Construction Robotics (2024) 8:19
https://doi.org/10.1007/s41693-024-00135-9

ORIGINAL PAPER

Intelligent path planning for civil infrastructure inspection with multi-rotor aerial vehicles

Lanh V. Nguyen[1] · Trung H. Le[1] · Ignacio Torres Herrera[1] · Ngai M. Kwok[2] · Quang P. Ha[1]

Received: 20 April 2024 / Accepted: 22 July 2024

Abstract
This paper presents the development of algorithms for high-level control and intelligent path planning of multi-rotor aerial vehicles (MAVs) in the tasks of inspecting civil infrastructure. After revisiting the multicopter modeling, we describe the hierarchy of high-level control for MAVs and develop optimization algorithms for generating optimal paths and enabling automatic flight during inspection tasks, making use of the digital twin technology. A co-simulation framework is then established to simulate and evaluate inspection mission scenarios, integrating these essential components. Real-world examples from built infrastructure illustrate this concept. An advantage of this approach is its ability to rigorously test, validate, verify, and evaluate MAV operations under abnormal conditions without requiring physical implementation or field tests. This significantly reduces testing efforts throughout the development cycle, ensuring optimal cooperation, safety, smoothness, fault tolerance, and energy efficiency. The methodology is validated through simulations and real-world inspection of a monorail bridge.

Keywords Multi-rotor aerial vehicles · Intelligent path planning · Civil infrastructure inspection · Digital twin

1 Introduction

With the rapid development of technology in the fourth industrial revolution, multi-rotor aerial vehicle (MAV) systems have garnered widespread attention across civilian and industrial sectors. The construction industry has embraced MAVs as invaluable tools throughout various project stages, including site monitoring, 3D mapping, building and damage assessment, and package delivery logistics. For example, researchers have demonstrated that MAVs can generate highly accurate 3D maps of construction sites, enabling project managers to assess progress and identify potential issues in real-time (Bulgakov et al. 2020). Additionally, MAVs equipped with high-resolution cameras can conduct building inspections, facilitating the detection of structural damage invisible to the naked eye (Bolourian et al. 2017). Furthermore, exploring MAVs in package delivery logistics has shown promising potential for reducing delivery times and costs (Grzybowski et al. 2020). Consequently, the prospect of widespread MAV adoption within the construction industry and beyond is increasingly apparent. For these applications to achieve their full potential, it is essential to implement a robust and optimized strategy for the path planning of MAVs.

A variety of MAV path planning methods have appeared in the literature. Although traditional techniques, such as the A^* algorithm and Dijkstra's algorithm, prove effective in specific situations, their scalability decreases as the search space expands (Tang et al. 2021; Prasad and Ramkumar 2022). In contrast, strategies using sampling like rapidly exploring random trees (RRT) and probabilistic roadmaps (PRM) have the advantage of producing feasible flight paths. However, they may have difficulty meeting MAV maneuverability constraints with some errors between planned and actual paths, as discussed in Pharpatara et al. (2016). Additionally, methods, such as visibility graphs and potential fields, could generate smooth and continuous paths but face the challenge of local minima, as documented in Blasi et al. (2022) and Pan et al. (2021).

✉ Lanh V. Nguyen
vanlanh.nguyen@uts.edu.au

[1] School of Electrical and Data Engineering, University of Technology Sydney (UTS), 15 Broadway, Ultimo 2007, Australia

[2] School of Engineering, Design and Built Environment, Western Sydney University (WSU), Penrith, NSW 2751, Australia

Published online: 13 August 2024

Heuristic optimization methods, widely used to tackle complex optimization tasks, have become promising in robotic path planning. In this array, particle swarm optimization (PSO) mimics swarm behavior (Phung and Ha 2021), artificial bee colony (ABC) draws inspiration from honeybee foraging (Lin et al. 2022), and genetic algorithms (GAs) develop possible solutions (Kok et al. 2012). Standing out in this range, the grey wolf optimization (GWO) algorithm is an innovative approach inspired by hierarchical group hunting behavior observed in grey wolves (Mirjalili et al. 2014). As noted in a recent study, GWO requires relatively few parameters to tune with a good convergence rate, making it attractive for practical optimization problems (Makhadmeh et al. 2023). This advantageous flexibility positions GWO as a convenient choice of optimizing routes for civil infrastructure inspection, especially in settings characterized by numerous obstacles and constraints.

For completing complex tasks, such as large-scale 3D inspection, the collaboration of multiple MAVs flying in formation has become essential (Wu et al. 2021; Zou et al. 2020). When multiple drones work together in a team, there is a significant improvement in their performance. For example, when an MAV team is engaged in inspection, combining the field of view of all group members allows for a much more extensive coverage area than an individual MAV can achieve, leading to improved inspection performance (Hu et al. 2019). Consequently, promoting the collaboration of multiple MAVs is encouraged to optimize processing time and operational capabilities in the mentioned applications.

The main problem in MAV formation control lies in resolving potential conflicts and interactions among group members while maintaining a desired formation under various constraints during task execution. Game theory, a branch within applied mathematics, offers a robust framework for examining strategic interactions among decision-makers (Roger 1991). It provides insights into how individuals make choices to maximize their outcomes while considering the actions of others. In cooperative path planning for UAVs, game theory can be employed to identify optimal solutions that balance individual and collective objectives. Therefore, this study utilizes a game theory-based approach for MAV path planning in large-scale 3D inspection missions. Another issue for MAVs remains how to carry out tasks within the civil sector autonomously, which involves navigating through a complex flying environment fraught with various elements, such as cooperative constraints, travel distances, vehicle dynamics, and obstacles. A combination of these factors could pose a significant challenge. To achieve certification for reliability through MAV testing, the vehicle must cover extensive distances, even under challenging conditions, which are time-consuming and hazardous. Moreover, creating and replicating such situations in real life takes a lot of work. It is essential to note that accurate flight testing is only applicable to a specific mechanical, electrical, and software configuration. Any changes to these elements necessitate repeating the testing process. Consequently, simulations will play an essential role in testing most MAV features, and having a framework capable of integrating multiple technological components and validating design iterations is imperative during the initial phases of development.

The concept of digital twin, first introduced in Grieves (2005), has gained increasing attention in recent years as a technological solution to the problem above. Digital twins refer to virtual digital representations of physical systems that can imitate actual objects' behavior. This technology holds the potential to revolutionize numerous industries by providing accurate real-time data on the behavior of the physical system and allowing for the virtual testing of different scenarios before implementing changes in the real world. Digital twins have demonstrated success across various applications, including industrial robots (Kaigom and Roßmann 2020), self-driving cars (Hu et al. 2022), and aerospace systems (Li et al. 2021). For instance, digital twins can be used to simulate the performance of a new robot design before building the physical prototype. By testing and optimizing the digital twin, engineers can identify potential issues and make adjustments to improve the robot's performance, reliability, and safety. Similarly, digital twins can be applied in the aerospace industry to simulate the behavior of aircraft components and systems, allowing engineers to optimize designs and test various scenarios.

Recently, a digital twin framework was introduced in the MAV domain (Yang et al. 2021), featuring three key components: the virtual environment, the physical MAV, and the service center. Therein, bidirectional communication between physical and virtual MAVs could be enabled, facilitating assembly tasks and updating physical models. Moreover, a digital twin-based intelligent, cooperative architecture for MAV swarms was proposed (Lei et al. 2020), wherein the digital twin model accurately replicates the physical entity, the MAV swarm, with high fidelity. This allows for comprehensive monitoring for its processes and incorporates a machine learning algorithm to explore global optimal solutions and regulate swarm behaviors.

In civil engineering, the application of digital twins extends to architecture, infrastructure, machinery, and construction processes, as highlighted in Jiang et al. (2021). Leveraging modeling, simulation, calculation, analysis tools, and advanced algorithms from a virtual perspective fosters the development of innovative construction methods. For example, Jiang et al. (2022) introduced a digital twin-enabled intelligent modular integrated construction system featuring a demonstrating testbed for cooperative decision-making and on-site assembly. However, current efforts mainly focus on replicating digital structures and tracking building states. Including MAV systems, Zhang

et al. (2022) conducted a digital twin study of a building inspection scenario for data validation and reanimation. This research entails realistic system workflow creation, comprehensive regulation of the digital twin settings, and the cross-application of geographic information systems (GIS). It consists of various procedures for virtual environment creation, including analyzing available GIS data, installing the simulator, applying GIS-based real-world environment transformation tools, and collecting data using MAVs.

While the current literature offers specific digital twin platforms for MAVs, the works mainly focus on system verification or communication. As far as we know, not much development has concentrated on dynamic control and trajectory-following MAV inspection tasks. Hence, this paper presents a digital twin that integrates the dynamic behaviors of standard UAVs and implements an MAV swarm along with a cooperative path-planning algorithm. The co-simulation structure involves four main components: the cooperative path planning algorithm, the trajectory tracking controller, the vehicle dynamics simulator, and the environment simulator. The intelligent path planning algorithm determines the optimal flight paths for MAVs, while the trajectory tracking controller ensures that the MAV follows the desired flight path. The vehicle dynamics simulator models the physical properties and dynamics of the MAV, and the environment simulator models the external environment, including infrastructure, obstacles, and weather conditions.

The present study aims to address the development of intelligent drones and a digital twin platform for MAVs used in civil infrastructure inspection. First, it derives the dynamics of the MAV and introduces a hierarchical MAV control system to generate optimal paths and enable automatic flight. Second, it establishes a co-simulation framework to simulate and evaluate inspection mission scenarios, integrating these essential components. Finally, intelligent drones and digital twin technology are incorporated into a real-world scenario, including tests with both single and multiple drones, affirming their effectiveness in authentic applications. These contributions collectively demonstrate the effectiveness of the proposed digital twin platform in enabling a more efficient and reliable MAV control and trajectory-following system. Initial findings from this research were reported in symposium and conference papers (Nguyen et al. 2023, 2024a, b).

The remainder of this paper is structured as follows: Sect. 2 presents preliminaries of MAV modeling and control. Following this, Sect. 3 details intelligent path planning techniques for civil infrastructure inspection. Subsequently, Sect. 4 introduces tool development and visualization. Simulation results are discussed in Sect. 5, followed by experimental validation in Sect. 6. Finally, Sect. 7 provides the conclusion to the study.

2 Preliminaries

This section outlines MAV operation, modeling, and control algorithms. A thorough understanding of MAV dynamics is necessary to control the MAV effectively. Meanwhile, trajectory tracking control is essential in achieving stable flight and accurate trajectory following, enabling automatic flight during inspection tasks.

2.1 Description of MAV

MAVs are driven by a specific number of rotor pairs symmetrically arranged on opposite sides of the airframe, as illustrated in Fig. 1. Each rotor is positioned at the end of an arm with a defined length, and the angles between these arms are uniformly distributed. By controlling the speed of these rotors, MAVs can execute six degrees of freedom in their motion, including roll, pitch, yaw, and altitude adjustments.

