2022 International Conference on Unmanned Aircraft Systems (ICUAS)
Dubrovnik, Croatia. June 21-24, 2022

2022 International Conference on Unmanned Aircraft Systems (ICUAS) | 978-1-6654-0593-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICUAS54217.2022.9836233

The Co4AIR Marathon – A Matlab Simulated Drone Racing Competition

Marius Dragomir
Department of Automation
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
dragomir.ma.marius@student.utcluj.ro

Vicu-Mihalis Maer
Department of Automation
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
vicu.maer@aut.utcluj.ro

Lucian Buşoniu
Department of Automation
Technical University of Cluj-Napoca
Cluj-Napoca, Romania
lucian.busoniu@aut.utcluj.ro

***Abstract*—We describe a UAV competition concept in which a Parrot Mambo drone must race over a sequence of colored markers in minimum time. The competition is implemented in Matlab, using the Simulink Support Package for Parrot Minidrones, and can be organized fully in simulation, although an optional real-drone component is included. Students with either control or computer-science backgrounds are accommodated by providing baseline solution modules for the part outside their expertise. We present the competition design, a baseline solution, and our experience with the first edition, which was held in 2021, including student feedback and lessons learned.**

***Index Terms*—student competition, education, unmanned aerial vehicles, Matlab, Simulink**

I. Introduction

An important topic in the area of robotics is the autonomous control of robot systems. Consider, for example, the problem of farming, where multiple types of robots are used in cooperative teams for different activities. There is a need to plant seeds, to spread different chemical substances over the farmlands, to protect the plants from possible invasive species, to monitor certain parameters, and to harvest the yield. In several of these tasks unmanned aerial drones were employed in the last years, due to their ability to cover large distances quickly, without touching the plants themselves [1]. In the overall context of rapid development of autonomous robotic systems, there is an increasing demand for engineers knowledgeable in several aspects of robotics.

Increased efforts to educate future specialists in robotics would contribute to the solution for this larger workforce demand. One type of activity that is known to work well because it often gathers the interest of possible participants is the organisation of competitions. The participants often maintain their motivation and focus over the course of the competition, which further facilitates learning.

The "Co4AIR Marathon – Drone Racing Competition" is a key intellectual output of the "Co4AIR – Computers, Cognition and Communication" Erasmus+ project (http://co4air.eu/). In this competition, student teams must develop machine vision, path planning, and control algorithms for a Parrot Mambo drone so as to complete in minimum time a race track consisting of a sequence of colored markers placed on the ground. The process of choice was a quadcopter drone as it is an unstable system, providing a challenging control problem to solve. At the same time, drones are very popular so the problem is motivating for the students. The race setting promotes a strong team investment into developing a good solution.

To lower the barrier of entry and reduce student costs, the competition can be run fully in simulation, although student teams can choose to also develop a real-drone solution. A full baseline solution is provided on which teams can improve. This also means that the teams can focus only on some parts of the solution, so we can accommodate e.g. students with expertise only in computer vision, or only in control. The scoring rules take this into account.

A. Existing competitions

Other competitions for autonomous drone racing exist, such as the Drone Racing League - Artificial Intelligence Robotic Racing (AIRR) circuit [2] and the AlphaPilot - Lockheed Martin AI Drone Racing Innovation Challenge [3]. These competitions are held physically, using state-of-the-art hardware. The teams participating here are tasked with developing perception, navigation and control algorithms for a standardized drone. The objective of the competition is to complete the racing track in the minimum time possible.

In 2022, the ICUAS conference proposes an autonomous drone competition [4], in the context of an autonomous fire-fighting aerial drone. The sub-tasks that compose the solution to the problem involve the exploration of the world, the detection of an ongoing fire, and the delivery of a payload to the affected area. The competition is be held in two phases: a simulated phase and a live, physical phase. The simulated phase takes place in a world simulated in Gazebo, while the physical trials are held in a motion capture arena that will be built during the ICUAS 2022. Unlike the ICUAS competition, the Co4Air competition focuses on racing on a track as it is discovered.

This work has been financially supported by the project Co4AIR - Computers, Cognition and Communication in Control: A strategIc paRtnership, funded by the European Commission through the Erasmus+ Key Action 203, Strategic Partnerships for Higher Education, and by the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, SeaClear support project number PN-III-P3-3.6-H2020-2020-0060.

