drones

MDPI

Article

HEA-Planner: A Hierarchical Environment-Aware Trajectory Planning Framework for Autonomous UAV Landing on Moving Platforms in Confined Dynamic Warehouse Environments

Xing Wu [1,*], Fengyang Qiu [1], Junzhe Chen [1], Peihuang Lou [1], Sheng Dai [2] and Zhu Duan [3]

[1] College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; qfy123@nuaa.edu.cn (F.Q.); chenjunzhe@nuaa.edu.cn (J.C.); meephlou@nuaa.edu.cn (P.L.)
[2] Nanjing LES Information Technology Co., Ltd., Nanjing 210014, China; andydai83@126.com
[3] School of Traffic and Transportation Engineering, Central South University, Changsha 410075, China; duanzhu@csu.edu.cn
* Correspondence: wustar5353@nuaa.edu.cn

Highlights

What are the main findings?

- A hierarchical environment-aware trajectory planning framework (HEA-Planner) is proposed, integrating EA-ACO global planning, EA-A* local replanning, and SG-MINCO/STC-MINCO trajectory optimization for the rapid descent phase of autonomous quadrotor landing on moving platforms in confined warehouse environments.
- Simulations across the three operational states verify the performance advantages of the framework, while physical flight experiments validate the perception–planning–control closed-loop capability and real-time execution on the onboard computer.

What are the implications of the main findings?

- By maintaining obstacle clearance in narrow corridors and bounding handover position error within 200 mm, HEA-Planner provides a practical trajectory planner for the rapid descent phase of two-stage landing systems in warehouse scenarios.
- The hierarchical decoupling of spatial obstacle clearance and temporal state synchronization provides an effective framework for autonomous landing in confined dynamic environments.

Abstract

Autonomous quadrotor landing on moving ground platforms is essential for cooperative air–ground operations in confined warehouse environments. However, existing trajectory planners often compromise corridor clearance in narrow aisles or fail to track maneuvering platforms under dynamic intrusions. To address these challenges, a hierarchical environment-aware trajectory planning framework (HEA-Planner) is presented. At the global layer, an Environment-Aware Ant Colony Optimization (EA-ACO) algorithm is formulated to generate corridor-centered reference paths using environmental feedback. At the local layer, an Environment-Aware A* (EA-A*) replanner is designed to coordinate reactive obstacle avoidance and platform interception via spatiotemporal risk repulsion and predictive attraction fields. At the trajectory optimization layer, a Spatially-Guided MINCO (SG-MINCO) solver is introduced to preserve corridor clearance, while a Spatiotemporal Coupled MINCO (STC-MINCO) solver is derived to resolve moving-platform rendezvous through analytical duration total derivatives. Simulations across three operational states demonstrate robust collision avoidance and dynamic synchronization.

Check for updates

Academic Editor: Weiran Yao

Received: 11 August 2026
Revised: 8 October 2026
Accepted: 10 October 2026
Published: 16 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Furthermore, physical flight experiments validate onboard closed-loop execution, achieving an average replanning latency of 33.6 ± 4.9 ms and a terminal touchdown error of 148.2 ± 37.3 mm across all trials (100% success rate), verifying the real-time feasibility of autonomous air–ground handover in restricted logistics workspaces.

Keywords: unmanned aerial vehicles; unmanned ground vehicles; UAV–UGV cooperative systems; moving-platform landing; hierarchical trajectory planning; confined dynamic environments

1. Introduction

In modern automated warehouses, air–ground collaboration between unmanned aerial vehicles (UAVs) and unmanned ground vehicles (UGVs) offers a flexible paradigm to enhance operational efficiency [1–3]. In these heterogeneous teams, aerial vehicles leverage three-dimensional mobility for high-shelf inventory inspection [4], while ground platforms handle material handling and mobile charging support [5]. The closed-loop synergy of this air–ground system hinges on the UAV's capability to land autonomously and precisely on moving platforms [6]. However, confined warehouse environments present strict constraints: narrow aisles with dense shelf clutter restrict flight clearance, while moving platforms demand tight spatiotemporal coordination. Concurrently, UGVs must traverse dynamic workspaces while converging toward a time-varying rendezvous state, posing severe challenges to real-time trajectory planning [7].

Autonomous UAV landing on moving platforms can be categorized into cooperative schemes, where platform motion is jointly planned [8], and unilateral tracking schemes, where the UAV must adapt autonomously using onboard vision [9]. In warehouse logistics, ground platforms typically execute independent tasks, necessitating unilateral tracking by the aerial vehicle. In this unilateral context, three primary paradigms have been established: pure visual servoing [10], single-step predictive interception [11], and staged landing [12]. In confined warehouse environments, pure visual servoing is vulnerable to line-of-sight (LoS) occlusions caused by high-shelving structures, whereas single-step predictive interception lacks robustness against dynamic uncertainties due to the absence of continuous feedback loops. Consequently, staged landing has become the dominant paradigm, which decouples the landing process into a rapid descent phase and a precision landing phase. In the rapid descent phase, the quadrotor approaches the moving platform from a distance, whereas in the precision landing phase, it aligns with the deck for final touchdown [13].

Beyond the multi-stage landing structure discussed above, existing studies are further categorized by how descent motion is generated, leading to control-based and planning-based technical routes. The control-based route closes the feedback loop directly from perception to actuation. Here, visual servoing tracks the platform via image feedback; model predictive control (MPC) [14,15] incorporates vehicle dynamics and operational constraints via receding-horizon optimization; and reinforcement learning (RL) [16–18] learns end-to-end control policies, showing adaptability under wind disturbances and sensing noise. Most control-based methods treat the landing maneuver as a single-stage process without isolating the descent phase. When evaluated against descent-phase requirements in confined dynamic warehouses, the control-based route presents clear limitations. Visual servoing lacks global spatial reasoning and degrades under LoS occlusions. MPC incurs heavy computational latency that scales with prediction horizon and clutter density, alongside sensitivity to model mismatch. Meanwhile, reinforcement learning optimizes expected cumulative rewards rather than explicit geometric boundaries, making hard corridor containment challenging. Conversely, planning-based techniques construct collision-free

trajectories in state space prior to execution. Hierarchical frameworks [19,20] decouple frontend spatial search from backend continuous optimization: the frontend generates discrete collision-free corridors, while the backend optimizes a smooth, dynamically feasible continuous trajectory for state synchronization. These approaches are complementary in staged landing: the trajectory planner generates spatiotemporal reference states for lower-level controllers, while terminal visual servoing executes physical touchdown. To contextualize the proposed framework, existing trajectory planning literature is reviewed across three aspects: global path planning, local dynamic replanning, and backend trajectory optimization.

Global path planning constructs the initial collision-free path for rapid descent in a static environment [21,22]. Standard graph search methods such as A* incur high computational costs when heuristic guidance degrades in large maps, while sampling-based algorithms (e.g., Rapidly-exploring Random Tree Star, RRT*) yield geometrically unsmoothed paths of suboptimal lengths [23]. Bio-inspired planners such as Ant Colony Optimization (ACO) and Genetic Algorithm (GA) [24] exhibit global search capabilities but suffer from slow convergence, parameter sensitivity, and susceptibility to local optima [25]. When applied to narrow warehouse corridors, conventional bio-inspired planners greedily minimize path length, causing planned paths to closely hug shelf boundaries without safety margins and increasing collision risk.

Within a receding-horizon framework, local dynamic replanning primarily serves to track the moving platform and react to dynamic obstacles encountered during flight. Bounding the search within a sliding window allows discrete algorithms such as A* and Dijkstra to be applied for local replanning [26], yet the resulting paths are time-unparameterized and require backend smoothing. Furthermore, geometric distance minimization causes paths to hug obstacle boundaries, leaving narrow margins for evasive maneuvers. Reactive guidance methods, such as artificial potential fields (APFs), the Dynamic Window Approach (DWA) [27], and vector-field guidance [28], translate sensor feedback directly into steering commands. However, their purely reactive nature can cause deadlocks in cluttered corridors, and the absence of terminal time constraints prevents spatiotemporal state synchronization with moving targets. Deep reinforcement learning (DRL) approaches [29,30] adapt well to dynamic environments, yet standard model-free policies lack explicit dynamic-feasibility guarantees and cannot consistently ensure exact terminal velocity matching on moving decks.

At the backend of the hierarchy, trajectory optimization transforms discrete paths into time-parameterized, dynamically feasible trajectories [31]. Duration allocation is a defining factor distinguishing mainstream trajectory parameterizations. Minimum Snap [32] prescribes segment durations heuristically prior to optimization. Meanwhile, B-spline and Bézier parameterizations [33,34] embed duration into basis functions, where duration adjustments cause conflicting gradients and topological sensitivity in narrow corridors. When terminal boundary conditions depend on arrival time, as in moving-platform landing, such parameterizations employ a predict-and-fix scheme: target motion is extrapolated and arrival time is frozen before optimization, decoupling duration allocation from spatial cost and causing interception errors to accumulate during target maneuvers. In contrast, the Minimum Control Effort (MINCO) parameterization [35] treats intermediate waypoints and segment durations as independent variables with analytical gradients, enabling joint spatial and temporal optimization. However, standard MINCO formulations assume static terminal boundaries, adjusting duration as if the target were stationary, and thus fail to achieve spatiotemporal state synchronization with maneuvering platforms.

This work addresses autonomous quadrotor landing on moving platforms in confined dynamic warehouse environments, focusing on two interrelated technical challenges. First, safe navigation in confined dynamic spaces requires maintaining obstacle clearance

in narrow corridors while executing real-time evasion of dynamic intrusions. Second, spatiotemporal state synchronization requires terminal matching of position, velocity, and arrival time with the maneuvering platform. However, existing methods address these challenges only partially. Control-based schemes [10] track terminal states without global obstacle reasoning under line-of-sight occlusions. End-to-end reinforcement learning policies [18] adapt to dynamic intrusions, yet regulate spatial positions rather than continuous terminal velocity vectors needed for smooth touchdown. Discrete global–local planners [27] achieve obstacle avoidance but yield time-unindexed paths lacking moving-target synchronization. Finally, optimization pipelines such as EGO-Planner [34] freeze arrival time prior to backend optimization, decoupling temporal allocation from spatial clearance and degrading tracking performance during target maneuvers.

To overcome these limitations, a hierarchical environment-aware trajectory planning framework (HEA-Planner) is proposed for autonomous quadrotor landing on moving platforms in confined dynamic warehouse environments. The primary contributions are threefold:

1. System integration: An FSM-supervised hierarchical architecture is designed for staged landing in confined dynamic warehouses. The framework coordinates Global Approaching (S_1), Dynamic Avoiding (S_2), and Target Synchronizing (S_3) to bring the quadrotor safely into the visual handover window.
2. Algorithmic adaptations: Discrete search and continuous trajectory optimization are adapted to narrow warehouse corridors. Specifically, obstacle clearance and turning costs are incorporated into EA-ACO pheromone updates to prevent wall-hugging; a spatiotemporal risk repulsion field and a predictive attraction field are integrated into EA-A* for dynamic evasion; and corridor bounding-box mappings with a waypoint guidance penalty are introduced into SG-MINCO to prevent corner-cutting at sharp turns.
3. Theoretical derivation: The Spatiotemporal Coupled MINCO (STC-MINCO) framework is formulated for Target Synchronizing (State S_3). Under moving terminal boundary conditions, analytical total derivatives of the trajectory cost with respect to segment durations are rigorously derived, enabling joint gradient-based optimization of spatial trajectory shape and flight duration.

The rest of this paper is organized as follows. Section 2 gives the problem statement and preliminary definitions. Section 3 describes the proposed HEA-Planner framework. Section 4 presents simulation and experimental results. Section 5 concludes the paper.

2. Problem Scenario and Preliminaries

2.1. Problem Overview and Mission Profile

In automated warehouses, heterogeneous UAVs and UGVs collaborate to execute warehouse logistics tasks. As illustrated in Figure 1a, the quadrotor UAV leverages its three-dimensional mobility for high-shelf inventory inspection [36], while the cooperative UGV platform executes ground-level material transport and serves as a mobile landing platform. Upon completing the inspection mission, the quadrotor is required to return and land on the moving platform, necessitating autonomous trajectory planning through confined spaces for battery recharging or mission handover.

To accomplish this landing mission, a two-stage strategy is adopted, as illustrated in Figure 1b. First, in the rapid descent phase, a continuous trajectory is generated to guide the quadrotor from an initial standoff distance toward the moving platform while achieving spatiotemporal state synchronization with platform motion. Second, a precision landing phase employs visual servoing for terminal touchdown [13]. This work focuses on the

rapid descent phase, where HEA-Planner generates collision-free trajectories to achieve spatiotemporal state synchronization, guiding the quadrotor into the visual handover window prior to the precision landing phase.

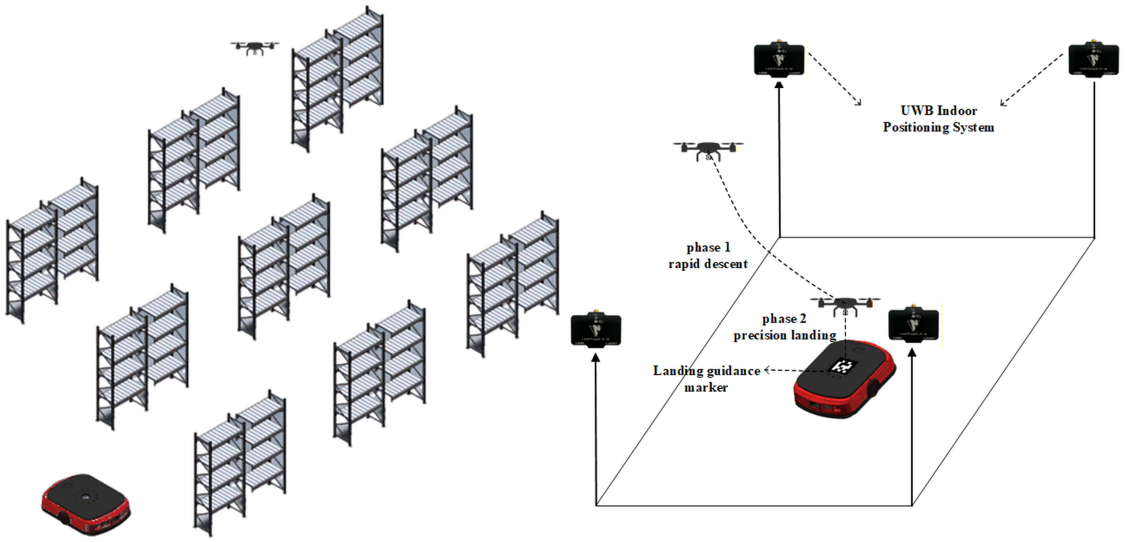


Figure 1. Task scenario and two-stage landing profile. (**a**) Inspection environment with shelves, narrow corridors, a quadrotor UAV, and a cooperative UGV platform. (**b**) Two-stage landing profile consisting of the rapid descent phase and the precision landing phase.

2.2. Coordinate Systems and Map Representation

To describe relative motion during the rapid descent phase, three coordinate frames are established: the inertial world frame $\mathcal{W}$ with its Z-axis pointing vertically upward, the UAV body-fixed frame centered at the quadrotor center of mass with its X-axis aligned with the forward heading, and the moving-platform frame $\mathcal{G}$ centered on the cooperative UGV platform.

The warehouse workspace $\Omega \subset \mathbb{R}^3$ is discretized into a 3D voxel grid map. The obstacle distribution comprises the static obstacle set Ω_{static} and the time-varying dynamic obstacle set $\Omega_{\text{dynamic}}(t)$. Incorporating the UAV footprint radius $R_{\text{footprint}}$, the collision-free navigation space $\Omega_{\text{free}}(t)$ is formulated as :

$$\Omega_{\text{free}}(t) = \{\mathbf{p} \in \Omega \mid \text{dist}(\mathbf{p}, \Omega_{\text{static}} \cup \Omega_{\text{dynamic}}(t)) \geq R_{\text{footprint}}\} \tag{1}$$

Path planning and safety corridor generation are performed within $\Omega_{\text{free}}(t)$.

2.3. Quadrotor Dynamics and Differential Flatness

The quadrotor operates as an underactuated, nonlinear dynamical system. The state vector is defined as $\mathbf{X} = [\mathbf{p}^T, \mathbf{v}^T, \mathbf{R}]^T$, where $\mathbf{p} = [x, y, z]^T \in \mathbb{R}^3$ is the position, $\mathbf{v} \in \mathbb{R}^3$ is the velocity, and $\mathbf{R} \in SO(3)$ is the rotation matrix. The control input vector is $\mathbf{U} = [f_T, \boldsymbol{\tau}^T]^T$, where $f_T \in \mathbb{R}$ is the total thrust and $\boldsymbol{\tau} \in \mathbb{R}^3$ denotes the control torque.

By defining the flat output vector as $\boldsymbol{\sigma} = [\mathbf{p}^T, \psi]^T$ (where ψ denotes the yaw angle), the state and control input trajectories are uniquely determined via algebraic mappings:

$$\mathbf{X}(t) = \Phi_x(\boldsymbol{\sigma}, \dot{\boldsymbol{\sigma}}, \ddot{\boldsymbol{\sigma}}, \dddot{\boldsymbol{\sigma}}), \quad \mathbf{U}(t) = \Phi_u(\boldsymbol{\sigma}, \dot{\boldsymbol{\sigma}}, \ddot{\boldsymbol{\sigma}}, \dddot{\boldsymbol{\sigma}}, \ddddot{\boldsymbol{\sigma}}) \tag{2}$$

Through differential flatness, smooth flat output trajectories algebraically map to collective motor thrust and body angular rates without mathematical singularities [32].

The translational position trajectory $\mathbf{p}(t)$ and yaw angle $\psi(t)$ are differentially decoupled in flight dynamics. Specifically, $\mathbf{p}(t)$ dictates the required thrust magnitude and vehicle attitude (roll and pitch), whereas $\psi(t)$ independently governs the heading orientation. Consequently, trajectory optimization focuses on the spatial curve $\mathbf{p}(t)$, with the yaw profile assigned downstream during trajectory execution.

