Proceedings of the 44th Chinese Control Conference
July 28-30, 2025, Chongqing, China

Optimal Smooth Coverage Trajectory Planning for Quadrotors in Cluttered Environment

Duanjiao Li[1], Yun Chen[1], Ying Zhang[1], Junwen Yao[1], Dongyue Huang[2*], Jianguo Zhang[2], Ning Ding[2]

1. Guangdong Power Grid Co. Ltd., Guangzhou 510220, P. R. China
E-mail: 13922751289@139.com

2. Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Shenzhen 518129, P. R. China
E-mail: huangdongyue@cuhk.edu.cn

Abstract: For typical applications of UAVs in power grid scenarios, we construct the problem as planning UAV trajectories for coverage in cluttered environments. In this paper, we propose an optimal smooth coverage trajectory planning algorithm. The algorithm consists of two stages. In the front-end, a Genetic Algorithm (GA) is employed to solve the Traveling Salesman Problem (TSP) for Points of Interest (POIs), generating an initial sequence of optimized visiting points. In the back-end, the sequence is further optimized by considering trajectory smoothness, time consumption, and obstacle avoidance. This is formulated as a nonlinear least squares problem and solved to produce a smooth coverage trajectory that satisfies these constraints. Numerical simulations validate the effectiveness of the proposed algorithm, ensuring UAVs can smoothly cover all POIs in cluttered environments.

Key Words: Traveling Salesman Problem, Genetic Algorithm, Smooth Trajectory Optimization

1 Introduction

In recent years, with the rapid development of manufacturing industries, unmanned systems have found widespread applications across various fields. Among them, quadrotors have been increasingly utilized in industrial applications such as aerial photography and surveying [1]. As electricity consumption continues to rise, the frequency of power grid maintenance has also increased. Given the high risks and costs associated with manual inspections, the importance of utilizing unmanned systems for autonomous power grid inspections has become increasingly evident [2], as shown in Fig 1. Substations, as critical components of the power grid system, play an essential role in ensuring seamless inspection across modules within the same facility or between different facilities. The units scheduled for inspection can be abstracted as a series of access points, with drones acting as agents tasked with visiting these points. Consequently, the key focus of this paper is to explore how quadrotors can efficiently cover all inspection points while minimizing energy costs.

In cluttered environments with obstacles, a broad range of motion planning strategies has been extensively studied. Most focus on optimality guarantees and static obstacle avoidance capabilities, such as Expansive Space Trees (EST) [3], Probabilistic Roadmap Methods (PRM) [4], Rapidly exploring Random Trees (RRT) [5], and optimal RRT (RRT*) [6]. Additionally, as noted in [7], approaches like multi-RRT* Fixed Node (RRT*FN) and genetic algorithm (GA) have been explored for UAV motion planning in cluttered environments. Yet, these methods fail to adequately address the trajectory generation problem for feasible UAV paths. Moreover, these methods lack real-time applicability and are less suitable for dynamic environments due to high computational costs. Additionally, some works, such as [10], consider not only the shortest path as the sole criterion when addressing the Traveling Salesman Problem (TSP), problem but also incorporate the UAV's dynamic model into the planning process. However, this approach does not take obstacle avoidance into account.

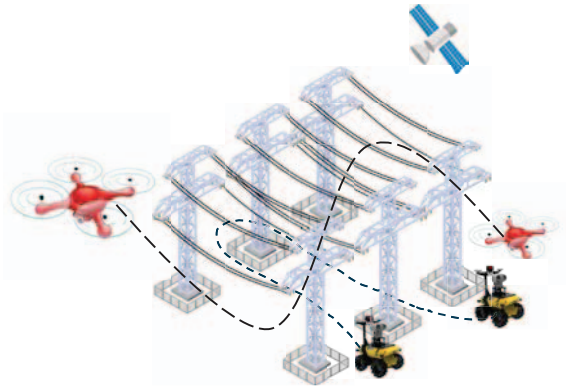


Fig. 1: Sketch map of unmanned system operating inspection in the power grid system.

For the trajectory generation of UAVs, several studies have been conducted. For instance, [8] generates trajectories with jerk limits, while [9] employs optimal smoothing B-splines and validates the idea through physical experiments. However, these works do not focus on the critical aspect of efficiently accessing predefined trajectory points.

