-convex and nonlinear constraints. In fact, as shown in Figure 4, the velocity vector has always to be kept within the region between two concentric circles $\mathcal{C}_{V_{i,min}}$ and $\mathcal{C}_{V_{i,max}}$, with radius equal to $V_{i,min}$ and $V_{i,max}$ respectively. For this reason, $\mathcal{C}_{V_{i,max}} - \mathcal{C}_{V_{i,min}}$ represents the set of admissible $\mathbf{V}_i$.

In order to define a linear and convex constraint, let consider the two polytopes $\mathcal{P}_{V_{i,min}}$ and $\mathcal{P}_{V_{i,max}}$ with $n_v \geq 3$ vertices, such that $\mathcal{C}_{V_{i,min}}$ is inscribed in $\mathcal{P}^i_{V_{min}}$ and $\mathcal{C}_{V_{i,max}}$ is circumscribed at $\mathcal{P}^i_{V_{max}}$ (see Figure 4). Therefore the set of admissible velocity vector, defined as $\mathcal{C}_{V_{i,max}} - \mathcal{C}_{V_{i,min}}$, is approximated by the set difference $\mathcal{P}^i_{V_{max}} - \mathcal{P}^i_{V_{min}}$.

Let $\mathfrak{P}^i = \{\mathfrak{P}^i_s, s = 1, 2, ..., n_v\}$ be the set of polytopes needed to define $\mathcal{P}^i_{V_{max}} - \mathcal{P}^i_{V_{min}}$ where:

$$\mathfrak{P}^i_s = \{\mathbf{V_i}(k+p+1|k) \in \mathbb{R}^2 : \quad \boldsymbol{\mu}^v_{i,s}\mathbf{V_i}(k+p+1|k) \leq \boldsymbol{\nu}^v_{i,s}\} \tag{17}$$

with $\boldsymbol{\mu}^v_{i,s}$ and $\boldsymbol{\nu}^v_{i,s}$ constant matrix and vector, respectively.

Consequently, the set of non-convex linear constraints $\mathbf{V_i}(k+p+1|k) \in \mathfrak{P}^i$ can be converted into a convex one, by using the so-called big-M reformulation [83] that results in a MIQP (Mixed Integer Quadratic Programming) reformulation of the problem. This approach requires to increase the number of optimization variables, by introducing n_v additional binary variables $\delta^v_{i,s}(k+p+1|k) \in \{0, 1\}$ $(s = 1, 2..., n_v)$, at each time step $k+p$ $(p = 0, ..., n_p)$. Velocity constraints (16) are then reformulated as follows:

$$\begin{cases} \boldsymbol{\mu}^v_{i,1}\mathbf{V_i}(k+p+1|k) \leq \boldsymbol{\nu}^v_{i,1} + \mathbf{M}(1 - \delta^v_{i,1}(k+p+1|k)) \\ \boldsymbol{\mu}^v_{i,2}\mathbf{V_i}(k+p+1|k) \leq \boldsymbol{\nu}^v_{i,2} + \mathbf{M}(1 - \delta^v_{i,2}(k+p+1|k)) \\ \vdots \\ \boldsymbol{\mu}^v_{i,n_v}\mathbf{V_i}(k+p+1|k) \leq \boldsymbol{\nu}^v_{i,n_v} + \mathbf{M}(1 - \delta^v_{i,n_v}(k+p+1|k)) \\ \sum_{s=1}^{n_v} \delta^v_{i,s}(k+p+1|k) = 1 \end{cases} \forall p = 0, ..., n_p \tag{18}$$

where $\mathbf{M} = [M_x, M_y]^T$, M_x and M_y being sufficiently large numbers.

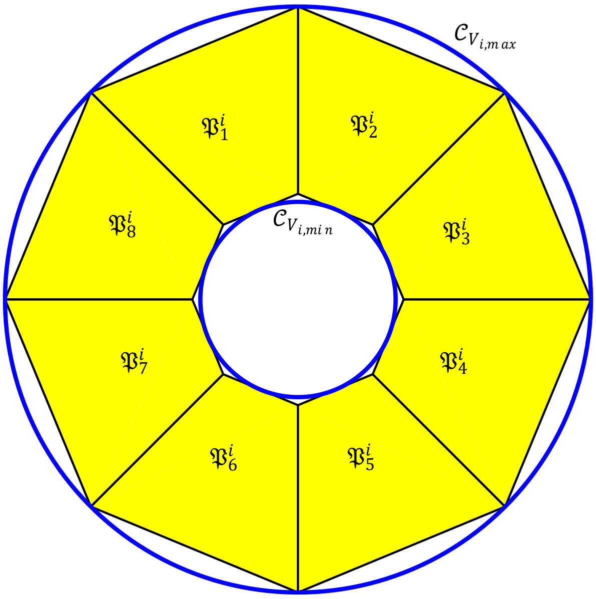


Figure 4. Velocity vector domain and its polytopic approximation. Blue circles represent the maximum and minimum allowed speed.

On the other hand, acceleration constraints can be defined as follows:

$$\begin{bmatrix} a_{i,min}^{\|}(k) \\ a_{i,min}^{\perp}(k) \end{bmatrix} \leq \begin{bmatrix} a_i^{\|}(k+p|k) \\ a_i^{\perp}(k+p|k) \end{bmatrix} \leq \begin{bmatrix} a_{i,max}^{\|}(k) \\ a_{i,max}^{\perp}(k) \end{bmatrix} \quad \forall p = 0, ..., n_p \tag{19}$$

where superscripts $\|$ and $\perp$ indicate tangential and normal components to the flight trajectory. The values of maximum and minimum accelerations, $a_{i,min}^{\|}(k)$, $a_{i,min}^{\perp}(k)$, $a_{i,max}^{\|}(k)$, and $a_{i,max}^{\perp}(k)$, are related to the available thrust and the turn rate capabilities of the aircraft. In particular, $a_i^{\|}(k)$ is produced with the excess of thrust with respect to drag force in the wing levelled forward flight approximation, whereas normal acceleration limits is related to the minimum turning radius r_i as follows:

$$a_i^{\perp}(k) \approx \frac{V_i(k)^2}{r_i} \tag{20}$$

In order to obtain linear constraints, accelerations have to be translated into the global reference system (flat Earth-fixed reference frame). Let $\mathbf{T}_i$ be the rotation matrix from the local reference frame to the global one. Such a matrix is precomputed at time k and assumed constant in the prediction horizon $\mathbf{T}_i\,(k+p|k) = \mathbf{T}_i\,(k|k)$. Consequently, the acceleration constraints can be formulated as the following set of linear inequalities:

$$\mathbf{T}_i\,(k|k) \begin{bmatrix} a_{i,min}^{\|}(k) \\ a_{i,min}^{\perp}(k) \end{bmatrix} \leq \begin{bmatrix} a_{i,x}(k+p|k) \\ a_{i,y}(k+p|k) \end{bmatrix} \leq \mathbf{T}_i\,(k|k) \begin{bmatrix} a_{i,max}^{\|}(k) \\ a_{i,max}^{\perp}(k) \end{bmatrix} \quad \forall p = 0, ..., n_p \tag{21}$$

Let's finally define an enlarged vector of design variables $\boldsymbol{\Omega_i(k)} = [\mathbf{u}_i(k|k)^T, ..., \mathbf{u}_i(k+n_c|k)^T, \delta_{i,s}^v(k+1|k), ..., \delta_{i,s}^v(k+n_p+1|k)]^T$. Problem 1 is then reformulated as follows:

Problem 2. *At each time step k, find the optimal sequence of design variables $\boldsymbol{\Omega_i(k)}$ such that it minimizes the cost function (15), in presence of linear constraints (18) and (21).*

Problem 2 turns out to be a Mixed Integer Quadratic Programming problem, which has efficient numerical solvers available [84,85].

According to the receding horizon paradigm, the output of the MPC is the control signal at $p = 0$, $\mathbf{u_i^*}(k|k) = \left[a^*_{i,x}(k|k), a^*_{i,y}(k|k)\right]^T$.

The OPP block computes the speed change u_{V_i} and the turn rate u_{ψ_i} to be passed to the aircraft FCS as follows:

$$\begin{aligned} u_{V_i}(k) &= \frac{2v_{i,x}(k)a^*_{i,x}(k|k)+2v_{i,y}(k)a^*_{i,y}(k|k)}{\sqrt{v_{i,x}(k)^2+v_{i,y}(k)^2}} \\ u_{\psi_i}(k) &= \frac{v_{i,x}(k)a^*_{i,y}(k|k)-v_{i,y}(k)a^*_{i,x}(k|k)}{v_{i,x}(k)^2+v_{i,y}(k)^2} \end{aligned} \tag{22}$$

5.2. Collision Avoidance Algorithm and Constraints

The Collision Avoidance Algorithm (CAA) is designed to guarantee that each aircraft in the formation is able to avoid collision with the other vehicles or with any existing obstacle.

Consider $N_o = N_{ob} + (N-1)$ equivalent obstacles $\mathfrak{C}_j(k)$ (N_{ob} real obstacles and $N-1$ aircraft of the swarm). Assume that each obstacle j is modeled as a circumference of radius $\bar{r}_j$, centered at point $\overline{\mathbf{C}}_j(k) = [\bar{x}_j(k), \bar{y}_j(k)]^T$.

The CAA verifies if any distance between the i-th aircraft and the obstacles is less than or equal to a fixed safety distance d_{safety}, evaluating the set of possible colliding obstacles $\mathcal{O}_i(k)$.

Definition 5. *The set of possible colliding obstacles $\mathcal{O}_i(k)$ at time k is defined as:*

$$\mathcal{O}_i(k) = \left\{\forall \mathfrak{C}_j(k) : \left|\mathbf{P}_i(k) - \overline{\mathbf{C}}_j(k)\right| - \bar{r}_j - \bar{r}_i \le d_{safety}\right\} \tag{23}$$

where $\bar{r}_i$ and $\bar{r}_j$ are the equivalent radius of the i-th aircraft and j-th obstacle respectively.

In accordance with the reactive scheme [86,87], only the nearest colliding obstacle within set $\mathcal{O}_i(k)$ is taken into account by the MPC [8,78] by adding a limited number of constraints to the Problem 2.

Without loss of generality, assume that the forthcoming potential collision, for the vehicle i, is going to occur with the aircraft $\bar{l}$, as shown in Figure 5.

For all the time instants in the prediction horizon $k + p$ ($p = 1, ..., n_p$), the UAV position is constrained to fall within a region delimited by two straight lines. This guarantees the desired safety distance d_{safety}.

The first boundary line $\mathfrak{r}^1_i(k+p|k)$ is normal to the vector $\mathbf{P}_{\bar{l}}(k+p|k) - \mathbf{P}_i(k+p|k)$ and tangent to the circumference $\mathfrak{C}_{\bar{l}}$ with radius $\bar{r}_{\bar{l}}$, centered in the predicted position $\mathbf{P}_{\bar{l}}(k+p|k) = [\hat{x}_{\bar{l}}(k+p|k), \hat{y}_{\bar{l}}(k+p|k)]^T$.

Since there is no cooperation between vehicles, the obstacle future positions $\mathbf{P}_{\bar{l}}(k+p|k) = [\hat{x}_{\bar{l}}(k+p|k), \hat{y}_{\bar{l}}(k+p|k)]^T$, over the prediction horizon ($p = 1, ..., n_p + 1$), are estimated assuming that both speed and heading are constant, i.e. $V_{\bar{l}}(k+p|k) = V_{\bar{l}}(k|k)$, $\psi_{\bar{l}}(k+p|k) = \psi_{\bar{l}}(k|k)$.
Line $\mathfrak{r}^1_i(k+p|k)$ is defined as follows :

$$\mathfrak{r}^1_i(k+p|k) : c_{\alpha,i}(k+p|k)\left(x - \hat{x}_{\bar{l}}(k+p|k)\right) + s_{\alpha,i}(k+p|k)\left(y - \hat{y}_{\bar{l}}(k+p|k)\right) - \bar{r}_{\bar{l}} = 0 \tag{24}$$

with

$$[c_{\alpha,i}(k+p|k), s_{\alpha,i}(k+p|k)]^T = \frac{\mathbf{P}_{\bar{l}}(k+p|k) - \mathbf{P}_i(k)}{||\mathbf{P}_{\bar{l}}(k+p|k) - \mathbf{P}_i(k)||} \tag{25}$$

Another line delimiting the region that can be occupied by the aircraft trajectory is that reported as $\mathfrak{r}^2_i(k+p|k)$ in Figure 5. This is tangent to the obstacle $\mathfrak{C}_{\bar{l}}$ and passes through a point $\mathbf{P}_i^{ref^*} = \mathbf{P}_i^{ref}(k+p^*|k) = (x_i^{ref}(k+p^*|k), y_i^{ref}(k+p^*|k))$, on the reference trajectory ($p^* \ge 1$), which is ahead the current aircraft position. This choice to look forward gives enough space and time to maneuver in order to avoid the forbidden regions.

