A Structure-Aware Reactive Motion Planner for Autonomous Nano-UAV Navigation in Confined 3D Tunnels Using Minimal Range Sensing

Abner Asignacion Jr.*, and Satoshi Suzuki
Chiba University, Japan

ABSTRACT

Inspection of confined tunnel environments remains challenging for small robotic platforms. While snake and soft robots can operate in highly constrained spaces, their effectiveness degrades in long, elevated, and vertically structured tunnels. Micro- and nano-UAVs offer a compelling alternative, but their deployment is severely constrained by limited sensing and onboard computation, rendering map-based planning impractical. This paper presents an autonomous structure-aware motion planning (SAMP) strategy for nano-UAV navigation in narrow tunnels using only sparse range sensing. The proposed method explicitly exploits short-horizon structural regularities in the sensed environment. A wall-extended local referencing mechanism is introduced to construct a virtual target ahead of the vehicle, enabling anticipatory motion while preserving a fully reactive and lightweight formulation. The approach requires no mapping, memory, or global optimization, making it suitable for resource-constrained aerial platforms. Extensive simulations and real-world experiments using a nano-UAV in modular tunnel environments demonstrate that SAMP achieves significantly improved traversal robustness compared to representative reactive baselines, particularly in scenarios involving horizontal–vertical transitions and complex junctions.

1 INTRODUCTION

Unmanned aerial vehicles (UAVs) have become widely-used for inspection and exploration in environments that are hazardous, inaccessible, or structurally complex. In particular, confined infrastructures such as tunnels, ducts, chimneys, and underground facilities present strong application demands for autonomous inspection [1, 2, 3, 4]. While ground-based platforms such as snake robots and soft robots can effectively traverse highly constrained spaces [5, 6, 7], their deployment becomes increasingly limited in long, elevated, and vertically structured tunnel networks. In contrast, micro- and nano-UAVs offer superior mobility and the ability to access multi-level structures, making them promising candidates for confined-space inspection.

Despite this potential, autonomous navigation of nano-UAVs in narrow tunnels remains fundamentally challenging. The severe size constraints of these platforms limit both sensing and onboard computation, often restricting the available perception to sparse range measurements and lightweight optical flow. At the same time, confined environments impose tight geometric constraints and complex aerodynamic interactions, including ground, ceiling, and wall effects, which significantly influence flight stability [8, 9]. These factors render conventional navigation pipelines impractical for nano-UAV systems, typically based on mapping, localization, and global planning.

Existing motion planning approaches in robotics, such as graph-based search, model predictive control, and map-based local planning, generally rely on dense sensing and computational resources [10, 11, 12, 13, 14, 15, 16]. While such methods have demonstrated success in larger UAV platforms, they are not directly applicable to resource-constrained nano-UAVs. Recent works have explored lightweight navigation strategies using minimal sensing, including optical-flow-based control and reactive obstacle avoidance [17, 18]. Reactive methods such as Bug algorithms and wall-following strategies offer computational simplicity, but they are inherently myopic, relying on immediate obstacle interactions without exploiting higher-level environmental structure. As a result, their performance degrades in tunnel environments involving junctions, vertical transitions, or extended corridors.

This paper argues that confined environments such as tunnels exhibit *implicit structural regularities* that can be exploited even under minimal sensing. Unlike general unstructured environments, tunnel-like spaces possess locally consistent geometric patterns (e.g., corridor continuity, wall alignment, and directional openings) that can provide navigational cues beyond instantaneous obstacle avoidance. However, existing lightweight planners largely fail to incorporate such structural information in a computationally efficient manner.

To address this gap, we propose a *Structure-Aware Motion Planning (SAMP)* strategy for nano-UAV navigation in confined environments using only sparse range sensing. The key idea is to introduce a short-horizon geometric interpretation of the environment by constructing a *wall-extended local*

*Email address(es): abner_asignacion@chiba-u.jp

reference ahead of the vehicle. This virtual reference point acts as a directional guide, enabling anticipatory motion while maintaining a fully reactive and lightweight formulation. Unlike conventional approaches that rely on wall contact, centerline balancing, or cautious probing, the proposed method explicitly leverages local structural cues to improve navigation consistency and robustness.