The Microsoft AirSim-based based competition, called ”Game of Drones” [5], took place in 2019 and allowed the participants to solve navigation planning, environment perception, or a combination of both tasks in a simulated environment – similarly to the Co4AIR competition. Game of Drones used a front-mounted camera which enables the perception of the poses of the gates that the drone needs to navigate through, and proposes the idea of 2 drones competing at the same time, thus requiring the avoidance of mid-air collisions. In contrast, the Co4Air competition uses a single drone with a bottom-mounted camera that captures pictures of the markers underneath. The fact that the drone needs to tilt in order to move in a certain direction increases the difficulty of detecting the markers.

An autonomous drone flight challenge organized at the IFAC 2020 World Congress [6] targeted a line following scenario, with a full deployment chain from Matlab code to embedded hardware for two low cost drone platforms including the Parrot Mambo. The competition organized by [7] targets more realistic scenarios, where the variability of the environment is important. In the ELROB competition [8], the drone should be able to recognize generic targets in the wild. A similar contest to the Co4AIR Marathon is [9], which uses a simulation environment in ROS/Gazebo, and the teams could evaluate their algorithms within this realistic environment for a task of navigating through floating frames.

A preliminary form of the Co4AIR drone racing contest was ClujUAV, a real-drone corridor navigation challenge held in the autumn of 2019 in Cluj-Napoca, Romania [10]. The results of this competition informed the design of the Co4AIR Marathon.

The Co4AIR proposes a competition that focuses both on machine vision and on navigation and control. The Co4AIR Marathon can be completely run in simulation, using the Parrot Minidrones Support Simulink package as a starting point. By using a virtual platform, any student with sufficient expertise can participate in the competition, as there is no longer any need for providing a physical drone and a working space where it can be safely used. Holding the contest completely online meant that an international competition could be held without travel costs. Another positive aspect of using a virtual model in Simulink is that control engineering students are usually well acquainted with the Matlab software stack, so they can focus on the new aspects of robotics instead of trying to learn to use a new tool. Another advantage of using the Simulink Support Package for Parrot Minidrones is that simulation-to-real transitions are as close to seamless as possible. The Parrot Mambo minidrone costs on the order of 100EUR, which further makes the Co4AIR competition accessible in the event that a physical, on-site challenge takes place.

Differently from other simulation competitions such as [9], and the simulation phase of real-drone competitions such as [4], we allowed students with expertise in either computer vision, control, or both, by allowing students to work only on some of the solution modules and use the baseline solution for the modules where they do not have expertise.

B. Structure of article

Next, Section II presents the design of the contest, including the evaluation procedure. Section III explains the baseline solution in some technical detail. Section IV outlines our experience with the first edition of the contest, as well as feedback from the competitors. Section V concludes the paper while synthesizing some ways in which the contest can be improved.

Some parts of this article are based on our internal, unpublished competition report [11].

II. Contest design

A. Objectives and concept

The objectives of the Co4AIR Marathon were to:

- Motivate students to design, build, and test a solution for a current practical control problem.
- Challenge the brightest students and allow to compare the various teaching systems through the contest result.
- After the end of the project, promote the contest for usage at other universities.

In the contest, a simulated Parrot Mambo drone races over a track defined by a sequence of square markers placed on the ground, in 3 different colours: red, green, blue (see Figure 1). To disambiguate situations in which multiple markers are present in the image of the drone camera, markers should be followed in the order of their colors: red, green, blue, red, etc. The environment provides the feed of the down-facing camera of the drone, as well as accurate position and attitude signals for the drone; and it allows setting the motor commands. The drone parameters will be fixed, but the sequence of marker positions will be unknown to the teams until the day of the contest.

The teams will have to design (a) a computer vision algorithm to detect the markers and produce waypoints for the controller; (b) a control algorithm that ensures the drone reaches each waypoint in the sequence to within a prespecified tolerance; or both (a) and (b). Teams may choose to do only (a) or only (b), in which case a default implementation of the missing component is supplied by the organizers by using the components from a baseline solution.

This baseline solution is provided to all the teams during the kickoff of the competition. The baseline is able to complete the objective of navigating from the start to the end of the race track in a sub-optimal manner, so that the teams can easily attempt to modify, redesign and improve components of the solution in such a way as to achieve an increased performance on the simulated race track.

The programming environment is Matlab/Simulink. Optionally, teams may also develop and demonstrate a real-life solution for the marker following task, with the true Parrot Mambo. To this end, teams should provide a video recording of the drone running the track, as well as any explanations required. The simulation task should be replicated as closely as possible.

