IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 7, 2020, accepted December 21, 2020, date of publication January 5, 2021, date of current version January 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3049433

Experimental Evaluation of a Team of Multiple Unmanned Aerial Vehicles for Cooperative Construction

FRAN REAL, ÁNGEL R. CASTAÑO, ARTURO TORRES-GONZÁLEZ, JESÚS CAPITÁN, PEDRO J. SÁNCHEZ-CUEVAS, (Member, IEEE), MANUEL J. FERNÁNDEZ, (Student Member, IEEE), MANUEL VILLAR, AND ANÍBAL OLLERO, (Fellow, IEEE)
GRVC Robotics Laboratory, University of Seville, 41092 Seville, Spain

Corresponding author: Jesús Capitán (jcapitan@us.es)

This work was supported in part by the Khalifa University, in part by the European Union's Interreg Atlantic Area Programme through the DURABLE Project, and in part by the Junta de Andalucia, FEDER Programme, through the MULTICOP Project, under Grant US-1265072.

ABSTRACT This article presents a team of multiple *Unmanned Aerial Vehicles* (UAVs) to perform cooperative missions for autonomous construction. In particular, the UAVs have to build a wall made of bricks that need to be picked and transported from different locations. First, we propose a novel architecture for multi-robot systems operating in outdoor and unstructured environments, where robustness and reliability play a key role. Then, we describe the design of our aerial platforms and grasping mechanisms to pick, transport and place bricks. The system was particularly developed for the *Mohamed Bin Zayed International Robotics Challenge* (MBZIRC), where Challenge 2 consisted of building a wall cooperatively with multiple UAVs. However, our approach is more general and extensible to other multi-UAV applications involving physical interaction, like package delivery. We present not only our results in the final stage of MBZIRC, but also our simulations and field experiments throughout the previous months to the competition, where we tuned our system and assessed its performance.

INDEX TERMS Multirobot systems, unmanned aerial vehicles, construction.

I. INTRODUCTION

In the last decades, the use of *Unmanned Aerial Vehicles* (UAVs) is becoming quite popular for a wide range of applications. Many of these applications, like for instance, autonomous construction [1], contact-based inspection [2] or package delivery [3], [4], require physical interaction with the environment. Multirotors are particularly suited for these tasks, as they can hover in-place and present *Vertical Take-Off and Landing* (VTOL) capabilities. Another recent trend is using cooperative teams of multiple UAVs for the aforementioned applications, as a way to optimize execution time and operational possibilities by running multiple tasks in parallel. However, the use of multi-UAV teams entails additional challenges such as inter-UAV communication or multi-UAV conflict resolution. Therefore, in order to achieve an efficient, safe and reliable cooperation, there is a need for multi-UAV architectures that are robust to account for faulty teammates and flexible enough to integrate heterogeneous platforms and accommodate new software modules in an easy manner.

Related to applications for cooperative multirotors, the *Mohamed Bin Zayed International Robotics Challenge* (MBZIRC)[1] is a competition that tries to boost research in order to close the gap between current robotics technology and actual requirements of potential applications, mainly in unstructured, dynamic and outdoor settings. For that, each MBZIRC edition proposes three new *Challenges* (and a *Grand Challenge* integrating them all), which are selected and designed by a panel of international experts coming

The associate editor coordinating the review of this manuscript and approving it for publication was Shafiqul Islam.

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

from the most relevant robotics groups worldwide. These Challenges are inspired by the idea of maximizing impact on real applications, and hence, in its second edition, MBZIRC focused on multi-robot teams for autonomous construction and disaster management. In particular, Challenge 2 consisted of building a wall cooperatively with a heterogeneous team of robots, namely several multirotors and an *Unmanned Ground Vehicle* (UGV). The task is actually quite complex, as it involves several sub-problems that need to be solved together in a scenario that is partially unknown: robust perception for 3D object tracking, physical interaction to pick and place bricks, multi-UAV planning and collision avoidance, etc.

We were members of the *Iberian Robotics* team, which was one of the 26 participants of MBZRIC 2020. Ours was a joint team between the University of Seville, the Instituto Superior Tecnico of Lisbon and the Advanced Center for Aerospace Technologies of Seville (CATEC). The complexity of the MBZIRC Challenges can be understood given the fact that only a reduced number of teams managed to perform complete autonomous missions, despite having participants from top universities around the world. Actually, the system presented in this article was the only one from all participant teams that scored in autonomous mode in the wall building part of the Grand Challenge.

In this article, we present our multi-UAV approach to solve the autonomous construction task defined in MBZIRC Challenge 2. We describe our system and how we solved the different sub-problems involved in building a wall cooperatively with a team of multirotors, both from a software and a hardware perspective. First, we propose a multi-robot architecture for cooperative missions with physical interaction. The architecture is not restricted to autonomous construction, but it goes beyond MBZIRC and can be used for other multi-robot applications implying physical interaction, such as package delivery. Then, we present our particular implementation for MBZIRC, describing how the general architecture is tailored to the multi-UAV wall building scenario in Challenge 2. In particular, we detail all the specific software modules that were developed for the Challenge, and how they are integrated into our multi-purpose architecture. This includes perception algorithms for brick and wall detection, tools for multi-UAV coordination and planning and controllers for pick and place operations. Moreover, we describe the design of our UAV platforms, including the hardware devices for physical interaction (i.e., picking bricks and placing them on the wall).

Our team was ranked 4^{th} in the MBZIRC Grand Challenge, but we achieved the best performance in the autonomous wall building part of the Grand Challenge. However, in this article we go beyond the competition and propose a generic multi-UAV architecture with a set of technologies reusable for other applications involving physical interaction. In summary, our main contributions are the following:

- We explore the state of the art (Section II) about UAV systems and technologies for applications with physical interaction like construction or package delivery. Then, using autonomous construction as motivating scenario (Section III), we introduce our novel software architecture for multi-robot missions (Section IV). The architecture puts special emphasis on robustness and reliability, considering faulty UAVs and communication losses. Moreover, it is adaptable and flexible, so that different types of robots, perception/control methods and tasks can be accommodated.
- We describe our original hardware devices to accomplish physical interaction with UAVs in autonomous construction (Section V). In particular, we detail the design of our multirotor vehicles and their payload for picking, transporting and placing bricks. Note that despite the fact that this hardware was thought for wall building, it could be reusable with some adaptation for similar tasks like package transportation.
- We present our experimental results in autonomous wall building (Section VI), before and during the MBZIRC competition. Our methods were extensively tested in simulation and mock-up scenarios to evaluate their performance, but also in the actual arena of the MBZIRC final stage. This allowed us to also assess the robustness of our multi-UAV system out of a controlled environment, with challenging conditions for UAVs such as high temperatures, wind gusts, poor GNSS signal, etc.
- Finally, all our developments and experimental campaigns throughout the year previous to MBZIRC, and our participation in the final stage, allowed us to gain valuable experience in multi-UAV systems, which we capture in our lessons learned (Section VII) and conclusions (Section VIII).

II. RELATED WORK

In the last years, competitions have turned out to be a noticeable way of promoting the development of new technologies to cope with present robotics problems. There are multiple competitions organized worldwide every year, dealing with a wide spectrum of types of robots and challenges. The *Defense Advanced Research Projects Agency* (DARPA) of the United States has organized multiple well-known challenges since 2004, covering applications like autonomous driving [5], [6] and subterranean exploration,[2] among others. Other competitions, like RockIn [7] or RoboCup [8], propose indoor scenarios to pursue reproducibility and repeatability in robotics benchmarks. MBZIRC is also a recently created competition to foster outdoor challenges involving ground and aerial robots. In its first edition in 2017, MBZIRC brought together 25 teams from around the world to solve three challenges. Specifically, *The Treasure Hunt* required a team of UAVs to search, pick and deliver somewhere else a set of color disks. Many of the team approaches [9]–[13] developed grasping and coordination capabilities for the UAVs that could also be of interest for autonomous construction and transportation.

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

There are existing works that address multi-UAV applications with physical interaction with the environment, for instance for construction [1], contact-based inspection [2] or package delivery [3], [4]. Also in domains like environmental monitoring or search and rescue [14], [15], the use of multiple UAVs that can deploy and recover sensors is of interest. Cloud-based solutions is another domain where multi-UAV architectures can be beneficial [16]. If physical interaction is required, a quite relevant aspect is the ability to grasp objects with a certain level of accuracy. These grasping devices need to consider the weight and space limitations of UAVs, as well as the characteristics of the object to grasp. One solution is to use magnets to grab ferromagnetic objects (or objects with a ferromagnetic plate) [17], [18]. Other more general solutions consider using suction with a vacuum pump [19] or passive robotic hands [20]. Some authors have also developed more complex designs with lightweight dual-arm systems for aerial manipulation [21].

Grasping objects with UAVs requires first to detect and track the objects robustly, in order to control grasping devices precisely. Some authors [22] use monocular camera systems to implement visual servoing on the image plane. Other works [23], [24] address the problem using stereo cameras to learn feature-based models of the object to be tracked and grasped. Convolutional Neural Networks can also be applied to object detection, both using visual images [25] or RGB-D cameras [26].

In addition, cooperative teams with multiple UAVs working in a common space and with shared resources, need for coordination and conflict-resolution capabilities. A common approach is to solve a *multi-robot task allocation* problem [27], [28], assigning non-conflicting tasks to different UAVs in an efficient manner. For instance, a distributed approach is presented in [29] to coordinate multi-UAV tasks in dynamic scenarios. The method is applied to autonomous structure assembly. Recently, other participant teams in MBZIRC 2020 have published their approaches for multi-UAV coordination in autonomous construction. Authors in [30] propose two alternative algorithms for task allocation while building a wall, considering the existence or not of communication. A decentralized planning and coordination framework based on hierarchical task representation was also presented in [31] for an aerial-ground robotic team.

From the software architecture point of view, there is no clear standard to follow for multi-UAV systems. Several designs and implementations have been developed over the years, with different pros and cons for specific situations. As common guidelines, such an architecture needs to be open, flexible and adaptable to other scenarios; and it should include mechanisms to implement and integrate easily new features or modules. OpenUAV [32] and XTDrone [33] are simulation testbeds that only solve hardware abstraction, and they are bounded to the use of PX4 [34] and MAVROS.[3] AEROSTACK [35] is a full-stack system that proposes a multi-layered architecture. Even though the framework is quite complete, the imposition of its own libraries decreases the possibility of integration with other middleware. The *MRS UAV System* [36] is another remarkable full-stack system, including functionalities for multi-UAV localization and navigation. Apart from the architecture, the system contributes with a complete set of components for ready-to-fly multi-UAV teams. Both AEROSTACK and the MRS UAV System provide also simulation environments based on *Gazebo*. In this article, we contribute with a novel architecture based on hierarchical levels. Our aim is to present a flexible structure with little overhead to ease the integration of alternative algorithms and components.