To account for vehicle physical capabilities, trajectory optimization is constrained by kinematic and actuator bounds. First, linear velocity and acceleration magnitudes are bounded by:

$$\|\dot{\mathbf{p}}(t)\| \leq v_{\max}, \quad \|\ddot{\mathbf{p}}(t)\| \leq a_{\max} \tag{3}$$

The total thrust vector is given by $\mathbf{f}(t) = m(\ddot{\mathbf{p}}(t) + g\mathbf{e}_z)$, where m denotes the vehicle mass, g is the gravitational acceleration, and $\mathbf{e}_z = [0, 0, 1]^T$ is the unit vertical vector. The collective thrust capability is bounded by:

$$\|\ddot{\mathbf{p}}(t) + g\mathbf{e}_z\| \leq \frac{f_{\max}}{m} \tag{4}$$

The tilt angle $\theta(t)$ between the thrust vector and $\mathbf{e}_z$ satisfies:

$$\cos\theta(t) = \frac{\mathbf{f}(t) \cdot \mathbf{e}_z}{\|\mathbf{f}(t)\|} = \frac{\ddot{z}(t) + g}{\|\ddot{\mathbf{p}}(t) + g\mathbf{e}_z\|} \tag{5}$$

To maintain aerodynamic stability and avoid extreme attitude excursions, the body tilt angle is bounded by a maximum threshold $\theta_{\max} < \pi/2$, yielding the cone constraint:

$$\ddot{z}(t) + g \geq \|\ddot{\mathbf{p}}(t) + g\mathbf{e}_z\| \cos\theta_{\max} \tag{6}$$

By virtue of differential flatness, the cone constraint in (6) algebraically restricts vehicle attitude $\mathbf{R}(t)$ within a bounded cone around the vertical axis, suppressing large tilt excursions. Physical feasibility is addressed across three complementary tiers: (i) in the optimizer, linear velocity (3), collective thrust (4), and tilt angle (6) are enforced on flat outputs; (ii) post-optimization, body angular rates $\boldsymbol{\omega}$ and single-rotor thrusts f_i ($i \in \{1, \ldots, 4\}$) are algebraically evaluated across the trajectory via differential flatness to verify $\|\boldsymbol{\omega}\| \leq \omega_{\max}$ and $f_{\min} \leq f_i \leq f_{\max}$, rejecting candidate trajectories that violate dynamic limits prior to execution; and (iii) at the tracking level, the onboard autopilot executes closed-loop attitude and angular rate tracking, with low-level mixer saturation management to handle physical motor limits.

3. Hierarchical Trajectory Planning Framework

A three-level hierarchical trajectory planning framework (HEA-Planner) is proposed for autonomous landing in confined dynamic warehouses: global path planning via EA-ACO, local dynamic replanning via EA-A*, and spatiotemporal trajectory optimization via SG-MINCO and STC-MINCO.

3.1. Framework Overview and FSM Supervision

Hierarchical trajectory planning with a Finite State Machine (FSM) has been widely adopted in aerial robotics to separate discrete task logic from continuous trajectory generation [19,20]. In HEA-Planner, the FSM acts as a supervisor to regulate the execution of appropriate planning stages based on real-time data. As illustrated in Figure 2, the landing operation in HEA-Planner is represented by three operational states:

1. State S_1 (Global Approaching): In a static environment or during the long-distance approaching phase, offline EA-ACO global planning generates a collision-free path. Next, the SG-MINCO backend generates a continuous trajectory according to the

reference path while respecting safe flight corridors (SFCs) constructed from Axis-Aligned Bounding Boxes (AABBs).

2. State S_2 (Dynamic Avoiding): Upon detecting dynamic obstacles, EA-A* replanning is applied to generate reactive detour paths in a local sliding window. The SG-MINCO optimizer recalculates detour trajectories online to maintain obstacle clearance.
3. State S_3 (Target Synchronizing): Once the quadrotor reaches the terminal landing area, the EA-A* module employs the predictive attraction field to track the moving platform. Then the STC-MINCO backend adjusts both the trajectory shape and its duration to achieve spatiotemporal state synchronization with the moving platform.

Transitions between states in the FSM are conditioned by spatial and temporal thresholds. Collision avoidance takes highest priority. When an incoming dynamic obstacle breaches both the temporal threshold $t_{\mathrm{CPA}} < \tau_{\mathrm{safe}}$ (time of closest approach) and the spatial clearance margin $d_{\mathrm{CPA}} < R_{\mathrm{coll}}$ (the finite-body collision test under constant relative velocity), the system transitions to State S_2 to plan an evasive detour. Otherwise, when there is no collision risk, the operation switches to State S_3 once the relative distance d_{rel} to the moving platform becomes smaller than D_{sync}. After clearing dynamic obstacles, the UAV switches to State S_3 if the moving platform stays in the terminal zone, or back to State S_1 to resume tracking the reference path. To suppress state chattering near threshold boundaries caused by sensor noise, positive hysteresis margins $\Delta\tau_{\mathrm{hys}}$ and ΔD_{hys} are used in transition conditions: State S_2 holds until $t_{\mathrm{CPA}} \geq \tau_{\mathrm{safe}} + \Delta\tau_{\mathrm{hys}}$ or $d_{\mathrm{CPA}} \geq R_{\mathrm{coll}}$, and State S_3 holds until $d_{\mathrm{rel}} > D_{\mathrm{sync}} + \Delta D_{\mathrm{hys}}$. The online planning procedure operates periodically with period Δt at current replanning timestamp t_c, and stitches new trajectories to the current one in accordance with the C^2 continuity condition in Equation (24b).

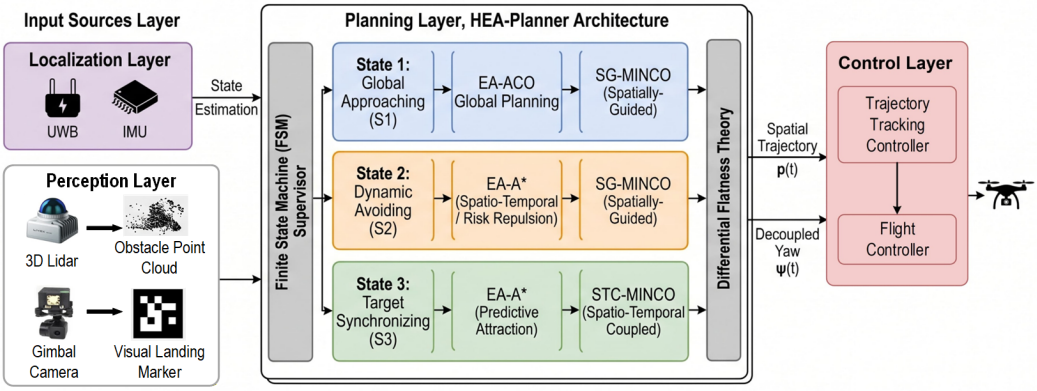


Figure 2. HEA-Planner architecture and layer interaction.

The complete online mission execution workflow is formalized in Algorithm 1.

Two operational characteristics of Algorithm 1 are notable. First, mission initialization is performed offline and only once: since the EA-ACO global search operates over the entire voxel map with a computation time of tens of seconds, it is fully decoupled from the online loop, within which only the EA-A* local replanner and the backend optimizers are invoked periodically with period Δt, thereby sustaining high-frequency receding-horizon execution. Second, consecutive replanning cycles are tightly coupled through three inherited quantities: the stitch state $\mathbf{s}_{\mathrm{stitch}}$ sampled from the committed trajectory (Equation (24b)), the previous velocity profile $\mathbf{v}_{\mathrm{prev}}$ for arrival-time calibration (Equation (16)), and the remaining flight duration $T_{\Sigma,\mathrm{prev}}$ that guides the predictive attraction heuristic (Equation (23)). Upon returning to S_1 from S_2 or S_3, the backend re-optimizes the remaining global reference path from $\mathbf{s}_{\mathrm{stitch}}$ via SG-MINCO, allowing the UAV to smoothly re-engage the global trajectory. If a replanning cycle fails to produce a valid trajectory—for instance, when node expansion reaches $N_{\max}$ or optimization fails to converge—HEA-Planner applies a two-tier safety

fallback. Under Tier 1, the supervisor checks whether the committed trajectory maintains forward clearance ($d_{\min} \geq R_{\text{coll}} = R_{\text{footprint}} + R_{\text{obs}}$). If so, the UAV continues along this path for the current interval Δt and retries replanning in the next cycle. Under Tier 2, if the committed trajectory risks breaching the safety margin ($d_{\min} < R_{\text{coll}}$), the system executes emergency braking with maximum deceleration $\mathbf{a} = -a_{\max}\frac{\mathbf{v}}{\|\mathbf{v}\|}$. This brings the UAV to a hover in place within a bounded distance $s_{\text{stop}} \leq \frac{\|\mathbf{v}\|^2}{2a_{\max}}$ until a viable path is re-established.

Algorithm 1 HEA-Planner Online Mission Execution (FSM-Supervised)

Input: Static map Ω_{static}, start $\mathbf{p}_s$, landing zone $\mathbf{p}_{\text{goal}}$, thresholds D_{sync}, τ_{safe}, hysteresis margins $\Delta\tau_{\text{hys}}$, ΔD_{hys}
Output: Committed trajectory sequence converging to the visual handover window
// Offline initialization (once per mission)
$\mathcal{P}_g \leftarrow$ EA-ACO$(\mathbf{p}_s, \mathbf{p}_{\text{goal}}, \Omega_{\text{static}})$ ▷ Section 3.2
$\mathcal{F}_g \leftarrow$ AABB-SFC$(\mathcal{P}_g)$; $\mathbf{p}(t) \leftarrow$ SG-MINCO$(\mathcal{P}_g, \mathcal{F}_g, \mathbf{s}_{\text{init}})$ ▷ Section 3.4.2
$S \leftarrow S_1$, $S_{\text{prev}} \leftarrow S_1$; cache $\mathbf{v}_{\text{prev}} \leftarrow \dot{\mathbf{p}}(t)$, $T_{\Sigma,\text{prev}} \leftarrow T_\Sigma$
// Online receding-horizon loop (period Δt)
while visual handover window not reached **do**
  $\{(\mathbf{p}_{\text{obs},i}, \mathbf{v}_{\text{obs},i})\} \leftarrow$ Obstacle cluster tracking; $\hat{\mathbf{s}}_{\text{target}} \leftarrow \text{EKF}(\mathbf{z}_G)$ ▷ Section 3.3.2
  $d_{\text{rel}} \leftarrow \|\mathbf{p} - \hat{\mathbf{p}}_{\text{target}}\|$; threat $\leftarrow \exists i$, $(t_{\text{CPA},i} < \tau_{\text{safe}}$ and $d_{\text{CPA},i} < R_{\text{coll}})$ ▷ Equations (18) and (19)
  if threat **then**
    $S \leftarrow S_2$
  else if $S = S_2$ **and** $\exists i$, $(t_{\text{CPA},i} < \tau_{\text{safe}} + \Delta\tau_{\text{hys}}$ and $d_{\text{CPA},i} < R_{\text{coll}})$ **then**
    $S \leftarrow S_2$
  else if $d_{\text{rel}} \leq D_{\text{sync}}$ **then**
    $S \leftarrow S_3$
  else if $S = S_3$ **and** $d_{\text{rel}} \leq D_{\text{sync}} + \Delta D_{\text{hys}}$ **then**
    $S \leftarrow S_3$
  else
    $S \leftarrow S_1$
  end if
  $\mathbf{s}_{\text{stitch}} \leftarrow$ state at $t_c + \Delta t$ on committed trajectory ▷ Equation (24b)
  if $S = S_1$ **and** $S_{\text{prev}} \neq S_1$ **then**
    $\mathbf{p}(t) \leftarrow$ SG-MINCO(remaining $\mathcal{P}_g, \mathcal{F}_g, \mathbf{s}_{\text{stitch}}$)
  else if $S = S_2$ **then**
    $\mathcal{P}_l \leftarrow$ EA-A*$_{\text{risk}}(\mathbf{s}_{\text{stitch}}, \mathcal{W}_{\text{local}}, \mathbf{v}_{\text{prev}}, \mathcal{P}_g)$ ▷ Section 3.3.1
    if $\mathcal{P}_l \neq \emptyset$ **then**
      $\mathbf{p}(t) \leftarrow$ SG-MINCO$(\mathcal{P}_l, \text{SFC}(\mathcal{P}_l), \mathbf{s}_{\text{stitch}})$ ▷ Section 3.4.2
    end if
  else if $S = S_3$ **then**
    $\mathcal{P}_l \leftarrow$ EA-A*$_{\text{attract}}(\mathbf{s}_{\text{stitch}}, T_{\Sigma,\text{prev}}, \hat{\mathbf{s}}_{\text{target}})$ ▷ Section 3.3.2
    $\mathbf{p}(t) \leftarrow$ STC-MINCO$(\mathcal{P}_l, \text{CapsuleSFC}(\mathbf{s}_{\text{target}}), \mathbf{s}_{\text{stitch}}, \mathbf{s}_{\text{target}}(\cdot))$ ▷ Section 3.4.3
  end if
  if a new trajectory $\mathbf{p}(t)$ is successfully generated **then**
    commit $\mathbf{p}(t)$; $\mathbf{v}_{\text{prev}} \leftarrow \dot{\mathbf{p}}(t)$, $T_{\Sigma,\text{prev}} \leftarrow T_\Sigma$
  else if committed trajectory maintains forward clearance ($d_{\min} \geq R_{\text{coll}}$) **then**
    retain committed trajectory for $[t_c, t_c + \Delta t]$ ▷ Tier 1: transient fallback
  else
    $\mathbf{p}(t) \leftarrow$ EmergencyBrake$(\mathbf{s}_{\text{stitch}}, -a_{\max})$; commit $\mathbf{p}(t)$ ▷ Tier 2: active safe-stop
  end if
  Track committed $\mathbf{p}(t)$ over $[t_c,\ t_c + \Delta t]$; $S_{\text{prev}} \leftarrow S$
end while

3.2. Global Path Planning via EA-ACO

Ant Colony Optimization (ACO) solves path planning problems by simulating ant foraging and pheromone exchange. In narrow warehouse environments, however, standard ACO faces key limitations: it relies on path length heuristics without considering obstacle clearance, uses fixed transition weights, and often converges prematurely to suboptimal paths with excessive turns.

To address these limitations, Environment-Aware Ant Colony Optimization (EA-ACO) is proposed. The algorithm introduces two primary modifications to the standard ACO workflow. First, an environment feedback factor is embedded into the state transition probability, guiding the search away from obstacle boundaries and suppressing sharp turns. Second, an environment-weighted pheromone update mechanism is designed to accelerate convergence by concentrating the population selection probability toward high-quality, safe paths.

3.2.1. Environment Feedback Factor Design

To enhance algorithmic responsiveness to environmental features, environment feedback factors γ_{ij} are embedded into the path selection model. Following the ant colony search paradigm, the transition probability of the proposed EA-ACO is formulated as:

$$P_{ij}^{k}(n) = \begin{cases} \frac{\tau_{ij}^{\alpha}(n)\eta_{ij}^{\beta}\gamma_{ij}^{\delta}}{\sum_{s\in \text{allowed}_k}\tau_{is}^{\alpha}(n)\eta_{is}^{\beta}\gamma_{is}^{\delta}} & j \in \text{allowed}_k \\ 0 & j \notin \text{allowed}_k \end{cases} \tag{7}$$

where P_{ij}^{k} is the transition probability from node i to node j for ant k; allowed_k represents the set of allowable next nodes, defined as the unvisited neighbors of node i with obstacle clearance $d_{\text{ob}} > R_1$ and turning deviation $|\Delta\psi| \leq \psi_{\max}$; τ_{ij} is the pheromone concentration on path (i, j); $\eta_{ij} = 1/d_{ij}$ is the reciprocal of the path (i, j) distance; α, β, δ are weighting parameters controlling the relative importance of pheromone, heuristic values, and environmental feedback factors. At each construction step, allowed_k is assembled prior to the roulette-wheel selection, such that only safe and cycle-free candidates participate in the probabilistic transition. If allowed_k is empty, the ant is trapped in a dead end and is abandoned for the current iteration without depositing pheromone.

The environment feedback factor is defined as:

$$\gamma_{ij} = f_{ij}^{d} \cdot f_{ij}^{\psi} \tag{8}$$

where f_{ij}^{d} represents the fixed obstacle impact factor and f_{ij}^{ψ} represents the path turning impact factor. These two factors guide path selection toward corridor centerlines and reduce sharp turns.

The fixed obstacle impact factor evaluates the Euclidean distance to the nearest fixed obstacle:

$$f_{ij}^{d} = \begin{cases} 0, & d_{\text{ob}} \leq R_1 \\ \frac{d_{\text{ob}} - R_1}{R_2 - R_1}, & R_1 < d_{\text{ob}} \leq R_2 \\ 1, & d_{\text{ob}} > R_2 \end{cases} \tag{9}$$

In this formulation (Equation (9)), d_{ob} stands for the Euclidean distance from candidate node j to the closest static obstacle, R_1 refers to the hard safety clearance corresponding to the UAV footprint radius ($R_1 = R_{\text{footprint}}$), whereas R_2 is the outer buffer distance corresponding to the composite collision clearance threshold ($R_2 = R_{\text{coll}}$). If the candidate node falls inside the R_1 region, then the obstacle factor is fixed to $f_{ij}^{d} = 0$ since such nodes

cannot be selected. With increasing clearance above R_1, the obstacle factor f_{ij}^{d} increases linearly up to 1. Thus, nodes with larger obstacle clearance from walls are prioritized.

The path turning factor f_{ij}^{ψ} evaluates geometric smoothness and suppresses sharp turns:

$$f_{ij}^{\psi} = \max\left(0,\ 1 - \frac{|\Delta\psi|}{\psi_{\max}}\right) \tag{10}$$

where $\Delta\psi$ is the deviation in heading and $\psi_{\max}$ is the maximum permissible turning angle. Nodes requiring turns beyond $\psi_{\max}$ receive a factor of zero and are excluded from the candidate set allowed$_k$, avoiding jagged search paths.

The combined factor γ_{ij} is computed using multiplication to form a soft logical conjunction. Because both static obstacle clearance and turn angles are geometric properties invariant across iterations, γ_{ij} is precomputed offline over the candidate edges before search iterations. If either obstacle clearance or turn smoothness does not satisfy the criteria, the product γ_{ij} approaches zero to enforce avoidance, reducing the risk of masking commonly encountered with additive weighting.

3.2.2. Environment-Weighted Pheromone Update Strategy

To guide convergence toward wider corridors, an environment-weighted rule updates global pheromone levels:

$$\tau_{ij}(n+1) = (1-\rho)\cdot\tau_{ij}(n) + f_{\gamma,ij}\sum_{k=1}^{K}\Delta\tau_{ij}^{k}(n) \tag{11}$$

$$\Delta\tau_{ij}^{k}(n) = \begin{cases} \frac{Q}{L_k} & \text{if ant } k \text{ reached the goal and traversed path } (i,j) \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

where $\tau_{ij}(n)$ is the pheromone level on path (i,j), ρ is the pheromone evaporation coefficient ($0 < \rho < 1$), K is the total number of ants, $f_{\gamma,ij} \in [0,1]$ is the edge-level environment-weighted scaling factor, Q is the pheromone enhancement constant, and L_k is the length of the path constructed by the k-th ant.