To prevent altering the parameters of the simulation, teams are only allowed to change specific blocks in the Simulink scheme, and at the end these blocks are integrated into the "clean" template. Therefore, changes to the expected inputs and outputs of these blocks or other modifications made to the rest of the project are not carried over. Instructions on these blocks, together with details on the competition and installation steps to get the baseline solution running, are provided to the teams in a guide document; the guide of the first edition is available at http://busoniu.net/files/co4airmarathonguide.pdf. As an appendix to this guide, the default control strategy for the drone is explained from a mathematical point of view.

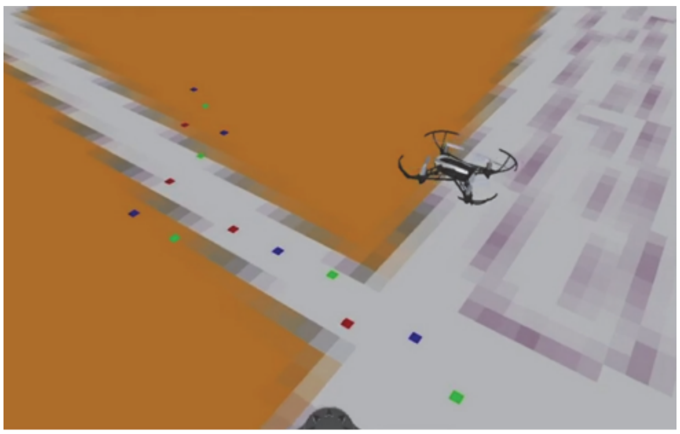

Fig. 1. Competition simulator showcasing the default race track and baseline solution, using the Parrot Minidrones Simulink Support Package.

The solutions would be put on trial on two different race tracks, one that is more linear, favoring solutions that can reach the end marker faster, and one that is more difficult to navigate, involving sharper turns, thus putting an emphasis on the stability and robustness of the solution. The race track is generated in a pseudo-random manner, using an algorithm that places consecutive points towards one direction of the simulated world. The next point in the race track is placed at a randomized distance, at a random angle from the current last marker placed. The randomized distance and angle are constrained to a given range. The distance and angle range were chosen in such a way that the drone would be able to see the next point while hovering. In order to generate the fast track, a smaller angle range was used, while the agility track was made by allowing wider angle variances in consecutive markers, which led to a track with very sharp turns.

B. Team composition and deadlines

Teams of 2 or 3 students (PhD, MSc, or BSc) are accepted. All students in a team should preferably come from the same institution, but student levels can mix. Due to the nature of the task, which mixes control and computer vision, teams with expertise in either systems and control, computer science, or a mix of both, are possible. Registration is done at least 5 weeks in advance of the competition using a registration form that includes team composition, institution, a team name and which among tasks (a), (b), or both the team aims to solve. A draft solution is to be sent 2 weeks in advance of the contest date, which is tested by the organizers for any integration issues, and feedback on correcting these issues is sent to the students. Then, the final solution is due one week in advance.

C. Evaluation

A jury formed by team supervisors and/or external experts evaluates the solutions on the competition day. Each team presents their technical approach in a limited time slot, answers any questions from the jury, and then their solution is demonstrated. It is the organizers who run the simulations, with the method submitted by each team integrated in the clean template. The final score and ranking of the teams in the competition is determined by the following formula:

$$P + (1 - \frac{T}{T_{max}}) + (1 - \frac{M}{M_{tot}}) + B + R \qquad (1)$$

where each variable means the following:

1) P is a score awarded by the judges for the technical approach used by the team and how they presented it before the competition. This activity is included in the competition schedule in order to provide the teams with a way to get to know each other, understand the approach of other teams and learn from their experience.
2) T is the time it took to reach the end of the race track and T_{max} is the maximum admitted value of T. This hard limit is imposed to allow the race trial to be interrupted in case it takes too long to complete. In the case that the drone crashes, then T is also set to T_{max}.
3) M is the number of colored markers that were missed during the trial. The drone needs to fly over the markers within a specified tolerance radius for the marker or be counted as touched. M_{tot} is the total of markers on the race track.
4) B represents bonus points given to the teams that attempted to modify both the computer vision component and the control component
5) R represents bonus points for teams that also attempted to implement and use their algorithm on real drones.