For each time step $k + p$ ($p = 1, ..., n_p$), consider the vector $\mathbf{P}_{\bar{l}}(k+p|k) - \mathbf{P}_i^{ref}(k+p^*|k)$. Its direction with respect to the x axis is defined by the angle $\beta_{i\bar{l}}(k+p|k)$.

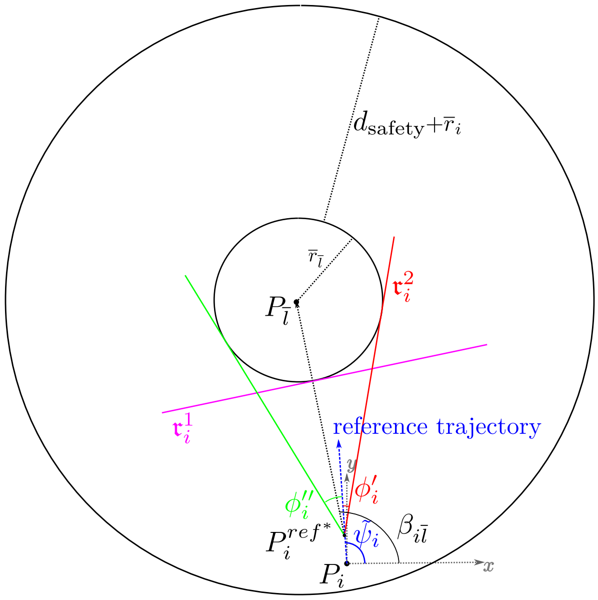


Figure 5. Collision Avoidance constraints (Time information has been omitted). Obstacle is described by circumference centered in the point $\mathbf{P}_j$; tangent lines $\mathfrak{r}_i^1$ and $\mathfrak{r}_i^2$ define the admissible region for the i-th future position.

Compute the angles:

$$\phi_i'(k+p|k) = -\arcsin(\bar{r}_{\bar{l}}/||\mathbf{P}_{\bar{l}}(k+p|k) - \mathbf{P}_i^{ref}(k+p^*|k)||)$$

$$\phi_i''(k+p|k) = \arcsin(\bar{r}_{\bar{l}}/||\mathbf{P}_{\bar{l}}(k+p|k) - \mathbf{P}_i^{ref}(k+p^*|k)||)\,.$$

The CAA algorithm selects the angle $\phi_i(k+p|k) \in \{\phi_i'(k+p|k), \phi_i''(k+p|k)\}$ minimizing the deviation from the desired heading $\tilde{\psi}_i(k)$.

Line $\mathfrak{r}_i^2(k+p|k)$ is defined as follows

$$\mathfrak{r}_i^2(k+p|k) : s_{\theta,i}(k+p|k)\left(x - x_i^{ref}(k+p^*|k)\right) - c_{\theta,i}(k+p|k)\left(y - y_i^{ref}(k+p^*|k)\right) = 0 \tag{26}$$

where $c_{\theta,i}(k+p|k) = \cos(\theta_i(k+p|k))$, and $s_{\theta,i}(k+p|k) = \sin(\theta_i(k+p|k))$, with $\theta_i(k+p|k) = \lfloor \beta_{i\bar{l}}(k+p|k) + \phi_i(k+p|k) \rfloor_{-\pi}^{\pi}$

It is worth noticing that constraints defined by $\mathfrak{r}_i^2(k+p|k)$ are not strictly needed to perform collision avoidance which could be managed in terms of speed optimization. However this kind of constraint, increases the overall performance of the algorithm implementing the anti-collision action well in advance with respect to the approaching obstacle. For example, without this constraint, in a symmetric condition, with the aircraft reference trajectory pointing to the center of the obstacle, the vehicle would tend to decrease its speed to the minimum, before performing a turn induced by the potential action spreading from the obstacle.

On the other hand, near the obstacle, this constraint may lead to an unfeasible problem being the aircraft initial position not in the feasible region. For this reason, the boundary line $\mathfrak{r}_i^2(k+p|k)$ is deactivated under certain conditions according to the following activation operator:

$$\Lambda_{i,\bar{l}}(k+p|k) = \begin{cases} 1 & \text{if } |\beta_{i\bar{l}}(k+p|k) - \psi_i(k)| \leq \pi/3 \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

The following equations define the admissible region over the prediction horizon $[k, k+n_p]$ as constraints to be added to the MPC Problem:

$$c_{\alpha,i}(k+p|k)\left(x-\hat{x}_{\bar{l}}(k+p|k)\right)+s_{\alpha,i}(k+p|k)\left(y-\hat{y}_{\bar{l}}(k+p|k)\right)-\bar{r}_{\bar{l}}=0 \tag{28}$$

$$\Lambda_{i,\bar{l}}(k+p|k)\left(s_{\theta,i}(k+p|k)\left(x-\hat{x}_i^{ref}(k+p^*|k)\right)-c_{\theta,i}(k+p|k)\left(y-\hat{y}_i^{ref}(k+p^*|k)\right)\right)\leq 0 \tag{29}$$

for each time step $k+p$, with $p=0,...,n_p$.

6. Numerical Results

To test and validate the proposed guidance system architecture, several numerical simulations were carried out using an ad-hoc simulator developed in a Matlab/Simulink environment. All of the operational scenarios are compliant with homogeneous swarms of micro/mini quadrotor-type UAVs having 10–15 min of flight endurance [88]. The proposed procedure was implemented using Operator Splitting Quadratic Program (OSQP) [89,90] to solve the MPC optimization problem.

The simulation parameters are presented in Table 1.

Table 1. Simulation parameters.

Description	**Value**
Cruise speed V_c [m/s]	5
Minimum speed V_{min} [m/s]	3
Maximum speed V_{max} [m/s]	10
Minimum normal acceleration $a_{min}^{\perp}$ [m/s^2]	-5
Maximum normal acceleration $a_{max}^{\perp}$ [m/s^2]	5
Minimum tangential acceleration $a_{min}^{\parallel}$ [m/s^2]	-10
Maximum tangential acceleration $a_{min}^{\parallel}$ [m/s^2]	10
Safety distance d_{safety} [m]	15
Aircraft size $\bar{r}_j$ $\forall j=1,...,N$ [m]	5
Desired distance $\bar{d}$ [m]	40
Weight matrix for tracking error **Q**	diag([10 10 10 10])
Weight matrix for control effort **R**	diag([1 1])
Number of steps of the prediction horizon n_p	10
Number of steps of the control horizon n_c	5

The position of each aircraft is monitored, checking if collisions are avoided and if the shape of the flight formation is re-established after passing obstacles. A numerical indicator, namely d_{mean}, is also calculated to check the algorithm effectiveness in keeping the swarm on a triangular mesh. It represents the average value of the distance between every pair of visible aircraft in the Delaunay triangulation and is defined as:

$$d_{\text{mean}}(k)=\frac{\sum_{i=1}^{N}\sum_{j\in\overline{\mathcal{Q}}_i(k)}d_{i,j}(k)}{\sum_{i=1}^{N}\text{card}(\overline{\mathcal{Q}}_i(k))} \tag{30}$$

To highlight the features of the proposed guidance algorithm, three scenarios were considered. Each scenario lies in a box of 1 × 1 km, simulating an environment with different obstacle arrangement and size.

Results with three, five and 10 flying vehicles are presented in Scenario #1, and 10 vehicles are used in Scenarios #2 and #3.

Aircraft initial conditions were assigned such that the swarm initial mesh is not composed of equilateral triangles as desired. Simulations were repeated 10 times for each scenario to avoid different initial conditions affecting the analysis of the results.

6.1. Scenario #1—Three Aircraft

The first scenario is made up of three circular obstacles of different sizes, placed between the fleet's starting position and the target point $\mathbf{P}_f$. Figure 6 shows the resulting trajectories obtained involving three aircraft; markers are used to show vehicles positions at the fixed time instants $t = 0$ s, $t = 30$ s, $t = 65$ s, $t = 95$ s and $t = 160$ s, whereas green lines are used to highlight the DTG arcs. Figure 7 shows the distances between vehicle 1 and the other aircraft of the swarm. Solid lines indicate the distances from vehicle 1 from any other visible aircraft; whenever the aircraft are no longer visible dashed gray lines are used.

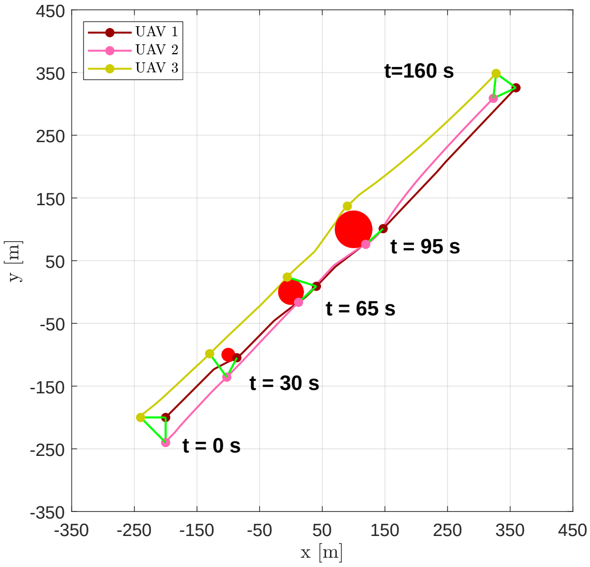


Figure 6. Scenario #1—three aircraft: UAVs' trajectories. Red circles represent the obstacles; colored markers indicate positions at different time instants; green straight lines highlight DTG arcs.

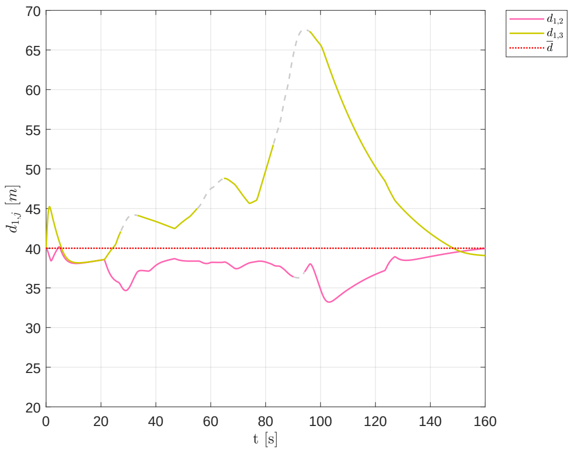


Figure 7. Scenario #1—three aircraft: distances between UAV 1 and the other swarm vehicles. Solid lines are used for visible aircraft, dashed gray lines are employed whenever they are no longer visible. The desired distance is highlighted using a dotted red line.

During the flight, in presence of obstacles, the swarm split into two parts (see aircraft positions at time instants $t = 32.9s$, $t = 59.7s$, and $t = 86.5s$ in Figure 8). Since aircraft 1 and 2 overcame the obstacles, passing on the same side, their mutual distance $d_{1,2}$ was always close to the desired distance. On the other hand, aircraft 3 passed on the other side of the obstacles and, consequently, distances $d_{1,3}$ and $d_{2,3}$ greatly deviated from the desired distance $\overline{d}$ near the obstacles. The goal of keeping a uniform distance between aircraft was achieved while avoiding any collision. As we can see, vehicle 2 almost always belonged to $\overline{\mathcal{Q}}_1$, the set of visible aircraft from vehicle 1. Vehicle 3, passing on the other side of the obstacles, became not visible from aircraft 1 during some time intervals. However, after passing the obstacles ($t \geq 95.9s$), the DTG was newly connected and aircraft tended to restore an equilateral triangles-based flight formation with edges equal to the desired distance $\overline{d}$.

