



Computational Intelligence in Electrical Engineering
Vol. 16, No. 1, 2025
pp. 19-44
Research Paper

Fast Trajectory Replanning of Racing UAVs in Restricted Space via Separating Hyperplanes and Disturbance Effects Estimation

Nour Abbas¹, Hossein Karimpour^{*2}

¹ PhD Candidate, Mechanical Engineering Department, Faculty of Engineering, University of Isfahan, Isfahan, Iran
no.abbas@eng.ui.ac.ir

² Assistant Professor, Mechanical Engineering Department, Faculty of Engineering, University of Isfahan, Isfahan, Iran
h.karimpour@eng.ui.ac.ir

Abstract :

Maintaining safe and reliable trajectories for multiple Unmanned Aerial Vehicles (UAVs) in environments with obstacles and external disturbances, such as wind, remains a significant challenge. This paper presents a novel trajectory replanning approach for a group of UAVs in the presence of wind disturbances. The proposed method is based on reflecting the impact of any uncertainty caused by disturbances as an uncertainty in the dynamic response of the quadrotor, through the application of the forward reachable sets (FRSs) theory on the motion equations. This results in a location ellipsoid error for each quadrotor. Combining this error ellipsoid with the quadrotor's physical dimensions via a Minkowski sum yields an expanded safety ellipsoid. To avoid inter-UAV collisions, a separating hyperplane is constructed between pairs of safety ellipsoids at each replanning instant. Furthermore, the adaptation of the trajectory to instantaneous conditions is made possible through the application of a virtual force field at each point on the trajectory. This force field accounts for both the intensity of wind disturbance and the proximity of the trajectory to surrounding obstacles and other quadrotors (delimited by hyperplanes), allowing the trajectory to be reconfigured while preserving some key optimal characteristics. Simulation results obtained in Matlab demonstrate the effectiveness of the proposed trajectory replanning framework and provide a comparative evaluation of its reduced computational cost.

Keywords: Multi Unmanned Aerial Vehicles, Wind Disturbance, Trajectory Replanning, Safe Ellipsoid, Hyperplane, Forward Reachable Sets.

1. Introduction

In recent years, Unmanned Aerial Vehicles

(UAVs), also known as drones, have gained significant attention across various fields due to their high adaptability, maneuverability, and agility. They have been widely employed in applications such as emergency rescue [1], package delivery [2], aerial photogrammetry [3], search-and-rescue operations [4], and other important tasks that enhance human life, making

¹ Submission date: 06, 09, 2025

Acceptance date: 18, 02, 2026

Corresponding author: Hossein Karimpour, Mechanical Engineering Department, Faculty of Engineering, University of Isfahan, Isfahan, Iran



This is an open access article under the CC BY-NC-ND/4.0/ License (<https://creativecommons.org/licenses/by-nc-nd/4.0/>).



<https://doi.org/10.22108/isee.2026.146562.1763>

drones essential tools in modern society.

Despite the advantages of a single UAV, certain applications require the use of multiple UAVs, which enhance task efficiency. UAVs operating in groups can quickly cover large areas in disaster zones for search and rescue (SAR) operations [5], as well as in agriculture and natural environments for tasks such as crop monitoring, pest detection, and precision spraying [6]. A group of UAVs can also reduce time and risks associated with manual inspection of infrastructure, such as power lines, pipelines, or bridges [7]. Additionally, UAVs can collaborate to deliver packages, leading to faster delivery and reduced costs [8].

The trajectory planning process for a group of UAVs involves designing optimized paths to move from the starting point to the target, while avoiding collisions with obstacles and other UAVs involved in the same mission. This ensures that all UAVs reach their destinations efficiently and safely, considering the constraints imposed on each vehicle.

There are three basic approaches for multi-robot coordination: centralized, decentralized, and hybrid planning approaches [9]. Using these approaches, along with optimization techniques, enables UAVs to perform their assigned tasks safely and efficiently. For comparison, these particularities can be enumerated:

- In the centralized planning method, the controller unit processes the data sent from each UAV and then generates and sends the required control instructions to each UAV based on the programmed task [10, 11]. The advantage of this method lies in the centralized decision-making process, which enables good coordination between agents. However, in cases of excessive participants, there are difficulties in data processing, and the risk of system failure increases if the central controller fails to make a decision.

- Unlike the centralized planning method, each UAV in decentralized planning relies on its own sensors and data communicated with other UAVs within the group to make and execute independent, environment-aware decisions [10, 12, 13]. In this case, the failure of one UAV does not necessarily lead to the failure of the entire system.

- In a hybrid approach, UAVs work together to

carry out specific tasks while maintaining a certain degree of autonomy, thus combining the advantages of the centralized approach for achieving optimal results and the independence of the decentralized approach for handling uncertainty [9].

Over the years, numerous algorithms have been proposed for UAV path planning to avoid obstacles and ensure safe and reliable operation in unknown environments. Path planning techniques can be broadly categorized into grid-based, sampling-based, artificial intelligence-based, cooperative, and non-cooperative techniques [14]. Grid-based methods include the A-star algorithm [15] and Dijkstra algorithm [16], while sampling-based methods include the Rapidly-exploring Random Tree (RRT) and its improved variants [17].

External disturbances that UAVs may be exposed to can affect their performance, safety, and reliability, and therefore should not be ignored but taken into account. These disturbances can arise from various sources, such as suspended loads [18], wind presence [19], and dynamic obstacles [20], making their impact on UAV operations a challenge that must be addressed promptly.

Concerning dynamic obstacles in the environment, the Curve-shortening flow method (CFM) was proposed in [21] to re-plan the trajectory; however, the UAV was modelled as a point mass, without considering any external wind disturbance. In the present paper, we propose a trajectory replanning approach for two individual UAVs flying through a restricted space under external wind disturbance, where the two UAVs are represented as “inflatable” ellipsoids to account for both collision and disturbances. The trajectories of both UAVs are replanned to avoid each other and potential collisions with obstacles by utilizing the concept of a separating hyperplane and the method proposed in [21].

The main contributions of this study can be summarized as follows:

1. A new approach is proposed for replanning the trajectories of a pair of UAVs racing into an obstructed space while accounting for external wind disturbances.
2. The generated trajectories demonstrate desirable characteristics, including trajectory length, high smoothness, minimal snap, and effective collision avoidance with obstacles.

3. A performance comparison with the Nonlinear Model Predictive Control (NMPC) technique highlights the proposed method's ability to maintain a strong trade-off between safety, efficiency, and smooth motion, while also achieving a significant reduction in computational cost.

The structure of the paper is as follows: Section 2 reviews the existing literature on multi-UAV trajectory planning in environments with obstacles and external disturbances. Section 3 presents the mathematical formulation of the proposed trajectory replanning approach. Section 4 provides the simulation results and a comparative analysis with the NMPC-based trajectory design method. Finally, Section 5 concludes the paper with a summary of the findings.

2. Related Work

2.1. Multi-UAV Trajectory Planning

Trajectory planning and path planning are two fundamental concepts in robotics and autonomous systems. While path planning algorithms, such as Dijkstra, A-star, Jump Point Search, and others, are designed to find a route from a starting point to a desired destination while avoiding obstacles, focusing primarily on the shape of the path without considering timing or motion parameters, trajectory planning is concerned with generating a smooth and safe path for the UAVs, taking into account not only the path but also the speed, acceleration, and precise timing of each UAV's position along the path.

There are three main methods for UAV trajectory planning: unconstrained, soft-constrained, and hard-constrained methods. In unconstrained methods, such as the method presented in [22], position, velocity, and acceleration are incorporated as constraints to generate the trajectory while considering the path points. In the soft-constrained method, for example, in [23], trajectory points are adjusted to maintain a safe distance from obstacles by exploiting the convex hull property of the B-spline curves, thereby ensuring the generation of a safe, collision-free trajectory. Hard-constrained methods, as discussed in [24, 25], typically involve two main steps: first, establishing a safe, obstacle-free flight path, and then designing the trajectory within the flight path.

As mentioned in the introduction, three main planning approaches are commonly used to address the multi-robot coordination problem: centralized, decentralized, and hybrid approaches.

In [11], a centralized multi-UAV planning method is proposed by integrating the deterministic Coordinated Multi-robot Exploration (CME) framework with a swarm-based optimization technique known as the Aquila Optimizer (AO). The proposed method is suitable for various mission scenarios, including reconnaissance, search, and rescue tasks.

A distributed trajectory generation algorithm for multiple UAVs operating in proximity is presented in [26]. In this work, UAVs are modeled using ellipsoids rather than spherical representations, as the latter may invalidate feasible trajectories in confined environments. The core idea is to partition the space into regions separated by hyperplanes using a Voronoi-based space partitioning approach. Employing Voronoi diagrams reduces the computational burden by utilizing a set of pre-defined, time-invariant hyperplanes. By solving the optimization problem, each UAV replans its trajectory using a receding horizon strategy. At each sampling time, each UAV generates its trajectory within a subset of its Voronoi cell based on the positional information of neighboring UAVs, continuing this process until all vehicles reach their target positions. For scenarios in which UAVs rely solely on local information, the method proposed in [27] can be employed. This approach computes the areas of Voronoi buffer cells using data obtained from a UAV-mounted laser range scanner to estimate obstacle positions, which are subsequently used as seeds for constructing the Voronoi buffer cells.

The author in [28] also employed the concepts of ellipsoids and hyperplanes to address the UAV motion planning problem in scenarios involving gap traversal and navigation through narrow spaces. The proposed method was shown to be effective for multiple UAVs as well. In this approach, the two UAVs are modeled as ellipsoids, and a separating hyperplane is generated between them to enforce collision avoidance constraints at each time step. The trajectory of each vehicle is represented as a Bezier curve, ensuring that these collision avoidance constraints are satisfied.

In a related context, [29] proposed an algorithm called Replanning using Linear Spatial Separations (RLSS), which employs linear spatial separations, representing the dynamic limits of UAVs, to ensure safety. This method relies solely on local relative measurements between UAVs rather than on inter-UAV communication. This algorithm computes hyperplanes between curve

segments and obstacles or other UAVs in the environment and subsequently uses them to define a convex polyhedral restricted space, within which trajectory segments are constrained.

To generate collision-free trajectories for a group of UAVs operating in environments with both static and dynamic obstacles, [30] proposed the Trajectory Planner in Multi-Agent and Dynamic Environments (MADER), a decentralized planning framework. At the end of each replanning iteration, every UAV broadcasts its trajectory to the others. To guarantee collision avoidance, a polyhedron is constructed around each trajectory at every time period, and separating hyperplane constraints are then incorporated into the optimization problem.

Communication delays between UAVs can lead to system failures and hinder mission completion. To address this issue, the researchers in [31] extended the work presented in [30] and proposed Robust MADER (RMADER), an asynchronous and decentralized multi-agent trajectory planner capable of generating collision-free trajectories in dynamic environments despite communication delays. RMADER employs a delay-checking mechanism that ensures at least one collision-free trajectory is maintained. However, it requires an upper bound on communication delay and does not account for complete communication outages.

While the UAVs in [26] are restricted to remain within their temporary Voronoi cell to avoid collisions, [32] addresses collision avoidance using a distributed model predictive control (DMPC) framework, in which an event-triggered replanning strategy is employed to generate real-time trajectories. In this approach, each UAV receives the most recent predicted trajectories of all other UAVs via a communication channel. Similarly, [33] proposed a novel algorithm for online distributed trajectory planning for multi-UAV, ensuring safe and dynamically feasible paths in crowded environments by using a linear safe corridor. The method begins with all UAVs sharing their current positions, goal points, and trajectories from the previous step; initial trajectories are then planned based on the previous replanning step (or from the current position if it is the first step). Collision avoidance constraints are subsequently constructed based on these initial trajectories. An intermediate target is then determined using the

grid-based path planner, where UAVs closer to the target are assigned higher priority. Finally, the safe trajectories are obtained by solving a distributed trajectory optimization problem formulated as a quadratic programming (QP) problem, incorporating both the safe flight corridor (SFC) and the linear safe corridor (LSC).

In [34], a new NMPC framework is introduced for multi-rotor aerial systems, specifically designed to address the high computational demands typically encountered in standard NMPC implementations. Rather than relying on stabilizing terminal costs or constraints, the method employs a growth-bound sequence derived from a customized running cost. This formulation enables the use of a shorter prediction horizon while still ensuring closed-loop stability. Numerical simulations demonstrate that the proposed approach achieves superior tracking accuracy, reduced computation time, and faster convergence compared to conventional NMPC techniques.

Based on the elastic bands method (EBM) [35], the Curve-shortening Flow Method (CFM) was proposed in [21] to refine and reshape UAV trajectories. In this approach, a UAV's path is represented as a sequence of interconnected nodes, each influenced by both attractive and repulsive forces. The attractive forces guide the path toward the locally shortest feasible trajectory, whereas the repulsive forces steer it away from obstacles. This dynamic interaction allows the trajectory to adapt to environmental changes, ensuring both optimality and safety.

2.2. UAVs in the Presence of External Disturbance

The concept of disturbance encompasses any internal or external factor that adversely affects a system's stability or performance. Analyzing disturbances has become a key research focus in UAV systems, as reducing or rejecting their effects is essential for enhancing both efficiency and safety in operational environments. Multi-UAV systems, in particular, are highly sensitive to various types of disturbances, including transmission delays, modeling uncertainties, and external influences such as wind or unforeseen obstacles.

There are two main strategies for addressing disturbances affecting UAVs: disturbance rejection approaches and trajectory replanning

methods.

Disturbance rejection techniques aim to actively compensate for disturbances while preserving the UAV's stability and performance. In contrast, replanning methods modify the UAV's trajectory immediately whenever the current plan becomes infeasible or unsafe due to environmental changes. Together, these strategies offer complementary mechanisms for enhancing the resilience of UAVs operating in uncertain and dynamic environments.

2.2.1. Disturbance Rejection Approaches

Several researchers have investigated various control strategies to reject disturbance, including model predictive control (MPC), active disturbance rejection control (ADRC), and adaptive control. For instance, in the observer-based H_∞ consensus control proposed in [36], a local control law is designed for each UAV based on its own state information as well as that of its neighbors. This approach explicitly considers both external disturbances and model uncertainties, with the resulting consensus required to satisfy the H_∞ performance criterion. Disturbances are estimated and compensated using the disturbance-observer-based control (DOBC) method [37], which comprises two components: one for disturbance compensation and another for stabilizing the nominal system to achieve the desired performance. Because these two components adhere to the separation principle (separating state estimation from control design), this approach is particularly suitable for dealing with external disturbances in multi-UAV systems.

