Appendix to "Trajectory Planning and Tracking of Hybrid Flying-Crawling Quadrotors"

Dongnan Hu, Ruihao Xia, Xin Jin, and Yang Tang, *Fellow, IEEE*

July 11, 2024

1 Appendix to Terrestrial-Aerial Trajectory Planning:

Algorithm 1 Terrestrial-Aerial Path Searching

```
Initialize();
while ¬P.empty() do
    n_c ← P.pop(), C.insert(n_c);
    if ReachGoal(n_c) ∨ AnalyticExpand(n_c) then
        return RetrievePath();
    if n_c.z ≥ z_threshold then
        primitives ← AerialExpand(n_c);
    else
        primitives ← TerrestrialExpand(n_c);
    nodes ← Prune(primitives);
    for n_i in nodes do
        if ¬C.contain(n_i) ∧ CheckFeasible(n_i) then
            g_temp ← n_i.g_air + n_c.g_c + EdgeCost(n_i);
            if ¬P.contain(n_i) then
                P.add(n_i);
            else if g_temp ≥ n_i.g_c then
                continue;
            n_i.parent ← n_c, n_i.g_c ← g_temp;
            n_i.f_c ← n_i.g_c + Heuristic(n_i);
```

The pseudo-code of proposed path-searching is illustrated in Algorithm 1, where P and C denote the open and closed set respectively. The variable g_c refers to the actual cost of the path from the start position to the current node, and g_{air} represents the aerial cost, set to a positive constant for nodes above a certain altitude and zero below, directing the planner towards ground paths. Each node n records attributes such as a primitive, the ending voxel of the primitive, the center position of the ending voxel, g_c, g_{air} and f_c. **AnalyticExpand**() computes an optimal path from the current node to the goal, ending the search in advance if the path is collision-free and dynamically feasible. Primitives are expanded iteratively and pruned (**Prune**()), retaining the least cost f_c. **CheckFeasible**() evaluates the safety and feasibility of remaining primitives, retaining the node with the lowest f_c. This process continues until any primitive reaches the goal or the **AnalyticExpand**() succeeds. **Edgecost**() calculates the cost

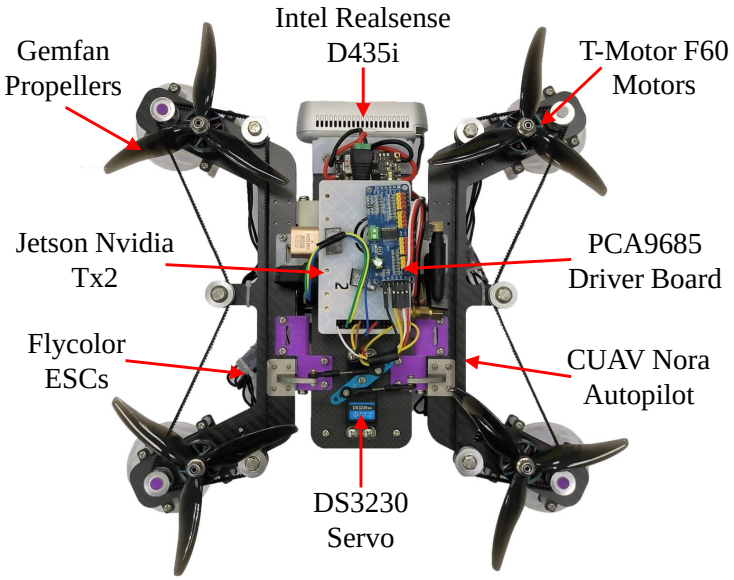

Figure 1: The detailed components of the HyFCQ. In flying mode, the size of the quadrotor is 34 × 32 × 11 cm, while in crawling mode, the size is 34 × 26 × 11 cm. The total weight is 1.6 kg.