The environment-weighted factor $f_{\gamma,ij}$ is formulated via a piecewise exponential mapping:

$$f_{\gamma,ij} = \begin{cases} \exp\left(1 - \frac{1}{\gamma_{ij}}\right), & \text{if } \gamma_{ij} > 0 \\ 0, & \text{if } \gamma_{ij} = 0 \end{cases} \tag{13}$$

As defined in Equation (13), the scaling factor $f_{\gamma,ij} \in [0,1]$ acts as an edge-quality filter on pheromone accumulation. In standard ACO, shorter paths deposit higher pheromone levels ($\Delta\tau \propto 1/L_k$), which inadvertently favors risky wall-hugging routes that cut corners along shelf boundaries. The exponential weighting in Equation (13) counteracts this tendency: while collision-free edges along open corridor centerlines receive normal pheromone reinforcement ($f_{\gamma,ij} \approx 1$), edges close to obstacles or involving sharp turns suffer steep exponential attenuation ($f_{\gamma,ij} \to 0$). Consequently, even if a wall-hugging path has a shorter total length, pheromone buildup on its hazardous segments is effectively suppressed, steering colony convergence toward safe, centerline-aligned trajectories. Algorithm 2 summarizes the complete EA-ACO procedure.

Algorithm 2 EA-ACO Global Path Planning

Input: Start $\mathbf{p}_s$, goal $\mathbf{p}_{\text{goal}}$, max iterations N
Output: Best-found reference path $\mathcal{P}^*$

Initialize pheromone $\tau_{ij}(0)$, parameters $\alpha, \beta, \delta, \rho, Q$, $\text{step}_{\max}$
$\mathcal{P}^* \leftarrow \emptyset$, $L^* \leftarrow \infty$
for $n = 1$ to N **do**
  for $k = 1$ to K **do**
    Place ant k at $\mathbf{p}_s$; $\mathcal{P}_k \leftarrow \emptyset$, $L_k \leftarrow 0$, $\text{reached}_k \leftarrow$ false, step $\leftarrow 0$
    while not at $\mathbf{p}_{\text{goal}}$ and step $< \text{step}_{\max}$ **do**
      $\text{allowed}_k \leftarrow \{j \mid j \text{ unvisited}, d_{\text{ob}}(j) > R_1, |\Delta\psi_j| \le \psi_{\max}\}$
      if $\text{allowed}_k = \emptyset$ **then**
        Break
      end if
      for each $j \in \text{allowed}_k$ **do**
        Compute $\eta_{ij} = 1/d_{ij}$, f_{ij}^d, f_{ij}^{ψ}, γ_{ij}
      end for
      Compute P_{ij}^k; select j via roulette wheel
      $\mathcal{P}_k \leftarrow \mathcal{P}_k \cup \{(i,j)\}$, $L_k \leftarrow L_k + d_{ij}$, step $\leftarrow$ step $+1$
      if $j = \mathbf{p}_{\text{goal}}$ **then**
        $\text{reached}_k \leftarrow$ true; break
      end if
    end while
    if reached_k and $L_k < L^*$ **then**
      $\mathcal{P}^* \leftarrow \mathcal{P}_k$, $L^* \leftarrow L_k$
    end if
  end for
  Update $\tau_{ij}(n+1)$ via Equation (11) using edge-level weights $f_{\gamma,ij}$
end for
return $\mathcal{P}^*$

*3.3. Local Dynamic Replanning via EA-A**

The local planning layer runs Receding Horizon Planning (RHP) within a local sliding window under FSM supervision to handle sudden dynamic disturbances and moving targets. Traditional A* algorithms, despite their search completeness, suffer from an excessive computational burden and lack safety margins in dynamic environments. To resolve these challenges, the Environment-Aware A* (EA-A*) local dynamic replanner is proposed, which switches between two operational modes: a spatiotemporal risk repulsion field for Dynamic Avoiding (S_2) and a predictive attraction field for Target Synchronizing (S_3).

3.3.1. Risk Repulsion Field for Dynamic Avoiding

Under State S_2, the local planner modifies the total evaluation function $F(x)$ by introducing a spatiotemporal risk repulsion term $G_{\text{risk}}(x, t_x)$ to generate evasive forces.

Based on the dynamic obstacle's kinematic state ($\mathbf{p}_{\text{obs}}, \mathbf{v}_{\text{obs}}$) and geometry obtained from the onboard perception and tracking pipeline—where dynamic point clouds are clustered to estimate the centroid position and fitted with an empirical bounding radius R_{obs}, and tracked via an Extended Kalman Filter (EKF)—the time-migrating repulsion field is formulated as:

$$G_{\text{risk}}(x, t_x) = C_{\text{risk}} \exp\Bigg(-\frac{1}{2} (\mathbf{p}_x - \mathbf{p}_{\text{pred}}(t_x))^T \boldsymbol{\Sigma}^{-1} \times (\mathbf{p}_x - \mathbf{p}_{\text{pred}}(t_x)) \Bigg) \cdot \Phi_{\text{risk}}(t_{\text{CPA}}) \tag{14}$$

$$\Phi_{\text{risk}}(t_{\text{CPA}}) = \begin{cases} \max\left(0, 1 - \frac{t_{\text{CPA}}}{\tau_{\text{safe}}}\right), & \text{if } t_{\text{CPA}} > 0 \text{ and } d_{\text{CPA}} < R_{\text{coll}} \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where C_{risk} is the repulsion gain scaling the dynamic obstacle penalty, τ_{safe} is the safety time horizon, and $\mathbf{p}_{\text{pred}}(t_x) = \mathbf{p}_{\text{obs}} + \mathbf{v}_{\text{obs}} t_x$ is the predicted obstacle position. To evaluate localized dynamic obstacle risk along candidate branches, the arrival timestamp t_x is estimated from cumulative path length S_x using the scalar speed profile $v_{\text{prev}}(s) = \|\dot{\mathbf{p}}_{\text{prev}}(t(s))\|$ from the previous optimization cycle:

$$t_x = \int_0^{S_x} \frac{1}{\max(v_{\text{prev}}(s), v_{\min})}\, ds \tag{16}$$

where $s \in [0, L_{\text{prev}}]$ is the arc length of the previous trajectory (saturating to $\bar{v}$ for $s > L_{\text{prev}}$), and $v_{\min} > 0$ is a small positive regularization threshold ($v_{\min} = 0.1\,\text{m/s}$) to avoid division-by-zero singularities during near-hover flight. In the initial cycle ($k = 0$), t_x is initialized via $S_x/\bar{v}$ using the nominal cruise speed $\bar{v}$. Because candidate branches may deviate geometrically from the prior path, the estimated timestamp t_x has an approximation discrepancy $\Delta t_x = |t_x - t_{\text{exec}}(x)|$. This timestamp serves only as heuristic guidance for frontend node risk evaluation ($G_{\text{risk}}(x, t_x)$). Trajectory execution timing is subsequently determined by the backend optimizer (Section 3.4.2), which optimizes the continuous segment duration vector $\mathbf{T} = [T_1, \ldots, T_M]^T$ in Equation (24) under kinodynamic and corridor constraints. In addition, periodic receding-horizon replanning updates the trajectory from fresh state feedback at each cycle, preventing timing approximation errors from accumulating over time, while the composite collision threshold $R_{\text{coll}} = R_{\text{footprint}} + R_{\text{obs}}$ provides geometric clearance against short-horizon velocity variations. To analytically capture the anisotropic risk, the covariance matrix is formulated as $\mathbf{\Sigma} = \mathbf{R}(\varphi)\text{diag}(\sigma_{\text{long}}^2, \sigma_{\text{lat}}^2, \sigma_{\text{vert}}^2)\mathbf{R}(\varphi)^T$. The lateral and vertical standard deviations are set to the obstacle baseline radius ($\sigma_{\text{lat}} = \sigma_{\text{vert}} = R_{\text{obs}}$, where R_{obs} is the empirical bounding radius fitted from the obstacle point cloud cluster), while the longitudinal standard deviation is dynamically stretched along the motion direction as $\sigma_{\text{long}} = R_{\text{obs}} + k_v\|\mathbf{v}_{\text{obs}}\|$, where k_v is a velocity expansion coefficient. The rotation matrix $\mathbf{R}(\varphi)$ aligns the primary axis with the obstacle's instantaneous yaw angle $\varphi = \text{arctan2}(v_y, v_x)$ around the Z-axis, formulated as:

$$\mathbf{R}(\varphi) = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{17}$$

The relative velocity is defined as $\mathbf{v}_{\text{rel}} = \mathbf{v}_{\text{prev}}(t_x) - \mathbf{v}_{\text{obs}}$, and the relative position is $\mathbf{p}_{\text{rel}} = \mathbf{p}_x - \mathbf{p}_{\text{pred}}(t_x)$. Assuming constant relative velocity over the short horizon, the Time of Closest Approach (t_{CPA}) is:

$$t_{\text{CPA}} = \begin{cases} \frac{-\mathbf{p}_{\text{rel}}^T \mathbf{v}_{\text{rel}}}{\|\mathbf{v}_{\text{rel}}\|^2}, & \text{if } \mathbf{p}_{\text{rel}}^T \mathbf{v}_{\text{rel}} < 0 \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

The predicted distance at t_{CPA} is then evaluated against the composite collision threshold $R_{\text{coll}} = R_{\text{footprint}} + R_{\text{obs}}$:

$$d_{\text{CPA}} = d(t_{\text{CPA}}) = \|\mathbf{p}_{\text{rel}} + \mathbf{v}_{\text{rel}} t_{\text{CPA}}\| \tag{19}$$

(for obstacle i, denoted as $d_{\text{CPA},i}$ in Algorithm 1). A dynamic collision risk is flagged only when $t_{\text{CPA}} > 0$ and $d_{\text{CPA}} < R_{\text{coll}}$, activating the risk penalty $\Phi_{\text{risk}}(t_{\text{CPA}})$ in Equation (15).

If the trajectories maintain clearance ($d_{\text{CPA}} \geq R_{\text{coll}}$) or are moving apart ($\mathbf{p}_{\text{rel}}^T \mathbf{v}_{\text{rel}} \geq 0$), the penalty remains zero.

To maintain clearance around dynamic obstacles, a state-space hard pruning policy is applied. For any candidate node x at $(\mathbf{p}_x, t_x)$, state validity is evaluated via the indicator function $\mathbb{I}(x)$:

$$\mathbb{I}(x) = \begin{cases} 0, & \text{if } \exists i,\ \|\mathbf{p}_x - \mathbf{p}_{\text{pred},i}(t_x)\| < R_{\text{coll},i} \\ 1, & \text{otherwise} \end{cases} \tag{20}$$

where $\mathbf{p}_{\text{pred},i}(t_x)$ is the predicted position of obstacle i at t_x, and $R_{\text{coll},i} = R_{\text{footprint}} + R_{\text{obs},i}$ is the composite collision threshold combining the UAV footprint radius $R_{\text{footprint}}$ and the obstacle empirical bounding radius $R_{\text{obs},i}$ fitted online from the clustered point cloud. If $\mathbb{I}(x) = 0$, the node is immediately pruned ($F(x) = \infty$). Otherwise, the total cost function $F(x)$ is updated by adding the spatiotemporal risk cost as:

$$F(x) = G_{\text{dist}}(x) + H(x) + w_{\text{env}} \cdot G_{\text{risk}}(x, t_x) \tag{21}$$

where $G_{\text{dist}}(x)$ is the cumulative distance cost, w_{env} is the environmental risk weight, and $H(x) = \|\mathbf{p}_x - \mathbf{p}_{\text{local_goal}}\|$ is the Euclidean heuristic guiding toward local goal $\mathbf{p}_{\text{local_goal}}$ (selected at a lookahead distance L_{ahead} along the global reference path $\mathcal{P}_g$ from the projection of the stitch state). This formulation combines spatiotemporal risk evaluation with discrete node pruning to maintain clearance around moving obstacles. Both the localized risk penalty $G_{\text{risk}}(x, t_x)$ and node validity directly incorporate arrival timestamps t_x (Equation (16)) projected along candidate search branches to evaluate dynamic obstacle risk at candidate nodes. To ensure real-time execution on the onboard computer, nodes in the graph search are indexed by 3D spatial coordinates $\mathbf{p}_x = (x, y, z)$, retaining a single arrival per voxel. Under receding-horizon replanning, dynamic obstacles are handled across successive replanning cycles. This spatial search formulation bounds frontend complexity to $O(N_{\max} \log N_{\max})$, while the continuous backend optimizer handles kinematic smoothness and constraint satisfaction.

3.3.2. Predictive Attraction Field for Target Synchronizing

In State S_3, to handle platform motion and sensor noise, an Extended Kalman Filter (EKF) running on the onboard computer estimates the platform's initial state $\hat{\mathbf{s}}_{\text{target}} = [\hat{\mathbf{p}}_{\text{target}}^T, \hat{\mathbf{v}}_{\text{target}}^T, \hat{\mathbf{a}}_{\text{target}}^T]^T$ at $\tau = 0$ by fusing onboard Ultra-Wideband (UWB) positioning with platform telemetry (wheel odometry and IMU). Based on this filtered state, a continuous target rendezvous state function $\mathbf{s}_{\text{target}}(\tau)$ is constructed over the local predictive horizon $\tau \in [0, \infty)$ using a local second-order Taylor expansion:

$$\mathbf{s}_{\text{target}}(\tau) = \begin{bmatrix} \mathbf{p}_{\text{target}}(\tau) \\ \mathbf{v}_{\text{target}}(\tau) \\ \mathbf{a}_{\text{target}}(\tau) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{p}}_{\text{target}} + \hat{\mathbf{v}}_{\text{target}}\tau + \frac{1}{2}\hat{\mathbf{a}}_{\text{target}}\tau^2 + \begin{bmatrix} 0 \\ 0 \\ \Delta z \end{bmatrix} \\ \hat{\mathbf{v}}_{\text{target}} + \hat{\mathbf{a}}_{\text{target}}\tau \\ \hat{\mathbf{a}}_{\text{target}} \end{bmatrix} \tag{22}$$

where Δz is a prescribed vertical hovering offset distinguishing the target state of the UAV from the geometric center of the moving platform, preparing for the subsequent vision-based precision landing phase. Treating target acceleration as piecewise constant over the local horizon ($\tau \in [0, T_\Sigma]$) sets predicted target jerk to $\mathbf{j}_{\text{target}}(\tau) \approx \mathbf{0}$, simplifying gradient evaluation in the trajectory optimizer. Any discrepancy between this quadratic extrapolation and higher-order target maneuvers is compensated in closed loop through high-frequency receding-horizon updates. Based on the remaining flight duration $T_{\Sigma,\text{prev}}$

computed from the previous backend optimization cycle (the temporal length from $\tau = 0$ to the end of the optimized trajectory), the frontend heuristic is guided by the predicted spatial position at $\tau = T_{\Sigma,\text{prev}}$. The dynamic heuristic function $H(x)$ is defined as the Euclidean distance to this predicted position:

$$H(x) = \|\mathbf{p}_x - \mathbf{p}_{\text{target}}(T_{\Sigma,\text{prev}})\| \tag{23}$$

Thus, the frontend search direction is aligned with the spatiotemporal optimum through a prediction-guided warm-start, establishing close feedforward–feedback coupling.

Algorithm 3 unifies the two local replanning modes into a single search framework. In State S_2, the search targets a lookahead waypoint on the global path $\mathcal{P}_g$ with the environmental risk weight enabled ($w = w_{\text{env}}$). In State S_3, the heuristic directs the search toward the predicted rendezvous position $\mathbf{p}_{\text{target}}(T_{\Sigma,\text{prev}})$ with the risk weight disabled ($w = 0$). Starting from the stitch state $\mathbf{s}_{\text{stitch}}$ to preserve kinematic continuity, EA-A* expands candidate neighbors within $\mathcal{W}_{\text{local}}$. Each expansion step updates arrival timestamps via Equation (16) and prunes colliding nodes where $\mathbb{I}(x') = 0$. EA-A* operates at high frequency as a local heuristic replanner to generate feasible obstacle-avoidance guide waypoints for the continuous backend optimizer. When a candidate node reaches the local goal within tolerance δ_{arr} (set to one grid resolution), the planner backtracks parent pointers to recover the local path. If node expansions exhaust the computational budget $N_{\max}$, the algorithm backtracks from the node with the lowest remaining heuristic cost to construct a partial guide path.

3.3.3. Receding-Horizon Execution and Fallback Strategy

The local planning layer operates as a Spatiotemporal Receding Horizon Planning (RHP) framework within a 3D sliding window $\mathcal{W}_{\text{local}}$ coupled with the sensor range. The spatiotemporal consistency and numerical tractability of this RHP paradigm are maintained by two main frontend-backend coupling mechanisms:

First, under State S_2, the local frontend extracts the lookahead target $\mathbf{p}_{\text{local_goal}}$ at distance L_{ahead} along the global reference path $\mathcal{P}_g$ as the local planning goal. When the local search is truncated due to intersecting the boundaries of $\mathcal{W}_{\text{local}}$ or reaching the node expansion cap $N_{\max}$, the algorithm selects the node that minimizes the remaining heuristic cost to the target as the terminal guide point. This creates a heuristic convergence trend toward the global track, guiding the UAV to re-engage the reference trajectory after evading dynamic intrusions. The resulting local path is then converted into convex SFCs that restrict the backend SG-MINCO optimization.

Second, under State S_3, spatiotemporal state synchronization is maintained via feedback-feedforward coupling. The frontend uses the remaining flight duration $T_{\Sigma,\text{prev}}$ to predict the interception point $\mathbf{p}_{\text{target}}(T_{\Sigma,\text{prev}})$. The backend optimization is driven by the chain-rule-based temporal gradient compensation and the hard-embedded terminal boundary conditions, mitigating dynamic tracking lag and aligning the trajectory with the dynamic boundaries of the corridor.

When local replanning fails to yield a valid path within the allocated horizon, the FSM supervisor executes the two-tier fallback policy detailed in Section 3.1: if the committed trajectory is predicted to breach the collision threshold ($d_{\min} < R_{\text{coll}}$), the UAV commands maximum deceleration $\mathbf{a} = -a_{\max}\frac{\mathbf{v}}{\|\mathbf{v}\|}$, bringing the vehicle to a hover in place within stopping distance $s_{\text{stop}} \leq \frac{\|\mathbf{v}\|^2}{2a_{\max}}$ until a viable path is re-established.