ADRC represents a significant advancement in feedback control system design, offering a robust alternative to traditional PID and model-based methods. By actively estimating and compensating for unknown dynamics and disturbances in real time, ADRC effectively overcomes the limitations associated with model dependency and parameter tuning. This approach demonstrates high robustness, maintaining consistent performance and stability even under significant variations in system parameters.

Nevertheless, the conventional ADRC framework still exhibits certain limitations, such as time-delay issues and a relatively high design complexity.

A standard ADRC framework consists of three main components: a Tracking Differentiator (TD), an Extended State Observer (ESO), and a state error feedback control law [38].

In [39], an enhanced ADRC approach is

introduced, which combines multiple types of Extended State Observers (ESOs) to enhance roll stability and disturbance rejection in vertical take-off and landing UAVs (VTOL-UAVs). Simulation results demonstrated that enhanced ADRC achieved faster response times, reduced overshoot, and lower root mean square error (RMSE) in tracking the desired roll angle.

[40] introduces an advanced control framework for Unmanned Aerial Vehicles (UAVs) operating in confined tunnel environments, where precise maneuvering and stable flight are essential for close-range inspection tasks. The major challenge in such missions arises from the complex, nonlinear aerodynamic disturbances and torque effects induced by tunnel walls, which can compromise flight stability and control accuracy. To overcome these challenges, the proposed approach integrates Model Predictive Control (MPC) with modified Control Barrier Function (CBF) constraints. The combination not only ensures safe operation near tunnel boundaries but also enhances the UAV's disturbance rejection capabilities, allowing it to effectively counteract wall-induced aerodynamic effects while maintaining the precision required for close-proximity operations.

2.2.2. Replanning Approaches

In addition to rejection-based strategies, various replanning approaches have been extensively explored. [41] introduces a decentralized path planning algorithm for a group of UAVs operating under weather-induced disturbances. Although each UAV is required to reach designated target points, external disturbances influence the energy consumption. This challenge is addressed by predicting the energy demands based on the estimated disturbances, followed by online path replanning to accommodate these changes. In [42], the external disturbances, primarily wind forces acting on the UAV, are estimated using the Visual-Inertial-Dynamics Fusion (VID-Fusion) method [43]. The resulting forward reachable sets (FRS) are modeled as ellipsoids, which are instantaneously inflated to account for disturbance effects. Path replanning is then performed to ensure that these safety ellipsoidal envelopes (the inflated geometrical shape of the UAV) remain within existing safe corridors, thereby guaranteeing collision avoidance.

In [44], the authors developed a control system for autonomous quadrotors with real-time trajectory re-planning capabilities. They implemented a fully integrated hardware–software

pipeline based on a modified PX4 Autopilot and a Raspberry Pi 5 companion computer. The proposed framework utilizes a minimum-snap trajectory generation method, exploiting the differential flatness property of quadrotors, which enables lightweight and on-board real-time re-planning.

In their study, the mission plan was assumed to be pre-defined based on available data; however, new obstacles could emerge during flight, necessitating trajectory re-planning. The results demonstrated an online trajectory re-planning and optimization framework capable of handling uncertain environments, where obstacles are initially unknown and must be addressed in real time during flight.

In [45], the authors proposed a risk-aware parallel motion planning framework capable of rapidly generating safe, smooth, and aggressive trajectories for quadrotors. The method samples kinodynamically feasible motions in the state space and connects them to form a guiding trajectory, which maintains topological consistency with the execution trajectory and can be computed efficiently in closed form.

To further enhance the trajectory, the framework incorporates a risk-aware Model Predictive Path Integral (RA-MPPI) optimization scheme, which refines the path using a gradient-free and robust approach. The RA-MPPI integrates the Conditional Value-at-Risk (CVaR) metric to automatically evaluate and manage potential risks, such as collisions or violations of dynamic constraints. Moreover, by leveraging Monte Carlo sampling and GPU-based parallelization, the algorithm achieves high computational efficiency.

Importantly, since the RA-MPPI optimization is performed in real time and continuously updates the motion plan based on the current state and perceived risks, the proposed method effectively realizes an online trajectory re-planning framework, enabling the quadrotor to dynamically adapt to changing environments and uncertainties.

In [46], the authors present an efficient framework aimed at enhancing the resilience of quadrotor flights. The proposed system integrates an improved disturbance observer-based safety controller within the autopilot to estimate external forces in real time. Based on these estimates, a high-safety planner is developed to generate

disturbance-aware trajectories that ensure safe operation under uncertain conditions.

The planner comprises a kinodynamic path search module that considers actuator limitations, along with a spatio-temporal joint trajectory optimizer for smooth and feasible motion planning. To guarantee collision avoidance, the method incorporates the uncertainty of external forces into the position estimation by computing Hamilton-Jacobi forward reachability sets of the error dynamics.

3. Proposed Method for Trajectory Replanning of Two UAVs under Unknown Wind Disturbances

In this paper, we propose a systematic framework for trajectory replanning of two UAVs operating under unknown wind forces. Initially, a trajectory is generated for each UAV, which is subsequently adjusted dynamically based on the presence/absence and intensity of wind disturbances. The proposed framework ensures that a safe distance from obstacles is maintained while also considering the interaction between the two UAVs operating in the same environment.

The overall scheme of the proposed system is illustrated in Fig. 1. The environment is modeled as an occupancy grid, constructed from data captured by a camera mounted on each UAV. Accordingly, an obstacle-free path is generated for each UAV through the Dijkstra path planning algorithm. Based on this path, an initial trajectory is then constructed using the Minimum Snap technique outlined in [47], ensuring both smoothness and dynamic feasibility. However, this trajectory may become impractical in the presence of external disturbances or when both UAVs operate simultaneously within the same environment.

The proposed approach addresses two main challenges: (i) both UAVs are subjected to external wind disturbances, and (ii) their simultaneous operation within the same environment introduces the risk of collision. To tackle the first challenge, the wind disturbance is modeled as a combination of two components: a nominal force estimated by VID-Fusion or other estimators, and a bounded noise term following the approach described in [42]. Using Hamilton-Jacobi reachability analysis, the resulting error FRSs are approximated as ellipsoids. By applying the Minkowski sum, the UAV ellipsoid and the

wind disturbance ellipsoid are combined to form a safe ellipsoid that accounts for wind effects. To address the second challenge of collision avoidance, a separating hyperplane is constructed between the two UAV ellipsoids at each replanning instant. Trajectory replanning is then performed based on four factors: the size of the safe ellipsoids, the separating hyperplane, obstacle specifications, and the current trajectory. This framework ensures that UAVs maintain a safe distance from each other and from obstacles, even under wind disturbances in a shared environment.

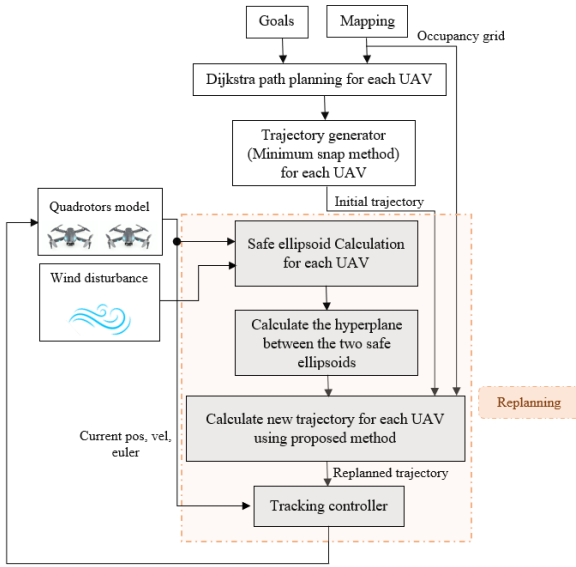


Fig. 1. General scheme of the proposed system

3.1. Mathematical Description

The mathematical formulation of the proposed method involves updating the dimensions of each UAV's safe ellipsoid, followed by the computation of the instantaneous separating hyperplane between them. The detailed procedure for replanning their respective trajectories based on the current environmental conditions is presented in the following sections.

3.1.1. Calculating the Dimensions of the Safe Ellipsoid

The system safety is ensured by computing the Forward Reachable Sets (FRSs) using Hamilton-Jacobi (HJ) reachability analysis. The FRS denotes the set of states that can be reached by the system under bounded disturbances from a given initial region around the current state. The HJ reachability analysis provides a quantitative description of FRS evolution, enabling a robust assessment of safety boundaries for trajectory planning. For completeness, a detailed explanation of HJ reachability analysis and FRSs

is provided in Appendix A.

By applying Hamilton-Jacobi reachability analysis to the error dynamics induced by wind disturbances acting on each UAV, the FRS's error is approximated as an ellipsoid. Subsequently, the Minkowski sum of the wind-disturbance ellipsoid and the UAV shape ellipsoid is computed to obtain an updated safe ellipsoid for trajectory replanning. A detailed description of the Minkowski sum of rotational ellipsoids is provided in Appendix B.

The dynamic model is considered under the assumption of precise Euler angle control, where the angular rates accurately track the desired commands. The full-body nonlinear dynamic model is formulated as follows:

$$\dot{\mathbf{p}} = \mathbf{v}, \quad (1-a)$$

$$\dot{\mathbf{v}} = \frac{1}{m} \left(\mathbf{R} \begin{bmatrix} 0 \\ 0 \\ T_c \end{bmatrix} - \mathbf{F}_{drag} + \mathbf{F}_{ext} \right) - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}, \quad (1-b)$$

$$\dot{\phi} = \dot{\phi}_c, \quad (1-c)$$

$$\dot{\theta} = \dot{\theta}_c, \quad (1-d)$$

$$\dot{\psi} = \dot{\psi}_c. \quad (1-e)$$

Based on the above equations, the UAV's overall nonlinear dynamic model can be concisely expressed as:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{F}_{ext}(t)), \quad (1-f)$$

where $\mathbf{x} = [\mathbf{X}, \mathbf{v}, \phi, \theta, \psi]^T \in \chi \subset \mathbb{R}^{n_x}$ is the state of the system, $\mathbf{X} = [X_x, X_y, X_z]^T$ is the position and $\mathbf{v} = [v_x, v_y, v_z]^T \in \mathbb{R}^3$ is the velocity of the UAV. ϕ, θ, ψ are the roll, pitch and yaw angles. $\mathbf{u} = [\dot{\phi}_c, \dot{\theta}_c, \dot{\psi}_c, T_c]^T \in \mathcal{U} \subset \mathbb{R}^{n_u}$ is the control input where $\dot{\phi}_c, \dot{\theta}_c, \dot{\psi}_c \in \mathbb{R}$ are the command rates of the Euler angles and T_c is thrust command of the UAV in the body frame.

$\mathbf{R} \in \mathbb{R}^{3 \times 3}$ is the rotation matrix parameterized by the Euler angles, $m \in \mathbb{R}$ is the mass of the quadrotor, and $g \in \mathbb{R}$ is the magnitude of gravitational acceleration.

Since the components of rotor drag and drag-like effects are difficult to model and depend on the quadrotor's velocity, they are incorporated into the system model to account for their influence on the disturbance bound.

For a simplified aerodynamic representation, the drag force $\mathbf{F}_{drag}$ is incorporated into the x-y body frame based on a first-order drag model.

As described in [42], $\mathbf{K}_{drag} = \text{diag}\{k_d, k_d, 0\}$ is defined, where k_d denotes the drag coefficient constant, and the drag force in the world frame is given by $\mathbf{F}_{drag} = \mathbf{R} \mathbf{K}_{drag} \mathbf{R}^T \mathbf{v}$.

The external force is defined as $\mathbf{F}_{ext} = \mathbf{b}_{ext} +$

$\mathbf{w}_{ext}$, where $\mathbf{b}_{ext}$ is a nominal constant force obtained by estimating the external force using VID-Fusion as in [43], and $\mathbf{w}_{ext}$ is a bounded noise with maximum bound (w_m), satisfying $\mathbf{w}_{ext} \in \mathbb{W} = \{\mathbf{w} \in \mathbb{R}^{n_w} : \|\mathbf{w}\|_\infty \leq w_m\}$.

A linearized model of the closed-loop error dynamics of the UAV is constructed around the nominal state $\mathbf{x}_0(t)$, the nominal input $\mathbf{u}_0(t)$, and the nominal external disturbance $\mathbf{b}_{ext,0}(t)$. The error state is defined as $\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{x}_0(t)$.

Assuming that the higher-order linearization errors can be neglected in the neighborhood of the nominal states, and that the feedback control law is expressed as $\mathbf{u}(t) = \mathbf{u}_0(t) + \mathbf{K}(t)\mathbf{e}(t)$, the resulting linearized error dynamics, in the presence of an external force, can be expressed following [42], as:

$$\dot{\mathbf{e}}(t) = \mathbf{\Gamma}(t)\mathbf{e}(t) + \mathbf{D}(t)\mathbf{F}_{ext}(t) \quad (2)$$

where:

$$\begin{aligned} \mathbf{\Gamma}(t) &= \mathbf{A}(t) + \mathbf{B}(t)\mathbf{K}(t), & \mathbf{A}(t) &= \frac{\partial f(\mathbf{x}, \mathbf{u}, \mathbf{b}_{ext})}{\partial \mathbf{x}}, \\ \mathbf{B}(t) &= \frac{\partial f(\mathbf{x}, \mathbf{u}, \mathbf{b}_{ext})}{\partial \mathbf{u}}, & \mathbf{D}(t) &= \frac{\partial f(\mathbf{x}, \mathbf{u}, \mathbf{b}_{ext})}{\partial \mathbf{F}_{ext}}, \end{aligned}$$

and $\mathbf{K}(t)$ denotes the feedback gain of the state feedback controller.

Considering the system dynamics sampled at intervals of t_s over N steps, the discrete-time state at step $k \in \{0, 1, 2, \dots, N-1\}$ is defined as $\mathbf{x}^{(k)} = [\mathbf{X}^{(k)}, \mathbf{v}^{(k)}, \phi^{(k)}, \theta^{(k)}, \psi^{(k)}]$. The UAV's shape can be represented by an ellipsoid centered at position $\mathbf{X}^k$ with orientation $\mathbf{R}^k$. This ellipsoid, characterized by a radius r and height h , can be described geometrically as:

$$\xi(\mathbf{X}, \mathbf{Q}) = \{\mathbf{X} | \mathbf{X}^T \mathbf{Q}^{-1} \mathbf{X} \leq 1\} \quad (3)$$

or:

$$\begin{aligned} \xi(\mathbf{X}^k, \mathbf{Q}_{uav}^k) \\ = \{\mathbf{X} | (\mathbf{X} - \mathbf{X}^k)^T (\mathbf{Q}_{uav}^k)^{-1} (\mathbf{X} - \mathbf{X}^k) \leq 1\}, \end{aligned} \quad (4)$$

$$\mathbf{Q}_{uav}^k = \mathbf{R}^k \mathbf{q} (\mathbf{R}^k)^T; \quad \mathbf{q} = \text{diag}\{r^2, r^2, h^2\}, \quad (5)$$

Here, the shape matrix of the UAV, $\mathbf{Q}_{uav}^k$, is calculated as in (5).