[3]https://github.com/mavlink/mavros

III. PROBLEM STATEMENT

Our work in this article is inspired by the Challenge 2 of MBZIRC 2020, which consists of building two walls made of colored bricks with a cooperative team of multiple UAVs (up to 3) and a UGV. There are four types of bricks with the same cross-section ($0.2m \times 0.2m$) and different lengths and weights: red ($0.3m$, $1.0kg$); green ($0.6m$, $1.0kg$); blue ($1.2m$, $1.5kg$); and orange ($1.8m$, $2.0kg$). All bricks have one or several (three for orange bricks) rectangular ferromagnetic plates of white color on their top layer, in order to be grabbed by the robots.

Each trial has a maximum duration of 25 minutes, and at the beginning, teams are given randomly generated blueprints for the two walls to be assembled:

- *Wall 1*. This wall is to be assembled by the UGV, and it consists of 45 bricks arranged in 5 layers: 20 red, 10 green, 5 blue and 10 orange. The wall is L-shaped, with two segments of 4-meter length each. There is an L-shaped checkered pattern on the ground that indicates the location where the wall should be built.
- *Wall 2*. This wall is to be assembled by the UAVs, and it consists of 46 bricks arranged in 2 layers: 24 red, 12 green, 6 blue and 4 orange. The wall is a "W" shape made up of 4 segments of 4-meter length each, and the first segment is reserved only for orange bricks. Each wall segment has on top a U-shaped channel made of Perspex to place bricks, in order to reduce the effects of the wind generated by UAV rotors. Each U-shaped channel is mounted on a $1.7m$ hollow brick facade.

The arena consists of an outdoor flat area with dimensions $60m \times 50m$ (see Figure 1). GNSS is allowed although the use of RTK has a penalty of 25%. Each UAV has to fit into a $1.2m \times 1.2m \times 0.5m$ box. The use of larger platforms (up to $1.7m \times 1.7m \times 1m$) was allowed later, but with an associated penalty. A ground station can also be used to monitor and control UAVs through a 5 GHz Wi-Fi network provided by the competition organizers. As it will be seen in Section V, the UAV platforms and the hardware design to detect, pick and place bricks is driven by these technical specifications of the Challenge.

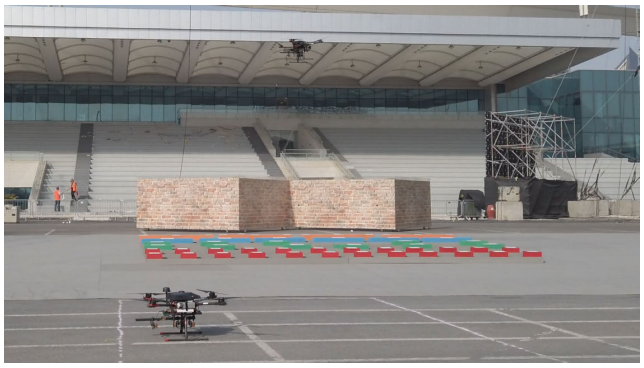


FIGURE 1. View of the arena for the MBZRIC Challenge 2. The structure to build Wall 2 has four segments, and the piles of bricks for the UAVs can be seen in front of the wall.

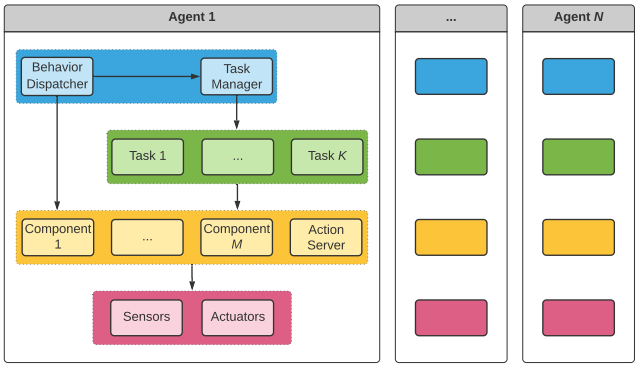


FIGURE 2. Scheme of the proposed multi-agent architecture. There are three different levels of software modules above the hardware level. Each Agent offers its own Actions and Tasks. Optionally, one of the Agents could be a Ground Station running centralized components for coordination.

Scoring is based on the number of bricks placed successfully on each wall. Depending on the brick color, 3, 4, 6 or 10 points are given; the larger the brick size, the more points are scored. Then, a penalty scheme is applied to the total score achieved, in order to account for bricks out of sequence at each wall segment. This means bricks placed in a different order than that specified in the blueprint. Moreover, a lower score can still be obtained for missions in manual mode, i.e., with any human intervention. More details about the scoring scheme and the Challenge description can be seen in the official MBZIRC website.[4]

At the beginning of each trial, the structures to build Wall 1 and Wall 2 are located at different places which are unknown for the multi-robot team. Bricks are arranged in piles per color, following a specific layout that is different for UGV and UAV bricks. Early versions of the rules promoted cooperation between the UAVs and the UGV more explicitly, but the final version establishes separated piles of bricks and walls for the UAVs and the UGV. Therefore, the only possible coordination is using UAV capabilities to localize faster the piles and wall for the UGV too. As the two problems are decoupled apart from that, we focus in this article on the solution for building Wall 2 with a multi-UAV team.

IV. SOFTWARE ARCHITECTURE AND MODULES

In this section, we propose a multi-agent architecture that is modular and flexible enough to be adapted to different scenarios. We present first the general architecture and concepts, and then, we describe all modules involved in our specific implementation for autonomous building in MBZIRC.

The proposed architecture is based on four main concepts: *Agents*, *Components*, *Actions* and *Tasks*. *Agents* are entities with perception and actuation capabilities through their onboard sensors and actuators, respectively. They can also communicate when possible and interact physically with the environment. They can be heterogeneous, i.e., each Agent offers a certain set of possibilities depending on its payload configuration. Therefore, the architecture enables the integration of UGVs and UAVs, as it was the case in MBZIRC.

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

Since we focus in this article on the multi-UAV cooperation part, our Agents will be UAVs from now on. Moreover, our system may have optionally a *Ground Station* hosting software modules, so we also consider it another Agent.

Components are software modules that provide specific functionalities implemented by tailored algorithms. Some of them run continuously in background, as for instance, those Components related with perception and estimation activities that provide information about the environment (e.g., an algorithm for object detection). Other Components just keep listening to run their algorithms on demand, as it is the case for certain planning activities; or to act timely, as it is the case for hardware actuator drivers. Nonetheless, Components may be activated or deactivated at any time, in order to save processing load or due to strategical 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., pick or place a brick). Also, each Agent offers a set of *Tasks*, which represent higher-level actions that can be executed by the Agent. Each Task has an undetermined duration and is implemented by means of a *Finite State Machine* (FSM), which encodes some kind of coordination and makes use of Components and Actions in order to accomplish its purpose. Indeed, Tasks are composable and can be made up of several sub-tasks to be executed sequentially. For instance, for the wall building case, we implemented a Task to pick a brick. That Task is in charge of sending the UAV to the pile of bricks, asking for permission to access (it is a shared resource) and calling the Action "pick", which controls physically the UAV to actually pick the brick.

Our architecture combines the aforementioned concepts to build the hierarchical structure with three levels that is depicted in Figure 2. At the lowest software level, 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 a Task. Note that, due to the

heterogeneity of the system, each Agent could run different Components or Actions.

At the medium level, each Agent exposes its own Tasks, which can be externally started, preempted or cancelled. By default, Agents are idle or hovering until any Task is invoked. The coordination of the mission takes place at the top level, through a software module called *Behavior Dispatcher*. This module implements the mission strategy by means of a script that describes the required steps to accomplish the mission. Thus, we follow somehow 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 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 centrally the mission, either from the Ground Station or from a specific Agent designed as leader. However, it is also possible to have multiple Behavior Dispatchers running on different Agents that are decoupled and do not need to cooperate among themselves (for instance, when communication is not available).

The architecture proposed is flexible and adaptable, as it offers an adequate level of abstraction. First, the concepts of Components and Actions allow us to abstract the system from specific hardware and functionalities, enabling the existence of heterogeneous Agents. Thus, Components and Actions can be easily added or replaced to incorporate different algorithms and functionalities or deal with new sensors and actuators. Second, Tasks can also be easily created or modified, shaping Agent behaviors according to mission specifications and design. Moreover, the possibility of using sub-tasks enhances composability and reusability in the architecture. Last, the overall strategy to solve missions is encoded in one or several Behavior Dispatchers, so the whole system behavior is adapted just configuring those modules. This is particularly relevant for multi-UAV systems operating in dynamic settings, as conditions may be variable, and mission designers need to be ready for different situations. Indeed, flexible and simple architectures are quite valuable for competitions like MBZIRC, where scoring schemes and rules can evolve even during the competition.

Other key aspects for the design of our architecture 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 bricks once the relevant ones are found. In the description of the particular Tasks implemented for autonomous construction missions, we will detail which failures were considered and how we handled them for each case.

Regarding communication, the architecture assumes unreliability and does not use blocking procedure calls so that 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 total communication loss and defined alternative behaviors in that case. More details about communication management in our multi-UAV system will be discussed in Section IV-E.

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 the architecture could be implemented in other frameworks. In particular, 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,[5] and Finite State Machines within the Tasks with the SMACH library.[6]

In the following sections, we describe a specific realization of our architecture for multi-UAV autonomous building. We provide details about the particular Components, Actions and Tasks required, as well as how the Behavior Dispatcher coordinates them until mission completion. This implementation was used in our participation in the wall building Challenge of MBZIRC and is available online as open-source code.[7]

A. COMPONENTS

This section describes the Components that we implemented for multi-UAV cooperative wall building in MBZIRC.

1) GRASPING PAYLOAD DRIVER

This Component runs on board each UAV and it is a driver to operate the hardware devices for pick and place actions (see Section V). As input, the driver can receive commands to magnetize or de-magnetize a magnet circuit to grab bricks, and to open or close the auxiliary gripper. The driver provides as output the status of the magnet (on/off), the gripper (open/closed) and the contact sensors (on/off). For a pick operation, the usual procedure is to switch on the magnet first, get closer to the brick until the contact sensor gets activated (that should be the ferromagnetic part of the brick touching the magnet), and then close the gripper. For placing an attached brick, the procedure is to de-magnetize and open the gripper simultaneously once the UAV reaches the right position to drop the brick.