In this paper, we address the defined problem: how to generate smooth and feasible trajectories for UAVs to cover all inspection targets under limited resources. We propose a two-scale solution. At the first scale, without considering obstacles, all target points are formulated as a TSP, which is an NP-hard problem, solved by a GA. At the second scale, based on the visitation sequence obtained from the first scale, we use an optimal B-spline to generate trajectories. During this process, obstacles represented by Euclidean Distance Transforms (EDT) are incorporated as constraints to ensure

*Corresponding author

The work was supported in part by Grant BN00202401031, and in part by the Longgang District Shenzhen's "Ten Action Plan" for Supporting Innovation Projects under Grants LGKCSDPT2024002 and LGKCSDPT2024003.

the generated trajectory is feasible for UAV following.

The remaining of the paper is organized as follows: problem statement is presented in Section II. The proposed algorithms are highlighted in Section III. The numerical simulations are conducted in Section IV. Eventually, some conclusions are remarked in the last section.

2 Problem Statement

This paper abstracts the inspection targets in a substation into multiple points, defined as points of interest (POIs), within a 2D cluttered environment containing numerous obstacles. The goal is to generate a smooth trajectory for a UAV to efficiently visit all the inspection points. A quadrotor starts from an initial position, visits all POIs, and then moves to the endpoint along the shortest path while avoiding obstacles. During this process, the UAV navigates along a cost-minimizing path while adhering to dynamic constraints, ensuring the generated trajectory is executable by the UAV. The proposed algorithm is scalable, capable of handling large and geometrically complex areas. Using sensor-equipped quadrotor to perform coverage tasks presents significant potential for future applications. These flying sensors can serve the same purpose as a network of static sensors, offering rapid, adaptive, and cost-effective solutions for diverse operational scenarios.

2.1 GA for TSP

In this problem, we aim to determine the minimum distance path starting from a home base point p_0, visiting all points of interest p_i for $i = 1, \ldots, N-1$, each exactly once. The objective is to determine a permutation $\mathcal{S}$ of the sequence:

$$\mathcal{S} = [s_0, s_1, \ldots, s_{N-1}], \tag{1}$$

For each route $s_i \in \mathcal{S}$, the total path length objective function $\mathcal{C}(s)$ is defined, which includes energy and time consumption factors. The total path length is represented as follows:

$$\mathcal{C}(s) = \sum_{i=0}^{N-1} d(s_i, s_{i+1}) E_\mathrm{f} + d(s_i, s_{i+1}) T_\mathrm{f} \tag{2}$$

where inter-distances $d(s_i, s_j)$ are determined from the set of POIs in the given space. E_f and T_f are the energy consumption factor and time consumption factor, respectively.

This problem, known as the TSP, is an NP-hard optimization problem, and we propose to use GA to solve this problem and obtain a near-optimal path. The proposed GA-based solution initializes a population of possible paths and iteratively improves the population by applying selection, crossover, and mutation operations. The optimization is subject to a defined fitness function which aims to minimize the total path length while ensuring that each point is visited only once and the path returns to the end point.

One might use in the genetic algorithm, the selection operation is based on individual fitness (i.e., the inverse of the objective function value). A roulette wheel selection method is used, and the The cumulative probability is defined and determined as follows:

$$\mathbf{P}_i = \sum_{j=1}^{i} \frac{1/\mathcal{C}(s)_i}{\sum_{j=1}^{N} (1/\mathcal{C}(s)_j)} \tag{3}$$

2.2 Optimized B-Spline

In this problem, we formulate the optimization points as a nonlinear least squares problem. The cost function is defined as a weighted summation of multiple components, including trajectory smoothness ϵ_s, obstacle avoidance ϵ_e, boundary points ϵ_b, time consumption ϵ_t, and POIs ϵ_p. The parameter ω with a suffix represents the weight assigned to the corresponding cost function. The goal is to generate a smooth coverage trajectory while minimizing the overall cost, defined as ϵ_total. The boundary functions are defined as $g(x)$ and $h(x)$. The mathematical formulation of the cost function is as follows:

$$\begin{aligned} \min_{x} \quad & \frac{1}{2} \|\epsilon_\mathrm{total}\|_2^2 \\ \text{s.t.} \quad & g(x) \leq 0, \\ & h(x) = 0. \end{aligned}$$

where

$$\epsilon_\mathrm{total} = \omega_\mathrm{s}\epsilon_\mathrm{s} + \omega_\mathrm{b}\epsilon_\mathrm{b} + \omega_\mathrm{e}\epsilon_\mathrm{e} + \omega_\mathrm{t}\epsilon_\mathrm{t} + \omega_\mathrm{p}\epsilon_\mathrm{p}. \tag{4}$$

