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

Systems Article

Autonomous fire-fighting with heterogeneous team of unmanned aerial vehicles

Fran Real[1], Ángel R. Castaño[1], Arturo Torres-González[1], Jesús Capitán[1], Pedro J. Sánchez-Cuevas[2], Manuel J. Fernández[1], Honorio Romero[1] and Aníbal Ollero[1]

[1]GRVC Robotics Lab, University of Seville, Seville, Spain
[2]SpaceR Research Group, University of Luxembourg, Luxembourg, Luxembourg

Abstract: This paper addresses autonomous fire-fighting missions with a heterogeneous team of *Unmanned Aerial Vehicles* (UAVs). We use UAVs with different capabilities to extinguish cooperatively fires on the ground and on facades of high-rise buildings. The former is done by deploying fireproof blankets, whereas a water-jet system is used to extinguish fires on building facades. First, we propose two novel mechanisms on board UAVs for fire extinction: a system to deploy a blanket on the ground automatically, and a device to shoot water at facade fires. Then, we describe our software architecture for cooperative multi-robot missions in outdoor and unstructured environments, where robustness and reliability play a key role. The system was particularly developed for the *Mohamed Bin Zayed International Robotics Challenge* (MBZIRC), where we achieved the best performance in the fire-fighting Challenge. However, the conceptual design of our extinguishing devices and our software architecture are more general and could be applied to other search and rescue situations. We present not only our results in the final stage of MBZIRC but also our simulations and field experiments throughout the months previous to the competition, during which we assessed our system and tuned its performance.

Keywords: aerial robotics, cooperative robots, emergency response

1. Introduction

The use of teams with multiple *Unmanned Aerial Vehicles* (UAVs) has been becoming a trend in the last few years for a wide spectrum of applications. In many of them, like e.g., fire-fighting (Merino et al., 2012), search and rescue (Alotaibi et al., 2019), or goods delivery in disaster situations, operation in hazardous environments is required, and UAVs are key to access difficult places efficiently. Moreover, multirotors are particularly suited for tasks that require an accurate physical interaction with their environment, as they can hover in place to inspect an area closely and actuate their payload mechanisms precisely. Another interesting aspect is the cooperation between heterogeneous vehicles

Received: 30 September 2020; revised: 12 April 2021; accepted: 14 April 2021; published: 04 November 2021.

Correspondence: Jesús Capitán, GRVC Robotics Lab, University of Seville, Camino de los Descubrimientos, 41092 Seville, Spain, Email: jcapitan@us.es

with complementary sensing and actuation capabilities. This capability can be helpful to minimize the time to complete the task, which is essential in the aforementioned applications.

However, operating heterogeneous teams of UAVs in outdoor and unstructured environments is quite challenging. First, there is a need for specific actuation systems that are efficient and accurate enough to perform reliably the given tasks. Second, the UAVs have to act in a coordinate fashion, even in situations with unreliable communication and faulty systems. Therefore, there is an interest in robust and flexible multi-UAV architectures that are able to integrate efficiently heterogeneous hardware and software platforms.

The *Mohamed Bin Zayed International Robotics Challenge* (MBZIRC)[1] is a competition that proposes in each edition three new *Challenges* and a *Grand Challenge* integrating them all. The Challenges are selected by a panel of experts coming from the top robotics labs worldwide, aiming to maximize the impact on potential real-world applications in outdoor and unstructured environments. Thus, Challenge 3 in MBZIRC 2020 was about fire-fighting: a team of heterogeneous multirotors and an *Unmanned Ground Vehicle* (UGV) had to cooperate in order to extinguish a series of fires in a building and its surrounding area. The task is rather complex, as multiple sub-problems are involved: detecting fires with perception technologies, implementing physical interaction through different types of fire extinguishers, planning the mission, coordinating team members, and so on.

We participated in the second edition of MBZIRC (Abu Dhabi, 2020), as part of the *Iberian Robotics* team, which brought together researchers from the University of Seville, the Instituto Superior Tecnico of Lisbon, and the Advanced Center for Aerospace Technologies of Seville (CATEC). From the numerous participant applications, we were selected for a final stage with 26 other teams from top universities and research centers worldwide. Despite being so selective and having the best robotics labs participating, few autonomous missions demonstrated good performance during the competition, which gives an idea of the complexity of the Challenges proposed.

In this paper, we present our multi-UAV fire-fighting system, which responds to MBZIRC Challenge 3. We describe our approach to solve the Challenge with its different sub-tasks, including our hardware design and software functionalities. However, we go beyond the competition, as we propose a system that could be extended for actual fire-fighting applications in different settings. We start explaining the design of our UAV platforms and the prototype mechanisms that we built as fire extinguishers. Then, we present our multi-robot software architecture for cooperative missions with physical interaction. Although the architecture is generic and not restricted to MBZIRC, we focus on the specific implementation we developed for the fire-fighting Challenge, detailing the modules implemented and their interactions. This includes perception algorithms for fire detection, controllers for task execution, and tools for multi-UAV coordination, among others.

The multi-UAV system presented in this paper achieved one of the best performances in the overall MBZIRC 2020, becoming the winner of Challenge 3. In addition, we strongly believe that our prototype hardware and software architecture can be valuable assets for real teams of fire-fighters. In summary, our main contributions are the following:

- We survey the existing literature on related systems for fire extinction and multi-UAV cooperation (Section 2). Then, motivated by the fire-fighting Challenge in MBZIRC (Section 3), we propose two novel mechanisms to extinguish fires with UAVs (Section 4). Even though our final prototypes are tailored to the Challenge, the underlying concepts could be used for different settings. Moreover, our UAV platforms are designed with an original payload exchange system that allows us to replace onboard actuators quickly, which would be crucial for real fire brigades.
- We present a novel software architecture for multi-robot missions (Section 5), specifically for dynamic, outdoor scenarios that require heterogeneous platforms. The main advantages of our architecture are its flexibility and modularity. The former, because it enables the integration of

[1] https://www.mbzirc.com

robots with different types of capabilities; the latter, because alternative functionalities could be easily implemented by replacing the corresponding modules.
- We tested our system under challenging conditions, like those in realistic fire-fighting scenarios, since the outdoor arena of MBZIRC presented poor GNSS signal, high temperatures, and wind gusts. Therefore, we have demonstrated the robustness and reliability of the system for adverse circumstances, also including faulty UAVs. We provide details of our experimental results before and during MBZIRC (Section 6), including simulations, lab tests in our mock-up scenario, and our performance in the actual trials of the competition.
- Finally, our work and experimental campaigns throughout the year previous to MBZIRC, together with our participation in the final stage, have allowed us to gain valuable experience in multi-UAV systems, which we capture in our lessons learned (Section 7) and conclusions (Section 8).

2. Related work

Competitions are revealing in the last years as a prominent manner of boosting the research and development of new technologies to solve current robotics problems. Numerous competitions are organized yearly worldwide, focusing on a wide range of challenges and types of robots. The *Defense Advanced Research Projects Agency* (DARPA) of the United States has organized multiple well-known challenges. Their DARPA *Grand Challenge* (Buehler et al., 2007) and *Urban Challenge* (Buehler et al., 2009) were devoted to autonomous driving, but more recent editions have focused on search and rescue. The DARPA *Robotics Challenge* (Krotkov et al., 2017) proposed the development of humanoid robots to solve complex tasks in disaster management scenarios, and the DARPA *Subterranean Challenge*[2] the exploration of underground tunnels and mines. The *RoboCup Rescue Robot League* (Pellenz et al., 2015) takes place yearly in indoor scenarios with the idea of promoting repeatiability, hence encouraging the creation of robotics benchmarks. Regarding multi-robot systems that operate outdoors, *euRathlon* (Winfield et al., 2016) was a European competition combining marine, aerial, and ground robots for search and rescue tasks. The recently created MBZIRC competition also addresses outdoor challenges where ground and aerial vehicles need cooperation. In its first edition in 2017, MBZIRC already brought together 25 teams that contributed with valuable approaches for multi-robot coordination in dynamic and unknown scenarios (Bähnemann et al., 2019; Beul et al., 2019; Castaño et al., 2019; Loianno et al., 2018; Spurný et al., 2019).

UAVs are a great asset for search and rescue operations. These applications involve hazardous scenarios with limited accessibility and communication, in which aerial vehicles can make a difference. Moreover, disaster management operations usually take place in large-scale scenarios that can be explored more efficiently with teams of multiple UAVs. For instance, in Scherer et al. (2015), a modular architecture of multi-UAV systems for search and rescue missions is presented; and Erdelj et al. (2017) propose multi-UAV teams to establish wireless communication links in disaster management scenarios. Regarding coordination, these multi-UAV teams need to accomplish their tasks in the minimum amount of time, to find the maximum number of survivors. This problem is rather challenging and can be mapped into a *multi-robot task allocation* problem (Nunes et al., 2017). In this sense, multiple algorithms to efficiently allocate search and rescue tasks in multi-UAV teams have been proposed (Alotaibi et al., 2019; Kurdi et al., 2016). A mixed integer linear program to generate optimal multi-UAV plans is introduced in Lee and Morrison (2015). Bevacqua et al. (2015) also propose a planning and execution system for human interaction with multi-UAV teams in rescue missions.

Fire-fighting is one of the main applications related with search and rescue in which multi-UAV systems have demonstrated their possibilities. In particular, heterogeneous teams with multiple UAVs have been proposed for forest fire monitoring (Ghamry et al., 2017; Merino et al., 2012), as these

[2] https://www.subtchallenge.com

teams can improve efficiency in large-scale scenarios. Detection and monitoring of forest fires with multiple UAVs is also addressed in Sherstjuk et al. (2018), and some authors have proposed complete systems for detection and extinction (Harikumar et al., 2019). Regarding indoor fire-fighting, there are less UAV systems. For example, a semi-autonomous indoor fire-fighting UAV is presented in Imdoukh et al. (2017). The pilot is able to view the environment through onboard cameras.

In terms of fire detection, different types of techniques and sensors have been used in the past. Mostly, fire detection has been performed based on color (Çelik & Demirel, 2009), temperature from infrared cameras (Pecho et al., 2019), or smoke (Yuan et al., 2019). More advanced methods combine those with other techniques, e.g. fusing information from color and motion features, not only to detect fires but also to estimate their motion (Yuan et al., 2016). Another example of multi-modal fire detection is using optical smoke, gas, and microwave sensors (Krüll et al., 2012).

Finally, apart from detecting and monitoring fires, there are also available solutions for fire extinction with UAVs. Since UAVs are usually limited in size and weight, most solutions only extinguish fires partially, unless they are small enough. Nonetheless, UAVs can help to control a fire until ground vehicles or larger manned aerial vehicles arrive. There are works proposing fire extinguishers on board UAVs (Imdoukh et al., 2017; Manimaraboopathy et al., 2017); while others have proposed dropping extinguishing balls (Soliman et al., 2019), collecting and releasing water like big fire-fighting planes (Qin et al., 2016), or carrying a hose connected to a water source or other fire retardant on the ground (Ando et al., 2018; Whitaker & Corson, 2017). In this paper, we contribute a team of multiple heterogeneous UAVs. For that, we propose two different systems for fire extinction: a water-jet system for high-rise building fires and a fireproof blanket deployment system. Moreover, we introduce a complete multi-UAV architecture for autonomous fire-fighting.

3. Problem statement

Challenge 3 of MBZIRC 2020, which motivates our work in this paper, consists of a team of multiple UAVs (up to 3) and a UGV that must cooperate in order to find and extinguish a series of fires inside and outside a mock-up high-rise building. The arena (60 m × 50 m) includes a structure of three stories (up to 18 m in height) that emulates the building. There are windows and doors to access inside the building with the robots. Figure 1 depicts the main elements of the Challenge. There are three different types of fires in the arena:

- *Ground fires*. These fires are placed outside the building at ground level, each on top of a wooden box with a squared section of 1 m × 1 m. The fire is emulated by means of a red silk flame that moves with a fan and an array of resistances forming a heat source. This type of fire must be extinguished by dropping a fireproof blanket on top, either with a UAV or the UGV.
- *Facade fires*. These fires are placed on the facades of the building at different fixed positions. Each fire consists of an actual flame generated with propane gas in a circular pipe. This type of fire has also a circular hole in the middle with a small deposit inside, and it must be extinguished by shooting water through the hole with a UAV.
- *Indoor fires*. These fires are placed inside the building, at each of the three floors. This type of fire is also emulated with heat and a silk flame, and it must be extinguished by shooting water in a deposit with a UAV (or with the UGV on the ground floor).