The proposed planner operates without mapping, memory, or global optimization, making it suitable for platforms with strict sensing and computational constraints. To evaluate its effectiveness, we conduct comparative simulation studies against other known motion planners. Validation is performed through both simulations and real-world experiments using a nano-UAV platform equipped with optical flow and multiranger sensors in modular tunnel environments with horizontal and vertical segments.

The main contributions of this work are summarized as follows:

- A novel **structure-aware reactive motion planning** framework for nano-UAVs operating under minimal sensing constraints.
- A **wall-extended local referencing** mechanism that enables anticipatory navigation without requiring mapping or global state estimation.
- A systematic **comparative evaluation** in confined tunnel environments, including real-world validation.

The remainder of this paper is organized as follows. Section II presents the proposed structure-aware motion planning method. Section III describes the simulation setup and evaluation framework. Section IV presents experimental results and discussion.

2 METHODOLOGY

We propose a *Structure-Aware Motion Planning (SAMP)* framework for nano-UAV navigation in confined tunnel environments using minimal range sensing. The approach is designed to operate under strict sensing and computational constraints, without relying on mapping, memory, or global optimization.

The key idea is to exploit short-horizon structural regularities in tunnel-like environments by constructing a local geometric reference ahead of the vehicle. This enables anticipatory motion while preserving a fully reactive formulation. The planner operates in two main components: (i) a structure-aware local planner based on wall-extended referencing, and (ii) a mode-switching framework for handling mixed horizontal and vertical tunnel geometries.

2.1 *Structure-Aware Local Planning*

Unlike conventional reactive methods such as wall following or centerline balancing, the proposed SAMP explicitly interprets the local corridor geometry to guide motion. Instead of reacting only to instantaneous obstacle measurements, the planner constructs a short-horizon *virtual reference point* (carrot point) ahead of the vehicle.

2.1.1 Frontal Structure Evaluation

To determine whether the local environment exhibits sufficient structural regularity, the controller evaluates the temporal consistency of forward range measurements. Let $\{r_F^{(i)}\}_{i=1}^N$ denote the recent forward measurements. Their mean and standard deviation are computed as

$$\bar{r}_F = \frac{1}{N}\sum_{i=1}^{N} r_F^{(i)}, \quad \sigma_{r_F} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}\left(r_F^{(i)} - \bar{r}_F\right)^2}. \tag{1}$$

A structure-aware mode is activated when the forward obstacle is both within a relevant range and temporally stable:

$$d_{\text{safe}} + \delta < \bar{r}_F < r_{\max}, \qquad \sigma_{r_F} < \sigma_{\text{th}}. \tag{2}$$

This condition ensures that the detected corridor geometry is sufficiently consistent to support short-horizon extrapolation.

2.1.2 Wall-Extended Local Reference

Once activated, the planner constructs a local reference point ahead of the UAV. The forward extent is defined as

$$\Gamma = \text{clip}(r_F - d_{\text{stop}}, 0, \Gamma_{\max}), \tag{3}$$

while the lateral offset is determined by the imbalance between left and right wall distances:

$$y_c = k_y(r_L - r_R). \tag{4}$$

The resulting carrot point is given by

$$(x_c, y_c) = (\Gamma, y_c). \tag{5}$$

This construction can be interpreted as a short-horizon extension of the corridor geometry, enabling the UAV to steer toward an anticipated free-space direction rather than reacting purely to local measurements.

The desired heading toward the carrot point is computed as

$$\psi = \tan^{-1}\left(\frac{y_c}{\max(x_c, \varepsilon)}\right), \tag{6}$$

and the planar velocity commands are defined as

$$v_x = v\cos\psi, \quad v_y = v\sin\psi, \tag{7}$$

with speed

$$v = \text{clip}(k_c x_c, 0, v_{\text{nom}}). \tag{8}$$

The yaw rate aligns the vehicle with the reference direction:

$$\omega_z = \text{clip}(k_\psi \psi, -\omega_{\max}, \omega_{\max}). \tag{9}$$

This formulation enables smooth, anticipatory motion while maintaining computational simplicity.