The values of P, R, and B were specified only on the day when the competition was held, so as to avoid a situation where the teams could over-fit solutions to a particular scoring function and race track. After discussions with the jury, organized prior to the competition, we arrived at the following parameters in the score function: $P = 0.5$, $B = 0.5$, $R = 0.25$. The rationale for P and B is that these components should be half of the real-race time and marker components, so that the actual race performance counts more. R was selected smaller because not all the teams had access to – or the financial possibility to acquire – the real drone.

Presentations and actual races were interspersed to maintain audience interest. To make the competition more exciting, it was also decided that the time score T and marker score M would be averaged between two race tracks: one where the markers are nearly along a line, which favors speed, and another where there the track given by the marker sequence often switches direction at various large angles, which favors maneuverability.

III. Baseline solution

A. Drone model

The Parrot Mambo drone was used for the Co4Air competition, a small, light quadrotor that is easy to interface with Simulink: Matlab provides official packages that support data transfer between a computer and the drone, and the ability to program and upload Simulink-based solutions to the physical drone. The drone is shown in Figure 2.

Fig. 2. Mambo parrot drone

In the Simulink environment, the drone model is simulated using two components: a component that solves the rotor dynamics and the effects of the environment on the movement of the drone, and a component that simulates the pose and movement of the quadrotor using a Simulink block that models the 6DOF equations of motion of a fixed mass body. The forces and torques computed by the dynamic component are passed to the 6DOF block which computes the next pose and velocity of the drone. Figure 3 shows the feedback loop that simulates this behavior.

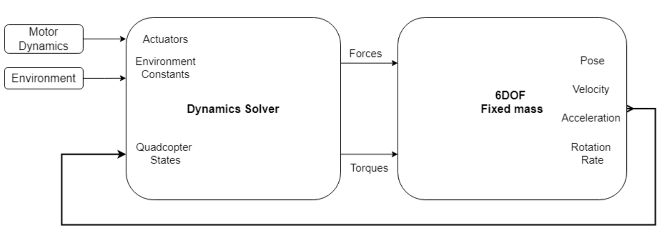

Fig. 3. Schematic of the simulated drone model, as defined in the Simulink Support Package for Parrot Minidrones

The relevant physical parameters of the Mambo Parrot drone are given in Table I.

TABLE I
PHYSICAL DRONE PARAMETERS

Variable	Symbol	Value
Mass	m	0.063 [kg]
Moment of Inertia X	I_x	$5.828 * 10^{-5}$
Moment of Inertia Y	I_y	$7.169*10^{-5}$
Moment of Inertia Z	I_z	$1*10^{-4}$

B. Simulation framework

The framework was built using the demos provided in the Matlab Parrot drone support package. The following list contains the main components of interest:

- Sensors
- flightControlSystem
- Nonlinear Airframe
- Environment
- Visualization

The $Sensors$ component reads the drone model state and the environments, then simulates the sensor signals. The $flightControlSystem$ contains the Simulink blocks that solve the image processing task, the planning task, and the pose control task. The $Airframe$ component contains the nonlinear model of the quadrotor (see again Figure 3), and the $Environment$ stores the constants used to define the simulated environment. The aspects of the environment that are modeled are the gravity field, the magnetic field, and the atmosphere. Finally, the $Visualization$ block displays the drone and the environment as the drone navigates in the simulation. The top-level feedback loop schematic is presented in Figure 4.

If the participant wishes to implement the controller designed in the simulation on the real drone, then he should upload the $flightControlSystem$ into the drone on board memory with minimal interface changes. This functionality is supported by the Matlab drone support package.

Both the drone model and the overall framework already exist in the Parrot support package. Next, we present the baseline solution that we developed for the Co4AIR competition.

C. Initial solution outline

The baseline solution provided to the teams consists of two main components: an algorithm that handles the perception of the markers and the path planning, and an algorithm that controls the movement of the drone [12]. The structure of the system is shown in Figure 5.

The first mentioned component is made of two subsystems, one for the detection and placement of the markers in the world coordinate reference frame, and one for the generation of a reference signal for the flight-control component.