At each trial, the multi-robot team has a maximum of 15 minutes to find and extinguish 2 ground fires, 3 facade fires and 3 indoor fires (one per floor). The ground fires are placed at unknown locations, while the *active* facade fires are selected randomly. The scoring is based on the degree of extinction for each fire in the arena. For facade or indoor fires, the score is proportional to the volume of water in the deposit, with 1 liter being enough to reach the maximum score. This maximum score varies depending on the fire altitude, to reward difficulty. For ground fires, the score is proportional to the surface covered with the blanket. A higher score is given if the fire is extinguished by a UAV. More

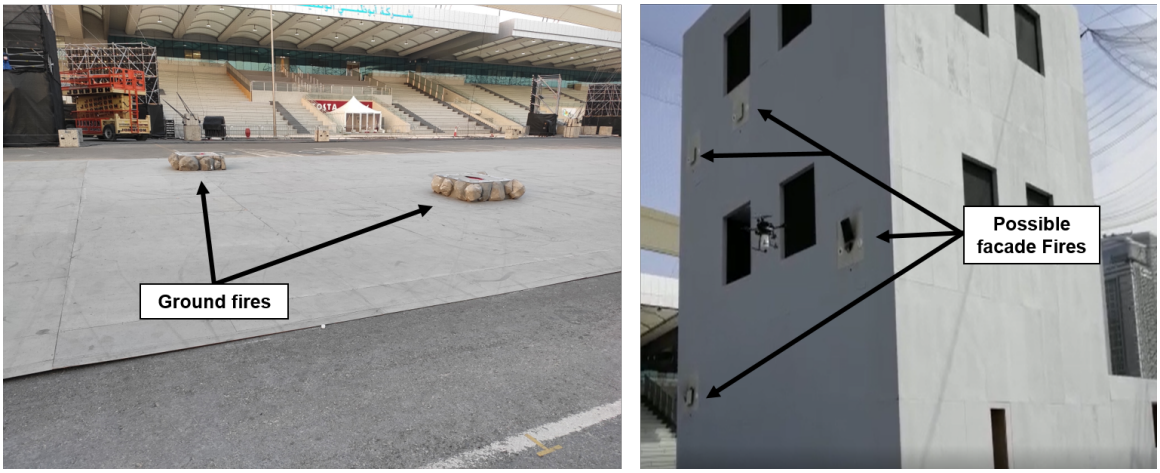

Figure 1. Pictures of the MBZIRC arena for Challenge 3. Left, general view with ground fires. Right, high-rise building with windows and facade fires.

details about the scoring scheme and the Challenge description can be seen on the official MBZIRC website[3].

GNSS is allowed outside the building but the use of RTK (Real-Time Kinematic positioning) incurs a penalty of 25%. UAVs must fit into a box of dimensions 1.2 m × 1.2 m × 0.5 m, although the organization allowed later the use of larger platforms (up to 1.7 m × 1.7 m × 1 m) with an associated penalty. A ground station for UAV monitoring and control is allowed, and communication between the UAVs and with the ground station is possible through a 5 GHz Wi-Fi network provided by the competition organizers. The technical specifications for this Challenge guided the design of our UAV platforms and the mechanisms to extinguish fires, as will be seen in Section 4.

Since the UAVs were given a higher score when extinguishing ground fires outdoors, our team decided to concentrate the UGV on the indoor fire at ground level; devoting a heterogeneous team of UAVs to water and blanket-based fire extinction. Thus, the UGV behavior was decoupled from the UAVs, and the remainder of this paper is focused on our multi-UAV solution for fire-fighting.

4. Hardware systems

This section describes the hardware systems of our multi-UAV team for MBZIRC Challenge 3. First, we describe the aerial platform and the hardware components that are common for all the UAVs. Then, we provide details of our two heterogeneous devices for fire extinction: a system to deploy fireproof blankets and a water-jet extinguisher. Our aerial platforms are equipped with a custom payload-exchange system in order to swap between the two types of fire extinguishers easily, which capability is key to rapid reconfiguration during fire-fighting missions. Even though we tailored our prototypes to address the competition requirements, their conceptual design could be reused for similar fire-fighting tasks in other settings.

4.1. Aerial platform

Our aerial platforms were multirotors made by the local manufacturer Proskytec (see Figure 2), based on the DJI E2000 propulsion system. Each UAV weights 6.5 kg (including batteries and avionics) and it can carry a payload of 3.5 kg. The platform fits without penalty within the restrictions imposed by

[3] https://www.mbzirc.com

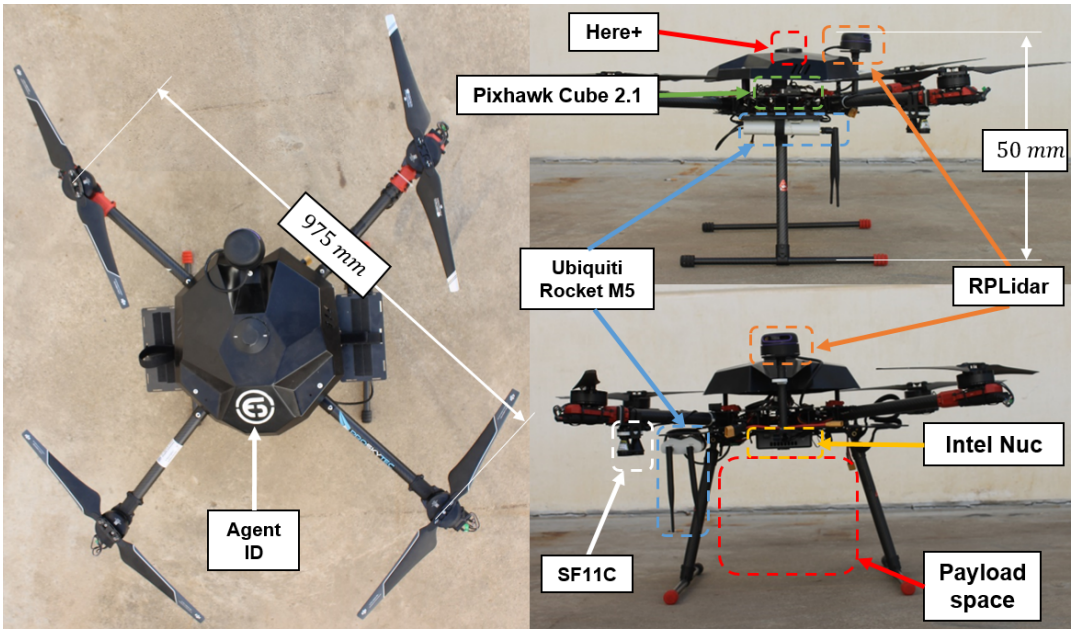

Figure 2. Several views of the aerial platform indicating the spatial distribution of the equipment on board.

the organization (1.2 m × 1.2 m × 0.5 m). The maximum flying time carrying the heaviest extinguisher is near 10 minutes. In practice, this was not an issue, since our UAVs needed to land after each fire extinction to replace the blanket or refill the water tank, so batteries could also be replaced.

Each UAV has a Pixhawk Cube 2.1 autopilot with a Here+ RTK GNSS receiver for navigation. Besides, we added a series of sensors necessary to accomplish this particular Challenge, with slightly different positions depending on the UAV type. Figure 2 depicts the spatial distribution of the onboard equipment. An Intel NUC-i7 with 16GB RAM centralizes computation onboard, being connected to all the other devices through USB ports. We replaced its original metallic case with a custom-made plastic case to reduce weight. A Slamtec RPLIDAR A3 is placed on top of the UAV facing forward to detect walls; and a Lightware SF11C laser altimeter is used for UAV altitude control and to measure precisely distance above ground fires before dropping a blanket. Each UAV has also two cameras for fire detection, an RGB-D Intel RealSense D435i and a TeraRanger Evo Thermal sensor. They both are facing forwards in the case of UAVs in charge of facade fires and downwards for UAVs extinguishing ground fires. For communication with a control computer on the ground, there is an onboard Ubiquiti Rocket M5 5.8 GHz radio link.

Finally, in our heterogeneous team, each UAV is equipped with one of the two types of fire extinguishers. Thanks to our payload-exchange system, the payload corresponding to each extinguisher can be mounted or unmounted rapidly to reconfigure the UAV. In particular, the payload is attached to the UAV frame using four fastenings that incorporate elastomeric blocks to dampen vibrations. Then, two light steel rods, one at each side, are inserted through the fastenings to fix the whole structure. The specific payloads associated with each type of fire extinguisher are described in next sections.

4.2. Blanket deployment system

This fire extinguisher is a system that can be placed aboard a UAV to transport and deploy a fireproof blanket. Our design consists of a mechanism that can expand itself automatically after hitting the ground, in order to cover the fire properly. We achieve this by transforming potential (due to the

UAV altitude) and elastic energy (stored in a set of internal springs) into a mechanical force that spreads out the blanket when it hits the ground. Our blanket deployment system is inspired by an *inverted* umbrella, as shown in Figure 3. It has a bullet-shape design to minimize air friction while falling down, increasing the energy before the impact. A major feature is that the whole system can be mounted quickly, so that blankets can be replaced efficiently. This was a requirement for the competition, where blankets were provided a few minutes before each trial, but also for actual fire-fighting operations, where time is precious.

The core of the system consists of a hollow *steel cylinder* (35 mm wide, 125 mm long) with eight tempered high-carbon steel *wires* that extend from its bottom and support the blanket, as depicted in Figure 4. Some clips are used to fix the blanket to these wires. Each wire is connected to a hard spring within the steel cylinder. During the mounting, these springs are compressed and the wires folded, in

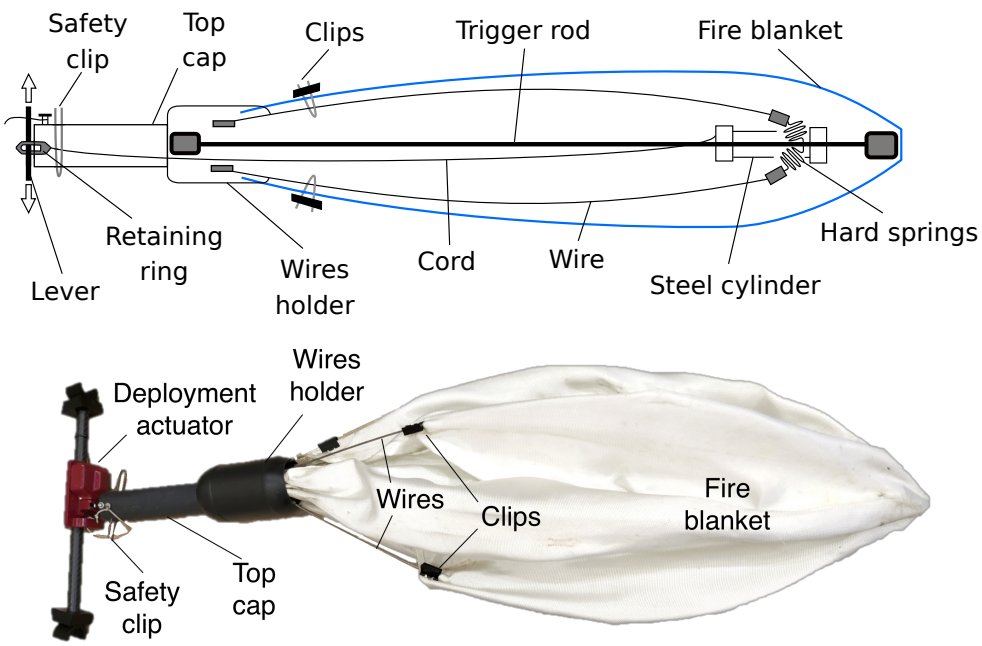

Figure 3. Top, diagram with the main parts of the blanket deployment system. Bottom, the actual system folded.