2.1.3 Fallback and Safety Behavior

When the forward structure is not sufficiently stable, the planner reverts to a simpler reactive strategy based on forward clearance and lateral centering. Additionally, velocity commands are modulated based on proximity to obstacles:

$$v_x \leftarrow \min(v_x, v_{\mathrm{slow}}), \quad \text{if } r_F < d_{\mathrm{slow}}. \tag{10}$$

All commands are further constrained by instantaneous safety envelopes derived from current range measurements, ensuring collision avoidance under all conditions.

2.1.4 Mode Switching for Vertical Tunnels

To enable navigation in tunnels with vertical shafts, the proposed method extends the same structure-aware logic through a mode-switching and virtual frame formulation. The planner operates in three modes: horizontal (H), ascending (ASC), and descending (DESC). Transitions are triggered based on the availability of free space. For example, ascent is selected when forward motion is constrained while sufficient upward clearance exists:

$$r_F < d_{turn}, \quad r_U > d_{open}. \tag{11}$$

Similarly, descent is selected when downward clearance dominates. Transitions back to horizontal motion occur when vertical motion becomes constrained and horizontal openings reappear.

A key component of the proposed framework is the use of a *virtual navigation frame*. Regardless of the active mode, the progress direction is always treated as a virtual forward direction:

$$(r_F, r_B, r_L, r_R) = \begin{cases} (r_F^0, r_B^0, r_L^0, r_R^0), & \text{H}, \\ (r_U^0, r_D^0, r_L^0, r_R^0), & \text{ASC}, \\ (r_D^0, r_U^0, r_L^0, r_R^0), & \text{DESC}. \end{cases} \tag{12}$$

This allows the same structure-aware planning logic to be reused across horizontal and vertical motion without modification.

The computed velocity commands are mapped back to the physical frame by swapping the progress axis:

$$(v_x, v_z) = \begin{cases} (v_{\mathrm{nom}},\ v_z^{\mathrm{alt}}), & \text{H}, \\ (v_z^{\mathrm{alt}},\ +v_{\mathrm{nom}}), & \text{ASC}, \\ (v_z^{\mathrm{alt}},\ -v_{\mathrm{nom}}), & \text{DESC}. \end{cases} \tag{13}$$

Safety constraints are enforced in the physical frame to guarantee collision avoidance.

The proposed SAMP framework combines reactive control with short-horizon structural reasoning, enabling robust navigation in confined environments using minimal sensing. By integrating wall-extended referencing with virtual frame remapping, the method achieves consistent performance across complex tunnel geometries while maintaining low computational cost.

Table 1: Control parameters used in both simulation and experiment.

Parameter	Symbol	Value
Forward gain	k_x	1.2
Lateral gain	k_y	1.8
Vertical gain	k_z	1.5
Yaw gain	k_ω	0.9
Nominal velocity	v_{nom}	0.16 m/s
Maximum velocity	v_{max}	0.25 m/s
Maximum yaw rate	ω_{max}	1.2 rad/s
Creep velocity	v_{creep}	0.06 m/s
Stop threshold	d_{stop}	0.18 m
Turn threshold	d_{turn}	0.24 m
Slowdown threshold	d_{slow}	0.36 m
Open-space threshold	d_{open}	0.50 m
Maximum look-ahead distance	Γ_{max}	0.35 m
Maximum vertical dwell time	$t_{\mathrm{nav}}^{\mathrm{max}}$	1.0 s
Future horizon	N	8

3 SIMULATION RESULTS

The simulation results were obtained using a MATLAB-based implementation of the proposed and baseline motion planning algorithms. The design parameters are summarized in Table 1.

The simulation environment consists of a narrow tunnel-like structure with multiple turns, representing confined inspection scenarios such as building ducts and underground passages. The tunnel has a square cross-section of $0.5\ \mathrm{m} \times 0.5\ \mathrm{m}$ and a total length of approximately $11\ \mathrm{m}$. The UAV starts from an initial location outside the tunnel and is required to navigate through the corridor toward the goal while avoiding collisions with the surrounding walls.

3.1 Velocity Profile Analysis in Horizontal Navigation

Figure 1 compares the trajectories generated by the evaluated motion planners, including Rapidly-exploring Random Trees (RRT), local grid-based A* reached the goal and a lightweight bug MP from [17]. The lightweight reactive planners maintained a smooth and centered trajectory through the corridor and successfully negotiated the turns using only local sensing information. This behavior is desirable in narrow tunnel environments, where maintaining clearance from the walls is critical for safe flight.