The FRS's error can be expressed as an approximate ellipsoid representing the external force disturbance, using the Hamilton-Jacobi reachability analysis method applied to the error dynamics as in (6). The shape matrix of the wind disturbance, $\mathbf{Q}_{dist}^k$, is computed from (7), and the maximal ellipsoidal reachable set of the error dynamics, $\mathbf{Q}_{0,uav}^k$, is obtained as in [48]:

$$\begin{aligned} \xi(\mathbf{X}^k, \mathbf{Q}_{dist}^k) \\ = \{\mathbf{X} | (\mathbf{X} - \mathbf{X}^k)^T (\mathbf{Q}_{dist}^k)^{-1} (\mathbf{X} - \mathbf{X}^k) \leq 1\} \end{aligned} \quad (6)$$

$$\mathbf{Q}_{dist}^k = [\exp(\mathbf{\Gamma}^k t_s) (\mathbf{Q}_{0,uav}^k \boxplus \mathbf{Q}_d^k) \exp(\mathbf{\Gamma}^{k,T} t_s)]_{3 \times 3} \quad (7)$$

Now, the Minkowski sum of the UAV ellipsoid in (4) and the wind-disturbance ellipsoid in (6) can be applied to obtain the safe ellipsoid $\xi(\mathbf{X}^k, \mathbf{Q}_{safe}^k)$, as illustrated in Fig. 2 and defined in (8). Here, $\mathbf{Q}_{safe}^k$ denotes the shape matrix of the safe ellipsoid, and $\boxplus$ represents the Minkowski sum symbol. Subsequently, the matrix $\mathbf{Q}_{uav}^k$ is updated according to (9).

$$\begin{aligned} \mathbf{Q}_{safe}^k &= \mathbf{Q}_{uav}^k \boxplus \mathbf{Q}_{dist}^k \\ &= (1 + \alpha) \mathbf{Q}_{uav}^k + \left(1 + \frac{1}{\alpha}\right) \mathbf{Q}_{dist}^k, \quad \alpha = \end{aligned} \quad (8)$$

$$\begin{aligned} \sqrt{\frac{\text{trace}(\mathbf{Q}_{dist}^k)}{\text{trace}(\mathbf{Q}_{uav}^k)}}, \\ \mathbf{Q}_{0,uav}^{k+1} = \mathbf{Q}_{0,uav}^k + \mathbf{Q}_d^k \end{aligned} \quad (9)$$

Once the safe ellipsoid is determined, its semi-axes can be computed from the shape matrix as follows:

$$s_i = \lambda_i^{-1/2} q_i \quad (10)$$

In this context, λ_i represents the eigenvalue corresponding to the i^{th} eigenvector $\mathbf{q}_i$. The length of each semi-axis is determined by the magnitude of its associated eigenvalue, while the orientation is governed by the associated eigenvector.



Fig. 2. The inner ellipsoid, $\xi(\mathbf{X}^k, \mathbf{Q}_{dist}^k)$, represents the ellipsoidal approximation of the FRSs. The middle ellipsoid, $\xi(\mathbf{X}^k, \mathbf{Q}_{uav}^k)$, and the outer ellipsoid, $\xi(\mathbf{X}^k, \mathbf{Q}_{safe}^k)$, represent the UAV ellipsoid and the safe ellipsoid, respectively.

3.1.2. Calculating the Separating Hyperplane between the Safe Ellipsoids of Two UAVs

After obtaining the safe ellipsoid of each UAV in the presence of wind disturbance, the next step is to construct a hyperplane that separates them. Specifically, a separating hyperplane for two ellipsoids $\xi_1(X^k, Q_{safe}^k)$ and $\xi_2(X^k, Q_{safe}^k)$ is defined as a hyperplane that places $\xi_1(X^k, Q_{safe}^k)$ on one side and $\xi_2(X^k, Q_{safe}^k)$ on the other, as illustrated in Fig. 3.

First, using CVX (a Matlab-based modeling system for convex optimization), the closest distance segment between two points of the two safe ellipsoids is calculated. Then a hyperplane is constructed perpendicular to this line at its midpoint. The resulting hyperplane is described by (11) and determines two (closed) half-spaces, defined by (12) and (13), each containing one ellipsoid.

$$H = \{X \in \mathbb{R}^n \mid \langle n, X \rangle = c\} \quad (11)$$

or:

$$A * x + B * y + C * z - D = 0$$

$$S_1 = \{X \in \mathbb{R}^n \mid \langle n, X \rangle \leq c\} \quad (12)$$

$$S_2 = \{X \in \mathbb{R}^n \mid \langle n, X \rangle \geq c\} \quad (13)$$

Here, A, B, and C denote the components of the normal vector $\vec{n} = (A, B, C)$ of the hyperplane, $X = [x, y, z]$ represents an arbitrary point on the hyperplane, and c is a scalar bias term.

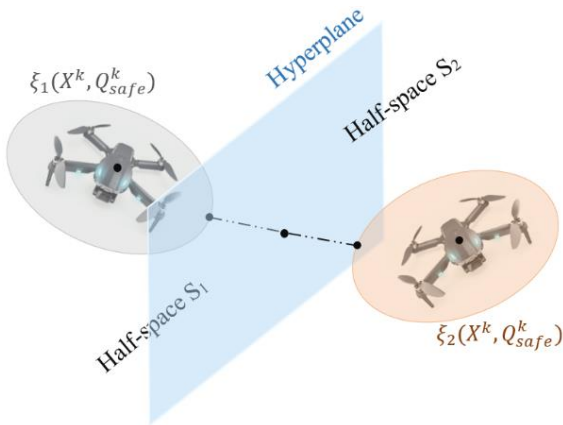


Fig. 3. A separating hyperplane between two safe ellipsoids, $\xi_1(X^k, Q_{safe}^k)$ and $\xi_2(X^k, Q_{safe}^k)$

3.1.3. Trajectory Replanning

The basic concept of our method is to apply “forces” that adjust the positions of trajectory points according to their proximity to obstacles, the relative distance of the two UAVs, and the magnitude of the external disturbance affecting each UAV at that moment.

Conceptually, the replanning mechanism can

be interpreted like the elastic band approach commonly used in motion planning [35]. In this framework, the discrete trajectory points are treated as nodes connected by virtual elastic links. Each link exerts a “tension force” proportional to the relative displacement between consecutive points, which acts to smooth the path and minimize sharp curvature.

The resultant vector from the current point p_i toward its neighboring points p_{i-1} and p_{i+1} thus reflects the local geometric curvature of the trajectory, similar to the restoring forces in an elastic band.

Simultaneously, additional “repulsive forces” are introduced to account for obstacle proximity, inter-UAV separation constraints, and external disturbances.

The updated position of each trajectory point is then obtained by balancing these competing effects, yielding a physically intuitive, force-based trajectory deformation process that continuously adapts to the surrounding environment.

At each point p_i along the trajectory, as illustrated in Fig. 4, two vectors are defined: $\vec{p_i p_{i+1}}$, directed from point p_i to the next point p_{i+1} , and $\vec{p_i p_{i-1}}$, connecting point p_i to the preceding point p_{i-1} .

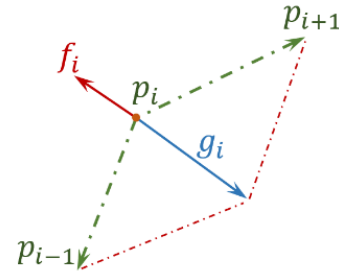


Fig. 4. The elastic force f_i acting on the path

The vector g_i , defined as the resultant of these two vectors, offers geometric information about the local structure of the trajectory. Its orientation indicates the curvature at the corresponding point, pointing inward toward the obstacle in convex regions and outward, away from the obstacle, in concave regions:

$$g_i \triangleq (p_{i-1}(t) - p_i(t)) + (p_{i+1}(t) - p_i(t)) \quad (14)$$

This vector, computed at each trajectory point, forms the basis for reshaping the trajectory. The remodeling process takes into consideration the spatial configuration of surrounding obstacles, denoted by O , and modifies the trajectory to ensure both safety and smoothness during

navigation.

$$\mathbf{f}_i = -K * \mathbf{g}_i(\mathbf{p}_i(t), \mathbf{O}(t)) \quad (15)$$

Acting like an elastic force $\mathbf{f}_i$, this vector is generated by the deformation (replanning) of the trajectory, with a shaping coefficient K used to regulate the magnitude of adjustment applied to the trajectory.

Assume that $D(\xi_{safe_i}, \mathbf{O})$ represents the minimum distance between the safe ellipsoid $\xi_i(\mathbf{X}, \mathbf{Q}_{safe})$ (hereafter referred to as ξ_{safe_i} representing the UAV under wind disturbance) and the obstacle $\mathbf{O}$. The value of this variable is set to zero whenever the trajectory point lies beyond the obstacle's reach r_{obs} :

$$D(\xi_{safe_i}, \mathbf{O}) = \max(0; r_{obs} - d(\xi_{safe_i}, \mathbf{O})) \quad (16)$$

In this expression, the term $d(\xi_{safe_i}, \mathbf{O})$ denotes the distance between the safe ellipsoid ξ_{safe_i} and the obstacle $\mathbf{O}$. Similarly, let $D(\xi_{safe_i}, \mathbf{H})$ represent the minimal distance between the safe ellipsoid $\xi_i(\mathbf{X}, \mathbf{Q}_{safe})$ and hyperplane $\mathbf{H}$. The value of this variable is set to zero when the trajectory point lies outside the influence domain of the hyperplane, denoted by r_{hyp} . In this context, the term $d(\xi_{safe_i}, \mathbf{H})$ denotes the distance between the safe ellipsoid ξ_{safe_i} and the hyperplane $\mathbf{H}$:

$$\begin{aligned} D(\xi_{safe_i}, \mathbf{H}) \\ = \max(0; r_{hyp} - |d(\xi_{safe_i}, \mathbf{H})|) \\ * \quad \text{sgn}(d(\xi_{safe_i}, \mathbf{H})) \end{aligned} \quad (17)$$

The external force $\mathbf{F}_i(\mathbf{p}_i(t), \mathbf{O}, \mathbf{H})$ is defined as the combination of two components, as outlined in (20). The first component is primarily determined by two factors: (i) the external wind disturbance $\mathbf{F}_{ext}$, and (ii) the repulsive force exerted by nearby obstacles $\mathbf{O}$, as formulated in (18). The second component arises from the repulsive influence of the hyperplane $\mathbf{H}$, as expressed in (19):

$$\mathbf{F}_i(\mathbf{p}_i(t), \mathbf{O}(t), \xi_{safe}(t)) = D(\xi_{safe_i}, \mathbf{O}) * \frac{-\nabla D(\xi_{safe_i}, \mathbf{O})}{|\nabla D(\xi_{safe_i}, \mathbf{O})|}, \quad (18)$$

$$\mathbf{F}_i(\mathbf{p}_i(t), \mathbf{H}(t), \xi_{safe}(t)) = D(\xi_{safe_i}, \mathbf{H}) * \frac{-\nabla D(\xi_{safe_i}, \mathbf{H})}{|\nabla D(\xi_{safe_i}, \mathbf{H})|}, \quad (19)$$

$$\begin{aligned} \mathbf{F}_i(\mathbf{p}_i(t), \mathbf{OH}) = & \mathbf{F}_i(\mathbf{p}_i(t), \mathbf{O}(t), \xi_{safe_i}(t)) \\ & + \mathbf{F}_i(\mathbf{p}_i(t), \mathbf{H}(t), \xi_{safe_i}(t)). \end{aligned} \quad (20)$$

A time-dependent solution, which accounts for wind disturbances and the presence of other UAVs in the same environment, is more appropriate for trajectory modification rather than a static one; the semi-static approach can be formulated by incorporating the time derivatives of the trajectory points into (14) to obtain (21):

$$\begin{aligned} \dot{\mathbf{p}}_i(t) = & \frac{1}{T} [\mathbf{g}_i + K^{-1} * \mathbf{F}_i(\mathbf{p}_i(t), \mathbf{OH})] \\ = & \frac{1}{T} \left[\mathbf{g}_i + K^{-1} * \right. \\ & \left. \left(\mathbf{F}_i(\mathbf{p}_i(t), \mathbf{O}(t), \xi_{safe_i}(t)) + \right. \right. \\ & \left. \left. \mathbf{F}_i(\mathbf{p}_i(t), \mathbf{H}(t), \xi_{safe_i}(t)) \right) \right], \end{aligned} \quad (21)$$

Here, T represents a dynamic parameter that controls how quickly the trajectory adapts. On the right-hand side of (21), two forces are considered: the internal elastic force $\mathbf{f}_i$ and the external force $\mathbf{F}_i$. The algorithm accounts for several inputs, including the strength of the wind disturbance, the quadrotor's proximity to nearby obstacles, and its distance from the hyperplane. These inputs enable the system to modify the trajectory more effectively and adaptively in changing environments.

First, the force resulting from both wind disturbances and nearby obstacles is calculated. A two-dimensional circular obstacle, as illustrated in Fig. 5, is used to clarify the state equations. Subsequently, more complex scenarios involving three-dimensional states and environments with multiple obstacles will be presented.