To explain further, the pitch angle, which controls the forward and backward movement of the MAV, is achieved by manipulating the relative speeds of the front and rear rotors. This, in turn, generates the total front force, represented as $(F_1 + F_2 + F_8)$, and the total rear force, represented as $(F_4 + F_5 + F_6)$. Meanwhile, the roll angle governs the lateral displacement of the MAV. It is controlled by adjusting rotor velocities of the right and left sides, leading to the generation of total right forces, represented as $(F_2 + F_3 + F_4)$, and total left forces, represented as $(F_6 + F_7 + F_8)$. The yaw torque, which manages the rotation of the MAV, is achieved by altering the average speed of clockwise and anticlockwise rotation. It is worth noting that all rotors contribute to the overall thrust input. Generally, MAVs function similarly, with differing features, such

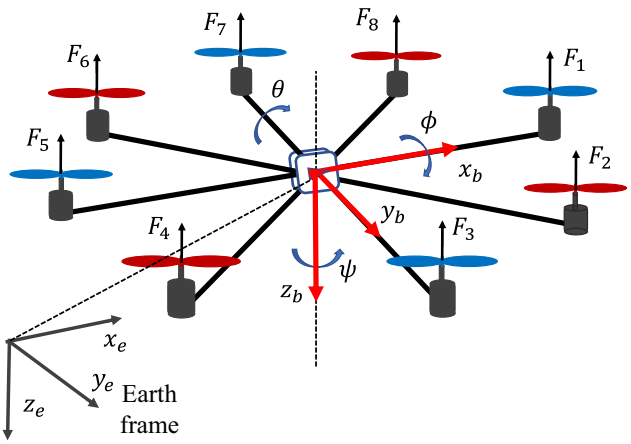


Fig. 1 Configuration of the MAV system

as rotor layout, size, and quantity. In practice, modifying the main configuration allows for creating many alternative variations of MAVs.

2.2 MAV model

Modeling expresses the physical principles governing a system's behavior or control system motion in mathematical terms. This enables numerical and graphical analysis of system performance through calculations and simulations. A parametric vehicle model is essential to allow proper control of MAVs. This paper considers a multi-rotor vehicle featuring $N_p \geq 2$ pairs of rotors, assuming rigidity and symmetry concerning the roll and pitch axes. Furthermore, we assume the center of gravity (CoG) of the drone is positioned at the origin of the body-fixed frame, with each propeller rotating in the opposite direction of its two neighboring ones.

As depicted in Fig. 1, an earth frame denoted as x_e, y_e, z_e is fixed to the ground, while a body frame denoted as x_b, y_b, z_b is affixed to the CoG of the vehicle. Both frames have their z axis oriented in a downward direction. A vector $X = (x, y, z)^T$ specifies the position of the CoG in the earth frame. Roll, pitch, and yaw motions are denoted by angles $(\phi, \theta, \psi)^T$. These angles are constrained for attitude control such that $\phi \in [-\pi/2, \pi/2]$, $\theta \in [-\pi/2, \pi/2]$, and $\psi \in [-\pi, \pi]$. The total thrust force F, expressed in the body coordinate system, is determined by

$$F = \sum_{i=1}^{2N_p} f_i = b \sum_{i=1}^{2N_p} \omega_i^2. \tag{1}$$

Here, $f_i = b\omega_i^2$ denotes the thrust force produced by the ith rotor, where ω_i stands for its rotational speed, and $b > 0$ represents the thrust coefficient. Furthermore, the torque vector components $\tau = [\tau_\phi \ \tau_\theta \ \tau_\psi]^T$, corresponding to rotational motion in the roll, pitch, and yaw directions, are derived as:

$$\tau_\phi = bl \sum_{i=1}^{2N_p} -\sin\left((i-\epsilon)\frac{\pi}{N_p}\right)\omega_i^2, \tag{2a}$$

$$\tau_\theta = bl \sum_{i=1}^{2N_p} \cos\left((i-\epsilon)\frac{\pi}{N_p}\right)\omega_i^2, \tag{2b}$$

$$\tau_\psi = \beta\left(\sum_{i=1}^{N_p} \omega_{2i}^2 - \sum_{i=1}^{N_p} \omega_{2i-1}^2\right) = \beta\left(\sum_{i=1}^{2N_p} (-1)^i \omega_i^2\right), \tag{2c}$$

where l denotes the length of the rotor arm, β represents the motional radius, and coefficient $\epsilon = 1$ for a "+" configuration MAV, or $\epsilon = 1/2$ for an "X" configuration MAV. Within the MAV dynamics, we account for external disturbances such as propeller gyroscopic torque, denoted as τ_p, and aerodynamic torques, denoted as τ_a, i.e., $d = \left[d_\phi \ d_\theta \ d_\psi\right]^T = \tau_p - \tau_a$, where d_ϕ, d_θ, and d_ψ are disturbance components.

It should be emphasized that while the thrust force F is the only control signal for the 3D translational dynamics, the active torques ultimately dictate the MAV's orientation. Henceforth, let the vector of virtual control inputs be denoted as $u = (u_1, u_2, u_3, u_4)^T$, where $u_1 = F$, $u_2 = \tau_\phi$, $u_3 = \tau_\theta$, and $u_4 = \tau_\psi$. Subsequently, a unified dynamical model for the MAV is provided below (Benzaid et al. 2016):

$$\ddot{x} = m^{-1}(C_\phi S_\theta C_\psi + S_\phi S_\psi)u_1, \tag{3a}$$

$$\ddot{y} = m^{-1}(C_\phi S_\theta S_\psi + S_\phi C_\psi)u_1, \tag{3b}$$

$$\ddot{z} = m^{-1} C_\phi C_\theta u_1 - g, \tag{3c}$$

$$\ddot{\phi} = I_x^{-1}\left[(I_y - I_z)\dot{\theta}\dot{\psi} + u_2 + d_\phi\right], \tag{3d}$$

$$\ddot{\theta} = I_y^{-1}\left[(I_z - I_x)\dot{\phi}\dot{\psi} + u_3 + d_\theta\right], \tag{3e}$$

$$\ddot{\psi} = I_z^{-1}\left[(I_x - I_y)\dot{\phi}\dot{\theta} + u_4 + d_\psi\right], \tag{3f}$$

where S_x and C_x respectively denote $\sin x$ and $\cos x$, m represents the total mass of the MAV, and $I = \mathrm{diag}(I_x, I_y, I_z)$ is the inertia matrix of the MAV.

2.3 MAV trajectory tracking control

The tracking controller ensures precision, safety, and reliability in autonomous MAV operations. It allows the MAV to navigate its environment effectively while performing tasks. A widely employed method for trajectory tracking control is the cascade control structure, depicted in Fig. 2. This structure breaks down the control system into loops, with MAV control organized hierarchically for different degrees of freedom (DOF). The inner loop focuses on attitude control to ensure stable flight and accurate trajectory tracking. In contrast, the outer loop oversees position control, crucial for maintaining the desired trajectory and target reaching without collisions. This separation into distinct loops within the hierarchical control scheme improves the efficiency of MAV movement control, enhances overall performance, and minimizes coupling between DOFs. The reduction in coupling leads to improved control stability and accuracy, underscoring the critical importance of the hierarchical control scheme for effective MAV control in real-world scenarios.

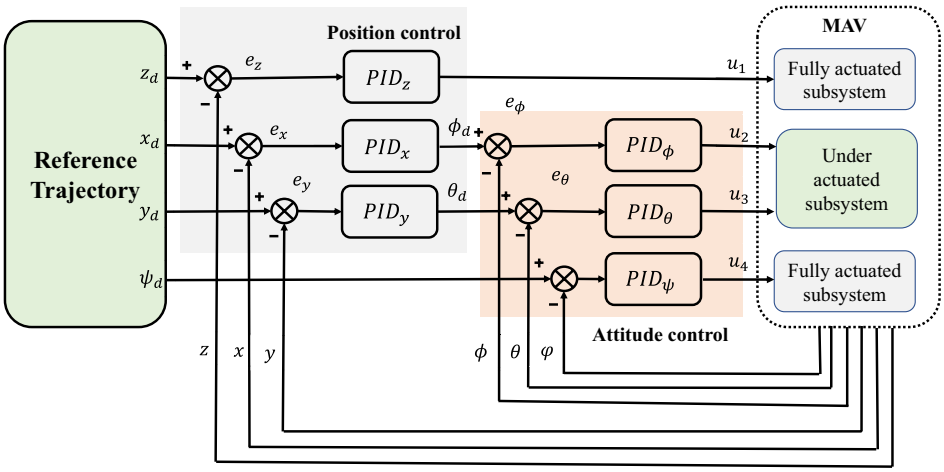


Fig. 2 MAV control architecture

In the control scheme, the fully actuated subsystem uses control signals u_1 and u_4 to control altitude and yaw motion, respectively. For the under-actuated subsystem, u_2 and u_3 control roll and pitch angles, respectively, which in turn control x and y positions. Remote ground operation or a trajectory planner typically provides the desired position for the MAV. A classical PID controller fine-tunes the MAV's output, which includes its position and orientation, adjusting the control signal based on the error between the desired and actual output. This PID controller comprises a proportional term, which modifies the control signal according to the current error, an integral term that adjusts the control signal based on the accumulation of past errors, and a derivative term that modifies the control signal based on the rate of change of the error. Known for its simple structure, straightforward implementation, commendable control performance, and robust design, the PID controller finds extensive industrial applications and particularly in flight control (Yue et al. 2022).

3 Intelligent path planning for civil infrastructure inspection with MAVs

In the previous section, trajectory tracking control during task execution was introduced to enable MAVs to effectively track a reference trajectory. However, addressing the practical necessity of determining optimal paths crucial for mission success requires introducing a higher path planning layer. This section develops intelligent path planning for civil infrastructure inspection with MAVs.

3.1 Path planning problem formulation

Consider an inspection task within a known environment, which includes obstacles like trees and light poles. Cylindrical models can represent these obstacles, as depicted in Fig. 3. The drone's position at node p is described by the vector $X(p) = (x(p), y(p), z(p))^T$, situated within an inertial frame xyz aligned with sea level, where the z axis points upwards. The challenge in MAV path planning for civil infrastructure inspection revolves around identifying the

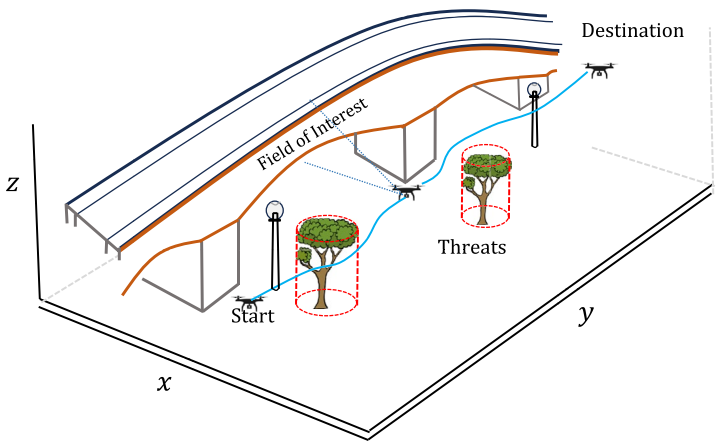


Fig. 3 Problem formulation

optimal path for the MAV, considering various constraints and prerequisites. To address this challenge, we formulate the path planning problem as an optimization process (Nguyen et al. 2022a):

$$X(0) \xrightarrow[s.t.\ J_s(\Pi)]{X(p)} X(end). \tag{4}$$

Here, $X(0)$ and $X(end)$ are the MAV's starting position and destination position, respectively. The path Π comprises a sequence of P waypoints represented as $X(p)$, where $p = 1, 2, \ldots, P$, which generally describes the flight trajectory of MAV.