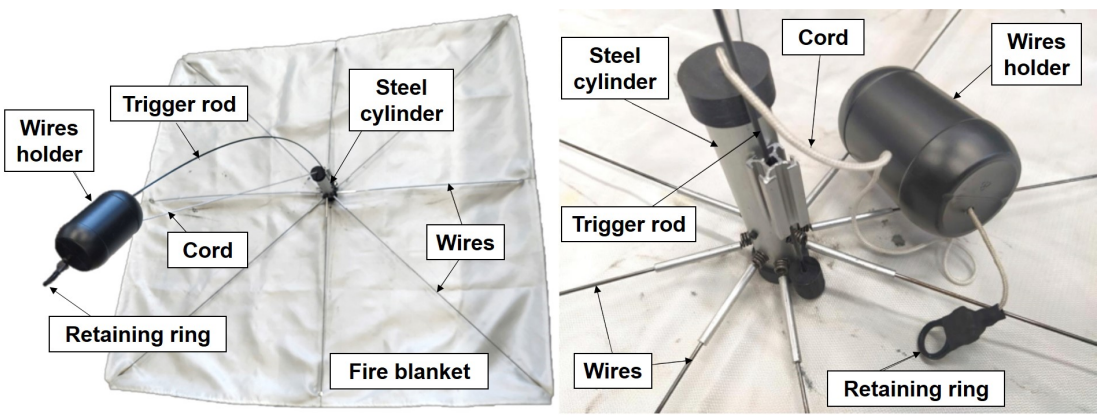

Figure 4. Left, a general view of the blanket deployment system unfolded. Right, a detailed view of the steel cylinder.

order to store the elastic energy that is released once they recover their original shape, spreading out the blanket. A 1.3 m long cord made of Dyneema is connected to the other extreme of the steel cylinder. At its tip, the cord ends with a steel *retaining ring* to hang the whole mechanism from the UAV. Once they are folded, the tips of the wires are held by a barrel-shaped piece (75 mm wide, 125 mm long) made of polypropylene plastic, called *wire holder*. The Dyneema cord goes through this plastic holder too. Besides, there is a 1 m long carbon fiber rod with plastic taps at both tips that goes along the whole umbrella. This *trigger rod* touches the middle of the blanket with one tip and the wire holder with the other, and it can move along a guide that is attached to the lateral of the steel cylinder. When the mechanism hits the ground, this rod pushes the wire holder up, releasing the wires tips, and spreading the blanket.

The deployment actuator consists of a PLA custom-made piece that is mounted on the landing gear of the UAV and that contains a servomotor inside. A lever attached to this servomotor goes through the retaining ring, in such a way that moving the lever releases both ring and cord, and the whole system falls down. There is also another plastic *top cap* that pushes downwards the wire holder and prevents the mechanism from opening accidentally due to vibrations or to the airflow. It also has two holes to insert a *safety clip*. Once the mechanism is installed on the UAV, with the lever of the servomotor in its place, this safety clip is removed before taking off. The whole system (including the blanket) is 1150 mm long, 300 mm wide, and weights 1540 g. It has to be deployed from a height higher than 2.5 m approximately and it takes less than 3 minutes to be assembled and installed on the UAV.

This deployment system has demonstrated to be effective, and it could be used beyond the competition to extinguish small fires. With a few modifications, the releasing mechanism could also be used to deploy other kinds of payloads, such as commercial packages or humanitarian aid parcels in disaster scenarios. The altitude of the delivery point is a parameter in our system, and it can be adapted to deliver the packages at ground level or from a higher distance in case of sensitive material (a parachute or a similar mechanism could be incorporated for a safe landing).

4.3. Water-jet extinguishing system

This fire extinguisher is a system to shoot water horizontally from the UAV. The prototype is shown in Figure 5, and it consists of a water tank, an electric pump, and two tubes for the water jets. The system performs a parabolic shot and it can be calibrated to concentrate the two water jets on a target located at a distance ranging from 1 m to 3 m. The water tank is made of rigid polyethylene and it can contain up to 6 liters. It has a 12 V 4.5 A submersible pump inside, which can pump up to

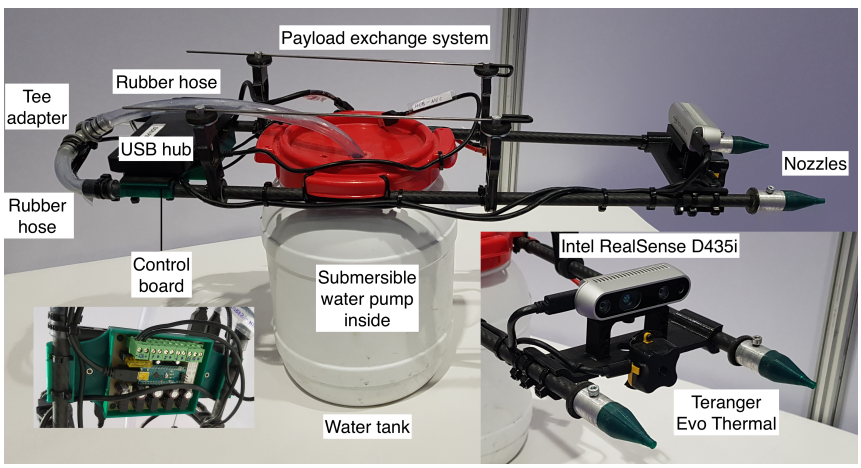

Figure 5. General and detailed views of the prototype built for our water-jet extinguishing system.

18 L/min. The water streams through flexible rubber hoses and a hydraulic tee adapter into the two carbon fiber tubes, which are 650 mm long and 12 mm width. The tubes have conical nozzles at their tip to eject water with higher pressure, reaching a longer range and keeping the water jet as straight as possible. We tested PLA custom-made nozzles of several sizes (from 1 mm to 3 mm) and inner shape (circular and ellipsoidal) to select the best solution in terms of range, discharge speed, and water-jet dispersion. After testing all the configurations, we selected 1.5 mm circular nozzles that enabled us to reach a 3 m range with little drop of the jets, and a 165 mL/s discharge speed.

The carbon fiber tubes are also part of the mechanical structure of the extinguishing system. The water tank is attached to both tubes at the payload's center-of-mass, and two custom-made PLA brackets add stiffness to the structure and calibrate the convergence angle between the two water jets. Moreover, another front bracket holds the RGB-D camera and the thermal sensor for fire detection, whereas a back bracket holds the electronics to control the extinguishing system. A USB hub centralizes connections of all these extinguisher devices with the onboard computer. A power connection for the 3S batteries is also needed. The whole water extinguishing system weights less than 2 kg and it can be mounted (or unmounted) in less than 2 minutes.

This system successfully pumped water into the target during our local tests and the rehearsals in Abu Dhabi, and it could be used to extinguish small fires at high altitudes in real scenarios beyond the competition. Furthermore, other water-based liquid solutions could be used (even as sprays with a few modifications), opening the possibilities to other applications, such as agriculture or disinfection services in remote locations.

5. Software architecture

In this section, we present our multi-agent software architecture, which is modular and flexible enough to be adapted to different scenarios. First, we introduce the general architecture and concepts and then, we provide details of the modules developed for our specific implementation of the architecture in the MBZIRC fire-fighting Challenge.

Our software architecture embodies four main concepts: *Agents*, *Components*, *Actions*, and *Tasks*. *Agents* are entities with perception and actuation capabilities through their onboard sensors and actuators, respectively. They also communicate when possible and interact physically with the environment. Agents are heterogeneous, i.e., each Agent offers a certain set of possibilities depending on its payload configuration. Therefore, the architecture facilitates integrating UGVs and UAVs, as required for the MBZIRC competition. Since we focus on the multi-UAV cooperation part in this paper, Agents will refer, from now on, to our aerial robots. Moreover, our system may optionally include a *Ground Station* that runs software modules, which entity we will also consider an Agent.

Components are software modules that provide specific functionalities implemented by tailored algorithms. Some of them run continuously in the background, as for instance, those Components related to perception and estimation activities that provide information about the environment (e.g., an algorithm for fire detection). Other Components just keep listening so that they can act in a timely fashion, as in the case of hardware actuator drivers. Nonetheless, Components may be deactivated at any time, in order to save processing load or due to strategic needs.

Depending on its hardware capabilities, each Agent can perform some basic *Actions*, which involve movement (e.g., take-off or landing) or physical interaction with the environment (e.g., extinguishing a fire). Also, each Agent offers a set of *Tasks* that represent higher-level behaviors implemented by a *Finite State Machine* (FSM). Tasks add two main features to Actions: composability and coordination. Tasks can be made up of several sub-tasks to create more complex actions (e.g., following a path with sequential *go-to* Actions). Moreover, they can implement mechanisms for multi-robot coordination, interacting with Actions and Components from other Agents. For instance, a Task to search for fires could use a centralized Component for conflict resolution, "blocking" a certain area where the UAV is searching for fires, thus preventing other UAVs from entering that space.

Our architecture combines the aforementioned concepts to build a hierarchical structure with three software levels on top of the hardware, as depicted in Figure 6. At the lowest software level,

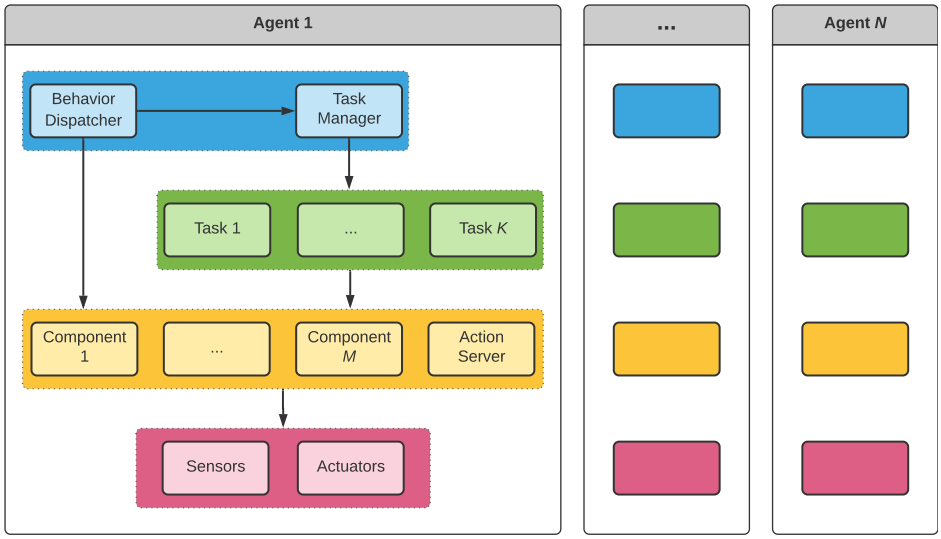

Figure 6. Scheme of the proposed multi-agent architecture. There are three different levels of software modules on top of the hardware. Each Agent offers its own Actions and Tasks. Optionally, one of the Agents could be a Ground Station running centralized components for coordination.

Components are run at each Agent, including the Ground Station, if it exists. Each Agent also runs an Action Server, which is a software module in charge of handling the execution of Actions when they are called by Tasks. Note that, due to system heterogeneity, each Agent can run different Components or Actions.

At the medium level, each Agent exposes its own Tasks, which act as interface with the rest of the Agents. Tasks can be externally started, preempted, or cancelled. Agents are idle or hovering by default. The main coordination of the mission takes place at the top level, through a software module called *Behavior Dispatcher*. This module implements the core strategy by means of a script that describes the required steps to accomplish the mission. Thus, we follow a master-slave approach, as Agents just keep waiting for commands from this Behavior Dispatcher, which calls their different Tasks in the right order. For that, a *Task Manager* is used to handle non-blocking calls. The Task Manager runs a different thread per Agent, so that the Dispatcher can start and preempt Tasks at multiple Agents simultaneously without getting blocked until Task completion.

It is key to highlight that our concept of Behavior Dispatcher is flexible: different allocations are possible depending on the situation. In general, a single Behavior Dispatcher would be in charge of coordinating the mission centrally, either from the Ground Station or from a specific Agent designated as leader. However, it is also possible to have multiple Behavior Dispatchers running on different, decoupled Agents that do not need to cooperate among themselves (for instance, when they have heterogeneous tasks that do not interfere).