The machine vision algorithm uses the data from a camera mounted under the drone. Areas where the known colours of the markers are detected in the image through the use of a

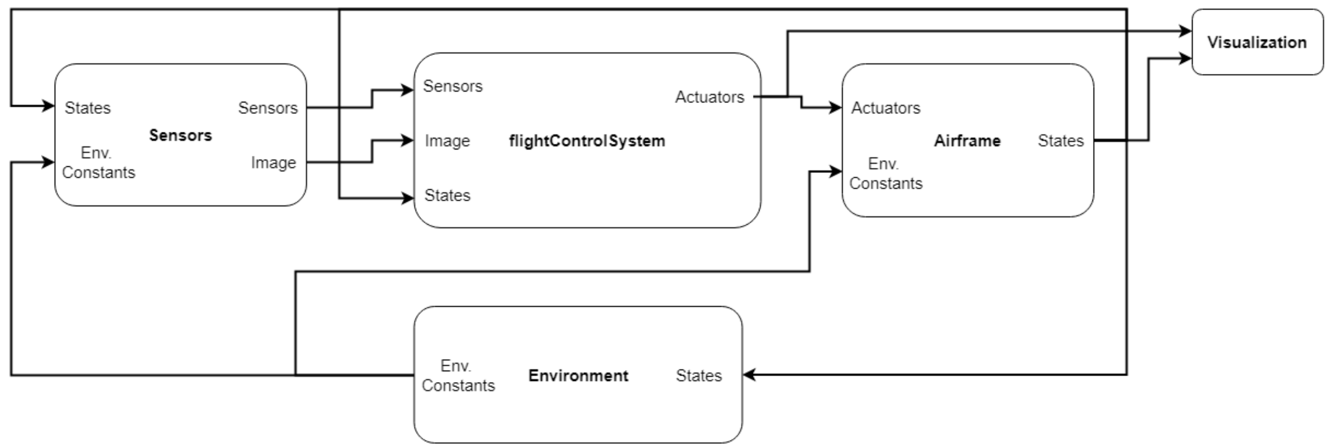

Fig. 4. Schematic of the simulation framework, as defined in the Simulink Support Package for Parrot Minidrones

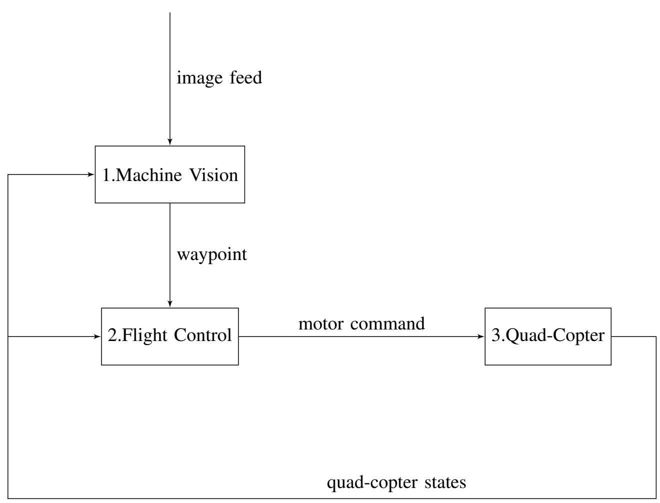

Fig. 5. Structure of the baseline solution

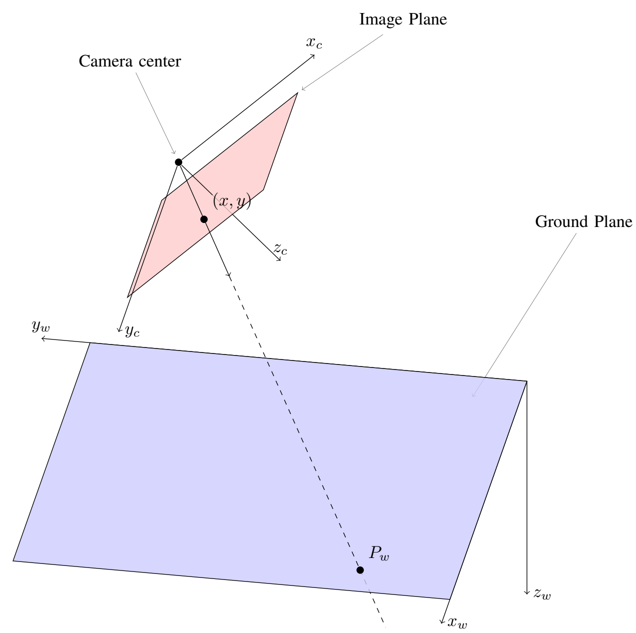

Fig. 6. Marker positioning.

thresholding algorithm with thresholds known a priori due to the constant colors of the simulated markers.