In a drone-based inspection task, the objective is to determine the most efficient path for the MAV from an initial position to a designated target while satisfying all requirements dictated by constraints, such as safety, travel distance, smoothness, and coverage area. By integrating these constraints, we formulate the optimization problem for path planning, with the cost function defined as:

$$J_s(\Pi) = \omega_1 J_{\text{co}} + \omega_2 J_{\text{sa}} + \omega_3 J_{\text{tr}} + \omega_4 J_{\text{sm}}, \tag{5}$$

where J_{co} signifies the coverage-related cost, J_{sa} represents the safety cost associated with T threats, J_{tr} denotes the traveling distance cost, and J_{sm} corresponds to the smoothness cost. The weight coefficients ω_i for $i = 1, 2, \ldots, 4$ serve as design parameters. Further details regarding the individual cost functions are outlined below.

3.1.1 Coverage cost

In civil infrastructure inspections, there is a growing inclination toward leveraging MAVs as mobile sensors, offering distinct advantages over traditional static monitoring techniques. These MAVs are essential in encompassing the area of interest and gathering helpful information, predominantly through visual data capture. With an identical horizontal-facing camera configuration, MAVs can cover a square field of view (FOV), forming a pyramid with a half angle θ. There are two exceptional cases for angles between the vertical medians of the faces and the height, represented as θ_1 and θ_2, as depicted in Fig. 4. To ensure thorough coverage, a point $\hat{p}$ falls within the MAV's FOV if it satisfies the following equations:

$$\|\hat{p} - c_i\| \leq d \tan\theta. \tag{6}$$

Here, c_i signifies the projected position of the MAV on the infrastructure surface, while d represents the distance from the MAV to that point. The main goals of the MAV involve maximizing coverage across the field of interest F within its field of view (FOV) and achieving a desired level of overlap to improve overall coverage efficiency.

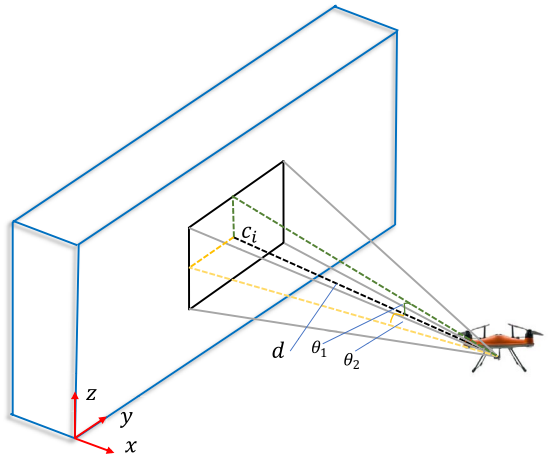


Fig. 4 MAV field of view

For the sake of convenience, we assume that the coordinate system xyz is fixed to one corner of the infrastructure, with the z axis pointing up, as shown in Fig. 4. During civil infrastructure testing, we plan MAV flights along the y-axis and maintain a constant speed for the MAV in the y-axis, yielding $y(k)$. Therefore, the vertical FOV remains the sole consideration in the path planning problem. Employing the camera model delineated in Fig. 4, we ascertain the maximum distance $d_{\max}$ of a captured surface that still conforms to the minimum target pixel resolution δ, as follows:

$$d_{\max} = \frac{r}{2\delta \tan\rho}, \tag{7}$$

where ρ represents the camera's aperture angle, and r denotes the camera resolution.

The coverage height H_c when the MAV is located at (x, y, z) is computed as:

$$H_c = \min(h^u, H_b^u) - \max(h^l, H_b^u), \tag{8}$$

where H_b^u and H_b^l denote the upper and lower bounds of the infrastructure, respectively. The upper and lower vertical coverage bounds of the MAV are calculated as

$$h^u = \begin{cases} z + x\tan\rho, & \text{if } x \leq d_{\max} \\ 0, & \text{if } x > d_{\max}, \end{cases} \tag{9a}$$

$$h^l = \begin{cases} z - x\tan\rho, & \text{if } x \leq d_{\max} \\ 0, & \text{if } x > d_{\max}. \end{cases} \tag{9b}$$

The overlapping height H_o is computed as:

$$H_o = H_o^u + H_o^l, \tag{10}$$

in which

$$H_o^u = \left|\mu(h^u - h^l) - \max(0, h^u - H_b^u)\right|, \tag{11a}$$

$$H_o^l = \left|\mu(h^u - h^l) - \max(0, H_b^l - h^l)\right|, \tag{11b}$$

where μ denotes the desired overlapping percentage.

Using (8) and (10), we can formulate the coverage cost as follows:

$$J_{co} = \beta_1 \sum_{p=1}^{P} [H_b(p) - H_c(p)] + \beta_2 \sum_{p=1}^{P} H_o(p), \tag{12}$$

where $H_b(p) = H_b^u(p) - H_b^l(p)$ stands for the height of the bridge at node p, and β_1 and β_2 are weighting factors adjusting the impact of the total coverage and overlapping heights, respectively.

3.1.2 Safety cost

During operations, the MAV must steer clear of potential collisions with obstacles in its operational environment, such as trees or utility poles. In this work, each obstacle is considered a threat and modeled as a cylinder with a radius of r_τ and a height of z_τ. Figure 5 depicts the obstacle representation and the safe distance for a flight segment $\overline{X(p)X(p+1)}$ relative to the obstacle (C_τ, r_τ, z_τ), where C_τ represents the center of the obstacle. Let $\overline{A(p)B(p)}$ signify the segment portion located at or below the obstacle's height z_τ, and let $d_s(p)$ denote the distance between C_τ and $\overline{A(p)B(p)}$. In Case 1, if both waypoints $X(p)$ and $X(p+1)$ are above the obstacle, $\overline{A(p)B(p)}$ does not exist, resulting in $d_s(p) = \infty$. In Case 2, if both waypoints $X(p)$ and $X(p+1)$ are below or at the obstacle's height, $\overline{A(p)B(p)} = \overline{X(p)X(p+1)}$. In Case 3, when one waypoint is positioned above the obstacle while the other is below or at the obstacle's height, the values of $A(p)$ and $B(p)$ are established as

$$\begin{cases} x_{A_p} = x_p + h_\tau \dfrac{x_{p+1} - x_p}{z_{p+1} - z_p}, \\ y_{A_p} = y_p + h_\tau \dfrac{y_{p+1} - y_p}{z_{p+1} - z_p}, \\ z_{A_p} = h_\tau, \end{cases} \tag{13}$$

$$\begin{cases} c_p = (z_p < z_{p+1}), \\ B(p) = c_p X(p) + [1 - c_p] X(p+1), \end{cases} \tag{14}$$

where, in this case, $A(p) = (x_{A_p}, y_{A_p}, z_{A_p})$ represents the intersection of the segment $\overline{X(p)X(p+1)}$ with the plane $z = h_\tau$, and $B(p)$ refers to the waypoint that has an altitude lower than or equal to the obstacle's height.

Let $\hat{A}(p) = (x_{A_p}, y_{A_p}, 0)$ and $\hat{B}(p) = (x_{B_p}, y_{B_p}, 0)$ denote the projections of $A(p)$ and $B(p)$ onto the plane Oxy, respectively. To determine the distance $d_s(p)$ from the center of the obstacle $C_\tau(x_c, y_c)$ to the path segment at node p, we calculate $|C - \hat{A}(p)|$ when $|\hat{A}(p) - \hat{B}(p)| = 0$. Alternatively, when $|\hat{A}(p) - \hat{B}(p)| \neq 0$ and $\mathcal{A} < 0$, $d_s(p)$ is computed as $\min(|C - \hat{A}(p)|, |C - \hat{B}(p)|)$. Here,

$$\mathcal{A} = [(x_{A_p} - x_{B_p})(x_c - x_{B_p}) + (y_{A_p} - y_{B_p})(y_c - y_{B_p})] \times [(x_{B_p} - x_{A_p})(x_c - x_{A_p}) + (y_{B_p} - y_{A_p})(y_c - y_{A_p})].$$

If $|\hat{A}(p) - \hat{B}(p)| \neq 0$ and $\mathcal{A} \geq 0$, the computation for $d_s(p)$ is as follows (Nguyen et al. 2022b):

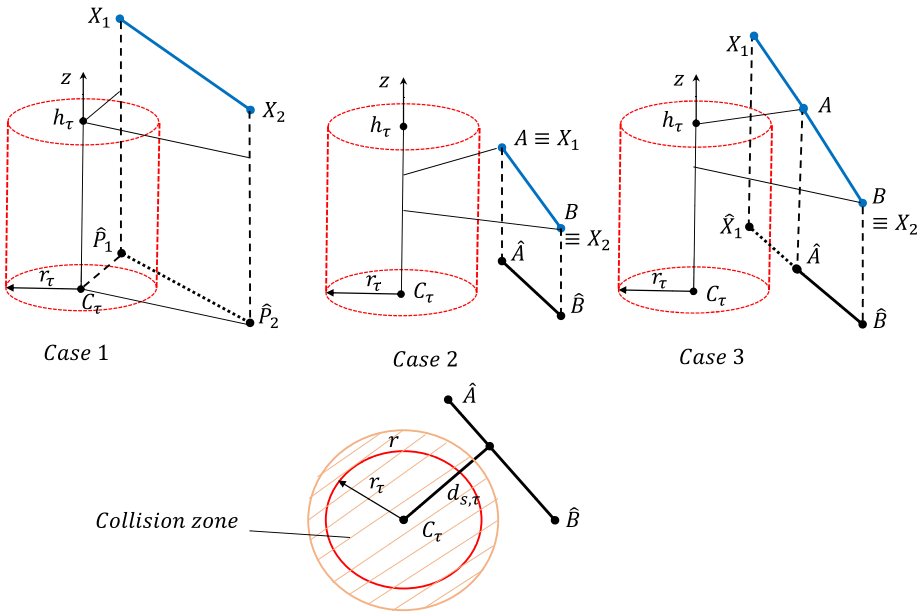


Fig. 5 Obstacle representation and safe distance

$$d_s(p)=\frac{\left|(x_{A_p}-x_{B_p})(y_{B_p}-y_c)-(x_{B_p}-x_c)(y_{A_p}-y_{B_p})\right|}{\sqrt{(x_{A_p}-x_{B_p})^2+(y_{A_p}-y_{B_p})^2}}. \quad (15)$$

The safety cost is then determined as

$$J_{sa}=\sum_{k=1}^{K-1}\sum_{\tau=1}^{\mathcal{T}}S_\tau(p), \quad (16)$$

where

$$S_\tau(p)=\begin{cases}0, & \text{if } d_s(p)>r_\tau+r_v\\ \infty, & \text{if } d_s(p)\le r_\tau+r_v.\end{cases} \quad (17)$$

Here, $S_\tau(p)$ represents the collision cost assigned to waypoint $X(p)$ concerning obstacle (C_τ, r_τ, z_τ), and r_v denotes the radius of the MAV.

3.1.3 Traveling cost

When planning a path, minimizing its length is essential to conserve time and energy, particularly for low-cost UAVs. In autonomous operations, a path typically comprises a series of waypoints uploaded to the MAV as references for the flight controller to adhere to. With P waypoints, the path can be delineated as a collection of $P-1$ line segments connecting these waypoints. The total path length is then the summation of these segments. Representing $X(p)$ as waypoint p along the path, the length pertaining to segment $\overline{X(p)X(p+1)}$ is computed as:

$$L(p)=||X(p+1)-X(p)||. \quad (18)$$

Consequently, the traveling cost is determined as follows:

$$J_{tr}=\sum_{p=1}^{P-1}L(p). \quad (19)$$

3.1.4 Smoothness cost

The constraints imposed by the MAV dynamics on motion dictate limitations in making sharp turns, requiring the algorithm to generate a smooth path. This objective is accomplished by limiting changes in turning and climbing angles.