The architecture proposed is flexible and adaptable, as it offers an adequate level of abstraction. First, Components and Actions allow us to abstract the system from specific hardware and functionalities, enabling the existence of heterogeneous Agents. Thus, they can be easily added or replaced to incorporate different algorithms and functionalities or to deal with new sensors and actuators. Second, Tasks can be reused and combined to compose more complex Agent behaviors hierarchically, depending on the mission specifications. Last, the overall strategy to solve each mission is encoded in one or several Behavior Dispatchers, so the whole system behavior can be adapted just by reconfiguring those modules. This capability is particularly relevant for multi-UAV systems operating in dynamic settings, as the conditions may change and mission designers must be ready for different situations. Indeed, flexible and simple architectures are quite valuable for competitions like MBZIRC, where the scoring schemes and rules can evolve even during the competition.

Other key aspects for our architecture design are reliability and robustness to robot or communication failures. For each Task, we consider a successful or failure outcome, so that the Behavior Dispatcher can account for that and apply contingency actions. Moreover, all Tasks are also preemptable, e.g., the Dispatcher may decide to stop searching for more fires once one is found, to concentrate on extinguishing it. We will describe in the next sections how failures are handled for each of the Tasks of our specific implementation for fire-fighting. Regarding communication, the architecture assumes unreliability and it does not use blocking procedure calls, so Agents can keep working without communication. Besides, inter-robot communication is minimized, being concentrated on as few Components as possible. Last, we consider the option of a total loss of communication and we defined alternative behaviors in that case. More details about the communication management in our multi-UAV system will be discussed in Section 5.5.

Finally, we implemented our multi-agent architecture using the *Robot Operating System* (ROS), but our concepts and hierarchical structure are agnostic to the middleware used, so that it can be implemented in other frameworks. In our case, Components were implemented as C++ ROS nodes for efficiency reasons, whereas Behavior Dispatchers were Python scripts to improve readability. Moreover, Action Servers were implemented using the ROS actionlib library[4], and FSM in the Tasks with the SMACH library[5].

Due to the flexibility and modularity integrating different modules, our software architecture could be used in a varied set of multi-UAV applications, such as surveillance, facade inspection, or package delivery. Nevertheless, in the next sections, we describe our specific implementation for the fire-fighting MBZIRC Challenge. We detail the particular Components, Actions, and Tasks implemented, as well as how the Behavior Dispatchers coordinate them to accomplish the mission. Our code is open-source and it is available online[6].

5.1. Components

This section explains our Components developed for multi-UAV fire-fighting in MBZIRC.

5.1.1. Fire extinguisher driver

This Component handles the fire extinguisher devices described in Section 4, namely, the system to deploy blankets and the water extinguisher. Depending on the UAV configuration, one of them is installed on board. For the blanket device, the driver can be commanded to activate the releasing mechanism to deploy a blanket. For the water extinguisher, the pump can be activated or deactivated in order to shoot water. In a first implementation, we installed a sensor at the bottom of the water tank to detect whether it was empty, but we had difficulties in avoiding water leakages. Therefore, we decided to estimate the time needed to empty the tank depending on the water volume refilled and the pump throughput, and we stopped the pump after that time in open loop.

5.1.2. Thermal fire detector

This Component runs on board each UAV to detect fires using images from a thermal camera, which are published at 15 Hz. Depending on the UAV configuration, the camera is facing downwards and the Component searches for fires on the ground; or forward, to search for fires on facades. Basically, the thermal images (see Figure 7) are binarized according to a temperature threshold that is adjusted manually, and the contours of the hot regions are computed in the resulting image, to generate bounding boxes for potential fires. In order to calculate fire 3D positions in the camera frame, a depth estimation is used. In the case of UAVs with the camera pointing downwards, the reading from the laser altimeter is used; for UAVs with the camera pointing forward, the 2D LIDAR is used to determine the distance to the facade.

[4] https://github.com/ros/actionlib
[5] https://github.com/ros/executive_smach
[6] https://github.com/grvcTeam/mbzirc2020

Figure 7. Example of the thermal detection of a ground fire. Left, thermal image. Right, same view from the visual camera.

At first, we thought of reducing the number of false positive detections by filtering out small hot regions, but we ended up discarding that solution not to miss fires. Actually, we realized during the tests that actual fires were producing rather small hot regions, given the low resolution of our thermal camera (32 pixels × 32 pixels). For the ground fires in the MBZIRC arena, the temperature difference with the surroundings was not so significant, as the arena floor was really hot due to the weather conditions in Abu Dhabi. Therefore, we had to increase the temperature threshold in the detector, hence reducing potential false positives.

Finally, we also experienced that the low resolution of the camera yielded an inaccurate 3D positioning of the fires. As this information was not precise enough for fire extinction, we developed additional fire detectors using visual images (to detect red ground fires and facade holes), and we used this thermal detector, in combination with them, to double-check the temperature of possible detections and confirm the actual existence of fire.

5.1.3. Color-based fire detector

This Component runs on board UAVs with the visual camera pointing downwards, in order to detect ground fires using color information. It provides fire 3D positions in the camera reference frame. An HSV (Hue-Saturation-Value) filter based on the one in Bruce et al. (2000) is applied for color segmentation. Each pixel belongs to a specific color if its HSV values fall within given HSV ranges describing that color. Those ranges are defined tuning the detector by hand with datasets of images containing actual fires. We adjusted the filter to detect the red color of the silk flame on ground fires. After the color filtering, some morphological operations (erosion and dilation) and an area-based filtering (too small or too large regions) are applied to the resulting images to remove outliers, and then, a list of colored items with their approximate bounding boxes is generated. Figure 8 shows an example of the detector outcome. The algorithm includes a tracking capability based on pixel distance to follow detections throughout consecutive image frames. Finally, color item positions on the image plane are transformed into 3D metric positions and orientations in the camera reference frame, using the camera intrinsic calibration parameters, and the estimated depth given by the UAV laser altimeter, to resolve the scale factor.

Even though this detector is tailored to the detection of the fires in the competition, it could be adapted to other applications. For instance, we used a similar color-based detector to pick up objects on the floor in another MBZIRC challenge (Real et al., 2021), which may be useful for many applications like package transportation.

Figure 8. Color-based detection of a ground fire.

5.1.4. Hole detector

This Component runs on board UAVs with the RGB-D camera facing forward, in order to detect holes of facade fires. Recall that these fires consist of a rectangular plate with a central hole to throw water into an internal deposit. They also have a concentric ring of real flames generated with propane gas. Color images are processed using the Hough circle transform (Yuen et al., 1990) to extract a list of candidate circles. As the size of the actual hole is known, depth information is used to estimate the expected size of the circle on the image and to filter out candidates. The depth information from the RGB-D camera is also combined with the output of the Wall Detector, to discard circles that are farther or closer than the building wall in front of the UAV, as the target hole should be there.

In our preliminary mock-up tests, we experienced that our detector was sometimes detecting the circle corresponding to the ring of flames (there was a circular gas pipe) instead of the hole, but this was not a problem in practice, as they were both concentric and equally useful to throw water into the deposit. However, once in the actual MBZIRC arena in Abu Dhabi, we realized that, for each facade fire, there were two extra holes at each side of the main one. These holes had a size similar to the central one, and their function was to eject air to simulate wind gusts, but they were not connected to the deposit. As our circle detector found hard to distinguish these circles from the main hole, we added a heuristic to the Action for fire extinction. We aimed the water shot at the middle circle if three were detected, and to the middle point of the line connecting the centers of the circles, if two were detected. Figure 9 shows an example image where the three holes are detected.

5.1.5. Wall detector

This Component runs on board UAVs with water extinguisher, to detect the building walls using the readings from the 2D LIDAR mounted on the UAV. Using the points from the laser scan, we apply a RANSAC algorithm (Fischler & Bolles, 1981) in order to fit straight lines. Then, we filter the possible lines by length, exploiting the fact that the building dimensions are known. This Wall Detector is used to localize the building with respect to the UAV, both when searching for facade fires and during a fire extinction. Since GNSS signal was poor near the building, it was key to maintain a safety distance while exploring a facade, and also to align the UAV to shoot water. This capability to navigate safely in front of a wall could also be useful in other applications like facade inspection.

5.1.6. Window detector

This Component uses information from the RGB-D camera on board the UAV to detect windows in the building facade. The four most prominent corners on the depth image are extracted (images with less than four corners are discarded) using the corner detector described in Shi and Tomasi (1994). Then, the intrinsic camera calibration parameters and depth information are used to obtain the

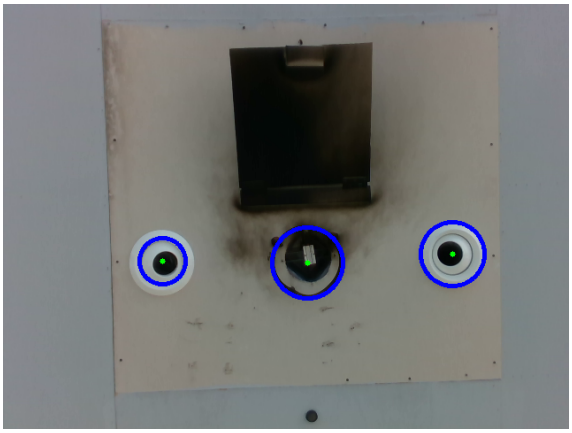

Figure 9. Output of the Hole Detector. The facade fire is not active but its central fire ring is detected with the RGB-D camera. The additional holes to simulate wind gusts are the same size and hence also detected.

3D position of the corners in the camera reference frame. Also, the Euclidean distance between the corners is checked to filter out detections according to the actual window sizes ($2\,\mathrm{m} \times 2\,\mathrm{m}$). The final output of the Window Detector are the 3D orientation of the vector perpendicular to the window plane and the 3D position of its center.

Since there were fires inside the upper floors of the building, we developed this Window Detector so that UAVs could throw water through the windows to put out those fires, or even entering the building. However, in order to minimize risks, and given the level of complexity, we discarded to go inside the building with the UAVs and concentrate on outdoor and facade fires. Moreover, once in the competition arena, we realized that indoor fires could not be detected nor extinguished from outside. Indeed, to the best of our knowledge, none of the participant teams entered with their UAVs in the building to extinguish fires.

5.1.7. UAL

The *UAV Abstraction Layer* (UAL) is a special Component that implements an abstraction layer to control a UAV. UAL is an open-source[7] library (Real et al., 2020) that we developed in our lab to ease the integration of different types of UAV autopilots. UAL offers common interfaces to provide UAV positioning and the possibility of sending basic commands to the autopilot, such as take-off, landing, position or velocity controllers, and so on.

In MBZIRC, our UAV platforms were using either the PX4 autopilot (Meier et al., 2015) or ArduPilot[8] underneath our Component UAL, using the corresponding state estimators and controllers integrated in both autopilots. We configured and tuned those autopilots to control our specific UAVs, but we did not implement customized modules in the low-level UAV control pipeline. Thanks to UAL, the type of autopilot running underneath is transparent for the rest of the system.

5.2. Actions

Each UAV is able to perform a set of low-level Actions that are handled by its onboard Action Server. These Actions are in charge of controlling the UAV to navigate or to operate its fire extinguisher. We implemented the following Actions for the MBZIRC Challenge.

[7] https://github.com/grvcTeam/grvc-ual
[8] https://ardupilot.org

Take-off/Land/GoTo These are basic movement Actions to take off the UAV at a certain altitude, to land it, or to navigate it to a given waypoint in the arena. Our UAL Component is used underneath to command the UAV autopilot.

ExtinguishGroundFire This Action is implemented in UAVs with the system to deploy blankets, which are the ones in charge of extinguishing ground fires. The Action assumes that the UAV has found a ground fire and it is located above that fire, and it activates the controller to deploy the blanket on top. Since the Thermal Fire Detector is not accurate enough for positioning, the Color-based Fire Detector and the UAV altimeter are used to center the UAV on top of the fire at a predetermined altitude. We calibrated our system with multiple trials to find out this optimal position relative to the fire, in which the surface covered by the blanket is maximized. As soon as this target position is reached within a given threshold and for a certain period of time (to assert stability), the releasing mechanism is commanded to deploy the blanket. We also discovered in our tests that the deployment is better when the UAV increases its altitude right after releasing the blanket, so we added this behavior. This is because the air gust induced downwards helps the blanket to fall down straight and open successfully on the fire. Once deployed, the UAV is high enough not to blow away the blanket.