Then, using a projective transformation, the markers are placed on the ground in the world reference frame, see also Figure 6. First, the axis d_w passing through the center of the camera C_w, on which the centroid of the marker in the image is found, is determined:

$$d_w = R^{-1} \cdot K^{-1} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} X_c/Z_c \\ Y_c/Z_c \\ 1 \end{bmatrix} \tag{2}$$

where:

X_c, Y_c, Z_c	position of the marker center in the camera reference frame
x, y	coordinates of marker centroid in image
K	intrinsic camera matrix
R	Rotation matrix between camera and world coordinate frames

After this, the world coordinates of the marker center P_w can be computed using (2) and:

$$P_w = C_w + s \cdot d_w \tag{3}$$

knowing that the marker is situated in the ground plane $\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \cdot P_w = 0$. The whole procedure is applied assuming that the exact position of the drone is known, which is the case in the simulated environment.

The trajectory planning is done in a point-to-point manner by applying a first-order filter on the coordinates of the target marker, which means that the control algorithm receives a position to reach, spends most of the control effort to reach the said point, and moves to the next one when a threshold distance to the current target is reached. Equation (4) shows how the reference trajectory r is computed using the target coordinates t and the configuration parameter of the filter α.

$$r(k) = \alpha \cdot t(k) + (1 - \alpha) \cdot r(k-1) \quad (4)$$

The second component, which handles the navigation of the drone, consists of a cascaded feedback loop where a PI controller is used for the control of the 3D position of the drone, which in turn generates a reference for the inner loop PD controller that controls the attitude of the drone. Note that the controller for the height of the drone runs decoupled from the attitude controller.

The initial solution provided was not sophisticated or well tuned, yet it was stable and managed to complete the initial demo race track, see Figure 7 which shows the realised trajectory.

Since the competition would be held online, the recommendation for the teams was to focus on path planning and control, as in the simulation environment the default machine vision solution sufficed, as there would be no usual artifacts that affect machine perception of visual data that appear in the physical application (such as different lighting conditions, foreign objects that obstruct markers and so on).

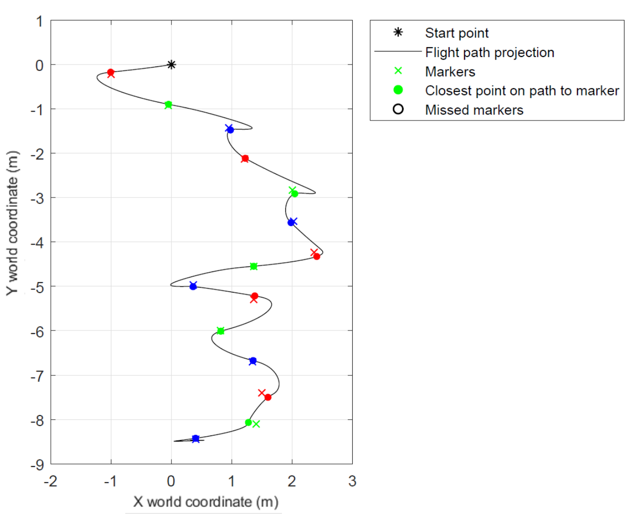

Fig. 7. Baseline solution trajectory, projected on the ground.

IV. First contest edition

A. Teams and rankings

For the first edition, we had five teams from four universities in four different countries. The teams, together with their composition and final ranking in the contest, are shown in Table II.

None of the teams fully implemented both the computer vision or control tasks, but nearly all of them worked on the path planning part. The jury decided to award all the teams that did this part 30% of the bonus points for B. On the more difficult mobility track one unforeseen circumstance arose: some solutions landed prematurely, ending the circuit run in a fast time, but missing most of the markers in the race track. To prevent favoring such failed trials in the evaluation due to their short time, the jury decided in on-the-spot discussions that for any solution that misses more than half the markers, the maximal time will be assigned.

TABLE II
Participant teams and team composition

Rank	Team	Institution	Study Level
1	QuadcopTeam	Universite Polytechnique Hauts de France	MSc
			MSc
			MSc
2	AirwolfRTS	University of Maribor	PhD
			PhD
			MSc
3	libellule	Universite Polytechnique Hauts de France	PhD
			BSc
			BSc
4	Millenial Mambo	Technical University of Cluj-Napoca	MSc
			BSc
			-
5	5C++	University of Wuerzburg	MSc
			MSc
			MSc

The first and second place received prizes in the form of hardware components related to the subject of drone control.