For the cost function in the objective formulation, the trajectory smoothness is directly evaluated using velocity, acceleration, and jerk as components of the cost function. The time consumption is addressed by introducing a time-scaling variable, which adjusts the time scale as needed. Additionally, the cost for POIs is directly calculated using the Euclidean distance. For the obstacle cost function, we utilize values derived from the EDT as the specific metric for this cost. A weighting function is applied to regulate its influence within the overall optimization. Additionally, a target safety distance is set to ensure that no safety issues arise when applied to objects of a certain volume.

2.3 Model of Quadrotors

Before introducing the model of quadrotors, we defined two coordinates, which are body frame, attached at the center of gravity of vehicle and inertial frame, represented by NED (North-East-Down), as shown in Fig. 2. The states of the model is described as $\boldsymbol{\nu} = [u\ v\ w\ p\ q\ r]^\mathrm{T}$, represented in the body frame, the pose variable $\boldsymbol{\eta} = [x\ y\ z\ \phi\ \theta\ \psi]^\mathrm{T}$, both represented in the inertial frame. Defined the transformation matrix as $\boldsymbol{J} = \mathrm{diag}([\boldsymbol{R}_\mathrm{b}^\mathrm{e}, \boldsymbol{S}^{-1}])$ [11].

As what stated in [12], the candidates UAV model is described as

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}\boldsymbol{\nu}, \tag{5}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{\tau}_\mathrm{d}. \tag{6}$$

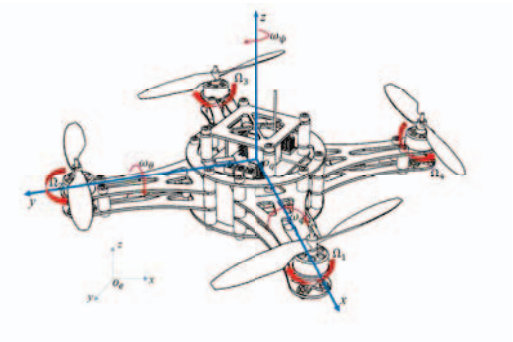


Fig. 2: Coordinates of the quadrotors

where M is the inertial matrix, $C(\nu)$ is the Coriolis and centripetal matrix, $g(\eta)$ is the restoring force and moment vector, τ is the input force/torque represented under the body frame, and τ_{d} is the disturbance. One notices that the input τ is indirectly affected by rotational speed of the motors.

3 Algorithms Description

In this section, we mainly introduce the algorithm used in planning the optimal smooth coverage trajectory, as shown in Fig 3. Firstly, we give a detailed description on the GA for optimizing the TSP. Then a detailed optimized b-spline algorithm is introduced.

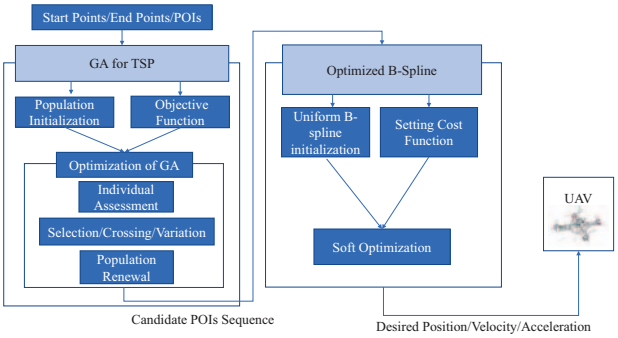


Fig. 3: Overview of the proposed planning algorithm.