2) COLOR-BASED BRICK DETECTOR

This Component runs on board each UAV and uses the color images from a camera pointing downwards to detect bricks. As output, it provides colors and 3D positions of detected bricks, relative to the UAV camera reference frame. The Component has two different operational modes: *detection*

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

and *tracking*. The detection mode is used while the UAV is navigating around the arena, for detecting new bricks. The tracking mode is used when the UAV is approaching a brick during a pick action. The Component can also be deactivated at certain moments to avoid false positive detections, e.g., when the UAV is already carrying a brick.

The first stage of the detection pipeline is common for both modes. First, an HSV (Hue-Saturation-Value) filter is applied for color segmentation. Each pixel is considered to be of a specific color if its HSV values fall within given HSV ranges describing that color. Those ranges are defined for each color (they should not overlap in the Hue dimension to distinguish bricks better) after tuning the detector by hand with sets of images containing the actual bricks. In particular, we configured a filter for each of the brick colors in the competition (i.e., red, green, blue and orange), but also one for the white color, to detect the inner white rectangle on the top layer of the bricks. After color filtering, some morphological operations (erosion and dilation) and 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. Moreover, 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. We also implemented a brick detector based on poinclouds generated from RGB-D images, but we experienced that its performance was decaying when the UAV was flying at higher altitudes, farther from the bricks. Therefore, as we realized during the first MBZIRC trials that brick colors were quite distinguishable w.r.t. the background color, we discarded the algorithm based on RGB-D images.

The detection pipeline finishes at this point for the detection mode, and the color elements found are used to feed the Component *Object Estimator*, which implements a centralized filter fusing detections from different UAVs to locate the piles of bricks. However, when attempting to pick a brick, this information was not enough, so we created the tracking mode with additional processing. In this mode, the color of the desired brick is set, and then, the closest list item with that color is selected for tracking. An algorithm based on pixel distance is used to track the detected item between consecutive image frames. Then, the detector also looks for a white region within the brick. Finally, position and orientation of the border between the brick color and the white region are provided as output, so that the UAV can use that measurement to place correctly the magnet on top of the white ferromagnetic plate. This is key when the UAV is so close that the camera can only see part of the brick. Figure 3 shows some examples of the brick detector working in the two operational modes.

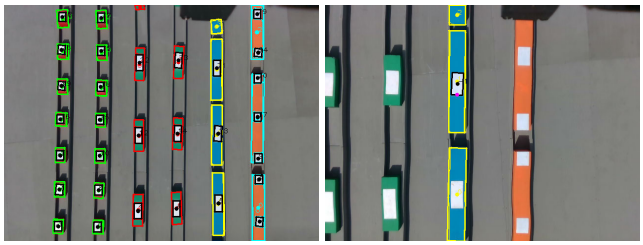


FIGURE 3. Output of the Color-based Brick Detector. Left, detection mode giving as output all colors found, including white rectangles. Right, tracking mode for blue color. All blue bricks are detected, but the closest one is explored to track the border with the white area (pink dot).

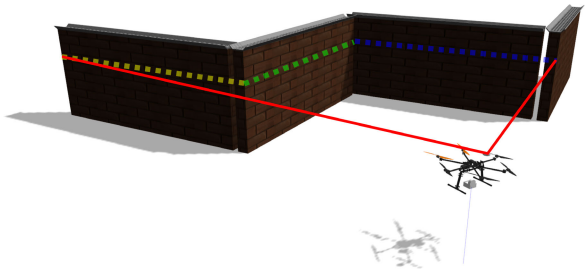


FIGURE 4. Visualization of the output of the Wall Detector. The 2D laser scan is processed to detect rectilinear segments (different colors) with the appropriate length.

3) WALL DETECTOR

The team of UAVs has to build its wall on a specific structure with 4 segments elevated a couple of meters from the ground, as described in Section III. Each of these wall segments has on top a U-shaped profile with a particular yellow color when seen from above. Therefore, the previous color-based detector would have been a first option to detect wall segments with the UAVs. However, we discovered that the walls had that color on their top layer on site, during the actual competition. As we did not have many trials to train and test that strategy for wall detection, we discarded it.

Another option to detect the wall segments is using depth images from an RGB-D camera, as segments have a clear elevation from the ground. However, we experienced that this sensor was not reliable enough to perform place operations, as it was partially occluded. The final solution that we used for wall detection was based on the readings from a 2D LIDAR mounted on top of each UAV. Using the points from the laser scan, we apply a RANSAC algorithm in order to fit straight lines. Then, we filter the possible lines by length, exploiting the fact that wall segments are known to be 4-meter long. A representation of the final output is shown in Figure 4. This wall detector is used both to localize wall segments in the arena during an initial exploration phase and to position relatively the UAV for placing bricks accurately.

In addition, it is worth mentioning that UAVs were also running onboard another Component to detect the wall for the UGV. As described in Section III, the UGV has to build its wall on top of an L-shaped pattern located on the ground and with known checkered colors. During the initial search phase, we use that Component to detect the L-shaped pattern and estimate its global position and orientation in the arena for the UGV. Basically, we use visual images from the UAV camera to segment the pattern colors and filter out

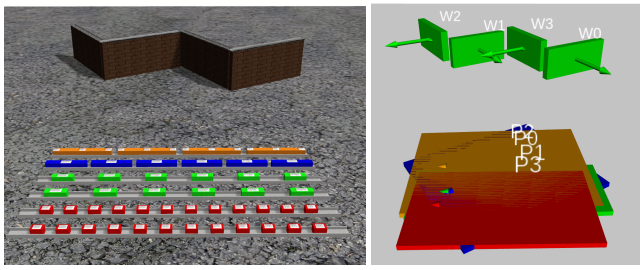


FIGURE 5. Left, view of a simulation of the arena, with the UAV wall and piles. Right, output of the Object Estimator: poses for the wall segments, and poses and colors for the piles.

regions by size, as the actual wall pattern dimensions are also known.

4) OBJECT ESTIMATOR

This Component consists of a stochastic filter to estimate the state of the objects in the arena that are relevant for the mission, namely piles of bricks and walls. The Component runs centrally on the Ground Station or the UAV designated as leader, and it receives observations from all the perception Components for brick and wall detection running on each UAV. The main function of this module is to fuse information coming from multiple detectors allocated on different UAVs, and to have an integrated estimation of the arena state. Moreover, the module acts as a filter to cope with noisy measurements, false positive detections, delayed observations, irregular detection rates, etc. We opted for a centralized filter because the number of UAVs was short and the communication bandwidth required to share their observations was not critical; while a decentralized approach may bring up issues related with inter-UAV communication and delays.

The output of this Component is a list of objects in the arena with an estimation of their state. There are three possible types of object: a pile of bricks with the same color, a wall segment for UAVs and the L-shaped pattern (this is estimated for the UGV). The state of each object consists of its 2D position in arena coordinates (z is assumed known), its yaw angle orientation, its 2D dimensions and its color (only for piles). All variables are continuous and estimated with a standard *Kalman Filter*; except for the color, which is a discrete variable $c \in \{red, blue, green, orange\}$, and it is estimated with a discrete *Bayes Filter*. Figure 5 depicts an example of the information provided by the Object Estimator.

Each time a detection is received, we apply a data association heuristic based on Euclidean distance (and color for the piles) to determine whether that observation should update an existing object in the list or a new object has to be created. As all objects are static, there is no prediction step in the Kalman Filter. For the update step, we use unitary matrices as observation models in case of wall detections. For brick detections, as we want to estimate the position of a pile containing multiple bricks of the same color, we apply a clustering operation to compute the pile centroid and dimensions taking into account all brick detections associated so far. The belief of the pile color is also updated with the observed brick color c_o, by means of a standard Bayes update:

$$p(c = i) = \eta \cdot p(c_o|c = i) \cdot p(c = i), \quad \forall i \tag{1}$$

where η is a normalizing constant and $p(c_o|c_t = i)$ is the probability of observing c_o given an actual color value $c = i$. We estimated empirically that the probability of detecting the actual color of a brick with our vision algorithm was of 0.9, which is the value that we used to compute the previous probability.

Finally, we include a couple of additional constraints in the Object Estimator to alleviate spurious detections. All observations located out of the limits of the competition arena or received with a delay longer than 2 seconds, are discarded. Moreover, objects that have been detected in less than 5 frames and have not been detected for 20 seconds, are considered false positives and removed from the list.

5) CONSTRUCTION PLANNER

This Component computes a plan to construct the wall, consisting of an ordered list of brick operations for the UAVs. As input, it receives a text file with the blueprint of the desired wall to build, which was provided by the MBZIRC organization before each trial. Then, this planner parses the file and generates a sequence of place operations, each one indicating the place position w.r.t. a wall segment reference frame and the brick color. This plan is used later by the Behavior Dispatcher to coordinate the different UAVs assigning them brick operations from the list in an ordered manner.

Our first strategy was to start placing the first brick of each wall segment alternatively, then the second, and so on until the completion of the bottom layer for all segments. After that, bricks belonging to the second layer would also be placed for each wall segment alternatively. The idea is to maximize the score by reducing the possibility of bricks conflicting with each other in the same segment, as the strategy of concentrating on completing each wall segment before going to the next one would create more difficulties. Nonetheless, this planner for construction can be modified in order to carry out different strategies depending on the situation. Indeed, during the competition, we adapted our initial strategy to focus first on green and blue bricks instead of red ones, as we wanted to optimize our score.

6) SHARED REGION MANAGER

As there are multiple UAVs navigating in the same scenario to pick and place bricks, we need to establish an autonomous system for conflict resolution in order to avoid collisions. This Component runs centrally on the UAV designated as leader or on the Ground Station, to handle conflicts. In our first implementation, each time a UAV wanted to navigate to another part of the arena, for searching, picking a brick or placing it, it had to reserve the 3D space region that was going to use; and then, no other UAV was allowed to reserve another overlapping region. Each reserved region was made up of

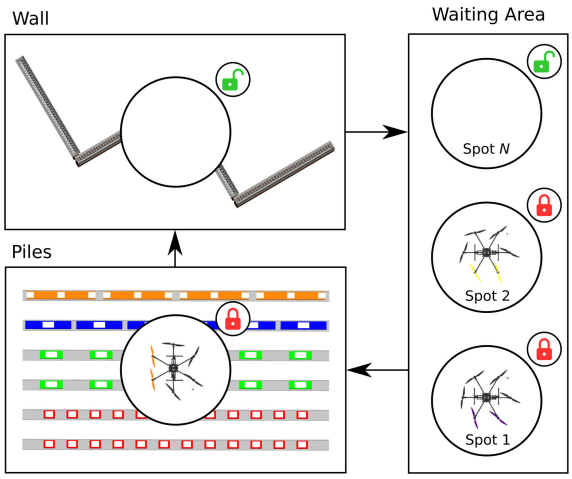


FIGURE 6. Procedure for a pick and place operation. The piles and the wall are shared resources (the wall is free in the example), and the UAVs can stay in one of the free spots of the waiting area while they get access. *N* waiting spots are used for a team with *N* UAVs.

a set of connected rectangular corridors. Thus, we avoided UAVs accessing simultaneously to the same critical places (e.g., a pile of bricks or a wall segment) and created safe corridors for navigation through the arena. Also, if access to a region was not granted, the corresponding UAV would try an alternative route or wait until the conflicting region is released. This multi-UAV coordination is implemented at Task level, so the different Tasks running on the UAVs are in charge of reserving and releasing these shared regions when accomplishing their activities.

