Test of Obstacle Avoidance Capabilities on an Autonomous Nano Quadcopter

Tagir Muslimov
Ufa State Aviation Technical University
Ufa, Russia
tagir.muslimov@gmail.com

Evgeny Kozlov
Ufa State Aviation Technical University
Ufa, Russia
geny.kozlov@mail.ru

Mark Akbashev
Ufa State Aviation Technical University
Ufa, Russia
markakbashev@yandex.ru

***Abstract*—This paper presents collision avoidance implementation on a Crazyflie 2.1 nano quadcopter featuring modular architecture. Autonomous navigation was implemented using special decks. One of those decks can accurately measure distance to surrounding objects in four directions. The second deck can measure the distance to the surface below the quad and pick up the coordinates onboard. A simple algorithm was developed to avoid collision with single obstacles; the algorithm has since been tested on an actual nano quadcopter. The paper presents the results of that test. According to the results, the tested equipment is suitable for indoor collision avoidance implementation on autonomous quadcopters.**

***Index Terms*—obstacle avoidance, autonomous navigation, UAV control**

I. INTRODUCTION

A. Motivation

Numerous papers cover collision avoidance implementation for unmanned aerial vehicles (UAVs). Several factors such as drone type, environment, and onboard equipment influence the way the problem could be solved. The problem is even more relevant in case of small and very small UAVs that have limited onboard capacity. Many papers cover planning the indoor flight path of such small quadcopters [1]. However, some tasks require use of quadcopters in flocks, e.g., rescue operations [2]. When grouped together, several UAVs have to avoid collisions with each other and with other obstacles. Thus, UAVs have to move in a group while also avoiding collisions [3]. In this case, preferred are the so-called reactive methods where the aircraft generates its flyby trajectory using environmental data and sensor readings. Collision avoidance algorithms for fixed-wing UAVs [4] are drastically different from such algorithms for rotary-wing UAVs [5]. This is due to the movement dynamics that creates benefits of better maneuverability.

B. Related Works

Paper [6] shows using decision trees for collision avoidance implementation; paper [7] covers proportional integral differential (PID) controllers that control quadcopter trajectory in obstacle-aware flight. A stereo cam enabled a micro quadcopter to avoid collisions in [8]. Paper [9] successfully implemented an artificial potential field method.

Thus, overview of related works draws a conclusion that they mainly focus on global localization systems. Armed with the global knowledge of the environment, a quadcopter can retrieve obstacle coordinates and geometry readings for much easier navigation. However, implementing collision avoidance on a limited-sensor quadcopter with no global knowledge of obstacle localization could be interesting. This is the subject matter of this work. We implemented a simple navigation algorithm for an autonomous nano quadcopter flying in an environment with obstacles while lacking access to a global navigation system. The goal hereof was to find whether such an algorithm was implementable.

II. COLLISION AVOIDANCE IMPLEMENTATION

A. Equipment Used

Crazyflie 2.1 is an open-source nano quadcopter used by researchers worldwide. It has a modular architecture, allowing use of various navigation systems. Thus, for global navigation it can use the Lighthouse positioning system. This system accurately calculates the coordinates onboard the aircraft. However, in this research we intentionally limited ourselves to the Multiranger rangefinder deck. According to the official documentation [10], this expansion deck can measure the distance in five directions around the quadcopter; the margin of error is a few mm at distances of up to 4 m. We also used Flow deck v2 that measures the distance to surface below with the same sensor as Multiranger. Besides, Flow deck v2 is equipped with an optical stream sensor that implements an extended Kalman filter to pick the quadcopter coordinates in flight. Such measurements tend to drift; however, they are accurate enough for a flyby maneuver over a short distance. Fig. 1 shows the general view with the Multiranger deck installed. Fig. 2 shows the general view with the Flow deck v2 installed.

The device can be controlled from a PC via a dedicated radio channel. The control algorithm can be programmed with a special Python API. Flight data are received over radio as well.

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (Agreement No. 075-15-2021-1016).