3.1 GA for TSP Optimization

The proposed algorithm employs a GA to solve the TSP, aiming to minimize the total path cost, which incorporates both energy and time consumption. The algorithm starts by initializing a population $\mathcal{P}$, consisting of random permutations of POIs excluding the fixed start and end points. The initial best sequence of the traveling POIs and cost are computed using an objective function, which calculates the energy consumption and travel time for each segment of the route. The optimization process runs for $\mathcal{G}_{\max}$ generations, where in each generation, the individuals are evaluated to determine their fitness based on the objective function. The best solution is updated if a better sequence is found. The 'PopulationEvoFunc' function is then used to evolve the population by selecting the top-performing individuals for crossover and mutation, resulting in offspring paths that are fixed by the 'RepairPath' function to ensure validity, i.e., that each POI is visited exactly once.

The cost function $\mathcal{C}$, as defined in (2), involves calculating the energy consumption and time required for traversal between POIs, summed over all consecutive POIs pairs along the path. During the evolution phase, crossover operations are performed by selecting crossover points between two individuals to produce offspring, which are then repaired to remove any duplicated POI while maintaining the fixed start and end points. The algorithm terminates after $\mathcal{G}_{\max}$ generations, returning the best path found and its associated cost, thus providing an optimized sequence of POIs, denoted as $\mathcal{S}^*$, visits that considers both energy and time efficiency.

3.2 Optimized B-Spline

In this subsection, we start by initializing a binary grid (`bw`) to represent the environment. Each obstacle (`obs`) is defined by its coordinates, and the corresponding grid index is computed using the `pos2grid` function. It converts coordinates `(x,y)` into grid indices `(idx,idy)` by dividing by step, rounding. For each obstacle, we mark the cells within its defined radius (r_{obs}) as occupied in the binary map. The resolution of the grid is determined by a step

Algorithm 1 Genetic Algorithm

Input: N, d, size($\mathcal{P}$), $\mathcal{G}_{\max}$, E_{f}, T_{f}
Output: $\mathcal{S}^*$, $\mathcal{C}_{\min}$
Initialization: $\mathcal{P} \in \mathbb{R}^{\mathrm{size}(\mathcal{P})\times(N-2)}$ for each individual (excluding start and end)
$\mathcal{S}^* \leftarrow [1, \mathcal{P}(1,:), N]$
$\mathcal{C}_{\min} \leftarrow$ `ObjectiveFunc`$(\mathcal{S}^*, d, E_{\mathrm{f}}, T_{\mathrm{f}})$
for $\mathcal{G}_{\max}$ **do**
  for $i = 1 \rightarrow$ size($\mathcal{P}$) **do**
    $\mathcal{S} \leftarrow [1, \mathcal{P}(i,:), N]$
    $\mathcal{C}(i) \leftarrow$ `ObjectiveFunc`$(\mathcal{S}, d, E_{\mathrm{f}}, T_{\mathrm{f}})$
  end for
  `currentMinCost, idx` $\leftarrow \min(\mathcal{C})$
  if `currentMinCost` $< \mathcal{C}_{\min}$ **then**
    $\mathcal{C}_{\min} \leftarrow$ `currentMinCost`
    $\mathcal{S}^* \leftarrow [1, \mathcal{P}(\mathrm{idx},:), N]$
  end if
  $\mathcal{P} \leftarrow$ `PopulationEvoFunc`$(\mathcal{P}, \mathcal{C}, N)$
end for

function POPULATIONEVOFUNC($\mathcal{P}, \mathcal{C}, N$)
  $\mathcal{P}_{\mathrm{s}} \leftarrow$ top 50% individuals based on costs
  $\mathcal{P}_{\mathrm{n}} \leftarrow \mathcal{P}_{\mathrm{s}}$
  while $|\mathcal{P}_{\mathrm{n}}| < |\mathcal{P}|$ **do**
    $i, j \leftarrow$ random selection from selected
    if $i \neq j$ **then**
      `Prt`$_{\mathrm{c}} \leftarrow$ random index
      `RPath1` $\leftarrow [\mathcal{P}_{\mathrm{s}}(i, 1:\mathrm{Prt_c}), \mathcal{P}_{\mathrm{s}}(j, \mathrm{Prt_c}{+}1:\mathrm{end})]$
      `RPath2` $\leftarrow [\mathcal{P}_{\mathrm{s}}(j, 1:\mathrm{Prt_c}), \mathcal{P}_{\mathrm{s}}(i, \mathrm{Prt_c}+1:\mathrm{end})]$
      `RPath1` $\leftarrow$ `RepairPath`$(\mathrm{RPath1}, N-2)$
      `RPath2` $\leftarrow$ `RepairPath`$(\mathrm{RPath2}, N-2)$
      $\mathcal{P}_{\mathrm{n}} \leftarrow$ append `RPath1, RPath2`
    end if
  end while
  Return: $\mathcal{P}_{\mathrm{n}}$
end function