Table 1: Description of The Main Components

Components	Part numbers	Key features
Onboard computer	Jetson Nvidia TX2	GPU: 256 CUDA cores
Flight autopilot	CUAV Nora	CPU: STM32H743
Stereo camera	Intel Realsense D435i	Range: 0.3-3 m
Motor	T-Motor F60	2550KV
ESC	Flycolor 50A	MCU: STM32G071
Propeller	Gemfan 51477	Diameter: 5 inch
Servo	DS3230	Torque: 320 $N \cdot cm$
Driver Board	PCA9685	Resolution: 12-bit
Battery	GS33004S30	Weight: 235 g

of a motion primitive during the expansion process. **Heuristic**() computes the cost of feasible trajectories from the current node to the goal to accelerate the search process.

2 Appendix to HyFCQ Platform:

The HyFCQ platform incorporates perception, computation, and actuation modules, as shown in Figure 1. The perception module includes a stereo camera for acquiring depth images. The computation module includes an onboard computer for processing tasks such as localization, mapping, trajectory planning, and tracking. The actuation module consists of an autopilot, electronic speed controllers (ESCs), motors, servo, and servo driver board. Detailed descriptions of the components are provided in Table 1.

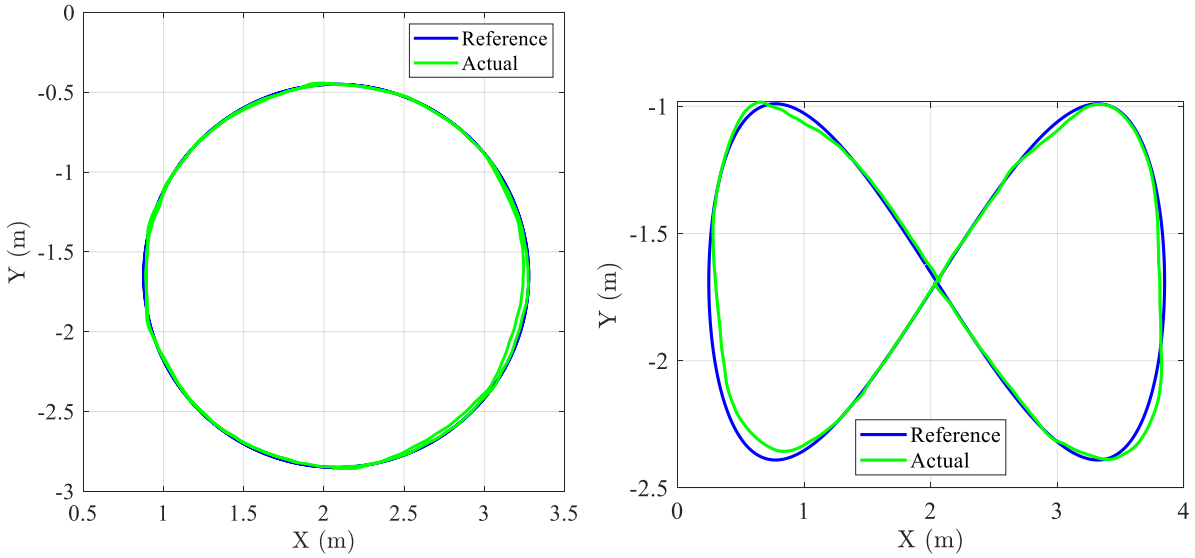

Figure 2: Left) The circular trajectory tracking. Right) The lemniscate trajectory tracking.