B. Example solution from one the teams

The team Millenial Mambo provided both a planning component and a new implementation for the control component.

1) Path planning component: The desired path for the navigation of the drone was generated as a spline trajectory that passes through the known markers. Upon finding a new marker on the ground, the spline is regenerated taking into account this new point to be included in the future path. The setting for the flight height was changed to the maximum that the simulated camera allowed, so that as many points as possible could be seen in order to generate a spline trajectory with as much information as possible. Figure 8 shows an example spline trajectory that was generated during run-time with partial future marker information.

The reference scheduler used a carrot-on-a-stick approach, which means that the reference point passed to the navigation component is always ahead of the drone on the desired path. The distance at which the future reference would be placed was a function of the absolute value of the derivative of trajectory ahead. This was implemented so the next reference would be farther from the drone when the area ahead had no sharp turns, thus promoting speed, while areas where turns had to be made would make the scheduler provide a closer point ahead, which would slow the drone down to allow it to make sharper turns.

2) Navigation control component: For position control an LQ tracking controller was designed and tuned in an aggressive manner. This solution promoted speed on the track, as it was noticed during early trials that with the default controller significant time was lost while waiting for the drone to stabilize over markers. The main idea was to try to preserve momentum while searching for the next point to reach. The drone was modelled as a MIMO system with

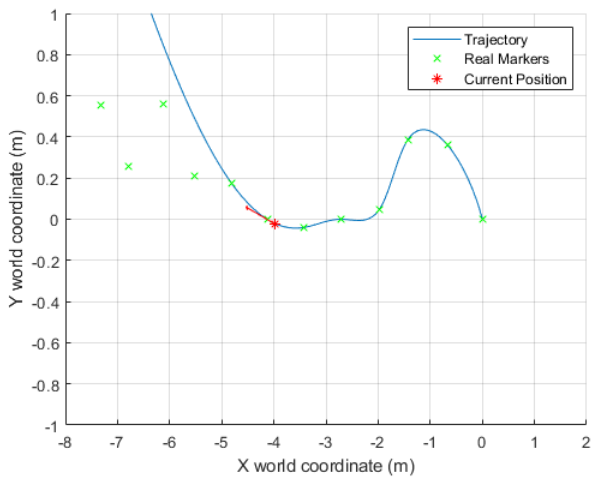

Fig. 8. Example trajectory generated with the spline method. The drone moves from right to left, is currently at the red point, and sees two markers in advance.

12 states: 3 XYZ position states, 3 linear velocities, 3 angles for the pitch, roll and yaw, and 3 angular velocities. As the system has 4 inputs (the rotation of 4 independent motors), 4 states were chosen for tracking. The states chosen were the XYZ coordinates and the yaw angle. The rest of the states will tend to be 0 during tracking, which helps implement hover stability.

The tuning of the LQ controller is done via two matrices: Q for the state errors weights, and R for the control signal weights. For Q, the weights were chosen in the following decreasing order (which can be interpreted as the decreasing order of importance of state tracking):

- Z position: for increased stability at the desired height
- X and Y position: to emphasize the fast navigation requirement
- yaw, pitch, roll
- derivatives of position and angles

The R matrix was tuned to allow the drone to maintain the desired height, and to rotate at a rate that enables it to navigate quickly and stably to the desired reference.

3) Performance of the solution: While this solution worked well on the speed track, achieving the fastest race time on it while narrowly missing some markers, it did not perform well on the mobility track. One possible improvement that could have been made consists of a way to compensate the extra distance needed to "touch" the markers in the turns, as the high velocity approach made the drone miss some of the markers by not turning in a large enough motion.

C. Feedback from the contestants

In order to better organize future editions of the Co4Air drone marathon contest, the participants were given a feedback form where they were asked to rate their overall satisfaction with the competition, and to express their opinion about aspects of the competition that they enjoyed or that they found lacking. We organized a survey with all contestants, where we attempted to gauge:

- Their experience with the contest, overall and specifically on the online format.
- How useful the contest was for them and whether they wish to see such events integrated in their curriculum.
- Their opinion on the scoring rules.
- What should be improved in general, for the online experience, etc.
- In what way should each participant's institution support them.
- How students prefer their contest performance to be recognized.