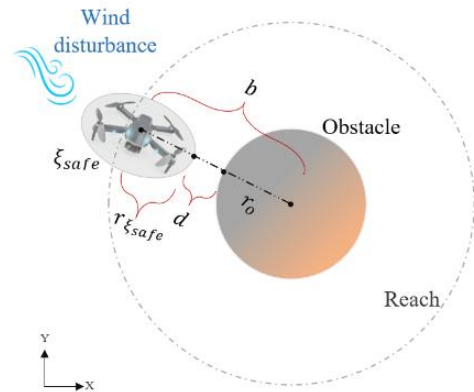


Fig. 5. A cylindrical obstacle in the presence of wind disturbance affecting the UAV

As illustrated in Fig. 5, the distance $d(\xi_{safe_i}, \mathbf{O})$, defined in (16) between the safe ellipsoid ξ_{safe_i} and the obstacle, can be computed as follows:

$$d_i(\xi_{safe_i}, \mathbf{O}) = b_i - r_o - r_{\xi_{safe_i}} \quad (22)$$

$$= \sqrt{(x_{p_i} - x_o)^2 + (y_{p_i} - y_o)^2} - r_o - r_{\xi_{safe_i}}$$

$$; b = \sqrt{(x_{p_i} - x_o)^2 + (y_{p_i} - y_o)^2},$$

Here, r_o represents the radius of the obstacle, whose center is located at (x_o, y_o) , and (x_{p_i}, y_{p_i}) are the coordinates of the i^{th} point along the trajectory. Accordingly, from (16), we can write:

$$D(\xi_{safe_i}, \mathbf{O}) = \max \left(0; r_{obs} - \sqrt{(x_{p_i} - x_o)^2 + (y_{p_i} - y_o)^2} + r_o + r_{\xi_{safe_i}} \right) \quad (23)$$

The force $\mathbf{F}_i(\mathbf{p}_i, \mathbf{O}, \xi_{safe_i})$ can be determined using (18) and (23), and is given by:

$$\begin{aligned} F_i(p_i, O, \xi_{safe_i}) &= F_{i,x}(p_i, O, \xi_{safe_i}) \hat{i} \\ &\quad + F_{i,y}(p_i, O, \xi_{safe_i}) \hat{j} \\ &= \frac{\max \left(0; r_{obs} - \sqrt{(x_{p_i} - x_o)^2 + (y_{p_i} - y_o)^2} + r_o + r_{\xi_{safe_i}} \right) * (x_{p_i} - x_o)}{\sqrt{(x_{p_i} - x_o)^2 + (y_{p_i} - y_o)^2}} \hat{i} \\ &\quad + \frac{\max \left(0; r_{obs} - \sqrt{(x_{p_i} - x_o)^2 + (y_{p_i} - y_o)^2} + r_o + r_{\xi_{safe_i}} \right) * (y_{p_i} - y_o)}{\sqrt{(x_{p_i} - x_o)^2 + (y_{p_i} - y_o)^2}} \hat{j} \end{aligned} \quad (24)$$

Secondly, for the computation of the hyperplane repulsive force, as defined in (17), the distance $d(\xi_{safe_i}, \mathbf{H})$ between the safe ellipsoid and the hyperplane is obtained as follows:

$$d(\xi_{safe_i}, \mathbf{H}) = \frac{A * x_{p_i} + B * y_{p_i} + C * z_{p_i} - D}{\sqrt{A^2 + B^2 + C^2}} \quad (25)$$

Here, A, B , and C represent the components of the hyperplane's normal vector, while D denotes the hyperplane's offset from the origin. The coordinates $x_{p_i}, y_{p_i}, z_{p_i}$ correspond to the position of the proximal point on the ellipsoid. Therefore, based on (17), the following expression can be written:

$$\begin{aligned} D(\xi_{safe_i}, \mathbf{H}) &= \max \left(0; r_{hyp} - \frac{|A * x_{p_i} + B * y_{p_i} + C * z_{p_i} - D|}{\sqrt{A^2 + B^2 + C^2}} \right) \\ &\quad * \text{sgn}(d(\xi_{safe_i}, \mathbf{H})) \end{aligned} \quad (26)$$

Finally, the force $\mathbf{F}_i(\mathbf{p}_i, \mathbf{H}, \xi_{safe_i})$ can be computed by substituting (26) into (19).

In scenarios where the trajectory is influenced solely by static obstacles and under steady conditions, the time derivatives of the trajectory points vanish. Consequently, (21) can be simplified and expressed as follows:

$$\begin{aligned} &\begin{bmatrix} -2 & 1 & 0 & \dots & 0 & 0 & 0 \\ 1 & -2 & 1 & \dots & 0 & 0 & 0 \\ 0 & 1 & -2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -2 & 1 & 0 \\ 0 & 0 & 0 & \dots & 1 & -2 & 1 \\ 0 & 0 & 0 & \dots & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ \vdots \\ p_{n-2} \\ p_{n-1} \\ p_n \end{bmatrix} * \begin{bmatrix} p_{start} \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ p_{goal} \end{bmatrix} \\ &= -K^{-1} \begin{bmatrix} F_1(p_1, O, \xi_{safe_1}) + F_1(p_1, H, \xi_{safe_1}) \\ F_2(p_2, O, \xi_{safe_2}) + F_2(p_2, H, \xi_{safe_2}) \\ F_3(p_3, O, \xi_{safe_3}) + F_3(p_3, H, \xi_{safe_3}) \\ \vdots \\ F_{n-2}(p_{n-2}, O, \xi_{safe_{n-2}}) + F_{n-2}(p_{n-2}, H, \xi_{safe_{n-2}}) \\ F_{n-1}(p_{n-1}, O, \xi_{safe_{n-1}}) + F_{n-1}(p_{n-1}, H, \xi_{safe_{n-1}}) \\ F_n(p_n, O, \xi_{safe_n}) + F_n(p_n, H, \xi_{safe_n}) \end{bmatrix} \end{aligned} \quad (27)$$

Here, $\mathbf{p}_0 = \mathbf{p}_{start}$ denotes the starting point of the trajectory, and $\mathbf{p}_{n+1} = \mathbf{p}_{goal}$ represents its goal point.

To regulate the trajectory deviations, the following equation is employed to constrain the total displacement relative to the initial position $(x_{p_i}^0, y_{p_i}^0)$:

$$\|\text{disp}\|_{\varphi} = \frac{1}{n} * \sum_{i=1}^n \sqrt{(x_{p_i} - x_{p_i}^0)^2 + (y_{p_i} - y_{p_i}^0)^2} \quad (28)$$

By expanding (27), the system of equations presented in (29) can be derived as follows:

$$\left\{ \begin{array}{l} -2x_{p_1} + x_{p_2} + x_{start} + K^{-1} * \left(F_{1,x}(p_1, O, \xi_{safe_1}) + F_{1,x}(p_1, H, \xi_{safe_1}) \right) = 0 \\ -2x_{p_2} + x_{p_1} + x_{p_3} + K^{-1} * \left(F_{2,x}(p_2, O, \xi_{safe_2}) + F_{2,x}(p_2, H, \xi_{safe_2}) \right) = 0 \\ -2x_{p_3} + x_{p_2} + x_{p_4} + K^{-1} * \left(F_{3,x}(p_3, O, \xi_{safe_3}) + F_{3,x}(p_3, H, \xi_{safe_3}) \right) = 0 \\ \dots = 0 \\ -2x_{p_{n-2}} + x_{p_{n-3}} + x_{p_{n-1}} + K^{-1} * \left(F_{n-2,x}(p_{n-2}, O, \xi_{safe_{n-2}}) + F_{n-2,x}(p_{n-2}, H, \xi_{safe_{n-2}}) \right) = 0 \\ -2x_{p_{n-1}} + x_{p_{n-2}} + x_{p_n} + K^{-1} * \left(F_{n-1,x}(p_{n-1}, O, \xi_{safe_{n-1}}) + F_{n-1,x}(p_{n-1}, H, \xi_{safe_{n-1}}) \right) = 0 \\ -2x_{p_n} + x_{p_{n-1}} + x_{goal} + K^{-1} * \left(F_{n,x}(p_n, O, \xi_{safe_n}) + F_{n,x}(p_n, H, \xi_{safe_n}) \right) = 0 \end{array} \right. \quad (29-a)$$

$$\left\{ \begin{array}{l} -2y_{p_1} + y_{p_2} + y_{start} + K^{-1} * \left(F_{1,y}(p_1, O, \xi_{safe_1}) + F_{1,y}(p_1, H, \xi_{safe_1}) \right) = 0 \\ -2y_{p_2} + y_{p_1} + y_{p_3} + K^{-1} * \left(F_{2,y}(p_2, O, \xi_{safe_2}) + F_{2,y}(p_2, H, \xi_{safe_2}) \right) = 0 \\ -2y_{p_3} + y_{p_2} + y_{p_4} + K^{-1} * \left(F_{3,y}(p_3, O, \xi_{safe_3}) + F_{3,y}(p_3, H, \xi_{safe_3}) \right) = 0 \\ \dots = 0 \\ -2y_{p_{n-2}} + y_{p_{n-3}} + y_{p_{n-1}} + K^{-1} * \left(F_{n-2,y}(p_{n-2}, O, \xi_{safe_{n-2}}) + F_{n-2,y}(p_{n-2}, H, \xi_{safe_{n-2}}) \right) = 0 \\ -2y_{p_{n-1}} + y_{p_{n-2}} + y_{p_n} + K^{-1} * \left(F_{n-1,y}(p_{n-1}, O, \xi_{safe_{n-1}}) + F_{n-1,y}(p_{n-1}, H, \xi_{safe_{n-1}}) \right) = 0 \\ -2y_{p_n} + y_{p_{n-1}} + y_{goal} + K^{-1} * \left(F_{n,y}(p_n, O, \xi_{safe_n}) + F_{n,y}(p_n, H, \xi_{safe_n}) \right) = 0 \end{array} \right. \quad (29-b)$$

The parameter K is selected to minimize the term $\|disp\|_\phi$ in (28), using Matlab's `fmincon` optimization function, subject to the constraints specified in the system of (29).

To determine the optimal value of the parameter K , the problem was formulated as a constrained nonlinear optimization task, in accordance with the implementation guidelines provided in the official MathWorks documentation.

In this formulation, the parameter K does not explicitly appear in the objective function (28), which depends only on the trajectory points. Nevertheless, K affects these points indirectly through the set of equality constraints defined in (29). The initial value of K , along with its lower and upper bounds, are also specified in the optimization process.

Once the value of K is determined, (21) is revisited to compute the parameter T and update the trajectory coordinates accordingly. Whenever the UAV approaches an obstacle or the hyperplane separating it from another UAV, the corresponding surfaces act as virtual walls constraining the UAV's motion. Accordingly, the obstacle and hyperplane force models are formulated in (30) and (31), respectively:

$$F_i(p_i, O) = r_{obs} - p_i(t) \quad (30)$$

$$F_i(p_i, H) = r_{hyp} - p_i(t) \quad (31)$$

By employing (30), (31), and (27), the following state equation can be formulated for the x-direction:

$$\begin{aligned} [\dot{x}_t]_{n \times 1} &= \frac{1}{T} * \left[[A]_{n \times n} * [x_t]_{n \times 1} + [P_{s,t}]_{n \times 1} + K^{-1} \right. \\ &\quad \left. * [r_{obs} - x_t]_{n \times 1} \right. \\ &\quad \left. + [r_{hyp} - x_t]_{n \times 1} \right], \quad (32) \\ [\dot{x}_t]_{n \times 1} &= \frac{1}{T} * \left(\underbrace{[A]_{n \times n} - 2 * K^{-1} * [I]_{n \times 1}}_S \right) * \\ &\quad [x_t]_{n \times 1} + \frac{1}{T} * \left[[P_{s,t}]_{n \times 1} + \frac{K^{-1}}{T} * (r_{obs} + r_{hyp}) * [I]_{n \times 1} \right], \end{aligned}$$

The matrix S is a tridiagonal Toeplitz system matrix, whose eigenvalues can be calculated using (33). For $i = 1$, it results that $\lambda_1 = \lambda_{max} = 3/T_{max}$, and (33) can be solved for T , where $T_{max,d}$ is the required settling time:

$$\lambda_i = \frac{1}{T} * [-2 + 2 \cos(i * \pi / (n + 1)) - 2 * K^{-1}] \quad (33)$$

$$T = \frac{1}{3} [2 - 2 \cos(\pi / (n + 1)) + 2 * K^{-1}] * T_{max,d} \quad (34)$$

By applying the backward Euler method, $\dot{p}$ is

discretized to solve the transition between two successive trajectory states, $\mathbf{p}^{(k)}$ and $\mathbf{p}^{(k+1)}$, with the discrete time t_k and a fixed time step $\Delta t = t_{k+1} - t_k$. Using this approach, the system of differential equations in (21) is solved, leading to the following expression:

$$\begin{aligned} \mathbf{p}_i^{(k+1)} &= \mathbf{p}_i^{(k)} \\ &+ \frac{1}{T} \left[\mathbf{A} * \mathbf{p}_i^{(k+1)} + \mathbf{K}^{-1} \right. \\ &* \left(\mathbf{F}_i^{(k)} \left(\mathbf{p}_i(t_k), \mathbf{O}(t), \xi_{safe_i}(t_k) \right) \right. \\ &\left. \left. + \mathbf{F}_i^{(k)} \left(\mathbf{p}_i(t_k), \mathbf{H}(t), \xi_{safe_i}(t_k) \right) \right) + \mathbf{p}_{s,t} \right] \Delta t \end{aligned} \quad (35)$$

Thus, $\mathbf{p}^{k+1}$ is finally obtained by solving the iterative equation in (36):

$$\begin{aligned} \underbrace{\left[\mathbf{I} - \frac{\Delta t}{T} * \mathbf{A} \right]}_{\mathbf{G}} * \underbrace{\mathbf{p}_i^{(k+1)}}_{\mathbf{X}} &= \mathbf{p}_i^{(k)} \\ &+ \frac{1}{T} \left[\mathbf{K}^{-1} \right. \\ &* \left(\mathbf{F}_i^{(k)} \left(\mathbf{p}_i(t_k), \mathbf{O}(t), \xi_{safe_i}(t_k) \right) \right. \\ &\left. \left. + \mathbf{F}_i^{(k)} \left(\mathbf{p}_i(t_k), \mathbf{H}(t), \xi_{safe_i}(t_k) \right) \right) + \mathbf{p}_{s,t} \right] \Delta t \end{aligned} \quad (36)$$

The matrix $\mathbf{G}$ is a tridiagonal coefficient matrix, structured as follows:

$$\begin{aligned} \mathbf{G} &= \begin{bmatrix} b & c & 0 & \dots & 0 \\ a & b & c & \ddots & \vdots \\ 0 & a & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & b & c \\ 0 & \dots & 0 & a & b \end{bmatrix}; \quad a = c = -\frac{\Delta t}{T}, \quad b \\ &= 1 + 2\frac{\Delta t}{T} \end{aligned} \quad (37)$$

The system of linear equations given in (36), expressed in the form $\mathbf{G} * \mathbf{X} = \mathbf{W}$, is solved using Thomas's method [49]. In this equation, the only unknown is $\mathbf{p}_i^{(k+1)}$, representing the updated trajectory points coordinate (x_i, y_i, z_i) .

4. Results

This study presents a performance metric designed to capture the primary objectives of the proposed approach, focusing on two key aspects:

1. **Trajectory enhancement:** To ensure energy efficiency and motion feasibility, both quadrotors should follow trajectories with minimal snap. Additionally, the replanned trajectories should closely approximate the optimal trajectories obtained via the Dijkstra algorithm, in terms of trajectory length, effective obstacle avoidance, and maintaining safe distances from other quadrotors.