Algorithm 3 EA-A* Local Dynamic Replanning

Input: Mode $m \in \{S_2, S_3\}$, stitch state $\mathbf{s}_{\text{stitch}}$, local window $\mathcal{W}_{\text{local}}$, velocity profile $\mathbf{v}_{\text{prev}}$, global reference $\mathcal{P}_g$; platform estimate $\hat{\mathbf{s}}_{\text{target}}$ and remaining duration $T_{\Sigma,\text{prev}}$ (if $m = S_3$)
Output: Local path $\mathcal{P}_{\text{local}}$

if $m = S_2$ **then**
  $\mathbf{p}_{\text{local_goal}} \leftarrow$ lookahead point on $\mathcal{P}_g$; $w \leftarrow w_{\text{env}}$
else if $m = S_3$ **then**
  Construct $\mathbf{s}_{\text{target}}(\cdot)$ from $\hat{\mathbf{s}}_{\text{target}}$; $\mathbf{p}_{\text{local_goal}} \leftarrow \mathbf{p}_{\text{target}}(T_{\Sigma,\text{prev}})$; $w \leftarrow 0$
end if
$H(x) \leftarrow \|\mathbf{p}_x - \mathbf{p}_{\text{local_goal}}\|$
OPEN $\leftarrow \{\mathbf{p}_{\text{stitch}}\}$, CLOSED $\leftarrow \emptyset$, $n_{\text{exp}} \leftarrow 0$, found $\leftarrow$ false
$G_{\text{dist}}(\mathbf{p}_{\text{stitch}}) \leftarrow 0$; $F(\mathbf{p}_{\text{stitch}}) \leftarrow H(\mathbf{p}_{\text{stitch}})$; parent$(\mathbf{p}_{\text{stitch}}) \leftarrow$ null
while OPEN $\neq \emptyset$ **and** $n_{\text{exp}} \leq N_{\max}$ **and not** found **do**
  $x \leftarrow \arg\min_{x \in \text{OPEN}} F(x)$; move x to CLOSED; $n_{\text{exp}} \leftarrow n_{\text{exp}} + 1$
  if $\|\mathbf{p}_x - \mathbf{p}_{\text{local_goal}}\| \leq \delta_{\text{arr}}$ **then**
    found $\leftarrow$ true
  else
    for each neighbor x' of x within $\mathcal{W}_{\text{local}}$ **do**
      $t_{x'} \leftarrow$ arrival timestamp calibrated via Equation (16)
      if $x' \in$ CLOSED **or** $\mathbb{I}(x') = 0$ **then**
        continue
      end if
      $g(x') \leftarrow G_{\text{dist}}(x) + \|\mathbf{p}_{x'} - \mathbf{p}_x\|$
      if $x' \notin$ OPEN **or** $g(x') < G_{\text{dist}}(x')$ **then**
        $G_{\text{dist}}(x') \leftarrow g(x')$; parent$(x') \leftarrow x$
        $F(x') \leftarrow G_{\text{dist}}(x') + H(x') + w \cdot G_{\text{risk}}(x', t_{x'})$; insert x' into OPEN
      end if
    end for
  end if
end while
$\mathcal{V}_{\text{cand}} \leftarrow (\text{OPEN} \cup \text{CLOSED}) \setminus \{\mathbf{p}_{\text{stitch}}\}$
if found **then**
  $\mathcal{P}_{\text{local}} \leftarrow$ backtrack parents from x
else if $\mathcal{V}_{\text{cand}} \neq \emptyset$ **then**
  $x_{\text{fg}} \leftarrow \arg\min_{v \in \mathcal{V}_{\text{cand}}} H(v)$; $\mathcal{P}_{\text{local}} \leftarrow$ backtrack parents from x_{fg}
else
  $\mathcal{P}_{\text{local}} \leftarrow \emptyset$
end if
return $\mathcal{P}_{\text{local}}$

3.4. Trajectory Optimization via SG-MINCO and STC-MINCO

The backend optimization layer translates discrete frontend paths into continuous, dynamically feasible polynomial trajectories. Based on the active FSM state, the framework uses two distinct formulations: under States S_1 and S_2 (Global Approaching and Dynamic Avoiding), Spatially-Guided MINCO (SG-MINCO) optimizes trajectories within safe flight corridors (SFCs); under State S_3 (Target Synchronizing), Spatiotemporal Coupled MINCO (STC-MINCO) achieves spatiotemporal state synchronization with the moving platform.

3.4.1. Trajectory Formulation and SFC Generation

The backend jointly optimizes the continuous spatial trajectory $\mathbf{p}(t)$ across M polynomial segments and the segment time allocation vector $\mathbf{T} = [T_1, \ldots, T_M]^T \in \mathbb{R}^M_{>0}$. Defining cumulative knot timestamps $t_0 = 0$ and $t_j = \sum_{l=1}^{j} T_l$ $(j = 1, \ldots, M)$, the general trajectory optimization problem is formulated as:

$$\min_{\mathbf{p}(t),\mathbf{T}} \quad J = \int_0^{T_\Sigma} \|\mathbf{p}^{(3)}(t)\|^2 dt + \rho_T T_\Sigma \tag{24a}$$

$$\text{s.t.} \quad \mathbf{s}_1(0) = \mathbf{s}_{\text{stitch}} \tag{24b}$$

$$\mathbf{p}(t) \in \mathcal{F}_j, \quad \forall t \in [t_{j-1}, t_j] \tag{24c}$$

$$\|\dot{\mathbf{p}}(t)\| \leq v_{\max}, \quad \forall t \in [0, T_\Sigma] \tag{24d}$$

$$\|\ddot{\mathbf{p}}(t) + g\mathbf{e}_z\| \leq \frac{f_{\max}}{m}, \quad \forall t \in [0, T_\Sigma] \tag{24e}$$

$$\mathbf{s}_M(T_\Sigma) = \mathbf{s}_{\text{target}}(T_\Sigma) \tag{24f}$$

where $T_\Sigma = \sum_{j=1}^{M} T_j$ denotes total flight duration and ρ_T weights the time regularizer. Across replanning cycles, the stitch condition in Equation (24b) enforces C^2 continuity ($[\mathbf{p}^T, \mathbf{v}^T, \mathbf{a}^T]^T$) with the active trajectory at $t_c + \Delta t$, preventing discontinuities in collective thrust and vehicle attitude. Within each optimized trajectory, the minimum-jerk formulation yields piecewise fifth-order polynomials that exhibit C^4 continuity across interior knots (junction conditions detailed in Appendix A.2). The terminal equality in Equation (24f) specifies target rendezvous.

To enforce the spatial safety constraint in Equation (24c), a sequence of safe flight corridors (SFCs), $\mathcal{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_M\}$, is generated along the reference waypoints. Each corridor is formed by expanding convex polyhedra around path segments, defined by the linear inequalities:

$$A_j \mathbf{p} \leq \mathbf{b}_j - r_{\text{corridor}} \mathbf{1} \tag{25}$$

or equivalently for each bounding hyperplane $m \in \{1, \ldots, K_j\}$ as $\mathbf{a}_{j,m}^T \mathbf{p} \leq b_{j,m} - r_{\text{corridor}}$, where $\mathbf{1} = [1, \ldots, 1]^T$ denotes the all-ones vector. Each hyperplane normal vector is normalized ($\|\mathbf{a}_{j,m}\| = 1$), such that $r_{\text{corridor}} = R_{\text{footprint}} + \delta_{\text{margin}}$ establishes an inward erosion margin protecting against vehicle dimensions and numerical discretization. Frontends with higher obstacle clearance naturally create wider corridors, giving the backend optimizer greater freedom to smooth curves and satisfy dynamic bounds.

3.4.2. Spatially-Guided MINCO (SG-MINCO)

Enforcing geometric inequality constraints in narrow corridors represents a primary computational bottleneck in trajectory optimization. Conventional methods formulate general convex polytopes for penalty gradients, which increases numerical overhead during inner iterations. Moreover, joint minimization of control effort and flight duration causes trajectories to cut tightly around corridor turns (a taut-string effect), reducing obstacle clearance near shelf boundaries.

To overcome these geometric bottlenecks, the SG-MINCO approach utilizes orthogonal characteristics of the warehouse layout. The safe flight corridor is initially segmented into Axis-Aligned Bounding Boxes (AABBs) centered at intermediate waypoints $\mathbf{P}_{\text{ref},i}$. An anisotropic diffeomorphic mapping via hyperbolic tangent functions is used to transform box boundaries into unconstrained coordinate spaces separately along the X, Y, and Z axes:

$$\mathbf{q}_i = \mathbf{P}_{\text{ref},i} + \mathbf{d}_{\text{box},i} \odot \tanh(\boldsymbol{\xi}_i) \tag{26}$$

where $\mathbf{d}_{\text{box},i} \in \mathbb{R}^3_{>0}$ specifies the AABB expansion half-lengths, $\odot$ denotes the Hadamard (element-wise) product, and $\boldsymbol{\xi}_i \in \mathbb{R}^3$ are unconstrained optimization variables. To compensate for the inward attraction of time and control penalty terms, the waypoint spatial guidance penalty $J_{\text{guide}}(\boldsymbol{\xi}) = \sum_{i=1}^{M-1} \|\mathbf{q}_i(\boldsymbol{\xi}) - \mathbf{P}_{\text{ref},i}\|^2$ attracts intermediate waypoints to the safe frontend centerline. While the tanh mapping in Equation (26) confines discrete

intermediate waypoints within their local bounding boxes, the continuous polynomial curves between waypoints must also satisfy dynamic feasibility and remain within the safe flight corridor. Building upon the MINCO trajectory framework [35], both actuator dynamic limits and continuous-time corridor boundary constraints along each segment are evaluated via numerical Simpson quadrature and enforced as hard inequality constraints within the Powell–Hestenes–Rockafellar (PHR) Augmented Lagrangian Method (ALM) framework [37]. The optimization problem is formulated as:

$$\min_{\boldsymbol{\xi},\tilde{\mathbf{T}}} \quad \mathcal{L}_{\mathrm{SG}}(\boldsymbol{\xi},\tilde{\mathbf{T}};\boldsymbol{\mu},\boldsymbol{\rho}) = J_{\mathrm{MINCO}}(\boldsymbol{\xi},\mathbf{T}(\tilde{\mathbf{T}})) + J_{\mathrm{ALM}}(\boldsymbol{\xi},\mathbf{T}(\tilde{\mathbf{T}});\boldsymbol{\mu},\boldsymbol{\rho}) + \lambda_{\mathrm{guide}} J_{\mathrm{guide}}(\boldsymbol{\xi}) \tag{27}$$

where $J_{\mathrm{MINCO}}(\boldsymbol{\xi},\mathbf{T})$ computes the base jerk energy, and J_{ALM} penalizes continuous corridor boundary penetration and dynamic limit violations (velocity, acceleration, thrust, and tilt angle) using the Powell–Hestenes–Rockafellar (PHR) augmented Lagrangian kernel, parameterized by dual multiplier vector $\boldsymbol{\mu} \geq \mathbf{0}$ and penalty parameter vector $\boldsymbol{\rho} > \mathbf{0}$. The inner subproblem is solved via the Limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm, while outer iterations update $\boldsymbol{\mu}$ and $\boldsymbol{\rho}$ (detailed in Appendix A.1). Segment duration positivity is enforced by the smooth diffeomorphism $T_j = \exp(\tilde{T}_j)$ with unconstrained variables $\tilde{T}_j \in \mathbb{R}$ (forming the unconstrained duration vector $\tilde{\mathbf{T}} \in \mathbb{R}^M$). In receding-horizon operation, dual multipliers are warm-started across replanning cycles, allowing ALM to achieve high-accuracy constraint satisfaction within a few outer iterations. Through the waypoint spatial guidance penalty, the trajectory effectively reduces corner-cutting, maintaining the clearance determined by the frontend.

3.4.3. Spatiotemporal Coupled MINCO (STC-MINCO)

During target synchronization (State S_3), tracking the time-varying trajectory of the moving platform couples spatial geometry with terminal arrival time T_Σ. Enforcing moving boundary conditions via explicit nonlinear equality constraints often induces ill-conditioned optimization and slow numerical convergence.

To circumvent this issue, STC-MINCO embeds the rendezvous state $\mathbf{s}_{\mathrm{target}}(T_\Sigma) = [\mathbf{p}_{\mathrm{target}}^T, \mathbf{v}_{\mathrm{target}}^T, \mathbf{a}_{\mathrm{target}}^T]^T$ directly into the linear boundary system of MINCO, converting boundary matching into an unconstrained space-time optimization problem:

$$\min_{\boldsymbol{\xi},\tilde{\mathbf{T}}} \quad \mathcal{L}_{\mathrm{STC}} = \mathcal{L}_{\mathrm{SG}} + w_t T_\Sigma \tag{28}$$

where $w_t T_\Sigma$ penalizes total flight duration. Because changing any segment duration T_i shifts the rendezvous timestamp T_Σ, the terminal boundary state drifts accordingly. Applying the multivariate chain rule, the total derivative of $\mathcal{L}_{\mathrm{STC}}$ with respect to segment duration T_i is:

$$\frac{d\mathcal{L}_{\mathrm{STC}}}{dT_i} = \frac{\partial \mathcal{L}_{\mathrm{STC}}\big|_{\mathrm{native}}}{\partial T_i} + \left(\frac{\partial \mathcal{L}_{\mathrm{STC}}}{\partial \mathbf{s}_f}\right)^T \begin{bmatrix} \mathbf{v}_{\mathrm{target}}(T_\Sigma) \\ \mathbf{a}_{\mathrm{target}}(T_\Sigma) \\ \mathbf{j}_{\mathrm{target}}(T_\Sigma) \end{bmatrix} + w_t \tag{29}$$

Here, $\frac{\partial \mathcal{L}_{\mathrm{STC}}\big|_{\mathrm{native}}}{\partial T_i}$ denotes the duration derivative holding boundary states fixed, and $\frac{\partial \mathcal{L}_{\mathrm{STC}}}{\partial \mathbf{s}_f} \in \mathbb{R}^9$ represents the adjoint sensitivity vector. Because $T_\Sigma = \sum_{j=1}^{M} T_j$, the duration sensitivity satisfies $\frac{\partial T_\Sigma}{\partial T_i} = 1$ for all segments, distributing the terminal state gradient uniformly across each segment duration. Under the onboard second-order Taylor prediction model (Equation (22)), target acceleration is treated as piecewise constant over the local horizon, yielding $\mathbf{j}_{\mathrm{target}} \approx \mathbf{0}$. The complete analytical derivation is provided in Appendix A.2. To maintain spatial clearance as the rendezvous location evolves during optimization,

a capsule-shaped safe flight corridor is constructed by sweeping the vehicle bounding volume along the relative velocity vector.

This coupled gradient enables L-BFGS to optimize trajectory geometry and segment durations concurrently, achieving continuous spatiotemporal synchronization without ill-conditioned boundary constraints.

4. Simulation and Experimental Validation

4.1. Simulation Environment Setup

A three-dimensional flight space simulating warehouse logistics environments is constructed with 20 fixed shelves (2.0 m × 1.0 m × 4.0 m) and four supporting columns (1.5 m × 1.5 m × 4.0 m) at the outer corners. Aisle width is set to 1.5 m. A geometric clearance threshold of 0.2 m defines the physical collision boundary (matching the UAV footprint radius $R_{\text{footprint}}$), where clearances approaching this threshold indicate safety-critical proximity. To reflect realistic onboard sensing constraints, a 3D LiDAR sensor is simulated within the environment without assuming prior knowledge of dynamic obstacle trajectories. The dynamic obstacles are perceived in real-time via a cluster-based object tracking pipeline applied to the raw point cloud. The system updates the obstacle's current position $\mathbf{p}_{\text{obs}}$ and estimates its smooth linear velocity $\mathbf{v}_{\text{obs}}$ at a frequency of 50 Hz, which is used for local dynamic replanning. All simulation studies were carried out on a workstation configured with an Intel Core i5-14600KF CPU, an NVIDIA GeForce RTX 4060 Ti GPU, and 32 GB RAM.

4.2. Simulation Verification

To evaluate the navigation performance and dynamic feasibility of HEA-Planner, three progressive simulation studies are conducted in the warehouse logistics environment, directly corresponding to the three operational FSM states: Global Approaching (S_1), Dynamic Avoiding (S_2), and Target Synchronizing (S_3). To support reproducibility and facilitate comparison, the key configurations of the environment, the UAV physical limits, and the configuration details for all evaluated path planning, dynamic replanning, and trajectory optimization algorithms are summarized in Table 1. The hyperparameter configurations for the corresponding algorithms in Table 1 are set through extensive simulation testing to strike a balance between safety, trajectory quality, and computational efficiency.

Table 1. Simulation parameter settings.

Parameter Name	Value
Flight range	$x \in [-6, 15]$, $y \in [-6, 22]$, $z \in [0, 4]$ m
UAV start point	$(15, 6.75, 3.5)$ m
UAV target point	$(1, 1.75, 0.5)$ m
Number of obstacles	Static: 24, Dynamic: 2
Obstacle dimensions	2.0 m × 1.0 m × 4.0 m, 1.5 m × 1.5 m × 4.0 m
Dynamic obstacles	Obs 1: $R_{\text{obs}} = 0.5$ m, $h = 2.0$ m, $v = 0.5$ m/s Obs 2: $R_{\text{obs}} = 0.5$ m, $h = 0.2$ m, $v = 2.0$ m/s
Search resolution	0.1 m
Path/Trajectory sampling	0.1 m, 0.05 s
EA-ACO parameters	$\rho = 0.3$, $K = 30$, $N = 200$, $\alpha = 2.0$, $\beta = 4.0$, $\delta = 4.0$, $Q = 1.0$, $\text{step}_{\max} = 500$, $R_1 = 0.2$ m, $R_2 = 0.7$ m, $\psi_{\max} = 90^\circ$
ACO parameters	$\rho = 0.3$, $K = 30$, $N = 200$, $\alpha = 2.0$, $\beta = 4.0$
GA parameters	$K = 800$, $N = 200$, $P_c = 0.6$, $P_m = 0.4$

Table 1. *Cont.*

Parameter Name	Value
RRT* parameters	$N = 40,000$, $\varepsilon = 1.0$, $P_{\text{goal}} = 0.3$, $r = 1.0$
MINCO parameters	$v_{\max}^{S_1} = 2.0\,\text{m/s}$, $v_{\max}^{S_2,S_3} = 1.2\,\text{m/s}$, $a_{\max}^{S_1} = 5.0\,\text{m/s}^2$, $a_{\max}^{S_2,S_3} = 3.0\,\text{m/s}^2$, $\theta_{\max}^{S_1} = 35.0^\circ$, $\theta_{\max}^{S_2,S_3} = 21.0^\circ$, $f_{\max}/m = 11.96\,\text{m/s}^2$, $\omega_{\max} = 2.5\,\text{rad/s}$, $f_i \in [0.5, 7.5]\,\text{N}$, $\lambda_{\text{guide}} = 10$, $\rho_T = 20$, $w_t = 10$
EA-A* parameters	$w_{\text{env}} = 2.0$, $C_{\text{risk}} = 15$, $k_v = 0.5\,\text{s}$, $\tau_{\text{safe}} = 2.0\,\text{s}$, $L_{\text{ahead}} = 2.0\,\text{m}$, $v_{\min} = 0.1\,\text{m/s}$, $N_{\max} = 1000$, $\mathcal{W}_{\text{local}} = 5.0 \times 5.0 \times 4.0\,\text{m}$
A* parameters	$N_{\max} = 1000$, $\mathcal{W}_{\text{local}} = 5.0 \times 5.0 \times 4.0\,\text{m}$
DWA parameters	$t_{\text{sim}} = 3.0\,\text{s}$, $\Delta v = 0.1\,\text{m/s}$, $w_{\text{head}} = 0.1$, $w_{\text{dist}} = 0.2$, $w_{\text{vel}} = 0.1$
AIFDS parameters	$\gamma = 0.99$, learning rate $= 1 \times 10^{-3}$, $\tau = 0.005$, $\sigma = 0.2$, $c = 0.5$, $d = 2$
PPO parameters	$\gamma = 0.99$, learning rate $= 3 \times 10^{-3}$, $\lambda = 0.95$, $\varepsilon = 0.2$

4.2.1. Simulation I: Performance Comparison of Global Approaching (State S_1)

To evaluate global approaching performance in State S_1, EA-ACO is benchmarked against standard ACO, GA, and RRT* across ten independent trials. All global planners operate on an identical 3D occupancy grid with matching start and goal coordinates, and backend optimizers share the same discrete reference waypoints and kinematic limits (Table 1). All performance metrics are evaluated as mean ± sample standard deviation ($N = 10$) to account for algorithmic stochasticity. As summarized in Table 2 and Figure 3, EA-ACO improves safety and smoothness over standard ACO with only a 1.6% increase in path length and a 5.3% computational overhead. Specifically, by incorporating the environment feedback factor and environment-weighted update strategy, EA-ACO mitigates wall-hugging behaviors, increasing the minimum obstacle clearance to more than 1.6 times that of standard ACO and reducing the average turning angle by 57.7%. Regarding computational stability, EA-ACO converges rapidly within a few generations. In contrast, RRT* suffers from high search variance due to random sampling, while GA often struggles to achieve full convergence within the solver timeout due to premature local convergence.