This system for space management was successfully tested in simulation, but during the actual competition, we implemented an alternative procedure to achieve a more efficient performance. In particular, we defined as shared resources the piles of bricks and the wall. We also established a waiting area next to them, to queue UAVs before they get access to the piles and after they place a brick. Each time a UAV wants to pick a brick, it first requests access to the piles area, and once the Shared Region Manager (autonomous software component) gives permission, the UAV moves there. Then, it releases its previous spot in the waiting area, and tries to pick the brick. In order to place a picked brick, the UAV waits until the wall is free; and moves there after permission is granted, freeing then the piles area. Once the brick is placed, the UAV locks a spot in the waiting area, moves there, and frees the wall area. We added the waiting area in order to avoid deadlocks between two UAVs trying to access each other locked shared resources, and there are N spots for teams with N UAVs, so they can never get blocked waiting for a free waiting spot. Figure 6 summarizes this procedure to access shared resources. Last, for the case of UAVs navigating through the arena, we realized that potential collisions could mostly occur during the initial search phase to localize piles and walls. Therefore, we used different non-overlapping paths for each UAV during that initial exploration to avoid conflicts.

7) UAL

There is a special Component implementing an abstraction layer to operate each UAV, called *UAV Abstraction Layer* (UAL). UAL is an open-source[8] library [37] that was developed in our lab to ease the integration of different types of UAV autopilots. Thus, 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 [34] or ArduPilot[9] underneath our Component UAL, using the corresponding state estimators and controllers integrated in both autopilots. We configured and tuned these 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.

B. ACTIONS

Each UAV is able to perform a set of low-level Actions that are handled by its Action Server running onboard. These Actions control the UAV motors to navigate or its grasping device for physical interaction with the bricks. We implemented the following Actions for the wall building Challenge.

`Take-off/Land/GoTo`

These basic movement Actions are in charge of taking off the UAV at a certain altitude, landing it or controlling it in order to navigate to a given waypoint in the arena. For that, our UAL is used underneath.

`Pick`

This Action receives as input a given color, and it controls the UAV to pick the closest brick found of that color. The Action is to be called when the UAV is flying over the pile of bricks, so it should be able to detect bricks of the specified color. First, the Brick Detector is switched to *tracking* mode to track the closest brick. Simultaneously, the magnet of the grasping mechanism is magnetized and the gripper opened. Then, a velocity controller is activated to descend the UAV while it aligns the gripper with the brick and centers the magnet with the white ferromagnetic plate. For that, feedback from the laser altimeter and the Brick Detector Component are used. Moreover, the descending velocity is inversely proportional to the displacement error w.r.t to the brick center.

There are two sets of parameters for velocity control, occurring the commutation between them at a given altitude level, with a certain hysteresis. The *upper* controller is softer, as its objective is to approach the brick while maintaining it within the camera image; the *lower* controller is more aggressive, as a higher accuracy is needed at the final phase and the UAV has to deal with the ground effect. Right after the white rectangle starts getting cropped on the image, the UAV orientation is fixed, and small angular misalignments with the

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

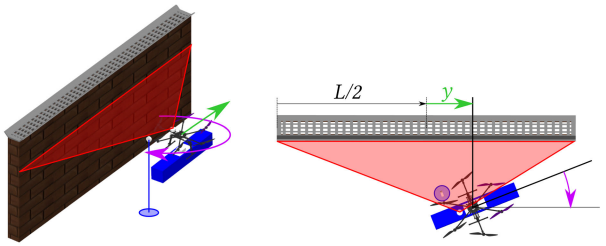


FIGURE 7. UAV alignment before a place operation: perspective (left) and top view (right). The UAV flies at low altitude to detect the wall segment and orientates parallel to it. Simultaneously, it moves longitudinally to reach the desired offset (y) regarding the wall segment middle, where the brick has to be placed. L is the total length of the wall segment.

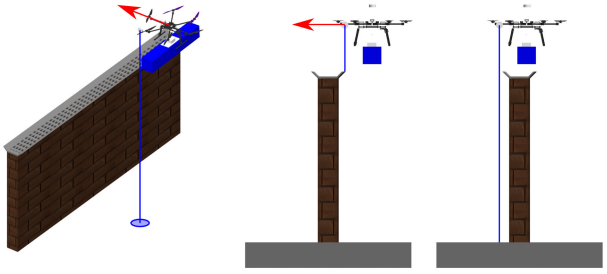


FIGURE 8. Sequence to place a brick. The UAV moves laterally over the wall (left image) until and it detects a first (middle image) and second leap (right image) in its altimeter reading. Given the position of the altimeter, the second leap indicates that the UAV is near the right position to drop the brick.

brick are corrected later when closing the gripper. Moreover, if the tracked brick is lost on the image for some time, the UAV starts ascending until the brick is found again or a maximum altitude is reached, which causes the Action to finish reporting a failure. Otherwise, as soon as the contact sensors in the grasping mechanism detect the brick, the gripper gets closed, grasping the brick. The UAV is commanded to go up to a given safe altitude, and the Action is considered successful.

`Place`

This Action assumes that the UAV is holding a brick and receives as input the longitudinal offset w.r.t a wall segment where the brick should be placed (e.g., zero offset means the middle of the wall segment). When the Action is called, the UAV is expected to have the corresponding wall segment to its left, which will be ensured at Task level. This last precondition is key, because the procedure to release the brick on top of the wall relies on the asymmetry of the brick position on board the UAV with respect to the laser altimeter.

The Action starts descending the UAV to a predetermined altitude where wall segments are visible for the Wall Detector (i.e., for the 2D LIDAR). Then, the most centered segment to the UAV left is targeted to place the brick; and the Wall Detector output is used to keep the UAV orientation aligned with the segment while it moves longitudinally, to reach the desired offset w.r.t. the segment's middle, and perpendicularly, to reach a known safety distance from the wall. This procedure is depicted in Figure 7. If the wall segment is lost or never detected, the Action ends reporting a failure. Once the UAV is well positioned, it is commanded to ascend to a predefined altitude higher than the wall, and approach it moving to its left. The laser altimeter is used to detect both edges of the wall segment, thanks to the two corresponding steps in the readings (see Figure 8). After the second leap in the altimeter reading, the UAV centers the brick laterally with the wall edges in open loop, demagnetizes the magnet, opens the gripper, and releases the brick on the wall. In case the wall is not detected after some sensible distance is travelled, it is considered lost and a failure is reported. The overall procedure is depicted in Figure 10.

C. TASKS

Apart from its Actions, each UAV is also able to perform a set of higher-level Tasks. These Tasks are exposed so that they can be called from the Behavior Dispatcher, regardless of its location, as it may run on that UAV too or somewhere else. Each Task consists of a FSM and makes use of the functionalities offered by the UAV Actions and Components. Tasks differ from Actions in the sense that they add on top a layer of intelligence and coordination. Actions execute physical operations with a single UAV by using its controllers, whereas Tasks can include additional behaviors related to mission strategy, failure handling or multi-UAV coordination. We implemented the following Tasks for the wall building Challenge.

`Take-off/Land/GoTo`. These Tasks call the corresponding Actions with the same name. However, if the Action fails, the Task waits a certain time and retries. Also, before/after moving to a given waypoint, access can be requested/released to the Shared Region Manager.

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

`Pick`. This Task receives as input the color of the desired brick and the corresponding pile position in the arena. Figure 9 depicts the FSM. In our final implementation, the Shared Region Manager handled a waiting area to access the piles. Therefore, the Task starts with the UAV in the waiting area and asking the Shared Region Manager for access to the piles area. When the UAV is given permission, it is sent above the pile, and once there, it frees its previous spot in the waiting area. Finally, a `Pick` Action with the desired color is called. The outcome of this Action is also the final outcome of the Task; but in case of failure, before finishing, a spot in the waiting area is locked, the UAV is sent there and the piles area released.

`Place`. This Task receives as input the position of a wall segment in the arena and the offset regarding its center to place the brick. The Task starts asking the Shared Region Manager for access to the the wall. Once granted, the UAV is sent to a position that leaves the target wall segment to its left, the piles area is freed, and the `Place` Action is called.

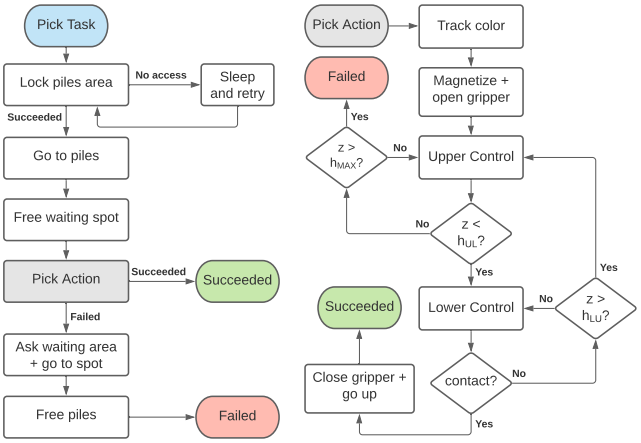


FIGURE 9. Left, FSM for the `Pick` Task. Right, flowchart of the `Pick` Action, which is called from the `Pick` Task. The switching between upper and lower controls is triggered by the altitude thresholds h_{UL} and h_{LU}. The Action fails if the UAV ascends above h_{MAX} without detecting a brick.

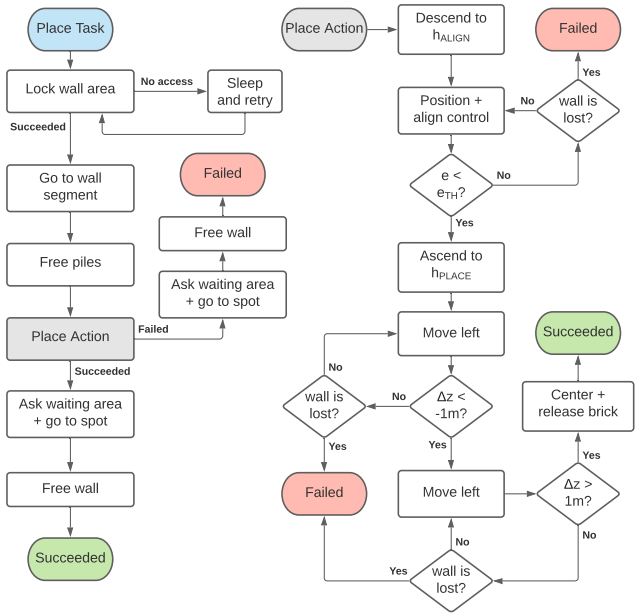


FIGURE 10. Left, FSM for the `Place` Task. Right, flowchart of the `Place` Action, which is called from the `Place` Task. The UAV first descends to $h_{eqnarray}$ to see the wall with the 2D LIDAR. When the overall error in position and angle is smaller than a given threshold e_{TH}, the UAV ascends at h_{PLACE} to place the brick. Then, two consecutive abrupt changes Δz in the readings from the laser altimeter are used to detect the wall edges and release the brick. If the wall is lost, the Action reports failure.