TABLE I
PARAMETERS

Parameter	Symbol	Value
Safety radius, [m]	ρ	0.4
Additional distance, [m]	ΔD	0.2
Collision avoidance speed, [m/s]	v_{avoid}	0.1

B. Simple Collision Avoidance Algorithm

The core idea behind the algorithm is as follows. The quadcopter is tasked to fly straight to the target and then land. It encounters an obstacle on the way and has to avoid it. Mission is considered completed successfully if the unit can take off at Point A, fly, and land once the estimated coordinates exceed the preset value. The unit must not collide with an obstacle in this process. For simplicity, we only tested a rectangular obstacle. Multiranger was used to detect the copter's entrance in the zone defined by the safety radius ρ around the obstacle. The drone would then start to move right until losing the beam; thus, it would detect the right boundary of the obstacle. Then it would need to fly a distance sufficient for collision avoidance; we set it at 0.2 m. A certain flight time would need to be determined accordingly. After beam loss, the quad would fly over this determined time to ensure having a safe path for further flight forward. When moving to the right of the obstacle, the drone would measure the distance to the left in order to pick the beam loss moment. Then it would move forward additionally and then go back to the initial rectilinear flight trajectory.

The algorithm is presented below as pseudocode. In this algorithm, D_{front} is the Multiranger-measured distance to the obstacle in front of the copter; D_{left} is the Multiranger-measured distance to the obstacle to the left of the copter. Table 1 shows the parameters of the experiment. These parameters were found by trial and error over a series of experiments. The algorithm was implemented using a high-level Python API with the open-source code provided by the Crazyflie developers. Then the setpoints were transmitted to the autopilot over a radio channel; the autopilot would process the setpoints using low-level velocity control. For a more detailed description of control implementation, refer to the official documentation [10].

Algorithm: OBSTACLE AVOIDANCE

```
Input: safety radius $\rho$, mission, drone position
       estimation $x.Estimate, y.Estimate$
Output: mission completion
initiate forward movement
while $D_{front} < \rho$ do
    initiate rightward movement at the set speed $v_{avoid}$
    if there was a loss of beam then
        move to the right by the additional distance $\Delta D$
proceed to move forward at the set speed $v_{avoid}$
while $D_{left} < \rho$ do
    continue moving at the set speed $v_{avoid}$ and keep a safe distance
    if there was a loss of beam then
        move forward by the additional distance $\Delta D$
go back to the initial path
complete the mission
return flight data
```

Fig. 1. Crazyflie 2.1 Nano quadcopter with Multiranger distance measurement deck

III. Results and Discussion

Fig. 3 shows the obstacle and the UAV in flight. In the tests, the drone was able to bypass the obstacle and land after returning to the initial flight path. The drone takes off, bypasses obstacles, and lands. Fig. 4 shows applying Kalman filters to estimate quadcopter coordinates in meters. Here, $stateEstimate.x$ is the X-axis coordinate, $stateEstimate.y$ is the Y-axis coordinate, $stateEstimate.z$ is the Z-axis coordinate, and time is measured in milliseconds. Fig. 5 is a three-dimensional (3D) representation of these estimates. Fig. 6 shows quadcopter-measured distance to the nearest objects in three directions (in millimeters). Here, $range.front$ is the distance to the nearest object in front, $range.left$ is the distance to the left, $range.zrange$ is the distance to the surface below the aircraft. Fig. 7 is a 3D representation of these estimates. Fig. 8 shows Euler angles as measured by the quadcopter in flight. Here, $stabilizer.roll$ is the roll angle, $stabilizer.pitch$ is the pitch angle, $stabilizer.yaw$ is the yaw angle; time is measured in milliseconds.

The graphs show that the coordinate estimates have poor accuracy and tend to drift. Notably, the Multiranger deck can measure distance to objects up to 4 m away; graphs in Fig. 6

Fig. 2. Crazyflie 2.1 Nano quadcopter with Optical Navigation Flowdeck V2

confirm that. Front-sensor beam loss and the later left-sensor beam loss can be accurately pinpointed to specific moments of time in the graph.