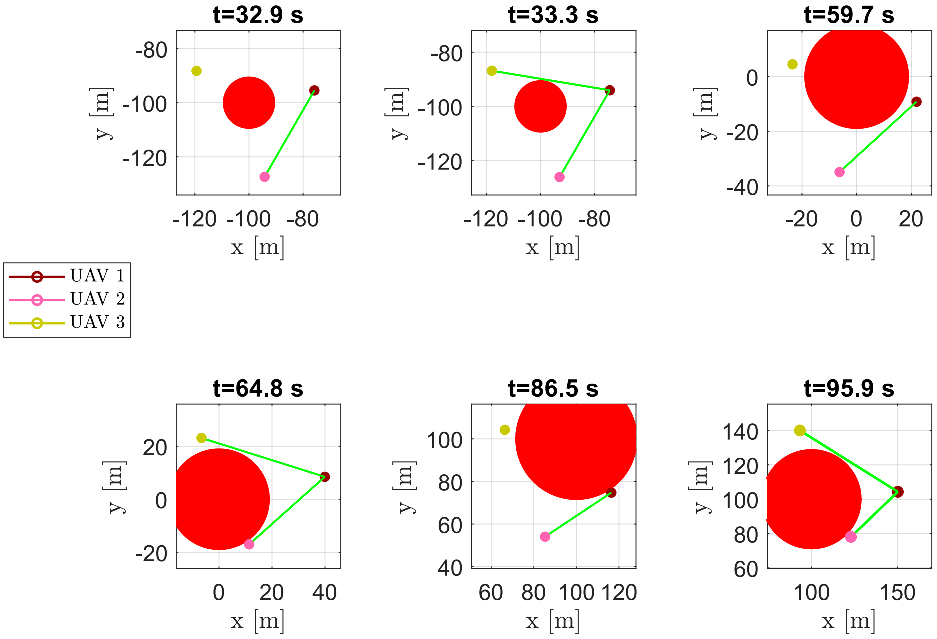


Figure 8. Scenario #1—three aircraft: snapshots of UAVs' positions at different time instants. Colored markers indicate UAVs' actual positions; straight green lines show the DTG arcs. Several swarm configurations during obstacle avoidance are highlighted.

Figure 9 shows the mean distance d_{mean}. As expected, this distance increased with respect to the desired one only in short time intervals, where the presence of obstacles led the aircraft to deviate more from the desired mutual distances.

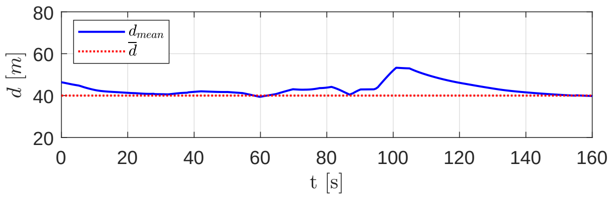
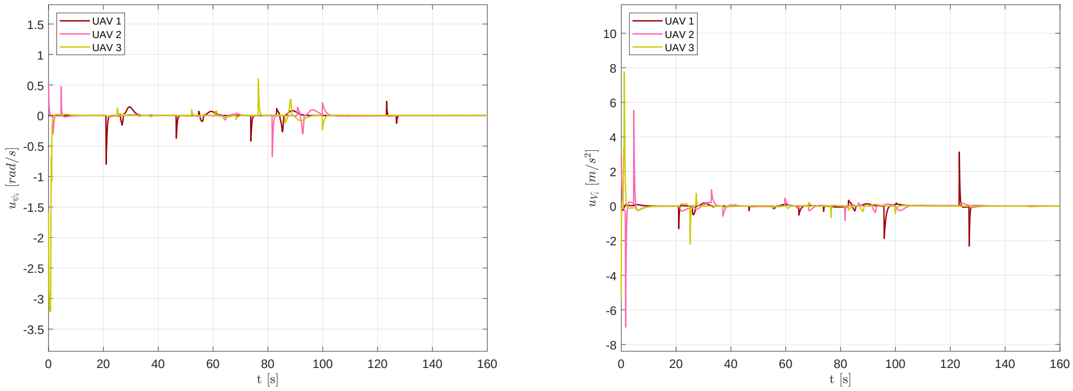


Figure 9. Scenario #1—three aircraft: mean distance between visible aircraft (blue solid line) compared with desired distance (red dotted line).

Time histories of the control signals u_ψ and u_V are presented in Figure 10. It is worth noticing that at the beginning of the simulation, the random initial position of the aircraft required them to accelerate and turn to establish the desired swarm shape. Furthermore, control signals presented peaks before passing obstacles at the activation of the collision avoidance and, after that, in order to re-compose the swarm.



(**a**) u_{ψ_i} (**b**) u_{V_i}

Figure 10. Scenario #1—three aircraft: control signal computed by TTCAA.

6.2. Scenario #1—Five Aircraft

The goal of keeping a uniform distance between aircraft was also achieved with a higher number of vehicles. Figure 11 shows the resulting trajectories obtained with five aircraft together with their positions (markers) and the DTG edges (green lines) at several time instants ($t = 0$ s, $t = 34$ s, $t = 60$ s, $t = 100$ s and $t = 160$ s).

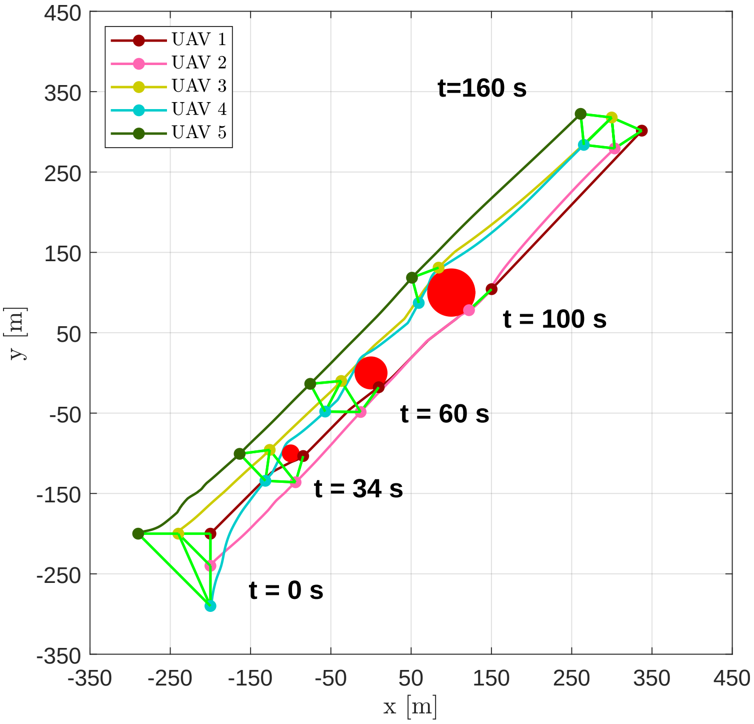


Figure 11. Scenario #1—five aircraft: UAVs trajectories. Red circles represent obstacles; colored markers indicate positions at different time instants; green straight lines highlight DTG arcs.

As shown in Figure 12, near obstacles the formation split into two parts, passing at the opposite sides of the obstacles; e.g., at time instant $t = 68.1s$ the DTG was not connected and was composed of two distinct sub-graphs made up of vehicles $\{3, 4, 5\}$ and $\{1, 2\}$, seperately.

In Figure 13 the distances between vehicle 1 and the others are shown. Aircraft 4 never belonged to the set of vehicles visible from UAV 1, $\overline{\mathcal{Q}}_1$, resulting in a distance $d_{1,4}$ (indicated with a dashed line) that was always greater than the desired distance. On the other hand, aircraft 2 was almost always visible from UAV 1, resulting in a distance approximately equal to the desired one during the flight, except in short time intervals, where collision avoidance prevented vehicles from the desired relative positioning. Vehicle 3, while passing on the opposite side of obstacles with respect to aircraft 1, tried to re-compose the swarm after them. Finally, at $t > 100s$, $d_{1,3}$ clearly tended to the desired distance $\overline{d}$.

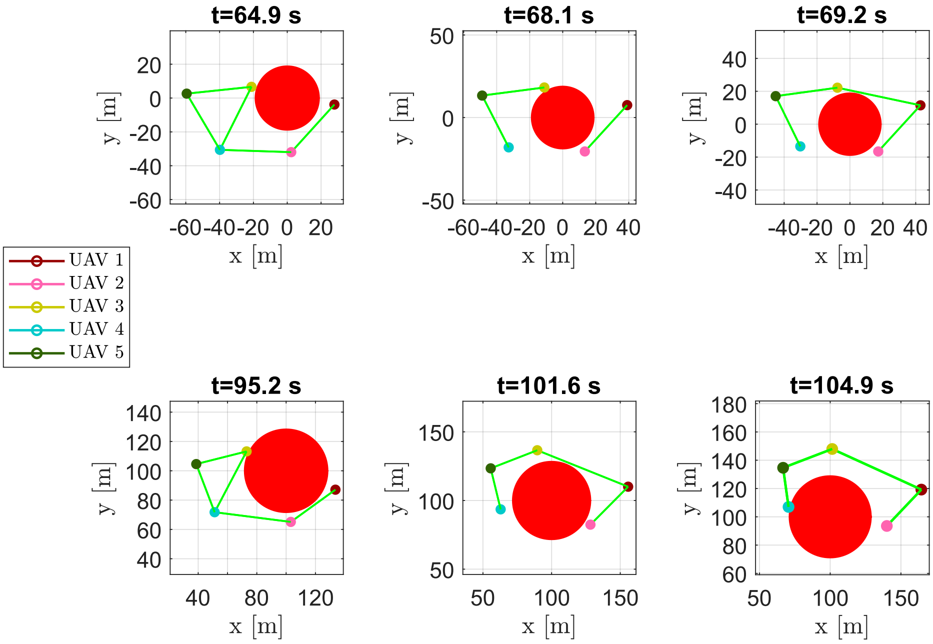


Figure 12. Scenario #1—five aircraft: snapshots of UAVs positions at different time instants. Colored markers indicate UAVs actual positions; straight green lines show the DTG arcs. During obstacle avoidance the DTG may became not connected, separating the formation into two distinct sub-graphs.

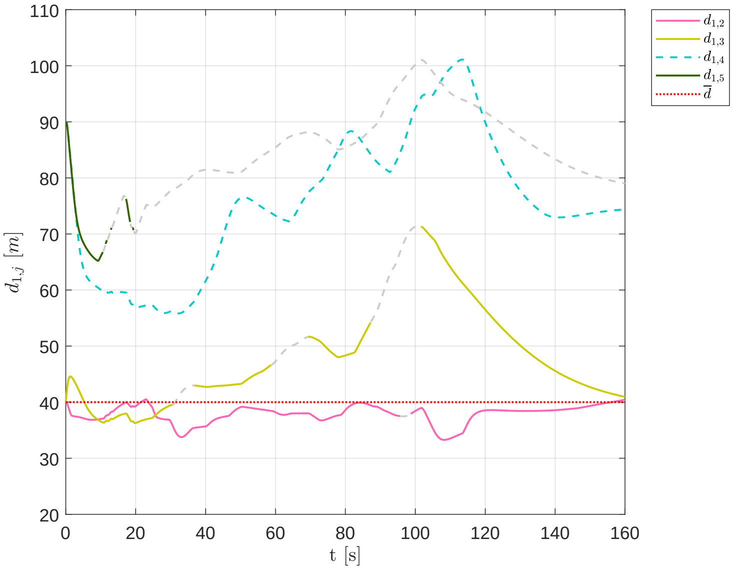


Figure 13. Scenario #1—five aircraft: distances between UAV 1 and the other swarm vehicles. Solid lines are used for visible aircraft, dashed gray lines are employed whenever they are no longer visible, whereas dashed colored lines are used for never visible UAVs. The desired distance is highlighted using a dotted red line.