The Task reports success if the Action succeeds and failure otherwise. In both cases a spot in the waiting area is locked to send the UAV, and the wall area released before finishing. A representation of the FSM for the `Place` Task is depicted in Figure 10.

D. BEHAVIOR DISPATCHER

We implemented a Behavior Dispatcher for wall building that ran on the Ground Station and coordinated our multi-UAV team in a centralized fashion. Figure 11 summarizes the

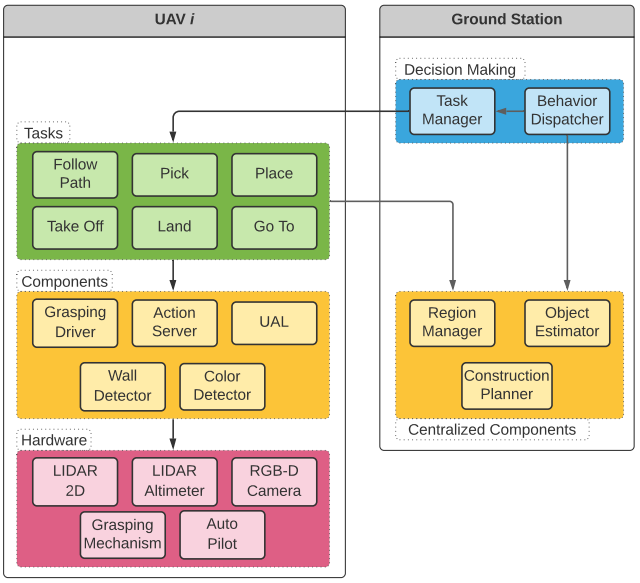


FIGURE 11. Specific implementation of our multi-agent architecture for the wall building challenge of MBZIRC.

final implementation of our architecture for the MBZIRC Challenge, including all modules developed. Before starting the mission, the Behavior Dispatcher discovers which UAVs are available by monitoring the wireless network. Then, given the number of UAVs (up to three in the competition trials), it implements the following behavior.

First, an exploration phase to search for piles and wall segments is accomplished by the UAVs. For that, the Behavior Dispatcher activates the Components for brick and wall detection on board the UAVs, and then, it starts a `FollowPath` Task for each UAV. The arena is divided into as many regions as UAVs, and non-overlapping paths are assigned to each UAV in order to cover all sub-divisions (Figure 12 depicts an example with two UAVs). The altitude level was configured by hand as a trade-off between field of view and brick detector performance. If the wall or any of the piles of bricks for UAVs are not found, the same paths are repeated sequentially. Also, the `FollowPath` Tasks are preempted in case the piles and wall are discovered before completion. Note that the Object Estimator keeps track of the detected wall segments separately. Therefore, we run periodically a clustering algorithm to group segments that are close enough and connected with 90° angles, to find the 4-segment, W-shaped wall as a whole. Once this is achieved, the wall is considered detected and a wall reference frame is defined for each of the segments, so that UAVs can position bricks relative to them.

After the exploration phase, UAVs start building the wall. The Behavior Dispatcher uses the Construction Planner to create a sequence of brick operations. This list is treated as a queue of tasks that need to be allocated to UAVs. Thus, until the end of the mission, the Dispatcher keeps extracting items from the queue and assigning them to idle UAVs, choosing the closest one if more than one is available. For that, the Behavior Dispatcher calls the `Pick` Task of the

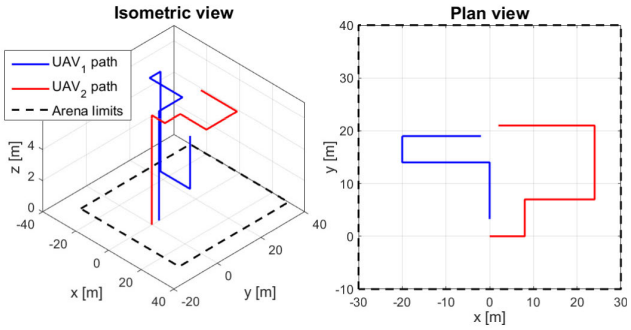


FIGURE 12. Example paths for an exploration phase with 2 UAVs. One of them includes a segment with lower altitude in order to detect the wall with the UAV 2D LIDAR.

corresponding UAV. If the Task fails, the brick item is returned to the queue as a pending operation and the UAV gets back to idle. Otherwise, the corresponding `Place` Task is called. If this Task fails, it means that the UAV did not find the corresponding wall segment. The brick item is returned to the queue as a pending operation and the UAV gets back to idle after dropping the actual brick.

Finally, recall that multi-UAV conflict resolution is performed at Task level. UAVs, when executing `Pick` and `Place` Tasks, request the Shared Region Manager access to the corresponding pile or wall segment. Moreover, during the building phase, each UAV flies at a different altitude while moving through the arena toward the pile or the wall, in order to avoid collisions.

E. COMMUNICATION

One key aspect in multi-UAV systems is communication, as using unreliable wireless channels is commonplace. We designed our system to be robust in this sense, making it as less dependant as possible on the existence of reliable communication. We did that increasing the autonomy of each UAV (relevant computation is mainly performed on board) and reducing the exchanged information only to what is really necessary. In general, our strategy is using communication channels when available to improve performance through coordination, but also articulate alternative behaviors in case of communication loss.

As it has already been explained, there are only three modules in our system that require inter-agent communication for UAV coordination, namely the Object Estimator, the Shared Region Manager and the Behavior Dispatcher. The remaining modules do not receive information coming from other Agents. In the MBZIRC competition, we ran these centralized modules on our Ground Station because the Wi-Fi communication link was stable, but note that the system would work in a transparent manner if they were located in one of the UAVs acting as leader.

Our multi-UAV team handled communication using the ROS master scheme. However, we used the ROS package *multimaster-fkie*[10] to improve system robustness. ROS relies on a single central master, which limits communication. Instead, the multi-master package allows each UAV to run its own ROS master, and it manages inter-master communication. In the event of a communication failure in one of the UAVs, others would keep working. The approximate bandwidth to communicate all necessary data from a UAV to the Ground Station was less than $30KB/s$ most of the time, going up to $190KB/s$ when detecting all the bricks. This datastream was mainly coming from brick and wall detections, UAV position, UAV Tasks status and UAV requests and releases for shared regions. The definition of all data packets exchanged by the modules in our architecture is on the *mbzirc_comm_objs* package that can be found in our public repository.[11]

In our first participation in MBZIRC (2017), we came across communication issues with the Wi-Fi network provided by the organizers. Thus, this occasion we prepared our system so that it could also run with no communication at all. First, when starting each trial, the Behavior Dispatcher was only taking into account for the overall strategy those UAVs responding with their status. Second, in a hypothetical event of total lack of communication, an instance of the Behavior Dispatcher could run on board each UAV, without coordinating with others. Each UAV would only consider its object detections, and shared resources (i.e., piles and wall segments) would be accessed in predetermined time slots. This approach is safe but less efficient, although we did not have to use it during the competition trials, as there were no serious issues with the Wi-Fi network this edition.

V. HARDWARE SYSTEMS

In this section, we provide details about our hardware systems for the MBZIRC Challenge 2. We describe the hardware architecture of our multirotor platforms, and then the grasping mechanism. Although the designed is tailored to the wall building Challenge of MBZIRC, the concept could be adapted for picking and transporting other similar objects.

A. AERIAL PLATFORM

Our aerial platforms were multirotors made by the local company Proskytec (see Figure 13), based on the DJI E2000 propulsion system. Each UAV weights $6.5kg$ (including batteries and avionics) and can carry a payload of $3.5kg$. These UAVs allowed us to fulfill the main physical requirements for the MBZIRC Challenge: (i) they fit within the size restrictions imposed by the organizers without penalty, $1.2m \times 1.2m \times 0.5m$; (ii) they can transport the heaviest bricks (orange, $2.0kg$); and (iii) their operational time in flight while picking and placing bricks is over 12 minutes. This is nearly half the maximum Challenge time, so we only needed to change batteries once during each trial.

Each UAV carries a Pixhawk Cube 2.1 autopilot and a Here+ RTK GNSS receiver for navigation. We also added the required sensors to accomplish this particular Challenge.

[10] https://github.com/fkie/multimaster_fkie

[11] https://github.com/grvcTeam/mbzirc2020

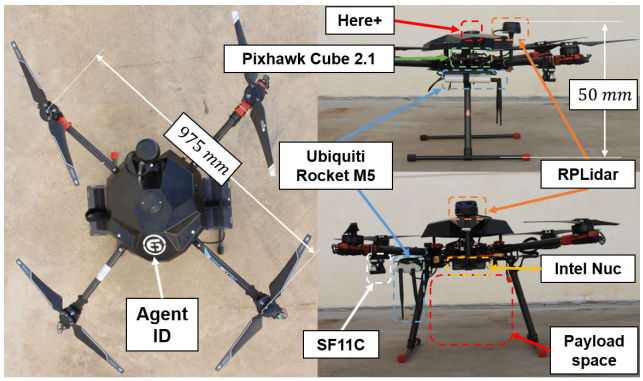


FIGURE 13. Several views of the aerial platform indicating the spatial distribution of the equipment on board.

Figure 13 shows the onboard equipment and its spatial distribution. An Intel NUC-i7 with 16GB RAM performs computation onboard, and it is connected to all other devices through USB ports. Its original metallic case has been replaced with a custom-made plastic case to reduce weight. An Intel RealSense D435i RGB-D camera is located pointing downwards for brick detection; and a Slamtec RPLIDAR A3 is placed on top of the UAV poiting forward, to detect the wall structure when flying below 1.5m. This sensor is connected to the onboard computer using a serial TTL UART to USB converter. Also, there is a Lightware SF11C laser altimeter that is used for altitude control and to detect wall edges during place operations. Last, a Ubiquiti Rocket M5 5.8GHz radio link is used for communication with the Ground Station.