Tests have shown that more complex collision avoidance methods such as the potential field method can be difficult to implement when using an autonomous navigation system. This is why research into such methods mostly relies on a global navigation system that accurately locates the quadcopter in relation to the endpoint, which is also known as point of attraction. Autonomous navigation systems such as the optical system that comes with Flow deck V2 tend to drift over time. This compromises coordinate accuracy in long-distance flight. Thus, for autonomous quadcopter control, we see a greater potential in combining simple rule-based algorithms with more complex solutions such as the potential field method. For better autonomous navigation, quadcopters should also be capable to refining their coordinate estimates in long-distance flight.

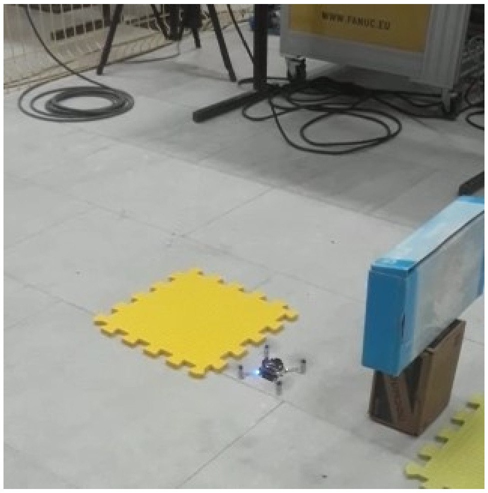

Fig. 3. Quadcopter flight with collision avoidance

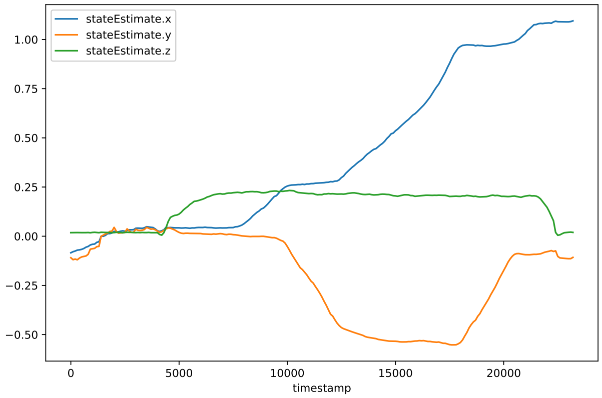

Fig. 4. Quadcopter's in-flight coordinate estimates

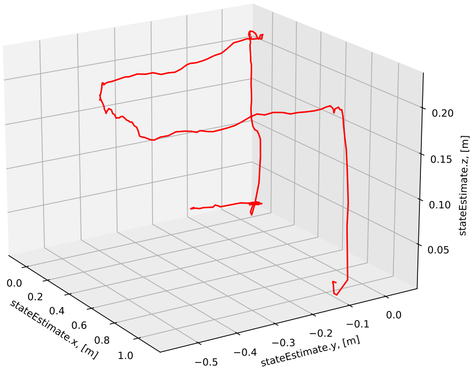

Fig. 5. Quadcopter's in-flight coordinate estimates in 3D

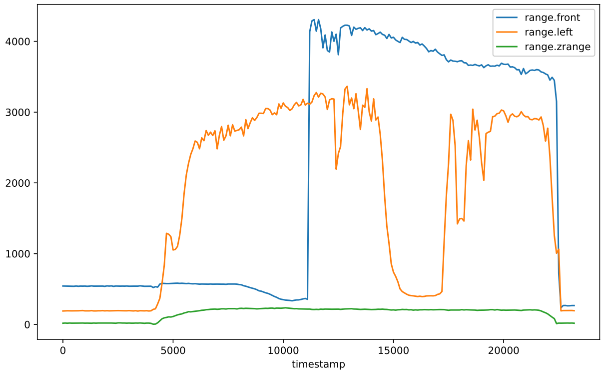

Fig. 6. Multiranger deck-measured distances to surrounding objects

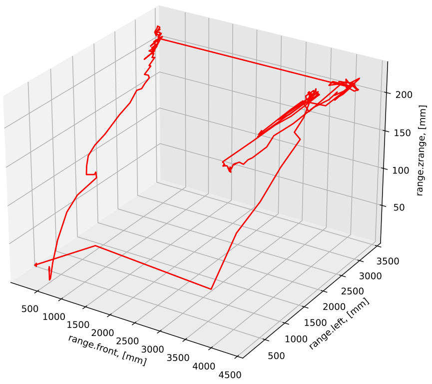

Fig. 7. Multiranger deck-measured distances to surrounding objects in 3D

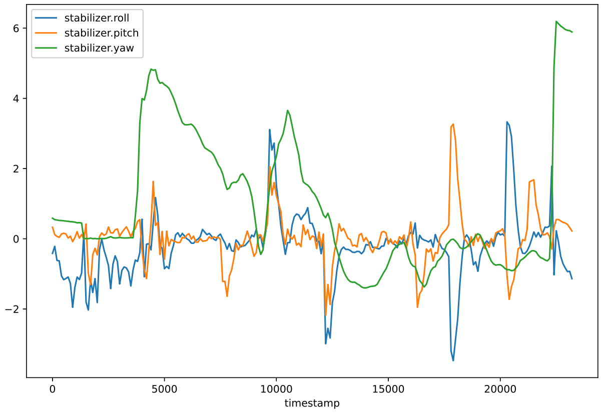

Fig. 8. Quadcopter's in-flight Euler angle estimates

IV. Conclusions

The paper demonstrates a successfully tested simple collision avoidance algorithm for an autonomous nano quadcopter. Flight tests were conducted, and the results showed that the quadcopter accomplished the intended mission without collisions. Further research could focus on more complex obstacle avoidance algorithms, such as the artificial potential field method.

References