Table 2: The Results Of Terrestrial Trajectory Tracking

Trajectory	$E_{ap}(m)$	$E_{mp}(m)$	$E_{ay}(rad)$	$E_{my}(rad)$
Circular	0.046	0.047	0.049	0.239
Lemniscate	0.048	0.064	0.123	0.550

3 Appendix to Terrestrial Trajectory Tracking Experiment:

To validate the effectiveness of the terrestrial trajectory tracking controller, we conduct tests using circular and lemniscate trajectories. The evaluation criteria for trajectory tracking performance include the average position error E_{ap}, the maximum position error E_{mp}, the average yaw angle error E_{ay} and the maximum yaw angle error E_{my}. The circular has a radius of 1.2 meters, while the lemniscate has a longitudinal length of 3.6 meters and a lateral width of 1.4 meters. Both trajectories can be represented as functions of the phase angle. With a fixed angular velocity, the linear velocity of the setpoint on the trajectory is uniquely determined. The angular velocity set for the circular tracking is 0.66 rad/s, corresponding to a linear velocity of 0.8 m/s. The angular velocity set for the lemniscate tracking is 0.78 rad/s, corresponding to a maximum linear velocity of 1.0 m/s.

We conduct 10 sets of experiments and performed statistical analysis on the data. The tracking errors and plots are shown in **Table** 2 and Figure 2 respectively. It can be seen that the proposed controller achieves stable tracking of the terrestrial trajectories effectively.

4 Appendix to Comparison of Terrestrial Trajectory Planning Experiment:

The simulation experiment is implemented by the Robot Operating System (ROS) on a laptop with AMD R7-4800H CPU, 16G RAM, and RTX2060 GPU. The operation system is Ubuntu18.04 LTS.

To verify that our trajectory meets the HyFCQ crawling motion requirements in the real world, a comparison of planning performance is made between our method and Zhang's un-

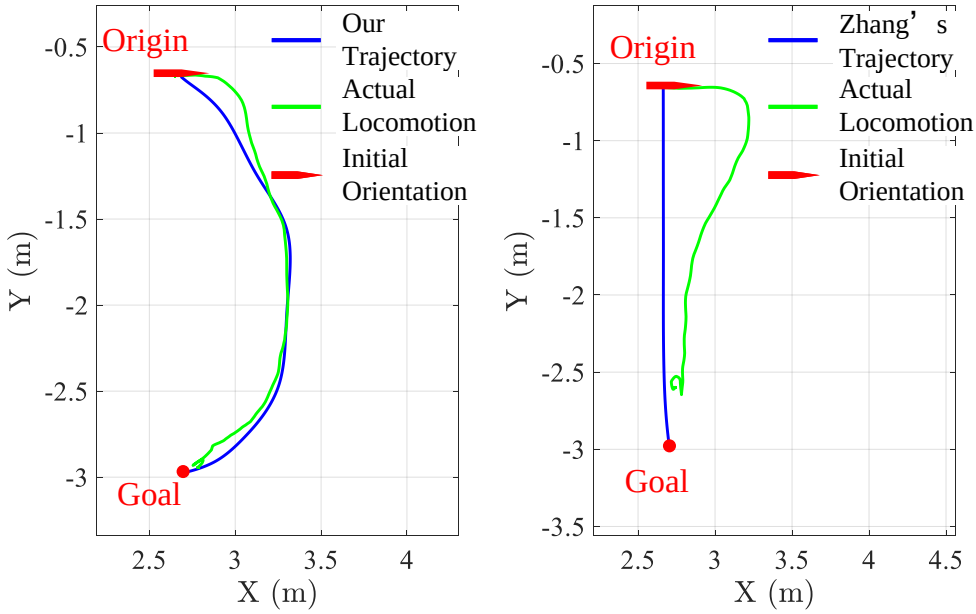

Figure 3: The Left) and Right) figures respectively represent our and Zhang's **c11** trajectory planning in real-world experiments. Based on the tracking performance, our trajectory closely aligns with the crawling motion of HyFCQs in nonholonomic conditions.