The turning angle between two consecutive segments, denoted as $\phi(p)$, is calculated using the formula:

$$\phi(p)=\cos^{-1}\left(\frac{\overrightarrow{P(p)}\cdot\overrightarrow{P(p+1)}}{|\overrightarrow{P(p)}|\cdot|\overrightarrow{P(p+1)}|}\right), \quad (20)$$

where $\overrightarrow{P(p)}=\mathrm{Proj}_{Oxy}(X(p+1)-X(p))$ represents the projection of the segment $(X(p+1)-X(p))$ onto the plane Oxy.

The climbing angle, denoted as $\varphi(k)$, between the path segment $(X(p+1)-X(p))$ and $\overrightarrow{P(p)}$ is computed as:

$$\varphi(p)=tan^{-1}\left(\frac{z(p+1)-z(p)}{\|\overrightarrow{P(p)}\|}\right). \quad (21)$$

Accordingly, the smoothness cost is formulated as:

$$J_{sm}=\beta_3\sum_{p=1}^{P-2}\phi(p)+\beta_4\sum_{p=1}^{P-1}|\varphi(p)-\varphi(p+1)|, \quad (22)$$

where β_3 and β_4 denote weighting factors adjusting the impact of the turning angle constraint and climbing angle, respectively. These factors collectively contribute to ensuring the smoothness of the resultant MAV path.

3.2 GWO for MAV path planning

The function $J_s(\Pi)$ outlined in (5) defines the cost function for the path planning task. This formulation converts path planning into an optimization challenge to minimize $J_s(\Pi)$. However, the complexity and multimodal characteristics inherent in $J_s(\Pi)$ render traditional methods such as hill climbing impractical, as they often get trapped in local maxima. Consequently, heuristic and metaheuristic techniques have gained traction for efficiently attaining high-quality solutions. Notably, the GWO algorithm has emerged as a potent population-based metaheuristic, drawing inspiration from the hunting behaviors observed in wolf packs (Mirjalili et al. 2014).

Within the GWO framework, a population of N individuals tackles an optimization problem of M dimensions. In the context of optimizing MAV path planning, each individual's flight trajectory is represented as a vector within the inertial frame (xyz), encoding the movement between waypoints. Specifically, for a MAV path labeled as i, composed of P nodes, the vector representation extends across $M=3P$ dimensions, formulated as follows:

$$\Pi_i=(x_{i1},y_{i1},z_{i1},x_{i2},y_{i2},z_{i2},\ldots,x_{iP},y_{iP},z_{iP}), \quad (23)$$

where $1\le i\le N$ denotes each individual within the population.

During the predation process, grey wolves encircle their prey, as depicted by the following vector of absolute value elements:

$$D_i=\left|\Pi_i(t)-C\circ\Pi_p(t)\right|, \quad (24)$$

where $\circ$ symbolizes the element-wise Hadamard product, t denotes the current iteration, and Π_p represents the position of the prey, corresponding to the optimal solution. The oscillation factor C is determined as

$$C = 2r_1, \tag{25}$$

where r_1 is a random vector with elements distributed uniformly in the range [0, 1].

The grey wolf's location is updated as

$$\Pi_i(t+1) = \Pi_i(t) - A \circ D. \tag{26}$$

Here, the convergence factor A is calculated as

$$A = 2ar_2 - a, \tag{27}$$

where r_2 is a random vector with elements uniformly distributed in the interval [0, 1], coefficient a decreases from 2 to 0 over iterations as per (Mirjalili et al. 2014):

$$a = 2 - \frac{2t}{T}, \tag{28}$$

and T represents the maximum number of iterations.

In practical scenarios where the precise location of the prey, Π_p, within the search space, remains unknown, the top three best solutions are denoted by Alpha, Beta, and Delta. Equation (24) is adapted as

$$\begin{cases} D_{\alpha_i} = \left|\Pi_i(t) - C_1 \circ \Pi_\alpha(t)\right|, \\ D_{\beta_i} = \left|\Pi_i(t) - C_2 \circ \Pi_\beta(t)\right|, \\ D_{\delta_i} = \left|\Pi_i(t) - C_3 \circ \Pi_\delta(t)\right|, \end{cases} \tag{29}$$

where C_j, for $j = 1, 2, 3$, is computed as described in (25), and Π_α, Π_β, and Π_δ represent the positions of Alpha, Beta, and Delta, respectively. Subsequently, the other grey wolves update their positions as follows:

$$\begin{cases} \Pi_{1_i} = \Pi_\alpha - A_1 \circ D_\alpha, \\ \Pi_{2_i} = \Pi_\beta - A_2 \circ D_\beta, \\ \Pi_{3_i} = \Pi_\delta - A_3 \circ D_\delta, \end{cases} \tag{30}$$

$$\Pi_i(t+1) = \frac{\Pi_{1_i} + \Pi_{2_i} + \Pi_{3_i}}{3}. \tag{31}$$

Here, A_j, for $j = 1, 2, 3$, is computed as given in (27).

Algorithm 1 outlines the pseudocode for implementing GWO for MAV path planning. It begins by encoding MAV flight paths, initializing positions randomly, and evaluating fitness to identify Alpha, Beta, and Delta. The main loop iteratively updates particle dynamics based on Alpha, Beta, and Delta positions. The algorithm continues until either fitness stabilizes or a preset iteration threshold is reached, balancing solution quality with computation time.

Algorithm 1 GWO for MAV path planning

Require: Search map and initial path planning information;
Initialize GWO parameters: N, T;
Set $t = 0$;
for $i = 1 : N$ do
  Generate random Π_i;
  Calculate cost $J(\Pi_i)$;
end for
Sort $J(\Pi)$ to determine $\Pi_\alpha, \Pi_\beta, \Pi_\delta$;
for $t = 1 : T$ do
  Update a;
  for $i = 1$ to N do
    Update A_j, C_j for $j = \{1, 2, 3\}$;
    Compute $D_{\alpha_i}, D_{\beta_i}, D_{\delta_i}$;
    Compute $X_{1_i}, X_{2_i}, X_{3_i}$;
    Update $\Pi_i(t+1)$;
    Calculate $J(\Pi_i(t+1))$;
    if $(J(\Pi_i(t+1)) \leq J(\Pi_\alpha))$ then
      Update Delta: $\Pi_\delta = \Pi_\beta$;
      Update Beta: $\Pi_\beta = \Pi_\alpha$;
      Update Alpha: $\Pi_\alpha = \Pi_i(t+1)$;
    end if
    if $(J(\Pi_\alpha) < J(\Pi_i(t+1)) \leq J(\Pi_\beta))$ then
      Update Delta: $\Pi_\delta = \Pi_\beta$;
      Update Beta: $\Pi_\beta = \Pi_i(t+1)$;
    end if
    if $(J(\Pi_\beta) < J(\Pi_i(t+1)) \leq J(\Pi_\delta))$ then Update Delta: $\Pi_\delta = \Pi_i(t+1)$;
    end if
  end for
end for
return Π_α.

3.3 Cooperative path planning

In practice, executing complex tasks often requires the collaboration of multiple MAVs working as a team to meet various requirements, such as large-scale 3D or façade-long infrastructure inspections. Cooperative control of a group of MAVs engaged in a robotic task can offer significant efficiency, reliability, and flexibility advantages compared to executing the task with single MAVs. Therefore, the path planning problem for multiple MAVs has garnered significant interest. The cost function of MAV_m in a cooperative task, $J(\Pi_m, \Pi_m^-)$, where Π_m^- corresponds to a set of paths taken by neighboring MAVs to MAV_m, consists of a single cost and a cooperative cost, determined by

$$J(\Pi_m, \Pi_m^-) = J_s(\Pi_m) + \beta J_c(\Pi_m, \Pi_m^-). \tag{32}$$

Here, β represents a weighting factor, and $J_s(\Pi_m)$ denotes the single MAV cost as defined in (5). Additionally, $J_c(\Pi_m, \Pi_m^-)$ stands for the cooperative-MAV cost, which is determined based on the specific cooperative task. This study focuses on large-scale 3D inspection using multiple MAVs, where maintaining MAV formation is important throughout the mission. The cooperative-MAV cost $J_c(\Pi_m, \Pi_m^-)$ in Eq. (32) is derived from MAV cooperation and formation requirements, incorporating formation cost and the avoidance of inter-vehicle collisions. It is defined as follows (Nguyen et al. 2022b):

$$J_c(\Pi_m, \Pi_m^-) = \sum_{p=1}^{P} E_m(p), \tag{33}$$

where $E_m(p)$ represents the formation error integrated with collision-free constraints.

With the cost function $J_m(\Pi_m, \Pi_m^-)$ established for each MAV, the cooperative path planning problem transforms into the task of simultaneously minimizing $J_m(\Pi_m, \Pi_m^-)$ for

Fig. 6 Digital twin model of the 3DR Solo Drone

paths $\Pi_m, m = 1, 2, \ldots, M$. This challenge arises because the cost depends not only on the path Π_m designed for MAV_m but also on the paths of its competitors Π_m^-. Meanwhile, recognizing game theory's effectiveness in conflict resolution and managing interactions, this study adopts a recent game theory-based particle swarm optimization (GPSO) approach (Nguyen et al. 2022b) to generate MAV paths efficiently. By employing GPSO, the cooperative path planning problem is addressed by identifying the equilibrium of a Stackelberg-Nash game within a PSO hierarchical optimization framework. The Pseudocode is given in Algorithm 2. Here, Π_l and Π_f correspond to the strategies of the leader and followers, respectively. $\overset{*}{\Pi}_l$ and $\overset{*}{\Pi}_f$ stand for their corresponding optimal solutions, and $n = 1, \ldots, N$ denotes the nth follower.