B. GRASPING MECHANISM

All UAVs are equipped with the same grasping mechanism to pick and transport bricks. Competition bricks (see Section III) had lengths ranging from 0.3m to 1.8m and weights between 1kg and 2kg, as well as a ferromagnetic plate on top (three the largest ones). Even though bricks could be picked using a magnetic gripper, given the issues that we found with electromagnetic grippers during our participation in MBZIRC 2017 [13], we decided to design a more reliable hybrid grasping mechanism, combining magnets with a contact gripper. The system, shown in Figure 14, consists of a carbon fiber bar with a magnet device attached, hanging in the middle, and a gripper made up of three parallel pincers. This solution is heavier than a simple magnetic gripper, but in our preliminary tests it turned out to be much more reliable, mainly with bigger bricks. The magnet holds the brick, but the pincers align it and grasp it firmly during flight, so that it does not detach with vibrations.

The magnetic component is based on the commercially available Magswitch MagJig 95 device. This device consists of two permanent magnets, one fixed and another that can be rotated using a mechanic switch. A magnetic field is activated or deactivated when the relative position of the two permanent magnets is changed. Figure 15 shows our design to couple the Magswitch to a Hitec HS-225BB servomotor for automatic rotation. Two small switches are also included for contact

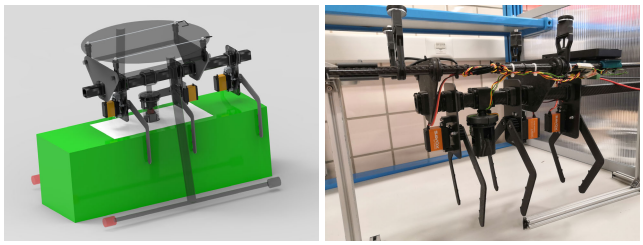


FIGURE 14. Left, rendered view of the grasping mechanism with a green brick. Right, real implementation.

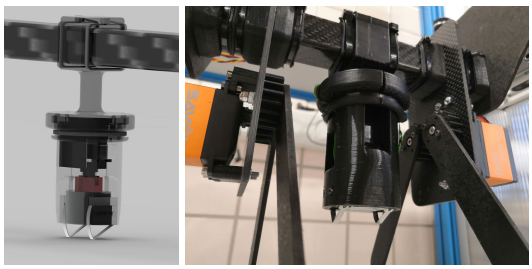


FIGURE 15. Final design integrating the Magswitch with a servomotor and the limit switches within a plastic (PLA) cylinder. Left, rendered view; right, actual implementation.

detection with the bricks. The whole magnetic component is attached to the main carbon fiber bar with a piece made of thermoplastic polyurethane (TPU), as this material is flexible enough to allow for compliant grasping of the bricks.

The other component of our grasping mechanism is a gripper consisting of three pincers attached to the main carbon fiber bar. Apart from aligning the brick correctly with the UAV frame, this gripper adds robustness to the grasping, allowing for failures in the magnet. Each pair of pincers has a Savox SA-1230SG servomotor for operation and three saw-teeth to increase their holding capacity. The design is such that grasped bricks can slide smoothly when the pincers open.

The whole grasping mechanism is powered by an independent 3S LiPo battery and controlled by a custom-made electronics board. This control board can be seen in Figure 16 and is based on an Arduino Nano connected to the main onboard computer through a serial port to send sensor readings and receive commands for the actuators. In particular, the board reads the contact switches in the magnet device, and it has outputs for the four servomotors, the one rotating the magnets and those operating the pincers.

VI. EXPERIMENTAL RESULTS

This section presents our experimental results testing the multi-UAV system for wall building. The evaluation includes results in our simulator as well as tests in mock-up scenarios before the competition. Our team performance during the final stage of MBZIRC is also summarized. The experimental workflow consisted of testing all software modules in simulation before conducting actual flights. However, as we considered that testing pick and place operations outdoors was critical for the final performance in the competition,

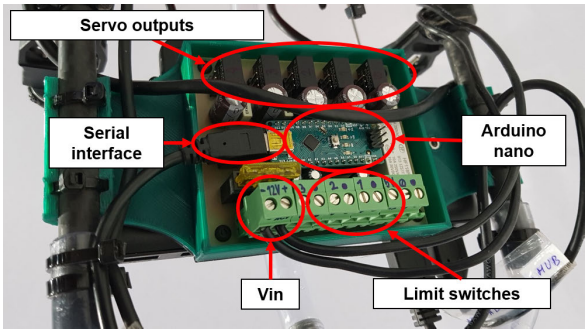


FIGURE 16. Control board for the grasping mechanism.

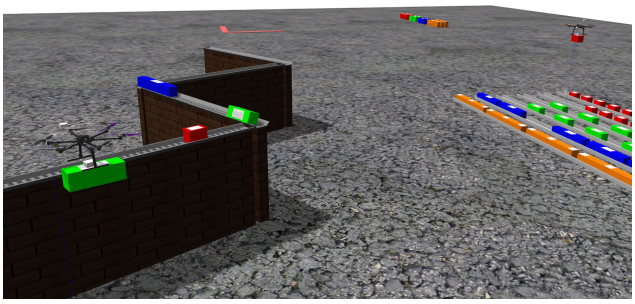


FIGURE 17. Image of an example simulation with the UAVs picking and placing bricks on the wall.

we concentrated first on setting a basic mock-up scenario for that. Afterwards, we interleaved simulation and real experiments to converge toward reliable multi-UAV missions.

A. SIMULATION EXPERIMENTS

We developed a simulation of our multi-UAV system, following the specifications of the actual MBZIRC arena. Figure 17 shows an image of one of our simulations. The simulation is based on Gazebo and it uses the *Software-In-The-Loop* (SITL) framework of PX4,[12] which allows us to also simulate the autopilot firmware with the rest of the system. The main objective is to test extensively the multi-UAV architecture without risking the actual platforms, in order to ease the integration of all software functionalities. Actually, thanks to the SITL, we can even perform quite realistic simulations of pick and place operations. Through our simulator, we validated mainly the modules related with the overall strategy and multi-UAV coordination, namely the Object Estimator, the Shared Region Manager, the Construction Planner, and the interaction of the Behavior Dispatcher with Actions and Tasks through the Action Server and the Task Manager, respectively.

Numerous hours of simulation were conducted to test every relevant change in software modules before and during the competition. In order to assess the performance of the whole system, we ran a batch of 20 simulations of 25 minutes each (maximum duration of the actual Challenge). We used two UAVs not to overload the simulation. The results are

[12]https://github.com/PX4/sitl_gazebo

TABLE 1. Performance metrics to evaluate the system in simulation. Average values over 20 simulations of 25 minutes with two UAVs are shown.

	`Pick` Action	`Place` Action
Brick count	16.95	16.00
Error rate [%]	0	6.5
Action duration [s]	19.18	74.08

presented in Table 1. No errors (i.e., crashes) occurred during pick operations, and UAVs kept trying until the brick was successfully picked. We include the average duration of each of the Actions; `Place` Actions take longer, as UAVs move slower with the bricks and they have to get aligned with the wall. Regarding the errors of `Place` Actions, they were mostly due to collisions with already placed bricks, which led to some of the bricks falling to the floor. The total number of bricks of any color picked and placed is also measured.

In summary, our system reached the necessary level of reliability in simulation to run multi-UAV missions in the actual arena with a reasonable safety margin. Despite the fact that simulations were pretty realistic, specially due to the SITL autopilot, there were no external color artifacts that could deceive our brick detector, and we did not simulate wind nor aerodynamic effects (such as ground effect) that could complicate flight control. Therefore, we still needed to calibrate thoroughly the UAV controllers to pick and place bricks in a real environment. For that, we carried out the mock-up experiments described in next section.

B. MOCK-UP EXPERIMENTS

We built mock-up facilities for the wall building MBZIRC Challenge next to our lab at the University of Seville. The scenario was large enough to fly three UAVs simultaneously and had bricks and a movable mock-up wall. Our main objective was to validate key parts of the system with our aerial platforms, mainly those involving brick/wall detection and physical interaction. With these experiments, we were able to adjust the autopilot controllers for our UAVs, adapt the mechanical design of the grasping mechanism after preliminary tests, and calibrate the systems and controllers involved in `Pick` and `Place` Actions.

Figure 18 shows several views of our mock-up scenario. Our bricks were made of *expanded polystyrene* with the same specifications (including weight) as the MBZIRC rules. We added some extra weight to the bricks with a 0.3*mm* steel sheet to meet the specifications. We also built a replica of a wall with two perpendicular segments, following MBZIRC dimensions. Moreover, in our first pick trials we discovered that bricks were moved away due to the UAV airflow. Therefore, we devised a wooden rail to place bricks on the ground and diminish that effect. Indeed, after we warned the MBZIRC organizers about this issue, they decided to use a similar design for the actual competition.

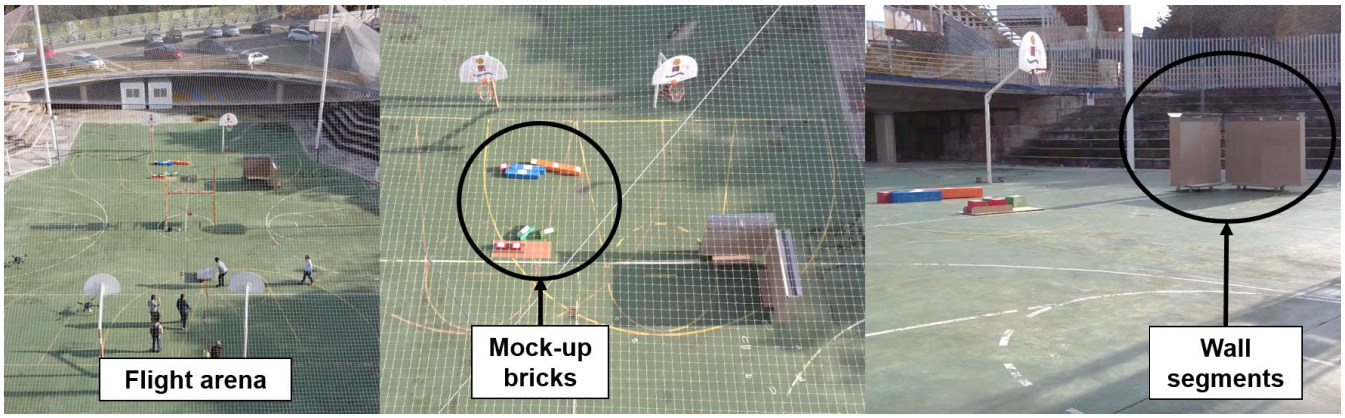


FIGURE 18. Different views of our mock-up scenario. There is a safety net covering the arena.