Table 2 summarizes the quantitative performance of the planners. The lightweight reactive planner achieved the shortest completion time and the lowest computational cost while maintaining a competitive path length. In contrast, RRT and local grid-based A* required substantially longer computation times due to repeated local replanning. Standard RRT and A* implementations generally assume access to an occupancy grid or geometric map; however, such information is difficult to obtain on nano-UAVs equipped only with sparse range sensors. Therefore, in this study, RRT and A* were

Table 2: Performance comparison of motion planners in MATLAB simulation

Planner	Time Completed	Path Length	Clearance	Ave. Speed	Comp. Time (μs)	Strength/Weakness
SAMP	90 s	12.5 m	0.10 m	0.1384 m/s	63.6	Minimal sensing, Fast
Bug	88 s	12.5 m	0.05 m	0.1429 m/s	72.2	Fast but low clearance
RRT	179 s	13.3 m	0.08 m	0.0744 m/s	217.0	Slow and inefficient
A*	133 s	16.6 m	0.08 m	0.1250 m/s	47.3	Poor at corners

implemented in a receding-horizon manner, where candidate waypoints were sampled within a locally inferred free-space corridor derived from range measurements. Instead of explicitly constructing a map, tunnel boundaries were imposed as feasibility constraints. Although this modification allowed both planners to progress toward the goal, their computational burden remained significantly higher than that of the lightweight reactive planner.

Figure 1 also shows the forward (v_x) and lateral (v_y) velocity profiles for representative planners. These profiles provide insight into the control behavior and motion characteristics of each method under confined navigation. These observations are worthy to note:

- **SAMP (Proposed)** The proposed SAMP exhibits a smooth and structured velocity profile. The forward velocity v_x remains close to the nominal value for most of the trajectory, with only brief reductions near turns or constrained regions. The lateral velocity v_y shows gradual and continuous variations, corresponding to anticipatory steering toward the local reference (carrot point). This behavior reflects the structure-aware nature of the planner: instead of reacting abruptly to obstacles, the UAV adjusts its motion based on short-horizon geometric inference. As a result, SAMP produces smooth transitions through corners and maintains stable forward progress with minimal oscillation.

- **Bug** In contrast, the Bug-based planner exhibits highly oscillatory lateral velocity v_y, characterized by rapid switching between positive and negative values. This behavior is a direct consequence of wall-following logic, where the controller repeatedly corrects its position relative to the wall without anticipating future geometry. The forward velocity v_x also shows intermittent drops, particularly near corners, indicating hesitation and inefficient progress. These oscillations lead to increased path inefficiency and reduced stability, especially in narrow passages where frequent corrections are required.

- **RRT** The RRT-based planner produces irregular and discontinuous velocity profiles. Both v_x and v_y exhibit sudden changes and spikes, reflecting the discrete and replanning-based nature of the algorithm. As the planner generates new waypoint sequences, the control commands change abruptly, resulting in non-smooth motion. Although RRT is able to reach the goal, the velocity profile indicates poor suitability for real-time control on nano-UAVs, as the lack of continuity can introduce instability and increased control effort.

- **Local Grid A*** Similarly, the local grid-based A* planner shows high-frequency fluctuations in both v_x and v_y, particularly during replanning phases. Large bursts in lateral velocity are observed when the planner updates its path, causing abrupt directional changes. These discontinuities arise from the grid-based representation and frequent local replanning, which do not explicitly enforce motion smoothness. While the planner successfully reaches the goal, the resulting motion is less stable and more computationally demanding compared to lightweight reactive approaches.

Overall, the velocity profiles highlight a key advantage of the proposed SAMP: it achieves smooth, continuous control behavior while maintaining reactivity and low computational complexity. In contrast, Bug suffers from oscillatory corrections due to its purely local nature, while RRT and A* introduce discontinuities due to replanning. These results demonstrate that incorporating short-horizon structural awareness enables more stable and efficient motion in confined environments under minimal sensing.

3.2 Velocity Profile Analysis in Mixed Tunnel Navigation

The velocity profiles in Fig. 2 further highlight the difference in motion behavior between SAMP and the Bug-based planner, particularly in the presence of vertical transitions.