2. **Computational Efficiency:** The

computation time for generating new trajectories for both quadrotors should be minimized, enabling rapid adaptation to dynamic environmental changes.

To verify that these requirements are satisfied, the following evaluation criteria are employed:

- **Trajectory Length Evaluation:** To evaluate the deviation from the original optimal trajectory computed using Dijkstra's algorithm, the total lengths of the trajectories before and after replanning are compared for both quadrotors. The results are reported in Table 1.

- **Computation Time Evaluation:** The method's ability to respond in real-time is evaluated by analyzing the time required to compute new trajectory points during the replanning process.

- **Method Comparison:** The proposed approach is further evaluated through comparison with the NMPC technique, demonstrating improved trajectory efficiency along with a significant reduction in computation time, as illustrated in Table 1.

This section presents the simulation environment used to evaluate the proposed method, which is implemented in Matlab/Simulink. The quadrotor platform considered in the simulations is the Crazyflie 2.1, a racing-type UAV. The vehicle has a mass of 0.027 kg , and moments of inertia given by $I_{xx} = 1.43 \times 10^{-5} \text{ kg.m}^2$, $I_{yy} = 1.43 \times 10^{-5} \text{ kg.m}^2$, $I_{zz} = 2.89 \times 10^{-5} \text{ kg.m}^2$. The physical dimensions of the quadrotor are $0.092 \times 0.092 \times 0.029 \text{ m}$ (Width $\times$ Height $\times$ Depth), measured from motor axis to motor axis, including the motor mount feet.

The applied disturbance bounds are determined based on realistic assumptions regarding wind disturbances, although they do not represent worst-case scenarios. These disturbances are chosen to correspond to the size and characteristics of the Crazyflie 2.1 drone, drawing on previous experimental data and assumptions reported in earlier studies. Moreover, the intensity of the wind disturbances is carefully selected to match the thrust capabilities of the UAV. These disturbance levels are chosen to ensure that the UAV can maintain its trajectory and stability without exceeding the system's thrust limits, thereby ensuring consistent and reliable performance across varying environmental conditions.

The UAV considered in this study is an underactuated system, as the number of

independent control inputs, namely the four rotor thrusts, is fewer than the total number of degrees of freedom, which is six. Specifically, the UAV can directly control its vertical translational motion and rotational motions, including roll, pitch, and yaw. In contrast, horizontal translational motions along the x and y axes are achieved indirectly through attitude control by adjusting the roll and pitch angles. This underactuation property is taken into account in the control design and analysis presented in the paper.

Because of this underactuation, the system exhibits zero dynamics, which correspond to the internal attitude dynamics that arise when the translational outputs are constrained or regulated to their desired values. Ensuring the stability of these zero dynamics is therefore essential for guaranteeing the overall stability of the UAV.

In the proposed PD-based control scheme, a cascade control architecture is adopted: the outer-loop PD controller generates the desired attitude references based on the position tracking errors, while the inner-loop PD controller stabilizes the attitude dynamics. As a result, the zero dynamics associated with the attitude subsystem remain stable due to the inner-loop PD controller. This cascade structure ensures the stability of the overall closed-loop system and enables accurate position tracking performance.

The proposed method addresses trajectory replanning for racing UAVs, considering both mutual collision avoidance and obstacle avoidance in the presence of wind disturbances. To achieve this objective, the replanning procedure is carried out through the following main stages:

1. Initially, a collision-free path is generated for each quadrotor using Dijkstra's algorithm. Based on these paths, initial trajectories are then constructed using the Minimum Snap method.

2. The safe ellipsoid, denoted as ξ_{safe} , is computed for each quadrotor, while the quadrotor shape matrix $Q_{0,uav}^{k+1}$ is continuously updated at each time step.

3. The separating hyperplane between the safe ellipsoids $\xi_1(X^k, Q_{safe}^k)$ and $\xi_2(X^k, Q_{safe}^k)$, corresponding to the first and second quadrotors, respectively, is then calculated.

4. The optimal value of parameter K , which determines the extent to which the trajectory

shape is modified during the replanning process, is determined.

5. The scalar parameter T , which influences the dynamic behavior of the trajectory adjustment process, is evaluated.

6. Finally, the updated coordinates (x, y, z) of the trajectory points at the next iteration p^{k+1} are computed.

Initially, a path for each quadrotor is generated using Dijkstra's algorithm, which serves as the foundation for designing the initial trajectory using the Minimum Snap technique. Subsequently, the safe ellipsoid ξ_{safe} for each quadrotor is computed according to (8), while the corresponding quadrotor's shape matrix is continuously updated at each time step according to (9).

The coordinates x_i , y_i , and z_i of the trajectory points of each of the two quadrotors, obtained from the Dijkstra path, are substituted into (22) to compute the distance $d(\xi_{safe_i}, \mathbf{O})$ between the safe ellipsoid and the obstacles. This distance is then used in (16) to evaluate $D(\xi_{safe_i}, \mathbf{O})$. When the safe ellipsoid is located outside the obstacle's reach (i.e., beyond its influence region), $D(\xi_{safe_i}, \mathbf{O})$ is set to zero. Finally, $D(\xi_{safe_i}, \mathbf{O})$ is substituted into (18) to determine the force $F_i(\mathbf{p}_i, \mathbf{O}, \xi_{safe})$ acting on the trajectory point $\mathbf{p}_i$. Similarly, $D(\xi_{safe_i}, \mathbf{H})$ is computed to evaluate the force $F_i(\mathbf{p}_i, \mathbf{H}, \xi_{safe})$ induced by the separating hyperplane $\mathbf{H}$. The resulting external force $F_i(\mathbf{p}_i, \mathbf{O}, \mathbf{H})$ influencing the trajectory is obtained by summing the obstacle-induced and hyperplane-induced forces, namely $F_i(\mathbf{p}_i, \mathbf{O}, \xi_{safe})$ and $F_i(\mathbf{p}_i, \mathbf{H}, \xi_{safe})$.

The force itself is influenced by the quadrotor's proximity to obstacles and to the other quadrotor, as well as the magnitude of the wind disturbance. The parameter K is determined using Matlab's `fmincon` function by minimizing the expression $\|\mathbf{disp}\|_\phi$ in (28), subject to the constraints specified in (29).

An occupancy map is employed to model an environment containing three obstacles that form a narrow corridor. Fig. 7b illustrates this environment, in which both quadrotors safely navigate from their respective starting points $[1, 8.5, 5]m$ and $[1, 10.5, 4]m$ to their assigned target points $[17.5, 7, 0]m$ and $[17.5, 12, 1]m$,

respectively, while being subjected to wind disturbances with components of $[0, 0.3, 0]N$ along the x, y, and z axes.

Fig. 6 shows the top view of the initial trajectories for the two quadrotors, with red and purple representing the first and second quadrotors, respectively. These trajectories were generated using the Minimum Snap Trajectory Generation approach, based on the paths obtained from the Dijkstra algorithm for each quadrotor. Nevertheless, as can be seen, the initial trajectories are suitable, as they bring the quadrotors dangerously close to obstacles. Moreover, it's essential to avoid any potential collisions between the two quadrotors during mission execution. The presence of wind disturbances further complicates the situation, making the planned paths less reliable and increasing the risk of collisions.

At each time step, a safe ellipsoid is constructed around each quadrotor, with red representing the first quadrotor and blue representing the second. A light blue hyperplane is then formed to separate the two ellipsoids, while the green dashed line indicates the shortest distance between their closest points at that specific moment.

Fig. 7 illustrates the initial (dashed) and replanned (solid) trajectories of the quadrotors, with red corresponding to the first quadrotor and blue to the second.

As shown in Fig. 7a, which provides a top view of the environment, the initial dashed trajectory is pushed away from obstacles as the quadrotor gets closer to those obstacles and is influenced by the wind disturbance, resulting in the solid-line replanned trajectory.

In the 3D view presented in Fig. 7b, the trajectories of the two quadrotors diverge due to the separating hyperplane generated between their respective safe ellipsoids. As the quadrotors approach each other, the replanned trajectory of the first quadrotor (red) rises, while the second quadrotor's trajectory (blue) descends, ensuring safe separation.

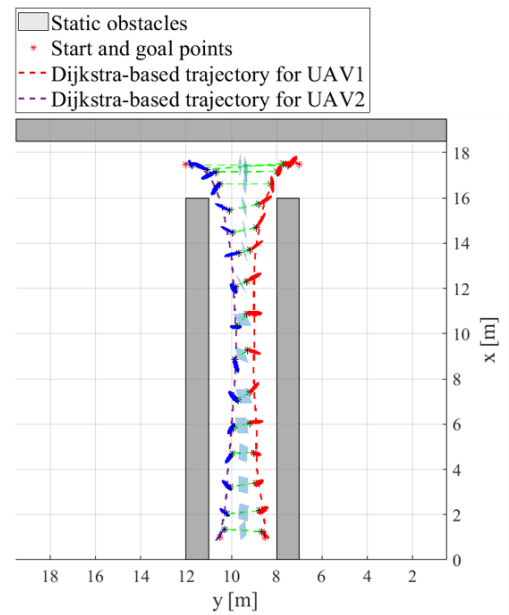
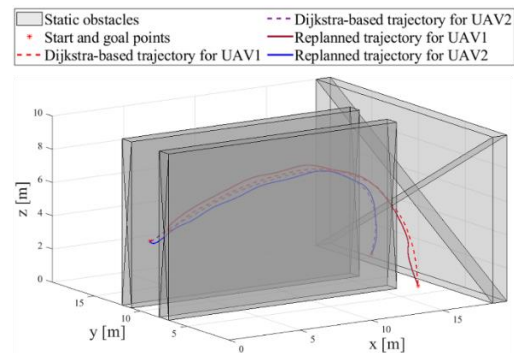
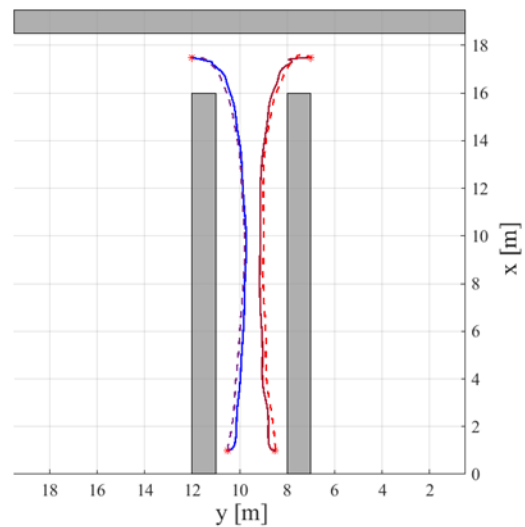


Fig. 6. The top view of the initial trajectory of the two quadrotors affected by wind disturbance



(a)



(b)

Fig. 7. Initial and replanned trajectory of the two quadrotors. (a) Top view, (b) 3D view

In an environment containing multiple obstacles and subjected to a wind disturbance with a magnitude of $[0.2, 0.1, 0]N$, Fig. 8 presents the

initial trajectories of both quadrotors as dashed lines. At each time step, a safe ellipsoid is generated around each quadrotor. The first quadrotor travels from $[2, 12, 0.5]m$ to $[16, 15, 3]m$, while the second moves from $[2, 12, 3]m$ to $[16, 15, 0.5]m$. The zoomed-in section in Fig. 8 highlights how the two safe

ellipsoids approach each other closely, indicating a potential collision risk.

Fig. 9 shows the trajectories after replanning using the proposed approach. The zoomed-in section reveals that the two quadrotors now maintain a larger separation, effectively reducing the risk of collision.

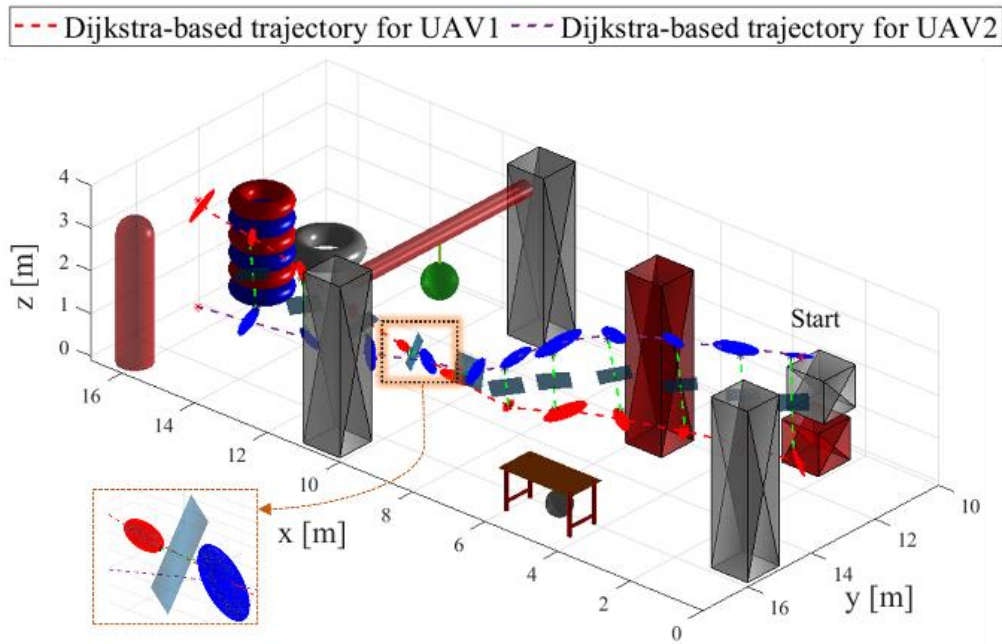


Fig. 8. Initial trajectory of the two quadrotors in the presence of wind disturbance

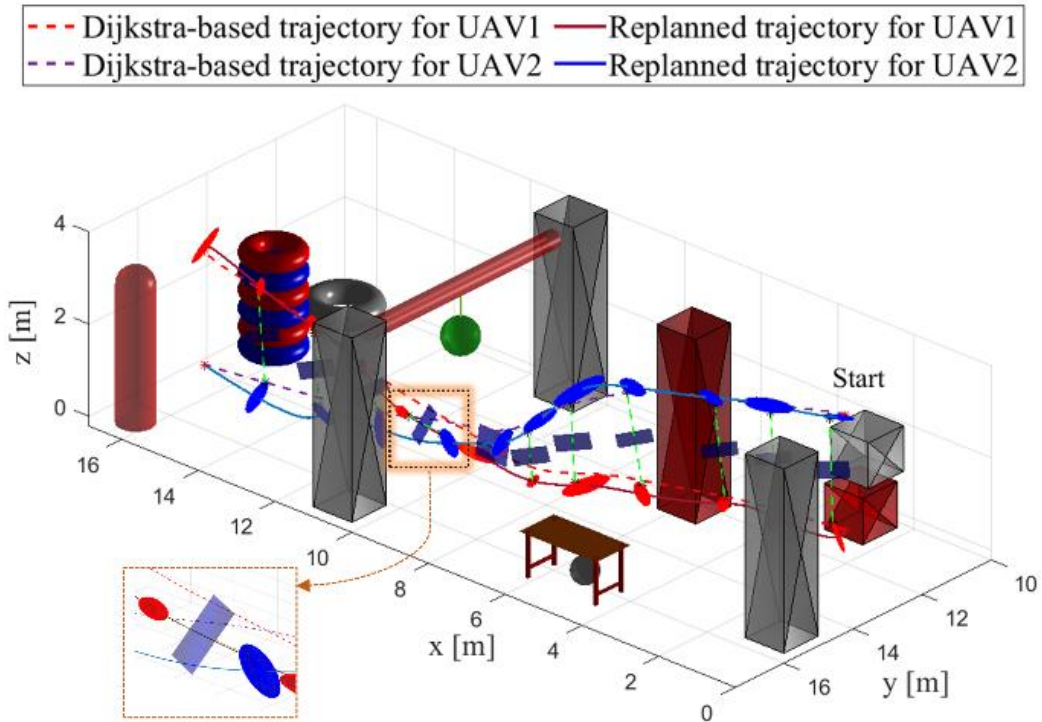


Fig. 9. Initial and replanned trajectory of the two quadrotors under wind disturbance

4.1. Comparison

The proposed approach was evaluated against the NMPC technique, which modifies the trajectory of the quadrotors in response to external disturbances. NMPC determines control inputs at each control step using model-based predictions combined with constrained optimization.

