Special Issue: The Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2020

Regular Article

Autonomous Aerial Robot for High-Speed Search and Intercept Applications

Alejandro Rodriguez-Ramos[1], Adrian Alvarez-Fernandez[1], Hriday Bavle[1], Javier Rodriguez-Vazquez[1], Liang Lu[1], Miguel Fernandez-Cortizas[1], Ramon A. Suarez Fernandez[1], Alberto Rodelgo[1], Carlos Santos[2], Martin Molina[1], Luis Merino[3], Fernando Caballero[3] and Pascual Campoy[1]

[1]Universidad Politécnica de Madrid, Centre for Automation and Robotics
[2]Universidad de Alcalá de Henares
[3]Universidad Pablo de Olavide

Abstract: In recent years, high-speed navigation and environment interaction in the context of aerial robotics has become a field of interest for several academic and industrial research studies. In particular, Search and Intercept (SaI) applications for aerial robots pose a compelling research area due to their potential usability in several environments. Nevertheless, SaI tasks involve a challenging development regarding sensory weight, onboard computation resources, actuation design, and algorithms for perception and control, among others. In this work, a fully autonomous aerial robot for high-speed object grasping has been proposed. As an additional subtask, our system is able to autonomously pierce balloons located in poles close to the surface. Our first contribution is the design of the aerial robot at an actuation and sensory level consisting of a novel gripper design with additional sensors enabling the robot to grasp objects at high speeds. The second contribution is a complete software framework consisting of perception, state estimation, motion planning, motion control, and mission control in order to rapidly and robustly perform the autonomous grasping mission. Our approach has been validated in a challenging international competition and has shown outstanding results, being able to autonomously search, follow, and grasp a moving object at 6 m/s in an outdoor environment.

Keywords: MBZIRC, UAV, autonomous, outdoor, grasping

1. Introduction

Research contributions in Unmanned Aerial Vehicles (UAVs) have increased considerably over the past decade, both in academic and industrial contexts (Gioioso et al., 2014; Klingbeil et al., 2014; Loianno et al., 2016; Bachrach, 2021). Indeed, these types of robotic systems, such as multirotor

Received: 29 September 2020; revised: 26 March 2021; accepted: 16 May 2021; published: 23 June 2022.

Correspondence: Alejandro Rodriguez-Ramos, Universidad Politécnica de Madrid, Centre for Automation and Robotics, Email: alejandro.rramos@upm.es

DOI: https://doi.org/10.55417/fr.2022044

systems, provide a high maneuverability and an elevated potential level of autonomy in the field of complex missions. Furthermore, the 6-Degrees Of Freedom (DOF) design of a multirotor allows for both precise and slow maneuvers, as well as fast and aggressive navigation or environment interaction, which can be considerably useful in the context of field robotics.

A notable amount of research has been carried out in the area of autonomous navigation or environment interaction (e.g., object grasping). Stated autonomous tasks involve the development and coordination of several complex software components, as well as the integration of miniaturized sensors that provide usable rich information in order to successfully complete the endeavor. In particular, sensors such as RGB cameras or LIDARs are able to provide high-dimensional information of the environment, which has to be efficiently processed onboard in order to perform the mission. In addition to the aforementioned challenges, high-speed environment navigation and/or interaction in the context of Search and Intercept (SaI) applications pose an enormous challenge due to the available onboard processing resources, sensor weight, and algorithms to compute the high-dimensional information, efficiently and in a real-time trend.

In this respect, regarding high-speed SaI applications, the complexity of the task increases with the required speed of the mission, since the algorithms have to be processed at a higher rate, and the dynamics of the system (e.g., moments of inertia) show up accordingly. Normally, the robotic system is modeled for control, incurring several assumptions that simplify the design. These assumptions may not be valid as speed increases, since the nonlinearities of the system are more notable, among others. Furthermore, the application of object interception with UAVs constitutes an increased challenge due to constraints on actuator design and its effect on the dynamics of the system, as well as the required perception and control algorithms, which enable the interaction maneuvers.

On this subject, a fully autonomous UAV solution for SaI applications is proposed in this work. Although a Global Positioning System (GPS) and a Real Time Kinematic (RTK) system were used in the outdoor scenario, the complete mission has been approached based on the RGB camera sensor as the main information source, which poses an additional challenge due to the environmental area size, conditions, and the amount of information to be processed at high speeds. In this study, the complete UAV system is described, as well as the actuator design for this particular task. In addition, all the software components that enabled the vision-based behavior have been detailed. The main contributions of the proposed system are summarized herein: i) A custom UAV has been built with a flexible hardware architecture along with a novel gripper design in order to perform the grasping maneuver. ii) A complete software architecture is designed in conjunction with the hardware design in order to efficiently and autonomously perform a complete mission in the context of SaI applications. iii) A thorough evaluation of the entire system has been carried out in both simulated and real flight environments, leading to a successful performance in an international robotics competition.

Concretely, the task of autonomous object grasping from a noncooperative target UAV that is flying at high speeds has been selected as the baseline task. This mission has been extracted from the 2020 Mohamed Bin Zayed International Robotics Competition (MBZIRC),[1] in which several members of the Computer Vision and Aerial Robotics (CVAR) group have participated. In particular, the task of autonomous object grasping corresponds to Challenge 1 (out of three challenges) defined in this competition. Our team ended up in the 3rd position in the Grand Challenge,[2] which is a parallel combination of the proposed three challenges of the competition.

The remainder of the paper is organized as follows: Section 2 provides a literature review, analyzing the most relevant related work. Section 3 states the problem that this work aims to solve. Section 4 describes our proposed hardware platform, detailing all the sensory and actuation resources. Section 5 describes the developed software stack and their components. Section 6 outlines the carried-out experiments and their corresponding results, and Section 7 remarks on and discusses

[1] https://www.mbzirc.com/challenge/2020
[2] https://www.mbzirc.com/winning-teams/2020/challenge4

the most relevant experimentation outcomes. Finally, Section 8 concludes the paper and indicates future lines of research.

2. Related Work

The present work has been developed in the context of fully autonomous aerial robotics, with a broad coverage of state-of-the-art techniques. On this basis, several studies in the literature provide methods and techniques that are adjacent to the present study. In the following section, the most relevant related work is included, with a special emphasis on the tangential aspects to the methods provided.

2.1. High-Speed Navigation for Autonomous UAVs

In the broad context of robotics, multirotor aerial robots in particular provide a wide maneuverability, being technically able to navigate at high speeds due to the low friction of air and their 6-DOF design. Nevertheless, autonomous navigation at such high speeds in cluttered or interactive environments poses an enormous challenge not yet solved by the research community. In this regard, there are several studies in the literature showing UAVs that are able to achieve notable velocities in fully autonomous mode.

Autonomous navigation in cluttered and/or unstructured environments has been thoroughly studied in the literature. However, high-speed navigation in such environments has been recently explored. In (Do et al., 2019), Parrot Bebop multirotor was able to handle dynamic obstacles in indoor environments at on average 1 m/s and using an RGB camera as the main sensor. In (Liu et al., 2016), a fast and efficient path planning algorithm for UAV navigation in unstructured environments was proposed. It provided a peak velocity of 3 m/s using an RGB-D sensor. A Smoothing And Mapping With Inertial State Estimation (SAMWISE) navigation system was proposed in (Steiner et al., 2017), achieving 20 m/s in an outdoor environment. It was aided by heterogeneous sensory inputs for state estimation, such as LIDAR, RGB-D, and Inertial Measurement Units (IMUs). In (Florence et al., 2020), an MPC scheme and local estimation were utilized for fast reaction, achieving a peak velocity of 5.5 m/s in simulation.

As multirotor hardware design is potentially able to provide agile or aggressive maneuvers, some studies have researched the generation of autonomous acrobatic movements. In (Kaufmann et al., 2020), acrobatics maneuvers were trained in simulation with no expert demonstrations and deployed in an outdoor environment. The ego-motion was provided by feature tracks, and the IMU allowed for motion recovery. In (Falanga et al., 2017), onboard localization and perception were utilized in order to fly through tilted narrow gaps. The window estimation was achieved through an RGB camera, and a maximum velocity of approximately 2.6 m/s was reached. In (Loianno et al., 2016), high-speed narrow gap crossing was achieved at a maximum speed of 4.5 m/s. Visual-Inertial Odometry (VIO) was used for state estimation, and the window was not being actively detected.

In the 2017 Mohamed Bin Zayed International Robotics Competition (MBZIRC), a fast landing maneuver on top of a moving platform challenge was proposed. Several research groups have tackled the challenge with a variety of techniques. In (Beul et al., 2017), a DJI Matrice 100 was proposed for landing at a maximum speed of 2 m/s. In (Cantelli et al., 2018), Tracking-Learning-Detection (TLD) was used along with a pretrained model for detection. The team reached 4th position, having autonomously landed in 140 s. In (Baca et al., 2017), computer vision techniques and an MPC scheme are used, having provided a complete landing maneuver in 25 s. In (Beul et al., 2019), two RGB cameras were utilized for detection, and a maximum speed of 2 m/s was achieved. In the stated reference, the team reached 1st position in the Grand Challenge and 3rd position in two subchallenges. In (Bähnemann et al., 2019), VIO and MPC methods were implemented, with a maximum velocity of 8 m/s.

In this work, the main sensor for target state estimation has been a 3-DOF RGB camera gimbal. In the complete approach, a maximum speed of approximately 6 m/s was reached in outdoor space when following a moving target.

2.2. Object Grasping for Multirotor Systems

Object interaction, and concretely object grasping, in the context of multirotor systems poses a notable challenge, since it requires several components to be performing simultaneously, from complex controllers to real-time perception algorithms. In this respect, several studies have provided object grasping solutions for certain scenarios.

Multiple grasping techniques and maneuvers are provided in the literature. A claw-inspired design has been used to perform the grasping of cylindrical objects with UAVs. In (Thomas et al., 2013), the 2-DOF gripper was inspired on the claw of an eagle. The dynamic model was determined using Lagrange mechanics, and the trajectory was computed by restricting positions in space. In (Ramon Soria et al., 2017), a 3-DOF arm and a RGB-D camera sensor were utilized for 3D estimation of the objects. In (Pounds and Dollar, 2014; Pounds et al., 2011c,a), a robotic arm with no DOF was designed for helicopter grasping. A PID scheme was used for control. In (Parra-Vega et al., 2013), multiple UAVs were used for object grasping, with advanced nonlinear control for swarms in simulation. In (Spurnỳ et al., 2019), an Unscented Kalman Filter (UKF) for object state estimation and prediction was used. MPC control was proposed and RTK was used for positioning. In (Spica et al., 2012), a planning strategy relying on differential flatness was then proposed to concatenate one or more grasping maneuvers by means of spline-based subtrajectories. In (Pounds et al., 2011b), a preliminary analysis and experiments for reliable grasping of unstructured objects with a robot helicopter were presented. Key problems associated with this task were discussed, including hover precision, flight stability in the presence of compliant object contact, and aerodynamic disturbances. In (Gioioso et al., 2014), a flying hand was presented, where a robotic hand consisting of a swarm of UAVs was able to grasp an object. Each UAV contributes to the grasping task with a single contact point at the tooltip. In (Kim et al., 2013), a quadrotor with a 2-DOF robotic arm was proposed. In (Bodie et al., 2019), an omnidirectional aerial manipulation platform for robust and responsive interaction with unstructured environments, toward the goal of contact-based inspection, was proposed. In (Yang and Lee, 2014), a novel backstepping-like end-effector tracking control law, which allowed different roles to be assigned for the center-of-mass control and for the internal rotational dynamics control according to task objectives, was designed. In (Fumagalli et al., 2014), an impedance controller for grasping with a 3-DOF arm was described.

In this study, a novel gripper, placed above the quadrotor's center of mass and statically fixed to its frame, has been proposed. This design allows for fast grasping maneuvers (from a lower position with respect to the target object) of hanging objects, and it enables a smooth movement throughout the whole grasping trajectory without the necessity of abrupt changes in position, velocity, or acceleration direction.

2.3. Vision-Based Target Following for Autonomous UAVs

Autonomous target following is a topic of increasing interest due to the successful UAV applications in the industry. On this subject, following an unknown or noncooperative target is a challenging task that involves complex perception and control systems. In the literature, a number of studies have addressed the task of autonomous object following.

Some approaches rely on an online user-defined Region-of-Interest (RoI) within the image plane during the execution of a real flight in order for the UAV to be able to follow a tracked object. A PID control scheme was used to control a multirotor aerial robot in (Pestana et al., 2014) based on the OpenTLD tracker in the image plane. In (Li et al., 2016b), the authors developed a novel frequency and image-plane-domain tracker with latter pose estimation of the object and standard PID control. A state-of-the-art tracker was used in (Mueller et al., 2016) and a standard PID control scheme within the image plane was established. The stated work also performed real-flight validation tests and developed a novel camera handover strategy in order to enable long-term operation through several UAVs. In (Haag et al., 2015), correlation filters for short-term tracking and a redetection strategy based on TLD were utilized for IBVS control.

Classical control schemes (e.g., PID) along with computer-vision techniques (e.g., color-based detectors) have also been explored in the literature. In (Teuliere et al., 2011), a color-based detector fed a particle filter that aided a pose-based PID control scheme. A miniature robotic blimp was used in (Yao et al., 2017) as the robotic platform, with human-pose estimation and standard PID control. In (Nägeli et al., 2018), human tracking was carried out over a long period of time and long distances by means of active infrared markers and the estimation of the pose of both human and UAVs without calibrated or stationary cameras. In (Mondragón et al., 2011), an IBVS controller and a color-based detector were developed for object following. In (Olivares et al., 2011), the fuzzy servo controller was explored along with a color-based detector. A nonlinear adaptive observer was used in (Choi and Kim, 2014) for target-object estimation, and a guidance law for target-object tracking was developed. In this case, the vision sensor was modeled and the complete approach was validated under numerical simulations.

