Autonomous Drone Racing Competition @ FUZZ IEEE 2021

1 Organizers

- **Dr. Andriy Sarabakha** (Munich School of Robotics and Machine Intelligence, Technical University of Munich, Germany), andriy.sarabakha@tum.de
- **Assoc. Prof. Erdal Kayacan** (Department of Engineering, Aarhus University, Denmark), erdal@ece.au.dk

2 Motivation

Drone racing is a recreational sport in which the goal is to pass through a sequence of gates in a minimum amount of time while avoiding collisions with the environment. In autonomous drone racing, one must accomplish this task by flying fully autonomously in an unknown environment relying only on the on-board resources. Thus, autonomous drone racing is an exciting case study that aims to motivate more researchers to develop innovative ways of solving complex problems. What makes drone racing such an interesting challenge is the cumulative complexity of each sub-problem to be solved, such as *perception*, *localisation*, *path planning* and *control*.

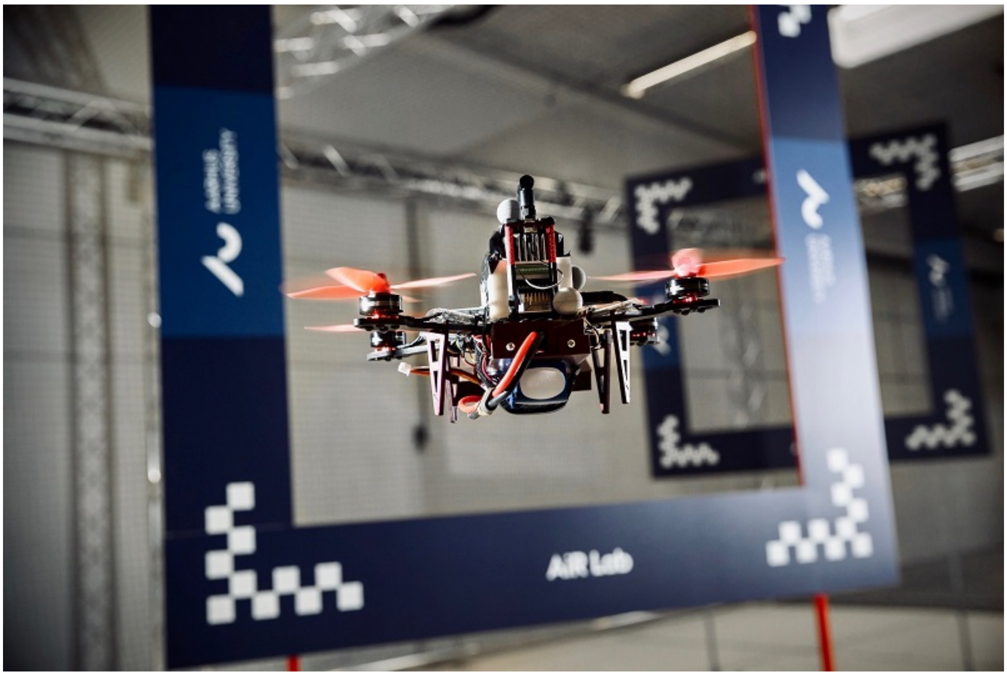

Figure 1: Autonomous drone racing track at Aarhus University.

3 Goal

The FUZZ IEEE 2021 Autonomous Drone Racing Competition is an aerial robotics challenge in which research groups will validate their fuzzy logic controllers in a challenging scenario – *autonomous drone racing.*