1) RESULTS FOR BRICK DETECTION

One of the most critical modules to be tested in real conditions is the Color-based Brick Detector. We trained this Component using images from each of the color bricks in our mock-up scenario, in order to use it testing `Pick` Actions. We were able to detect bricks consistently for all colors except for green ones, as the floor of our arena was green too. Given these issues with green bricks, we did not use them for pick experiments.

Since colors and lighting conditions were slightly different in the actual MBZIRC arena, we repeated the training of the Brick Detector when we arrived there. In order to assess the accuracy of the Brick Detector, we recorded a dataset with more than 1,800 frames (95 seconds) in the MBZIRC arena, with the UAV flying above the pile of bricks at an altitude of around 10m. The dataset contained bricks of all colors considering different lightning conditions (as rehearsals were scheduled at different times of the day) and we ran our detector for each of them. According to the results, there were no significant errors in our Brick Detector. The illumination conditions were stable during the competition, and the colors in the MBZIRC arena were vivid enough, so we were able to tune the detector adequately to distinguish all colors even at different times of the day. We only observed detection failures in less than a 1% of the frames, mainly red bricks missed, which were the ones with less area on the image. Another error that we observed was fusing two blue bricks as one detection, as they were stacked closer in the rail. In Figure 3, it can be seen the pile layout, and how red rectangles are smaller and blue ones are closer. In any case, these errors were just spurious and were not really an issue for pick operations.

2) PICK AND PLACE OPERATIONS

We considered that performing physical interaction reliably with the UAV was essential to succeed in the final Challenge. Therefore, we put a lot of efforts in testing extensively `Pick` and `Place` Actions in our mock-up scenario. We conducted more than 20 pick and place Actions using bricks of different colors, achieving a success rate of 84% in pick operations and 75% in place operations. We consider a pick operation as a failure when the UAV became unstable trying to pick the brick and there was a need for manual intervention. A place operation is considered a failure if the brick was not dropped on top of the wall but outside. Figure 19 shows various sequences of a UAV picking bricks of different colors in our mock-up experiments. Even though we were able to pick orange bricks, the UAV was quite close to its payload limit, so we eventually decided not to try orange bricks in the actual competition to reduce risks.

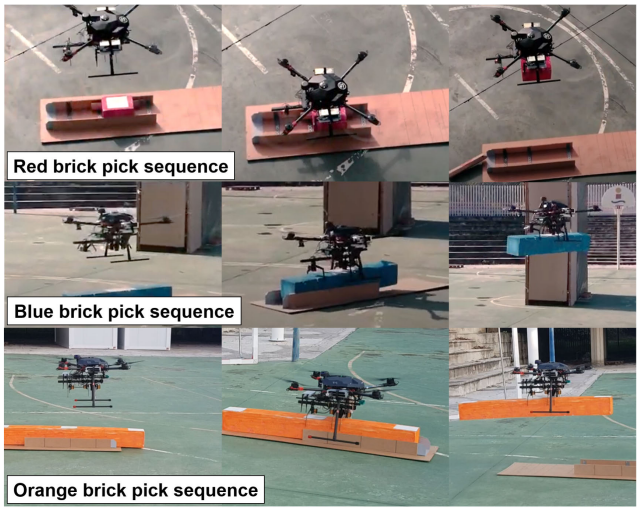


FIGURE 19. Sequences of images of the UAV picking bricks of different colors in Seville.

A major issue that we found in our first tests picking bricks was that the UAV could get unstable if the gripper got somehow stuck in the rail containing the bricks. Then, we decided to include a passive accommodation system in order to provide a certain level of elasticity and force accommodation to the whole grasping mechanism. This system helped us to conduct pick operations even in those cases in which the brick was not totally horizontal, as it happened during the competition. Figure 20 shows how the grasping mechanism can be accommodated to the brick.

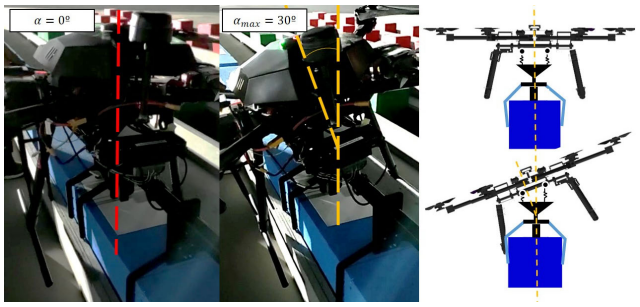


FIGURE 20. The grasping mechanism has certain elasticity to accommodate when picking a brick. A maximum angle of 30° can be reached.

Figure 21 depicts an example of a UAV performing a complete pick and place autonomous operation. The UAV trajectory is shown from the moment that it starts a `Pick` Action for a red brick, until it performs the `Place` Action. The transitions between different Actions are indicated with numbers in the figure. The experiment starts with the UAV above a red brick and a `Pick` Action is commanded. At that moment, the velocity controller starts to control the horizontal and vertical speed of the UAV, trying to center the grasping mechanism with the center of the brick while descending. The outcome of the velocity controllers (Figure 21, right) shows that the pick operation lasts 23 seconds, and once the brick is picked the UAV starts ascending with the brick until point ① is reached. Then, the UAV is commanded a `GoTo` Action to fly close to the wall, whose position was previously detected in a search phase. When the UAV is close to the wall (point ②), the `Place` Action starts: the UAV descends until it detects the wall with the 2D LIDAR; then, it gets aligned parallel to the wall while positioning itself correctly, it ascends again, flies over the wall to detect the edges with the laser altimeter, and drops the brick (point ③). After that, the `Place` Action is over and the UAV ascends to point ④ before starting a new pick operation over the pile of bricks. A video summarizing some of our main experiments for picking and placing bricks can be found at https://grvc.us.es/downloads/videos/MBZIRC-CH2.mp4.

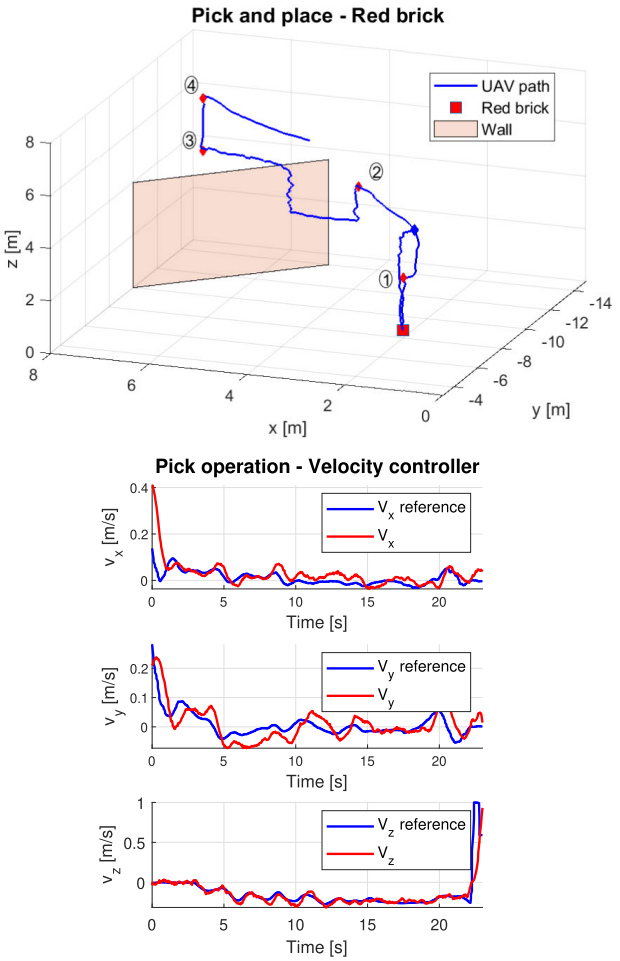


FIGURE 21. Detail of an experiment picking and placing a red brick. Top, 3D trajectory of the UAV, which starts at the blue diamond marker. Bottom, output of the velocity controllers during the pick operation.

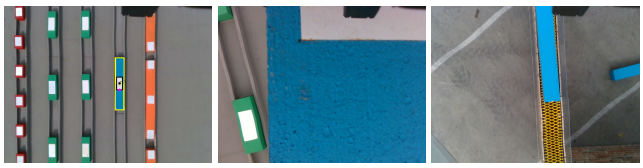


FIGURE 22. Sample images from the camera on board the UAV while picking and placing a blue brick during one of the MBZIRC trials.

C. RESULTS FROM THE MBZIRC COMPETITION

In Abu Dhabi, we spent ten days integrating and testing our system before the final trials of the competition. Although the organization provided an indoor arena for UAV testing, we could not use it to adjust our UAV controllers for pick operations, as our UAVs relied on GNSS for positioning and autonomous flight. Therefore, we opted for going to an outdoor airfield owned by the Abu Dhabi Radio Control Club. This opportunity turned out to be crucial, since we were able to test our UAVs after the initial mounting and tune our autopilot controllers.

The competition lasted six days. The three first days were for rehearsals, then we had two trials for Challenge 2 (wall building) and one single trial for the Grand Challenge. We used the first rehearsals to record logs with images of the bricks and test the detection of the wall. Brick colors and illumination conditions were slightly different from those in our mock-up experiments, so we trained our detectors again. During the last rehearsal, we were able to place two blue bricks in complete autonomous mode. Since we discovered that our team was the only one able to pick bigger blue bricks, we focused on these bricks, as their score was higher. Figure 22 shows a sequence of images taken from one of our UAVs picking and placing a blue brick in the MBZIRC arena.

During the two official trials, however, we had hardware issues with the UAVs. The rails with the bricks were slightly higher than expected and our grippers were hitting them. Despite this, we managed to pick several bricks, but we

were not able to place them successfully. It seemed that the batteries were not working properly, and the UAVs went down after a few minutes flying, without having enough time to perform the complete mission. In the Grand Challenge, we had another opportunity to test our system for wall building. The Grand Challenge consisted of a mixture of three different Challenges simultaneously, so it was even more complex to achieve a good result. This time, we changed our strategy and tried to go for green bricks (they were lighter), and we managed to place one brick on the wall autonomously. We picked another one, but time ran out before arriving to the wall. In fact, we were the only team that was able to place a brick autonomously during the Grand Challenge, so we achieved the best performance for wall building, and we ranked 4^{th} for the overall Grand Challenge.