GoToFacadeFire This Action is implemented in UAVs with the water extinguisher, which are the ones that need to get close to facades. It assumes that the UAV is in front of a facade and it is used to navigate the UAV to another waypoint in that facade while searching fires. The target waypoint is given as input. The Action differs from the standard `GoTo` because the navigation is performed with different sensors. `GoTo` uses the autopilot position controller (through UAL) relying on GNSS localization. However, the GNSS signal near the building was not reliable, so we implemented this `GoToFacadeFire` Action for safer navigation. In particular, a velocity control is performed to reach the given waypoint, using the UAV altimeter for altitude feedback and the 2D LIDAR to keep a safety distance from the facade.

ExtinguishFacadeFire This Action is implemented in UAVs with the water extinguisher. It assumes that the UAV is in front of an active facade fire and has to be controlled to perform the extinction operation. The Action uses the output coming from the Hole Detector to center the UAV and perform the shot. We calibrated our system with multiple trials to determine the optimal position relative to the hole from where the amount of water in the deposit is maximized, but still keeping a safe distance to the facade. In case of detecting two holes (see details in Section 5.1.4), the middle point in the line connecting their centers is used to position the UAV. In case of detecting three, the one in the middle is aimed. The UAV orientation is also key for a successful shot. Thus, the Wall Detector is used to keep the UAV oriented perpendicularly to the facade. Once the target position and orientation are reached within some threshold, the water pump is activated.

During our experimental tests, we realized that the water splashed around the hole, and even on the camera lenses, was significantly jeopardizing the accuracy of the Hole Detector, which made it hard for us to use this detector as feedback for UAV positioning. Therefore, as soon as the ideal UAV position and orientation are reached, the UAV is "locked" there, while the water pump is throwing water. We use the readings from the altimeter to keep the altitude fixed, and the Wall Detector output to keep the relative distance with the facade. The operation lasts the estimated time to empty the water tank and then, the UAV needs to land to refill the tank. We thought that this strategy was more efficient than throwing part of the water at different fires within the same flight. Once the UAV lands, the camera lenses can be cleaned for optimal operation.

5.3. Tasks

Apart from its Actions, each UAV is also able to perform a set of higher-level Tasks. These Tasks represent its interface with other Agents and they are to be called by a Behavior Dispatcher running

on the same UAV or somewhere else. Tasks are implemented by a FSM and they can combine the use of several Actions or Components to reach their objective. Contrary to Actions, Tasks have the capacity to create more complex behaviors by combining other Tasks and Actions. They also allow for multi-UAV coordination, for instance, using Components to exchange information with other Agents. Thus, in a Task for extinguishing fires, a UAV could reserve a certain facade so that others do not operate there to avoid conflicts and then, apply sequential Actions to approach the fire and proceed with the extinction.

In our final implementation for the MBZIRC fire-fighting Challenge, we did not have the need to use centralized conflict resolution for the UAVs. There were just 3 heterogeneous UAVs that we assigned to non-conflicting tasks, one for the ground fires and the others to deal with fires in different facades. Besides, there was not much chance to combine sequential behaviors, as our UAVs needed to land after each extinction to refill the water tank or to replace the blanket. However, even though we do not exploit all its potential in this Challenge, the architecture is generic enough to address more complex situations. For the MBZIRC fire-fighting Challenge, we implemented the following Tasks.

Take-off/Land/GoTo These Tasks call the corresponding Actions with the same name. However, if the Action fails, the Task waits for a certain time and it retries. They could also interact with a Component to solve conflicts with other UAVs for navigation, but we did not need to implement this functionality in our final version.

FollowPath This Task is a composition of several `GoTo` Tasks. It receives a list of waypoints representing a path and it executes sequentially the corresponding `GoTo` Tasks.

ExtinguishGroundFire This Task receives as input a searching path to look for a fire on the ground and extinguish it. It calls the `FollowPath` Task to navigate through the list of waypoints until a fire is detected on the ground. The Color-based Fire Detector is used to detect and locate the fire, and the Thermal Detector to confirm whether is really hot (to filter out false positives). If a hot fire is found, the Task calls the `ExtinguishGroundFire` Action. Otherwise, it keeps searching for fires throughout the given path.

GoToFacadeFire This Task calls the corresponding Action with the same name to navigate to a given waypoint following the facade and searching for an active fire. It uses the Hole Detector to find potential facade fires on its way, and the Thermal Detector to check whether the fire is active. If no active fire is detected during the navigation to the waypoint, a failure is reported.

ExtinguishFacadeFire This Task just calls the corresponding Action with the same name. It is assumed that the UAV is facing an active facade fire and the Task tries to extinguish it.

5.4. Behavior dispatcher

In this Challenge, UAVs are heterogeneous and each one can only be dedicated to either ground or facade fires. Due to the involved risk and the scoring scheme, we discarded entering in the building to take care of indoor fires. Since the remaining fires are located at different areas and there are only 3 UAVs, we can allocate them to non-conflicting tasks, one extinguishing ground fires in the surroundings of the building, and the others taking care of different facades. Therefore, we implemented a Behavior Dispatcher running on each UAV, resulting in a totally distributed multi-UAV system. Figure 10 summarizes the final implementation of our architecture for the MBZIRC Challenge, including all the developed modules.

The Behavior Dispatcher on board the UAV with blanket calls the `ExtinguishGroundFire` Task to navigate through a predefined route covering the zone of the arena with ground fires. Within this Task, the UAV stops at the first detected fire and deploys the blanket. If no fire is found, the Task reports a failure and it is called again sequentially. Otherwise, after a successful outcome, it means that the UAV has released the blanket, and it lands to be manually recharged with another one.

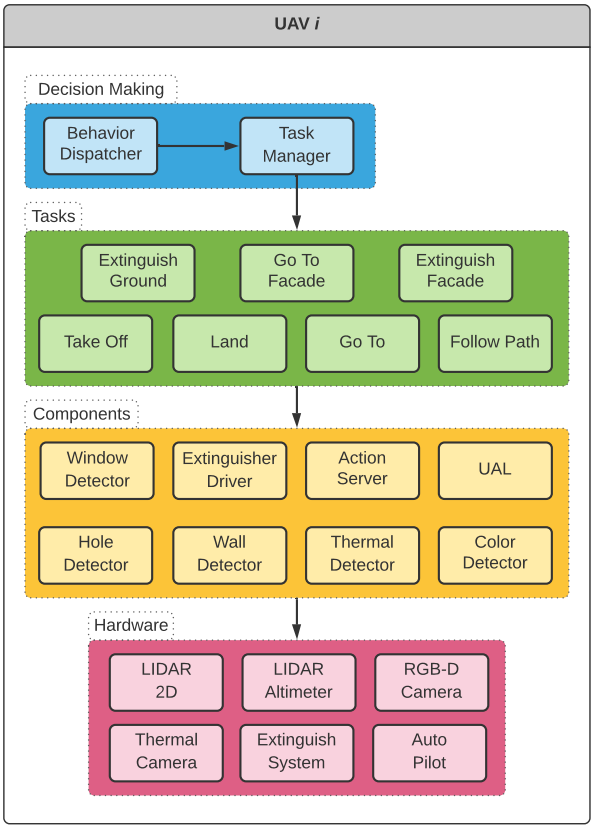

Figure 10. Specific implementation of our multi-agent architecture for the fire-fighting Challenge of MBZIRC.

The Behavior Dispatcher on board UAVs with water extinguisher sends the UAV to its preassigned facade with a `FollowPath` Task. Once there, the `GoToFacadeFire` Task is called sequentially to travel to key waypoints in the facade searching for active fires. These waypoints were set manually in order to cover all the areas in the facade with potential fires. The UAV keeps repeating that behavior until an active fire is found. Then, the `ExtinguishFacadeFire` Task is called and the UAV lands so that its water tank is manually refilled. In subsequent rounds, the UAV could skip the key waypoints corresponding to already extinguished fires, but we did not implement that behavior because in practice we found more efficient to keep throwing water to the same facade fire once detected, as the organization only set one fire per facade.

5.5. Communication

Communication is one of the most critical aspects in multi-UAV systems, and unreliable wireless channels are commonly used. Therefore, in order to increase robustness, we tried to design our system to be as less dependant as possible on the existence of a reliable communication. This is done by increasing the autonomy of each UAV (performing all the required computation on board), and reducing the exchanged information as much as possible. Our overall strategy in multi-UAV systems is using communication channels when available to improve the performance through coordination, but also encoding alternative behaviors in case of a lack of communication.

Actually, in the first edition of MBZIRC (2017), we experienced issues with the Wi-Fi communication network provided by the organization. Therefore, in this occasion we prepared our system to be able to run assuming not communication at all. As explained in Section 5.4, we ran distributed Behavior Dispatchers on each UAV without communication with centralized modules. The area

covered by each UAV is preassigned, i.e., the outdoor ground fires or a specific facade, so conflicts between the UAVs are avoided. Also, as there is only one UAV operating at each region, it does not need to share its fire detections with others. This strategy was enough for the MBZIRC Challenge, as there were only 3 UAVs operating. Nonetheless, more complex rules could be set in case of larger teams. For instance, subdividing the outdoor area in different sections, or using different flight altitudes for the UAVs.

In addition, the communication management in our multi-UAV team was based on the ROS master scheme. In order to increase the system robustness, we used the ROS package *multimaster-fkie*[9]. This package overcomes one the main drawbacks of ROS, which relies on the existence of a single and central master. However, the multi-master scheme allows each UAV to run its own ROS master, and it handles the ROS communication between the different masters (the UAVs were exchanging information about RTK GNSS corrections). In case of a communication failure in one of the UAVs, the others would keep working.

6. Experimental results

In this section, we present our experimental results for fire-fighting with our heterogeneous team of UAVs. We developed a simulated environment for integration, but our main tests were carried out in mock-up scenarios built for the competition. Our team performance during the final stage of MBZIRC, where we achieve the best performance in the fire-fighting Challenge, is also summarized.

6.1. Simulations

We developed a simulated fire-fighting arena following the MBZIRC specifications (see Figure 11). The simulation is based on Gazebo, using the PX4 *Software-In-The-Loop* (SITL) framework[10] to run the same autopilot firmware as in the real UAVs. The main objective of this simulation is to integrate all the software functionalities and test our multi-UAV architecture. Mainly, the interaction of Behavior Dispatchers with Actions and Tasks through the Action Server and the Task Manager, respectively.

Numerous hours of simulation, before and during the competition, allowed us to test extensively the architecture in a safe and consistent manner, and to validate the overall strategy for the Challenge. However, simulating the physical interaction with the environment was difficult. Even though we devised a simulated version of the water extinguishing system by shooting small *water balls* with the UAVs, we did not replicate exactly our prototypes for fire extinction. Thus, we used the simulation to

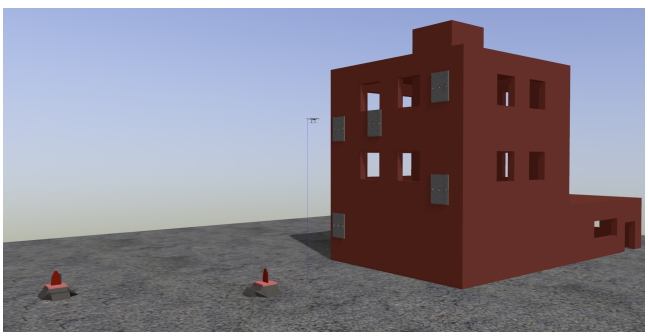

Figure 11. A picture of our simulation with two ground fires and the high-rise building with fire facades.

[9] https://github.com/fkie/multimaster_fkie
[10] https://github.com/PX4/sitl_gazebo

reach an appropriate level of reliability in the software architecture, but we conducted experiments in an outdoor mock-up scenario to calibrate the fire detectors and the actual fire extinguishers. This experiments are described in the next section.

6.2. Mock-up experiments

We built mock-up facilities next to our lab at the University of Seville to test our aerial platforms. The scenario was large enough to fly three UAVs simultaneously, and it had a movable ground fire and a small building with a couple of facade fires. The objective was to validate key parts of the system. These are mainly those related with UAV control and physical interaction, i.e., the detection of fires, the deployment of blankets, and the extinction of facade fires. With these experiments, we adjusted the autopilot controllers for our UAVs and we adapted the hardware design of our fire extinguishers after preliminary tests. We also calibrated the fire extinguishers, which means estimating the optimal altitude to throw blankets, adjusting the angle of the water jets, estimating the maximum water tank volume that our UAVs were able to carry, and estimating the time needed to empty that tank.