In this work, a camera gimbal aids a machine learning based algorithm for target detection, among others. An MPC scheme was integrated for obstacle-free trajectory following of the moving target.

2.4. Fully Autonomous Aerial Sensors and Onboard Computation

Fully autonomous systems require a certain amount of both computational and sensory resources. Nevertheless, depending on the employed techniques, an autonomous UAV can incorporate either heavier exteroceptive and/or lighter proprioceptive sensors for performing the high-level mission. Several strategies have been found in the literature in this respect.

A LIDAR sensor provides rich and precise information about the surrounding environment. Due to this fact, LIDAR is a commonly utilized sensor in fully autonomous UAV designs. In (Tomic et al., 2012), a Hokuyo laser rangefinder UTM-30LX and two onboard computers (1.6 GHz Intel Atom, 1 GB DDR2 and ARM Cortex-A8, 720 MHz, 512 MB RAM) were used for indoor and outdoor urban search and rescue applications. In (Sampedro et al., 2019b), a Hokuyo laser rangefinder UTM-30LX and an Intel NUC6i5SYK featuring a 2.9 GHz Intel Core i5-6260U CPU were integrated. In the stated work, an additional Intel Realsense R200 RGB-D camera and Lightware SF10/A altimeter were included to enable a higher versatility of the system for search and rescue applications in indoor environments. An AISKopter continuously rotating 3D laser scanner and an Intel Core i7, 8GB RAM onboard computer were selected in (Klingbeil et al., 2014). Additionally, a global shutter camera, a stereo camera pair , and ultrasonic sensors were also integrated.

Other techniques use stereo camera pairs as the main sensor for perception in autonomous missions. In (Schmid et al., 2013), the stereo images were processed in an Intel Core2Duo SU9300@1.86GH and Spartan 6 LX75 FPGA. The selected onboard computer was an OMAP3530 ARM Cortex A8@720MHz for indoor and outdoor navigation applications. Bebop 2 was utilized in (Bavle et al., 2018) for navigation in indoor environments, along with an Intel R200 RGB-D camera and a Parrot Slam Dunk stereo camera pair. In (McGuire et al., 2017), a 4g micro stereo camera pair and an STM32F4 processor were integrated for obstacle avoidance and reactive navigation. In (Perez-Grau et al., 2018), Orbbec Astra stereo vision was utilized for navigation in GPS-denied areas.

Monocular strategies for autonomous navigation are challenging tasks, since the spatial information is not directly available in the raw information provided by the sensor. Nevertheless, stated RGB cameras are low-cost and passive sensors that provide high-dimensional information that can be exploited with different approaches. In (Loianno et al., 2016), a downward facing VGA camera with a 160º field of view, a VGA stereo camera, and a 4 K camera were integrated for estimation, control, and planning for aggressive flight (running on a Qualcomm Snapdragon board). In (Ross et al., 2013), an offboard monocular approach for reactive navigation in cluttered natural environments with an ArDrone 2.0 was proposed. In (von Stumberg et al., 2017), an offboard monocular simultaneous localization and mapping (SLAM) -based technique was implemented for navigation in cluttered indoor environments with an ArDrone 2.0.

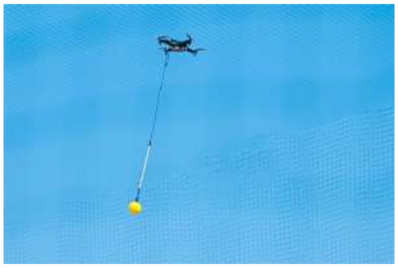


Figure 1. Target UAV with tethered ball in the real competition environment.

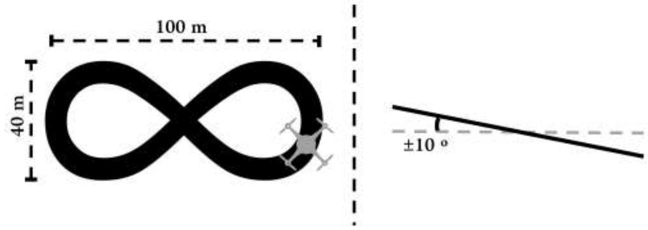


Figure 2. Structure of the designed competition arena for Challenge 1 in the 2020 MBZIRC competition.

In this study, the main sensors for ego-motion and target state estimation have been an RGB camera gimbal, an IMU, an RTK positioning device, and two VGA stereo pairs. The selected onboard computer has been an NVIDIA Jetson AGX Xavier. The complete strategy is primarily based on an RGB camera gimbal for searching, detection, target state estimation, and a grasping maneuver.

3. Problem Statement

In this work, the task of autonomous object grasping has been selected as the baseline application. The stated task has been extracted from Challenge 1, proposed in the 2020 MBZIRC competition. It consists of the grasping of an object that is mounted on a target UAV. The target UAV is flying within a volume of 100 m × 40 m × 20 m, describing a lemniscate shape inside a tilted 3D plane, as shown in Figure 2. In the competition, the whole Challenge 1 is designed to last 15 min per trial, and the velocity of the target UAV is scheduled to 8 m/s for the first 8 min and 5 m/s for the next 7 min. In Figure 1, the target UAV with a tethered ball in the real competition environment has been depicted.

The overall objective of the challenge was to autonomously perform a series of interaction tasks using aerial vehicles at high speeds. The object to be grasped corresponded to a ball of an approximate radius of 11 cm, which is hanging from a cable of approximately 1.5 m in length. Once the item has been removed from the tether, it must then be delivered to a prespecified 10 m × 10 m landing location inside the arena.

Furthermore, the UAV (or a team of a maximum of three UAVs) will have to locate and physically interact with a set of static objects (balloons of 45 cm in diameter), which are of unknown color, randomly distributed within the outdoor flight arena, and tethered to the ground. For added difficulty, once these static items are detected, the UAV will have to navigate and burst each balloon using any component installed onboard.

4. Hardware Configuration

4.1. Aerial Vehicle Platform

The aerial platform used for Challenge 1 of the MBZIRC competition was a modified version of the DJI Matrice 210 RTK v2 UAV, shown in Figure 3. This robust platform was chosen as the main

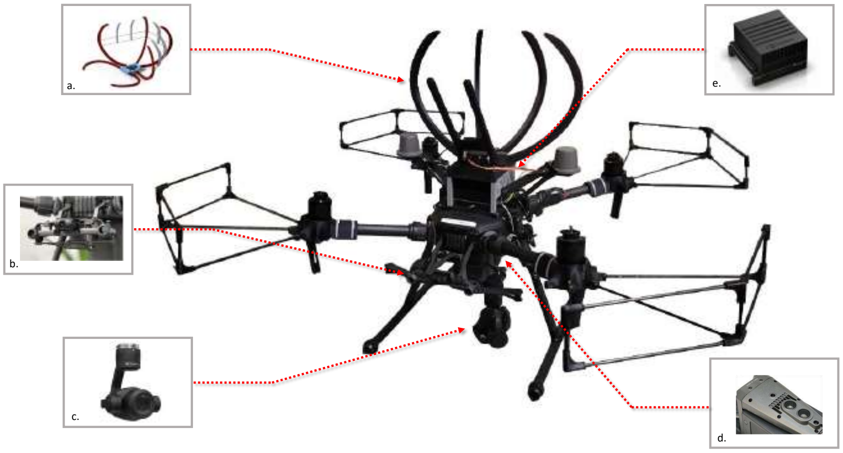


Figure 3. Aerial platform used for Challenge 1 with the onboard components: (a) Gripper with a Servo Mechanism. (b) DJI RGB Stereo Pair. (c) DJI X4s RGB Camera. (d) Sonar Altimeter & Bottom Camera. (e) Nvidia Xavier.

aerial vehicle due to the demanding flight performance required by the different tasks, i.e., flights up to 10 m/s, large payload, and extended flight times. The aerial robot was equipped with RTK and GPS receivers, a DJI Visual Position System (VPS), a sonar altimeter, and a high resolution Zenmuse X4s onboard camera featuring a 20MP, 1" sensor and an 84 degree FOV high-resolution lens.

Additionally, the aerial platform was equipped with the Single Board Computer (SBC) NVIDIA Jetson AGX Xavier with an 8-core ARM v8.2, 64-bit CPU running Ubuntu Linux 18.04 Bionic Beaver for onboard computing, and a 512-Core Volta GPU for image processing tasks. All interprocess communications were handled using the Melodic Morenia distribution of the Robotic Operating System (ROS) middleware.

As previously mentioned, the objective of Challenge 1 was to retrieve an item tethered to a separate target UAV in motion. To this end, the DJI Matrice 210 UAV was equipped with a custom-designed gripping mechanism, shown in Figure 3, which enabled the grasping of the item. It was also equipped with custom-designed protections in order to protect the propellers of the aerial robot against collisions from the target UAV or the tethered item. Sections 4.2 and 4.3 explain in detail the design mechanisms for the gripper and the protections, respectively.

4.2. Gripper Design

A bio-inspired design has been selected for the gripper actuator. There exist carnivorous plant species, such as the Venus Flytrap [Figure 4(a)], which are able to trap flying insects without the necessity of performing fast grasping movements. The stated species have two moving parts that are able to be fully closed, trapping the insect that was flying in between. To prevent the insects from escaping, there are sharp and interlocking ends in the borders of each part.

The gripper design is required to provide a wide volume for catching the ball with the minimum weight possible. Also, it is required to allow the tether to pass through the gripper in order to successfully release the ball from the target drone and trap it into the gripper. The gripper is composed of two main parts, one mobile and one static. On the one hand, the cage, which is the static part, consists of four long *fingers* followed by five small *fingers* bounded together with carbon fiber bars and epoxy resin. Moreover, nylon threads were used for covering the bottom part of the cage, leaving free space between the upper part of the long fingers. This free space between fingers allows the ball tether to flow and go through the cage. On the other hand, the mobile part of the gripper consists on one planar surface built with two *fingers* joined with nylon. When the gripper

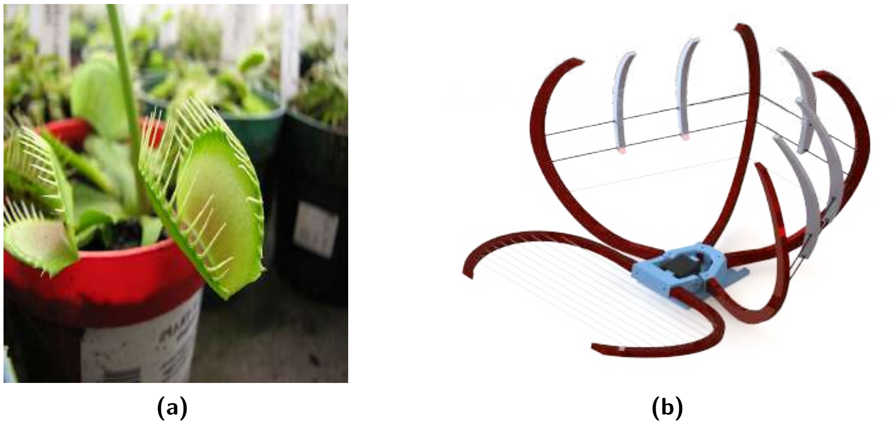


Figure 4. (a) Venus Flytrap. (b) Gripper CAD.

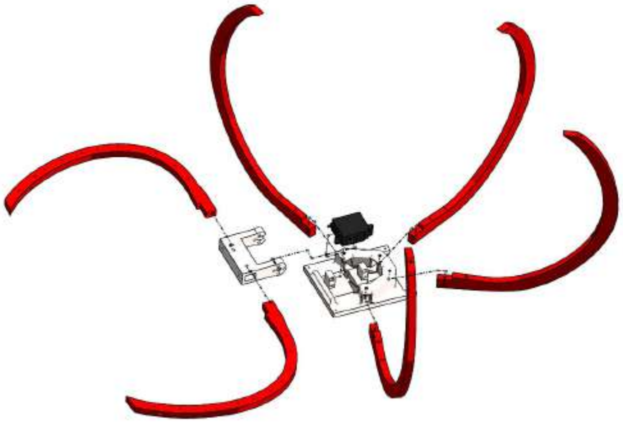


Figure 5. Exploded view of the main structural components of the gripper.

is closing, this planar surface rotates into the cage, catching the ball inside it. A servomotor was used for closing the gripper. The stated servo is controlled trough an Arduino Nano, in charge of closing and opening the gripper whenever the high-level mission components require the catching maneuver. Finally, a single point laser sensor pointing upwards was placed inside the gripper in order to detect if the ball is trapped inside.

PolyLactic Acid (PLA) 3D-printed pieces for the whole design of the gripper have been utilized, except for the carbon fiber bars and the nylon wires. PLA was used, over other 3D printing materials, due to its ease of handling and its rigidness. Carbon fiber bars were used because of their light weight and tenacity, and they were attached to the *fingers* using epoxy resin. To cover open patches (to avoid the ball dropping out of the gripper), nylon threads were used. Nylon can resist the impacts without stretching or breaking.

4.3. Protection Design

As explained in Section 3, the challenge was composed of one main task and another complementary task. However, both tasks required physical interaction between the aerial vehicle and other objects, such as balls or balloons. This physical interaction implies the risk of propellers touching hard objects, such as the pole that sustained the balloons, or the bar that attached the ball to the moving UAV. These structures may cause a propeller to break, which could lead to a UAV full failure or crash.