To ensure a fair comparison, both the proposed method and the NMPC technique employed the same system dynamics model, UAV parameters, and actuator constraints. Identical external disturbance profiles and obstacle configurations were applied to both approaches.

The NMPC formulation was solved using Matlab's optimization toolbox, with the prediction horizon and sampling period chosen to guarantee numerical stability. All simulations were executed on the same computer under identical computational settings.

This configuration ensures that the comparison focuses purely on the control performance and computational efficiency of the two algorithms.

To evaluate performance against the proposed method, the NMPC framework was configured with a sampling time of $T_s = 0.21 \text{ sec}$, a prediction horizon of 8 steps, and a control horizon of 2 steps. Custom inequality constraints were incorporated into the NMPC formulation to represent both obstacles and separating hyperplanes.

The sampling time interval is selected based on the system dynamics and the level of accuracy required for trajectory adjustment. In particular, the chosen interval reflects a trade-off between computational efficiency and precision, ensuring a stable and accurate system response within the specified time horizon.

Decreasing the sampling time enhances accuracy and improves the system's stability time, but increases computational load. Conversely, increasing the sampling time reduces the computational burden but may lead to slower system stabilization and less precise trajectory adjustments. Additionally, the processing capability and computational speed of the hardware installed on the quadrotor impose limitations on the achievable sampling rate.

Wind disturbances acting on the quadrotors were modeled as an average force of $[0.1, 0.2, 0] N$, integrated within the quadrotors'

dynamic model, and modulated during simulation according to $[0.1, 0.2, 0] * (1 + 0.2 * \sin t) N$.

Initially, Dijkstra's algorithm was employed to compute a separate path for each quadrotor. The first quadrotor moves from the starting point $[0, 9, 1]m$ to the destination $[10.5, 7, 0]m$, while the second quadrotor travels from $[0, 10.2, 1]m$ to $[10.5, 12, 0]m$. These paths were then used to construct smooth trajectories, depicted by red dashed lines (right side for the first quadrotor, and left side for the second), by solving a quadratic program based on the minimum snap trajectory optimization approach.

The trajectories derived from the Dijkstra paths were adjusted using the proposed approach to account for wind disturbances and the presence of the other quadrotor, producing the blue trajectories. On the other hand, the NMPC method generated the pink trajectories under the same conditions. A summary of the simulation outcomes is presented in Table 1 for comparison.

According to Table 1, the trajectories generated from the Dijkstra-based paths have total lengths of $12.1880 m$ and $12.0751 m$ for the first and second quadrotors, respectively. After replanning with the proposed method, the lengths increased slightly to $12.2694 m$ for the first quadrotor and $12.3809 m$ for the second. In contrast, the NMPC approach produced shorter trajectories measuring $9.3515 m$ and $9.3394 m$ for the first and second quadrotors, respectively.

Despite producing shorter trajectory lengths, the NMPC approach results in trajectories that pass uncomfortably close to obstacles, as illustrated in Fig. 10, particularly in the vicinity of the final obstacle. In contrast, the proposed method consistently preserves safer clearance margins throughout the entire flight. The minimum snap trajectory is formulated by minimizing the integral of the squared snap, defined as the fourth derivative of position, over the motion duration. The objective is to minimize snap values, resulting in highly smooth motion profiles. Additionally, trajectory smoothness can also be evaluated by computing the integral of the squared crackle, corresponding to the fifth derivative of position, over time. Lower values of this metric indicate smoother trajectory profiles.

As reported in Table 1, the proposed approach

achieves minimum snap values of $1.0722 \text{ m}^2/\text{s}^7$ and $0.4689 \text{ m}^2/\text{s}^7$ for the first and second quadrotors, respectively, along with corresponding smoothness values of $0.1027 \text{ m}^2/\text{s}^9$ and $0.0413 \text{ m}^2/\text{s}^9$. In contrast, the NMPC method yields higher snap values of $9.1318 \text{ m}^2/\text{s}^7$ for the first quadrotor and $6.7270 \text{ m}^2/\text{s}^7$ for the second, in addition to increased smoothness values of $2.1982 \text{ m}^2/\text{s}^9$ and $1.5626 \text{ m}^2/\text{s}^9$. These results demonstrate that the proposed method generates significantly smoother and more efficient trajectories compared to the NMPC approach.

In terms of computational efficiency, the proposed method shows a pronounced advantage over NMPC. The total computation time required to generate the trajectories for both quadrotors using the proposed technique was 26.3636 sec , with a replanning time of only 0.0088919 sec . Conversely, the NMPC method required 52.487373 sec for trajectory generation and 0.260695 sec for replanning.

To assess the safety performance of both approaches, the average minimum distance between the quadrotors and the surrounding obstacles was computed.

As summarized in Table 1, the NMPC-based trajectories achieved an average minimum distance of 0.226 m , whereas the proposed method achieved a larger clearance of 0.3005 m .

This outcome indicates that the proposed approach tends to maintain greater clearance from obstacles, reflecting an enhanced level of safety.

Nevertheless, additional experiments with more quadrotors and a wider range of scenarios are required to validate the robustness of this safety improvement under different conditions.

One of the key advantages of the proposed method is its ability to regulate sensitivity to external disturbances. In scenarios characterized by strong wind conditions or complex environmental factors, increasing the system's sensitivity becomes crucial to maintaining a safe clearance around obstacles. Furthermore, in environments containing a large number of obstacles, higher sensitivity enables more effective obstacle avoidance and enhances overall

safety. This adjustable sensitivity provides the system with greater adaptability to varying environmental conditions, resulting in more reliable and secure operation across diverse settings.

As mentioned previously, the parameter K is determined by minimizing the displacement $\|\mathbf{disp}\|_\phi$ in (28) using Matlab's `fmincon` function, subject to the constraints defined in system (29).

The initial value of K , along with its lower and upper bounds and the parameter $T_{max,d}$, directly affect the dynamic coefficient T , as formulated in (34). Consequently, T governs both the extent of trajectory deviation and the system's sensitivity to external disturbances. In particular, larger values of T lead to greater trajectory deviations, which enhances the system's ability to respond to environmental disturbances.

The objective of this study is to compare the trajectory generated by the proposed method for different values of the parameter $T_{max,d}$ (and consequently T) with that obtained using the NMPC approach, to evaluate the system's sensitivity under the same wind disturbance conditions. The configuration shown in Fig. 10 is used as a basis for comparing the influence of $T_{max,d}$ variation, under the same wind disturbance and the same obstacles ($K = 0.111$ and $T_{max,d} = 0.5 [\text{sec}]$ for this particular scene). Increasing the settling time $T_{max,d}$ leads to a longer trajectory and a larger clearance from obstacles, which reflects an increased sensitivity of the system to wind disturbances. On the other hand, decreasing $T_{max,d}$ reduces the overall trajectory length, resulting in trajectories that come closer to obstacles. The corresponding changes in the trajectory are shown in Fig. 11a and Fig. 11b, and summarized in Table 2.

These variations are achieved while maintaining acceptable values for minimum snap, smoothness, and computation time. Moreover, these performance metrics remain favorable in comparison with the NMPC method, as summarized in Tables 1 and 2.

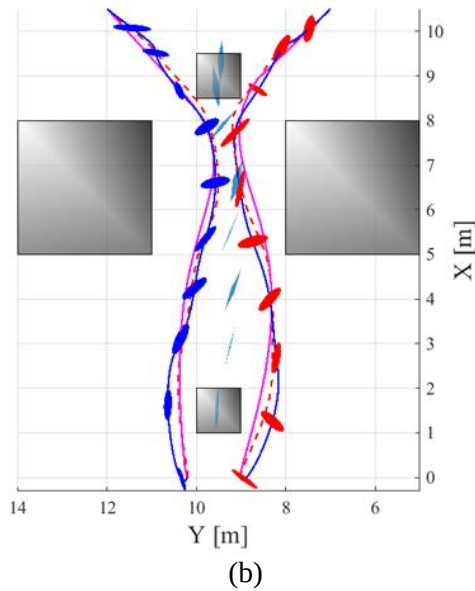
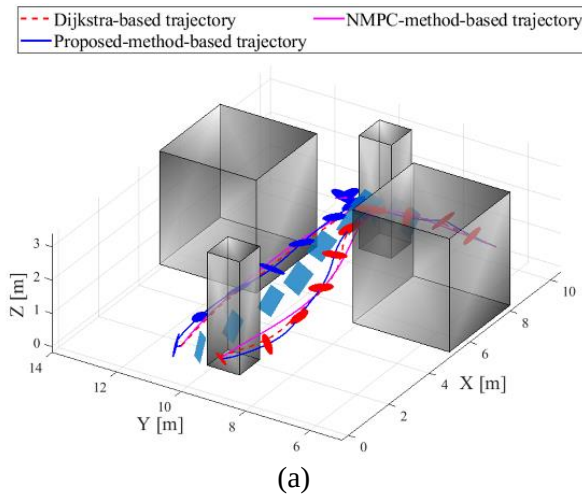


Fig. 10. (a) 3D view, (b) Top view showing the replanning of trajectories derived from the Dijkstra path (red dashed trajectories) using the proposed method (blue trajectories) and the NMPC method (pink trajectories)

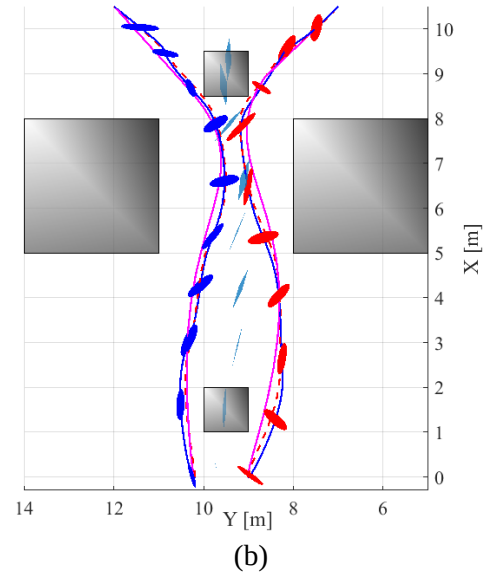
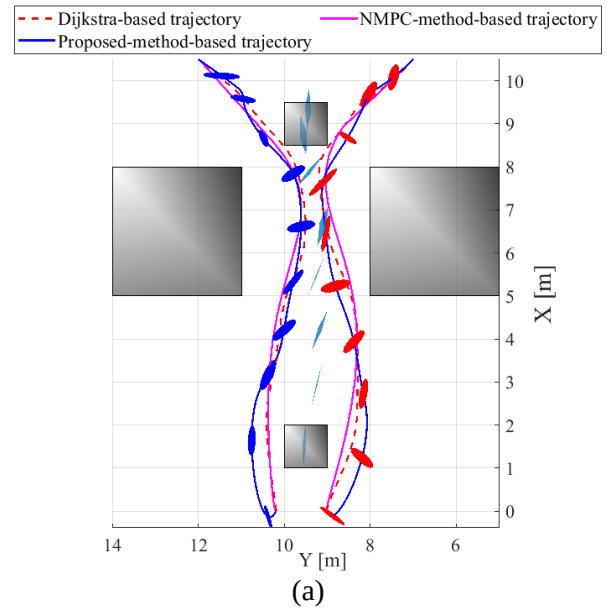


Fig. 11. Top view showing the replanning of trajectories derived from the Dijkstra path (red dashed trajectories) using the NMPC method (pink trajectories). (a) The proposed method (blue trajectories) $T_{max,d} = 0.99[sec]$, (b) The proposed method (blue trajectories) $T_{max,d} = 0.334[sec]$

Table (1): Evaluation of trajectory characteristics and simulation outcomes using the proposed approach versus the NMPC technique

	The initial trajectory resulting from the Dijkstra path		NMPC		Proposed method $T_{max,d} = 0.5[sec]$ $K = 0.111$	
	Uav1	Uav2	Uav1	Uav2	Uav1	Uav2
Total trajectory length [m]	12.1880	12.0751	9.3515	9.3394	12.2694	12.3809
Minimum snap of trajectory [m^2/s^7]	-	-	9.1318	6.7270	1.0722	0.4689
Smoothness of trajectory [m^2/s^9]	-	-	2.1982	1.5626	0.1027	0.0413
Minimum Distance to Obstacles [m]	-	-	0.186	0.266	0.313	0.288
Average Minimum Distance [m]	-	-	0.226		0.3005	
Overall Computation Time for Both Trajectories[sec]	-	-	52.487373		26.3636	
Replanning time [sec]	-	-	0.260695		0.0088919	

Table (2): Evaluating the sensitivity of the proposed method for the same magnitude of wind disturbance $[0.1, 0.2, 0]$ N and for different values of the settling time $T_{max,d}$

	Proposed method			
	$T_{max,d} = 0.99[sec]$ $K = 0.111$		$T_{max,d} = 0.334[sec]$ $K = 0.111$	
	Uav1	Uav2	Uav1	Uav2
Total trajectory length [m]	12.4166	12.6470	12.1842	12.1838
Minimum snap of trajectory [m^2/s^7]	1.0467	0.5848	1.1663	0.4211
Smoothness of trajectory [m^2/s^9]	0.0995	0.0517	0.1124	0.0374
Minimum Distance to Obstacles [m]	0.400	0.365	0.227	0.210
Average Minimum Distance [m]	0.3825		0.2185	
Overall Computation Time for Both Trajectories [sec]	24.4669		27.6348	
Replanning time [sec]	0.008189		0.013051	

Building upon the previous example, a more challenging scenario is investigated to evaluate the robustness of the proposed method. In this scenario, two UAVs operate within the same environment, which contains four obstacles, and are subject to wind disturbances.