function OBJECTIVEFUNC($\mathcal{S}, d, E_{\mathrm{f}}, T_{\mathrm{f}}$)
  $E \leftarrow 0, T \leftarrow 0$
  for $i = 1 \rightarrow$ size($\mathcal{S}$) $- 1$ **do**
    $E \leftarrow E + E_{\mathrm{f}} \cdot d(\mathcal{S}(i), \mathcal{S}(i+1))$
    $T \leftarrow T + T_{\mathrm{f}} \cdot d(\mathcal{S}(i), \mathcal{S}(i+1))$
  end for
  $\mathcal{C} \leftarrow E + T$
  Return: $\mathcal{C}$
end function

function REPAIRPATH(`path`, N)
  `RPath` $\leftarrow$ `path`
  for $i = 1 \rightarrow$ size `(path)` **do**
    if $\sum$(`RPath` $==$ `RPath`(i)) > 1 **then**
      `unusedPrt` $\leftarrow \{2, \ldots, N+1\} \setminus$ `RPath`
      `RPath`$(i) \leftarrow$ `unusedPrt`(1)
    end if
  end for
  Return: `RPath`
end function

Table 1: UAV Parameters

Mass (kg)	0.5
Moment of Inertia (kg · m^2)	Diag(2.93,2.32,4) $\times 10^{-3}$
Wheelbase (m)	0.3
Thrust Coefficient (N/rpm^2)	6.11 $\times 10^{-8}$
Moment Coefficient (Nm/rpm^2)	1.50 $\times 10^{-9}$

size (λ), which influences the granularity of obstacle representation. Finally, we compute the `EDT` of the grid, which provides a distance map indicating the proximity of each unoccupied cell to the nearest obstacle, giving a continuous representation of obstacle influence in the environment.

Algorithm 2 Obstacle Map

Input: `obs`, λ, r_{obs}
Output: `EDT`
Initialize binary map `bw` of the environment.
Set step λ and obstacle radius r_{obs}.
for each obstacle in `obs` **do**
    Calculate grid index for obstacle position: `(idx, idy)` ← `pos2grid(obs)`
    for each `x_offset` and `y_offset` within `obs_radius` **do**
        if distance is within `obs_radius` **then**
            Calculate new indices in grid: `(idx, idy)`
            if `idx`, `idy` are within the bounds of `bw` **then**
                Set `bw[idx, idy]` = 1
            end if
        end if
    end for
end for
`EDT` ← $\lambda \cdot$ `bwdist(bw, 'euclidean')`
Return: `EDT`

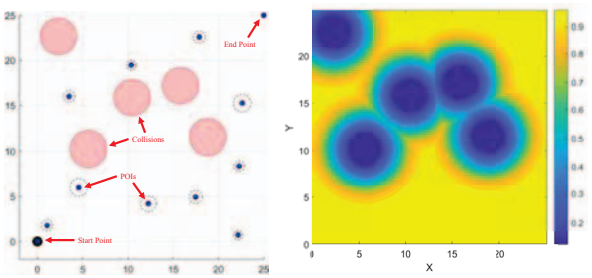
Then, we construct a cost function based on the established EDT obstacle map, and incorporate it into the B-spline optimization process for trajectory planning. Additionally, we consider other costs such as trajectory smoothness, as described in (1). Furthermore, the sequence of visiting points obtained from the GA is also incorporated as a cost term. The overall goal of this process is to solve for the optimal control points for the trajectory.

4 Numerical Simulations

In the simulation, we applied our algorithm in a quadrotor in a clutter environment with traveling all POIs, with specific parameters listed in Table 1. As the proposed planning algorithm is 2D and depending on mapping, assuming that all the positions of the collisions are pre-known, and all the POIs are located at the height of 5 m.

The scenario in this numerical simulation is depicted in Fig 4. Specifically, as shown in Fig. 4(a), red areas represent obstacles, while the blue points enclosed by dashed lines indicate the POIs. The dashed lines denote the coverage range, simulating the maximum field of view of UAV. The starting and ending points are also marked. The UAV begins at the starting point, visits each POI while avoiding obstacles, and ultimately reaches the endpoint to complete the inspection task. Additionally, the cost map generated using the EDT is shown in Fig. 4(b).