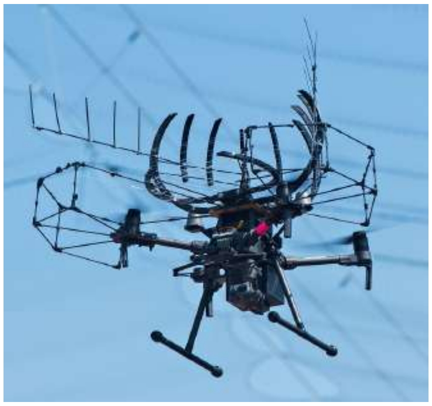
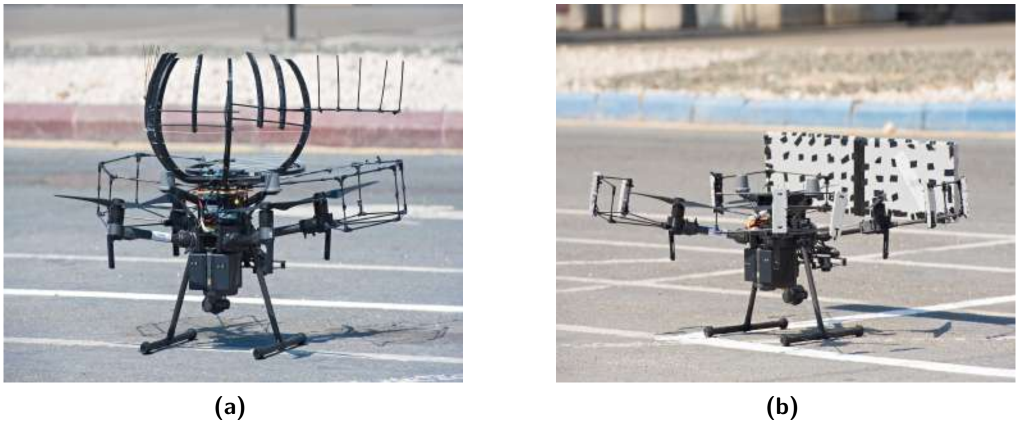


Figure 6. The complete proposed platform, with the gripper and all protections included in the real competition environment.



(a)

(b)

Figure 7. (a) Full covering front protections. (b) Contact covering protections with needle plates.

The risk of failure was a strong motivation for the design of custom protections for both aerial platforms. Stated protections have to accomplish design constraints. First, the propellers must be protected from physically touching any pole, bar, or ball. Second, they must be lightweight and easy to assemble, disassemble, and repair. Two configurations for the protection design were built, which are related to the task to be carried out, and they are as follows:

- **Full covering**: This configuration is meant for the ball grasping task, where the UAV is required to intercept a target UAV with a ball attached to it. In the stated task, the protections fully cover the UAV arm, including the propellers, preventing them from breaking by touching other objects while they are rotating at high speeds. These protections are attached to the UAV arms, and to a plate placed between the onboard computer and the gripper. Attaching the structure to these points improves the rigidness of the protections, preventing them from bending and colliding with the propellers. Additionally, all the gaps between the carbon fiber bars were covered with nylon, avoiding the risk of any object impacting with the propellers [Figure 7(a)].
- **Contact covering**: For the objective of bursting balloons, the main threat are the poles that support said balloons. These poles were static on the ground, which reduced the risk of any object colliding with the tops of the propellers. In this case, lighter protections, which were easier to replace when needed, were selected. In this case, four isolated protections, one attached to each UAV arm, covered the propellers in case the robot collided with the poles (Figure 8).

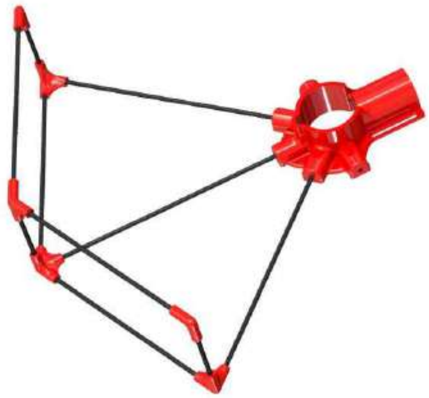


Figure 8. CAD of one isolated contact covering protection.

To pierce the balloons, multiple extruded polystyrene plates with several needles were attached along all the protections, with one wide plate between the front protections in order to burst the balloons by frontally colliding with them (Figure 8).

5. System Description

To perform the complex tasks encountered in Challenge 1 of the MBZIRC, a robust and flexible system architecture is essential for efficient development and validation of both software modules and hardware interfaces. The system architecture that is presented in this work has been developed using the Aerostack framework (Sanchez-Lopez et al., 2017, 2016).

Aerostack is an open-source multipurpose software framework for the development of autonomous multirobot unmanned aerial systems created by the Computer Vision and Aerial Robotics (CVAR) group. The Aerostack modules are mainly implemented in C++ and Python languages and are based on a Robot Operating System (ROS) for intercommunication between the different components. To provide higher degrees of autonomy, Aerostack's software framework integrates state-of-the-art concepts of robotics and artificial intelligence. Aerostack has been used in the development of complex robotic systems related, for example, to natural user interfaces (Fernandez et al., 2016), surface inspection (Molina et al., 2018), coordinated multirobot systems (Sampedro et al., 2016), landing on moving platforms (Rodriguez-Ramos et al., 2017), and search-and-rescue missions (Sampedro et al., 2019a). For a deeper understanding of Aerostack, we refer the reader to the extensive documentation and publications that are available on its website.[3] It should be noted that, although the Aerostack framework is able to provide predefined components and intercommunication methods to provide autonomy to UAVs, the components that are described in this work were completely designed and developed for the aim of this competition. All the components described in this study represent a novel contribution of this work.

Using the Aerostack software framework, a new system architecture design was developed for the tasks presented in the challenge. Figure 9 shows the functionalities that have been implemented in this work. In the figure, colored rectangular boxes represent data processing units (or processes in short) that are implemented as ROS nodes. They are organized in the following main components:

- *Sensor-Actuator Interfaces:* to receive data from sensors from the aerial platform and send commands to robot actuators.
- *Communication Channel:* Based on the Aerostack framework, our architecture uses a common communication channel that contains shared dynamic information between processes. This

[3] www.aerostack.org

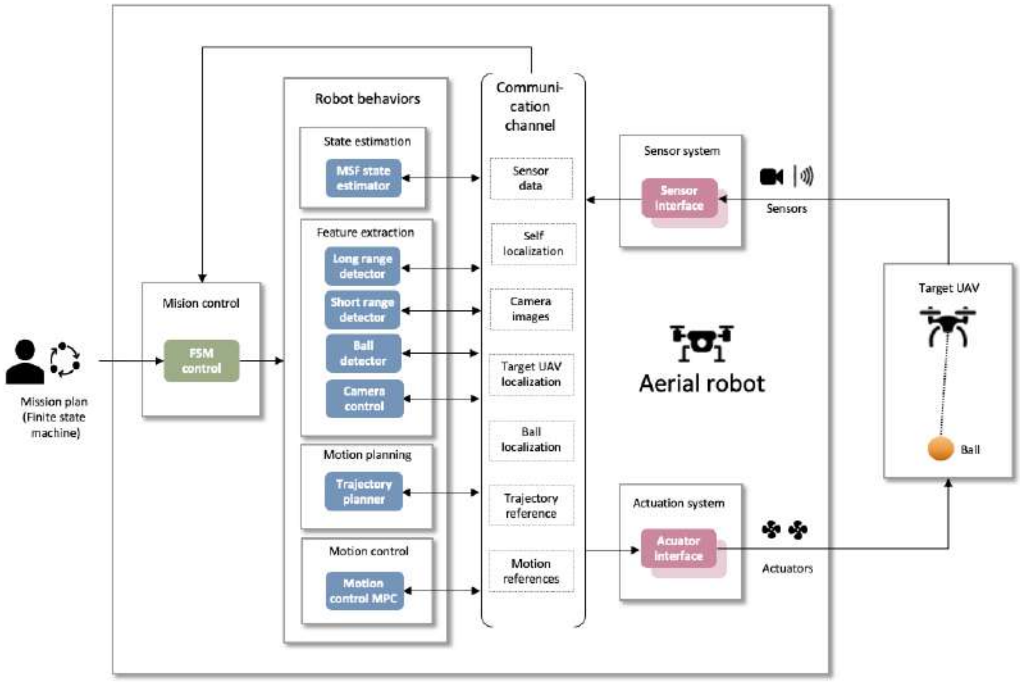


Figure 9. System architecture design used for Challenge 1 (based on the Aerostack software framework).

channel facilitates process interoperability and helps to reuse components across different types of aerial platforms. The channel is implemented with a set of ROS topics and ROS messages.

- *Robot Behaviors:* Robot behaviors implement the robot functional abilities including state estimation, feature extraction, motion control, and motion planning.
- *Mission Control:* Mission control executes a mission plan specified in a formal description. In this implementation, we use a finite state machine.

The next sections describe in more detail the components related to feature extraction, state estimation, motion planning, motion system, and mission control.

5.1. Feature Extraction System

A critical element required to achieve the proposed tasks is the capability of robustly detecting the target UAV and objects, as well as accurately estimating their location with respect to the aerial platform. The Feature Extraction System (FES) of the system architecture incorporates all of the perception software modules, which are based on computer vision and deep learning techniques. The FES can be divided into five base modules, which are used during different stages of the challenge: (i) the Long-Range UAV Detector, (ii) the Short-Range Detector, (iii) the Target Position Estimation, (iv) the Camera Gimbal Controller, and (v) the Interleaved Execution Policy.

5.1.1. Long-Range UAV Detector

As mentioned in Section 3, the area in which the challenge was carried out had dimensions of 100 m × 40 m. Therefore, in order to detect the target UAV in motion, with such a wide scope area, a long-range UAV detector module has been implemented. The long-range detector is in charge of the detection of moving objects when they appear small in the image plane, e.g., the case of a small UAV flying in a far distance from the camera (see Figure 10).

In these type of scenarios, where there are no semantics that can be exploited due to the small appearance of the object within the image plane, other techniques such as background motion

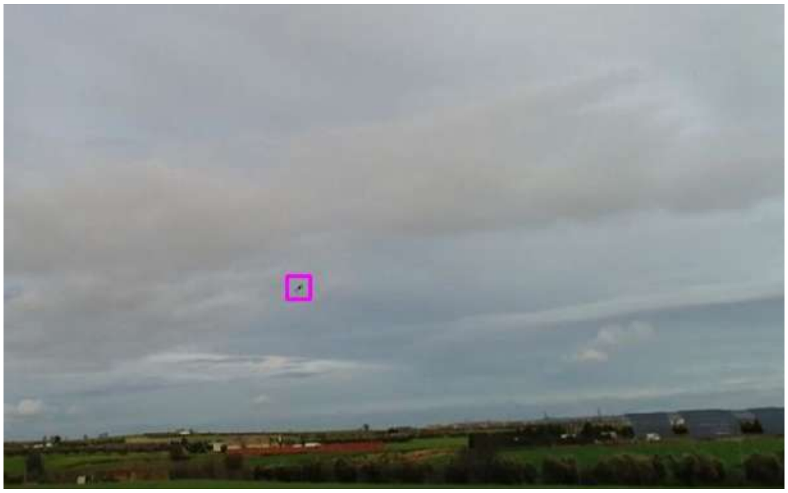


Figure 10. Example of a long-range UAV successful detection. The target UAV appears reduced within the image plane.

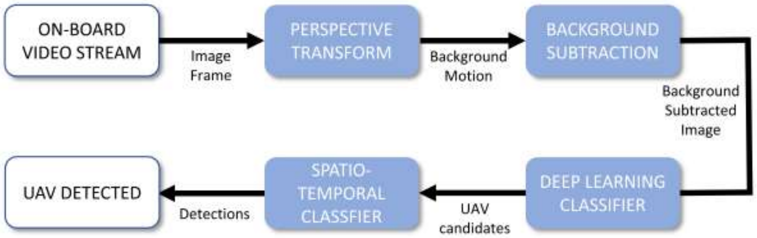


Figure 11. Diagram of the proposed method for the Far UAV detector in Challenge 1.

estimation and moving object detection can be incorporated (Li et al., 2016a). The processing pipeline is built on four steps, which are summarized as follows:

1. *Perspective Transform:* The image frames acquired from the onboard camera are used to estimate the background motion between frames. Considering a globally smooth motion with camera projection, the background motion can be calculated using a perspective transform model. For further details of the perspective transform model, refer to (Li et al., 2016a).
2. *Background Subtraction:* Since the background and the moving objects in the image can be assumed to have fairly different motion models, the moving objects in a far plane can be obtained by compensating the motion of the background. In this regard, with the pixel-wise background estimation model, the background can be subtracted from the original motion image.
3. *Deep Learning Classifier:* After subtracting the background from the original image, a deep learning classifier can be used to detect the candidates for moving objects in the highlighted patches of the background subtracted image.
4. *Spatiotemporal Classifier:* Finally, to find the local motion of the moving objects, a Lucas-Kanade optical flow algorithm has been applied to the classified candidates from the previous step, and hand-engineered spatiotemporal features can be extracted to identify actual UAVs in the image.

The architecture of the deep learning classifier is a standard Convolutional Neural Network (CNN) with two convolutional layers and five fully connected layers. The last fully connected layer has one linear unit in order to provide a classification of the target as UAV or background.