[1] V. M. Respall, D. Devitt, R. Fedorenko, and A. Klimchik, "Fast Sampling-based Next-Best-View Exploration Algorithm for a MAV," Proc. - IEEE Int. Conf. Robot. Autom., vol. 2021–May, pp. 89–95, 2021.

[2] R. V. Meshcheryakov et al., "An application of swarm of quadcopters for searching operations," IFAC-PapersOnLine, vol. 52, no. 25, pp. 14–18, Jan. 2019.

[3] V. Izhboldina and I. Lebedev, "Group movement of UAVs in environment with dynamic obstacles: a survey," Int. J. Intell. Unmanned Syst., Mar. 2022.

[4] V. D. Berdonosov, A. A. Zivotova, D. O. Zhuravlev, and Z. H. Naing, "Implementation of the Speed Approach for UAV Collision Avoidance in Dynamic Environment," in 2018 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon), 2018, pp. 1–6.

[5] C. Berger, P. Rudol, M. Wzorek, and A. Kleiner, "Evaluation of reactive obstacle avoidance algorithms for a quadcopter," in 2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV), 2016, pp. 1–6.

[6] C.-W. Chen, P.-H. Hsieh, and W.-H. Lai, "Application of decision tree on collision avoidance system design and verification for quadcopter," Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci., vol. XLII-2/W6, no. 2W6, pp. 71–75, Aug. 2017.

[7] A. V. Parfiryev, I. N. Ischuk, A. V. Dushkin, and S. D. Smolyakova, "Algorithm for Controlling the Trajectory of an Unmanned Aerial Vehicle with the Possibility of Flying Around Obstacles," in 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), 2020, pp. 2395–2400.

[8] A. U. Haque and A. Nejadpak, "Obstacle Avoidance Using Stereo Camera," May 2017.

[9] R. L. Galvez et al., "Obstacle avoidance algorithm for swarm of quadrotor unmanned aerial vehicle using artificial potential fields," in TENCON 2017 - 2017 IEEE Region 10 Conference, 2017, pp. 2307–2312.

[10] "Bitcraze Official Website." [Online]. Available: https://www.bitcraze.io/.