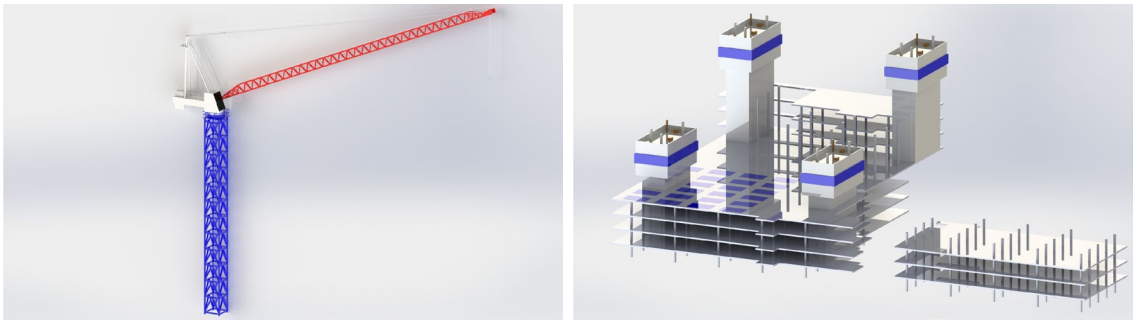
Algorithm 2 Leader-follower optimization hierarchy (Nguyen et al. 2022b)

Fix Π_l;
for $n = 1 : N$ **do**
  for $n = 1 : N$ **do**
    Fix $\Pi_{f_n}^-$;
    $\overset{*}{\Pi}_{f_n}(\Pi_l, \Pi_{f_n}^-) = \arg\min_{\Pi_{f_n}} J_{fn}(\Pi_l, \Pi_{fn}, \Pi_{f_n}^-)$;
    Obtain $\overset{*}{\Pi}_{f_n}(\Pi_l, \Pi_{f_n}^-)$ and $\overset{*}{\Pi}_{f_n}^-(\Pi_l, \Pi_{f_n})$;
  end for
  Substitute $\Pi_{f_n}^- = \overset{*}{\Pi}_{f_n}^-(\Pi_l, \Pi_{f_n})$ into $\overset{*}{\Pi}_{f_n}(\Pi_l, \Pi_{f_n}^-)$;
  Obtain $\overset{*}{\Pi}_{f_n}(X_l, \overset{*}{\Pi}_{f_n}^-)$;
end for
Obtain $\overset{*}{\Pi}_f(\Pi_l)$;
$\overset{*}{\Pi}_l = \arg\min_{\Pi_l} J_l(\Pi_l, \overset{*}{\Pi}_f(\Pi_l))$;
Obtain $\overset{*}{\Pi}_l$;
Substituting $\Pi_l = \overset{*}{\Pi}_l$ into $\overset{*}{\Pi}_f(\Pi_l)$;
Obtain $\overset{*}{\Pi}_f(\overset{*}{\Pi}_l)$;

4 Tool development and visualization

This section develops a prototypical digital twin package explicitly designed for MAV systems, encompassing flying environments, vehicle dynamics, control algorithms, and cooperative path planning. The study focuses on creating a virtual representation of real-world systems, updated from real-time data, to aid in control, conditional monitoring, and decision-making through a learning-based simulation platform. To achieve this, software tools are developed to describe system dynamics, operational environments, and other control and tracking requirements across various MAV mission circumstances. The proposed digital twin package offers several advantages over traditional physical testing methods. Notably, it enables thorough testing, validation, and verification of MAV control and monitoring in abnormal conditions, all without the need for physical implementation or field experiments for the entire system. This significantly mitigates the need for extensive testing efforts throughout the development cycle, ensuring optimal cooperation, safety, smoothness, fault tolerance, and energy efficiency.

(a) Crawler excavators [34] (b) Cement mixer truck [35]



(c) Tower crane [36] (d) Uncompleted building

Fig. 7 CAD models of construction objects

The process begins with creating a 3D CAD model of the MAV in Solidworks, incorporating dynamic parameters. Then, the drone model is transferred to Matlab/Simulink, integrated with a virtual scenario. Here, this is a construction site developed in Simulink Simscape Multibody, with imported CAD files for various obstacles.

4.1 Digital twin of the MAV model

The initial step involves constructing a digital model of a real-world MAV, such as the 3DR Solo utilized in this study, through component-based and parametric modeling techniques to visualize the model. This facilitates modular design while considering compatibility with the digital twin platform. The quadcopter's structural frame incorporates an attached battery, replicated legs forming the landing gear, and a propulsion system comprising two pairs of contra-rotating brushless DC motors and corresponding propellers. Furthermore, the materials and surface textures of each component are specified. This process ensures that the modeling software accurately computes essential parameters, such as mass, moment of inertia, and other intrinsic values, while visually representing its physical twin, as depicted in Fig. 6.

The software facilitates the extraction of dynamic parameters for the quadcopter from manufacturer specifications and measurements. These numerical values can be stored in a compatible file format for future utilization. Obtaining precise inertial figures for a drone type, whether through physical measurements or analytical calculations, requires instrumentation or validation work. Here, by utilizing CAD modeling, the developed digital twin can support various drones on this highly realistic platform.

4.2 Virtual flight environment

Creating a digital twin for MAV systems involves establishing a virtual flight environment that faithfully replicates the motion of aerial vehicles. Unlike existing simulators like AirSim, Gazebo, jMAVSim, and Flight-gear for visualized representations of MAV dynamics, often at a considerable computational expense, our platform is compatible with simulation tools, able to facilitate the crafting 3D models of complex objects, hence reducing the need for additional software.

To illustrate the capabilities of the digital twin platform, simplified versions of construction objects were developed to simulate a hypothetical construction site. These objects, including crawler excavators, cement mixer trucks, jib tower cranes and unfinished buildings, recreate real-life

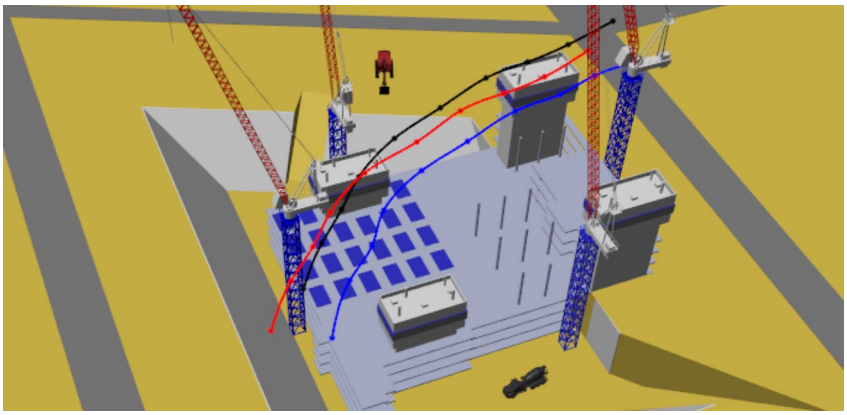


Fig. 8 Virtual flight environment

construction activities in virtual scenarios. For instance, excavation, concrete pouring operations, and heavy lifting and material moving, require excavators, mixer trucks, and tower cranes. These objects are selected from open source designs (Mohamed https://grabcad.com/library/volvo-excavator-4; Leonardo https://grabcad.com/library/truck-cement-mixer-1; Tech https://grabcad.com/library/tower-crane-15), as shown respectively in Figure 7a–c. Additionally, the construction site includes unfinished buildings with exposed concrete floors and columns. A general floor plan is component-based and created for the upper levels, with supportive pillars arranged accordingly, as shown in Fig. 7d. Furthermore, the tower blocks of the building complex are considered as obstacles for aerial missions.

By integrating the above elements into the simulation environment, the digital twin platform provides a more comprehensive and realistic representation of the construction site. This allows users to test and tweak different build scenarios before deploying in the real world. Additionally, the desired waypoints obtained from the path planning algorithm are depicted as spherical markers, and the interpolated polynomial trajectory is shown to analyze the response of the MAV. A virtual environment is then established, as depicted in Fig. 8.

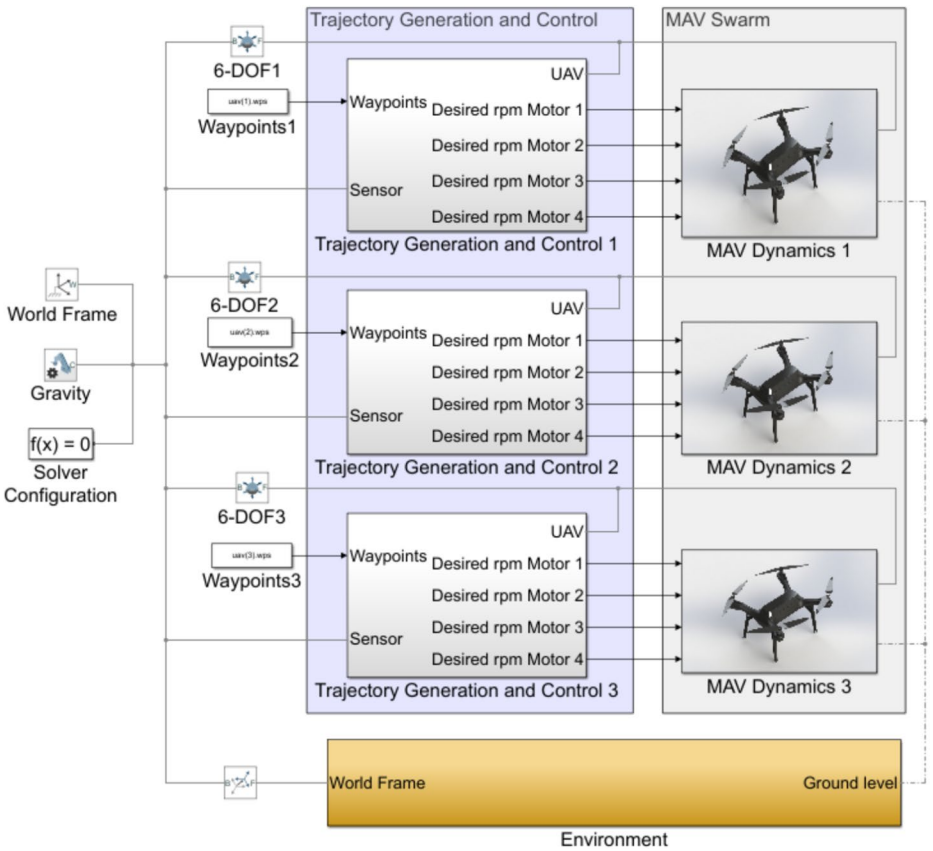


Fig. 9 Visualization model

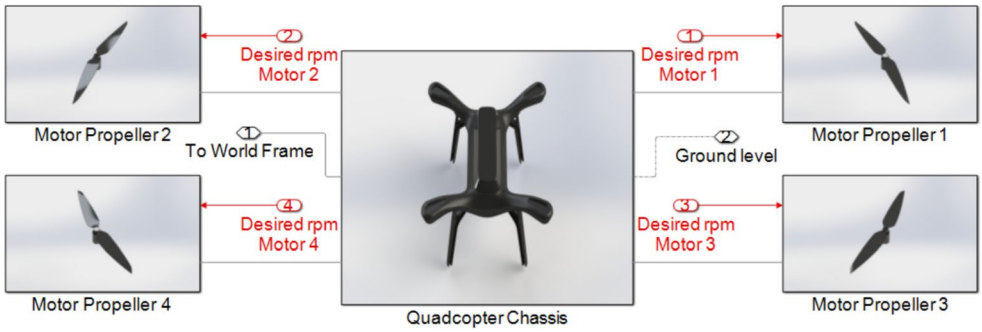


Fig. 10 MAV subsystem

4.3 Visualization model

To validate the efficacy of the proposed approach, we implement the path planning and trajectory control algorithms using Matlab/Simulink in conjunction with the MAV dynamic model. Following this, the virtual aerial vehicle is subjected to thorough evaluation tests to assess the performance of the trajectory generation, each control subsystem, alongside the MAV considering environmental factors. These components are illustrated in Figure 9. Further details regarding each subsystem are provided in the following.

Figure 10 illustrates the subsystem of MAV dynamics. This subsystem includes the structural frame and propulsion system of the quadcopter at its highest level, as detailed in Sect. 4.1. Here, the Simscape Multibody Link plug-in is used to export the digital twin of the drone, making use of the data transfer advantage of our visualization platform. Consequently, an extensible markup language (.xml) file is generated, outlining the intrinsic geometric relationships between each part within a CAD assembly model. Upon loading this file, a Simscape Multibody model of the multi-rotor is automatically generated. Minor adjustments are made to ensure alignment between the CAD model, created with standard practices in 3D modeling software, and the simulation requirements of the digital twin platform. The arrangement of the propulsion system is designed to be graphically intuitive for user-friendly interaction. The stationary components of the drone, such as the four legs and battery, are grouped within the "Quadcopter Chassis" block. This process may differ for various multi-rotor vehicles and is contingent upon the desired level of detail in the simulations.