The proposed SAMP exhibits smooth and coordinated velocity profiles across all three axes. The forward velocity v_x remains close to the nominal value, with only moderate reductions near turns and transition regions. Hence, the lateral velocity v_y varies gradually, reflecting continuous steering toward the wall-extended local reference. Importantly, the vertical velocity v_z shows structured and intentional activation only during vertical transitions. The transitions between horizontal and vertical motion are smooth and temporally consistent, indicating that the planner can anticipate and commit to vertical motion based on local geometric cues. This results in stable traversal of vertical segments without oscillatory switching.

In contrast, the Bug-based planner exhibits highly discontinuous velocity profiles in all axes. The lateral velocity v_y shows rapid switching behavior, corresponding to re-

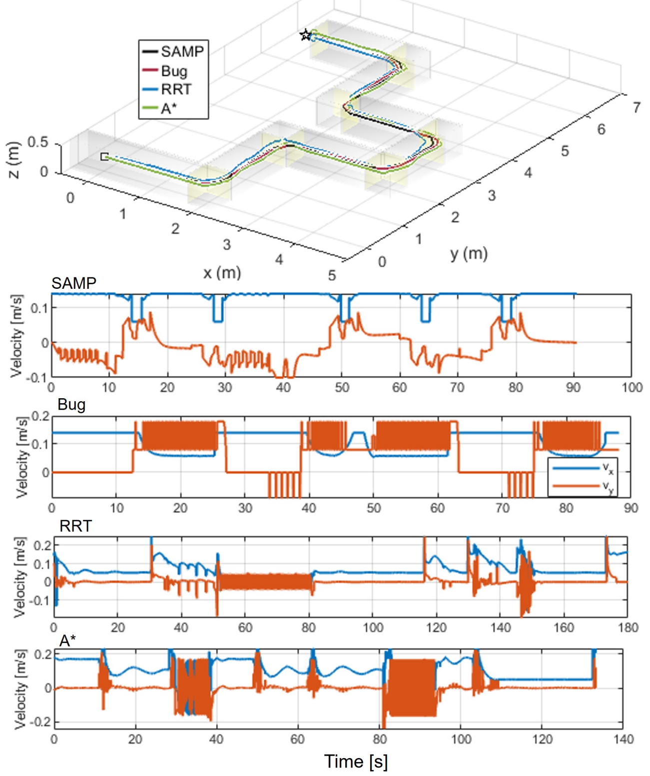

Figure 1: Comparison of motion planners in the MATLAB tunnel simulation environment. The proposed structure-aware motion planning (SAMP) and the Bug-based planner are highlighted as lightweight methods suitable for nano-UAV deployment. SAMP achieves a smoother and more centered trajectory through the corridor, while the Bug planner exhibits oscillatory behavior near turns due to its purely reactive nature.

peated wall-following corrections. This oscillatory behavior becomes more pronounced near corners and junctions. The vertical velocity v_z exhibits frequent sign changes and intermittent activation, indicating instability in selecting and maintaining vertical motion. Rather than committing to a clear ascent or descent, the planner repeatedly switches between directions, leading to inefficient and prolonged traversal. The forward velocity v_x also shows frequent interruptions, reflecting hesitation and reduced progress due to the lack of anticipatory behavior.

These results demonstrate that purely reactive strategies such as Bug are insufficient for mixed horizontal–vertical tunnel navigation, where consistent mode commitment is required. In contrast, SAMP leverages short-horizon structural awareness to generate smooth, coordinated motion across all axes. The resulting velocity profiles are significantly more stable, enabling reliable traversal of complex confined environments under minimal sensing.