Table 2. Performance comparison of global path planning algorithms.

Metric	EA-ACO	ACO	GA	RRT*
Path Length (m)	26.80 ± 0.57	26.37 ± 0.36	28.89 ± 0.93	26.88 ± 1.05
Comp. Time (s)	37.6 ± 3.2	35.7 ± 3.6	42.8 ± 4.9	29.1 ± 5.7
Avg Turn (°)	12.7 ± 2.0	30.0 ± 4.0	17.7 ± 3.7	22.3 ± 2.7
Avg Clearance (m)	0.82 ± 0.02	0.57 ± 0.02	0.75 ± 0.05	0.67 ± 0.01
Min Clearance $d_{\min}$ (m)	0.60 ± 0.04	0.23 ± 0.04	0.43 ± 0.07	0.35 ± 0.01

Building upon the discrete reference waypoints generated by EA-ACO, the backend trajectory optimization layer evaluates Spatially-Guided MINCO (SG-MINCO) for continuous trajectory generation against a standard MINCO formulation (EA-ACO-MINCO) without spatial guidance. As detailed in Figure 4, standard MINCO prioritizes minimizing execution time but exhibits corner-cutting at the U-turn corridor, reducing the minimum obstacle clearance to 0.25 m, which leaves a narrow safety margin of only 0.05 m beyond the vehicle footprint ($R_{\text{footprint}} = 0.2\,\text{m}$). In contrast, with the spatial guidance penalty J_{guide}, SG-MINCO maintains a minimum obstacle clearance of 0.51 m (+104%), keeping the trajectory near corridor centerlines. Trajectory smoothness is quantified by the average jerk $J_{\text{avg}} = \frac{1}{T}\int_0^T \|\mathbf{p}^{(3)}(t)\|\,dt$, reflecting control effort and actuator demand. Specifically, HEA-Planner achieves a trajectory length of 26.32 m and flight time of 18.7 s, with average/peak

velocity of 1.34/1.61 m/s, average/peak acceleration of 0.64/1.54 m/s^2, and average/peak jerk of 0.85/4.24 m/s^3. In contrast, standard MINCO achieves a shorter time (16.8 s) but leaves a smaller clearance. SG-MINCO enforces all kinematic and dynamic states within admissible bounds via the Augmented Lagrangian Method, preserving obstacle clearance with a minor increase in flight time (+1.9 s).

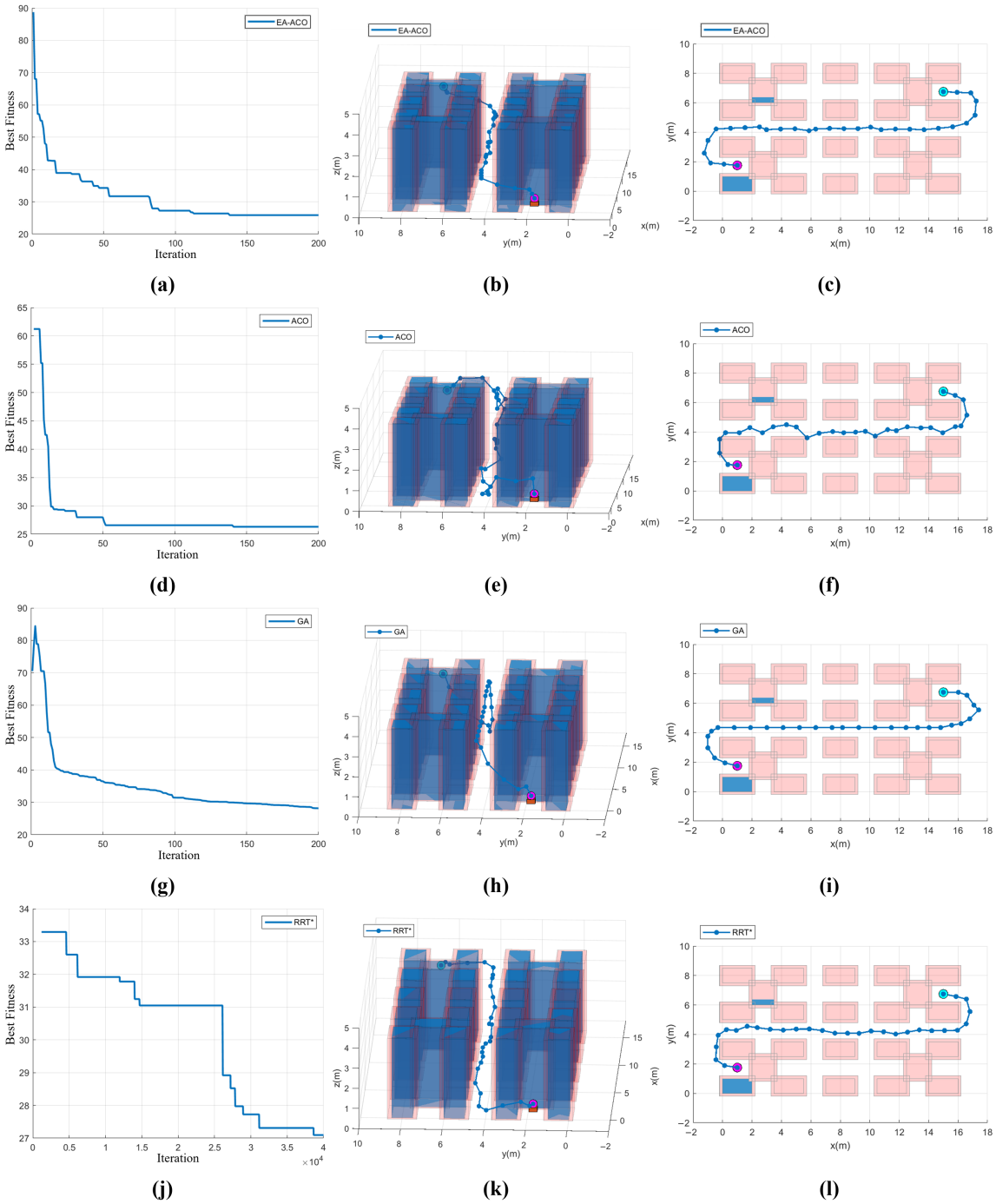


Figure 3. Performance and path comparison of global path planning algorithms: (**a**–**c**) EA-ACO, (**d**–**f**) standard ACO, (**g**–**i**) GA, and (**j**–**l**) RRT*, displaying the convergence curves of best-found path length (**left column**), 3D path planning views (**middle column**), and top-view 2D projections (**right column**).

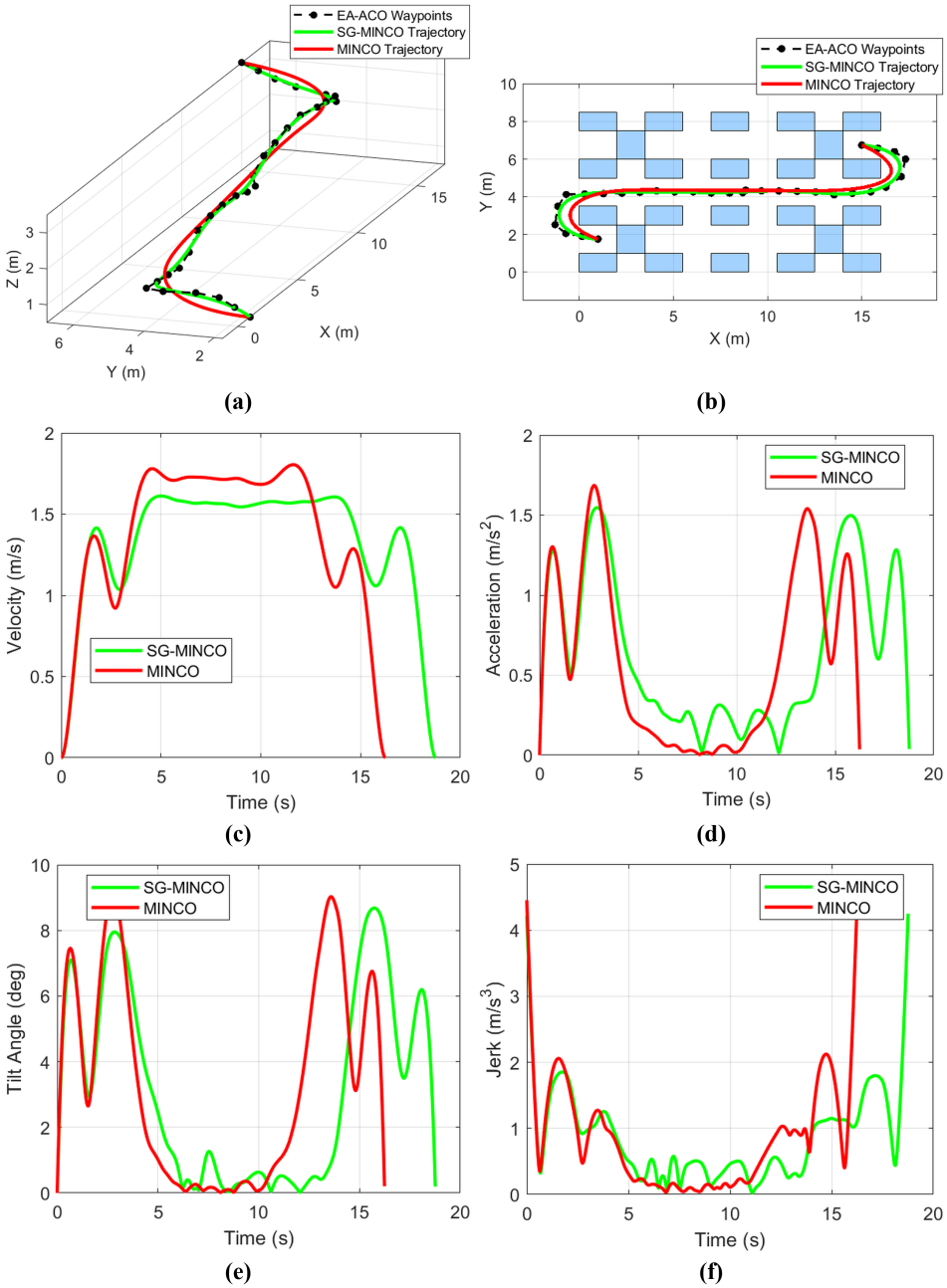


Figure 4. Comparison of trajectory smoothing and kinematic parameter profiles. (**a**) 3D trajectories. (**b**) Top-view projection. (**c**) Velocity magnitude. (**d**) Acceleration magnitude. (**e**) Body tilt angle. (**f**) Jerk magnitude.

4.2.2. Simulation II: Performance Comparison of Dynamic Obstacle Avoidance (State S_2)

This experiment evaluates HEA-Planner against four baselines: a standard baseline coupling conventional A* search with standard MINCO trajectory optimization (A*-MINCO), Dynamic Window Approach (DWA), Adaptive Interfered Fluid Dynamical System with Twin Delayed DDPG (AIFDS-TD3) [38], and Proximal Policy Optimization (PPO), under sudden dynamic intrusions. All five planners receive real-time obstacle estimates from the simulated LiDAR tracking pipeline (50 Hz) without preview of future dynamic obstacle trajectories, subject to the dynamic limits in Table 1. As shown in Table 3, Figures 5 and 6, model-free policies like PPO suffer from significant acceleration violations (6.49 m/s^2, 116.3% over-saturation) and control chattering (an average jerk of 10.08 m/s^3) under imminent threats. For discrete search methods, adopting a receding-horizon strategy substantially reduces computation time, enabling A*-based algorithms for local planning.

However, the standard A*-MINCO baseline suffers from limited predictive capability at obstacle boundaries. In contrast, by formulating a spatiotemporal risk repulsion field, HEA-Planner establishes predictive soft guidance that mitigates computing spikes, achieving an average total replanning latency (frontend path planning plus backend trajectory optimization) of 32.9 ms to support high-frequency online execution.

In terms of geometric safety, the 3D evasive flight trajectories and distance monitoring curves depicted in Figure 6 demonstrate that direct-velocity baselines (DWA, AIFDS-TD3, PPO) drop to hazardous levels ($\sim$0.1 m) due to uncoordinated local sensing (Figure 6b), while A*-MINCO induces corridor corner-cutting (0.34 m clearance). Conversely, by synthesizing the predictive risk potential fields with the waypoint spatial guidance J_{guide}, HEA-Planner guides the continuous trajectory to deform around high-risk regions in advance (Figure 6a). This dual-strategy coordination suppresses sharp clearance drops, yielding a smooth trajectory that increases the minimum clearance to 0.77 m while capping peak acceleration at 2.82 m/s^2 and average jerk at 3.85 m/s^3.

Table 3. Performance comparison of dynamic obstacle avoidance methods.

Metric	HEA-Planner	A*-MINCO	DWA	AIFDS-TD3	PPO
Avg Replan Time (ms)	32.9	38.1	15.0	29.4	31.0
Max Replan Time (ms)	40.4	42.9	18.1	38.6	45.5
Min Clearance $d_{\min}$ (m)	0.77	0.34	0.12	0.07	0.11
Flight Time (s)	38.3	37.3	36.7	34.9	36.5
Max Accel (m/s^2)	2.82	2.66	3.00	3.00	6.49
Avg Jerk (m/s^3)	3.85	8.07	20.01	7.89	10.08

4.2.3. Simulation III: Performance Comparison of Target Synchronization (State S_3)

This experiment evaluates target synchronization under $\sigma_{\text{noise}} = 0.05$ m sensor noise. This phase aims to reach a visual handover window (0.5 m above the moving platform) rather than physical touchdown, requiring terminal spatiotemporal synchronization defined by the horizontal handover position error $e_p = \sqrt{\Delta x^2 + \Delta y^2} < 200$ mm and the terminal velocity mismatch $e_v = \|\mathbf{v} - \mathbf{v}_{\text{target}}\| < 0.1$ m/s.

In variable-speed tests, the ground platform followed a continuous cosine velocity profile spanning $0.5 \sim 1.0$ m/s with a period of 4.0 s and a total motion duration of 45 s. Both HEA-Planner and PPO operated without prior trajectory knowledge, relying solely on real-time state estimates. At each cycle, HEA-Planner used the onboard EKF estimate $\hat{\mathbf{s}}_{\text{target}}$ and the local second-order Taylor prediction (Equation (22)) with $\mathbf{j}_{\text{target}} \approx \mathbf{0}$, resolving acceleration variations through receding-horizon replanning at 20 Hz. The benchmark PPO policy received the same noisy state estimate as HEA-Planner, comprising the estimated moving-platform state and the UAV's own state. It directly commanded 3D velocities subject to the dynamic limits of State S_3 (Table 1) at the same 20 Hz inference frequency. It was trained to convergence over 2×10^6 interaction steps with dense relative-tracking and velocity-matching rewards, together with action-rate smoothness penalties that suppress excessive acceleration and control chattering (PPO hyperparameters in Table 1). In constant-speed tests, both methods were evaluated over an identical discrete sweep of platform velocities (0.3, 0.5, 0.8, and 1.0 m/s), with ten synchronization trials per level. Random seeds were used to generate the initial x-directional position offsets of the moving platform, and a trial was counted as successful only if the handover tolerance envelope was reached before the platform completed its trajectory. As summarized in Table 4, HEA-Planner sustains a 100% handover success rate up to 0.8 m/s (retaining 70% at 1.0 m/s). In contrast, PPO maintains 100% success only up to 0.5 m/s, beyond which steady-state tracking lag degrades the success rate to 60% at 0.8 m/s and 20% at 1.0 m/s.