Figure 11 illustrates the environment subsystem, comprising blocks for visualizing the terrain and surroundings of the construction site, as well as the trajectories of the MAV swarm. In the Simscape Multibody simulation, the world origin is connected to every block visually represented in the animation, with the "World Frame" serving as the input to all internal blocks. The topography of the construction site is obtained from the TouchTerrain web application (Hasiuk et al. 2017) in standard triangulation language (.stl) file format and incorporated into the digital twin platform through a script that reads and converts the .stl file into a compatible data type for the "Rigid Terrain Grid Surface" block. Additionally, the "Construction Site" block directly integrates stationary CAD models of background objects into the relevant test cases. Meanwhile, the "Trajectory" block facilitates the placement of markers for waypoints and visualization of planned paths. Utilizing these blocks enables the creation of environments for various MAV missions within the Mechanics Explorer.

Figure 12a illustrates the outer layer of the "Trajectory Generation and Control" block for a single MAV. In this diagram, the "Waypoint Allocation" block receives the waypoints the cooperative path planning algorithm generates. The block's logic sequentially assigns a new destination to the vehicle once it reaches an acceptable range to a waypoint. Additionally, the block endeavors to hover the vehicle at the final waypoint. The positional references are then transmitted to the controllers. The position control initially receives the desired longitudinal and latitudinal

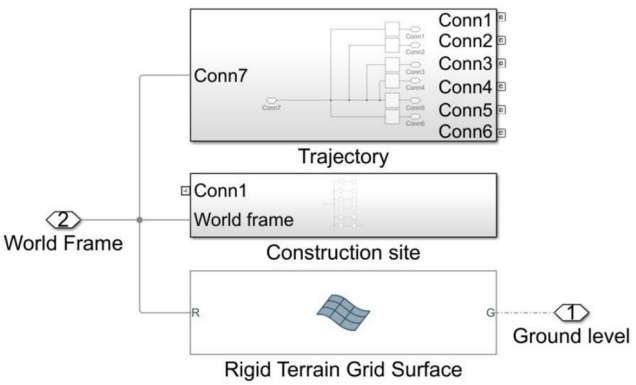
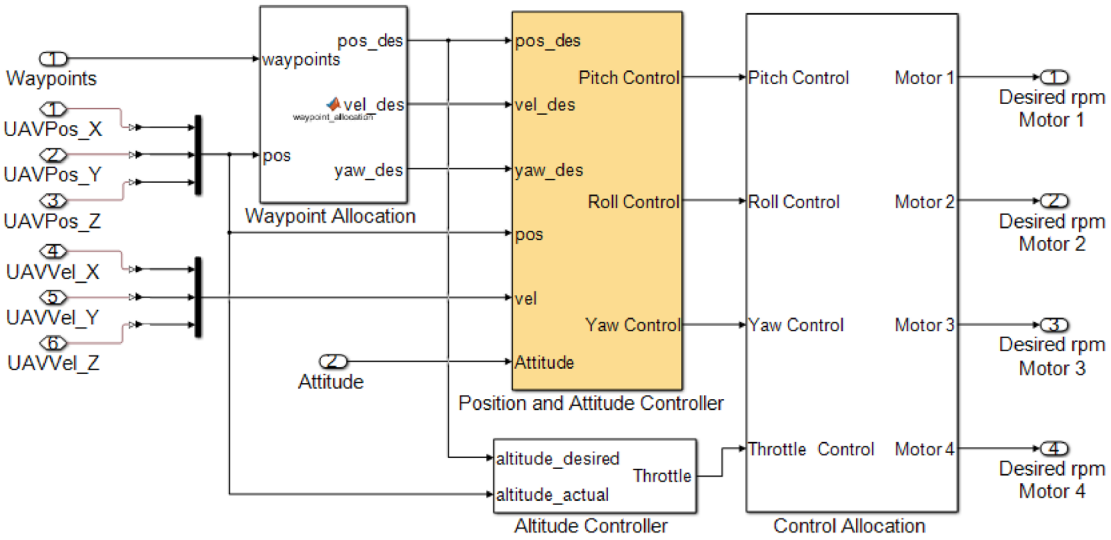
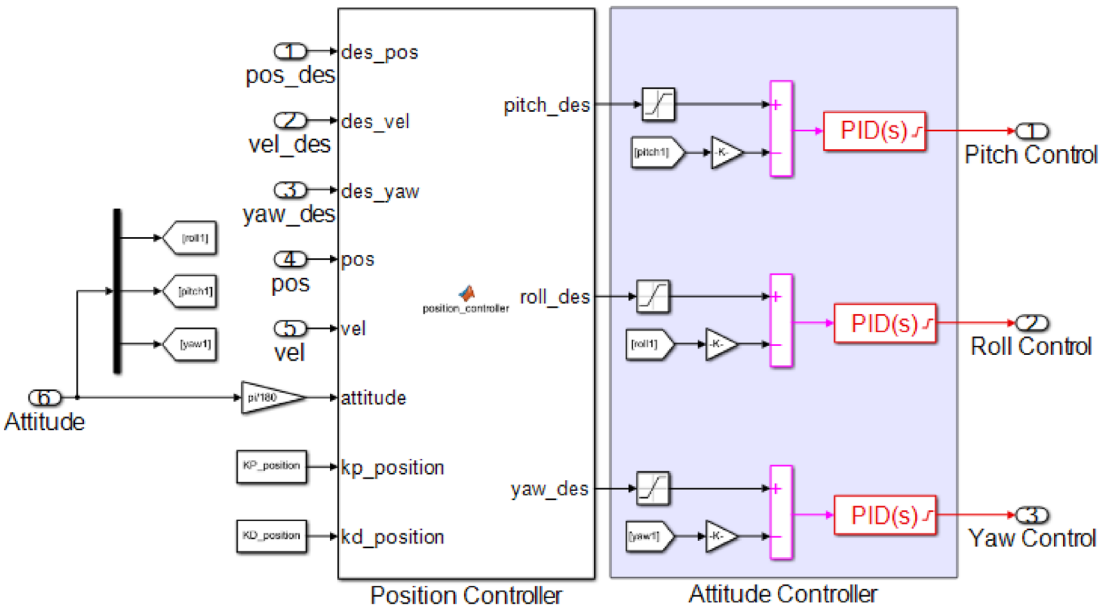


Fig. 11 Environment subsystem



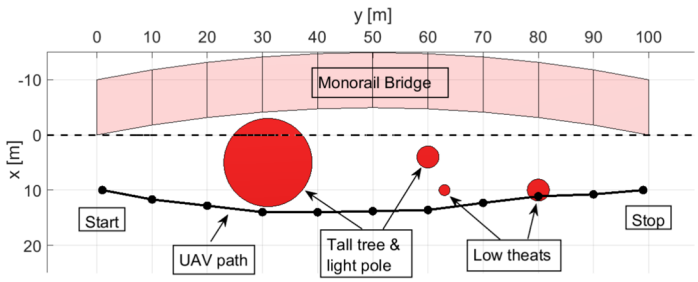
(a) Outer layer



(b) Position and attitude controller block

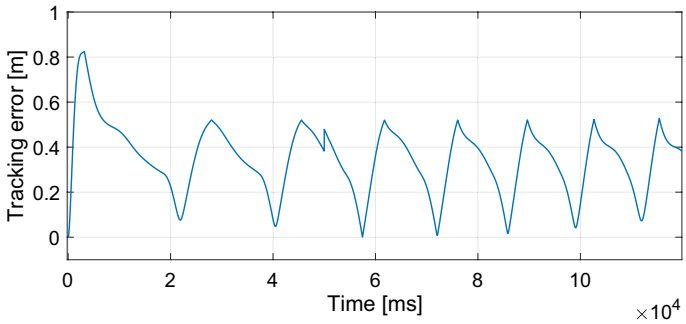
Fig. 12 Trajectory generation and control block

Fig. 13 Generated path



(a) Virtual MAV during the experiment

(b) Virtual bridge image captured from MAV



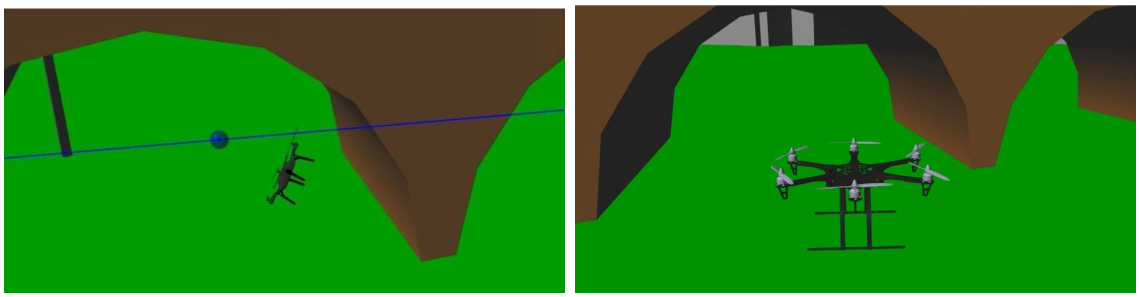
(c) Tracking error

Fig. 14 Simulation result

values in the "Position and Attitude Controller" block, elaborated in Fig. 12b. This block executes three main steps. Firstly, it applies corrections to position errors using proportional-derivative (PD) gains. Next, it converts from x and y positions to the required orientation, represented as roll-pitch-yaw angles. Finally, the attitude proportional-integral-derivative (PID) controller aims to adjust the current rotational values accordingly. Concurrently, the desired altitude is provided to the "Altitude Controller" at the outer layer for appropriate adjustments to the resulting thrust. The outputs of the attitude and altitude controllers are converted into more intuitive motor revolutions for subsequent operations conducted in the "Control Allocation" block. This block determines the required RPM for each specific motor, which is then utilized by the "MAV Dynamics" subsystem.

5 Simulation results

In this section, we delve into simulation case studies focusing on monorail bridge inspection and construction site monitoring, showcasing the application of intelligent drone systems and digital twins in modern civil infrastructure management.