We built mock-up fires trying to replicate the ones specified by the MBZIRC rules, as it can be seen in Figure 12. Our mock-up ground fire consisted of a metallic squared plate heated by a fire generated with a gas circuit underneath. We also painted a red circle on top of the metallic plate, to imitate the color of the ground fire in the real MBZIRC arena. Besides, we built a small mock-up building ($1\,\text{m} \times 3\,\text{m} \times 4\,\text{m}$) made with a scaffolding structure, where we placed three facade fires.

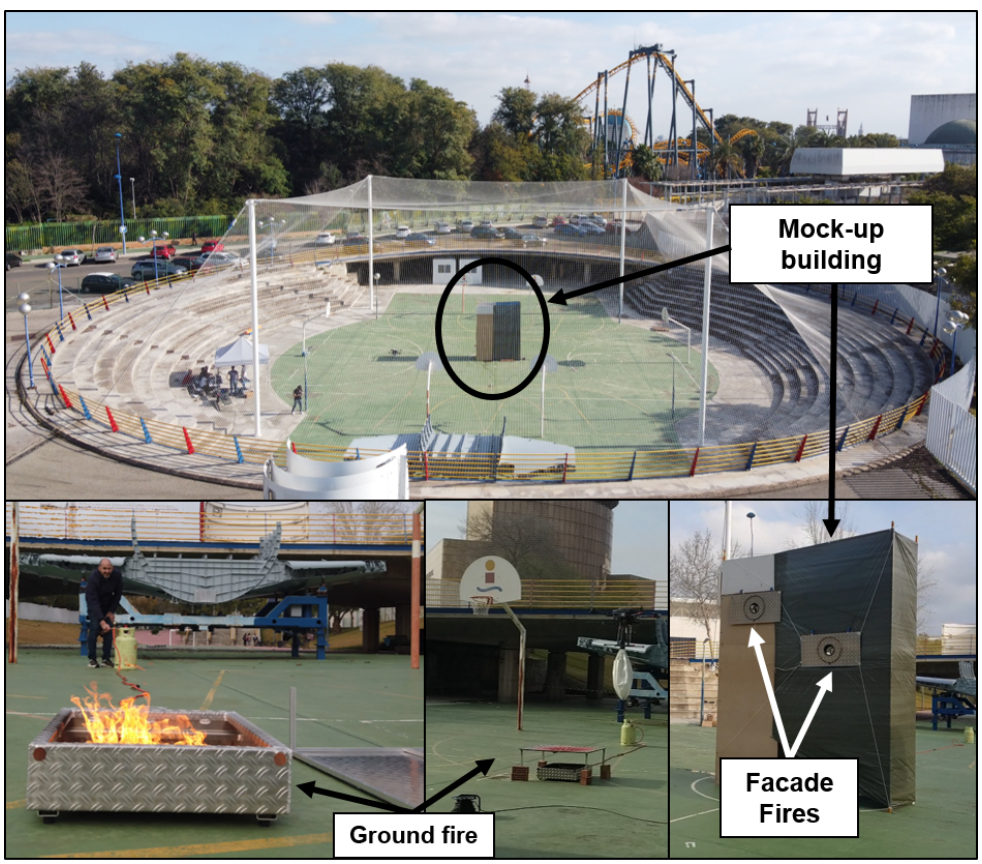

Figure 12. Top, a general view of the mock-up scenario. A safety net covers the whole arena. Bottom, different views of the ground and facade fire mock-ups used in our experiments.

Each mock-up facade fire consisted of a metallic plate with a hole in the middle connected to a jar for water collection. There was a heating resistance within the hole, and a gas ring pipe surrounding the outer part of the hole to produce real fire.

6.2.1. Results for fire detection

We tested and adjusted the Thermal Fire Detector with our ground and facade mock-up fires, and we integrated it with the fire extinction functionalities. Then, in Abu Dhabi, we tuned the detector again, as the fires in the MBZIRC arena presented differences in terms of temperature. In order to assess the accuracy, we analyzed datasets with more than 5,000 thermal images, taken from a UAV pointing at both active and inactive fires on the ground and facades. We obtained, in the case of facades detections, a 10.3% of false negatives (i.e., not detecting fire when there is) and we did not get any false positives (i.e., detecting fire when there is not). While the UAVs were shooting water to the fires, we observed a decrease of around 10 °C in the measured temperature, but not affecting significantly the detector accuracy. For fires on the ground, we achieved a 44.2% of false negatives and a 3.3% of false positives. We detected a 2.7% of false detections due to partial occlusions on the image caused by the oscillations of the blanket mechanism, but the effect was not significant.

In general, we found two main issues with the fire detection based on temperature. First, the sensor sensitivity was not enough to find a threshold that optimized the trade-off between false negative and false positive detections. Also, this sensitivity was affected by the distance to the fire, decreasing the measured temperature when farther. Second, its precision to provide 3D positions of the fires was rather low due to the lack of an accurate camera calibration. Therefore, we eventually opted for using the Thermal Fire Detector in a conservative manner, only to confirm heat once a fire had been detected by visual-based means.

We also tested the Color-based Fire Detector, searching for the red color on the plate of the ground fire. This detector demonstrated to be accurate enough to accomplish ground fire extinctions. This mock-up is the one that presented largest differences with the official one in the competition, as the latter had a red silk flag moved by a fan, instead of a static red circle. Thus, we needed to test and tune the detector again during the rehearsals in Abu Dhabi, though we achieved a similar level of accuracy. In particular, we analyzed a dataset with more than 6,000 frames with ground fires, taken from a UAV in Abu Dhabi, flying between 3 and 7 meters above the ground. From those altitudes, the detector performed really well, just missing some detections due to partial occlusions caused by the hanging blanket appearing on the image.

In the case of the Hole Detector, the false positive detections need to be addressed more carefully, as the UAV may get too close to the building facade when approaching a hole. Therefore, we filtered them properly, using the distance range measurements coming from the onboard 2D LIDAR. In this sense, the detector was reliable enough. We post-processed a dataset containing more than 11,000 images from facade fires in the MBZIRC arena, and we measured an overall rate of 31.1% false negatives and 0.1% false positives. The false positives were extremely rare, except for when the extinguisher was shooting water, that the rate raised up to 12.7%. The false negatives were more common, because the algorithm did not detect all the holes at every frame. The rate of false negatives also raised up to 86.1% while shooting water. As we locked the UAV position before starting to eject water, this degradation was not eventually so relevant for the extinction operation. Our main issue detecting holes was that we were also detecting the two additional holes of similar dimensions that the official facade fires in MBZIRC presented. Nonetheless, we tackled that issue applying some heuristics to decide where to point the water jets, as already explained in Section 5.1.4. Finally, since we decided to minimize risks not going through the building windows to tackle the indoor fires, we did not devote time to build a whole mock-up building with windows, and we only tested the Window Detector in simulation.

6.2.2. Results extinguishing fires

We considered that the key for this Challenge was achieving a reliable performance with the physical fire extinguishers. Therefore, we tested extensively our devices in extinction operations, in order to

improve and calibrate them properly. We started operating them manually on board the UAVs to then, validate them with autonomous experiments in our mock-up scenario.

Blanket deployment The `ExtinguisGroundFire` Action for autonomous blanket deployment was repeated 10 times with our mock-up ground fire in Seville. In those experiments, the fire was always detected and the blanket successfully deployed, covering more than the 50% of the fire in a 80% of the trials. As expected, one of the main issues with the blanket was its pendulum effect, causing oscillations of the UAV during flight. Note that the whole mechanism is a long object of 1.5 kg hanging from a 6.5 kg vehicle, so the involved dynamics during its movement are not negligible. We alleviated this effect by limiting the maximum velocity and acceleration of the UAVs, and establishing a restrictive criteria for UAV stabilization before dropping the blanket.

Figure 13 shows sequences of two blanket deployments on ground fires, one in our mock-up scenario and another in the MBZIRC arena in Abu Dhabi. Both of them represent successful deployments in which more than the 50% of the fire is covered. Although the ground fires in the final competition differed significantly from our mock-ups, we were able to adapt our fire detector adequately.

Figure 14 depicts a plot of the UAV trajectory and the results of its velocity controllers while the UAV is extinguishing a ground fire. In the first segment (magenta line), the UAV is searching for the fire. Once the fire is detected, the vehicle starts to descend and centers itself with respect to the fire, carrying out the *ExtinguishGroundFire* Action (cyan line). As it was aforementioned, the oscillations in the velocity controller during the Action were mainly produced by the transition from a forward flight (searching for a fire) to a hover flight (extinguishing a fire), apart from wind disturbances. It can also be seen how the oscillations decrease after the UAV is hovering for a while. Then, when the UAV reaches a specific height and stays centered within a threshold during a specified time, it drops the blanket (red marker) and goes up (blue line).

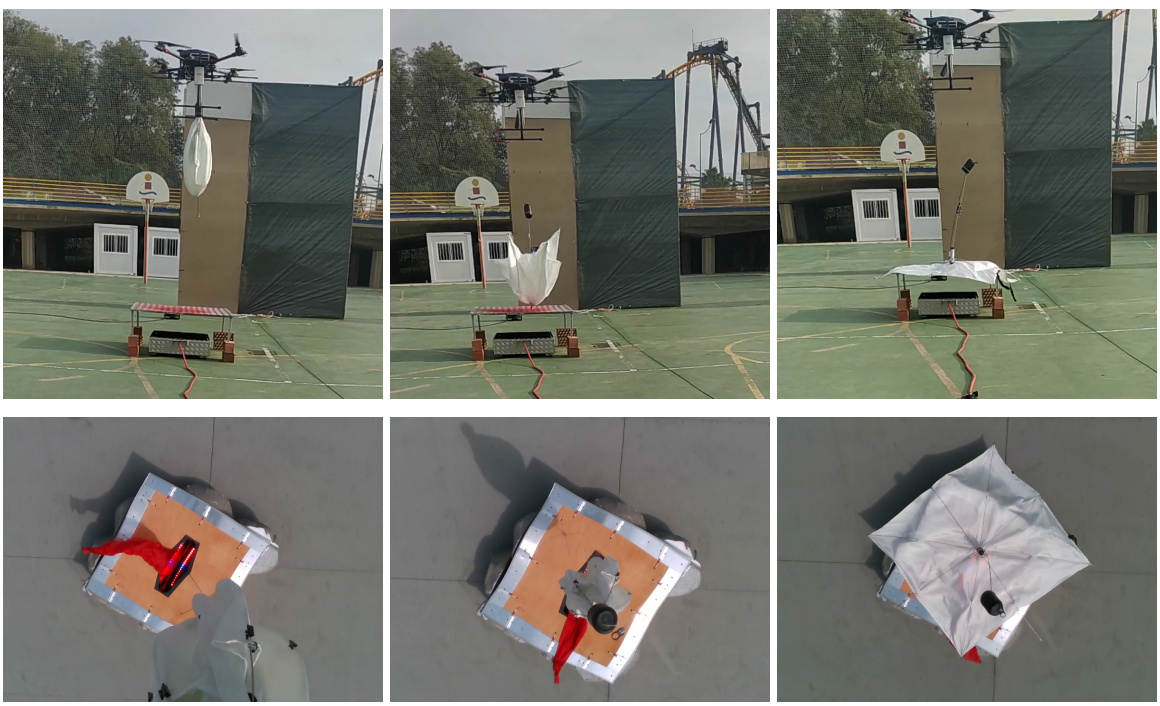

Figure 13. Top, sequence of the deployment of a blanket on our ground fire mock-up. Bottom, images from the onboard UAV camera of a blanket deployment during one of the rehearsals in the actual competition.