It is worth noting that DTG, and consequently the swarm configuration, changes over time: for instance, at $t \leq 10$ s, as shown in Figure 13, aircraft 1 belonged to two triangles, one with vehicles 3 and 5 and the other with 2 and 3, while at approximately $t > 100$ s it belonged only to a triangle with 2 and 3.

Figure 14 shows the mean distance d_{mean} between vehicles over DTG. Although such a distance showed slight deviations in short time intervals, due to a change in swarm configuration in the presence of obstacles, it tended always to the desired distance $\overline{d}$.

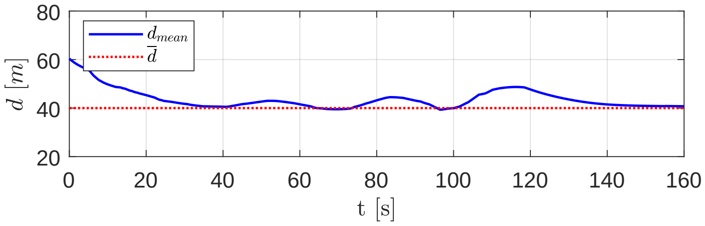


Figure 14. Scenario #1—five aircraft: mean distance between visible aircraft (blue solid line) compared with desired distance (dotted red line).

6.3. Scenario #1—10 Aircraft

Figure 15 shows the trajectories of a swarm of 10 aircraft, highlighting their positions (markers) and edges of DTG (green lines) at several time instants ($t = 0$ s, $t = 35$ s, $t = 75$ s, $t = 110$ s and $t = 160$ s).

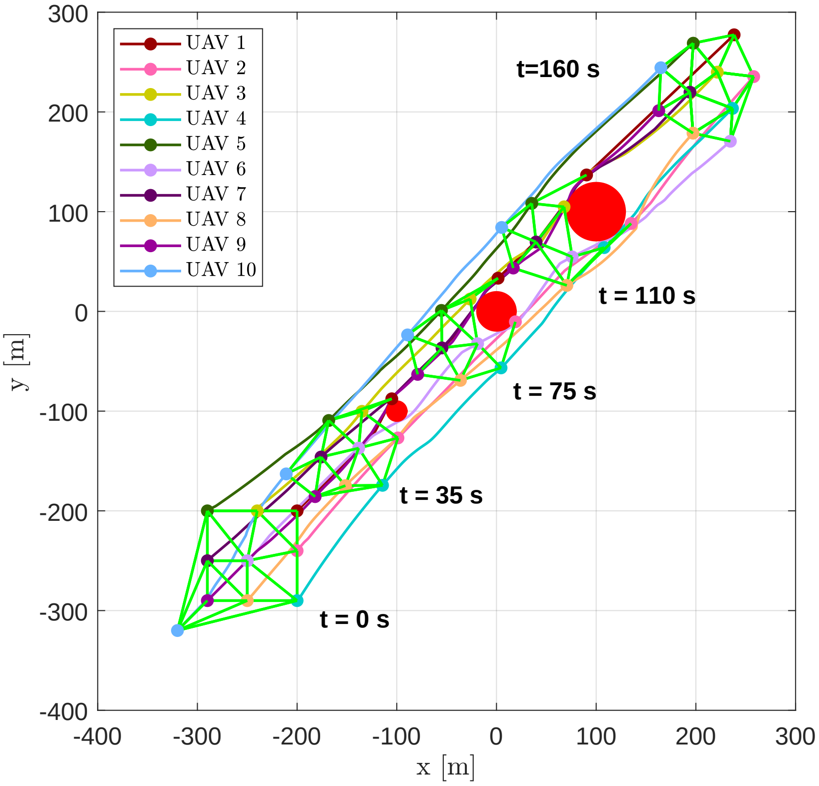


Figure 15. Scenario #1—10 aircraft: UAV trajectories. Red circles represent obstacles; colored markers indicate positions at different time instants; green straight lines highlight DTG arcs.

In Figure 16, snapshots of swarm configuration at time instants $t = 51$ s, $t = 63.6$ s, $t = 73$ s, $t = 80.2$ s, $t = 100.9$ s and $t = 106.6$ s are shown. It is worth noting that, although the vehicles passed on both sides of the obstacles, virtually splitting the swarm in two parts, the DTG remained connected, giving the ability to create a regular mesh also around the obstacles.

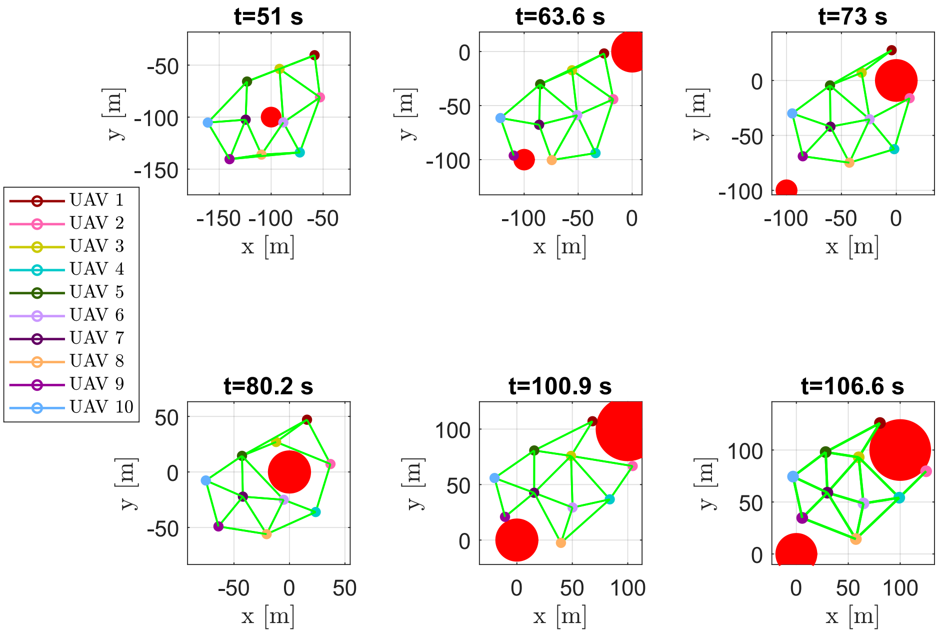


Figure 16. Scenario #1—10 aircraft: snapshots of aircraft positions at different time instants. Colored markers indicate UAVs' current positions; straight green lines show the arcs of modified DTG. Chosen time instants highlight swarm connectivity during obstacle avoidance.

In Figure 17, the distances of each vehicle from aircraft 1 are shown, highlighting that UAVs 4, 6, 7, 8, 9 and 10 never belonged to $\overline{\mathcal{Q}}_1$ (their distances are represented as colored dashed lines), whereas the remaining aircraft showed a continuously changing swarm configuration, finding an equilibrium only after the last obstacle ($t > 120$ s).

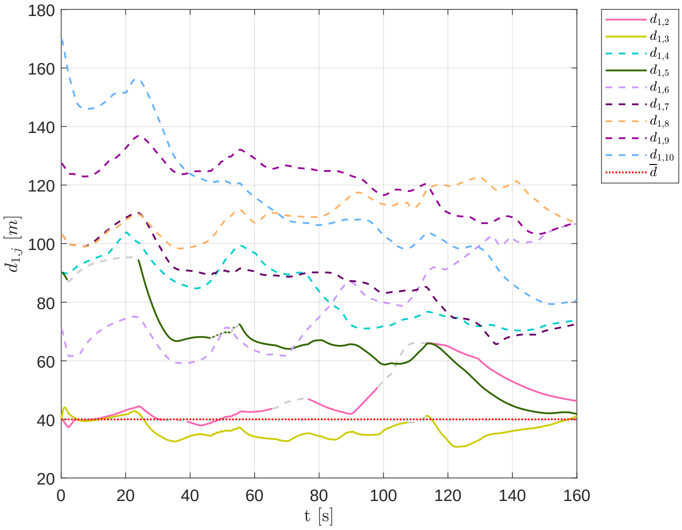


Figure 17. Scenario #1—10 aircraft: distances between UAV 1 and the other swarm vehicles. Solid lines are used for visible aircraft and dashed gray lines are employed whenever they are no longer visible, whereas dashed colored lines are used for never visible UAVs. The desired distance is highlighted using a dotted red line.

Figure 18 shows the distances of each vehicle from aircraft 10. Vehicles 5, 7 and 9 always belonged to $\overline{\mathcal{Q}}_{10}$ (their distances are represented as solid lines), whereas aircraft 4 and 8 belonged to the set of visible neighbors only at the beginning of the simulation.

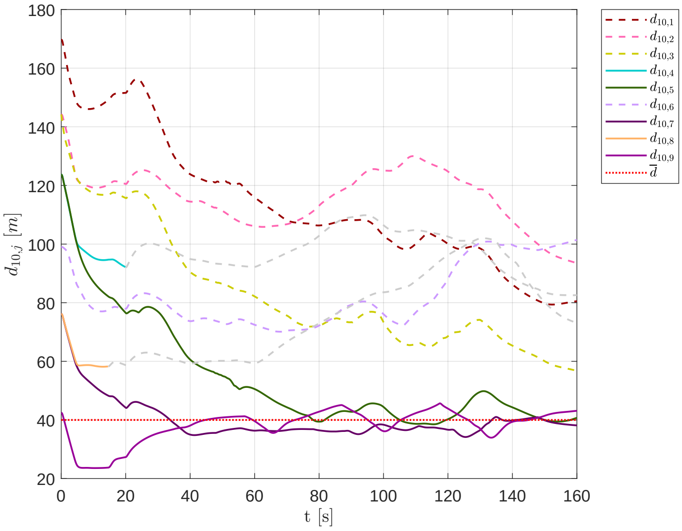


Figure 18. Scenario #1—10 aircraft: distances between UAV 10 and the other swarm vehicles. Solid lines are used for visible aircraft and dashed gray lines are employed whenever they are no longer visible, whereas dashed colored lines are used for never visible UAVs. The desired distance is highlighted using a dotted red line.

Looking at Figures 17 and 18, it is clear that aircraft tried to compose regular triangle-based sub-swarms with visible neighbors, also during the obstacle avoidance, maintaining the mutual distances as equally as possible to the desired distance.

Figure 19 presents the average distance between visible aircraft. As shown, aircraft were able to keep a mutual average distance very close to the desired one with slight deviations only when overcoming obstacles.

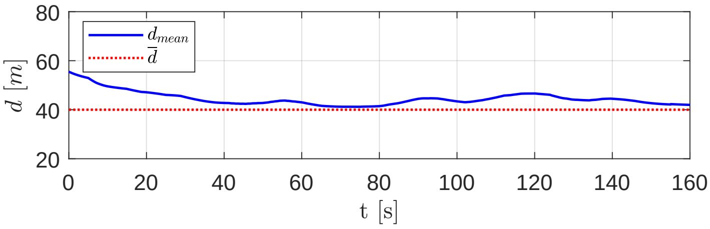


Figure 19. Scenario #1—10 aircraft: mean distance between visible aircraft (blue solid line), compared with desired distance (red dotted line).

6.4. Scenario #2—10 Aircraft

Scenario #2 is made up of four larger obstacles placed between the starting and the target points, in order to spread the swarm before a narrow channel that forces the aircraft to re-group. After that, the remaining big obstacle forces the swarm to divide again before reaching the target. This kind of scenario is an interesting test to stress the collision avoidance algorithm and prove the swarm's capability to rendezvous after the obstacles. Any sub-swarm, due to the presence of the obstacles, is automatically re-linked to the overall DTG to re-estabilish the aircraft formation.

Figure 20 shows trajectories of the vehicles, highlighting the positions and the DTG at several time instants ($t = 0$ s, $t = 100$ s, $t = 180$ s, $t = 280$ s and $t = 350$ s).