We apply the proposed algorithm to the UAV and scenario described above, and the resulting numerical simulations are shown below. As depicted in Fig. 5, the green line represents the smooth coverage trajectory planned by the algorithm, while the blue line illustrates the actual trajectory executed by the UAV. Details of each state of the UAV are presented in the Fig. 7. The first row shows the roll, pitch, and yaw angles over time. The second row illustrates the velocities. The third row depicts the position, respectively. The blue line represents the reference trajectory from the planner, while the red line indicates the UAV trajectory execution.



(a) (b)

Fig. 4: Maps for operating. (a) The top view of the map for operating. (b) Cost map generated through EDT.

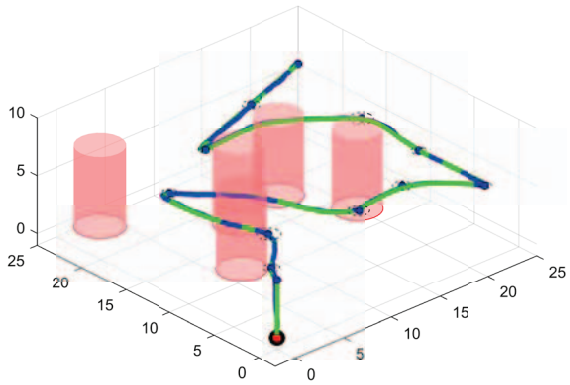
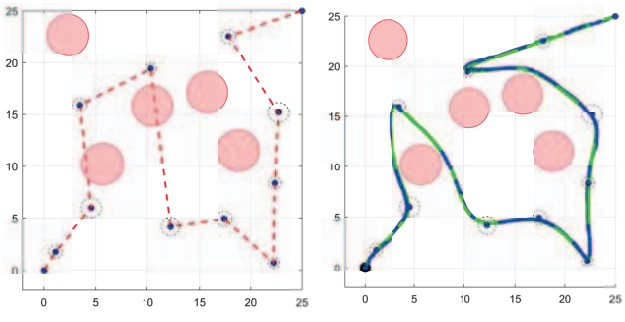


Fig. 5: The trajectory generated in the full cycle numerical simulation. The green line denotes the planned trajectory. The blue line denotes the estimated UAV trajectory.



(a) (b)

Fig. 6: Results of the simulations. (a) The top view of the simulation after the optimization of GA. (b) The top view of the simulation after the optimization of b-spline.

In Fig. 6(a), the POIs are first processed through the GA to determine an initial sequence of visitations, represented by the red dashed line. Since obstacle constraints are ignored during this optimization, the resulting sequence serves

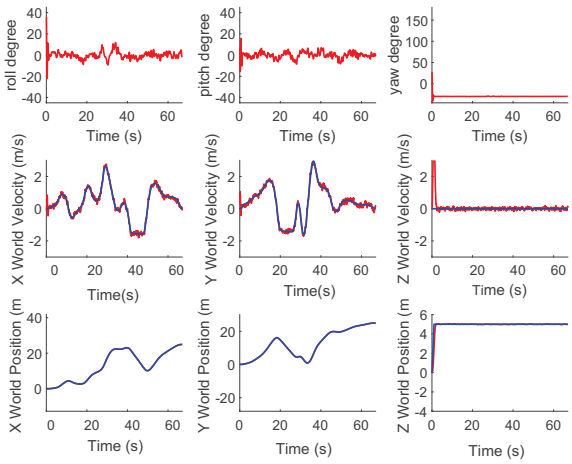


Fig. 7: Numerical simulation results of the state variables during the simulation.

as a preliminary solution. Next, the GA-optimized sequence is fed into the optimized B-spline algorithm, producing a smooth trajectory that the UAV can feasibly follow, as shown in Fig. 6(b). It is noticeable that the POIs visitation order changes after incorporating the optimized B-spline. This change arises from the need to account for obstacle avoidance, minimize traversal costs, and ensure smooth UAV operation, leading to adjustments in the cost function. It is also reflected in the detailed state variations. The UAV avoids large angular maneuvers, which not only contributes to energy savings but also aids in maintaining alignment with inspection targets during operation. These advantages are particularly significant in real-world industrial scenarios, ensuring higher inspection quality and operational efficiency.