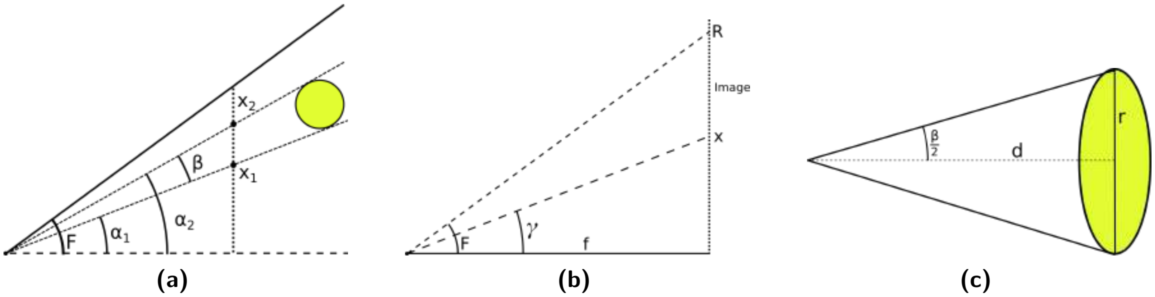


Figure 12. Diagram of the proposed method object position estimation. (a) An ellipsoid surrounding the object is considered, and the resulting position is calculated using the bounding box of the detection in image coordinates. The vertical line depicts the image plane. (b) Close-up diagram of image references. (c) Close-up diagram of distance estimation.

5.1.2. Short-Range Detector

The Long-Range Detector presented in the previous section has been used to find and approach the target UAV during the challenge. Nevertheless, the techniques used to detect objects in a far plane are not suitable for shorter distances. For object detection on close behavior (less than 15–20 m) a modified version of Tiny YOLO v3 (Redmon and Farhadi, 2018) has been used. Tiny YOLO v3 is an extremely fast fully convolutional one-shot object detector, which can process full images (at 720p resolution) to output a set of bounding boxes corresponding to the objects in the image along with a confidence value and corresponding class. The modification consists in changing the stride of the up-sample layer from 4 to 2, increasing the resolution of the feature maps in order to improve the detection of small objects, and as a consequence, increasing the detection range. Since Tiny YOLO v3 can be used as a multiclass detector, the same detector has been used to identify the target UAV, as well as objects of interest (UAV, ball, and balloons).

5.1.3. Target Position estimation

The target object 3-DOF position estimation (i.e., target UAV, ball, or balloon) is performed using a basic trigonometric calculation to estimate an approximate 3D position relative to the aerial vehicle. We define an ellipsoid around the target UAV of width and height r_w and r_h, respectively. We assume that the projection of the ellipsoid in the image plane corresponds with the bounding boxes provided by the detector. Letting x_1 and x_2 be the bottom and top coordinates of the bounding box, we can calculate the angles α_1 and α_2 [Figure 12(c), Equation (4)]. Then we can estimate the distance d using this height an the angular coordinate α_x. After that, we repeat this procedure using the width of the ellipsoid and bounding box, and we average both values in order to obtain the final 3D estimation (d, α_x, α_y) .

$$F = \frac{FOV}{2} \tag{1}$$

$$R = \frac{resolution}{2} \tag{2}$$

$$f = \frac{R}{tan(F)} \tag{3}$$

$$\gamma = tan^{-1}\left(\frac{x \times tan(F)}{R}\right) \tag{4}$$

$$\beta = \alpha_2 - \alpha_1 \tag{5}$$

$$\alpha_x = \frac{\alpha_1 + \alpha_2}{2} \tag{6}$$

$$d = \frac{r}{tan(\frac{\beta}{2})} \tag{7}$$

For the case of the Short-Range Detector (see Section 5.1.2), the stated trigonometric strategy was complemented by the Efficient Perspective-n-Point (EPnP) algorithm[4] [2] in order to reduce the variance of the estimation, given the high variability found in the bounding box detection within the image plane. On this basis, the four corners of the bounding box in the image are paired with the respective points in the world (centering the axes at the center of the item). Afterwards the plane that dissects the item (parallel to the image plane) is used, and the resulting 3D corners of a rectangle of the size of the object are matched to the detected bounding box 2D corners to acquire the translation to the camera frame of reference. Finally, a linear Kalman Filter (KF) is integrated in order to filter the raw position estimates and make them usable by other components in the architecture.

On the side of the Long-Range UAV Detector (see Section 5.1.1), although the position of the UAV within the image plane can be detected (a single point), we are unable to estimate the size of the object on the image plane. Thus, the distance of the UAV with respect to the camera frame of reference cannot be computed. In this scenario, the stated distance has been fixed to a constant value, and the estimated pose is used by other modules as a waypoint that indicates the direction in which we need to move to get closer to the target UAV.

5.1.4. Camera Gimbal Controller

This component of the FES is in charge of controlling the camera gimbal in order to provide the desired behavior in every stage of the mission. The controller scheme is composed of an image-based Proportional-Derivative (PD) angular velocity controller (pitch and yaw axes). The error in the image plane is translated with geometric projections into an error in angle in the camera frame of reference. Then, it is compensated by the action of this controller. This module receives two different types of modes where the FES can be: SHORT_RANGE_MODE or LONG_RANGE_MODE. Accordingly, the behavior changes as follows:

- *LONG_RANGE_MODE:* In this mode, the camera gimbal is controlled to hold a fixed upwards pitch angle of 10°. In this position, the camera gimbal is pointing towards an approximate direction of the sky in order first to maximize the detection probabilities of both Long-Range and Short-Range Detectors, and second to perform the grasping maneuver from a lower position with respect to the target object.
- *SHORT_RANGE_MODE:* In this mode, the camera gimbal is controlled to hold the detected target (UAV or ball) in the center of the image in order to maximize the number of frames in which the target object is being detected during the high-speed mission execution. The altitude of the UAV is being adjusted in parallel in order to maintain the UAV in a lower position with respect to the target ball or UAV. The control of the altitude involved a constant offset of altitude below the target UAV.

5.1.5. Interleaved Execution Policy

For the sake of efficient coordination, during the execution of a mission, the FES components share a policy to decide at which time slot each detector is executed. Figure 13 shows the execution flow of this policy. At the start of the mission, the Long-Range Detector is executed when an image frame is received (since the location of the target UAV is unknown and assumed at a far distance) and the amount of image frames is recorded using a counter (NUM_FRAMES). At this moment, the FES is considered to be in LONG_RANGE_MODE.

After the execution of the Long-Range UAV Detector, the modulus operation is applied to NUM_FRAMES with the divisor F_t (number of sporadic short-range detections in between long-range detections). As such, after every F_t frame, the Short-Range Detector is executed to evaluate whether a detection (either UAV or ball detection) is made. Once the number of consecutive frames

[4] https://github.com/cvlab-epfl/EPnP

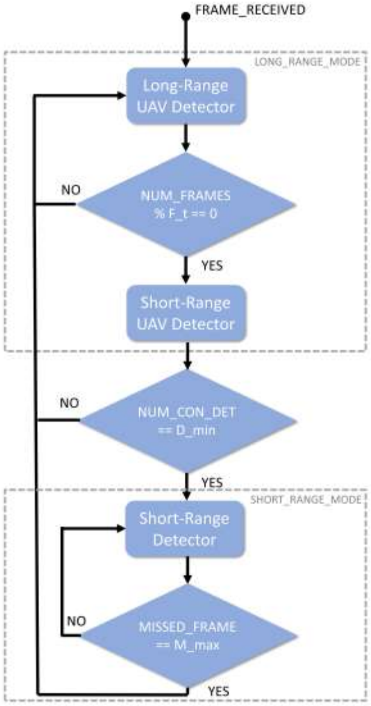


Figure 13. Feature extraction system component execution flowchart.

where a short-range detection is found (NUM_CON_DET) equals a minimum threshold (D_min), the Short-Range Detector is executed at every frame (FES changes to SHORT_RANGE_MODE) until the amount of frames without a detection (MISSED_FRAMES) reaches a maximum value of allowed misses (M_max). All of these constant values were experimentally defined to provide desired performance for a given experiment. It should be highlighted that the Gimbal Controller is being continuously executed, and it follows an internal policy to decide the reference angle. Also, the Position Estimator is executed every time a new detection is completed.

5.2. State Estimation System

Challenge 1 required our aerial robot to perform aggressive and precise maneuvers in order to intercept the object carried by the target aerial robot flying at high speeds (between 5 and 8 m/s). Thus, fast and robust state estimation was a key element of the autonomous system in order to perform the mission successfully.

The state estimation system of our architecture, shown in Figure 14, consists of the robust and modular Multi-Sensor Fusion (MSF) framework in ROS (Lepetit et al., 2009). The framework is based on an Extended Kalman Filter (EKF) and has the capability to integrate an unlimited number of sensors, as well as to compensate for any delayed sensor measurements. It can also estimate the bias in the sensor measurements and allow for online self-calibration of the sensor suite. The state prediction is performed by fusing the measurements received from the IMU sensors, which provide the measurements of linear accelerations and angular velocities in the three axes at a frequency of $\sim$ 400 Hz, and the position estimates about the x, y, and z axis provided by the RTK GPS sensor at $\sim$10 Hz. For additional robustness, the framework has been modified in order to integrate the linear velocities about the x, y, and z axis of the aerial robot estimated by an onboard Visual Inertial Odometry (VIO) algorithm running at $\sim$ 50 Hz and the altitude of the aerial robot estimated by a sonar sensor. The MSF is thus able to estimate the complete high-frequency odometry information of the aerial robot at $\sim$ 400 Hz.

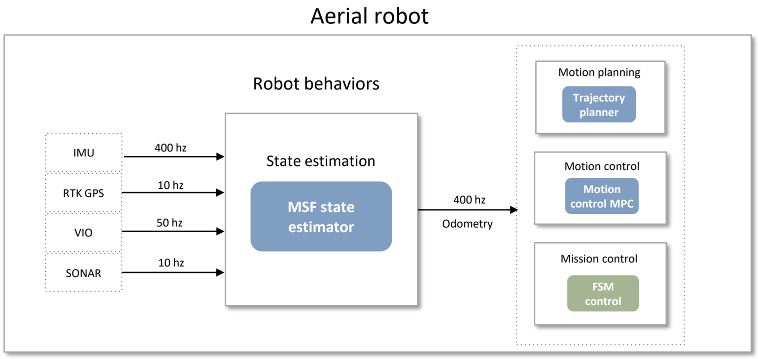


Figure 14. Block diagram for the Multi Sensor Fusion state estimation module.

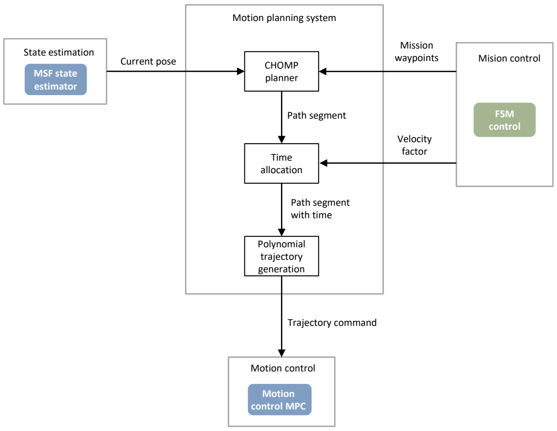


Figure 15. Schematic diagram for the motion planning system.

5.3. Motion Planning System

The Motion Planning System is in charge of defining a trajectory along which the aerial vehicle will have to navigate in every state of the mission in order to end up performing a precise catch of the ball. A modified CHOMP planner (Ratliff et al., 2009) is used for planning a collision-free path for catching, approaching, and following the target UAV and items. The waypoints and the velocity factor for the planner are given by the mission control system, the poses of the target drone and items estimation are given by the feature extraction system, and the initial point is taken as the current pose of the UAV, which is given by the MSF odometry measurements. A schematic diagram of the motion planning system can be seen in Figure 15.

First, the line segment between the initial and goal points is interpolated into finite points, where every point can be thought of as a state. The cost function of the CHOMP planner is composed of an obstacle objective function $f_o(\xi)$, a smooth objective function $f_s(\xi)$, the boundary objective functions $f_b(\xi)$ for X and Y, and the altitude objective functions $f_a(\xi)$ for Z. The main contribution of this equation is to generate a dynamically feasible collision avoidance trajectory, and within the defined altitude range that would be in the boundary of the flight area. In Equation (8), the smooth objective functions take charge of dynamic feasibility, and the obstacle objective functions take charge of obstacle avoidance. The boundary objective functions are used to penalize the trajectory if it is outside of the flight area, while the altitude objective functions are used to prevent the drone from flying to the ground or the ceil. The final objective function can be expressed as

$$f(\xi) = w_1 f_s(\xi) + w_2 f_o(\xi) + w_3 f_b(\xi) + w_4 f_a(\xi) \tag{8}$$

with

$$f_b(\xi) = c_{x_{max}}(\xi) + c_{x_{min}}(\xi) + c_{y_{max}}(\xi) + c_{y_{min}}(\xi) \tag{9}$$

$$f_a(\xi) = c_{z_{max}}(\xi) + c_{z_{min}}(\xi) \tag{10}$$

$$f_s(\xi) = \frac{1}{2}\int_0^1 \|\frac{\mathrm{d}}{\mathrm{d}s}\xi(s)\|^2 \mathrm{d}s \tag{11}$$

$$f_o(\xi) = \int_0^1 c_o(\xi(s))\|\frac{\mathrm{d}}{\mathrm{d}s}\xi(s)\|\mathrm{d}s \tag{12}$$

where w_1, w_2, w_3, and w_4 are the weights for each one of the objective functions, ξ is the path, and $\xi(s)$ is a function mapping the path length s to the UAV configurations. In the real flight, w_1, w_2, w_3, and w_4 are set as 1, 5, 10, and 15, respectively.