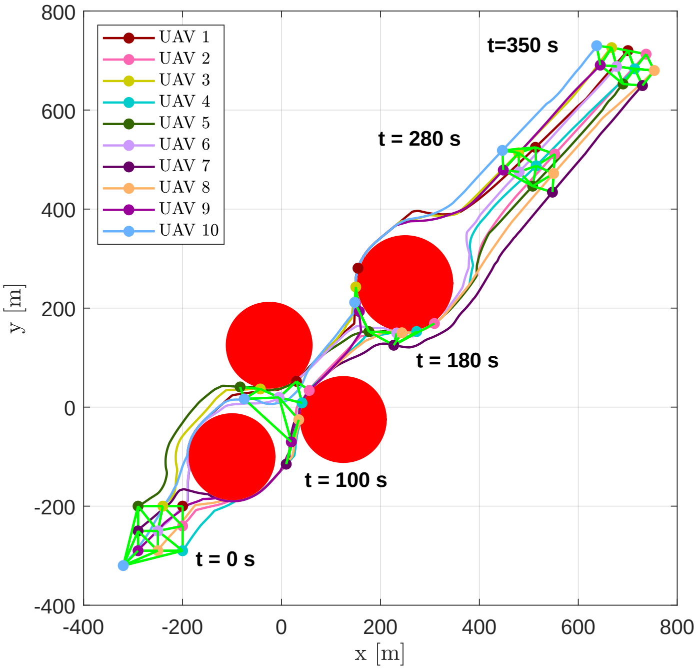


Figure 20. Scenario #2—10 aircraft: UAV trajectories. Red circles represent obstacles; colored markers indicate positions at different time instants; green straight lines highlight DTG arcs.

As expected, after the first obstacle, the swarm divided into four sub-swarms, composed of vehicles $\{1, 3, 5, 6, 10\}$, $\{7, 8, 9\}$, $\{2\}$ and $\{4\}$ (see aircraft positions at time instant $t = 49.9$ s in Figure 21). To pass the channel between the following obstacles the swarm had to re-organize its shape (see the snapshots at time instants $t = 89.9$ s and $t = 121.6$ s in Figure 21). After that, another split happened in order to overcome the last obstacle; the new sub-swarms, composed of vehicles $\{2, 4, 5, 6, 7, 8\}$ and $\{1, 3, 9, 10\}$, merged at the end of the obstacle avoidance. The result highlights the freedom of the swarm to compose different sub-swarms without assigning a prescribed structure, which is difficult to obtain using a leader (or virtual leader)–follower paradigm.

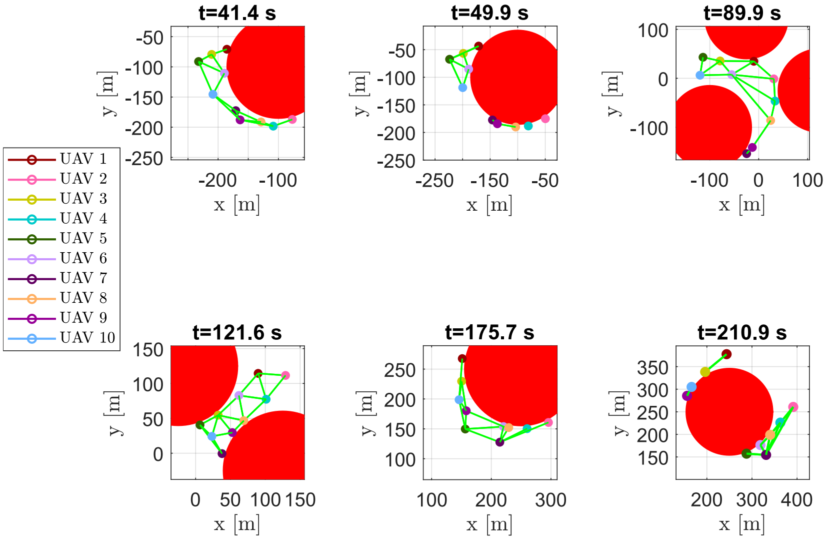


Figure 21. Scenario #2—10 aircraft: snapshots of UAVs' positions at different time instants. Colored markers indicate UAVs' actual positions; straight green lines show the DTG arcs. Chosen time instants highlight how the swarm is able to adapt its shape to the environment. At time instants $t = 49.9$ s and $t = 210.9$ s the DTG was not connected, and the swarm split into several sub-swarms.

Figures 22 and 23 show the aircraft distances from aircraft 1 and 7. A continuous change of swarm configuration can be noticed from the repeated switching between solid and dashed lines.

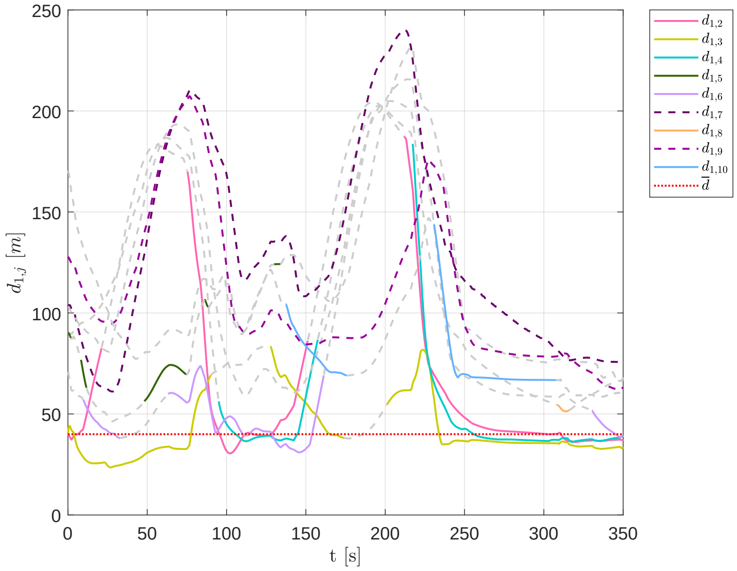


Figure 22. Scenario #2—10 aircraft: distances between UAV 1 and the other swarm vehicles. Solid lines are used for visible aircraft and dashed gray lines are employed whenever they are no longer visible, whereas dashed colored lines are used for never-visible UAVs. The desired distance is highlighted using a dotted red line.

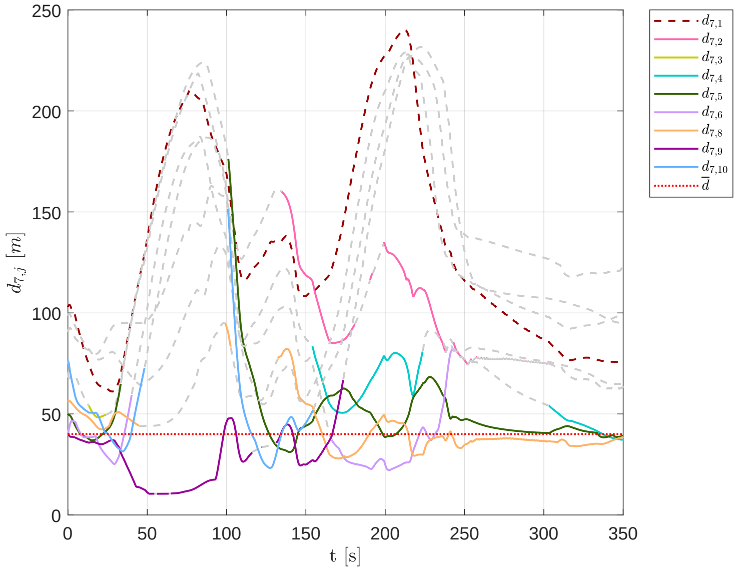


Figure 23. Scenario #2—10 aircraft: distances between UAV 7 and the other swarm vehicles. Solid lines are used for visible aircraftand dashed gray lines are employed whenever they are no longer visible, whereas dashed colored lines are used for never-visible UAVs. The desired distance is highlighted using a dotted red line.

It is worth noting that, after obstacles, the swarm tended to a configuration with mutual distances between visible aircraft that was equal to the desired one, e.g., aircraft 1 created equilateral triangles with aircraft 2, 3 and 4. This trend is confirmed by Figure 24, which shows the average distance between neighboring visible aircraft.

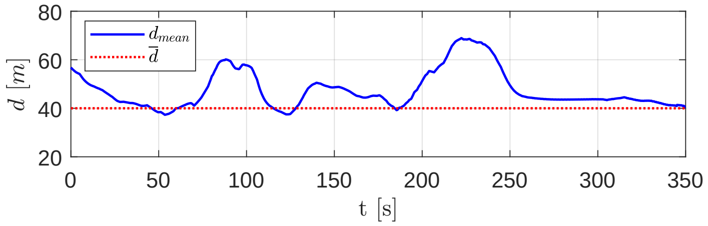


Figure 24. Scenario #2—10 aircraft: mean distance between visible aircraft (blue solid line) compared with desired distance (red dotted line).

6.5. Scenario #3—10 Aircraft

The third scenario allows to prove the capability of the swarm to rapidly adapt its shape to the environment. The first group of obstacles consists of two large circles, creating a narrow corridor before the third obstacle, which is placed to force the swarm to spread. The second group is composed of a full area of smaller obstacles forcing the swarm to split into several sub-swarms.

Figure 25 shows trajectories of the vehicles, highlighting their positions and the DTG arcs at several time instants ($t = 0$ s, $t = 80$ s, $t = 170$ s, and $t = 280$ s).

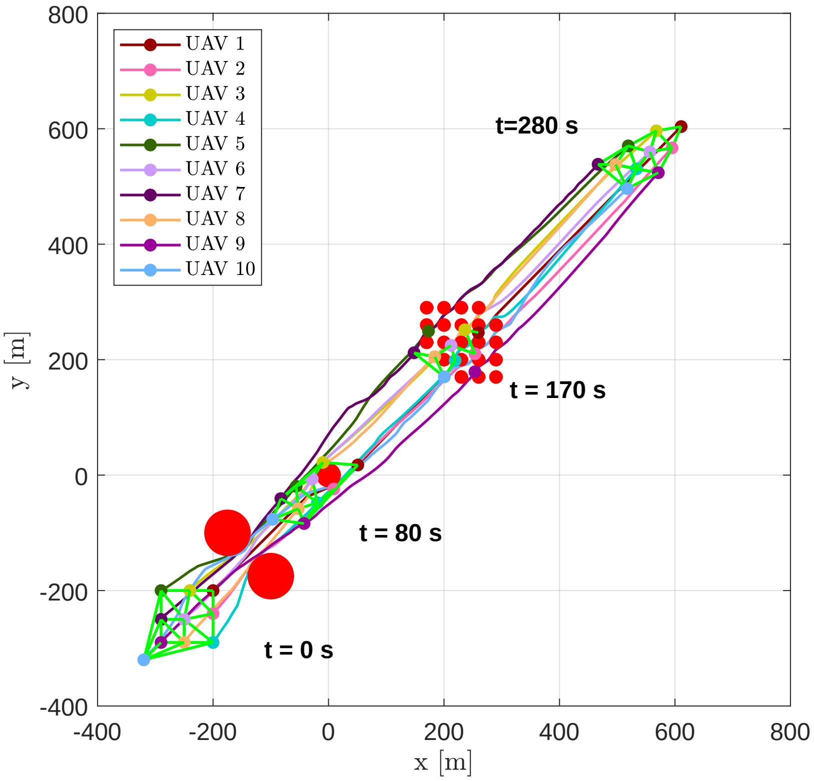


Figure 25. Scenario #3—10 aircraft: UAV trajectories. Red circles represent obstacles; colored markers indicate the UAVs positions at different time instants; green straight lines highlight DTG arcs.

At the beginning of the simulation, the swarm was forced to compact itself to pass between two big obstacles simulating the presence of a narrow crossing channel; then, the reduced size of the third obstacle permited the swarm to recompose a compact shape and pass it, losing the DTG connectivity at some time instants, as reported in Figure 26 at $t = 104.8$ s. Once aircraft reached the area full of obstacles, they scattered in order to overcome the obstacles, creating several time-varying sub-swarms as highlighted in Figure 26 at $t \in [160.8, 196.4]$ s.