(a) Quadcopter

(b) Hexacopter

Fig. 15 MAV in a faulty operation

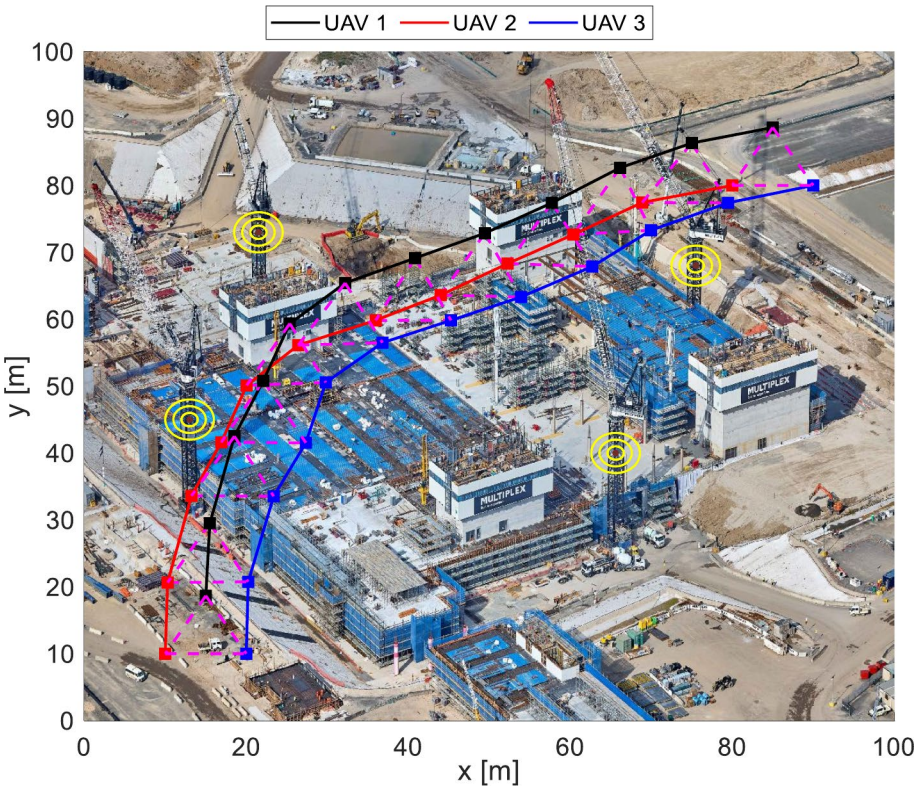


Fig. 16 Generated cooperative path (Nguyen et al. 2023)

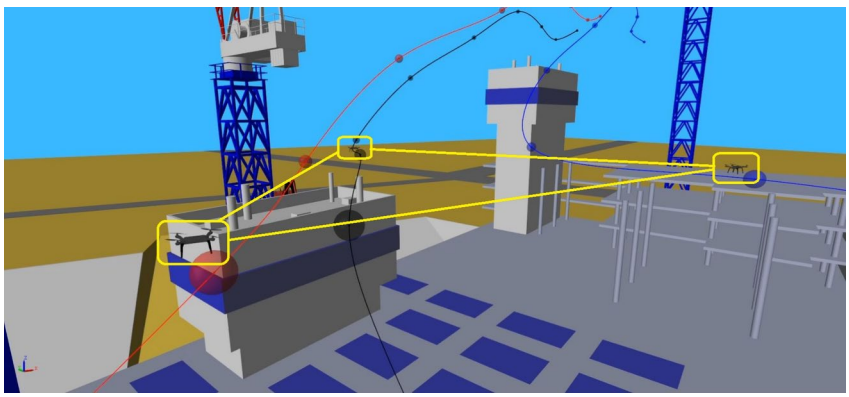


Fig. 17 Visualization results

5.1 Case study 1: Monorail bridge inspection

To illustrate the effectiveness of our proposed method, we carried out a virtual inspection of a monorail system situated at Wentworth Park, NSW, Australia, with coordinates −33.875266° S, 151.192769° E. Based on satellite-captured images (Google Maps https://www.google.com/maps/@-33.875266,151.192769,145m/data=!3m1!1e3?entry=ttu), we modeled the viaduct supporting the entire railway system and strategically placed obstacles, such as trees and light poles. We aimed to deploy an MAV with designated take-off and landing locations to perform an inspection task. For a single MAV, we relaxed the cooperative constraint. Employing the WGO algorithm, we generated optimized, collision-free routes while taking into account environmental obstacles, as illustrated in Fig. 13. We implemented red cylinders to denote threat zones around obstacles to ensure safe MAV maneuvering. These cylinders represented collision-free areas, ensuring safe navigation in real-world scenarios where obstacles are present. Furthermore, we employed a cascade trajectory tracking controller to ensure precise adherence to the MAV's planned routes. Continuously monitoring the MAV's position and velocity, this controller compared them to the planned trajectories and executed necessary adjustments to maintain the MAV on its designated path.

During the test, an MAV equipped with cascade controllers successfully executed its inspection mission by accurately tracking its reference trajectory. We introduced random disturbances into the simulation model to evaluate the controller's resilience against external factors like winds, assessing the UAV's response. These disturbances, generated from a random signal generator function, simulated external effects such as winds or aerodynamic changes. By varying the intensity and frequency of these disturbances, we could assess the control performance under different environmental conditions and its ability to maintain a stable flight in adverse situations. The proposed digital twin platform could allow us to obtain useful information of UAV performance, control algorithms, and practical applications. For instance, Figure 14 illustrates the case of an MAV following its reference trajectory while effectively avoiding obstacles, such as tall trees and power poles. In this simulation, disturbances were injected at $t = 50$ s. Figure 14c illustrates robustness

Fig. 18 Experimental setup 1 (Nguyen et al. 2024b)

Fig. 19 Imported trajectory (Nguyen et al. 2024b)

(a) UAV during experiment (b) Bridge image captured from MAV

Fig. 20 Experimental result (Nguyen et al. 2024a)

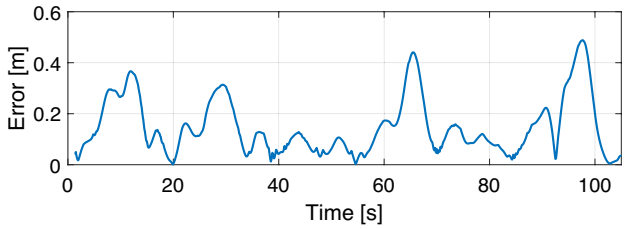


Fig. 21 Tracking error

of the controller with a steady-state error of less than 0.5 m, even subject disturbances, and its ability to maintain a stable flight with accurate trajectory tracking when inspecting a monorail bridge in the presence of obstacles. These results highlight the potential application of the digital twin technology in advancing bridge inspection technologies and engineering practices.

To further demonstrate the effectiveness of the digital twin platform, we simulated a motor failure in a quadcopter by intentionally severing the electrical signal to one motor mid-flight. This resulted in a crash, as shown in Figure 15a, indicating the quadcopter's inability to maintain stability with a failed motor. To enhance the hovering fault tolerance, a general approach besides control techniques is to increase the number of motors in multi-rotor aerial vehicles (Michieletto et al. 2017). We have tested on our visualization platform the safety landing of a hexacopter after the malfunction of a motor by redistributing the thrust applied to the drone from its reconfiguration to a quadcopter, as depicted in Figure 15b.

5.2 Case study 2: Construction site inspection

To further demonstrate the effectiveness of our proposed approach, we implemented a virtual inspection scenario at the Western Sydney International Nancy-Bird Walton Airport, which is currently a large infrastructure project in Badgerys Creek, NSW, Australia. Figure 16 (Western Sydney Airport https://westernsydney.com.au/construction) shows the construction site as of September 2022. In this simulation, we integrated 3D models to map the airport construction environment into its virtual representation of our digital twin platform. From the site imagery, the terrain along with as-built structures were replicated to virtually reflect the on-site scenario at the moment. Upon creating a visual representation to resemble the the reality as depicted in Figure 16, all prominent features including the

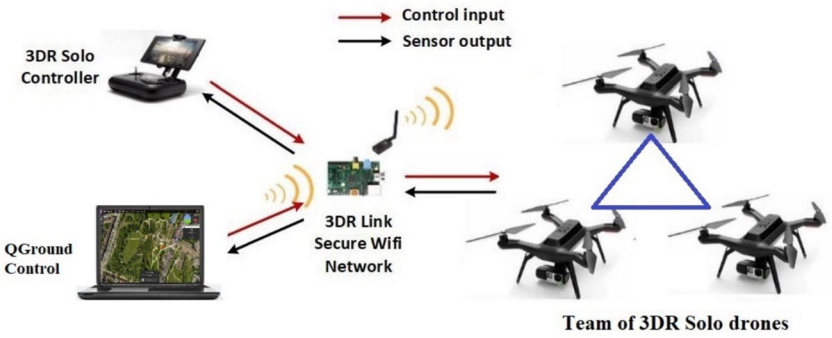


Fig. 22 Experimental setup of multiple MAVs (Nguyen et al. 2022b)

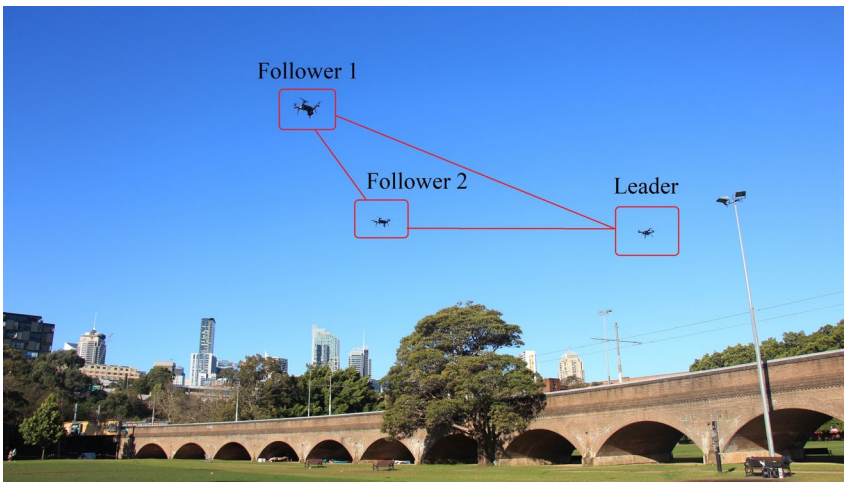


Fig. 23 UAV triangular formation during experiments (Nguyen et al. 2022b)

construction equipment, towering blocks, the main building and luffing jib cranes were carefully imported to the simulation, making it at most a mirror the site layout. Special attention was paid to the positions of those components having a certain height, such as tower cranes, which were considered as obstacles in the flight path of the drones.

The objective is to deploy a fleet of MAVs configured with specific geometrical shapes, guiding them from starting positions to designated endpoints for an on-site inspection mission. Utilizing the GPSO algorithm, optimized routes for the team are generated, ensuring they remain collision-free and maintain their formation amidst environmental obstacles, as depicted in Fig. 16. To safeguard the navigation of the drones around obstacles while adhering to their desired formation, yellow cylinders are deliberately placed around the tower mast to mark threat zones. Moreover, the altitudes of the MAVs are constrained based on the heights of unfinished buildings and the minimum height of the crane's boom. Employing a cascade trajectory tracking controller, each MAV accurately follows its planned route by continuously monitoring its position and velocity, making necessary adjustments to keep it on course.

In initial testing, three MAVs with cascaded controllers successfully performed their mission, tracking the desired trajectory as expected during normal operations. To evaluate the robustness of the controller against external disturbances, we also injected a signal random generator function into the model as a random disturbance to show the team ability to maintain the tracking performance. Through visualization, it can be observed that the controller adapted to varying external conditions such that the MAVs were able to safely follow their reference trajectory without collision, as depicted in Figure 17. However, in some circumstances, it could also reveal a tracking error associated with a control scheme, indicating via a deviation from the desired trajectory to be observed on the visualization platform. Thus, by assessing this information, one can apply and test an advanced control technique to improve the tracking performance across different scenarios. This underscores the advantage of the proposed digital twin platform in the facilitation of the high-level control design for MAV path planning in civil infrastructure inspection.

The simulation results highlight the promising application of the digital twin technology in validating optimal routes and trajectories for MAVs, ensuring safe and efficient completion of an inspection mission. This real-life scenario suggests the capability of the visualization platform in testing high-level control schemes for MAVs in the inspection of a dynamic environment such as construction sites, where moving equipment and fast-evolving structures could pose difficulties for path planning and trajectory tracking. By providing a virtual replica of the environment, the digital twin model aids in optimizing the flight path of MAV fleets before deploying to the actual location, thereby minimizing the risk of accidents and enhancing high quality as well as overall efficiency of operations.