A total of 9 feedback forms were collected. The overall feedback was positive, as the general satisfaction of the contestants was either rated as good (6) or very good (3).

When asked if they would like if their studies would regularly include contests designed in a similar manner to the Co4AIR Marathon – Drone Racing Competition, and all the contestants agreed with the statement, with 8 votes selecting the strong agreement option.

When asked if the work effort invested by the contestants in the competition helped them in their studies, research or job, 8 of the 9 contestants agreed.

Perhaps surprisingly, just a basic ranking of their team was the top-voted option for performance recognition. Apparently, actual prizes in cash or hardware are not that important for the students. This means that in the future editions of the contest the sponsorship should be used towards a better organization of the event itself, rather than prizes.

V. Conclusions and Future Improvements

The feedback forms contained many possible improvements, including the following highlights:

- An introduction event where the competition, procedure, baseline solution, etc. are presented and the teams get to know each other. This can be included right after the registration stage.
- A specific way of integrating the contest in the curriculum, via a special lecture dedicated to it and ECTS scores. This is highly recommended for institutions that plan to send participants to future editions.
- The students were surprised by the very challenging "maneuverability" track and suggested that we should include more example tracks with the baseline solution.
- A "live score" table where people can see their performance on the fly.
- Students suggested knowing the scoring weights in advance. Due to the reasons explained before, we prefer to not do this, but a middle ground can be to explain which parts of the solution will be more important without giving exact numbers.

The overarching line of the feedback above is that students find our contest useful, and would absolutely like to see such an event in their curriculum. We therefore believe that the Co4AIR Marathon is a useful educational tool. To move on to the next step, after revising the contest design as described above, we aim to promote adoption of this contest both in our universities and others. This very paper is a main component of our promotion strategy.

References

[1] D. C. Tsouros, S. Bibi, and P. G. Sarigiannidis, “A review on uav-based applications for precision agriculture,” *Information*, vol. 10, no. 11, p. 349, 2019.

[2] The Drone Racing League, “The drone racing league autonomous competition.” https://thedroneracingleague.com/airr/, 2022. [Online; accessed 20-Feb-2022].

[3] Lockheed Martin, “Innovation challenge AlphaPilot.” https://www.lockheedmartin.com/en-us/news/events/ai-innovation-challenge.html, 2022. [Online; accessed 20-Feb-2022].

[4] ICUAS, “Unmanned Aerial Vehicle (UAV) Competition.” http://www.uasconferences.com/2022_icuas/uav-competition/, 2022. [Online; accessed 20-Feb-2022].

[5] R. Madaan, “Game of Drones at NeurIPS 2019: Simulation-based drone-racing competition built on AirSim.” https://www.microsoft.com/en-us/research/blog/game-of-drones-at-neurips-2019-simulation-based-drone-racing-competition-built-on-airsim/, 2019. [Online; accessed 01-May-2022].

[6] Mathworks, “MathWorks Minidrone Competition.” https://de.mathworks.com/academia/student-competitions/minidrones/ifac-2020.html, 2019. [Online; accessed 19-Oct-2019].

[7] S. Geuther, L. Glaab, T. Johnson, R. Haq, and D. Hare, “Safeguard with autonomous navigation demonstration technical closeout report,” 2020.

[8] ELROB, “The European Land Robot Trial.” https://www.elrob.org/, 2019. [Online; accessed 19-Oct-2019].

[9] A. Bermudez, R. Casado, G. Fernandez, M. Guijarro, and P. Olivas, “Drone challenge: A platform for promoting programming and robotics skills in k-12 education,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 1, 2019.

[10] L. Tamas, C. Sándor, S. Pável, E. Wieser, A. Blaga, P. Boda, A.-O. Fülöp, A. Ursache, A. Z. Zöld, A. Kopacz, B. Lázár, *et al.*, “The ClujUAV student competition: A corridor navigation challenge with autonomous drones,” in *Proceedings 21st IFAC World Congress (IFAC-20)*, 2020.

[11] L. Busoniu, “Report on the Co4AIR Marathon – Drone Racing Competition.” http://co4air.eu/sites/co4air.eu/files/co4air_report_on_co4-marathon.pdf, 2021. [Online; accessed 19-Feb-2022].

[12] V. M. Maer, “Design and reference solution of an autonomous quadcopter racing competition,” BSc thesis, Technical University of Cluj-Napoca, 2020.