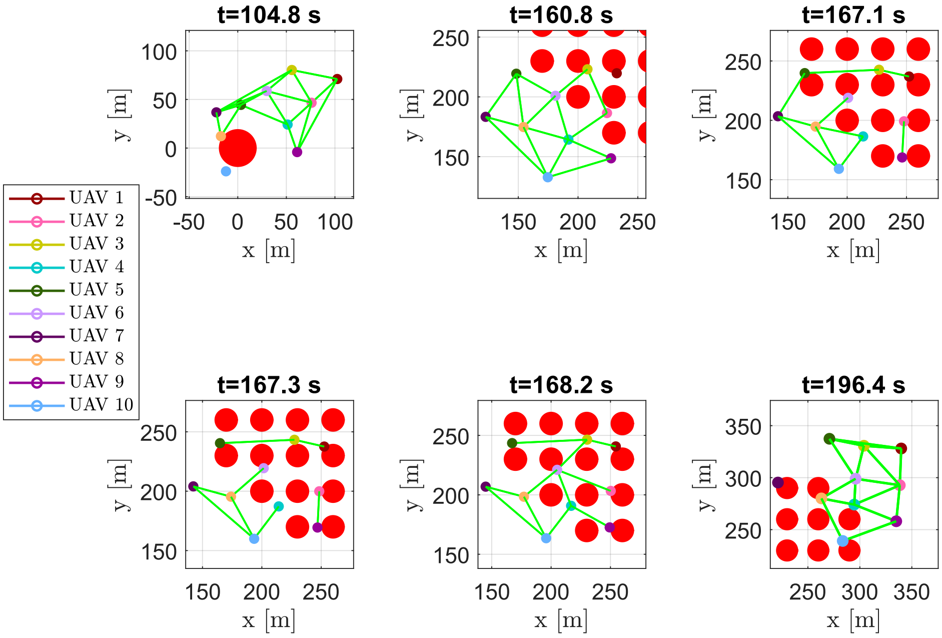


Figure 26. Scenario #3—10 aircraft: snapshots of UAV positions at different time instants. Colored markers indicate UAVs' actual positions, straight green lines show the DTG arcs. Snapshots highlight how the DTG connectivity rapidly changes in presence of obstacles.

This behavior is also confirmed by Figures 27 and 28, which show the distances of each vehicle visible from aircraft 1 and 6, respectively. As shown in Figure 27, in the time interval $t \in [160, 180]s$, the set of visible aircraft $\overline{\mathcal{Q}}_1$ was often empty, leaving the vehicle to fly and overcome obstacles without any swarm rule.

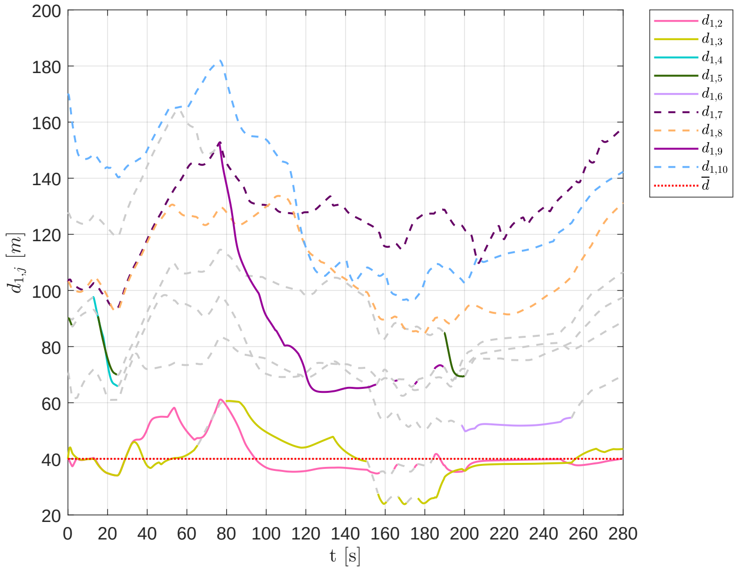


Figure 27. Scenario #3—10 aircraft: distances between UAV 1 and the other swarm vehicles. Solid lines are used for visible aircraft and dashed gray lines are employed whenever they are no longer visible, whereas dashed colored lines are used for never visible UAVs. The desired distance is highlighted using a dotted red line.

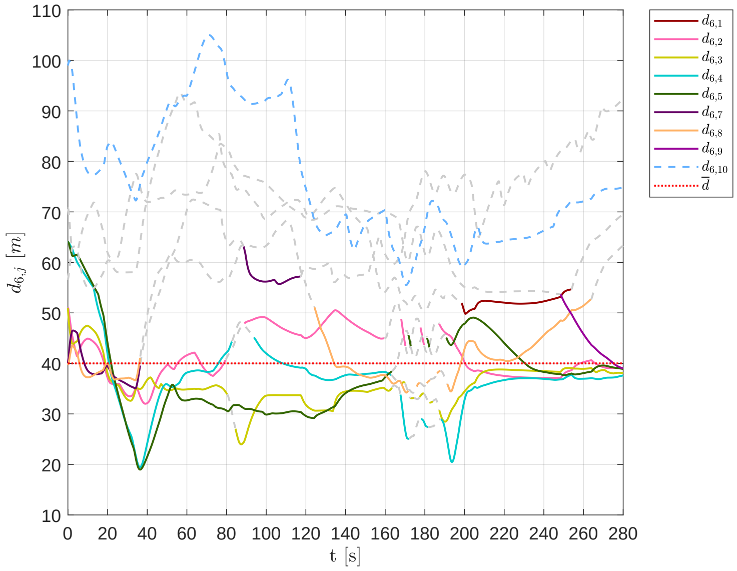


Figure 28. Scenario #3—10 aircraft: distances between UAV 6 and the other swarm vehicles. Solid lines are used for visible aircraft and dashed gray lines are employed whenever they are no longer visible, whereas dashed colored lines are used for never visible UAVs. The desired distance is highlighted using a dotted red line.

Finally, the UAVs re-estabilished the formation in absence of further obstacles. Figure 29 shows the average distance between visible aircraft. There was a clear tendency of it to become equal to the desired distance $\overline{d}$.

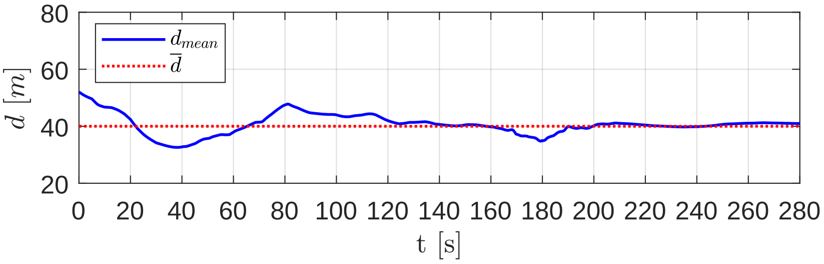


Figure 29. Scenario #3—10 aircraft: mean distance between visible aircraft (blue solid line) compared with desired distance (red dotted line).

During the validation test campaign, the proposed algorithm showed its properties as expected and discussed in the introduction. In the absence of obstacles, aircraft were able to fly together without creating a particular shape of the formation but maintaining mutual distances equal to the desired one. In presence of obstacles, such behavior resulted in a compact formation that split only near the obstacles to avoid collisions.

The absence of a fixed structure and a fixed leader or virtual leader makes the procedure really decentralized, lowering typical implementation and management difficulties [37,91]. During the flight, vehicles can freely change the topology of the swarm, creating sub-swarms as needed and re-composing the overall formation when possible.

It is worth noting that the use of the Delaunay triangulation makes the swarm cooperatively solve a particular packing problem [92] without a fixed external boundary. The objective of this packing problem is to find the shape of the swarm that can be exactly divided in equilateral triangles and whose vertices represent the vehicles positions. The result tended to a hexagonal packing that is the most efficient in terms of occupied area [93].

The swarm packing was tested in the absence of obstacles in order to show the packing density, which is defined as follows:

$$density = \frac{N\pi\bar{d}^2}{4A_c} \tag{31}$$

where A_c is the area of the region containing all the circles with a diameter equal to the desired distance. Figure 30 shows that the packing density was almost constant also in the presence of a swarm with a large number of vehicles and tended to be approximately equal to 1. Although in a packing problem the hexagonal packing density represents a maximum, here the circles can be overlapped, i.e., vehicles may have mutual distances shorter than the desired one. For this reason, the density reached values around 1.

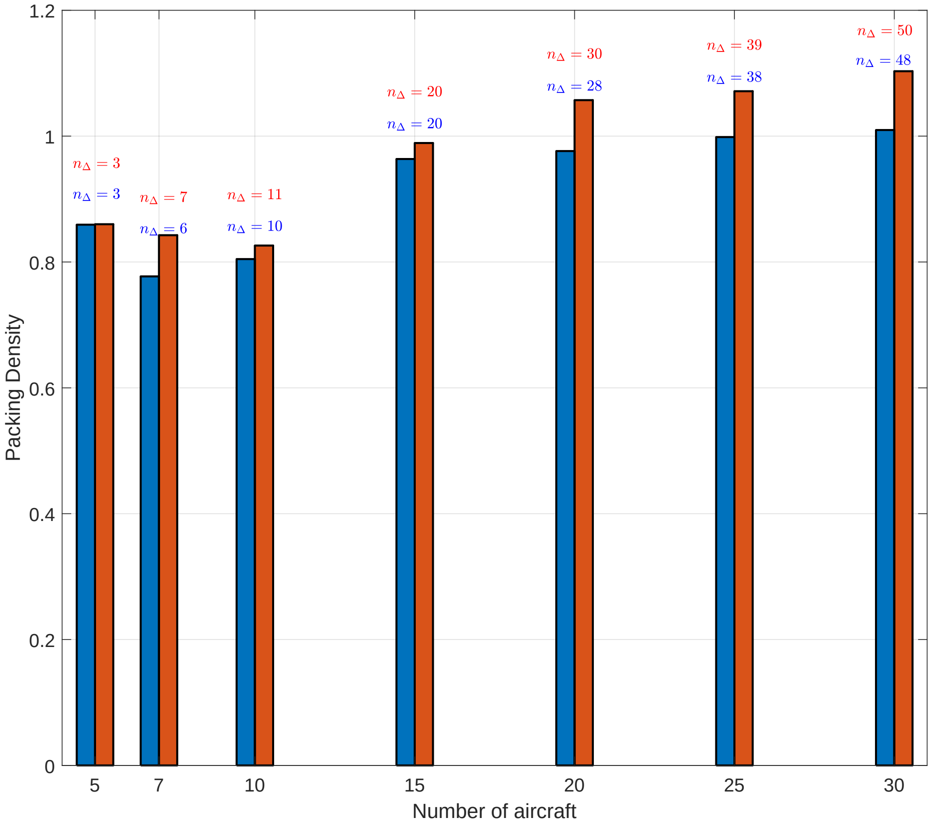


Figure 30. Minimum (blue bars) and maximum swarm packing densities (red bars) in the absence of obstacles. n_Δ describes the number of triangles in a DTG. The values close to 1 highlight the efficiency of the Delaunay-based swarm guidance strategy.

Another important feature is that the proposed approach involves a reasonably low computational burden. As shown in Figure 31, the MPC, which is the most demanding part of the algorithm from a computational point of view, never required more than 100 ms on an Intel i7-based laptop for the proposed numerical simulations. This result suggests the possible implementation of the algorithm even on slower CPU architectures, lowering the guidance algorithm sampling frequency.

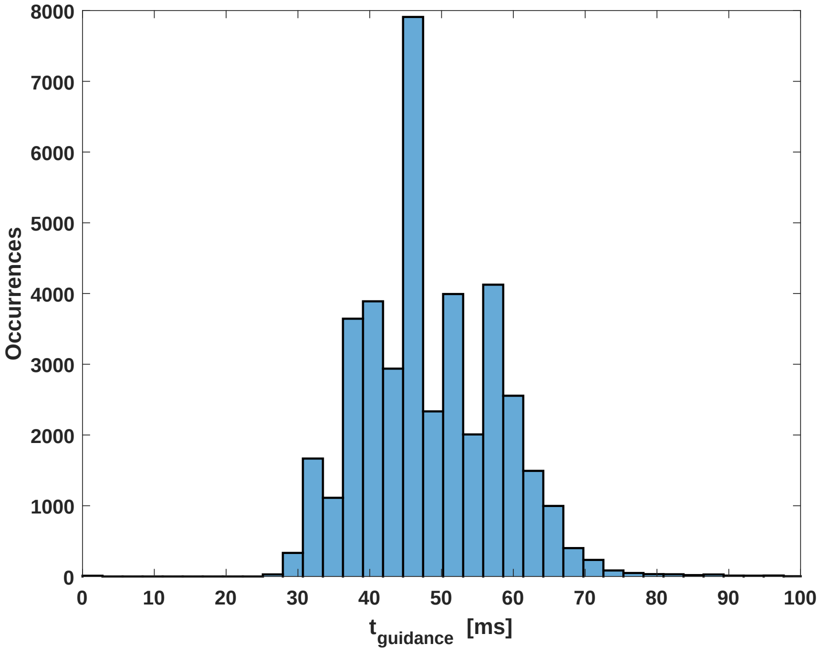


Figure 31. Occurrences of the MPC computational time for the proposed simulations.