6 Experimental validation

This section describes the testbeds used to validate the proposed technique in inspecting a monorail bridge. The experiment utilized a 3DR Solo drone with a GoPro Hero 4 camera, illustrated in Fig. 18. The drone's hardware features two Cortex M4 168 MHz processors for accurate control, supplemented by an ARM Cortex A9 running the Arducopter flight operating system (Hoang et al. 2020). The Mission Planner ground control station facilitated autonomous flight planning and data analysis. The compact form factor and 4K resolution capabilities of the GoPro Hero 4 camera made it ideal for capturing detailed visuals in hard-to-reach areas during construction inspections. The software facilitated by Mission Planner streamlined the workflow by enabling waypoint import, flight path planning, and data logging, as demonstrated in Fig. 19.

Figure 20 presents the MAV during the experiment and the bridge image captured from the MAV, exhibiting a coincidence of the field test results with those obtained from previous digital twin simulations. The alignment of structural features, the positioning of the MAV relative to the bridge, and the overall scene resemblance are notably consistent between the experiment and the simulations, indicating the advantage of the proposed approach to path planning for MAV-based inspection. Such consistency in trajectory-following and comprehensive coverage further validates the generation of optimal paths for MAVs and streamlines the process to facilitate the delivery of bridge inspection data in less time.

In Fig. 21, the experimental results depict the tracking error between the desired path and the actual path observed during the bridge inspection. The analysis reveals that the MAV follows the predefined path, exhibiting small tracking errors of less than 0.5 m. These disparities are mainly attributed to minor inaccuracies in GPS positioning rather than with the drone's tracking controller, as the typical accuracy for GPS signals can be expected at a daily global average user range error (URE) of less than 2.0 m with 95% probability (U.S. Government https://www.gps.gov/systems/gps/performance/accuracy/). Therefore, the errors here are assumed predominantly due to GPS inaccuracies. These observations highlight the effectiveness and practicality of our proposed path-planning method for performing bridge inspections.

To further enhance inspection performance, we conducted an experiment utilizing three 3DR Solo drones. The setup comprised remote controllers, a ground control station, and communication hardware, as illustrated in

Fig. 22. Similar to controlling a single MAV, the coordinates of waypoints received from the GPSO path planning algorithm are initially converted into longitude, latitude, and altitude coordinates. The speed profiles of the MAVs are set with a reference speed of $V_{ref} = 1$ m/s. Subsequently, the waypoints and corresponding speeds are transferred to the MAVs via Mission Planner software for autonomous execution. In Fig. 23, the drones are depicted forming a triangular shape during a bridge inspection task. The objectives of cooperative path planning and formation maintenance of the three drones are successfully achieved using GPSO and a prototypical digital twin.

7 Conclusion

This paper has presented intelligent path-planning algorithms for MAV for civil infrastructure inspection. The planning of an MAV path flying along a built infrastructure asset in complex environments is formulated as an optimization problem to minimize the cost function of coverage, safety, distance, smoothness, and also formation constraints. The developed GWO and game-based path planning approaches have effectively generated optimal paths for MAVs, especially in threat-inclusive scenarios. Extensive simulation results validated by field trials have confirmed the practicality of the proposed approach and its advantage in assessing their path viability against potential collision risks before real-world deployment to enhance the overall safety of multiple MAVs in cooperative missions. Experiments with single and multiple MAVs have confirmed the effectiveness of MAV path planning in the presence of obstacles. The synergy between visualization tests and real-world task deployment can contribute to improved accuracy, reduced costs, and mitigated risk in critical tasks, such as infrastructure inspection and maintenance. Our future work will expand the framework to deal with more complicated environments in various inspection conditions and enhance its flexibility in practical applications.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

Benzaid K, Mansouri N, Labbani-Igbida O (2016) A generalized dynamical model and control approach applied to multirotor aerial systems. In: 2016 8th International Conference on Modelling, Identification and Control (ICMIC). IEEE, pp 225–230

Blasi L, D'Amato E, Mattei M, Notaro I (2022) UAV path planning in 3D constrained environments based on layered essential visibility graphs. IEEE Trans Aerosp Electron Syst 59:2359–2375

Bolourian N, Soltani MM, Albahria AH, Hammad A (2017) High-level framework for bridge inspection using LiDAR-equipped UAV. In: ISARC. Proceedings of the international symposium on automation and robotics in construction, vol 34 . IAARC Publications

Bulgakov A, Sayfeddine D, Bock T, Fares A (2020) Generation of orthomosaic model for construction site using unmanned aerial vehicle. In: ISARC. Proceedings of the international symposium on automation and robotics in construction, vol 37. IAARC Publications, pp 900–904

Google Maps. https://www.google.com/maps/@-33.875266,151.192769,145m/data=!3m1!1e3?entry=ttu. Accessed 10 June 2024

Grieves MW (2005) Product lifecycle management: the new paradigm for enterprises. Int J Prod Dev 2(1–2):71–84

Grzybowski J, Latos K, Czyba R (2020) Low-cost autonomous UAV-based solutions to package delivery logistics. In: Advanced, contemporary control: proceedings of KKA 2020-The 20th Polish Control Conference, Łódź, Poland. Springer, pp 500–507

Hasiuk FJ, Harding C, Renner AR, Winer E (2017) TouchTerrain: a simple web-tool for creating 3D-printable topographic models. Comput Geosci 109:25–31. https://doi.org/10.1016/j.cageo.2017.07.005

Hoang VT, Phung MD, Dinh TH, Ha QP (2020) System architecture for real-time surface inspection using multiple UAVs. IEEE Syst J 14(2):2925–2936

Hu M, Liu W, Lu J, Fu R, Peng K, Ma X, Liu J (2019) On the joint design of routing and scheduling for vehicle-assisted multi-UAV inspection. Future Gener Comput Syst 94:214–223

Hu Z, Lou S, Xing Y, Wang X, Cao D, Lv C (2022) Review and perspectives on driver digital twin and its enabling technologies for intelligent vehicles. IEEE Trans Intell Veh 7:417–440

Jiang F, Ma L, Broyd T, Chen K (2021) Digital twin and its implementations in the civil engineering sector. Autom Constr 130:103838

Jiang Y, Li M, Guo D, Wu W, Zhong RY, Huang GQ (2022) Digital twin-enabled smart modular integrated construction system for on-site assembly. Comput Ind 136:103594

Kaigom EG, Roßmann J (2020) Value-driven robotic digital twins in cyber-physical applications. IEEE Trans Ind Inform 17(5):3609–3619

Kok J, Gonzalez LF, Kelson N (2012) FPGA implementation of an evolutionary algorithm for autonomous unmanned aerial vehicle on-board path planning. IEEE Trans Evol Comput 17(2):272–281

Lei L, Shen G, Zhang L, Li Z (2020) Toward intelligent cooperation of UAV swarms: when machine learning meets digital twin. IEEE Netw 35(1):386–392

Leonardo: truck cement mixer. https://grabcad.com/library/truck-cement-mixer-1. Accessed 10 June 2024

Li L, Aslam S, Wileman A, Perinpanayagam S (2021) Digital twin in aerospace industry: a gentle introduction. IEEE Access 10:9543–9562

Lin S, Li F, Li X, Jia K, Zhang X (2022) Improved artificial bee colony algorithm based on multi-strategy synthesis for UAV path planning. IEEE Access 10:119269–119282

Makhadmeh SN, Al-Betar MA, Doush IA, Awadallah MA, Kassaymeh S, Mirjalili S, Zitar RA (2023) Recent advances in Grey Wolf Optimizer, its versions and applications. IEEE Access 12:22991–23028

Michieletto G, Ryll M, Franchi A (2017) Control of statically hoverable multi-rotor aerial vehicles and application to rotor-failure robustness for hexarotors. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp 2747–2752

Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61

Mohamed S. Volvo excavator. https://grabcad.com/library/volvo-excavator-4. Accessed 10 June 2024

Nguyen LV, Herrera IT, Le TH, Phung MD, Aguilera RP, Ha QP (2022a) Stag hunt game-based approach for cooperative UAVs. In: ISARC. Proceedings of the international symposium on automation and robotics in construction, vol 39. IAARC Publications, pp 367–374

Nguyen LV, Phung MD, Ha QP (2022b) Game theory-based optimal cooperative path planning for multiple UAVs. IEEE Access 10:108034–108045

Nguyen L, Le T, Ha Q (2023) Prototypical digital twin of multi-rotor UAV control and trajectory following. In: Proceedings of the 40th international symposium on automation and robotics in construction. International Association for Automation and Robotics in Construction (IAARC), pp 148–155

Nguyen LV, Le TH, Nguyen TD, Kwok NM, Ha QP (2024a) Monorail bridge inspection using digitally-twinned UAVs. In: E3S Web of Conferences, vol 496. EDP Sciences, p 04004

Nguyen L, Le T, Ha Q (2024b) Grey wolf optimization-based path planning for unmanned aerial vehicles in bridge inspection. In: 2024 IEEE/SICE International Symposium on System Integration (SII). IEEE, pp 810–815

Pan Z, Zhang C, Xia Y, Xiong H, Shao X (2021) An improved artificial potential field method for path planning and formation control of the multi-UAV systems. IEEE Trans Circuits Syst II Express Briefs 69(3):1129–1133

Pharpatara P, Hérissé B, Bestaoui Y (2016) 3-D trajectory planning of aerial vehicles using RRT. IEEE Trans Control Syst Technol 25(3):1116–1123

Phung MD, Ha QP (2021) Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization. Appl Soft Comput 107:107376

Prasad NL, Ramkumar B (2022) 3-D deployment and trajectory planning for relay based UAV assisted cooperative communication for emergency scenarios using Dijkstra's algorithm. IEEE Trans Veh Technol 72(4):5049–5063

Roger BM et al (1991) Game theory: analysis of conflict, vol 66. The President and Fellows of Harvard College, USA

Tang G, Tang C, Claramunt C, Hu X, Zhou P (2021) Geometric A-star algorithm: an improved A-star algorithm for AGV path planning in a port environment. IEEE Access 9:59196–59210

Tech R. Tower crane. https://grabcad.com/library/tower-crane-15. Accessed 10 June 2024

U.S. Government: GPS accuracy. https://www.gps.gov/systems/gps/performance/accuracy/. Accessed 10 June 2024

Western Sydney Airport: Western Sydney Airport under construction. jpg—as of September 2022. https://westernsydney.com.au/construction. Accessed 28 Sept 2022

Wu J, Luo C, Luo Y, Li K (2021) Distributed UAV swarm formation and collision avoidance strategies over fixed and switching topologies. IEEE Trans Cybern 52(10):10969–10979

Yang Y, Meng W, Li H, Lu R, Fu M(2021) A digital twin platform for multi-rotor UAV. In: 2021 40th Chinese Control Conference (CCC). IEEE, pp 7909–7913

Yue T, Zuo X, Wang L, Geng J, Zhang H (2022) Similarity relations of PID flight control parameters of scaled-model and full-size aircraft. IEEE Trans Aerosp Electron Syst 58(4):2950–2960

Zhang J, Wang R, Yang G, Liu K, Gao C, Zhai Y, Chen X, Chen BM (2022) Sim-in-real: digital twin based UAV inspection process. In: 2022 International Conference on Unmanned Aircraft Systems (ICUAS). IEEE, pp 784–801

Zou Y, Zhang H, He W (2020) Adaptive coordinated formation control of heterogeneous vertical takeoff and landing UAVs subject to parametric uncertainties. IEEE Trans Cybern 52(5):3184–3195