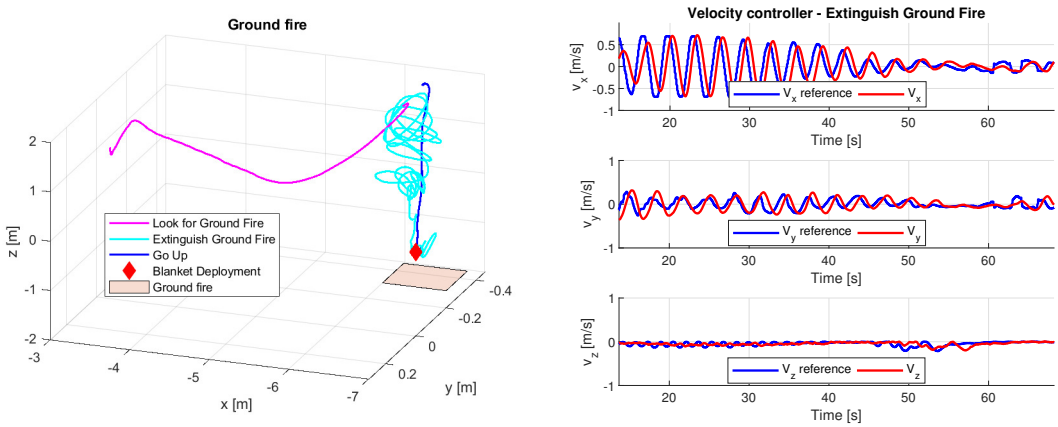

Figure 14. Detail of an experiment extinguishing a ground fire. Left, the trajectory of the UAV is shown while it looks for the fire, descends toward it until the blanket deployment, and goes up. Right, the velocity controllers during the descent.

Figure 15. The water-jet system working during our rehearsals in Seville (top), and during the competition in Abu Dhabi (bottom).

Facade fire extinguishing We also tested our water-jet extinguisher extensively in our mock-up scenario in Seville. These experiments allowed us to test different configurations for the water jets and the amount of water in the tank. We performed more than 30 experiments carrying between 1 and 3 liters onboard. In all of them the UAV had to detect the facade fire, get aligned with the wall and centered with the hole, and shoot water. In our best trial, we managed to fill 1.2 liters (with the tank filled with 2 liters) within the hole, but in average 400 mL to 600 mL. We eventually decided to carry 1.5 liters in the tank in the Abu Dhabi trials, not to overload the aerial platforms. Figure 15 shows our water-jet system working during our experiments in Seville, and during the rehearsals in Abu Dhabi.

Figure 16 depicts an experiment of a UAV conducting an extinction operation with a facade fire in our mock-up scenario. The UAV starts at point (1) and it executes a `GoToFacadeFire` Action to a waypoint in front of an active fire (2). Then, the UAV detects the fire and executes an `ExtinguishFacadeFire` Action where it gets aligned with the facade and centered with respect to the hole, using the wall and hole detectors. Once certain thresholds in its relative

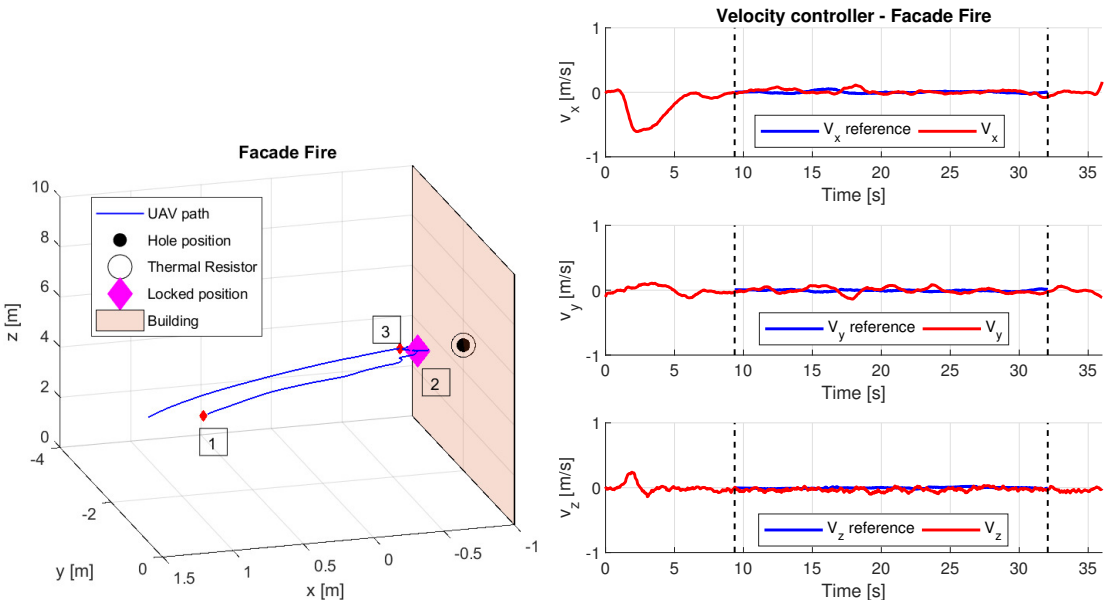

Figure 16. Detail of an experiment extinguishing a facade fire. Left, the UAV trajectory. It approaches the facade, detects the fire, and locks its position once it is centered with the hole to start shooting water. Right, the velocity controller receives references during the `ExtinguishFacadeFire` Action, to center its position with respect to the hole.

Figure 17. One of our aerial platforms during our field experiments in Abu Dhabi.

positioning are achieved, the UAV locks its position (magenta marker) and starts ejecting water. The velocity controller shows how the UAV is able to maintain its position while shooting water, even though it is losing mass due to the water ejection. After emptying the tank, the UAV goes away from the building (3). Finally, a video with a set of our experiments can be found at https://grvc.us.es/downloads/videos/MBZIRC-CH3.mp4.

6.3. Results from the MBZIRC competition

We spend ten days in Abu Dhabi integrating and testing our multi-UAV system before the MBZIRC trials. Even though there was an indoor flight arena provided by the organizers, it was not enough to adjust our controllers, as we were relying on GNSS for positioning and autonomous navigation. Thus, we decided to go to an airfield managed by the Abu Dhabi Radio Control Club (see Figure 17), in

order to fly outdoors. These flights turned out to be key to test our aerial platforms and tune the autopilot controllers after the initial mounting.

The competition was organized in three rehearsals and two trials to score. However, we could not test all the functionalities properly during the rehearsals, since the organization was still tuning up the arena. Thus, the facade fires could not be activated until the second day, and we did not realize until we saw them in Abu Dhabi that they had holes installed to simulate wind gusts (those were not in the specifications), which jeopardized our approach for fire detection. Moreover, the appearance of the ground fires was modified during the rehearsals, changing the color of their wooden cage. Another relevant issue was the fact that the inactive facade fires were sometimes still hot if they had been activated in the previous rehearsal, increasing the number of false positive detections.

Nevertheless, we managed to deploy autonomously two blankets on ground fires during the rehearsals. Then, we decided not to test further the deployment mechanism, and we focused on recording logs to fine tune the detection of ground fires. Regarding the facade fires, we adapted our Hole Detector to take into account the additional holes aforementioned, and we succeeded filling autonomously around 300 mL of water within the deposit of an active fire in one of the rehearsals. After the rehearsals, we were unsuccessful in the first trial, where we did not score. We experienced some issues with our fire detectors, and none of the UAVs tried to carry out an extinction operation. However, we had a quite successful second trial. One of the UAVs detected an active facade fire and shot water, though little got inside because the jet was not pointing properly; while the UAV devoted to the ground fires performed excellently. It found the two ground fires and deployed blankets autonomously on both of them. With this successful trial, we achieved the best score among the participating teams, and we became winners of the Challenge.

7. Lessons learned

Our first option was to use PX4 as autopilot firmware since, among other advantages, we find its SITL tool particularly valuable in making simulations closer to reality. However, we experienced sensor issues (mainly with the external magnetometer) when integrating PX4 in some of our Pixhawk Cube units. Thus, we opted for flying those units with ArduPilot instead, which solved the problem. Our UAL module proved quite helpful for this heterogeneous configuration, as it dealt with firmware particularities, making the type of firmware used transparent for the rest of the system. Furthermore, we decided to use GNSS for UAV localization. Since we used velocity- and position-control on top of the autopilot attitude controller, an accurate tuning of the later was critical, and we chose to use RTK corrections, even if penalized. Nonetheless, the GNSS signal was not quite reliable in the MBZIRC arena during the trials, making it impossible to reach RTK *fixed* status in some areas. Therefore, we conclude that we might have achieved a better overall performance with other methods for localization using the onboard sensors.

One of the reasons for our success in the fire-fighting MBZIRC Challenge was the good performance of our prototype hardware for fire extinction. In particular, the system to deploy blankets outperformed others in terms of efficacy in handling ground fires. We think that a careful hardware design was key in this specific Challenge. Our custom payload-exchange system also turned out to be rather helpful, allowing us to reconfigure UAVs rapidly, and enabling us to react to hardware failures and even to exchange fire extinguishers at mission execution time. Moreover, the construction of thoroughly realistic mock-up systems was crucial to test the fire extinguishers and calibrate them adequately.

We experienced some issues with our fire detectors during the competition. In general, our thermal sensor was not an optimal choice, as it demonstrated insufficient resolution for precise fire detection. Besides, the fire mock-ups in the actual MBZIRC arena stayed hot for some time after having been switched off, which condition led to false positive detections. Therefore, we ended up trusting mainly our alternative detectors that used visual image processing and checking temperature just for final confirmation. Our fire detectors and extinguishers were tailored to the competition rules, and we see room for improvement by making them work in more general situations. For instance, smoke-based

detectors should be quite useful in reality, even though the competition decided to avoid smoke generation. Nonetheless, we believe that the conceptual design of our prototypes could be adapted to be reused in other scenarios. Regarding our overall strategy to address the Challenge, we tended, as most teams did, to simplify complex behaviors throughout the competition, in order to increase our scoring possibilities. In this sense, the allocation of UAVs to different areas to avoid collisions was a wise decision, as it allowed us to optimize our hardware resources. Also, it made us more independent of potential communication failures.

In terms of software, our multi-robot architecture was a success. On the one hand, its flexibility and capacity to integrate alternative modules make it a valuable asset for other potential scenarios. Actually, we used the same architecture to handle other Challenges in MBZIRC. On the other hand, we reached an optimal trade-off between abstraction and simplicity, which is key in competitions, where complex architectures are less likely to achieve the required level of robustness. Thus, we were able to shift between different strategies easily, adapting to changes in the competition rules. Implementing our architecture with ROS was also positive. Our approach was to build a set of composable entities (i.e., Components, Actions, and Tasks) that allowed us to create complex behaviors and integrate heterogeneous algorithms, but at the same time, to leverage the reliability of a well-established framework such as ROS. Last, our abstraction of modules involving physical interaction (Actions) allowed us to test them extensively in a separate and more efficient manner.

In general, ROS has become a standard for the robotics community, since it simplifies the communication between processes and it favors system modularity. One critical issue for multi-UAV systems is its communication scheme, relying on a centralized master node. However, we tackled this problem using the ROS multi-master functionality, with a dedicated master on each UAV. Notably, the recent release of ROS 2 will alleviate significantly communication issues, as it is distributed by design, removing the master. Finally, we decided to use Python for the implementation of our higher-level modules, i.e., Tasks and Behavior Dispatchers. Python provides flexibility for rapid prototyping of high-level scripts, but it requires a more thorough procedure for code validation. Indeed, we came across several failures that were only revealed at mission execution time, with the consequent risk. Many of these issues would have been detected at compilation time with a compiled language such as C++.

8. Conclusions

This paper has presented a multi-UAV system for fire-fighting missions. The team is heterogeneous and the UAVs are equipped either with a system to deploy blankets on ground fires or with an extinguisher to shoot water at facade fires. We designed and implemented these two heterogeneous devices for fighting fires with UAVs, and we have also developed a multi-robot architecture to orchestrate cooperative missions involving physical interaction in outdoor and partially unknown environments. Our system is inspired by the MBZIRC Challenge 3 about fire-fighting, and we provide all the details of our specific implementation and experimental results for that Challenge.

Our results in the competition were outstanding, as we won Challenge 3 (see Figure 18), based on the efficacy of our fire extinguishers, which outperformed others. Although the accuracy of the fire detectors was enough for the competition, we think that putting some effort into generalizing them would contribute significantly to a more general-purpose system. In this sense, we would like to experiment with higher-quality thermal cameras and smoke-based detection algorithms.

In general, we have gained valuable experience and knowledge thanks to our participation in MBZIRC, which allowed us to improve significantly our aerial platforms and software modules for multi-UAV missions. Even though this work has focused on MBZIRC, we believe our prototype hardware devices and software architecture can be extended to be applied to other UAV applications in the search and rescue domain. Thus, we plan as future work to improve the blanket-deploying system by reducing oscillations while it hangs from the UAV. We would also like to integrate in our system more explicit methods for multi-UAV collision avoidance so that they can operate more tightly coupled.