Equation (10) are taken as examples to describe the boundary objective functions and the altitude objective functions. The detail about $c_{z_{max}}$ and $c_{z_{min}}$ in Equation (10) can be seen from Equations (13) and (14). Z_{max} and Z_{min} in Equations (13) and (14) mean the maximum and minimum altitude the drone can reach, while $Z(\xi(s))$ is the current altitude of the drone. $c_{x_{max}}$, $c_{x_{min}}$, $c_{y_{max}}$, and $c_{y_{min}}$ in Equation (9) are similar to $c_{z_{max}}$ and $c_{z_{min}}$. The only difference is that we need to replace the Z in the formula with X or Y,

$$c_{z_{max}}(\xi(s)) = \begin{cases} (Z(\xi(s)) - Z_{max})^2, & Z(\xi(s)) \geq Z_{max} \\ 0, & Z(\xi(s)) < Z_{max} \end{cases} \tag{13}$$

$$c_{z_{min}}(\xi(s)) = \begin{cases} 0, & Z(\xi(s)) > Z_{min} \\ (Z(\xi(s)) - Z_{min})^2, & Z(\xi(s)) \leq Z_{min} \end{cases} \tag{14}$$

Given that the target UAV is considered as a cylinder obstacle, in order to avoid planning paths above or below the target, the obstacle cost function parameters are expressed as

$$c_o(\xi(s)) = \begin{cases} \frac{r(\xi(s))}{3}(1 - \frac{dist(\xi(s))}{r(\xi(s))})^3, & dist(\xi(s)) \leq r(\xi(s)) \\ 0, & dist(\xi(s)) > r(\xi(s)) \end{cases} \tag{15}$$

$$r(\xi(s)) = \frac{R}{\cos(\arctan(\frac{dist_z(\xi(s),c)}{dist(\xi(s)}))}. \tag{16}$$

In Equations (15) and (16), the distance from the current position of the UAV to the center of the cylinder obstacle is denoted by $dist(\xi(s))$, R is the radius of the obstacle, c is the position of the center of the obstacle, and $dist_z(\xi(s), c)$ is the altitude distance from the current position of the UAV to the center of the obstacle. The pose and radius estimations of the target UAV and item are obtained from the feature extract system. $c_o(\xi(s))$ is set as 0 when the target drone pose estimation is not given.

With these functions, the cost functional gradients of every state are calculated and a gradient descent approach is used to do path optimization. Afterwards, an ROS trajectory generation package (Richter et al., 2016) is used to generate the desired trajectory to the target UAV or item, given the path of the CHOMP planner. To generate a trajectory tracking strategy instead of path following, different velocities for certain stages of the mission were defined. With the following function, the segment times were modified:

$$T_{seg}(\xi_{k-1}, \xi_k) = \frac{d \times (1 + f_{maxv} \times v_{max})}{2 \times v_{max} \times f_v \times a_{max} \times e^{\frac{-d}{2 \times v_{max}}}} \tag{17}$$

where $T_{seg}(\xi_{k-1}, \xi_k)$ is the segment time between ξ_{k-1} and ξ_k, d is the distance from ξ_{k-1} to ξ_k, f_{maxv} is the maximum velocity factor, f_v is the velocity factor that is given by the mission control system, and v_{max} and a_{max} are the maximum velocity and maximum acceleration the drone can reach, respectively. Different values of velocity factors are given based on the maneuvers of fast flight with high tracking errors, slow flight for precise maneuver, and target drone searching maneuver. In

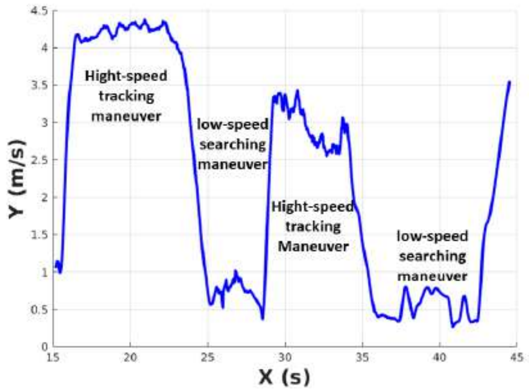


Figure 16. The velocity-changing during the high-speed tracking maneuvers and low-speed searching maneuvers. The velocity factors are set as 4 for high-speed tracking and 1 for low-speed searching, respectively.

the real flight, f_{maxv} is set as 6.5 while f_v is set as 4 for fast-tracking and 1 for flying slow to search the target and to do the precise maneuvers (searching and ball catching). The maximum velocity and maximum acceleration are set as 15 m/s and 20 m/s^2. Figure 16 shows the velocity-changing process when switching between fast-tracking and precise maneuvers. The velocities in the graph are the sum of squares of X, Y, and Z three-axis speeds.

5.4. Motion Control System

The Motion Control System is in charge of generating and communicating the necessary actuator commands for the UAV to perform the desired trajectories, generated by the aforementioned planning system. Given the complex requirements of Challenge 1, reliable trajectory tracking control is a key element of the overall system. To address this problem, a Nonlinear Model Predictive Controller (NMPC) (Kamel et al., 2017b) has been implemented. The main advantage of the MPC is the fact that it calculates the optimal control actions taking into account both the actual and future points of the trajectory within its prediction horizon and the constraints of the UAV. This is achieved by optimizing a finite time-horizon cost function, but only implementing the current control action and then optimizing again in the next iteration.

A NMPC has been implemented based on the full system dynamics of the DJI Matrice 210 v2 RTK instead of a Linear MPC, which only considers a linearized model, because UAV behavior is better described by a set of nonlinear differential equations in order to capture the aerodynamic and coupling effects. As demonstrated in (Kamel et al., 2017a), the NMPC presents a better disturbance rejection capability, step response, tracking performance, and computational effort than LMPC. Thus it ensures precise trajectory tracking in realistic environments where the flight performance may be compromised by external disturbances. To enhance the predictive capability of our NMPC, an online EKF disturbance observer that compensates for changes in mass of the UAV and wind has been designed.

Since the DJI Matrice 210 v2 RTK does not support directly sending angular rate commands, the MPC controller outputs are smooth pitch and roll commands for horizontal motion and altitude. Although the position and heading of the UAV can be controlled simultaneously by the MPC, these DOFs have been decoupled. Thus, for simplicity, a proportional control scheme for the yaw-axis has been used.

To configure our NMPC, all the constraints in velocities and accelerations of the UAV are indicated and the operating frequency of the control is set at 100 Hz. This control signal is sent to the low level control that is done by the DJI onboard computer at 400 Hz. To reduce the computational load of the NMPC, the internal prediction frequency is fixed at 10 Hz following the strategy presented in (Kamel et al., 2017b), thus with a prediction horizon of only 20 steps, the trajectory of the next

2 s is optimized in the controller. As for the Q and R matrices of the NMPC cost function, we set a more aggressive controller for the FOLLOW_SHORT_RANGE and CATCH_BALL states, setting a more robust control for the rest of the states. In this way, the use of large accelerations in the most critical part of the mission is enhanced, and the consumption through smaller accelerations when the UAV is far from the target is reduced.

5.5. Mission Control System

In general, a robot mission execution can be automatically controlled with the help of a representation such as Finite State Machines (FSM), hierarchical finite state machines, behavior trees, etc. In this work, a solution based on a FSM for the architecture designed for Challenge 1 has been implemented, which is adequate according to the complexity of the problem, since it can be represented with a small number of states and events. This representation was very useful to experiment with low programming effort with different versions of the mission control.

The implemented FSM uses a number of states in order to control different parts of the system. For example, some states activate and deactivate the Long-Range UAV Detector and Short-Range Detector. This is needed because the computation load is limited and all detectors cannot run in real time at the same time. States can also establish the required UAV speed. The UAV may need to increase the velocity in order to reach the target UAV or the ball. However, when there is not a target detected, the UAV uses low speeds to maximize the performance of the detectors. In addition, there are also states to establish navigation goals to move the UAV to different positions (e.g., to search the other drone or to reach a target).

The following list summarizes the most important states used in the FSM:

- *START_STATE*: The UAV is landed. In this state, the GPS, RTK, and cameras are initiated.
- *SEARCH*: The UAV does a predefined trajectory that was designed to maximize the probability to detect the target UAV. The trajectory is in the form of a semiellipse, very close to the long axis border and in the middle of this axis. This is because it is known that the lemniscate will cover 80% of the arena, therefore the center of it should be situated in the center part of the arena, and this trajectory put this area in the FOV of the camera. Also, the trajectory has a low altitude, so the camera is pointing to the sky. This maximizes the accuracy of the detectors because the background is uniform and there are fewer objects that can move. Finally, during this state, the maneuvers are at very low speed and acceleration order to get sharp images. In this state, both UAV detectors are active
- *FOLLOW_LONG_RANGE*: Every time there is a new long-range detection, the UAV moves at a high speed to a point 5 m towards the detection in the XY plane, while the altitude of the UAV is kept at a constant and low value in order to maximize the performance of the detectors.
- *FOLLOW_SHORT_RANGE*: The UAV moves at a high speed towards the detection, maintaining 4 m from the target. In this state, the Long-Range UAV Detector is disabled so the Short-Range Detector take all the computation power.
- *CATCH_BALL*: The UAV moves towards the ball detection trying to align the ball position with the gripper. Once the ball goes out of the camera field of view, it continues following the planned trajectory. In this state, the trajectory planner generates the required movements and the Long-Range UAV Detector is disabled so that the Short-Range Detector takes all the computation power.
- *LAND*: The UAV lands in a given position and the mission ends.

Table 1 shows a list of example events that were used in the FSM as transitions between states. Some of these events are related to the perception performance considering the number of consecutive image frames in which an object is detected. For example, a short-range detection is generated when the target UAV is recognized in two frames from the last four consecutive frames. Figure 17 illustrates part of the transitions in the FSM.

Table 1. Example events used in the finite state machine.

Event	Description
START_MISSION	The human operator gives the order to the UAV to start the mission
LONG_RANGE_UAV_DETECTED	There are 3 detections on 5 consecutive frames from the Long-Range Detector.
SHORT_RANGE_UAV_DETECTED	There are 2 detections on 4 consecutive frames from the Short-Range Detector.
BALL_DETECTED	The ball is detected.
DETECTION_LOST	No new detections in 5 consecutive frames
SUCCESSFUL_CATCH	Ball is detected by the laser sensor inside gripper.

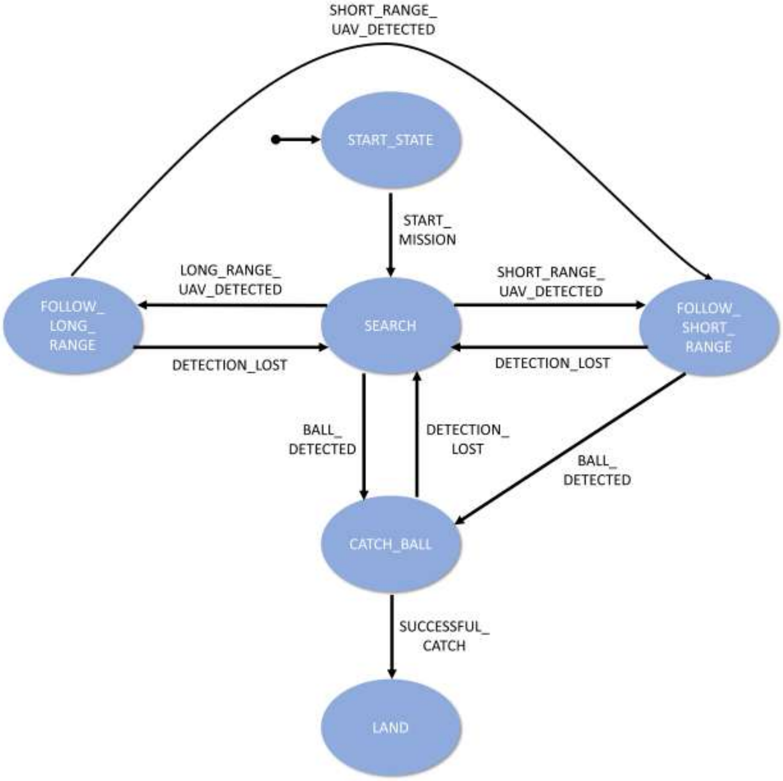


Figure 17. View of the states and transitions of the finite state machine.

6. Experiments and Results

In this section, we discuss several experimental tests that have been performed to test and validate all of the proposed solutions that have been integrated in the system pipelines, and the major results achieved for Challenge 1 are presented.

6.1. Experimental Setup

Since the challenge was to be performed in an outdoor environment, an open airfield provided a workspace large enough (larger than 100 m × 40 m × 20 m) to test several scenarios that could be encountered during the real competition trials. The initial tests performed were focused on gradually validating the individual components of the system architecture, presented in Section 5. Nevertheless, the experiments and tests provided in this section correspond to the report of a global mission execution.

Table 2. Data augmentation parameters used for training the Short-Range Detector using the Darknet framework.

Parameter	General training	Fine tuning
Saturation	2.0	1.5
Exposure	2.0	1.5
Hue	0.5	0.1

To perform the experiments, a DJI Matrice 100 is commanded to perform a lemniscate reference trajectory (approximately 80% of the total allowed area for the challenge) at different velocities and at 15 m reference altitude, replicating the challenge environment. The DJI Matrice 100 carried a yellow balloon, tethered to a carbon fiber stick of approximately 1.5 m in length.

All the computation for the DJI Matrice 210 v2 RTK is performed onboard with the hardware setup described in Section 4, and the mission is fully autonomous. A safety pilot has been constantly aware of the state of the mission execution in order to avoid risks in case of failure.

6.2. Training and Test of the Feature Extractors

There are two main feature extractors, which are part of the FES, and they are in charge of aiding both the Long-Range UAV Detector and the Short-Range Detector.