Table 3: Position Error Comparision

Method	$v_m(m/s)$	$a_m(m/s^2)$	$E_{ap}(m)$	$E_{mp}(m)$
Proposed	1.0	0.8	**0.077**	**0.156**
Zhang's			0.347	0.595
Proposed	1.2	1.0	**0.080**	**0.174**
Zhang's			0.354	0.598

der different velocity and acceleration limitations in nonholonomic conditions. The terrestrial tracking controller designed by this paper is utilized for trajectory tracking with both planning methods. A lateral goal is set, and trajectories along with the locomotion path of the quadrotor are recorded. One experiment performance is shown in Figure 3, in which the maximum velocity and acceleration of the trajectories are limited to 1.0 m/s and 0.8 m/s^2 respectively. An evident observation from this figure is that our trajectory, in contrast to Zhang's, restricts changes in yaw angle, thus avoiding generating excessive steering motion over short distances. Table 3 presents the average position error E_{ap} and maximum position error E_{mp} of the experiments. Where v_m and a_m represent maximum velocity and acceleration respectively. Zhang's E_{ap} is approximately 4.4 times greater than ours, while the E_{mp} is about 3.4 times greater than ours. These results suggest that Zhang's trajectories are kinodynamic infeasible for HyFCQs, whereas ours are feasible.

5 Appendix to Hybrid Terrestrial-Aerial Navigation Experiment:

Figure 4 presents the trajectory tracking results of the two hybrid terrestrial-aerial navigation experiments, comparing the actual and reference trajectories in the X, Y, and Z dimensions.

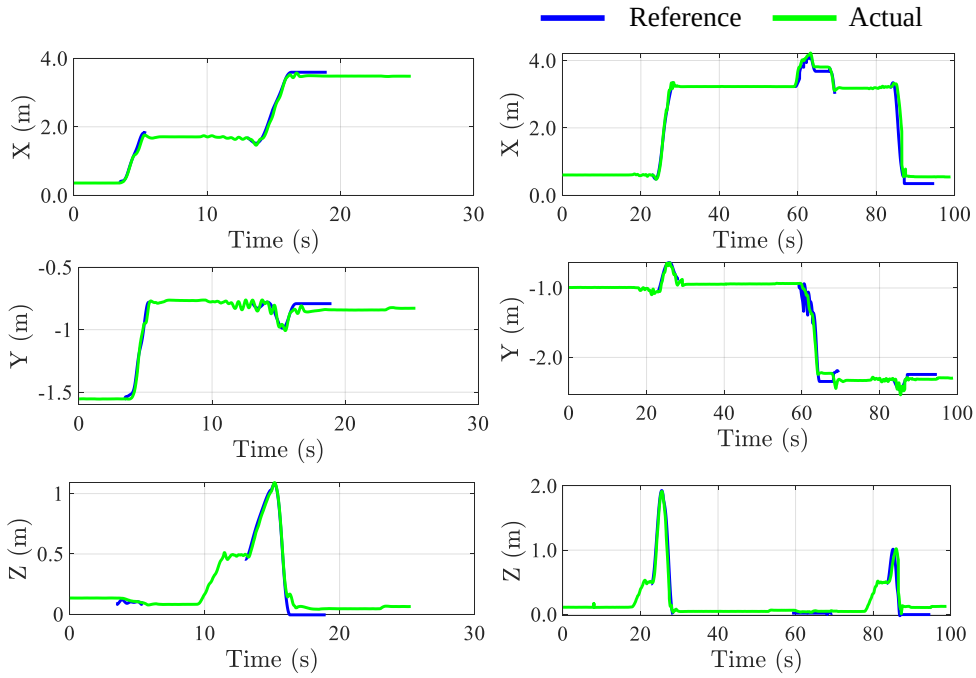

Figure 4: Actual and reference trajectories in the X, Y, and Z dimensions of the two navigation experiments. The Left) figure displays results from the first experiment, while the Right) figure shows results from the second experiment. The interruptions in the reference trajectory are due to the drone undergoing transformations, taking off, or landing, phases during which the trajectory planner does not engage in navigation tasks.

The average tracking errors for the first and second experiments are 0.102 m and 0.111 m, respectively. The outcomes of these experiments validate the feasibility of the proposed trajectory planning and tracking method.