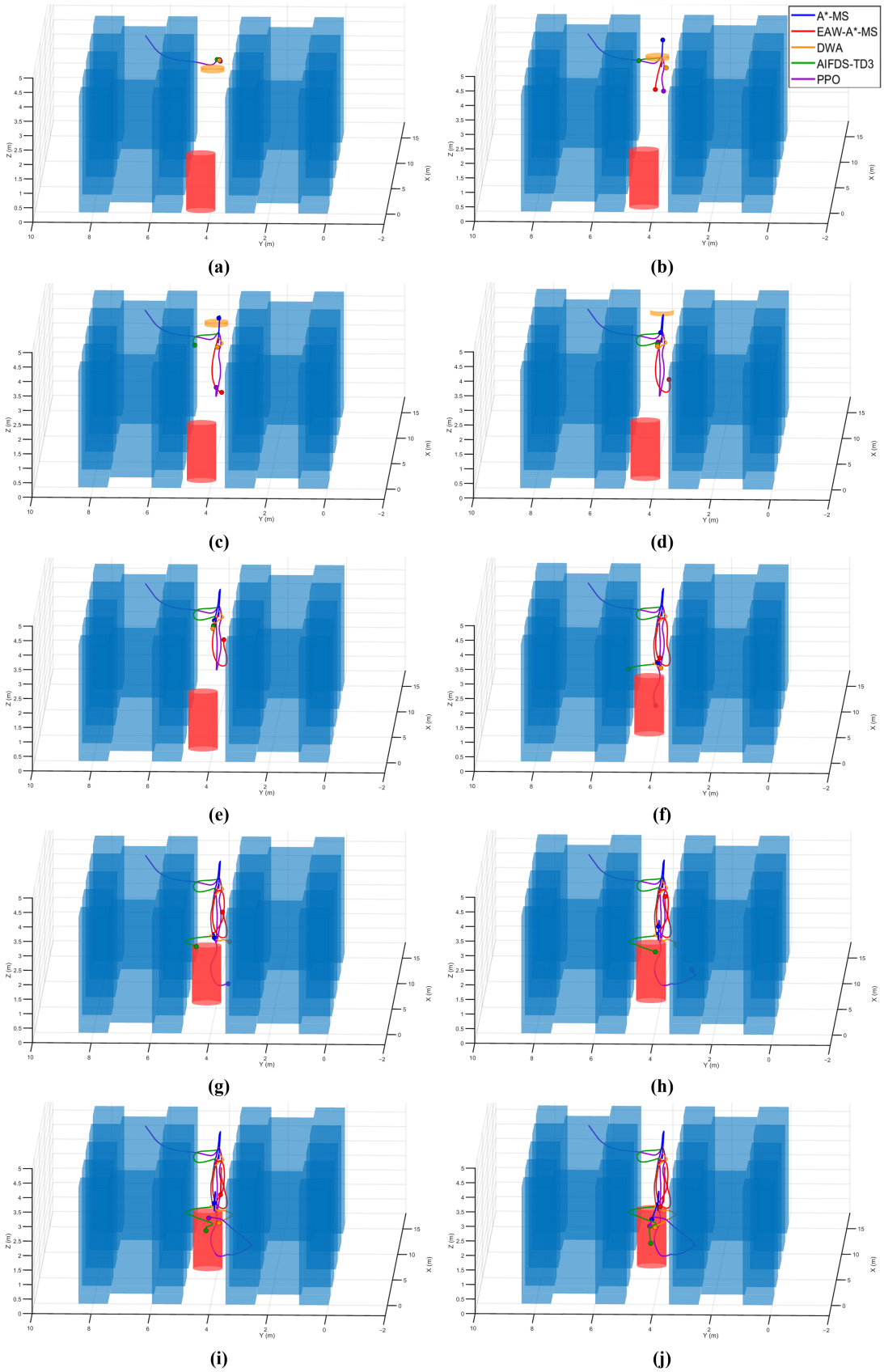


Figure 5. Spatiotemporal evolution snapshots of the dynamic obstacle avoidance process from $t = 9$ s to $t = 23$ s. (**a**–**j**) Sequential time-lapse snapshots showing reactive obstacle avoidance in the narrow corridor across five planning algorithms. Red and orange cylinders denote the ground dynamic obstacle (Obs 1) and airborne dynamic obstacle (Obs 2), respectively.

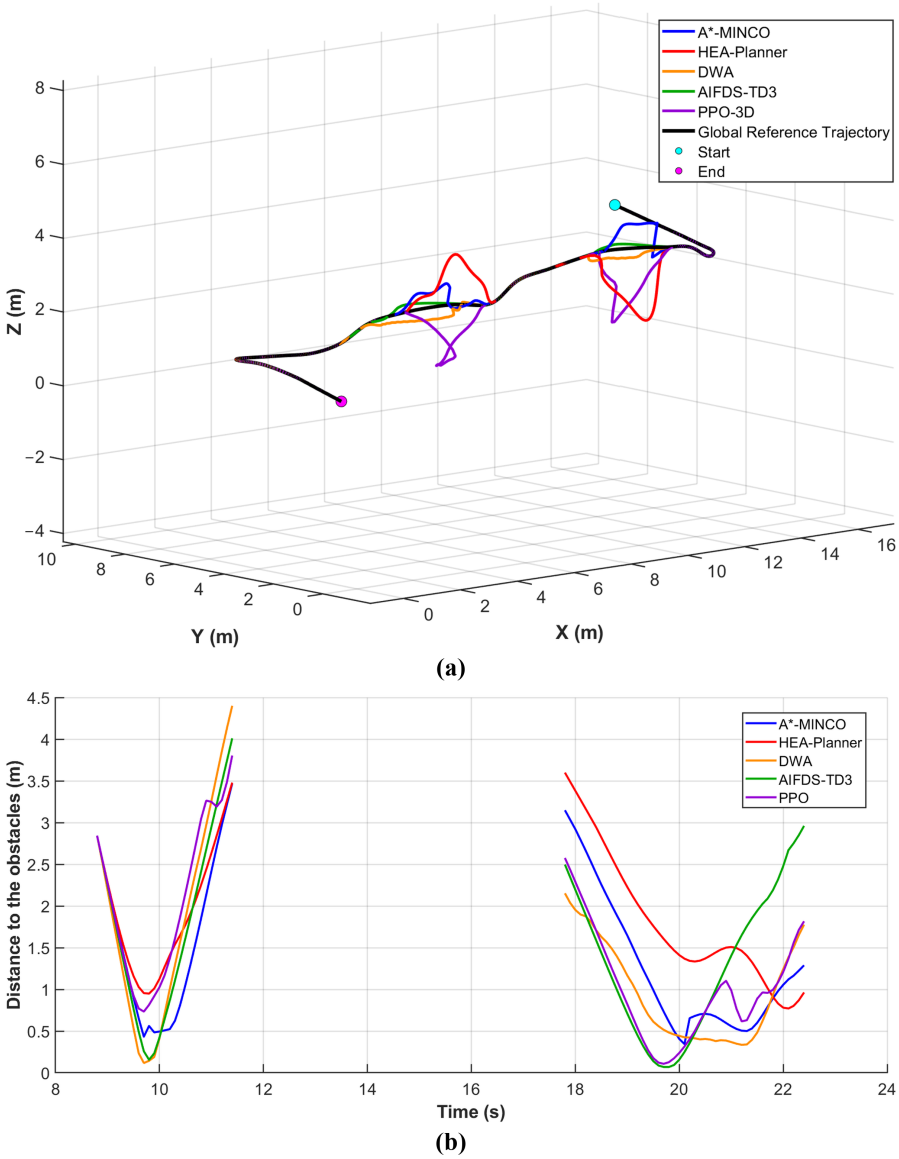


Figure 6. Quantitative comparison of dynamic obstacle avoidance trajectories. (**a**) Comparison of 3D obstacle avoidance trajectories. (**b**) Distance monitoring curves to dynamic obstacles.

As illustrated in Figure 7, under variable-speed platform motion up to 1.0 m/s, HEA-Planner first follows the globally planned reference trajectory and then, upon entering State S_3, synchronizes with the moving platform, reaching the terminal handover window at $t = 32.8$ s measured from the start of the simulation ($e_p = 0.19$ m, $e_v = 0.09$ m/s, average jerk 5.48 m/s^3). In contrast, at the exact moment HEA-Planner completes handover ($t = 32.8$ s), the benchmark PPO exhibits a position tracking lag of $e_p = 1.19$ m due to target acceleration; PPO experiences persistent tracking oscillations (average jerk 9.49 m/s^3). PPO fails to achieve dynamic handover during the platform motion phase, entering the tolerance envelope only at $t = 45.0$ s, by which point the platform has already come to rest. This improvement stems from the predictive attraction field in EA-A* and the duration gradient coupling in STC-MINCO: embedding target motion priors into the analytical duration derivative allows STC-MINCO to adjust trajectory geometry and duration T_Σ concurrently. As a result, the planner achieved terminal handover with an average replanning time of 34.4 ms.

Table 4. Performance comparison of moving-platform target synchronization in Simulation III ($N = 10$ trials per speed level).

Metric	HEA-Planner	PPO
(a) Constant-speed sweep: success rate ($n = 10$ per level)		
Success rate at 0.3 m/s (%)	100	100
Success rate at 0.5 m/s (%)	100	100
Success rate at 0.8 m/s (%)	100	60
Success rate at 1.0 m/s (%)	70	20
(b) Variable-speed benchmark (0.5 $\sim$ 1.0 m/s cosine)		
Time to reach handover envelope (s)	32.8	45.0
Position error at HEA-Planner's handover instant (m)	0.19	1.19
Velocity error at HEA-Planner's handover instant (m/s)	0.09	0.17
Average jerk (m/s^3)	5.48	9.49
Average computation time (ms)	34.4	21.6

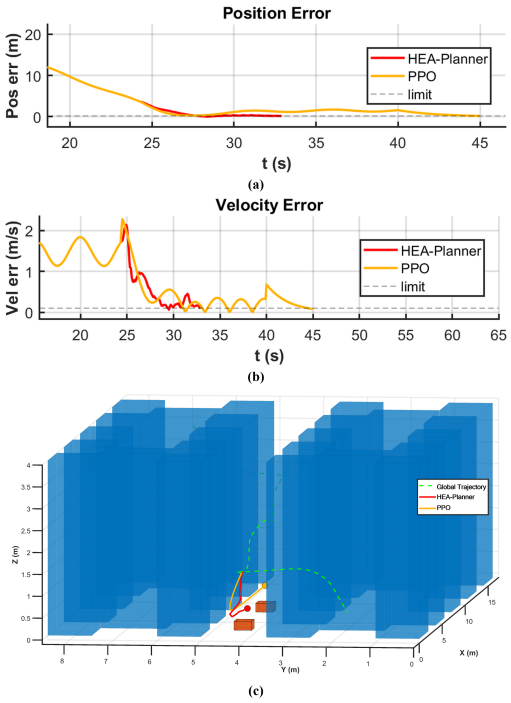


Figure 7. Tracking errors and trajectory comparison during moving-platform target synchronization and visual handover (State S_3). (**a**) Handover position error curve. (**b**) Handover velocity error curve. (**c**) Trajectory comparison. The orange rectangular prism indicates the moving landing platform (UGV).

4.3. Ablation Study

To isolate the contribution of each algorithmic component to navigation safety and moving-platform synchronization, ablation experiments were conducted across the three operational finite state machine (FSM) states: Global Approaching (S_1), Dynamic Avoiding (S_2), and Target Synchronizing (S_3). Using the same environment, platform and sensor settings as Simulations I–III, each state was evaluated over $N = 20$ randomized trials. The randomization is state-specific. In S_1, the start and goal positions are randomly sampled on opposing sides of the warehouse to ensure each trial traverses the full aisle network. In S_2, obstacle speeds are independently sampled from $[0.5, 2.0]$ m/s along the central corridor. In S_3, random seeds determine the moving-platform velocity profile as described below. All variants within each state were evaluated on identical seeded task instances; Table 5 summarizes the quantitative results. In State S_3, trials used the Simula-

tion III sensor noise ($\sigma_{\text{noise}} = 0.05$ m), handover altitude (0.5 m), and tolerance envelope, together with a seeded cosine velocity profile of period 4.0 s and mean speed 0.75 m/s. The per-seed velocity amplitude was randomized within $[0.15, 0.40]$ m/s, giving instantaneous speeds of 0.35∼1.15 m/s, and the initial x-directional position offset of the platform was drawn from $[4, 8]$ m. A trial was counted as successful only if the handover tolerance was reached before the platform completed its trajectory.

Table 5. Quantitative ablation results of HEA-Planner across operational FSM states ($N = 20$ randomized trials per state).

State S_1: Global Approaching (EA-ACO + SG-MINCO)					
Variant Configuration	**Min Clear (m)**	**ΔLength (m)**	**Avg Turn (°)**	**Comp. Time (s)**	**Success Rate**
(1) Standard ACO + Standard MINCO	0.19 ± 0.03	-1.29	51.3 ± 7.9	40.8 ± 7.9	55.0%
(2) EA-ACO w/o Turning Factor f^{ψ} + SG-MINCO	0.43 ± 0.06	$+0.32$	63.5 ± 16.8	39.1 ± 6.8	100.0%
(3) EA-ACO w/o Obstacle Factor f^{d} + SG-MINCO	0.31 ± 0.08	-0.24	17.1 ± 6.8	33.4 ± 2.8	100.0%
(4) EA-ACO + MINCO w/o Waypoint Guidance J_{guide}	0.28 ± 0.07	-0.53	23.6 ± 8.0	38.2 ± 4.4	90.0%
(5) Full Proposed: EA-ACO + SG-MINCO	0.40 ± 0.09	0 (18.94 ± 6.79)	23.6 ± 8.0	30.4 ± 5.3	100.0%
State S_2: Dynamic Avoiding (EA-A* + SG-MINCO)					
Variant Configuration	**Min Clear (m)**	**Flight Time (s)**		**Replan Time (ms)**	**Success Rate**
(6) EA-A* w/o Risk Field G_{risk} + SG-MINCO	0.11 ± 0.01	6.86 ± 0.08		29.0 ± 3.7	30.0%
(7) EA-A* w/o Velocity Inheritance + SG-MINCO	0.64 ± 0.05	7.56 ± 0.07		31.4 ± 4.4	100.0%
(8) EA-A* + MINCO w/o Waypoint Guidance J_{guide}	0.53 ± 0.05	7.62 ± 0.19		32.2 ± 8.8	100.0%
(9) Full Proposed: EA-A* + SG-MINCO	0.71 ± 0.05	7.41 ± 0.08		31.3 ± 4.5	100.0%
State S_3: Target Synchronizing (EA-A* + STC-MINCO)					
Variant Configuration	**Handover Pos Err (m)**	**Handover Vel Err (m/s)**	**Handover Time (s)**	**Replan Time (ms)**	**Success Rate**
(10) EA-A* + Fixed-Time MinSnap	1.61 ± 0.56	0.68 ± 0.40	11.21 ± 0.64	73.7 ± 24.6	20.0%
(11) EA-A* + STC-MINCO w/o Total Derivative	0.18 ± 0.02	0.08 ± 0.02	8.84 ± 0.58	34.4 ± 7.4	100.0%
(12) EA-A* w/o Predictive Attraction + STC-MINCO	0.19 ± 0.01	0.09 ± 0.01	9.04 ± 0.62	34.9 ± 6.5	100.0%
(13) Full Proposed: EA-A* + STC-MINCO	0.18 ± 0.01	0.08 ± 0.01	7.69 ± 0.74	32.7 ± 4.9	100.0%

In State S_1, disabling environmental heuristics (Variant 1) or the turning factor f^{ψ} (Variant 2) increases average turning angles to 51.3° (+117%) and 63.5° (+169%), while elevating search time (40.8 ± 7.9 s and 39.1 ± 6.8 s, respectively). Omitting the obstacle clearance factor f^{d} (Variant 3) or waypoint guidance J_{guide} (Variant 4) causes paths to hug shelf boundaries or cut corners around sharp turns, reducing minimum clearance to 0.31 m and 0.28 m (90% success rate in Variant 4). In contrast, the complete pipeline (Variant 5) achieves 0.40 ± 0.09 m clearance with a 100% success rate and a smooth turning profile ($23.6 \pm 8.0°$).

In State S_2, removing the risk repulsion field G_{risk} (Variant 6) reduces dynamic clearance to 0.11 m and drops the success rate to 30.0%. Without proactive repulsion, the quadrotor continues along a straight trajectory rather than diverting early into lateral escape bays. Disabling velocity inheritance (Variant 7) or waypoint guidance (Variant 8) degrades dynamic clearance to 0.64 m and 0.53 m. The full pipeline (Variant 9) preserves 0.71 ± 0.05 m clearance at a 31.3 ± 4.5 ms replanning latency.

In State S_3, each trial begins at the transition into the synchronizing state and terminates once the relative states enter the handover tolerance envelope to transition into visual servoing; the elapsed time between these two events is reported as the handover time. Except for the fixed-arrival-time baseline (Variant 10, pre-estimating duration without joint time optimization), which suffers an 80.0% failure rate with severe divergence ($e_p = 1.61 \pm 0.56$ m) due to motor thrust violations under platform acceleration, all remaining configurations (Variants 11–13) consistently achieve successful handover ($e_p \approx 0.18 \sim 0.19$ m, $e_v \approx 0.08 \sim 0.09$ m/s), making handover time the primary discriminator of synchronization efficiency. Under joint time optimization, omitting analytical total-derivative coupling (Variant 11) neglects target velocity gradients, extending handover time to 8.84 ± 0.58 s (+15.0%). Omitting predictive attraction (Variant 12) leads to a tail-chasing pursuit profile, prolonging handover time to 9.04 ± 0.62 s (+17.6%). Fast closed-loop replanning compensates for open-loop gradient inaccuracies across cycles, allowing Variants 11 and 12 to maintain 100% success at the expense of convergence speed. The full STC-MINCO formulation (Variant 13) achieves synchronization in the shortest handover time of 7.69 ± 0.74 s with an average replanning time of 32.7 ± 4.9 ms.

4.4. Experimental Validation

To validate the real-world closed-loop feasibility and real-time execution of the proposed framework on the onboard computer, physical flight experiments were conducted in an $8.0 \times 5.0 \times 3.0$ m indoor flight arena across two scenarios: static obstacle avoidance landing (Scenario 1) and autonomous landing on a moving platform (Scenario 2). In Scenario 1, the static obstacles were arranged to form a confined corridor with a clear spacing of 1.5 m between adjacent obstacles. Following the two-stage landing strategy, Stage 1 employed HEA-Planner to guide the quadrotor around obstacles and achieve spatiotemporal state synchronization within a visual handover window (0.5 m above the landing platform, tolerances $e_p < 200$ mm and $e_v < 0.1$ m/s), after which Stage 2 executed visual servoing for physical touchdown. The cooperative UGV carries a 600×600 mm landing platform with a central 200×200 mm AprilTag marker. At this handover height, the downward gimbal camera (85° FOV) yields a ground footprint of ≈0.9 m diameter, ensuring full marker capture within the handover envelope. A trial is considered successful if the quadrotor remains collision-free, enters the handover window, and completes touchdown within 200 mm of the deck center.

The experimental vehicle is a quadrotor UAV with a takeoff mass of 1.80 kg, a diagonal wheelbase of 380 mm, and outer dimensions of $420 \times 420 \times 250$ mm. The hardware setup comprises an onboard computer (NVIDIA Jetson Orin NX) running Ubuntu 20.04 and ROS Noetic, a Pixhawk 2.4.8 flight controller, a downward-facing gimbal camera, and a UWB transceiver module. Prior to flight, the static environment was mapped using a handheld LiDAR device and stored as a 3D voxel grid map for planning. For trajectory tracking, the onboard computer runs HEA-Planner at 20 Hz to plan continuous trajectories, transmitting reference states with velocity feedforward to the Pixhawk flight controller for inner position (100 Hz) and attitude (250 Hz) tracking. In Scenario 2, a wheeled UGV ($800 \times 800 \times 350$ mm) acts as the moving platform, following a continuous cosine velocity profile (0.3∼0.8 m/s, period 4.0 s). The UAV estimates the moving platform's state via an onboard EKF that fuses relative UWB measurements (RMSE ≈ 0.05 m) with UGV odometry and IMU telemetry; once detected, AprilTag visual poses are incorporated into the filter, achieving an estimation RMSE of 0.03 m. Target motion was extrapolated via the second-order Taylor model (Equation (22)). Ground-truth trajectories were recorded by an optical motion-capture system. To evaluate the sim-to-real gap and trajectory feasibility, trajectory tracking errors were computed offline as an external evaluation metric by comparing the optical motion-capture ground-truth flight paths against the onboard planned trajectories. Note that Stage 1

visual handover is independently triggered by onboard relative state estimation rather than external motion-capture tracking metrics. Figure 8a,b show the experimental setups, Figure 8c,d present the planned and executed trajectories, and quantitative results across repeated trials ($N = 5$) are summarized in Table 6. Terminal touchdown error is defined as the planar distance $e_{p,\text{touchdown}} = \sqrt{\Delta x^2 + \Delta y^2}$ from the platform center at touchdown.