The results from Fig. 7 demonstrate the effectiveness of the proposed algorithm in generating smooth and collision-free trajectories for UAV operations in cluttered environments. The roll and pitch angles exhibit minor oscillations, reflecting robust stabilization, while the yaw remains stable, indicating consistent heading control. The X and Y velocities show periodic variations aligned with acceleration and deceleration phases, ensuring energy-efficient motion between POIs. The positional trajectories in X and Y directions are smooth and closely follow the planned path, confirming high tracking accuracy. Additionally, the constant Z position highlights reliable altitude maintenance, essential for stable sensor operations. Overall, the control system handles disturbances effectively, ensuring real-time applicability and energy-efficient performance during inspection tasks.

5 Conclusion

In this paper, we have proposed an optimization planning algorithm for UAVs to generate smooth and coverage trajectories for visiting a series of POIs in cluttered environments. Specifically, we have first employed a GA to solve the TSP for candidate POIs and subsequently optimized the results using an optimal B-spline, incorporating cost functions such as smoothness and obstacle avoidance. Through the application of the proposed algorithm in UAV simulations, we have demonstrated its effectiveness. The simulation results have shown that during the full cycle, the UAV has not exhibited any significant angular maneuvers, with the maximum attitude angle variation remaining below 20 degrees. Additionally, the velocity profile has been smooth without any step-like changes. These simulation outcomes have proven beneficial for the application of this algorithm in UAV inspection tasks, laying a solid foundation for future real-world deployments.

References

[1] D. Huang, M. Dou, X. Liu, X. Wang, C. Wang, and B. M. Chen, Aqua Slide: An Underwater Leveling Motion Scheme for M-UAAV Utilizing Singularity, in *IEEE Transactions on Industrial Electronics*, in press, 2024.

[2] Z. Zhou, C. Zhang, C. Xu, F. Xiong, Y. Zhang, and T. Umer, Energy-Efficient Industrial Internet of UAVs for Power Line Inspection in Smart Grid, in *IEEE Transactions on Industrial Informatics*, 14(6), 2705 - 2714, 2018.

[3] J. M. Phillips, N. Bedrossian, and L. E. Kavraki, Guided Expansive Spaces Trees: a search strategy for motion- and cost-constrained state spaces, in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2004: 3968—3973.

[4] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, Probabilistic roadmaps for path planning in high-dimensional configuration spaces, in *IEEE Transactions on Robotics and Automation*, 12(4), 566 – 580, 1996.

[5] P. Pharpatara, R. Pepy, B. Hérissé, and Y. Bestaoui, Missile trajectory shaping using sampling-based path planning, in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013: 2533–2538.

[6] S. Karaman, and E. Frazzoli, Optimal Kinodynamic Motion Planning using Incremental Sampling-Based Methods, in *Proceedings of IEEE Conference on Decision and Control (CDC)*, 2010:7681–7687.

[7] Y. Bouzid, Y. Bestaoui, and H. Siguerdidjane, Quadrotor-UAV Optimal Coverage Path planning in cluttered environment with a limited onboard energy, in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2017: 979–984.

[8] S. Lai, M. Lan and B. M. Chen, Efficient safe corridor navigation with jerk limited trajectory for quadrotors, in *Proceedings of 2018 37th Chinese Control Conference (CCC)*, 2018: 10065-10070.

[9] S. Lai, M. Lan, and B. M. Chen, Optimal constrained trajectory generation for quadrotors through smoothing splines, in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018: 4743-4750.

[10] N. Michel, A. Patnaik, Z. Kong, and X. Lin, Energy-Optimal Planning of Waypoint-Based UAV Missions–Does Minimum Distance Mean Minimum Energy?. in *arXiv preprint* arXiv:2410.17585, 2024.

[11] G. Cai, B. M. Chen, and T. H. Lee, *Unmanned rotorcraft systems*. Springer Science & Business Media, 2011.

[12] D. Huang et al., First Principle Modeling of a Morphable Unmanned Aerial-Aquatic Vehicle: Mirs-Alioth, in *Proceedings of 2024 IEEE 18th International Conference on Control & Automation (ICCA)*, 2024: 540-545