Figure 18. The members of the Iberian Robotics team during the award ceremony of MBZIRC 2020.

Acknowledgments

This work has been partially funded by the Khalifa University, the European Union's Interreg Atlantic Area Programme (DURABLE project), and by the MULTICOP project (Junta de Andalucia, FEDER Programme, US-1265072). The authors would also like to thank all the members of the Iberian Robotics team for their support developing and integrating our systems for the MBZIRC competition, specially to Manuel Villar, Victor Vega, Rafael Salmoral, Alejandro Sanchez and Alejandro Braza. Special thanks to Joe Bracho and the Abu Dhabi RC Club for giving us the opportunity to fly at their airfield.

ORCID

Fran Real https://orcid.org/0000-0002-7533-6153
Ángel R. Castaño https://orcid.org/0000-0002-6235-9524
Arturo Torres-González https://orcid.org/0000-0003-0155-7671
Jesús Capitán https://orcid.org/0000-0002-7534-0187
Pedro J. Sánchez-Cuevas https://orcid.org/0000-0001-9791-4148
Aníbal Ollero https://orcid.org/0000-0003-2155-2472

References

Alotaibi, E. T., Alqefari, S. S., & Koubaa, A. (2019). LSAR: Multi-UAV collaboration for search and rescue missions. *IEEE Access*, *7*, 55817–55832. https://doi.org/10.1109/access.2019.2912306

Ando, H., Ambe, Y., Ishii, A., Konyo, M., Tadakuma, K., Maruyama, S., & Tadokoro, S. (2018). Aerial hose type robot by water jet for fire fighting. *IEEE Robotics and Automation Letters*, *3*(2), 1128–1135. https://doi.org/10.1109/lra.2018.2792701

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. https://doi.org/10.1002/rob.21824

Beul, M., Nieuwenhuisen, M., Quenzel, J., Rosu, R. A., Horn, J., Pavlichenko, D., Houben, S., & 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. https://doi.org/10.1002/rob.21817

Bevacqua, G., Cacace, J., Finzi, A., & Lippiello, V. (2015). Mixed-initiative planning and execution for multiple drones in search and rescue missions. *Proceedings of the Twenty-Fifth International Conference on International Conference on Automated Planning and Scheduling*, 315–323.

Bruce, J., Balch, T., & Veloso, M. (2000). Fast and inexpensive color image segmentation for interactive robots. *2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, *3*, 2061–2066. https://doi.org/10.1109/iros.2000.895274

Buehler, M., Iagnemma, K., & Singh, S. (Eds.). (2007). *Springer Tracts in Advanced Robotics: Vol. 36. The 2005 DARPA grand challenge: The great robot race*. Springer. https://doi.org/10.1007/978-3-540-73429-1

Buehler, M., Iagnemma, K., & Singh, S. (Eds.). (2009). *Springer Tracts in Advanced Robotics: Vol. 56. The DARPA urban challenge: Autonomous vehicles in city traffic*. Springer. https://doi.org/10.1007/978-3-642-03991-1

Castaño, Á. R., Real, F., Ramón-Soria, P., Capitán, J., Vega, V., Arrue, B. C., Torres-González, A., & Ollero, A. (2019). Al-robotics team: A cooperative multi-unmanned aerial vehicle approach for the Mohamed Bin Zayed International Robotic Challenge. *Journal of Field Robotics*, *36*(1), 104–124. https://doi.org/10.1002/rob.21810

Çelik, T., & Demirel, H. (2009). Fire detection in video sequences using a generic color model. *Fire Safety Journal*, *44*(2), 147–158. https://doi.org/10.1016/j.firesaf.2008.05.005

Erdelj, M., Natalizio, E., Chowdhury, K. R., & Akyildiz, I. F. (2017). Help from the sky: Leveraging UAVs for disaster management. *IEEE Pervasive Computing*, *16*(1), 24–32. https://doi.org/10.1109/mprv.2017.11

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*(6), 381–395. https://doi.org/10.1145/358669.358692

Ghamry, K. A., Kamel, M. A., & Zhang, Y. (2017). Multiple UAVs in forest fire fighting mission using particle swarm optimization. *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 1404–1409. https://doi.org/10.1109/icuas.2017.7991527

Harikumar, K., Senthilnath, J., & Sundaram, S. (2019). Multi-UAV Oxyrrhis marina-inspired search and dynamic formation control for forest firefighting. *IEEE Transactions on Automation Science and Engineering*, *16*(2), 863–873. https://doi.org/10.1109/tase.2018.2867614

Imdoukh, A., Shaker, A., Al-Toukhy, A., Kablaoui, D., & El-Abd, M. (2017). Semi-autonomous indoor firefighting UAV. *2017 18th International Conference on Advanced Robotics (ICAR)*, 310–315. https://doi.org/10.1109/icar.2017.8023625

Krotkov, E., Hackett, D., Jackel, L., Perschbacher, M., Pippine, J., Strauss, J., Pratt, G., & Orlowski, C. (2017). The DARPA robotics challenge finals: Results and perspectives. *Journal of Field Robotics*, *34*(2), 229–240. https://doi.org/10.1002/rob.21683

Krüll, W., Tobera, R., Willms, I., Essen, H., & von Wahl, N. (2012). Early forest fire detection and verification using optical smoke, gas and microwave sensors. *Procedia Engineering*, *45*, 584–594. https://doi.org/10.1016/j.proeng.2012.08.208

Kurdi, H., How, J., & Bautista, G. (2016). Bio-inspired algorithm for task allocation in multi-UAV search and rescue missions. *AIAA Guidance, Navigation, and Control Conference*, 1377. https://doi.org/10.2514/6.2016-1377

Lee, S., & Morrison, J. R. (2015). Decision support scheduling for maritime search and rescue planning with a system of UAVs and fuel service stations. *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, 1168–1177. https://doi.org/10.1109/icuas.2015.7152409

Loianno, G., Spurny, V., Thomas, J., Baca, T., Thakur, D., Hert, D., Penicka, R., Krajnik, T., Zhou, A., Cho, A., Saska, M., & Kumar, V. (2018). Localization, grasping, and transportation of magnetic objects by a team of MAVs in challenging desert-like environments. *IEEE Robotics and Automation Letters*, *3*(3), 1576–1583. https://doi.org/10.1109/lra.2018.2800121

Manimaraboopathy, M., Christopher, H. S. V., Vignesh, S., & Tamil selvan, P. (2017). Unmanned fire extinguisher using quadcopter. *International Journal on Smart Sensing and Intelligent Systems*, *10*(4), 471–481. https://doi.org/10.21307/ijssis-2017-264

Meier, L., Honegger, D., & Pollefeys, M. (2015). PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 6235–6240. https://doi.org/10.1109/icra.2015.7140074

Merino, L., Caballero, F., Martínez-de-Dios, J. R., Maza, I., & Ollero, A. (2012). An unmanned aerial system for automatic forest fire monitoring and measurement. *Journal of Intelligent and Robotic Systems*, *65*, 533–548. https://doi.org/10.1007/s10846-011-9560-x

Nunes, E., Manner, M., Mitiche, H., & Gini, M. (2017). A taxonomy for task allocation problems with temporal and ordering constraints. *Robotics and Autonomous Systems*, *90*, 55–70. https://doi.org/10.1016/j.robot.2016.10.008

Pecho, P., Magdolenová, P., & Bugaj, M. (2019). Unmanned aerial vehicle technology in the process of early fire localization of buildings. *Transportation Research Procedia*, *40*, 461–468. https://doi.org/10.1016/j.trpro.2019.07.067

Pellenz, J., Jacoff, A., Kimura, T., Mihankhah, E., Sheh, R., & Suthakorn, J. (2015). RoboCup rescue robot league. In R. A. C. Bianchi, H. L. Akin, S. Ramamoorthy, & K. Sugiura (Eds.), *Lecture Notes in Computer Science: Vol. 8992. RoboCup 2014: Robot World Cup XVIII* (pp. 673–685). Springer. https://doi.org/10.1007/978-3-319-18615-3_55

Qin, H., Cui, J. Q., Li, J., Bi, Y., Lan, M., Shan, M., Liu, W., Wang, K., Lin, F., Zhang, Y. F., & Chen, B. M. (2016). Design and implementation of an unmanned aerial vehicle for autonomous fire-fighting missions. *2016 12th IEEE International Conference on Control and Automation (ICCA)*, 62–67. https://doi.org/10.1109/icca.2016.7505253

Real, F., Castaño, Á. R., Torres-González, A., Capitán, J., Sánchez-Cuevas, P. J., Fernández, M. J., Villar, M., & Ollero, A. (2021). Experimental evaluation of a team of multiple unmanned aerial vehicles for cooperative construction. *IEEE Access*, *9*, 6817–6835. https://doi.org/10.1109/access.2021.3049433

Real, F., Torres-González, A., Ramón-Soria, P., Capitán, J., & Ollero, A. (2020). Unmanned aerial vehicle abstraction layer: An abstraction layer to operate unmanned aerial vehicles. *International Journal of Advanced Robotic Systems*, *17*(4), 1–13. https://doi.org/10.1177/1729881420925011

Scherer, J., Yahyanejad, S., Hayat, S., Yanmaz, E., Andre, T., Khan, A., Vukadinovic, V., Bettstetter, C., Hellwagner, H., & Rinner, B. (2015). An autonomous multi-UAV system for search and rescue. *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, 33–38. https://doi.org/10.1145/2750675.2750683

Sherstjuk, V., Zharikova, M., & Sokol, I. (2018). Forest fire-fighting monitoring system based on UAV team and remote sensing. *2018 IEEE 38th International Conference on Electronics and Nanotechnology (ELNANO)*, 663–668. https://doi.org/10.1109/elnano.2018.8477527

Shi, J., & Tomasi. (1994). Good features to track. *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 593–600. https://doi.org/10.1109/cvpr.1994.323794

Soliman, A. M. S., Cagan, S. C., & Buldum, B. B. (2019). The design of a rotary-wing unmanned aerial vehicles–payload drop mechanism for fire-fighting services using fire-extinguishing balls. *SN Applied Sciences*, *1*(10), 1259. https://doi.org/10.1007/s42452-019-1322-6

Spurný, V., Báča, T., Saska, M., Pěnička, R., Krajník, T., Thomas, J., Thakur, D., Loianno, G., & 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. https://doi.org/10.1002/rob.21816

Whitaker, T. M., & Corson, M. (2017, September 19). *Tethered unmanned aerial vehicle fire fighting system* (U.S. Patent No. 9,764,839). Google Patents. https://patents.google.com/patent/US20170043872A1/en

Winfield, A. F. T., Franco, M. P., Brueggemann, B., Castro, A., Limon, M. C., Ferri, G., Ferreira, F., Liu, X., Petillot, Y., Roning, J., Schneider, F., Stengler, E., Sosa, D., & Viguria, A. (2016). euRathlon 2015: A multi-domain multi-robot grand challenge for search and rescue robots. In L. Alboul, D. Damian, & J. M. Aitken (Eds.), *Lecture Notes in Computer Science: Vol. 9716. Towards autonomous robotic systems* (pp. 351–363). Springer. https://doi.org/10.1007/978-3-319-40379-3_36

Yuan, C., Liu, Z., & Zhang, Y. (2016). Vision-based forest fire detection in aerial images for firefighting using UAVs. *2016 International Conference on Unmanned Aircraft Systems (ICUAS)*, 1200–1205. https://doi.org/10.1109/icuas.2016.7502546

Yuan, C., Liu, Z., & Zhang, Y. (2019). Learning-based smoke detection for unmanned aerial vehicles applied to forest fire surveillance. *Journal of Intelligent & Robotic Systems*, *93*(1-2), 337–349. https://doi.org/10.1007/s10846-018-0803-y

Yuen, H. K., Princen, J., Illingworth, J., & Kittler, J. (1990). Comparative study of Hough Transform methods for circle finding. *Image and Vision Computing*, *8*(1), 71–77. https://doi.org/10.1016/0262-8856(90)90059-E

How to cite this article: Real, F., Castaño, Á. R., Torres-González, A., Capitán, J., Sánchez-Cuevas, P. J., Fernández, M. J., Romero, H., & Ollero, A. (2021). Autonomous fire-fighting with heterogeneous team of unmanned aerial vehicles. *Field Robotics*, *1*, 158–185.