Unlike the previous case, the target is not a single fixed point, but rather reaching the green region. The trajectory adjustments resulting from the application of the proposed method to modify the Dijkstra-generated path under wind disturbance are shown in Fig.12a and Fig. 12b.

In Fig. 12a, the target point of the second quadrotor, represented by the blue trajectory, is fixed at $[10.5, 12, 0]$ m, and the Dijkstra trajectory is adjusted to account for wind disturbances while ensuring obstacle avoidance. In contrast, Fig. 12b depicts a slightly different target for the second quadrotor, located at $[10.5, 11, 0]$ m, which represents another feasible point within the green region. A notable advantage of the proposed method is its ability to choose between multiple candidate Dijkstra trajectories based on the severity of the wind disturbance. Specifically, the method evaluates the size of the resulting safe ellipsoid and its proximity to surrounding obstacles before generating the corrected trajectory. By comparing these criteria, the algorithm selects the less risky Dijkstra trajectory for modification, even if this choice results in a longer overall trajectory. This design prioritizes safety, emphasizing collision avoidance and robustness over minimizing the trajectory length.

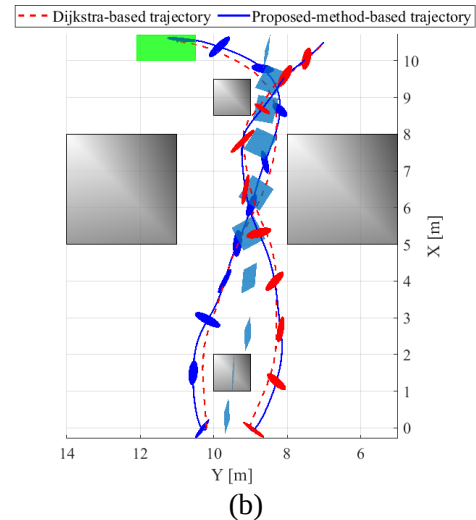
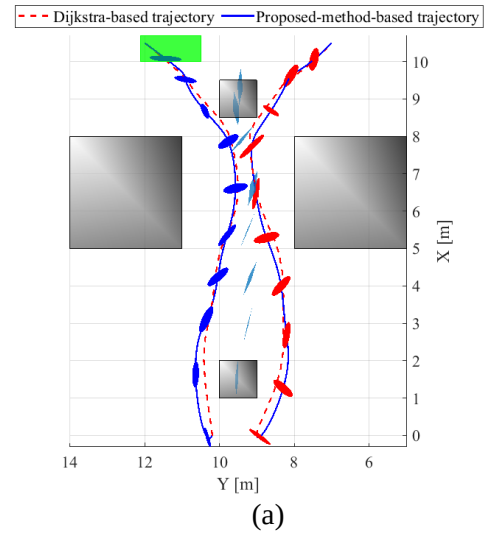


Figure. 12. Wind-aware modification of initial Dijkstra trajectories using the proposed method in the presence of obstacles. (a) Trajectory modification with the target selected from the green feasible region at $[10.5, 12, 0]$ m, (b) Trajectory modification with the target selected from the green feasible region at $[10.5, 11, 0]$ m

While this paper addresses the estimation and replanning of trajectories for two quadrotors simultaneously, several important limitations remain. First, the performance and stability of the system under strong and unstable wind conditions, which can cause significant variations, have not been fully examined. This highlights the need for further research to evaluate and improve the algorithms under more extreme environmental conditions, ensuring safe and reliable operation across a wider range of scenarios.

Second, the operational conditions in highly constrained and complex environments, where the movement space of the quadrotors is extremely limited, the risk of interference or collisions is high, or the number of quadrotors is large, have not been thoroughly investigated. In such scenarios, additional challenges emerge, including more precise collision avoidance, enhanced coordination among the quadrotors, and optimal management of resources and trajectories.

Addressing these challenges requires the development of more advanced algorithms and more accurate modeling techniques, which are essential to enabling broader and safer practical applications of multiple quadrotors in crowded environments.

Finally, it is important to note a limitation of the proposed disturbance estimation approach. In the current model, wind disturbance is represented as the combination of a nominal force estimated by VID-Fusion (or other estimators) and a bounded noise term. However, this estimation is based on the average intensity of the wind rather than its instantaneous value, which may lead to minor inaccuracies when wind conditions change rapidly.

5. Conclusion

Despite notable advancements in motion planning for UAVs, many techniques still face challenges in addressing specific issues. One of the primary concerns is ensuring effective collision avoidance, especially in dynamic and cluttered environments where wind disturbances or interactions with other UAVs may occur.

This paper introduces a systematic framework for replanning the trajectory of two UAVs racing in a restricted space, in the presence of unknown wind forces. While preserving essential features such as obstacle avoidance and safety, the initial UAV trajectories are modified using a virtual force field. This adjustment is based on (i) the safe ellipsoid for each UAV, generated by the Minkowski sum of the FRSs error approximation

due to wind disturbance and the ellipsoid representing their physical dimensions, (ii) the separating hyperplane between the resulting safe ellipsoid of each UAV, (iii) the specifications and dimensions of the obstacles, and (iv) the pre-planned trajectories.

The method is simulated in Matlab, and the outcomes indicate its efficiency in achieving safe trajectory replanning, even in the presence of external disturbances and other UAVs operating in the same environment. Compared to the NMPC method, the proposed strategy exhibited enhanced trajectory smoothness and notably reduced computational time.

Appendix A

Mathematical Background of Hamilton-Jacobi Reachability and Forward Reachable Sets [48, 50]:

Let $l: \mathbb{R}^n \rightarrow \mathbb{R}$ denote a convex function which satisfies $\mathcal{X}_0 = \{y | l(y) \leq 0\}$. The function l is used to check whether a particular state belongs to $\mathcal{X}_0$.

Consider the cost functional $J: \mathbb{R}^n \times \mathbb{R}^m \times [t_0, t_N] \rightarrow \mathbb{R}$ defined as:

$$J(x, \omega, t) = l(x(t_0)) + \int_{t_0}^t L(\omega(\tau)) d\tau \quad (\text{A-1})$$

Here, the function $L: \mathbb{R}^m \rightarrow \mathbb{R}$ represents the running cost for the boundedness of disturbances, ensuring that:

$$L(\omega) = \begin{cases} 0 & \omega \in \mathcal{W}, \\ \infty & \omega \notin \mathcal{W}. \end{cases} \quad (\text{A-2})$$

The value function $V: \mathbb{R}^n \times [t_0, t_N] \rightarrow \mathbb{R}$ is the minimum of the cost over all possible disturbances:

$$V(x, t) = \min_{\omega} J(x, \omega, t) \quad (\text{A-3})$$

The following theorem characterizes a temporal-spatial variation of the value function.

Theorem A.1 Consider a nonlinear dynamics $f: \mathbb{R}^n \times \mathcal{W} \times [t_0, t_N] \rightarrow \mathbb{R}^n$ as $\dot{x}(t) = f(x(t), \omega(t), t)$, where $x(t) \in \mathbb{R}^n$ is state, $\omega(t) \in \mathcal{W} \subset \mathbb{R}^m$ is the unknown disturbance, $t_0 \in \mathbb{R}$ is the initial time, and $t_N > t_0 \in \mathbb{R}$ represents a finite time horizon. n and $m \in \mathbb{R}$ are the dimension of the state and disturbance, respectively. $\mathcal{W}$ is a convex and bounded set of disturbances. Suppose f is Lipschitz in (x, t) and l is Lipschitz in x . For $(x, \tau) \in \mathbb{R}^n \times [t_0, t]$, V in (A-3) is the unique viscosity solution to:

$$\frac{\partial V(x, \tau)}{\partial \tau} + H\left(x, \frac{\partial V(x, \tau)}{\partial x}, \tau\right) = \quad (\text{A-4})$$

$o \text{ in } \mathbb{R}^n \times (t_0, t)$,
 $V(x, \tau) = l(x) \text{ on } \mathbb{R}^n \times \{\tau = t_0\}$,

In this formulation, the Hamiltonian $H: \mathbb{R}^n \times \mathbb{R}^n \times [t_0, t] \rightarrow \mathbb{R}$ is defined as:

$$H(x, \lambda, \tau) := \max_{\omega \in \mathcal{W}} (\lambda \cdot f(x, \lambda, \tau)). \quad (\text{A-5})$$

The reachability problem indicates that there exists a sequence of disturbances capable of driving the state x from the subzero level set of $V(x, t_0)$ to the level set of $V(x, t)$.

In other words, the optimal disturbance sequence $\omega^*(\tau)$ can steer the state x from the zero-level set of $V(x, t_0)$ to the zero-level set of $V(x, t)$. Consequently, the FRS can be characterized in terms of the value function as follows:

$$\mathcal{X}(t) = \{x | V(x, t) \leq 0\} \quad (\text{A-6})$$

Appendix B

Minkowski sum of rotational ellipsoids [51]:

Consider two ellipsoids, A_1 and A_2 . For the i^{th} rotational ellipsoid, the shape matrix $V_i \in \mathbb{R}^{3 \times 3}$ is given by $V_i = U_i S_i U_i^T$, where $S_i = \text{diag}[1/\alpha^2, 1/\beta^2, 1/\beta^2]$, with α and β representing the semi-major and semi-minor axes.

The rotation matrices corresponding to each rotational ellipsoid are expressed as $U_1 = I \in \mathbb{R}^{3 \times 3}$, $U_2 = [[\cos \theta, -\sin \theta, 0], [\sin \theta, \cos \theta, 0], [0, 0, 1]] \in \mathbb{R}^{3 \times 3}$, where θ denotes the rotation angle between the major semi-axes.

The position vector $v_i \in \mathbb{R}^3$ of each rotational ellipsoid is assumed to be the zero vector, as it is not considered in the computation of the Minkowski sum. Therefore, the ellipsoid can be represented by the following equation, as illustrated in Fig. B.1:

$$\begin{aligned} q(v_i, V_i) &= \{x \in \mathbb{R}^3 | (x - v_i)^T V_i (x - v_i) \leq 1\}, \\ &= \{x \in \mathbb{R}^3 | x^T V_i x \leq 1\}. \end{aligned} \quad (\text{B-1})$$

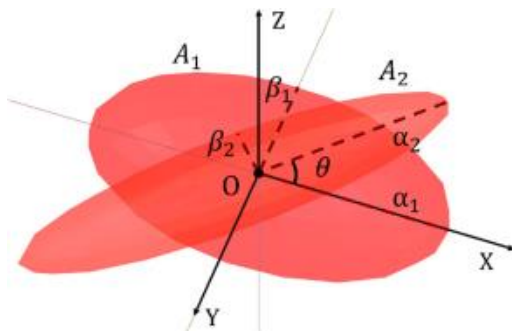


Fig. B.1. Arrangement of the rotational ellipsoids A1 and A2 with the XYZ coordinate system
 The Minkowski sum, considered as the

combination of sets, is defined as the set of all possible vector sums of points in A_1 and A_2 , where A_1 and A_2 are subsets of the three-dimensional Euclidean space $\mathbb{R}^3$ representing the two rotational ellipsoids.

To simplify the concept, we begin by picking an arbitrary element $s_1 \in A_1$ from the set A_1 . The Minkowski sum with the set A_2 is then given by:

$$s_1 + A_2 = s_1 + s_2 \in \mathbb{R}^3 | s_2 \in A_2. \quad (\text{B-2})$$

Considering the above Minkowski sum for every element s_1 in A_1 , we obtain the following expression, which represents the combination of all possible vector sums:

$$\bigcup_{s_1 \in A_1} s_1 + A_2 = \bigcup_{s_1 \in A_1} s_1 + s_2 \in \mathbb{R}^3 | s_2 \in A_2 \quad (\text{B-3})$$

A position vector r , which composes the contour of the Minkowski sum of convex sets, can be expressed as the sum of the position vectors r_1 and r_2 of the two subsets. In this case, the normal vector remains the same. As illustrated in Fig. B.2, if n is the unit normal vector at r , the following equality holds:

$$r \cdot n = r_1 \cdot n + r_2 \cdot n \quad (\text{B-4})$$

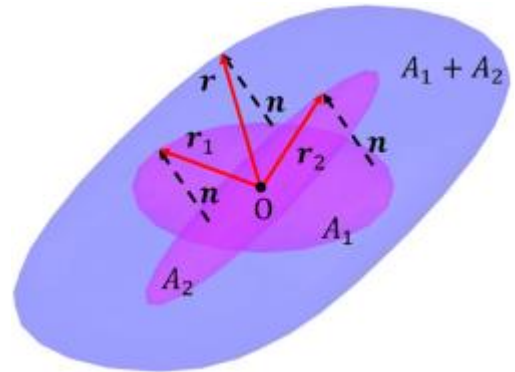


Fig. B.2. The sum of position vectors r_1 and r_2 from ellipsoids A_1 and A_2 from the normal vector n , forming a point on the Minkowski sum.

Since the boundary of a rotational ellipsoid, which is differentiable among closed convex sets, has a distinct unit normal vector at each point, the Minkowski sum is computed by considering all possible unit normal vectors in three-dimensional space.

Acknowledgment

The authors gratefully acknowledge the editorial board and reviewers of the Computational Intelligence in Electrical Engineering Journal for their valuable suggestions, which significantly enhanced the original manuscript.