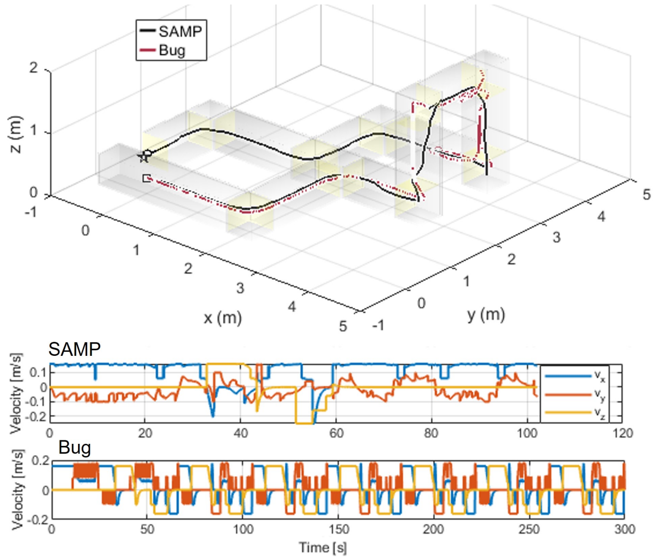

Figure 2: Comparison between the proposed structure-aware motion planning (SAMP) and a Bug-based reactive planner in a mixed horizontal–vertical tunnel environment. (Top) 3D trajectories from start to goal. SAMP achieves smooth and consistent traversal, including stable vertical transitions, while the Bug planner exhibits oscillatory behavior and inefficient motion, particularly near corners and vertical junctions. (Bottom) Corresponding velocity profiles in the forward (v_x), lateral (v_y), and vertical (v_z) directions. SAMP maintains smooth and continuous velocity variations, whereas the Bug planner shows frequent switching and high-frequency oscillations due to its purely reactive wall-following behavior.

4 EXPERIMENTAL VALIDATION

Real-world flight experiments were conducted using a Crazyflie 2.1 nano-UAV to evaluate the proposed lightweight motion planners under minimal sensing conditions. The platform has a mass of approximately 28 g and a motor-to-motor distance of 0.08 m. The UAV is equipped with onboard IMU and pressure sensors, together with a Multiranger deck and Flow deck for sparse range sensing, optical-flow-based velocity estimation, and altitude stabilization. No global map, motion capture system, 3D LiDAR, or vision-based localization was used during autonomous navigation.

Figure 3 shows the experimental tunnel environments and representative flight trajectories. The navigation task in all experiments was to autonomously traverse the tunnel and reach the end of the course using only onboard sparse sensing. Two tunnel configurations were considered: a horizontal-to-ascending tunnel transition (Environment A, 2.0 m) and a mixed horizontal-vertical tunnel (Environment B, 3.5 m). Figures 3(A.2) and (B.2) compare representative trajectories generated by SAMP and the Bug-based planner. In both environments, SAMP maintained smoother and more centered trajectories, while the Bug-based planner exhibited larger de-