7. Conclusions

In this paper, a decentralized guidance system for swarms of aircraft is proposed. It is based on both a Delaunay triangulation topology, to realize a regular mesh of the formation, and a potential field technique to compute the desired position for each aircraft. The tracking of the desired positions and collision avoidance is then based on a receding horizon model predictive control to optimize the overall performance of the swarm. Vehicles need only to exchange information about position, speed and heading, whereas their on-board control algorithms run independently. Since the fleet formation is not pre-assigned, in the presence of obstacles the swarm can split into several independent sub-swarms to avoid collisions. However, far from the obstacles, the proposed approach allows the vehicles to re-join and re-establish the flight formation. The vehicles tend to self-organize the swarm in order to assume a regular triangle-based formation.

Numerical results, involving different scenarios and fleet configurations, showed the effectiveness of the guidance and control system and underlined several features of the proposed technique.

In particular, the algorithm makes the UAVs formation easily adaptable to different operational environments by changing the swarm configuration and, consequently, the DTG connectivity over time.

This feature is difficult to obtain using an assigned structure or virtual structure as in a leader–follower paradigm. Furthermore, using our approach, the final configuration represents the solution to an optimization problem similar to the packing problem. The use of triangles and, in particular, Delaunay triangulation, gives the swarm the ability to increase the efficiency in terms of packing density. Finally, in terms of future applications, the relatively low computational burden would allow an implementation of the proposed approach also on an embedded low-cost control board.

Author Contributions: Conceptualization, methodology, investigation, formal analysis, writing, S.R.B., E.D., I.N., L.B. and M.M. Data curation, visualization, software development, S.R.B., E.D. and I.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the REGIONE CAMPANIA, SCAVIR Project "Advanced Configurations Studies for an Innovative Regional Aircraft" CUP B43D18000210007.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Raymer, D. *Aircraft Design: A Conceptual Approach*; American Institute of Aeronautics and Astronautics, Inc.: Reston, VA, USA, 2012.
2. Roskam, J. *Airplane Design*; DARcorporation: Lawrence, KS, USA, 1985.
3. Gudmundsson, S. *General Aviation Aircraft Design: Applied Methods and Procedures*; Butterworth-Heinemann: Oxford, UK, 2013.
4. Valerdi, R. Cost metrics for unmanned aerial vehicles. In Proceedings of the AIAA 16th Lighter-Than-Air Systems Technology Conference and Balloon Systems Conference, Arlington, VA, USA, 26–28 September 2005; p. 7102.
5. Maza, I.; Ollero, A.; Casado, E.; Scarlatti, D. Classification of Multi-UAV Architectures. In *Handbook of Unmanned Aerial Vehicles*; Valavanis, K.P., Vachtsevanos, G.J., Eds.; Springer: Dordrecht, The Netherlands, 2015; pp. 953–975. [CrossRef]
6. Kushleyev, A.; Mellinger, D.; Powers, C.; Kumar, V. Towards a swarm of agile micro quadrotors. *Auton. Robot.* **2013**, *35*, 287–300. [CrossRef]
7. Dudek, G.; Jenkin, M.; Milios, E.; Wilkes, D. A taxonomy for multi-agent robotics. *Auton. Robot.* **1996**, *3*. [CrossRef]
8. Cao, Y.U.; Fukunaga, A.S.; Kahng, A. Cooperative mobile robotics: Antecedents and directions. *Auton. Robot.* **1997**, *4*, 7–27. [CrossRef]
9. Schneider-Fontan, M.; Mataric, M. Territorial multi-robot task division. *IEEE Trans. Robot. Autom.* **1998**, *14*, 815–822. [CrossRef]
10. Yang, J.; Thomas, A.G.; Singh, S.; Baldi, S.; Wang, X. A semi-physical platform for guidance and formations of fixed-wing unmanned aerial vehicles. *Sensors* **2020**, *20*, 1136. [CrossRef]
11. Guo, X.; Lu, J.; Alsaedi, A.; Alsaadi, F.E. Bipartite consensus for multi-agent systems with antagonistic interactions and communication delays. *Phys. A Stat. Mech. Its Appl.* **2018**, *495*, 488–497. [CrossRef]
12. Lee, W.; Kim, D. Autonomous shepherding behaviors of multiple target steering robots. *Sensors* **2017**, *17*, 2729.
13. Forestiero, A. Bio-inspired algorithm for outliers detection. *Multimed. Tools Appl.* **2017**, *76*, 25659–25677. [CrossRef]
14. Wang, J.; Ahn, I.S.; Lu, Y.; Yang, T.; Staskevich, G. A distributed estimation algorithm for collective behaviors in multiagent systems with applications to unicycle agents. *Int. J. Control. Autom. Syst.* **2017**, *15*, 2829–2839. [CrossRef]
15. Olfati-Saber, R. Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Autom. Control* **2006**, *51*, 401–420. [CrossRef]
16. Su, H.; Wang, X.; Lin, Z. Flocking of multi-agents with a virtual leader. *IEEE Trans. Autom. Control* **2009**, *54*, 293–307. [CrossRef]
17. Sun, F.; Turkoglu, K. Distributed real-time non-linear receding horizon control methodology for multi-agent consensus problems. *Aerosp. Sci. Technol.* **2017**, *63*, 82–90. [CrossRef]
18. Vásárhelyi, G.; Virágh, C.; Somorjai, G.; Tarcai, N.; Szörényi, T.; Nepusz, T.; Vicsek, T. Outdoor flocking and formation flight with autonomous aerial robots. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 3866–3873.
19. Bennet, D.J.; MacInnes, C.; Suzuki, M.; Uchiyama, K. Autonomous three-dimensional formation flight for a swarm of unmanned aerial vehicles. *J. Guid. Control. Dyn.* **2011**, *34*, 1899–1908. [CrossRef]
20. Varela, G.; Caamaño, P.; Orjales, F.; Deibe, A.; Lopez-Pena, F.; Duro, R.J. Swarm intelligence based approach for real time UAV team coordination in search operations. In Proceedings of the IEEE 2011 Third World Congress on Nature and Biologically Inspired Computing, Salamanca, Spain, 19–21 October 2011; pp. 365–370.
21. Alfeo, A.L.; Cimino, M.G.; De Francesco, N.; Lazzeri, A.; Lega, M.; Vaglini, G. Swarm coordination of mini-UAVs for target search using imperfect sensors. *Intell. Decis. Technol.* **2018**, *12*, 149–162. [CrossRef]
22. Saska, M.; Vonásek, V.; Chudoba, J.; Thomas, J.; Loianno, G.; Kumar, V. Swarm distribution and deployment for cooperative surveillance by micro-aerial vehicles. *J. Intell. Robot. Syst.* **2016**, *84*, 469–492. [CrossRef]
23. Acevedo, J.J.; Arrue, B.C.; Maza, I.; Ollero, A. Cooperative large area surveillance with a team of aerial mobile robots for long endurance missions. *J. Intell. Robot. Syst.* **2013**, *70*, 329–345. [CrossRef]

24. Renzaglia, A.; Doitsidis, L.; Chatzichristofis, S.A.; Martinelli, A.; Kosmatopoulos, E.B. Distributed Multi-Robot Coverage using Micro Aerial Vehicles. In Proceedings of the 21st Mediterranean Conference on Control and Automation, Chania, Greece, 25–28 June 2013; pp. 963–968.
25. Garcia-Aunon, P.; del Cerro, J.; Barrientos, A. Behavior-Based Control for an Aerial Robotic Swarm in Surveillance Missions. *Sensors* **2019**, *19*, 4584. [CrossRef]
26. Kosuge, K.; Sato, M. Transportation of a single object by multiple decentralized-controlled nonholonomic mobile robots. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'99), Kyongju, Korea, 17–21 October 1999; Volume 3, pp. 1681–1686. [CrossRef]
27. Tartaglione, G.; D'Amato, E.; Ariola, M.; Rossi, P.S.; Johansen, T.A. Model predictive control for a multi-body slung-load system. *Robot. Auton. Syst.* **2017**, *92*, 1–11. [CrossRef]
28. Bernard, M.; Kondak, K. Generic slung load transportation system using small size helicopters. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, 12–17 May 2009; pp. 3258–3264. [CrossRef]
29. Michael, N.; Fink, J.; Kumar, V. Cooperative manipulation and transportation with aerial robots. *Auton. Robot.* **2011**, *30*, 73–86. [CrossRef]
30. Palunko, I.; Cruz, P.; Fierro, R. Agile Load Transportation: Safe and Efficient Load Manipulation with Aerial Robots. *IEEE Robot. Autom. Mag.* **2012**, *19*, 69–79. [CrossRef]
31. Bangash, Z.A.; Sanchez, R.P.; Ahmed, A.; Khan, M.J. Aerodynamics of Formation Flight. *J. Aircr.* **2006**, *43*, 907–912. [CrossRef]
32. Ariola, M.; Mattei, M.; D'Amato, E.; Notaro, I.; Tartaglione, G. Model predictive control for a swarm of fixed wing uavs. In Proceedings of the 30th Congress of the International Council of the Aeronautical Sciences (ICAS), Daejeon, Korea, 25–30 September 2016.
33. Yun, B.; Chen, B.M.; Lum, K.Y.; Lee, T.H. Design and implementation of a leader-follower cooperative control system for unmanned helicopters. *J. Control Theory Appl.* **2010**, *8*, 61–68. [CrossRef]
34. Bayraktar, S.; Fainekos, G.; Pappas, G. Experimental cooperative control of fixed-wing unmanned aerial vehicles. In Proceedings of the 2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601), Nassau, Bahamas, 14–17 December 2004; Volume 4, pp. 4292–4298. [CrossRef]
35. Paul, T.; Krogstad, T.R.; Gravdahl, J.T. Modelling of UAV formation flight using 3D potential field. *Simul. Model. Pract. Theory* **2008**, *16*, 1453–1462. [CrossRef]
36. Dasgupta, P. A Multiagent Swarming System for Distributed Automatic Target Recognition Using Unmanned Aerial Vehicles. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2008**, *38*, 549–563. [CrossRef]
37. D'Amato, E.; Mattei, M.; Notaro, I. Bi-level flight path planning of UAV formations with collision avoidance. *J. Intell. Robot. Syst.* **2019**, *93*, 193–211. [CrossRef]
38. Oh, K.K.; Park, M.C.; Ahn, H.S. A survey of multi-agent formation control. *Automatica* **2015**, *53*, 424–440. [CrossRef]
39. Reynolds, C.W. Flocks, herds and schools: A distributed behavioral model. In Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, Anaheim, CA, USA, 27–31 July 1987; pp. 25–34.
40. Bellingham, J.S.; Tillerson, M.; Alighanbari, M.; How, J.P. Cooperative path planning for multiple UAVs in dynamic and uncertain environments. In Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, USA, 10–13 December 2002; Volume 3, pp. 2816–2822.
41. Garrido, S.; Moreno, L.; Lima, P.U. Robot formation motion planning using fast marching. *Robot. Auton. Syst.* **2011**, *59*, 675–683. [CrossRef]
42. Gómez, J.V.; Lumbier, A.; Garrido, S.; Moreno, L. Planning robot formations with fast marching square including uncertainty conditions. *Robot. Auton. Syst.* **2013**, *61*, 137–152. [CrossRef]
43. Yang, T.T.; Liu, Z.Y.; Chen, H.; Pei, R. Formation control of mobile robots: State and open problems. *Zhineng Xitong Xuebao (CAAI Trans. Intell. Syst.)* **2007**, *2*, 21–27.
44. Peng, Z.; Wen, G.; Rahmani, A.; Yu, Y. Leader–follower formation control of nonholonomic mobile robots based on a bioinspired neurodynamic based approach. *Robot. Auton. Syst.* **2013**, *61*, 988–996. [CrossRef]
45. Consolini, L.; Morbidi, F.; Prattichizzo, D.; Tosques, M. Leader–follower formation control of nonholonomic mobile robots with input constraints. *Automatica* **2008**, *44*, 1343–1349. [CrossRef]