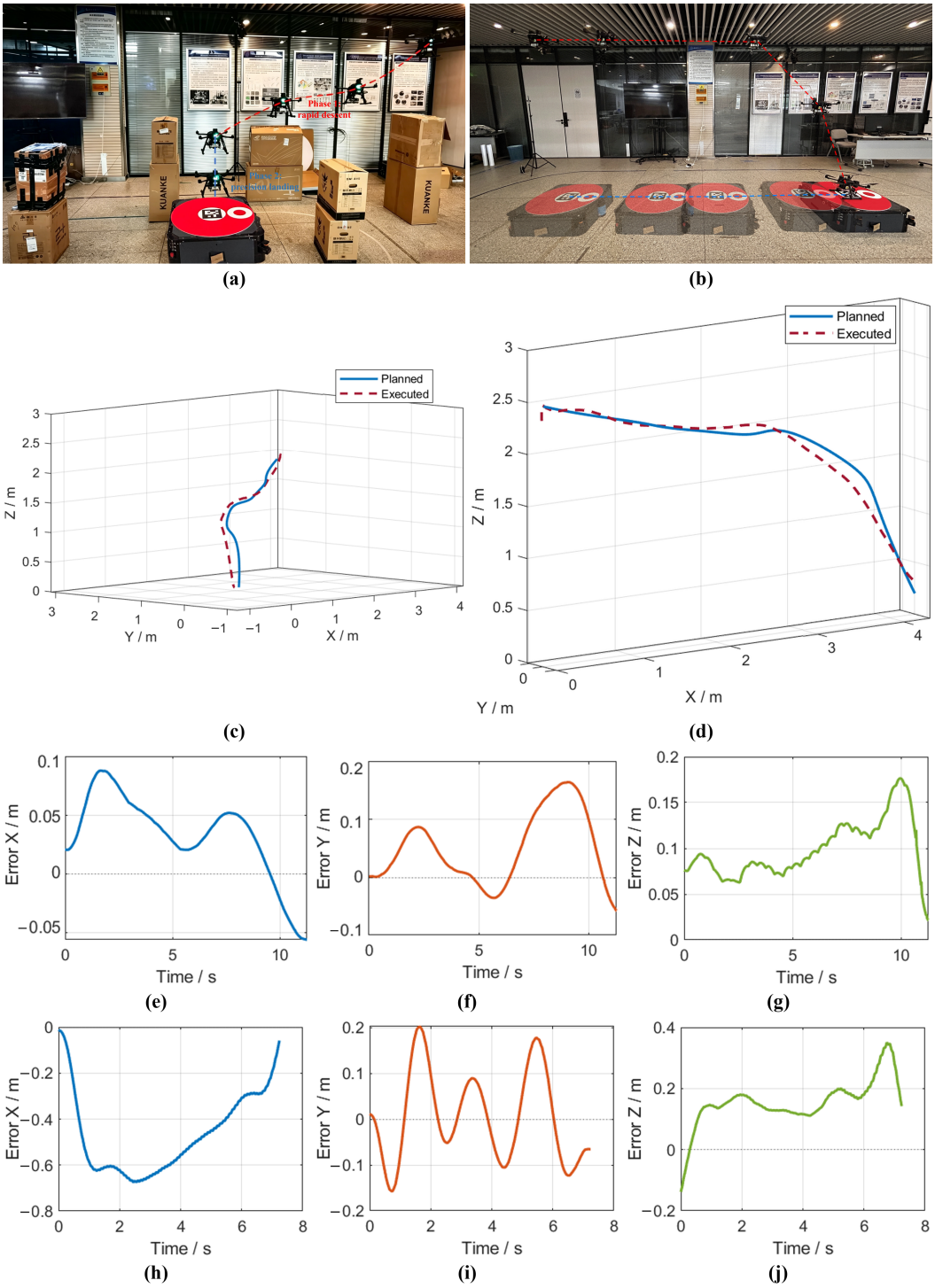


Figure 8. Flight experiments and quantitative tracking performance in two test scenarios. (**a**,**b**) Experimental flight setups and composite photographs in Scenario 1 and Scenario 2. (**c**,**d**) 3D planned and executed flight trajectories in Scenario 1 and Scenario 2. (**e**–**g**) Stage 1 tracking error curves along the *X*-, *Y*-, and *Z*-axes in Scenario 1. (**h**–**j**) Stage 1 tracking error curves along the *X*-, *Y*-, and *Z*-axes in Scenario 2.

Table 6. Quantitative results of physical flight experiments across repeated trials ($N = 5$ per scenario).

Scenario	Metric	T1	T2	T3	T4	T5	Overall
Scenario 1 (Static)	Min Clearance $d_{\min}$ (m)	0.68	0.89	1.04	0.61	0.52	0.75 ± 0.21
	Stage 1 Handover Time (s)	13.3	12.6	11.7	15.4	12.4	13.1 ± 1.4
	Terminal Touchdown Error (mm)	91.8	70.2	112.4	103.6	80.6	91.7 ± 17.0
	Success Rate (%)	100%	100%	100%	100%	100%	100% (5/5)
Scenario 2 (Moving Platform)	Stage 1 Handover Time (s)	7.3	8.6	7.8	6.4	9.0	7.8 ± 1.0
	Avg Replan Time (ms)	35.6	31.5	39.8	34.6	26.7	33.6 ± 4.9
	Terminal Touchdown Error (mm)	101.5	176.4	161.0	185.6	116.3	148.2 ± 37.3
	Success Rate (%)	100%	100%	100%	100%	100%	100% (5/5)

In Scenario 1, the quadrotor maintained a minimum obstacle clearance of 0.75 ± 0.21 m (0.52∼1.04 m, Table 6). Stage 1 handover time, measured from the start of the onboard flight program, averaged 13.1 ± 1.4 s (11.7∼15.4 s). Terminal touchdown error was 91.7 ± 17.0 mm, within the landing platform boundary across all trials (100% success rate). As shown in the planned and executed trajectories (Figure 8c) and Stage 1 tracking errors (Figure 8e–g), tracking deviations under SG-MINCO remained below 0.2 m along all three axes and converged near zero before visual handover.

In Scenario 2, the quadrotor achieved Stage 1 visual handover within 7.8 ± 1.0 s (6.4∼9.0 s, Table 6), while STC-MINCO replanning averaged 33.6 ± 4.9 ms (26.7∼39.8 ms), well within the 50 ms budget for 20 Hz closed-loop execution. Stage 2 visual servoing then completed touchdown with a terminal touchdown error of 148.2 ± 37.3 mm (101.5∼185.6 mm, 100% success rate). Figure 8d,h–j show the executed trajectory and Stage 1 tracking errors. A transient tracking lag of up to 0.67 m occurred along the longitudinal flight axis during moving-platform maneuvers (Figure 8h), primarily caused by UAV localization state estimation errors during maneuvers, as well as controller response delay. Flight safety is backed by the Tier 2 emergency-braking fallback, which triggers an active hover if clearance breaches the safety threshold. Throughout the flight, lateral tracking error remained within $|e_y| \leq 0.20$ m (Figure 8i). Before entering the visual handover window, tracking errors along both horizontal axes converged below 0.06 m. All five trials satisfied the handover criteria and completed successful touchdown.

4.5. Discussion on Limitations in Real-World Applications

While experimental evaluations demonstrate the feasibility of HEA-Planner in laboratory and simulated warehouse environments, several practical challenges remain for industrial deployment. First, onboard sensors with a limited field of view (FOV) are susceptible to blind-corner occlusions in densely packed warehouse aisles, which can shorten the safe reaction window for local receding-horizon replanning. Second, severe non-line-of-sight (NLOS) conditions and multi-path reflections caused by metallic shelf structures indicate that industrial applications will need to replace UWB-only tracking with robust multi-modal SLAM architectures. Finally, agile flight through narrow logistics corridors induces unmodeled aerodynamic ground and wall effects, creating transient physical disturbances that challenge the conservative safety margin ($R_{\text{footprint}}$) in tight workspaces. Accordingly, the collision-free performance reported here characterizes operation within an explicit operational envelope, defined by three physical conditions: (i) dynamic obstacles are detected with sufficient lead time for an evasive trajectory to be replanned before closest approach; (ii) the dynamic obstacle's relative velocity remains approximately piecewise-constant over a single replanning interval ($\Delta t = 50$ ms at 20 Hz); and (iii) closed-loop tracking deviations remain bounded within the planned clearance corridor. This defines the operational envelope within which the results of Section 4.2 are established. Overcoming these limitations will require integrating multi-sensor perception pipelines

with adaptive control laws, presenting an important research direction for autonomous air–ground collaboration in complex logistics environments.

5. Conclusions

This paper presents HEA-Planner, a hierarchical environment-aware trajectory planning framework for autonomous quadrotor landing on moving platforms in confined dynamic warehouse environments. Regulated by an FSM supervisor, the framework combines an EA-ACO global planner for reference paths, an EA-A* local replanner for dynamic obstacle avoidance, and a backend optimization layer with SG-MINCO and STC-MINCO for spatiotemporal trajectory generation. In this architecture, EA-ACO reduces wall-hugging via environment feedback factors to provide reference paths along corridor centerlines; EA-A* achieves dynamic responsiveness in a local sliding window by switching between risk repulsion and predictive attraction fields; and STC-MINCO propagates terminal boundary motion through analytical total derivatives to optimize trajectory geometry and arrival time concurrently. Simulation and physical flight experiments demonstrate that compared to standard ACO, EA-ACO reduces average turning angle by 57.7% and increases minimum obstacle clearance to more than 1.6 times that of standard ACO. In dynamic obstacle avoidance simulations, HEA-Planner achieved an average total replanning latency of 32.9 ms; in physical flight experiments on the moving platform, STC-MINCO replanning averaged 33.6 ± 4.9 ms, both well within the 50 ms budget for 20 Hz onboard closed-loop execution. For autonomous landing on the moving platform, the system achieved Stage 1 spatiotemporal state synchronization within the 200 mm handover tolerance envelope, followed by Stage 2 visual servoing yielding a terminal touchdown position error of 148.2 ± 37.3 mm on the moving-platform deck. Future work will focus on transitioning the framework to full-scale industrial warehouse settings, extending it to multi-UAV cooperative landing operations, and coordinating with UGV motion planning toward cooperative heterogeneous air–ground systems.

Author Contributions: Conceptualization, X.W., F.Q., P.L. and Z.D.; methodology, X.W. and F.Q.; software, F.Q. and J.C.; validation, F.Q. and J.C.; formal analysis, F.Q.; investigation, X.W., F.Q. and J.C.; resources, X.W., P.L. and S.D.; data curation, F.Q. and Z.D.; writing—original draft preparation, X.W., F.Q. and J.C.; writing—review and editing, X.W., P.L. and S.D.; visualization, F.Q. and Z.D.; supervision, X.W., P.L. and S.D.; project administration, X.W.; funding acquisition, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (General Program) [grant number 52475521].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: During the preparation of this manuscript, the authors used generative AI tools for English language editing and polishing. The authors have reviewed and edited the output and take full responsibility for the content of this publication. The authors would also like to thank the reviewers for their constructive comments and suggestions.

Conflicts of Interest: Author Sheng Dai was employed by Nanjing LES Information Technology Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Appendix A. Mathematical Derivations for SG-MINCO and STC-MINCO

Appendix A.1. Continuous-Time Augmented Lagrangian Formulation for Corridor and Dynamic Constraints

In the MINCO framework, a piecewise fifth-order polynomial trajectory $\mathbf{p}(t)$ in $\mathbb{R}^3$ over M segments is expressed for segment $j \in \{1, \ldots, M\}$ as:

$$\mathbf{p}_j(t) = \mathbf{c}_j^T \boldsymbol{\beta}_0(t - t_{j-1}), \quad t \in [t_{j-1}, t_j] \tag{A1}$$

where $\mathbf{c}_j \in \mathbb{R}^{6\times 3}$ is the polynomial coefficient matrix of segment j, and $\boldsymbol{\beta}_0(\tau) = [1, \tau, \tau^2, \tau^3, \tau^4, \tau^5]^T$ is the polynomial basis vector with local time $\tau = t - t_{j-1} \in [0, T_j]$. Higher-order time derivatives (velocity, acceleration, jerk, snap) are evaluated as $\mathbf{p}_j^{(k)}(t) = \mathbf{c}_j^T \boldsymbol{\beta}_k(\tau)$, where $\boldsymbol{\beta}_k(\tau) = \frac{d^k \boldsymbol{\beta}_0(\tau)}{d\tau^k}$.

Intermediate waypoints $\mathbf{q}_i = \mathbf{p}(t_i)$ $(i = 1, \ldots, M-1)$ are confined within their respective bounding boxes by the tanh mapping in Equation (26). Continuous corridor containment ($\mathbf{p}(t) \in \mathcal{F}_j$) and kinematic limits (velocity, thrust, and tilt angle) along the curve are enforced as hard inequality constraints via the Powell–Hestenes–Rockafellar (PHR) augmented Lagrangian method (ALM) [37].

For segment j, corridor containment is defined by K_j linear inequalities $\mathbf{a}_{j,m}^T \mathbf{p} \le b_{j,m} - r_{\text{corridor}}$ $(m = 1, \ldots, K_j)$, compactly written as $A_j \mathbf{p} \le \mathbf{b}_j - r_{\text{corridor}} \mathbf{1}$ with all-ones vector $\mathbf{1} = [1, \ldots, 1]^T$, where each hyperplane normal is normalized ($\|\mathbf{a}_{j,m}\| = 1$) and $r_{\text{corridor}} = R_{\text{footprint}} + \delta_{\text{margin}}$ provides an inward safety margin. The corresponding scalar violation function is:

$$c_{j,m}^{\text{corr}}(\mathbf{p}_j(\tau)) = \mathbf{a}_{j,m}^T \mathbf{p}_j(\tau) - b_{j,m} + r_{\text{corridor}} \le 0 \tag{A2}$$

Dynamic feasibility limits along the trajectory are formulated as inequality constraints:

$$c_v(\dot{\mathbf{p}}_j(\tau)) = \|\dot{\mathbf{p}}_j(\tau)\|^2 - v_{\max}^2 \le 0, \quad c_f(\ddot{\mathbf{p}}_j(\tau)) = \|\ddot{\mathbf{p}}_j(\tau) + g\mathbf{e}_z\|^2 - \left(\frac{f_{\max}}{m}\right)^2 \le 0 \tag{A3}$$

For a scalar inequality constraint $c_i(x) \le 0$ associated with dual Lagrange multiplier $\mu_i \ge 0$ and quadratic penalty factor $\rho_i > 0$, the PHR augmented Lagrangian kernel is defined as:

$$\Psi(c_i, \mu_i, \rho_i) = \begin{cases} \mu_i c_i + \frac{\rho_i}{2} c_i^2, & \text{if } \mu_i + \rho_i c_i \ge 0 \\ -\frac{\mu_i^2}{2\rho_i}, & \text{if } \mu_i + \rho_i c_i < 0 \end{cases} = \frac{1}{2\rho_i}\left[\max(0, \mu_i + \rho_i c_i)^2 - \mu_i^2\right] \tag{A4}$$

The derivative of the PHR kernel with respect to the constraint function c_i is continuous (C^1):

$$\frac{\partial \Psi(c_i, \mu_i, \rho_i)}{\partial c_i} = \max(0,\ \mu_i + \rho_i c_i). \tag{A5}$$

This formulation eliminates gradient discontinuities across the feasibility boundary.

To evaluate constraint integrals along continuous polynomial curves, each segment interval $[0, T_j]$ is discretized into K_{quad} quadrature points $\tau_\kappa = \kappa \frac{T_j}{K_{\text{quad}}}$ $(\kappa \in \{0, \ldots, K_{\text{quad}}\})$ via Simpson's quadrature with weights ω_κ. The augmented Lagrangian functional in Equation (27) is approximated as:

$$J_{\text{ALM}} \approx \sum_{j=1}^{M} \sum_{\kappa=0}^{K_{\text{quad}}} \omega_\kappa \frac{T_j}{K_{\text{quad}}} \left[\sum_{m=1}^{K_j} \Psi\left(c_{j,m}^{\text{corr}}(\mathbf{p}_j(\tau_\kappa)), \mu_{j,m,\kappa}^{\text{corr}}, \rho_{\text{corr}}\right) + \sum_{d\in\{v,f\}} \Psi\left(c_d(\tau_\kappa), \mu_{d,j,\kappa}, \rho_{\text{dyn}}\right) \right] \tag{A6}$$

Constraint gradients with respect to coefficient matrices $\mathbf{c}_j$ and segment durations T_j are propagated via the chain rule into the MINCO adjoint system.

The ALM solver iterates between inner primal minimization and outer dual/penalty updates:

1. Inner minimization: At outer iteration k, the unconstrained objective $\mathcal{L}_{\mathrm{SG}}(\boldsymbol{\xi}, \tilde{\mathbf{T}}; \boldsymbol{\mu}^{(k)}, \boldsymbol{\rho}^{(k)})$ is minimized with respect to $(\boldsymbol{\xi}, \tilde{\mathbf{T}})$ using L-BFGS to a gradient tolerance $\|\nabla \mathcal{L}\| \leq \epsilon_g$.
2. Dual multiplier update: For each constraint point i, the Lagrange multiplier is updated via first-order dual ascent:

$$\mu_i^{(k+1)} = \max\left(0,\ \mu_i^{(k)} + \rho_i^{(k)} c_i(x^{(k+1)})\right) \tag{A7}$$

3. Penalty parameter update: Let $v^{(k+1)} = \max_i \max(0, c_i(x^{(k+1)}))$ denote the maximum constraint violation. If the violation does not decrease sufficiently ($v^{(k+1)} > \eta_{\mathrm{dec}} v^{(k)}$), the penalty parameter is inflated:

$$\rho_i^{(k+1)} = \begin{cases} \gamma_{\mathrm{infl}} \rho_i^{(k)}, & \text{if } v^{(k+1)} > \eta_{\mathrm{dec}} v^{(k)} \\ \rho_i^{(k)}, & \text{otherwise} \end{cases} \tag{A8}$$

   with inflation ratio γ_{infl}.
4. Stopping tolerance and warm-start: The outer loop terminates when $v^{(k+1)} \leq \epsilon_{\mathrm{feas}}$ or when the outer iteration cap N_{outer} is reached. In the online replanning pipeline, dual multipliers $\boldsymbol{\mu}$ and penalty weights $\boldsymbol{\rho}$ are warm-started from the preceding cycle. Because the UAV state changes smoothly across successive replanning intervals, the warm-started multipliers are close to the optimal dual solution; the solver typically reaches feasibility within $1 \sim 2$ outer iterations without the numerical ill-conditioning of pure penalty formulations.