4 Challenge

The technical challenge of this competition focuses on the design of various fuzzy logic controllers which can operate on nonlinear regions of the drone's dynamical model at high speeds. The number and pose (position and orientation) of the gates will be predefined, and the drone will have to cross them in a given sequence. For the gate detection, an on-board wide-angle camera with 90° field-of-view is used. Trajectories at fixed velocities through the centres of all gates will be generated using the curve fitting method and available to the participants. However, each team is free to implement its own trajectory generation strategy to achieve better control performances. The real-time noisy localisation, which includes estimated position ($\hat{x}$, $\hat{y}$, and $\hat{z}$) and attitude (roll $\hat{\phi}$, pitch $\hat{\theta}$, yaw $\hat{\psi}$), will be available to simulate visual-inertial odometry which deteriorates its performances at high speeds due to motion blur and accelerometer's noise. The task of each team is to develop a fuzzy logic controller which will take the desired trajectory and current pose of the drone, as input, and will provide the commanded attitude (ϕ^*, θ^*, ψ^*) and thrust T^*, as control commands, as depicted in Fig. 2. For more details, please refer to [1].

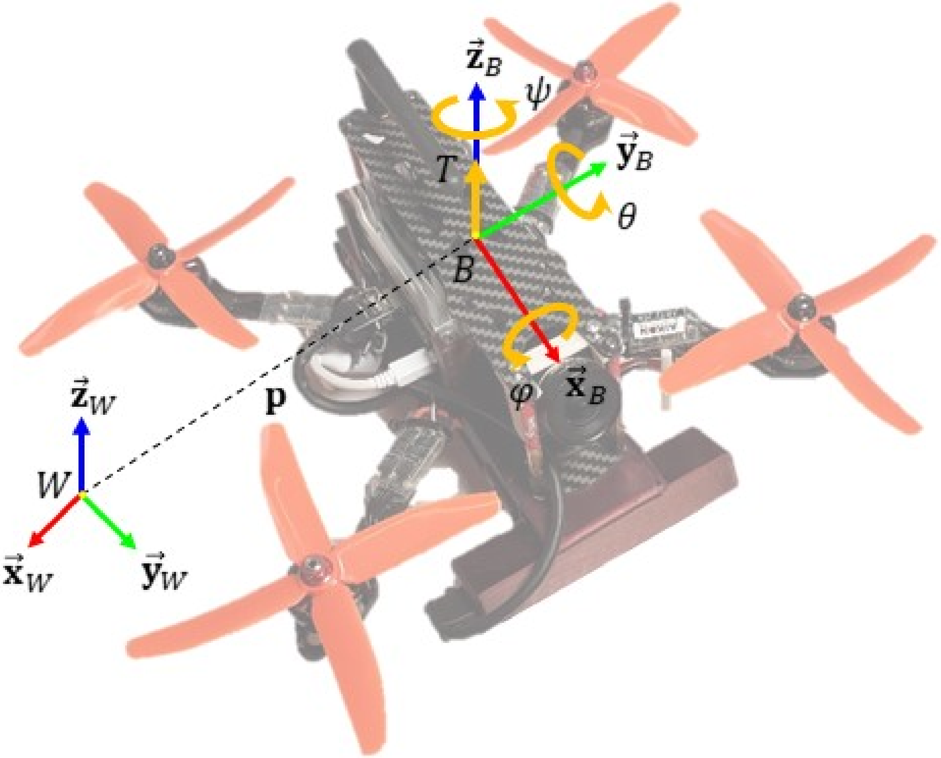

Figure 2: Racing drone with its reference frames and control inputs.

5 Environment

The simulation environment of the racing track with racing gates is implemented in MATLAB and tested in MATLAB R2020b. A sample environment is shown in Fig. 3. The simulation package contains five files:

1. *MAIN.m*: main file for starting the simulation;
2. *trajectory.m*: trajectory generation through the center of each gate (the desired speed over the trajectory can be adjusted by a meta parameter inside the file);
3. *controller.m*: pose controller which **participants will have to implement** (a sample PD controller is provided);
4. *uav.m*: dynamical model of the drone and its low-level controller;
5. *environment.m*: visualisation of the racing arena and performance evaluation (to speed up the visualisation, the animation can be disable by a meta flag inside the file).

Please note that only the content of *controller.m* and *trajectory.m* can be modified.