6.2.1. Training Methodology

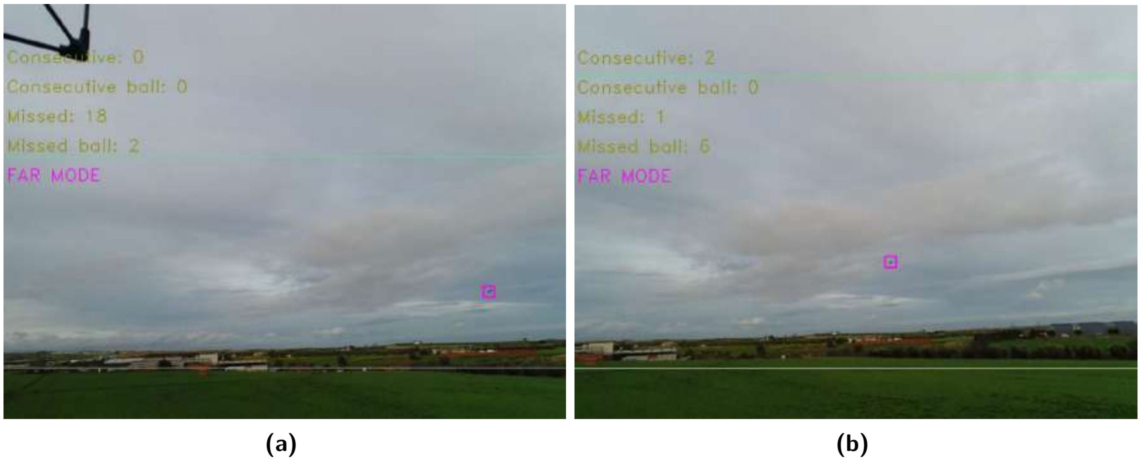
The training methodology is different in every detector case, despite being based on CNN architectures. The main aspects are as follows:

- *Long-Range UAV Detector:* This component is built on several processing steps. One of them involves the generation of a CNN architecture for UAV/Background classification. This model has been trained with a binary cross-entropy loss and Adam optimizer with a learning rate of 0.001. Every batch during the training phase has been composed of a 50%–50% balanced UAV-background patch sample in order to stabilize training due to the high amount of background samples in the original dataset. The batch size has been 1024, and dropout has been included in every fully connected layer. The dataset used for training has been the Purdue UAV Dataset.[5]
- *Short-Range Detector:* This component consists on a single neural network that performs the detection on a single inference pass. The training procedure consists of three steps:
  1. *Imagenet pretraining:* To get rich features and lower the training time, we use a pretrained darknet53 (Redmon and Farhadi, 2018) on Imagenet. We copy the weights from darknet53 to the matching layers of tiny YOLOv3.
  2. *General detector training:* The network is trained using our baseline dataset. The aim of this dataset is to obtain a general detector that can detect the objects of interest (UAVs, balls, and balloons) in many different situations and environments. The dataset contains almost 4000 images of different models of UAVs, and 1000 images of balloons and balls gathered online that we labeled manually. For training the network, we use the Darknet[6] framework. We trained the network during 10 000 steps using an Adam optimizer, with a learning rate of 0.001, lowering it to 10% and 1% at 8000 and 9000 steps, respectively. We used 20% of the training dataset for validation purposes and early stopping. The data augmentation parameters are shown in Table 2.
  3. *Domain-specific fine tuning:* Once the detector is performing properly, we perform a fine tuning step (lowering the initial learning rate to 0.0001) using a second dataset from the final environment. We recorded this dataset using the platform camera during the real flights using the currents objects to detect (DJI f450 and DJI m100 UAVs, green and white

[5] https://engineering.purdue.edu/~bouman/UAV_Dataset/
[6] https://github.com/alexeyab/darknet

Table 3. Metrics for the CNN classifier of the Long-Range UAV Detector. Precision, Recall, F-score, and average Frequency (Hz) results are provided.

Precision	0.477
Recall	0.497
F1-score	0.487
Frequency (Hz)	15.2



(a) **(b)**

Figure 18. Target UAV detection results from the Long-Range UAV Detector module. The UAV detections are depicted in magenta.

balloons and a yellow ball) in order to lower the domain gap between our general dataset and the real environment. Also, in order to improve the Intersection Over Union (IOU) of the detections, we increase the IOU threshold from the YOLO loss function from 0.7 to 0.9, resulting in more accurate bounding boxes and improving the pose estimation in subsequent steps at a marginal increase of missed detections. The data augmentation parameters are shown in Table 2.

6.2.2. Feature extractors results

The FES modules are considered key elements in this work, and they are required to successfully accomplish the tasks proposed in the challenge. As such, extensive testing was carried out to evaluate the different components of the FES. In Figure 18, successful detections of the target UAV moving in a far distance from the camera are depicted.

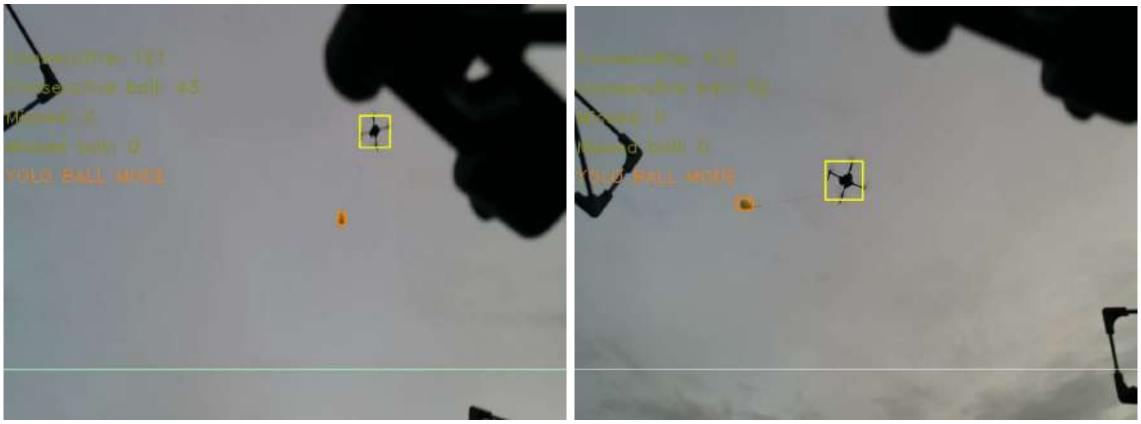
The aerial vehicle was initially flying to a distance of ∼50 m from the target UAV. As can be seen in Figures 18(a) and 18(b), at this distance the Far detector was able to successfully detect the target UAV in the image frame and an approximate pose was obtained, where the magenta Region-Of-Interest (ROI) indicates the detection of the target UAV in the image frame.

Furthermore, in order to formally evaluate the performance of the CNN classifier of the Long-Range UAV Detector component, average Frequency (Hz), Precision, Recall, and F-score are provided in Table 3. The provided frequency value corresponds to the complete Long-Range UAV Detection component, that is, all the processing steps described in Section 5.1.1.

In Figure 19, successful detection of both the target UAV and the object (ball), in a closer distance (∼10 m), is shown. The Short-Range Detector was able to detect and identify both the target UAV and the object (ball) tethered to the vehicle. In these tests, the aerial vehicle was manually flown towards and away from the target UAV, where the FES components would activate depending on the detector execution flowchart presented in Section 5.1.5.

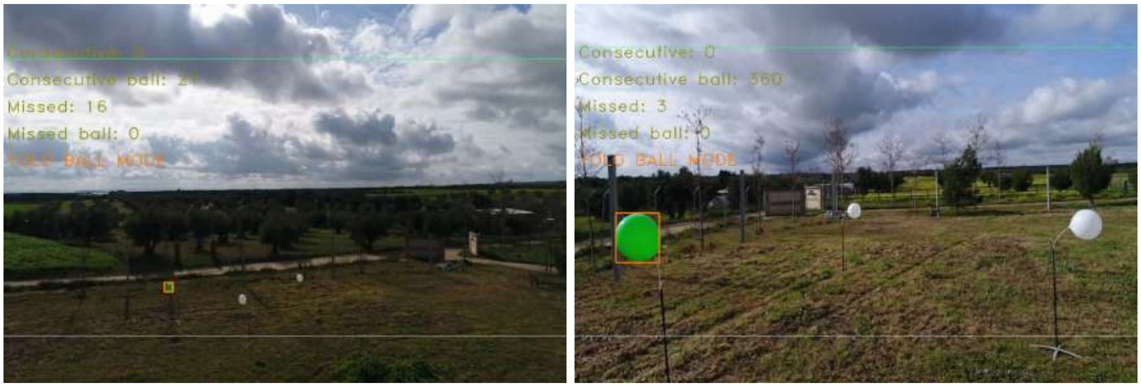
Table 4. Metrics of the final Short-Range Detector for a 0.25 confidence threshold and 0.5 IOU threshold.

AP (UAV)	0.893
AP (ball)	0.922
AP (balloon)	0.924
mAP	0.912
Average IOU	0.573
Precision	0.9
Recall	0.81
F1-score	0.86
Frequency (Hz)	22.54



(a) (b)

Figure 19. Target UAV and Item detection results from the Close and Item detector module.



(a) (b)

Figure 20. Balloon item detection using the Close and Item detector module.

As additional results, a complementary task for the challenge was required, and involved the autonomous bursting of ground-tethered balloons. As shown in Figure 20, the balloons were properly detected by the Short-Range Detector.

In Table 4 we report common object detection metrics for the Short-Range Detector. The AP (Average Precision) is calculated using a 0.5 IOU threshold, while the precision and recall are calculated using a 0.25 confidence threshold.

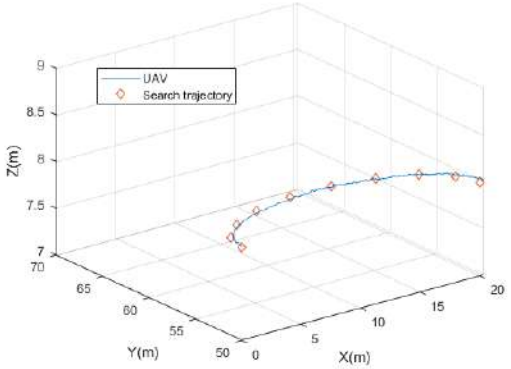
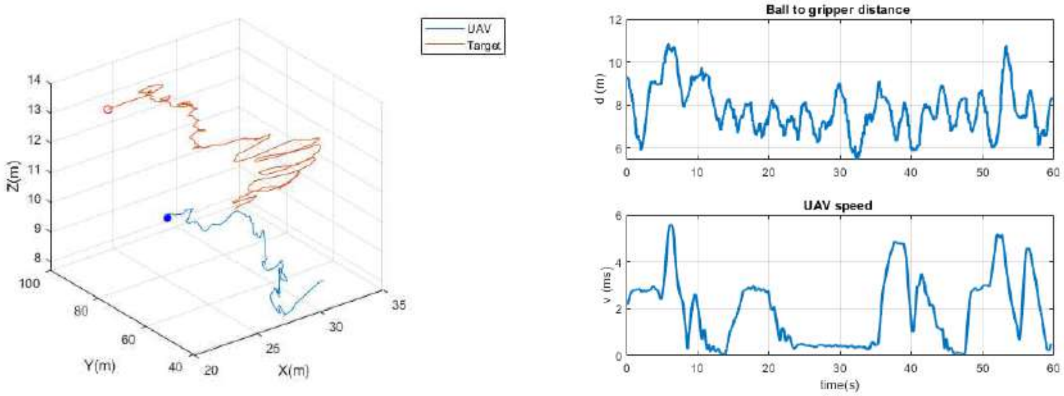


Figure 21. Trajectory tracking results for the SEARCH state. This is a predefined trajectory where the follower UAV tries to detect in LONG_RANGE or SHORT_RANGE the target UAV.



(a) **(b)**

Figure 22. Trajectory tracking results for the FOLLOW SHORT RANGE state. (a) Trajectories of the target UAV and our proposed UAV. (b) Distance between the target UAV and our proposed UAV and velocity of our proposed UAV.

6.3. Target Search, Follow and Approach Results

The technical specifications for Challenge 1 stated that the target UAV was flying in a lemniscate curve at a maximum velocity of 8 m/s. Thus, in order to perform the SEARCH maneuver, the Motion Planning was used to command a trajectory to the Motion Control from a specified series of waypoints following a semiellipsoidal shape in the x-y plane (see Figure 21).

The results of the trajectory tracking test are presented in Figure 21. As previously mentioned, the MSF module from the FES is in charge of fusing the measurements from the RTK, VIO, and IMU sensors to output an estimation of the pose of the aerial vehicle. Using the fused measurements from the MSF module, the MPC module of the Motion Control was able to perform the desired flight trajectories even in nonideal conditions compensating for wind gusts of up to 15 m/s.

In Figure 22, the FOLLOW SHORT RANGE stage is depicted. The figure shows the attempted evasion of the target UAV [see Figure 22(a)] and the maneuver of the proposed UAV to maintain a distance of about 8 m at all moments by applying an average speed of 2 m/s [see Figure 22(b)]. It should be noted that at all times a security distance is maintained with respect to the target UAV when the distance is less than 6 m, reducing the speed significantly (instants of time from 25 to 35 s).