The values of the ALM parameters adopted in this work are $\epsilon_g = 10^{-4}$, $\eta_{\mathrm{dec}} = 0.25$, $\gamma_{\mathrm{infl}} = 2.5$, $\epsilon_{\mathrm{feas}} = 10^{-3}$, $N_{\mathrm{outer}} = 2$, and $K_{\mathrm{quad}} = 16$.

Appendix A.2. Analytical Total-Derivative Derivation for STC-MINCO

In the minimum-jerk MINCO parameterization, each segment $\mathbf{p}_j(\tau) = \sum_{k=0}^{5} \mathbf{c}_{j,k} \tau^k$ ($\tau \in [0, T_j]$, $j = 1, \ldots, M$) is defined by coefficient vectors $\mathbf{c}_{j,k} \in \mathbb{R}^3$. At each interior waypoint $\mathbf{q}_j$ ($j = 1, \ldots, M-1$), position pinning enforces $\mathbf{p}_j(T_j) = \mathbf{p}_{j+1}(0) = \mathbf{q}_j$. Applying the Euler-Lagrange equation to the unconstrained jerk objective $\int \|\mathbf{p}^{(3)}(t)\|^2 dt$ yields natural transversality conditions:

$$\mathbf{p}_j^{(k)}(T_j) = \mathbf{p}_{j+1}^{(k)}(0), \quad k \in \{1, 2, 3, 4\} \tag{A9}$$

establishing C^4 continuity across interior knots. Together with the three initial conditions ($\mathbf{s}_1(0) = \mathbf{s}_{\mathrm{stitch}}$) and three terminal conditions ($\mathbf{s}_M(T_M) = \mathbf{s}_f$), these $6(M-1)$ waypoint and junction conditions form a determined linear system of $6M$ equations:

$$\mathbf{M}(\mathbf{T})\mathbf{C} = \mathbf{B}(\mathbf{q}, \mathbf{s}_{\mathrm{init}}, \mathbf{s}_f) \tag{A10}$$

where $\mathbf{M}(\mathbf{T}) \in \mathbb{R}^{6M \times 6M}$ is a nonsingular banded matrix, $\mathbf{C} \in \mathbb{R}^{6M \times 3}$ is the coefficient matrix, and $\mathbf{B} \in \mathbb{R}^{6M \times 3}$ contains $\mathbf{s}_{\mathrm{init}} = \mathbf{s}_{\mathrm{stitch}}$, intermediate waypoints $\mathbf{q} = [\mathbf{q}_1, \ldots, \mathbf{q}_{M-1}]^T$, and terminal state $\mathbf{s}_f = \mathbf{s}_{\mathrm{target}}(T_\Sigma) = [\mathbf{p}_f^T, \mathbf{v}_f^T, \mathbf{a}_f^T]^T$.

For the cost functional $\mathcal{L}_{\text{STC}}(\mathbf{C}(\mathbf{q}, \mathbf{T}, \mathbf{s}_{\text{init}}, \mathbf{s}_f), \mathbf{T}, T_\Sigma) = J_{\text{MINCO}} + J_{\text{ALM}} + \lambda_{\text{guide}} J_{\text{guide}} + w_t T_\Sigma$, standard MINCO evaluates the gradient with respect to segment durations T_i holding the terminal state $\mathbf{s}_f$ fixed, defining the native partial derivative:

$$\left.\frac{\partial \mathcal{L}_{\text{STC}}}{\partial T_i}\right|_{\text{native}} = \text{Tr}\left(\left(\frac{\partial \mathcal{L}_{\text{STC}}}{\partial \mathbf{C}}\right)^T \left.\frac{\partial \mathbf{C}}{\partial T_i}\right|_{\mathbf{s}_f \text{ fixed}}\right) + \frac{\partial J_{\text{ALM}}}{\partial T_i} \tag{A11}$$

evaluated in $O(M)$ time via the banded adjoint system $\mathbf{M}(\mathbf{T})^T \mathbf{\Lambda} = \frac{\partial \mathcal{L}_{\text{STC}}}{\partial \mathbf{C}}$.

In STC-MINCO, the terminal boundary state is explicitly time-varying: $\mathbf{s}_f = \mathbf{s}_{\text{target}}(T_\Sigma)$, where $T_\Sigma = \sum_{j=1}^{M} T_j$. Applying the multivariate total-derivative chain rule, perturbing any segment duration T_i propagates through three distinct paths:

$$\frac{d\mathcal{L}_{\text{STC}}}{dT_i} = \left.\frac{\partial \mathcal{L}_{\text{STC}}}{\partial T_i}\right|_{\text{native}} + \left(\frac{\partial \mathcal{L}_{\text{STC}}}{\partial \mathbf{s}_f}\right)^T \frac{d\mathbf{s}_f}{dT_i} + \frac{\partial (w_t T_\Sigma)}{\partial T_i} \tag{A12}$$

Since T_Σ is the linear sum of all segment durations, it holds that:

$$\frac{\partial T_\Sigma}{\partial T_i} = \frac{\partial}{\partial T_i}\left(\sum_{j=1}^{M} T_j\right) = 1, \quad \forall i \in \{1, \ldots, M\} \tag{A13}$$

Consequently, differentiating the terminal rendezvous state yields:

$$\frac{d\mathbf{s}_f}{dT_i} = \frac{\partial \mathbf{s}_{\text{target}}(T_\Sigma)}{\partial T_\Sigma}\frac{\partial T_\Sigma}{\partial T_i} = \dot{\mathbf{s}}_{\text{target}}(T_\Sigma) \cdot 1 = \begin{bmatrix} \mathbf{v}_{\text{target}}(T_\Sigma) \\ \mathbf{a}_{\text{target}}(T_\Sigma) \\ \mathbf{j}_{\text{target}}(T_\Sigma) \end{bmatrix} \tag{A14}$$

Hence, each segment duration T_i receives an identical terminal state coupling gradient.

The adjoint sensitivity vector with respect to the terminal boundary condition is obtained directly from the terminal three rows of the adjoint matrix $\mathbf{\Lambda} \in \mathbb{R}^{6M \times 3}$, denoted as $\mathbf{\Lambda}_{\text{tail}} = [\boldsymbol{\lambda}_p, \boldsymbol{\lambda}_v, \boldsymbol{\lambda}_a]^T \in \mathbb{R}^{3\times 3}$. In the linear system $\mathbf{M}(\mathbf{T})\mathbf{C} = \mathbf{B}(\mathbf{q}, \mathbf{s}_{\text{init}}, \mathbf{s}_f)$ where $\mathbf{M} \in \mathbb{R}^{6M\times 6M}$ and $\mathbf{B} \in \mathbb{R}^{6M\times 3}$, the moving terminal boundary state $\mathbf{s}_f = [\mathbf{p}_f, \mathbf{v}_f, \mathbf{a}_f]^T \in \mathbb{R}^{3\times 3}$ constitutes precisely the final three rows of the right-hand matrix $\mathbf{B}$, satisfying $\mathbf{B}^T_{6M-2+k,:} = \mathbf{s}_{f,k+1}$ ($k \in \{0, 1, 2\}$, corresponding to $\mathbf{p}_f$, $\mathbf{v}_f$, and $\mathbf{a}_f$) with identity mapping $\frac{\partial \mathbf{B}^T_{6M-2+k,:}}{\partial \mathbf{s}_{f,k+1}} = \mathbf{I}_3$. By the adjoint identity $\mathbf{M}^T\mathbf{\Lambda} = \frac{\partial \mathcal{L}_{\text{STC}}}{\partial \mathbf{C}}$, the derivative with respect to terminal boundary conditions is given by:

$$\frac{\partial \mathcal{L}_{\text{STC}}}{\partial \mathbf{s}_f} = \begin{bmatrix} \boldsymbol{\lambda}_p \\ \boldsymbol{\lambda}_v \\ \boldsymbol{\lambda}_a \end{bmatrix} = \begin{bmatrix} \mathbf{\Lambda}^T_{6M-2,:} \\ \mathbf{\Lambda}^T_{6M-1,:} \\ \mathbf{\Lambda}^T_{6M,:} \end{bmatrix} \in \mathbb{R}^9 \tag{A15}$$

where $\boldsymbol{\lambda}_p, \boldsymbol{\lambda}_v, \boldsymbol{\lambda}_a \in \mathbb{R}^3$ correspond to the position, velocity, and acceleration terminal boundary equations, requiring $O(1)$ operations once $\mathbf{\Lambda}$ is solved.

Under the onboard EKF second-order Taylor prediction model (Equation (22)), target acceleration is treated as piecewise constant over the local horizon, giving $\mathbf{j}_{\text{target}}(T_\Sigma) \approx \mathbf{0}$. Substituting these relations into Equation (A12) yields Equation (29). Numerical consistency of both the terminal adjoint sensitivity $\mathbf{\Lambda}_{\text{tail}}$ and the total duration derivative $\frac{d\mathcal{L}_{\text{STC}}}{dT_i}$ was verified against central finite differences with perturbation step $\epsilon = 10^{-6}$. For the terminal adjoint sensitivity, the maximum relative errors are 1.25×10^{-9} (position), 2.19×10^{-9} (velocity), and 1.31×10^{-8} (acceleration), with a mean relative error of 2.23×10^{-9}. For the total duration derivatives across multi-segment configurations ($M \in \{4, 6, 8\}$), the maximum relative error is below 5.0×10^{-10} (mean relative error below 3.2×10^{-10}).

References

1. Wei, M.; Wang, L.; Xu, L.; Chang, Z. Trajectory Planning for Joint Communications and Tracking in UAV-UGV Enabled Air–Ground Collaborative System. *IEEE Trans. Netw. Sci. Eng.* **2026**, *13*, 6276–6292. [CrossRef]
2. Huang, T.; Chen, Y.; Chen, Z. Air-Ground Collaborative Inspection Path Planning for Heterogeneous Tasks. *Trans. Inst. Meas. Control* **2026**, *online ahead of print*. [CrossRef]
3. Kim, K.; Kim, S.; Kim, J.; Jung, H. Drone-Assisted Multimodal Logistics: Trends and Research Issues. *Drones* **2024**, *8*, 468. [CrossRef]
4. Beul, M.; Droeschel, D.; Nieuwenhuisen, M.; Quenzel, J.; Houben, S.; Behnke, S. Fast Autonomous Flight in Warehouses for Inventory Applications. *IEEE Robot. Autom. Lett.* **2018**, *3*, 3121–3128. [CrossRef]
5. Chaffilla, R.; Alvito, P.; Basiri, M. Collaborative Infrastructure-Free Aerial–Ground Robotic System for Warehouse Inventory Data Capture. *Drones* **2025**, *9*, 792. [CrossRef]
6. Morales, J.; Castelo, I.; Serra, R.; Lima, P.U.; Basiri, M. Vision-Based Autonomous Following of a Moving Platform and Landing for an Unmanned Aerial Vehicle. *Sensors* **2023**, *23*, 829. [CrossRef] [PubMed]
7. Pore, E.; Patle, B.K.; Thorat, S.; Patel, B. Drone-Enabled Practices in Modern Warehouse Management: A Comprehensive Review. *Drones* **2026**, *10*, 189. [CrossRef]
8. Ding, J.; Zhang, H.T.; Jiang, W.; Liu, X.; Hu, B.B. Mutual-Rendezvous Control and Feature-Compatible Landing Optimization for Heterogeneous UAV–USV Fleets. *IEEE Trans. Ind. Inform.* **2026**, *22*, 348–359. [CrossRef]
9. Semerikov, S.O.; Nechypurenko, P.P.; Vakaliuk, T.A.; Mintii, I.S.; Kolhatin, A.O. Vision-Based Autonomous UAV Landing: A Comprehensive Review of Technologies, Techniques, and Applications. *J. Intell. Robot. Syst.* **2025**, *111*, 115. [CrossRef]
10. Falanga, D.; Zanchettin, A.; Simovic, A.; Delmerico, J.; Scaramuzza, D. Vision-Based Autonomous Quadrotor Landing on a Moving Platform. In Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017; pp. 200–207. [CrossRef]
11. Wang, P.; Wang, C.; Wang, J.; Meng, M.Q.H. Quadrotor Autonomous Landing on Moving Platform. *Procedia Comput. Sci.* **2022**, *209*, 40–49. [CrossRef]
12. Ghommam, J.; Saad, M. Autonomous Landing of a Quadrotor on a Moving Platform. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 1504–1519. [CrossRef]
13. Zhang, Y.; Wu, Z.; Wei, T. Precise Landing on Moving Platform for Quadrotor UAV via Extended Disturbance Observer. *IEEE Trans. Intell. Veh.* **2025**, *10*, 2468–2478. [CrossRef]
14. Feng, Y.; Zhang, C.; Baek, S.; Rawashdeh, S.; Mohammadi, A. Autonomous Landing of a UAV on a Moving Platform Using Model Predictive Control. *Drones* **2018**, *2*, 34. [CrossRef]
15. Panjavarnam, K.; Ismail, Z.H.; Tang, C.H.H.; Sekiguchi, K.; Casas, G.G. Model Predictive Control for Autonomous UAV Landings: A Comprehensive Review of Strategies, Applications and Challenges. *J. Eng.* **2025**, *2025*, e70085. [CrossRef]
16. Aikins, G.; Jagtap, S.; Nguyen, K.D. A Robust Strategy for UAV Autonomous Landing on a Moving Platform under Partial Observability. *Drones* **2024**, *8*, 232. [CrossRef]
17. Amendola, J.; Cenkeramaddi, L.R.; Jha, A. Drone Landing and Reinforcement Learning: State-of-Art, Challenges and Opportunities. *IEEE Open J. Intell. Transp. Syst.* **2024**, *5*, 520–539. [CrossRef]
18. Wang, C.; Wang, J.; Wei, C.; Zhu, Y.; Yin, D.; Li, J. Vision-Based Deep Reinforcement Learning of UAV-UGV Collaborative Landing Policy Using Automatic Curriculum. *Drones* **2023**, *7*, 676. [CrossRef]
19. Long, T.; Cao, Y.; Xu, G.; Meng, Z.; Sun, J.; Wang, Z. Real-Time Multi-Quadrotor Trajectory Generation via Distributed Receding Architecture and Hierarchical Planning in Complex Environments. *Isa Trans.* **2023**, *136*, 715–726. [CrossRef] [PubMed]
20. Zhou, B.; Gao, F.; Wang, L.; Liu, C.; Shen, S. Robust and Efficient Quadrotor Trajectory Generation for Fast Autonomous Flight. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3529–3536. [CrossRef]
21. Yang, L.; Qi, J.; Xiao, J.; Yong, X. A Literature Review of UAV 3D Path Planning. In Proceedings of the 11th World Congress on Intelligent Control and Automation, Shenyang, China, 27–30 June 2014; pp. 2376–2381. [CrossRef]
22. Aggarwal, S.; Kumar, N. Path Planning Techniques for Unmanned Aerial Vehicles: A Review, Solutions, and Challenges. *Comput. Commun.* **2020**, *149*, 270–299. [CrossRef]
23. Peng, T.; Chen, Z.; Zhou, Y. A RRT Path Planning Algorithm Based on A* for UAV. In *Proceedings of the 4th International Conference on Informatics Engineering & Information Science (ICIEIS2021), Tianjin, China, 14 February 2022*; SPIE: Bellingham, WA, USA, 2022; Volume 12161, pp. 34–39. [CrossRef]
24. Brand, M.; Masuda, M.; Wehner, N.; Yu, X.H. Ant Colony Optimization Algorithm for Robot Path Planning. In Proceedings of the 2010 International Conference on Computer Design and Applications (ICCDA), Qinhuangdao, China, 25–27 June 2010; pp. 436–440. [CrossRef]
25. Tanaya, P.I.; Mahdiana, D. Ant Colony Algorithms for Mobile Robot Path Planning: A State-of-the-Art Review. *ROTASI* **2022**, *24*, 7–13.

26. Ferguson, D.; Stentz, A. Using Interpolation to Improve Path Planning: The Field D* Algorithm. *J. Field Robot.* **2006**, *23*, 79–101. [CrossRef]
27. Bai, X.; Jiang, H.; Cui, J.; Lu, K.; Chen, P.; Zhang, M. UAV Path Planning Based on Improved A∗ and DWA Algorithms. *Int. J. Aerosp. Eng.* **2021**, *2021*, 4511252. [CrossRef]
28. Byun, W.; Huh, S.; Jang, H.; Yu, S.; Lim, S.; Lee, S.; Nam, W. Vector Field-Based Robust Quadrotor Landing on a Moving Ground Platform. *Aerospace* **2025**, *12*, 590. [CrossRef]
29. Gu, J.; Wang, Y. Multi-UAV Formation Control via Deep Reinforcement Learning and Multi-Step Experience Storage in Dense Obstacle Environments. *Trans. Inst. Meas. Control* **2026**, *48*, 4202–4214. [CrossRef]
30. Ladosz, P.; Mammadov, M.; Shin, H.; Shin, W.; Oh, H. Autonomous Landing on a Moving Platform Using Vision-Based Deep Reinforcement Learning. *IEEE Robot. Autom. Lett.* **2024**, *9*, 4575–4582. [CrossRef]
31. Gao, F.; Wang, L.; Zhou, B.; Zhou, X.; Pan, J.; Shen, S. Teach-Repeat-Replan: A Complete and Robust System for Aggressive Flight in Complex Environments. *IEEE Trans. Robot.* **2020**, *36*, 1526–1545. [CrossRef]
32. Mellinger, D.; Kumar, V. Minimum Snap Trajectory Generation and Control for Quadrotors. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 2520–2525. [CrossRef]
33. Sun, W.; Sun, P.; Ding, W.; Zhao, J.; Li, Y. Gradient-Based Autonomous Obstacle Avoidance Trajectory Planning for B-spline UAVs. *Sci. Rep.* **2024**, *14*, 14458. [CrossRef] [PubMed]
34. Zhou, X.; Wang, Z.; Ye, H.; Xu, C.; Gao, F. EGO-Planner: An ESDF-Free Gradient-Based Local Planner for Quadrotors. *IEEE Robot. Autom. Lett.* **2021**, *6*, 478–485. [CrossRef]
35. Wang, Z.; Zhou, X.; Xu, C.; Gao, F. Geometrically Constrained Trajectory Optimization for Multicopters. *IEEE Trans. Robot.* **2022**, *38*, 3259–3278. [CrossRef]
36. Metu, C.; Zou, T. Review of Vision-Based AI Approaches for Infrastructure Inspection Using Uncrewed Aircraft Systems (UAS). *Autom. Constr.* **2026**, *187*, 106976. [CrossRef]
37. Nocedal, J.; Wright, S.J. *Numerical Optimization*, 2nd ed.; Springer: New York, NY, USA, 2006. [CrossRef]
38. Zhang, Y.; Wang, H. Adaptive Interfered Fluid Dynamic System Algorithm Based on Deep Reinforcement Learning Framework. In *Proceedings of the 2021 International Conference on Autonomous Unmanned Systems (ICAUS 2021)*; Wu, M., Niu, Y., Gu, M., Cheng, J., Eds.; Springer: Singapore, 2022; pp. 1388–1397. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.