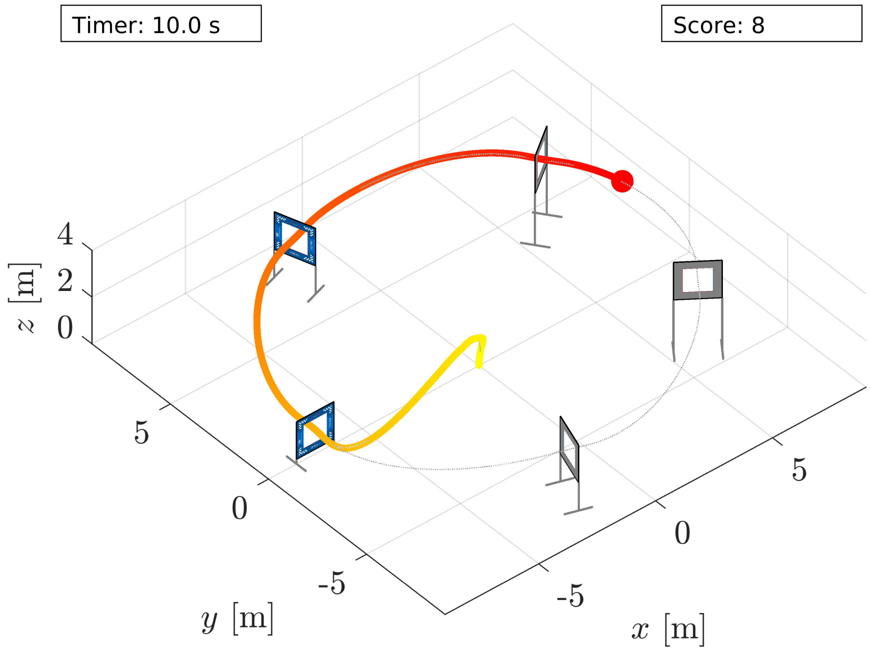

Figure 3: Racing arena simulated in MATLAB.

Remark. *The position and orientation of the gates in the testing arena will be different than in the arena provided to the teams to develop their controllers.*

Remark. *For any technical issues with the simulation environment, please open a new issue on https://github.com/andriyukr/drone_racing_competition/issues or send an email enquiry to the Organizers.*

6 Evaluation

The controllers will be evaluated in a similar MATLAB environment but the position of the gates will be different. The main prerequisite is the real-time implementation of the controller, in other words, **the controller must run minimum at 100**Hz. Each team will be given 60 seconds to cross as many gates as possible in a predefined sequence. The evaluation rules are as follows:

- for each successfully crossed gate: **+1 point**;
- for each skipped gate: **−0.5 points**;
- losing the next gate to cross in the field of view of the drone: **−1 point/sec**;
- after a collision with gate/ground/ceiling/walls: **the competition will end** (the points collected till the crash will be counted);
- surviving till the end of the challenge: **+5 points**.

The Organization Committee will evaluate the developed controllers on the testing racing track. The results will be announced during FUZZ IEEE 2021 conference in July 2021.

7 Prizes

- A student MATLAB license for **the winning team**.
- A small surprise from MathWorks for **the winning team**.
- A webinar organised by MathWorks during FUZZ IEEE 2021 for **all participating teams**.

8 Application

Prospective participants must submit an application via the following form. Successfully registered teams will receive a package containing the simulation environment. The latest version of the simulation environment can be downloaded also from `https://github.com/andriyukr/drone_racing_competition`.

9 Submission

The teams must submit the following MATLAB files:

1. *controller.m*,
2. (optional) *trajectory.m*,

to the Organizers via email by **May 31, 2021**.

Remark. *It is also possible to submit a compressed folder containing support files for the controller and trajectory generator.*

References

[1] Andriy Sarabakha. Learning Control of Unmanned Aerial Vehicles Using Artificial Intelligence-Based Methods. `https://hdl.handle.net/10356/143057`, 2020.