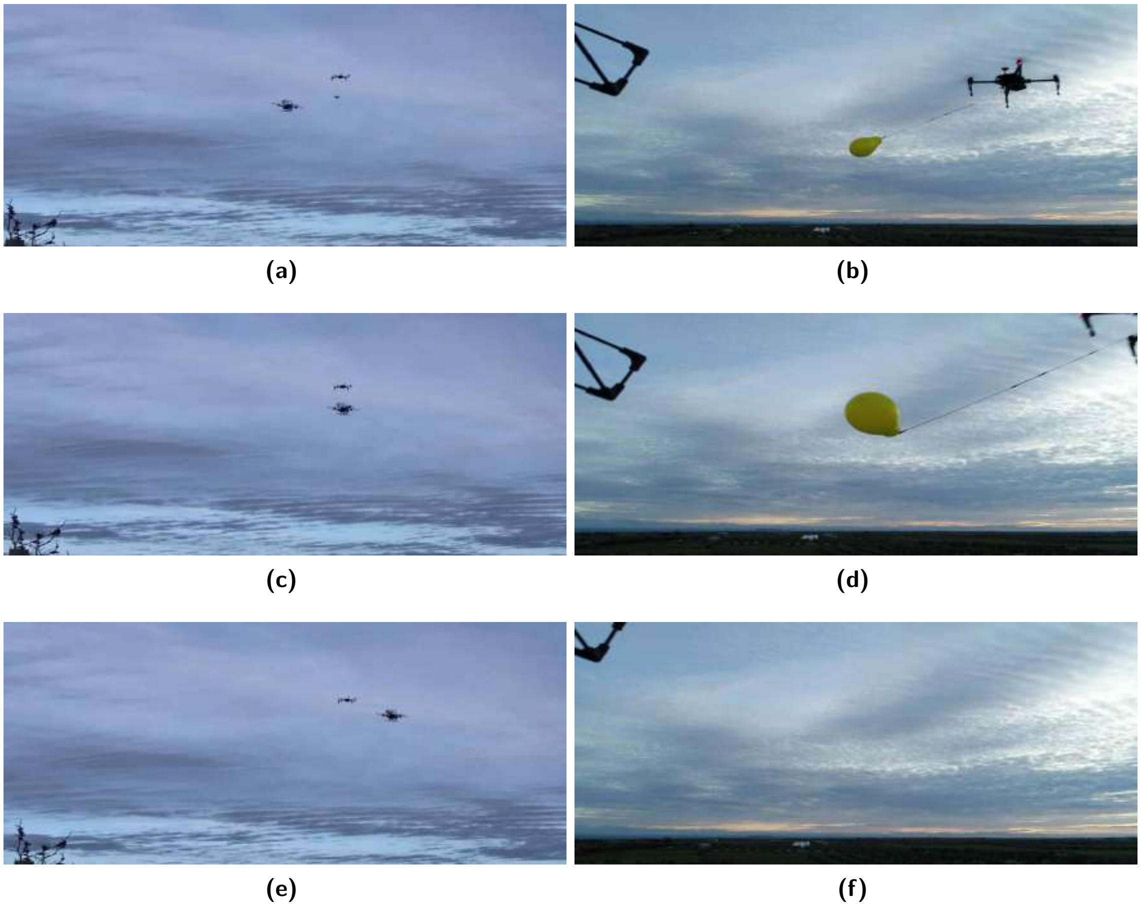
6.4. Grasping Maneuver Results

The most challenging task of Challenge 1 is to grasp the item that is tethered to the target UAV in motion. Therefore, a test scenario was implemented to validate the performance of the overall proposed strategy in a fully autonomous real-world grasping maneuver mission.

In this scenario, a DJI Matrice M100 UAV was used as a target UAV which has a yellow ball (a balloon for safety) tethered to the end of a 2 m long carbon fiber rod. As in the actual mission, the aerial platform is required to search, detect, follow, and interact with the target UAV in order to retrieve the desired item.

First, the target UAV was flown to a distance of approximately 50 m from the aerial platform in flight. Afterwards, the FSM implemented in the Mission Control (presented in Section 5.5) was activated to perform the fully autonomous mission. The first states that were executed performed a grid search of the target UAV in the open airspace. Once the Long-Range UAV Detector achieved the minimum number of detections necessary, the aerial vehicle proceeded to move towards the target UAV in order to acquire a lock with the Short-Range Detector.

Once the Short-Range Detector obtained a desired number of consecutive detections, the estimated pose of the tethered object was continuously commanded to the Motion Planning system as to generate the desired trajectories needed to intersect the object with the gripping mechanism. The results of the grasping maneuver test performed with the proposed system strategy are presented in Figure 23. Figures 23(a)–23(f) show a sequence of images of the aerial vehicle performing the


(a)

(b)

(c)

(d)

(e)

(f)

Figure 23. Results from the grasping maneuver test. (a)(c)(e) Sequence view of the aerial vehicle grasping maneuver. (b)(d)(f) Images from the onboard camera used for detection of the target UAV and item.

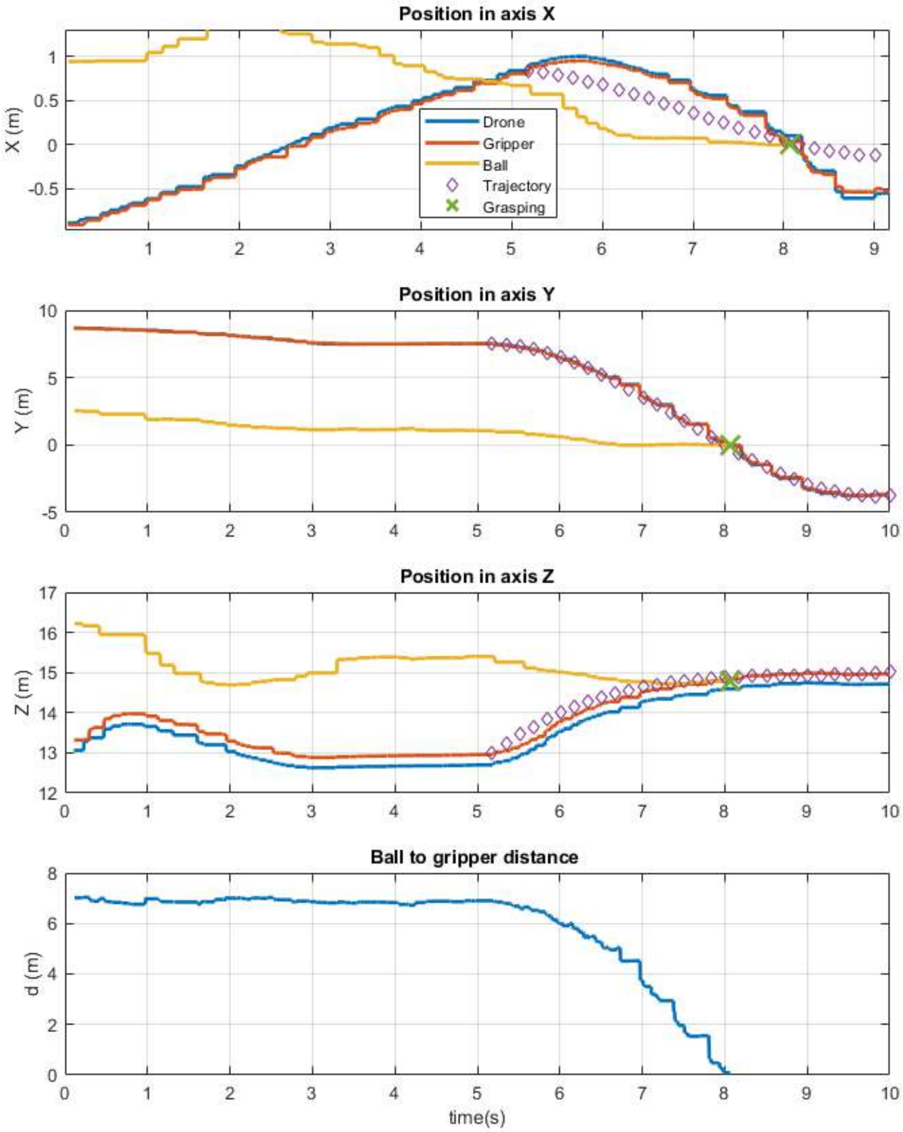


Figure 24. Trajectories of our UAV, gripper, and ball in x, y, and z axes for the grasping maneuver test during a real flight. At the beginning of the test, our UAV is following the ball keeping a distance of approximately 7 m. At 5.1 s our UAV starts the grasping manoeuver, approximately 3 s later the distance between the gripper and the ball is reduced to 0, meaning that the grasping maneuver was successful.

item grasping maneuver during the test flight. Figure 24 shows the trajectory of the aerial vehicle and pose of the ball object. As can be seen, the overall autonomous system strategy proposed and implemented for the grasping task of Challenge 1 enables the item to be retrieved from the target UAV successfully. Figure 23 shows how the FOLLOW state is maintained until the moment of time 5.2 s. In that instant, CATCH BALL state is activated and the trajectory is planned for the gripper to catch the ball at the time instant 8 s.

7. Discussion

An extensive validation of the proposed hardware and software UAV platform has been thoroughly carried out. All of the components have been detailed, and a global mission has been performed and validated, with a separate explanation of the results. From a global perspective, the proposed UAV platform has shown outstanding results, being able to grasp a tethered ball from a moving UAV at approximately 6 m/s and resulting in the 3rd position for the Grand Challenge in the 2020 MBZIRC.

As explained in previous sections, the complete approach has been mainly based on the RGB camera gimbal to support the global mission. In this regard, the FES components have provided successful results. The test metrics provided in Tables 3 and 4 indicate a proper performance of the components. Also, the results obtained in Sections 6.2, 6.3, and 6.4 demonstrate that the perception components developed for the FES are capable of detecting the target UAV and objects with varying lighting conditions, in addition to validating the FES component execution and pose estimation modules. It should be noted that every image detection led to a consequent position estimation, which is directly forwarded to the rest of the modules in order to complete the execution of the mission. Although the FES components provided adequate performance, the RGB camera gimbal resolution limited the behavior. For instance, an optical-zoom RGB camera gimbal can provide a wider versatility.

Regarding the search, follow, and approach results, all of the components involved showed an adequate behavior, being run onboard with tight computational resources. As shown in Figures 21 and 22, the target UAV has been located during the SEARCH stage and latterly followed during the FOLLOW and APPROACH states. Although a safety distance has been maintained during its execution, the target UAV has been continuously followed during the whole trajectory at high speeds. The target UAV has been followed with peak velocities of approximately 6 m/s, with onboard and vision-based components. In the different trials, the target UAV can be missed and the FOLLOW and APPROACH can be switched to the SEARCH state. This is mostly due to errors in detection or position estimation, and/or to abrupt changes in the target trajectory, which hardens the motion control.

Finally, with regard to the grasping maneuver, the tethered ball was successfully caught. Figure 23 shows the proper approximation of the proposed UAV, decreasing the distance in every axis. There is a period of time where the RGB camera gimbal is not able to perceive the ball due to the relative positions of the ball and camera (close to the end of the maneuver). At this stage, the position is being actively estimated in order to enable the complete catching maneuver. As shown, the distance to the ball is decreased to approximately zero, and the ball is properly grasped at high speeds and in a continuous trajectory trend. The grasping maneuver can fail in a trial, mostly due to detections or position estimation errors and/or sudden movements of the target ball, which complicates the maneuver.

8. Conclusions

In this work, the hardware and software components that were implemented as the proposed strategies to solve Challenge 1 of the 2020 Mohamed Bin Zayed International Robotics Challenge (MBZIRC) have been detailed. The team has achieved 3rd position for the Grand Challenge. The proposed challenges presented complex environments and scenarios, which were required to be completed using fully autonomous robotic systems. As such, a diverse knowledge of various fields has been necessary to implement efficient and robust solutions.

Challenge 1 has been solved by means of a complete UAV platform with custom software and gripper actuator designs. All the software components were launched onboard with tight computational budgets and primarily with vision-based techniques. The experiments have been carried out in an outdoor space, replicating the competition environment and constraints. The whole challenge mission has been successfully executed and detailed, providing adequate results for every component of the hardware and software architecture, and being able to grasp a tethered ball from a moving UAV at approximately 6 m/s.

The future work to be performed entails the improvement of the vision components in order to provide faster and more accurate estimations of the targets. The position estimator of the FES component can be improved to decrease the position error, and the detectors can be optimized to increase their frequency. Additionally, the weight has to be drastically reduced in order to enable maneuvers at even higher speeds. Different gripper designs and coupled control strategies will be explored in future work.

Acknowledgments

This work has been supported by the Spanish Ministry of Science, Innovation and Universities (COMCISE RTI2018-100847-B-C21, MCIU/AEI/FEDER, UE) and the Khalifa University under Contract No. 2020-MBZIRC-10. The research of two authors is also supported by the Chinese Scholarship Council and the Regional Ministry of Education, Youth and Sport of the Community of Madrid and the European Social Fund, respectively.

ORCID

Alejandro Rodriguez-Ramos https://orcid.org/0000-0002-3257-4602
Adrian Alvarez-Fernandez https://orcid.org/0000-0003-3590-7983
Hriday Bavle https://orcid.org/0000-0002-1732-0647
Javier Rodriguez-Vazquez https://orcid.org/0000-0003-0305-7806
Liang Lu https://orcid.org/0000-0002-8078-8760
Miguel Fernandez-Cortizas https://orcid.org/0000-0002-3822-075X
Ramon A. Suarez Fernandez https://orcid.org/0000-0003-4102-5899
Carlos Santos https://orcid.org/0000-0001-6471-5310
Martin Molina https://orcid.org/0000-0001-7145-1974
Luis Merino https://orcid.org/0000-0003-4927-8647
Fernando Caballero https://orcid.org/0000-0001-8869-2846
Pascual Campoy https://orcid.org/0000-0002-9894-2009

References

Baca, T., Stepan, P., & Saska, M. (2017). Autonomous landing on a moving car with unmanned aerial vehicle. In *2017 European Conference on Mobile Robots (ECMR)*, pp. 1–6. IEEE.

Bachrach, A. (2021). Skydio autonomy engine: Enabling the next generation of autonomous flight. In *2021 IEEE Hot Chips 33 Symposium (HCS)*, pages 1–43. IEEE Computer Society.