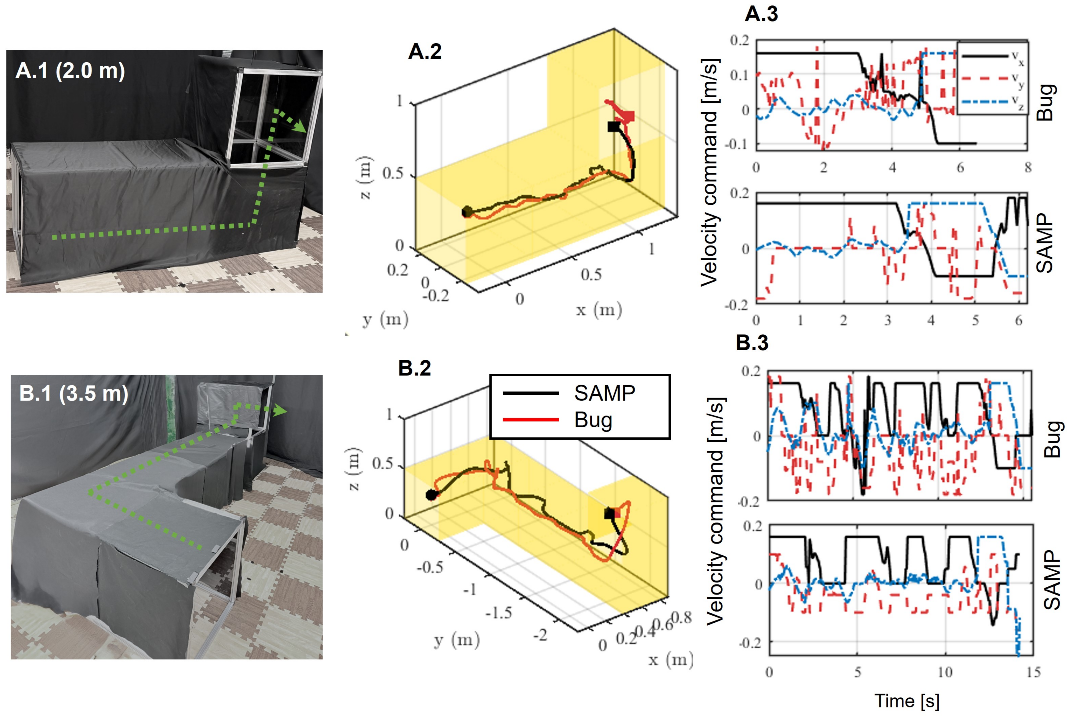

Figure 3: Experimental validation of lightweight motion planners in confined tunnel environments using a Crazyflie nano-UAV equipped with sparse onboard sensing. (A.1) Horizontal-to-ascending tunnel environment. (A.2) Representative experimental trajectories for SAMP and the Bug-based planner in Environment A. (A.3) Corresponding velocity profiles showing forward (v_x), lateral (v_y), and vertical (v_z) motion components. (B.1) Mixed vertical-horizontal tunnel environment. (B.2) Representative experimental trajectories for SAMP and the Bug-based planner in Environment B. (B.3) Corresponding velocity profiles for the evaluated planners. The proposed SAMP maintains smoother trajectories and more continuous velocity behavior, whereas the Bug-based planner exhibits stronger oscillatory corrections due to purely reactive wall-following behavior under sparse sensing.

viations and oscillatory corrections, particularly near tunnel transitions and corners.

The corresponding velocity profiles in Figs. 3(A.3) and (B.3) further highlight the behavioral differences. SAMP produced smoother forward, lateral, and vertical velocity transitions, whereas the Bug-based planner showed frequent oscillations caused by purely reactive wall-following behavior without anticipation of tunnel geometry.

The experimental success rates also demonstrate the advantage of incorporating short-horizon structural awareness. In Environment A, the Bug-based planner achieved a success rate of $5/10$, while SAMP achieved $8/10$. In the more challenging Environment B, Bug achieved only $1/10$ successful runs, whereas SAMP achieved $6/10$. Failures were mainly caused by aerodynamic disturbances inside the tunnels, imperfect range measurements, and local interpretation errors. Some runs resulted in wall collisions, while others caused the UAV to retreat toward previously traversed regions instead of progressing toward the goal.

Overall, the experimental results demonstrate that incorporating lightweight structural awareness significantly improves navigation robustness and motion smoothness for nano-UAV tunnel navigation under sparse sensing conditions.

5 Conclusion

This paper presented a Structure-Aware Motion Planner (SAMP) for nano-UAV navigation in confined tunnel environments using only sparse range sensing. The proposed approach introduces a wall-extended local referencing mechanism that enables anticipatory yet lightweight reactive navigation without requiring mapping, memory, or global optimization. Simulation and real-world experiments demonstrated that SAMP achieves smoother motion, improved stability, and higher traversal success rates compared to conventional reactive baselines. Overall, this work demonstrates the feasibility of autonomous tunnel navigation for resource-constrained nano-UAVs using minimal onboard sensing and highlights the potential of lightweight structure-aware planning for confined and GPS-denied environments.

ACKNOWLEDGEMENTS

This work was supported by the New Energy and Industrial Technology Development Organization (NEDO) under Project JPNP22002.

REFERENCES

[1] Jan Quenzel, Matthias Nieuwenhuisen, David Droeschel, Marius Beul, Sebastian Houben, and Sven Behnke. Autonomous mav-based indoor chimney inspection with 3d laser localization and textured surface reconstruction. *Journal of Intelligent & Robotic Systems*, 93(1):317–335, 2019.

[2] Chee How Tan, M Ng, Danial Sufiyan Bin Shaiful, Shane Kyi Hla Win, WJ Ang, Sai Kit Yeung, HB Lim, Minhn N Do, and S Foong. A smart unmanned aerial vehicle (uav) based imaging system for inspection of deep hazardous tunnels. *Water Practice & Technology*, 13(4):991–1000, 2018.