References

- [1] P. Tang, J. Li, H. Sun, "A review of electric UAV visual detection and navigation technologies for emergency rescue missions", *Sustainability*, Vol. 16, No. 5, 2024. <https://doi.org/10.3390/su16052105>
- [2] F. Betti Sorbelli, "UAV-based delivery systems: A systematic review, current trends, and research challenges", *Journal on Autonomous Transportation Systems*, Vol. 1, No. 3, pp. 1-40, 2024. <https://doi.org/10.1145/3649224>
- [3] X. Wang, C. Demartino, Y. Narazaki, G. Monti, B. F. Spencer Jr, "Rapid seismic risk assessment of bridges using UAV aerial photogrammetry", *Engineering Structures*, Vol. 279, 2023. <https://doi.org/10.1016/j.engstruct.2023.115589>
- [4] S. Moro, F. Linsalata, M. Manzoni, M. Magarini, S. Tebaldini, "Enhancing user localization with an integrated sensing and communication (isac) system: An experimental UAV search-and-rescue use case", *Remote Sensing*, Vol. 16, No. 16, 2024. <https://doi.org/10.3390/rs16163031>
- [5] J. Ahlskog, M.-T. Bahodi, A. Lugmayr, and T. Merritt, "Fostering trust through user interface design in multi-drone search and rescue", in *Proceedings of the Second International Symposium on Trustworthy Autonomous Systems*, pp. 1-11, 2024. <https://doi.org/10.1145/3686038.3686052>
- [6] R. I. Mukhamediev, K. Yakunin, M. Aubakirov, I. Assanov, Y. Kuchin, A. Symagulov, V. Levashenko, E. Zaitseva, D. Sokolov, Y. Amirgaliyev, "Coverage path planning optimization of heterogeneous UAVs group for precision agriculture", *IEEE Access*, Vol. 11, pp. 5789-5803, 2023. <https://doi.org/10.1109/ACCESS.2023.3235207>
- [7] C. Gao, X. Wang, X. Chen, B. M. Chen, "A hierarchical multi-UAV cooperative framework for infrastructure inspection and reconstruction", *Control Theory and Technology*, Vol. 22, No. 3, pp. 394-405, 2024. <https://doi.org/10.1007/s11768-024-00202-0>
- [8] G. Attenni, V. Arrigoni, N. Bartolini, G. Maselli, "Drone-based delivery systems: A survey on route planning", *IEEE Access*, Vol. 11, pp. 123476-123504, 2023. <https://doi.org/10.1109/ACCESS.2023.3329195>
- [9] L. Poudel, S. Elagandula, W. Zhou, Z. Sha, "Decentralized and centralized planning for multi-robot additive manufacturing", *Journal of Mechanical Design*, Vol. 145, No. 1, 2023. <https://doi.org/10.1115/1.4055735>
- [10] L. Wang, W. Huang, H. Li, W. Li, J. Chen, W. Wu, "A review of collaborative trajectory planning for multiple unmanned aerial vehicles", *Processes*, Vol. 12, No. 6, 2024. <https://doi.org/10.3390/pr12061272>
- [11] F. Gul, A. Mir, I. Mir, S. Mir, T. U. Islaam, L. Abualigah, A. Forestiero, "A centralized strategy for multi-agent exploration", *IEEE Access*, Vol. 10, pp. 126871-126884, 2022. <https://doi.org/10.1109/ACCESS.2022.3218653>
- [12] B. Zhou, H. Xu, S. Shen, "Racer: Rapid collaborative exploration with a decentralized multi-UAV system", *IEEE Transactions on Robotics*, Vol. 39, No. 3, pp. 1816-1835, 2023. <https://doi.org/10.1109/TRO.2023.3236945>
- [13] L. Bartolomei, L. Teixeira, M. Chli, "Fast multi-UAV decentralized exploration of forests", *IEEE Robotics and Automation Letters*, Vol. 8, No. 9, pp. 5576-5583, 2023. <https://doi.org/10.1109/LRA.2023.3296037>
- [14] A. Basiri, V. Mariani, G. Silano, M. Aatif, L. Iannelli, L. Glielmo, "A survey on the application of path-planning algorithms for multi-rotor UAVs in precision agriculture", *The Journal of Navigation*, Vol. 75, No. 2, pp. 364-383, 2022. <https://doi.org/10.1017/S0373463321000825>
- [15] J. Gao, Y. Zheng, K. Ni, Q. Mei, B. Hao, L. Zheng, "Fast path planning for firefighting UAV based on A-Star algorithm", in *Journal of Physics: Conference Series*, Vol. 2029, No. 1: IOP Publishing, 2021. <https://doi.org/10.1088/1742-6596/2029/1/012103>
- [16] N. L. Prasad, B. Ramkumar, "3-D deployment and trajectory planning for relay-based-UAV-assisted cooperative communication for emergency scenarios using Dijkstra's algorithm", *IEEE Transactions on Vehicular Technology*, Vol. 72, No. 4, pp. 5049-5063, 2022. <https://doi.org/10.1109/TVT.2022.3224304>
- [17] F. Yang, X. Fang, F. Gao, X. Zhou, H. Li, H. Jin, Y. Song, "Obstacle avoidance path planning for UAV based on improved RRT algorithm", *Discrete Dynamics in Nature and Society*, Vol. 2022, No. 1, 2022. <https://doi.org/10.1155/2022/4544499>
- [18] X. Han, R. Miyazaki, T. Gao, K. Tomita, A.

- Kamimura, "Controller design and disturbance rejection of multi-quadcopters for cable suspended payload transportation using virtual structure", IEEE Access, Vol. 10, pp. 122197-122210, 2022. <https://doi.org/10.1109/ACCESS.2022.3222031>
- [19] S. I. Azid, S. A. Ali, M. Kumar, M. Cirrincione, A. Fagiolini, "Precise trajectory tracking of multi-rotor UAVs using wind disturbance rejection approach", IEEE Access, Vol. 11, pp. 91796-91806, 2023. <https://doi.org/10.1109/ACCESS.2023.3308297>
- [20] Z. A. Ali, H. Zhangang, D. Zhengru, "Path planning of multiple UAVs using MMACO and DE algorithm in dynamic environment", Measurement and Control, Vol. 56, No. 3-4, pp. 459-469, 2023. <https://doi.org/10.1177/0020294020915727>
- [21] M. Huptrych, S. Röck, "Real-time path planning in dynamic environments for unmanned aerial vehicles using the curve-shortening flow method", International Journal of Advanced Robotic Systems, Vol. 18, No. 1, 2021. <https://doi.org/10.1177/1729881420968687>
- [22] N. A. Letizia, B. Salamat, A. M. Tonello, "A novel recursive smooth trajectory generation method for unmanned vehicles", IEEE Transactions on Robotics, Vol. 37, No. 5, pp. 1792-1805, 2021. <https://doi.org/10.1109/TRO.2021.3053649>
- [23] K. Wang, Z. Meng, Z. Wang, Z. Wu, "A trajectory generation method for multi-rotor UAV based on adaptive adjustment strategy", Applied Sciences, Vol. 13, No. 6, 2023. <https://doi.org/10.3390/app13063435>
- [24] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments", IEEE Robotics and Automation Letters, Vol. 2, No. 3, pp. 1688-1695, 2017. <https://doi.org/10.1109/LRA.2017.2663526>
- [25] F. Gao, W. Wu, Y. Lin, S. Shen, "Online safe trajectory generation for quadrotors using fast marching method and Bernstein basis polynomial", in 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 344-351, 2018. <https://doi.org/10.1109/ICRA.2018.8462878>
- [26] B. Sabetghadam, R. Cunha, A. Pascoal, "A distributed algorithm for real-time multi-drone collision-free trajectory replanning", Sensors, Vol. 22, No. 5, 2022. <https://doi.org/10.3390/s22051855>
- [27] K. Motonaka, S. Miyoshi, "Obstacle avoidance using buffered Voronoi cells based on local information from a laser range scanner", Advanced Robotics, Vol. 37, No. 1-2, pp. 73-86, 2023. <https://doi.org/10.1080/01691864.2022.2121616>
- [28] B. Sabetghadam, R. Cunha, A. Pascoal, "Trajectory generation for drones in confined spaces using an ellipsoid model of the body", IEEE Control Systems Letters, Vol. 6, pp. 1022-1027, 2021. <https://doi.org/10.1109/LCSYS.2021.3088406>
- [29] B. Şenbaşlar, W. Hönig, N. Ayanian, "RLSS: Real-time multi-robot trajectory replanning using linear spatial separations", arXiv preprint arXiv:2103.07588, 2021. <https://arxiv.org/abs/2103.07588>
- [30] J. Tordesillas, J. P. How, "MADER: Trajectory planner in multiagent and dynamic environments", IEEE Transactions on Robotics, Vol. 38, No. 1, pp. 463-476, 2021. <https://doi.org/10.1109/TRO.2021.3080235>
- [31] K. Kondo, R. Figueroa, J. Rached, J. Tordesillas, P. C. Lusk, J. P. How, "Robust mader: Decentralized multiagent trajectory planner robust to communication delay in dynamic environments", IEEE Robotics and Automation Letters, Vol. 9, No. 2, pp. 1476-1483, 2023. <https://doi.org/10.1109/LRA.2023.3342561>
- [32] C. E. Luis, M. Vukosavljev, A. P. Schoellig, "Online trajectory generation with distributed model predictive control for multi-robot motion planning", IEEE Robotics and Automation Letters, Vol. 5, No. 2, pp. 604-611, 2020. <https://doi.org/10.1109/LRA.2020.2964159>
- [33] J. Park, D. Kim, G. C. Kim, D. Oh, H. J. Kim, "Online distributed trajectory planning for quadrotor swarm with feasibility guarantee using linear safe corridor", IEEE Robotics and Automation Letters, Vol. 7, No. 2, pp. 4869-4876, 2022. <https://doi.org/10.1109/LRA.2022.3152702>
- [34] M. A. Goma, O. De Silva, G. K. Mann, R. G. Gosine, "Nonlinear MPC without terminal costs or constraints for multi-rotor aerial vehicles", IEEE Control Systems Letters, Vol. 6, pp. 440-445, 2021.

- <https://doi.org/10.1109/LCSYS.2021.3078809>
- [35] T. Kot, R. Wierbica, P. Oščádal, T. Spurný, Z. Bobovský, "Using elastic bands for collision avoidance in collaborative robotics", IEEE Access, Vol. 10, pp. 106972-106987, 2022. <https://doi.org/10.1109/ACCESS.2022.3212407>
- [36] X. Huang, Y. Chen, J. Zhan, "Observer-based consensus control for multi-agent systems with measurement noises and external disturbances", International Journal of Robust and Nonlinear Control, Vol. 32, No. 1, pp. 344-357, 2022. <https://doi.org/10.1002/rnc.5821>
- [37] W. Yu, H. Wang, H. Hong, G. Wen, "Distributed cooperative anti-disturbance control of multi-agent systems: an overview", Science China Information Sciences, Vol. 60, No. 11, 2017. <https://doi.org/10.1007/s11432-017-9141-x>
- [38] Y.-H. Tu, R.-F. Wang, W.-H. Su, "Active disturbance rejection control—New trends in agricultural cybernetics in the future: A comprehensive review", Machines, Vol. 13, No. 2, 2025. <https://doi.org/10.3390/machines13020111>
- [39] H. Marwan, A. J. Humaidi, H. Al-Khazraji, "Active Disturbance Rejection Control for Unmanned Aerial Vehicle", International Journal of Robotics and Control Systems, Vol. 5, No. 2, pp. 1278-1296, 2025. <https://doi.org/10.31763/ijrcs.v5i2.1829>
- [40] V. Mundheda, D. Datta, H. Kandath, "Control barrier function-based predictive control for close proximity operation of UAVs inside a tunnel", in 2025 International Conference on Unmanned Aircraft Systems (ICUAS), IEEE, pp. 975-981, 2025. <https://doi.org/10.1109/ICUAS65942.2025.11007781>
- [41] W. Zhao, B. Bonakdarpour, "Decentralized multi-UAV routing in the presence of disturbances", arXiv preprint arXiv:1807.04823, 2018. <https://doi.org/10.48550/arXiv.1807.04823>
- [42] Y. Wu, Z. Ding, C. Xu, F. Gao, "External forces resilient safe motion planning for quadrotor", IEEE Robotics and Automation Letters, Vol. 6, No. 4, pp. 8506-8513, 2021. <https://doi.org/10.1109/LRA.2021.3110316>
- [43] Z. Ding, T. Yang, K. Zhang, C. Xu, F. Gao, "Vid-fusion: Robust visual-inertial-dynamics odometry for accurate external force estimation", in 2021 IEEE International Conference on Robotics and Automation (ICRA), IEEE, pp. 14469-14475, 2021. <https://doi.org/10.1109/ICRA48506.2021.9560898>
- [44] Y. Kovryzhenko, N. Li, E. Taheri, "A Control System Design and Implementation for Autonomous Quadrotors with Real-Time Re-Planning Capability", Robotics, Vol. 13, No. 9, 2024. <https://doi.org/10.3390/robotics13090136>
- [45] X. Zhang, J. Lu, Y. Hui, H. Shen, L. Xu, B. Tian, "Rapa-planner: Robust and efficient motion planning for quadrotors based on parallel ra-mppi", IEEE Transactions on Industrial Electronics, 2024. <https://doi.org/10.1109/TIE.2024.3508056>
- [46] S. Liu, K. Guo, X. Yu, "A Planning and Control Architecture for Resilient Flights Under External Disturbances", IEEE Transactions on Aerospace and Electronic Systems, 2025. <https://doi.org/10.1109/TAES.2025.3535856>
- [47] D. Mellinger, V. Kumar, "Minimum snap trajectory generation and control for quadrotors", in 2011 IEEE International Conference on Robotics and Automation, IEEE, pp. 2520-2525, 2011. <https://doi.org/10.1109/ICRA.2011.5980409>
- [48] H. Seo, D. Lee, C. Y. Son, C. J. Tomlin, H. J. Kim, "Robust trajectory planning for a multirotor against disturbance based on Hamilton-Jacobi reachability analysis", in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, pp. 3150-3157, 2019. <https://doi.org/10.1109/IROS40897.2019.8968126>
- [49] W. Lee, "Tridiagonal matrices: Thomas algorithm", MS6021, Scientific Computation, University of Limerick, Vol. 1054, 2011. https://sites.nd.edu/gfu/files/2019/10/ms6021_thomas.pdf
- [50] H.-s. Seo, "Real-time Safety-guaranteed Trajectory Planning using Hamilton-Jacobi Reachability Analysis", Seoul National University, Graduate School, 2021. <https://space.snu.ac.kr/handle/10371/175172>
- [51] R. Yamada, T. Tsuji, T. Hiramitsu, H. Seki, T. Nishimura, Y. Suzuki, T. Watanabe, "Fast and precise approximation of Minkowski sum of two rotational ellipsoids with a superellipsoid", The Visual Computer, Vol. 40, No. 7, pp. 4609-4621, 2024. <http://doi.org/10.1007/s00371-024-03445-9>