Bähnemann, R., Pantic, M., Popović, M., Schindler, D., Tranzatto, M., Kamel, M., Grimm, M., Widauer, J., Siegwart, R., & Nieto, J. (2019). The eth-mav team in the mbz international robotics challenge. *Journal of Field Robotics*, 36(1):78–103.

Bavle, H., Manthe, S., De La Puente, P., Rodriguez-Ramos, A., Sampedro, C., and Campoy, P. (2018). Stereo visual odometry and semantics based localization of aerial robots in indoor environments. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1018–1023. IEEE.

Beul, M., Houben, S., Nieuwenhuisen, M., and Behnke, S. (2017). Fast autonomous landing on a moving target at mbzirc. In *2017 European Conference on Mobile Robots (ECMR)*, pp. 1–6. IEEE.

Beul, M., Nieuwenhuisen, M., Quenzel, J., Rosu, R. A., Horn, J., Pavlichenko, D., Houben, S., and Behnke, S. (2019). Team nimbro at mbzirc 2017: Fast landing on a moving target and treasure hunting with a team of micro aerial vehicles. *Journal of Field Robotics*, 36(1):204–229.

Bodie, K., Brunner, M., Pantic, M., Walser, S., Pfändler, P., Angst, U., Siegwart, R., and Nieto, J. (2019). An omnidirectional aerial manipulation platform for contact-based inspection. *arXiv:1905.03502.*

Cantelli, L., Guastella, D., Melita, C., Muscato, G., Battiato, S., D'urso, F., Farinella, G., Ortis, A., and Santoro, C. (2018). Autonomous landing of a uav on a moving vehicle for the mbzirc. In *Human-Centric Robotics: Proceedings of CLAWAR 2017: 20th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*, pp. 197–204. World Scientific.

Choi, H. and Kim, Y. (2014). Uav guidance using a monocular-vision sensor for aerial target tracking. *Control Engineering Practice*, 22:10–19.

Do, T., Carrillo-Arce, L. C., and Roumeliotis, S. I. (2019). High-speed autonomous quadrotor navigation through visual and inertial paths. *The International Journal of Robotics Research*, 38(4):486–504.

Falanga, D., Mueggler, E., Faessler, M., and Scaramuzza, D. (2017). Aggressive quadrotor flight through narrow gaps with onboard sensing and computing using active vision. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5774–5781. IEEE.

Fernandez, R. A. S., Sanchez-Lopez, J. L., Sampedro, C., Bavle, H., Molina, M., and Campoy, P. (2016). Natural user interfaces for human-drone multi-modal interaction. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1013–1022. IEEE.

Florence, P., Carter, J., and Tedrake, R. (2020). Integrated perception and control at high speed: Evaluating collision avoidance maneuvers without maps. In *Algorithmic Foundations of Robotics XII*, pp. 304–319. Springer.

Fumagalli, M., Naldi, R., Macchelli, A., Forte, F., Keemink, A. Q., Stramigioli, S., Carloni, R., and Marconi, L. (2014). Developing an aerial manipulator prototype: Physical interaction with the environment. *IEEE Robotics & Automation Magazine*, 21(3):41–50.

Gioioso, G., Franchi, A., Salvietti, G., Scheggi, S., and Prattichizzo, D. (2014). The flying hand: A formation of uavs for cooperative aerial tele-manipulation. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4335–4341. IEEE.

Haag, K., Dotenco, S., and Gallwitz, F. (2015). Correlation filter based visual trackers for person pursuit using a low-cost quadrotor. In *2015 15th International Conference on Innovations for Community Services (I4CS)*, pp. 1–8. IEEE.

Kamel, M., Burri, M., and Siegwart, R. (2017a). Linear vs nonlinear mpc for trajectory tracking applied to rotary wing micro aerial vehicles. *IFAC-PapersOnLine*, 50(1):3463–3469.

Kamel, M., Stastny, T., Alexis, K., and Siegwart, R. (2017b). Model predictive control for trajectory tracking of unmanned aerial vehicles using robot operating system. In *Robot Operating System (ROS)*, pp. 3–39. Springer.

Kaufmann, E., Loquercio, A., Ranftl, R., Müller, M., Koltun, V., and Scaramuzza, D. (2020). Deep drone acrobatics. *arXiv:2006.05768*.

Kim, S., Choi, S., and Kim, H. J. (2013). Aerial manipulation using a quadrotor with a two dof robotic arm. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4990–4995. IEEE.

Klingbeil, L., Nieuwenhuisen, M., Schneider, J., Eling, C., Droeschel, D., Holz, D., Läbe, T., Förstner, W., Behnke, S., and Kuhlmann, H. (2014). Towards autonomous navigation of an uav-based mobile mapping system. In *Proceedings of the International Conference on Machine Control & Guidance (MCG)*.

Lepetit, V., Moreno-Noguer, F., and Fua, P. (2009). Epnp: An accurate o (n) solution to the pnp problem. *International Journal of Computer Vision*, 81(2):155–166.

Li, J., Ye, D. H., Chung, T., Kolsch, M., Wachs, J., and Bouman, C. (2016a). Multi-target detection and tracking from a single camera in unmanned aerial vehicles (uavs). In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4992–4997. IEEE.

Li, R., Pang, M., Zhao, C., Zhou, G., and Fang, L. (2016b). Monocular long-term target following on uavs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 29–37.

Liu, S., Watterson, M., Tang, S., and Kumar, V. (2016). High speed navigation for quadrotors with limited onboard sensing. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1484–1491. IEEE.

Loianno, G., Brunner, C., McGrath, G., and Kumar, V. (2016). Estimation, control, and planning for aggressive flight with a small quadrotor with a single camera and imu. *IEEE Robotics and Automation Letters*, 2(2):404–411.

McGuire, K., De Croon, G., De Wagter, C., Tuyls, K., and Kappen, H. (2017). 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.

Molina, M., Frau, P., and Maravall, D. (2018). A collaborative approach for surface inspection using aerial robots and computer vision. *Sensors*, 18(3):893.

Mondragón, I. F., Campoy, P., Olivares-Mendez, M. A., and Martinez, C. (2011). 3d object following based on visual information for unmanned aerial vehicles. In *IX Latin American Robotics Symposium and IEEE Colombian Conference on Automatic Control*, pp. 1–7. IEEE.

Mueller, M., Sharma, G., Smith, N., and Ghanem, B. (2016). Persistent aerial tracking system for uavs. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1562–1569. IEEE.

Nägeli, T., Oberholzer, S., Plüss, S., Alonso-Mora, J., and Hilliges, O. (2018). Flycon: real-time environment-independent multi-view human pose estimation with aerial vehicles. *ACM Transactions on Graphics (TOG)*, 37(6):1–14.

Olivares, M., Mondragon, I., Campoy, P., Mejias Alvarez, L., and Martinez, C. (2011). Aerial object following using visual fuzzy servoing. In *Proceedings of the 1st Workshop on Research, Development and Education on Unmanned Aerial Systems (RED-UAS 2011)*, pp. 61–70. Centro Avanzado de Tecnologas Aeroespaciales (CATEC).

Parra-Vega, V., Sanchez, A., Izaguirre, C., Garcia, O., and Ruiz-Sanchez, F. (2013). Toward aerial grasping and manipulation with multiple uavs. *Journal of Intelligent & Robotic Systems*, 70(1):575–593.

Perez-Grau, F. J., Ragel, R., Caballero, F., Viguria, A., and Ollero, A. (2018). An architecture for robust uav navigation in gps-denied areas. *Journal of Field Robotics*, 35(1):121–145.

Pestana, J., Sanchez-Lopez, J. L., Saripalli, S., and Campoy, P. (2014). Computer vision based general object following for gps-denied multirotor unmanned vehicles. In *2014 American Control Conference*, pp. 1886–1891. IEEE.

Pounds, P. E. and Dollar, A. M. (2014). Aerial grasping from a helicopter uav platform. In *Experimental Robotics*, pp. 269–283. Springer.

Pounds, P. E., Bersak, D. R., and Dollar, A. M. (2011a). Grasping from the air: Hovering capture and load stability. In *2011 IEEE International Conference on Robotics and Automation*, pp. 2491–2498. IEEE.

Pounds, P. E., Bersak, D. R., and Dollar, A. M. (2011b). Practical aerial grasping of unstructured objects. In *2011 IEEE Conference on Technologies for Practical Robot Applications*, pp. 99–104. IEEE.

Pounds, P. E., Bersak, D. R., and Dollar, A. M. (2011c). The yale aerial manipulator: grasping in flight. In *2011 IEEE International Conference on Robotics and Automation*, pp. 2974–2975. IEEE.

Ramon Soria, P., Arrue, B. C., and Ollero, A. (2017). Detection, location and grasping objects using a stereo sensor on uav in outdoor environments. *Sensors*, 17(1):103.

Ratliff, N., Zucker, M., Bagnell, J. A., and Srinivasa, S. (2009). Chomp: Gradient optimization techniques for efficient motion planning. In *2009 IEEE International Conference on Robotics and Automation*, pp. 489–494. IEEE.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv:1804.02767*.

Richter, C., Bry, A., and Roy, N. (2016). Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments. In *Robotics Research*, pp. 649–666. Springer.

Rodriguez-Ramos, A., Sampedro, C., Bavle, H., Milosevic, Z., Garcia-Vaquero, A., and Campoy, P. (2017). Towards fully autonomous landing on moving platforms for rotary unmanned aerial vehicles. In *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 170–178. IEEE.

Ross, S., Melik-Barkhudarov, N., Shankar, K. S., Wendel, A., Dey, D., Bagnell, J. A., and Hebert, M. (2013). Learning monocular reactive uav control in cluttered natural environments. In *2013 IEEE International Conference on Robotics and Automation*, pp. 1765–1772. IEEE.

Sampedro, C., Bavle, H., Sanchez-Lopez, J. L., Fernández, R. A. S., Rodríguez-Ramos, A., Molina, M., and Campoy, P. (2016). A flexible and dynamic mission planning architecture for uav swarm coordination. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 355–363. IEEE.

Sampedro, C., Rodriguez-Ramos, A., Bavle, H., Carrio, A., de la Puente, P., and Campoy, P. (2019a). A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques. *Journal of Intelligent & Robotic Systems*, 95(2):601–627.

Sampedro, C., Rodriguez-Ramos, A., Bavle, H., Carrio, A., de la Puente, P., and Campoy, P. (2019b). A fully-autonomous aerial robot for search and rescue applications in indoor environments using learning-based techniques. *Journal of Intelligent & Robotic Systems*, 95(2):601–627.

Sanchez-Lopez, J. L., Fernández, R. A. S., Bavle, H., Sampedro, C., Molina, M., Pestana, J., and Campoy, P. (2016). Aerostack: An architecture and open-source software framework for aerial robotics. In *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 332–341. IEEE.

Sanchez-Lopez, J. L., Molina, M., Bavle, H., Sampedro, C., Suárez Fernández, R. A., and Campoy, P. (2017). A multi-layered component-based approach for the development of aerial robotic systems: the aerostack framework. *Journal of Intelligent & Robotic Systems*, 88(2):683–709.

Schmid, K., Tomic, T., Ruess, F., Hirschmüller, H., and Suppa, M. (2013). Stereo vision based indoor/outdoor navigation for flying robots. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3955–3962. IEEE.

Spica, R., Franchi, A., Oriolo, G., Bülthoff, H. H., and Giordano, P. R. (2012). Aerial grasping of a moving target with a quadrotor uav. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4985–4992. IEEE.

Spurnỳ, V., Báča, T., Saska, M., Pěnička, R., Krajník, T., Thomas, J., Thakur, D., Loianno, G., and Kumar, V. (2019). Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles. *Journal of Field Robotics*, 36(1):125–148.

Steiner, T. J., Truax, R. D., and Frey, K. (2017). A vision-aided inertial navigation system for agile high-speed flight in unmapped environments: Distribution statement a: Approved for public release, distribution unlimited. In *2017 IEEE Aerospace Conference*, pp. 1–10. IEEE.

Teuliere, C., Eck, L., and Marchand, E. (2011). Chasing a moving target from a flying uav. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4929–4934. IEEE.

Thomas, J., Polin, J., Sreenath, K., and Kumar, V. (2013). Avian-inspired grasping for quadrotor micro uavs. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 55935, p. V06AT07A014. American Society of Mechanical Engineers.

Tomic, T., Schmid, K., Lutz, P., Domel, A., Kassecker, M., Mair, E., Grixa, I. L., Ruess, F., Suppa, M., and Burschka, D. (2012). Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue. *IEEE Robotics & Automation Magazine*, 19(3):46–56.

von Stumberg, L., Usenko, V., Engel, J., Stückler, J., and Cremers, D. (2017). From monocular slam to autonomous drone exploration. In *2017 European Conference on Mobile Robots (ECMR)*, pp. 1–8. IEEE.

Yang, H. and Lee, D. (2014). Dynamics and control of quadrotor with robotic manipulator. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5544–5549. IEEE.

Yao, N., Anaya, E., Tao, Q., Cho, S., Zheng, H., and Zhang, F. (2017). Monocular vision-based human following on miniature robotic blimp. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3244–3249. IEEE.

How to cite this article: Rodriguez-Ramos, A., Alvarez-Fernandez, A., Bavle, H., Rodriguez-Vazquez, J., Lu, L., Fernandez-Cortizas, M., Fernandez, R. A. S., Rodelgo, A., Santos, C., Molina, M., Merino, L., Caballero, F., & Campoy, P. (2022). Autonomous aerial robot for high-speed search and intercept applications. *Field Robotics, 2*, 1320–1350.

Publisher's Note: Field Robotics does not accept any legal responsibility for errors, omissions or claims and does not provide any warranty, express or implied, with respect to information published in this article.