[3] B. He, X. Ji, G. Li, and B. Cheng. Key technologies and applications of uavs in underground space: A review. *IEEE Transactions on Cognitive Communications and Networking*, 10(3):1026–1049, 2024.

[4] R. Zhang, G. Hao, K. Zhang, and Z. Li. Unmanned aerial vehicle navigation in underground structure inspection: A review. *Geological Journal*, 58(6):2454–2472, 2023.

[5] Xuesu Xiao and Robin Murphy. A review on snake robot testbeds in granular and restricted maneuverability spaces. *Robotics and Autonomous Systems*, 110:160–172, 2018.

[6] Sangchul Han, Sanguk Chon, JungYeong Kim, Jeahong Seo, Dong Gwan Shin, Sangshin Park, Jin Tak Kim, Jinhyeon Kim, Maolin Jin, and Jungsan Cho. Snake robot gripper module for search and rescue in narrow spaces. *IEEE Robotics and Automation Letters*, 7(2):1667–1673, 2022.

[7] Peter Wharton, Tsam Lung You, George P Jenkinson, Richard Suphapol Diteesawat, Nguyen Hao Le, Edith-Clare Hall, Martin Garrad, Andrew T Conn, and Jonathan Rossiter. Tetraflex: A multigait soft robot for object transportation in confined environments. *IEEE Robotics and Automation Letters*, 8(8):5007–5014, 2023.

[8] Darius J Carter, Lauren Bouchard, and Daniel B Quinn. Influence of the ground, ceiling, and sidewall on micro-quadrotors. *AIAA Journal*, 59(4):1398–1405, 2021.

[9] Pedro Sanchez-Cuevas, Guillermo Heredia, and Anibal Ollero. Characterization of the aerodynamic ground effect and its influence in multirotor control. *International Journal of Aerospace Engineering*, 2017(1):1823056, 2017.

[10] Matěj Petrlík, Tomáš Báča, Daniel Heřt, Matouš Vrba, Tomáš Krajník, and Martin Saska. A robust uav system for operations in a constrained environment. *IEEE Robotics and Automation Letters*, 5(2):2169–2176, 2020.

[11] Mihail Pivtoraiko, Daniel Mellinger, and Vijay Kumar. Incremental micro-uav motion replanning for exploring unknown environments. In *2013 IEEE international conference on robotics and automation*, pages 2452–2458. IEEE, 2013.

[12] Danilo Saccani, Leonardo Cecchin, and Lorenzo Fagiano. Multitrajectory model predictive control for safe uav navigation in an unknown environment. *IEEE Transactions on Control Systems Technology*, 31(5):1982–1997, 2022.

[13] Xinhua Wang, Mingyan Cheng, Shuai Zhang, and Huajun Gong. Multi-uav cooperative obstacle avoidance of 3d vector field histogram plus and dynamic window approach. *Drones*, 7(8):504, 2023.

[14] Shengjie Fu, Chang Lin, Zhaoyuan Yao, Tianliang Lin, and Qihuai Chen. Obstacle-avoidance trajectory planning method for excavators in confined spaces based on improved rrt* algorithm. *Scientific Reports*, 2026.

[15] Mehak Raibail and Abdul Hadi Abd Rahman. Path planning of multi-robots in confined spaces. In *International Conference on Robot Intelligence Technology and Applications*, pages 200–208. Springer, 2023.

[16] Hui Ye, Junjie Cao, Xiaofei Yang, and Shuyi Shao. Safety-critical model predictive control for quadcopter uav subject to wind disturbances and measurement errors in confined environments. *Aerospace Science and Technology*, page 111790, 2026.

[17] Kimberly N. McGuire, Christophe De Wagter, Karl Tuyls, Hilbert J. Kappen, and Guido C. H. E. de Croon. Minimal navigation solution for a swarm of tiny flying robots to explore an unknown environment. *Science Robotics*, 4(35):eaaw9710, 2019.

[18] Kimberly N. McGuire, Guido C. H. E. De Croon, Christophe De Wagter, Karl Tuyls, and Hilbert J. Kappen. Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone. *IEEE Robotics and Automation Letters*, 2(2):1070–1076, 2017.