46. Tan, K.H.; Lewis, M.A. Virtual structures for high-precision cooperative mobile robotic control. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'96), Osaka, Japan, 8 November 1996; Volume 1, pp. 132–139.
47. Lewis, M.A.; Tan, K.H. High precision formation control of mobile robots using virtual structures. *Auton. Robot.* **1997**, *4*, 387–403. [CrossRef]
48. Balch, T.; Arkin, R.C. Behavior-based formation control for multirobot teams. *IEEE Trans. Robot. Autom.* **1998**, *14*, 926–939. [CrossRef]
49. Balch, T.; Hybinette, M. Behavior-based coordination of large-scale robot formations. In Proceedings of the Fourth International Conference on MultiAgent Systems, Boston, MA, USA, 10–12 July 2000; pp. 363–364.
50. Balch, T.; Hybinette, M. Social potentials for scalable multi-robot formations. In Proceedings of the Millennium Conference, IEEE International Conference on Robotics and Automation, Symposia Proceedings (Cat. No. 00CH37065) (2000 ICRA), San Francisco, CA, USA, 24–28 April 2000; Volume 1, pp. 73–80.
51. Marino, A.; Parker, L.E.; Antonelli, G.; Caccavale, F. A decentralized architecture for multi-robot systems based on the null-space-behavioral control with application to multi-robot border patrolling. *J. Intell. Robot. Syst.* **2013**, *71*, 423–444. [CrossRef]
52. Baizid, K.; Giglio, G.; Pierri, F.; Trujillo, M.A.; Antonelli, G.; Caccavale, F.; Viguria, A.; Chiaverini, S.; Ollero, A. Behavioral control of unmanned aerial vehicle manipulator systems. *Auton. Robot.* **2017**, *41*, 1203–1220. [CrossRef]
53. Barve, A.; Nene, M.J. Survey of Flocking Algorithms in multi-agent Systems. *Int. J. Comput. Sci. Issues (IJCSI)* **2013**, *10*, 110.
54. Chung, S.J.; Paranjape, A.A.; Dames, P.; Shen, S.; Kumar, V. A survey on aerial swarm robotics. *IEEE Trans. Robot.* **2018**, *34*, 837–855. [CrossRef]
55. Vásárhelyi, G.; Virágh, C.; Somorjai, G.; Nepusz, T.; Eiben, A.E.; Vicsek, T. Optimized flocking of autonomous drones in confined environments. *Sci. Robot.* **2018**, *3*, eaat3536. [CrossRef]
56. Cordeiro, T.F.; Ferreira, H.C.; Ishihara, J.Y. Robust and Synchronous Nonlinear Controller for Autonomous Formation Flight of Fixed Wing UASs. In Proceedings of the 2019 International Conference on Unmanned Aircraft Systems (ICUAS), Atlanta, GA, USA, 11–14 June 2019; pp. 378–387.
57. Singh, S.N.; Zhang, R.; Chandler, P.; Banda, S. Decentralized nonlinear robust control of UAVs in close formation. *Int. J. Robust Nonlinear Control. IFAC-Affil. J.* **2003**, *13*, 1057–1078. [CrossRef]
58. Campa, G.; Gu, Y.; Seanor, B.; Napolitano, M.R.; Pollini, L.; Fravolini, M.L. Design and flight-testing of non-linear formation control laws. *Control Eng. Pract.* **2007**, *15*, 1077–1092. [CrossRef]
59. Cordeiro, T.F.; Ferreira, H.C.; Ishihara, J.Y. Non linear controller and path planner algorithm for an autonomous variable shape formation flight. In Proceedings of the IEEE 2017 International Conference on Unmanned Aircraft Systems (ICUAS), Miami, FL, USA, 13–16 June 2017; pp. 1493–1502.
60. Rezaee, H.; Abdollahi, F.; Talebi, H.A. $\mathcal{H}_\infty$ Based Motion Synchronization in Formation Flight With Delayed Communications. *IEEE Trans. Ind. Electron.* **2014**, *61*, 6175–6182. [CrossRef]
61. Han, K.; Lee, J.; Kim, Y. Unmanned aerial vehicle swarm control using potential functions and sliding mode control. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2008**, *222*, 721–730. [CrossRef]
62. Quintero, S.A.; Collins, G.E.; Hespanha, J.P. Flocking with fixed-wing UAVs for distributed sensing: A stochastic optimal control approach. In Proceedings of the IEEE 2013 American Control Conference, Washington, DC, USA, 17–19 June 2013; pp. 2025–2031.
63. Zhihao, C.; Longhong, W.; Jiang, Z.; Kun, W.; Yingxun, W. Virtual target guidance-based distributed model predictive control for formation control of multiple UAVs. *Chin. J. Aeronaut.* **2020**, *33*, 1037–1056.
64. Xi, W.; Baras, J.S. MPC based motion control of car-like vehicle swarms. In Proceedings of the IEEE 2007 Mediterranean Conference on Control & Automation, Athens, Greece, 27–29 June 2007; pp. 1–6.
65. Peng, Z.; Li, B.; Chen, X.; Wu, J. Online route planning for UAV based on model predictive control and particle swarm optimization algorithm. In Proceedings of the IEEE 10th World Congress on Intelligent Control and Automation, Beijing, China, 6–8 July 2012; pp. 397–401.
66. Zhang, D.; Xie, G.; Yu, J.; Wang, L. Adaptive task assignment for multiple mobile robots via swarm intelligence approach. *Robot. Auton. Syst.* **2007**, *55*, 572–588. [CrossRef]
67. Altshuler, Y.; Yanovsky, V.; Wagner, I.A.; Bruckstein, A.M. Efficient cooperative search of smart targets using uav swarms. *Robotica* **2008**, *26*, 551. [CrossRef]

68. Najm, A.A.; Ibraheem, I.K.; Azar, A.T.; Humaidi, A.J. Genetic Optimization-Based Consensus Control of Multi-Agent 6-DoF UAV System. *Sensors* **2020**, *20*, 3576. [CrossRef]
69. Duan, H.; Luo, Q.; Shi, Y.; Ma, G. Hybrid particle swarm optimization and genetic algorithm for multi-UAV formation reconfiguration. *IEEE Comput. Intell. Mag.* **2013**, *8*, 16–27. [CrossRef]
70. Bai, C.; Duan, H.; Li, C.; Zhang, Y. Dynamic multi-UAVs formation reconfiguration based on hybrid diversity-PSO and time optimal control. In Proceedings of the 2009 IEEE Intelligent Vehicles Symposium, Xi'an, China, 3–5 June 2009; pp. 775–779.
71. Li, X.; Chen, H. An interactive control algorithm used for equilateral triangle formation with robotic sensors. *Sensors* **2014**, *14*, 7229–7247. [CrossRef]
72. Li, X. A Triangular Formation Strategy for Collective Behaviors of Robot Swarm. In *Computational Science and Its Applications—ICCSA 2009*; Gervasi, O., Taniar, D., Murgante, B., Laganà, A., Mun, Y., Gavrilova, M.L., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5592, pp. 897–911. [CrossRef]
73. Elkaim, G.H.; Kelbley, R.J. A lightweight formation control methodology for a swarm of non-holonomic vehicles. In Proceedings of the 2006 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2006.
74. Defoort, M.; Polyakov, A.; Demesure, G.; Djemai, M.; Veluvolu, K. Leader-follower fixed-time consensus for multi-agent systems with unknown non-linear inherent dynamics. *IET Control Theory Appl.* **2015**, *9*, 2165–2170. [CrossRef]
75. Ying, Z.; Xu, L. Leader-follower formation control and obstacle avoidance of multi-robot based on artificial potential field. In Proceedings of the IEEE 27th Chinese Control and Decision Conference (2015 CCDC), Qingdao, China, 23–25 May 2015; pp. 4355–4360.
76. Barnes, L.; Fields, M.; Valavanis, K. Unmanned ground vehicle swarm formation control using potential fields. In Proceedings of the 2007 Mediterranean Conference on Control Automation, Athens, Greece, 27–29 June 2007; pp. 1–8. [CrossRef]
77. Beaulieu, A.; Givigi, S.N.; Ouellet, D.; Turner, J.T. Model-Driven Development Architectures to Solve Complex Autonomous Robotics Problems. *IEEE Syst. J.* **2018**, *12*, 1404–1413. [CrossRef]
78. D'Amato, E.; Mattei, M.; Notaro, I. Distributed Reactive Model Predictive Control for Collision Avoidance of Unmanned Aerial Vehicles in Civil Airspace. *J. Intell. Robot. Syst.* **2020**, *97*, 185–203. [CrossRef]
79. Sloan, S. A fast algorithm for constructing Delaunay triangulations in the plane. *Adv. Eng. Softw. (1978)* **1987**, *9*, 34–55. [CrossRef]
80. Kownacki, C.; Ambroziak, L. Local and asymmetrical potential field approach to leader tracking problem in rigid formations of fixed-wing UAVs. *Aerosp. Sci. Technol.* **2017**, *68*, 465–474. [CrossRef]
81. Sun, J.; Tang, J.; Lao, S. Collision avoidance for cooperative UAVs with optimized artificial potential field algorithm. *IEEE Access* **2017**, *5*, 18382–18390. [CrossRef]
82. Rawlings, J.B.; Mayne, D.Q. Model predictive control: Theory and design. *Nob Hill Pub.* **2009**.
83. Griva, I.; Nash, S.G.; Sofer, A. *Linear and Nonlinear Optimization*; Siam: Philadelphia, PA, USA, 2009; Volume 108.
84. Lazimy, R. Mixed-integer quadratic programming. *Math. Program.* **1982**, *22*, 332–349. [CrossRef]
85. Takapoui, R.; Moehle, N.; Boyd, S.; Bemporad, A. A simple effective heuristic for embedded mixed-integer quadratic programming. *Int. J. Control* **2020**, *93*, 2–12. [CrossRef]
86. D'Amato, E.; Notaro, I.; Mattei, M. Reactive Collision Avoidance using Essential Visibility Graphs. In Proceedings of the 2019 6th International Conference on Control, Decision and Information Technologies (CoDIT), Paris, France, 23–26 April 2019; pp. 522–527.
87. Huang, H.; Savkin, A.; Li, X. Reactive Autonomous Navigation of UAVs for Dynamic Sensing Coverage of Mobile Ground Targets. *Sensors* **2020**, *20*, 3720. [CrossRef]
88. Moschetta, J.M. The aerodynamics of micro air vehicles: Technical challenges and scientific issues. *Int. J. Eng. Syst. Model. Simul. 48* **2014**, *6*, 134–148. [CrossRef]
89. Stellato, B.; Banjac, G.; Goulart, P.; Bemporad, A.; Boyd, S. OSQP: An Operator Splitting Solver for Quadratic Programs. *Math. Program. Comput.* **2020**. [CrossRef]
90. Stellato, B.; Naik, V.V.; Bemporad, A.; Goulart, P.; Boyd, S. Embedded mixed-integer quadratic optimization using the OSQP solver. In Proceedings of the European Control Conference (ECC), Limassol, Cyprus, 12–15 June 2018. [CrossRef]

91. Perez-Montenegro, C.; Scanavino, M.; Bloise, N.; Capello, E.; Guglieri, G.; Rizzo, A. A mission coordinator approach for a fleet of uavs in urban scenarios. *Transp. Res. Procedia* **2018**, *35*, 110–119. [CrossRef]
92. Pach, J.; Agarwal, P.K. *Combinatorial Geometry*; John Wiley & Sons: Hoboken, NJ, USA, 2011; Volume 37.
93. Chang, H.C.; Wang, L.C. A simple proof of Thue's Theorem on circle packing. *arXiv* **2010**, arXiv:1009.4322.