VII. LESSONS LEARNED

We believe that the choice of our architecture for multi-UAV challenges was a success. On the one hand, it can be adapted to other potential applications due to its flexibility. Indeed, we used the same architecture to implement other Challenges in MBZIRC. On the other hand, we reached an optimal trade-off between abstraction and simplicity, which is crucial in competitions, where complex architectures are less likely to achieve the required level of robustness. Thus, we were able to transit between different strategies easily, adapting to changes in the competition rules. Moreover, using ROS to implement our architecture was helpful. Our approach was to build a set of composable entities (i.e., Components, Actions and Tasks) that allow us to create complex behaviors and integrate heterogeneous algorithms, but at the same time, leverage the reliability of a well-established framework 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 the last years, ROS has become a standard in robotics development. Although it definitely simplifies communication between processes and promotes system modularity, its centralized architecture (with all its network depending on a single ROS master) is a drawback for multi-UAV systems. Nonetheless, we addressed this issue with the ROS multi-master package, setting a specific master on each UAV. Furthermore, ROS 2 is already alleviating significantly communication problems, since it removes the master and is distributed by design. For rapid prototyping, we decided to use Python for the implementation of the higher-level modules, i.e., Tasks and Behavior Dispatchers. Even though the flexibility of Python is desirable for the development of high-level scripts, there were several code errors that were only discovered at mission execution time, with the consequent risk. Using a compiled language like C++, we would have detected many of these errors at compilation time. Therefore, it is essential to perform more thorough tests in simulation before validating Python implementations.

In terms of autopilot firmware, our first idea was to use PX4, as we strongly value its SITL tools to make simulations closer to reality. However, some of our Pixhawk Cube units had issues with some sensors (mainly the external magnetometer) when running PX4. Therefore, we opted for flying those units with ArduPilot, which solved this problem. Indeed, our UAL module proved to be quite useful in this heterogeneous configuration, as it dealt with firmware particularities and made transparent the use of any of them for the rest of the system. We also opted for relying on GNSS for UAV localization. We decided to use RTK corrections even if penalized, as we thought that an accurate tuning of the autopilot attitude controllers was critical so that our position/velocity controllers performed well. However, we ended up experiencing unreliable GNSS coverage in the MBZIRC arena, and we were not able to reach RTK *fixed* in some areas. Therefore, we conclude that we might have achieved a better overall performance using other methods for localization with the onboard sensors.

In general, our algorithms for brick and wall detection worked fine for the competition. We think that using our color-based detector for the wall would have eased the initial exploration phase, but we realized late that the wall was of a clear yellow color from above. Of course, our detectors were tailored to the competition objects, and we see room for improvement, endowing them with the capacity of detecting more generic objects, e.g., using learning-based approaches. Nonetheless, we believe that our architecture can simply accommodate these potential extensions by replacing the adequate components. In terms of overall strategy to solve the Challenge, we simplified initial complex behaviors during the competition to optimize our scoring chances, as most teams. Thus, we ended up switching to a simpler Shared Region Manager instead of reserving 3D space for UAV navigation; and we were prepared with alternative strategies under the event of a total lack of communication. Our software architecture endowed us with this flexibility.

Finally, we learned in our previous participation in MBZIRC (2017) not to rely only on magnetic grippers, and our new hybrid grasping mechanism demonstrated a significantly better performance for picking objects in the construction Challenge. The non-rigid structure where the magnets were mounted added the flexibility needed to accommodate the grasping appropriately. Also, the contact pincers made the whole grasping more reliable, correcting minor misalignments of the brick and holding it more firmly during transportation. Actually, we improved our rate of successful pick operations from 50% in 2017 [13] to 84%.

VIII. CONCLUSION

In this article, we presented a multi-UAV system for autonomous construction. For that, we proposed a multi-robot architecture to articulate cooperative missions involving physical interaction, focusing on outdoor partially unknown environments. We also contributed with the design and development of grasping devices to pick and place objects with UAVs. Our system is inspired by the MBZIRC Challenge 2 about wall building, and we provide all the details of our

specific implementation and experimental results for that Challenge.

Given the overall performance of other teams in the competition and the complexities involved, we regard our results as positive. Only a few teams managed to perform complete pick and place operations in autonomous mode successfully. In general, all the experience and knowledge derived from our participation in MBZIRC was quite helpful to improve our aerial platforms and software modules for multi-UAV missions. Even though this work focuses on MBZIRC, we believe that the software architecture and the hardware devices proposed represent a remarkable asset to be generalized to other UAV applications involving physical interaction. Actually, we plan as future work to improve our multi-UAV architecture with more specific components for UAV navigation and multi-UAV coordination. Thus, we would like to integrate our own algorithms for trajectory generation and following, as well as collision avoidance; as we relied on built-in autopilot functionalities so far.

ACKNOWLEDGMENT

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 Honorio Romero, Victor Vega, Rafael Salmoral, Alejandro Sanchez and Alejandro Braza. Special thanks to Joe Bracho and the Abu Dhabi RC Club for giving them the opportunity to fly at their airfield.

REFERENCES

[1] F. Augugliaro, S. Lupashin, M. Hamer, C. Male, M. Hehn, M. W. Mueller, J. S. Willmann, F. Gramazio, M. Kohler, and R. D'Andrea, "The flight assembled architecture installation: Cooperative construction with flying machines," *IEEE Control Syst.*, vol. 34, no. 4, pp. 46–64, Aug. 2014.

[2] M. Tognon, H. A. T. Chavez, E. Gasparin, Q. Sable, D. Bicego, A. Mallet, M. Lany, G. Santi, B. Revaz, J. Cortes, and A. Franchi, "A truly-redundant aerial manipulator system with application to Push-and-Slide inspection in industrial plants," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1846–1851, Apr. 2019.

[3] J. Grzybowski, K. Latos, and R. Czyba, "Low-cost autonomous UAV-based solutions to package delivery logistics," in *Advanced, Contemporary Control*. Cham, Switzerland: Springer, 2020, pp. 500–507.

[4] D. N. Das, R. Sewani, J. Wang, and M. K. Tiwari, "Synchronized truck and drone routing in package delivery logistics," *IEEE Trans. Intell. Transp. Syst.*, early access, May 22, 2020, doi: 10.1109/TITS.2020.2992549.

[5] M. Buehler, K. Iagnemma, and S. Singh, *The 2005 DARPA Grand Challenge: The Great Robot Race*, vol. 36. Berlin, Germany: Springer, 2007.

[6] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, vol. 56. Berlin, Germany: Springer, 2009.

[7] F. Amigoni, E. Bastianelli, J. Berghofer, A. Bonarini, G. Fontana, N. Hochgeschwender, L. Iocchi, G. Kraetzschmar, P. Lima, M. Matteucci, P. Miraldo, D. Nardi, and V. Schiaffonati, "Competitions for benchmarking: Task and functionality scoring complete performance assessment," *IEEE Robot. Autom. Mag.*, vol. 22, no. 3, pp. 53–61, Sep. 2015.

[8] D. Rodriguez, H. Farazi, G. Ficht, D. Pavlichenko, A. Brandenburger, M. Hosseini, O. Kosenko, M. Schreiber, M. Missura, and S. Behnke, "RoboCup 2019 AdultSize Winner NimbRo: Deep learning perception, in-walk kick, push recovery, and team play capabilities," in *RoboCup 2019: Robot World Cup XXIII*, S. Chalup, T. Niemueller, J. Suthakorn, and M.-A. Williams, Eds. Cham, Switzerland: Springer, 2019, pp. 631–645.

[9] G. Loianno, V. Spurny, J. Thomas, T. Baca, D. Thakur, D. Hert, R. Penicka, T. Krajnik, A. Zhou, A. Cho, M. Saska, and V. Kumar, "Localization, grasping, and transportation of magnetic objects by a team of MAVs in challenging desert-like environments," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1576–1583, Jul. 2018.

[10] V. Spurný, T. Báča, M. Saska, R. Pěnička, T. Krajník, J. Thomas, D. Thakur, G. Loianno, and V. Kumar, "Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles," *J. Field Robot.*, vol. 36, no. 1, pp. 125–148, Jan. 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21816

[11] R. Bähnemann, M. Pantic, M. Popović, D. Schindler, M. Tranzatto, M. Kamel, M. Grimm, J. Widauer, R. Siegwart, and J. Nieto, "The ETH-MAV team in the MBZ international robotics challenge," *J. Field Robot.*, vol. 36, no. 1, pp. 78–103, Jan. 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21824

[12] M. Beul, M. Nieuwenhuisen, J. Quenzel, R. A. Rosu, J. Horn, D. Pavlichenko, S. Houben, and S. Behnke, "Team NimbRo at MBZIRC 2017: Fast landing on a moving target and treasure hunting with a team of micro aerial vehicles," *J. Field Robot.*, vol. 36, no. 1, pp. 204–229, Jan. 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21817

[13] Á. R. Castaño, F. Real, P. Ramón-Soria, J. Capitán, V. Vega, B. C. Arrue, A. Torres-González, and A. Ollero, "Al-robotics team: A cooperative multi-unmanned aerial vehicle approach for the mohamed bin zayed international robotic challenge," *J. Field Robot.*, vol. 36, no. 1, pp. 104–124, Jan. 2019. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21810

[14] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging UAVs for disaster management," *IEEE Pervas. Comput.*, vol. 16, no. 1, pp. 24–32, Jan. 2017.

[15] E. T. Alotaibi, S. S. Alqefari, and A. Koubaa, "LSAR: Multi-UAV collaboration for search and rescue missions," *IEEE Access*, vol. 7, pp. 55817–55832, 2019.

[16] A. Koubaa, B. Qureshi, M.-F. Sriti, Y. Javed, and E. Tovar, "A service-oriented cloud-based management system for the Internet-of-drones," in *Proc. IEEE Int. Conf. Auto. Robot Syst. Competitions (ICARSC)*, Apr. 2017, pp. 329–335.

[17] U. A. Fiaz, M. Abdelkader, and J. S. Shamma, "An intelligent gripper design for autonomous aerial transport with passive magnetic grasping and dual-impulsive release," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2018, pp. 1027–1032.

[18] A. S. Nedungadi and M. Saska, "Design of an active-reliable grasping mechanism for autonomous unmanned aerial vehicles," in *Modelling and Simulation for Autonomous Systems*, J. Mazal, A. Fagiolini, and P. Vasik, Eds. Cham, Switzerland: Springer, 2020, pp. 162–179.

[19] C. C. Kessens, J. Thomas, J. P. Desai, and V. Kumar, "Versatile aerial grasping using self-sealing suction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 3249–3254.

[20] A. McLaren, Z. Fitzgerald, G. Gao, and M. Liarokapis, "A passive closing, tendon driven, adaptive robot hand for ultra-fast, aerial grasping and perching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 5602–5607.

[21] A. Suarez, A. E. Jimenez-Cano, V. M. Vega, G. Heredia, A. Rodriguez-Castaño, and A. Ollero, "Design of a lightweight dual arm system for aerial manipulation," *Mechatronics*, vol. 50, pp. 30–44, Apr. 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0957415818300011

[22] J. Thomas, G. Loianno, K. Sreenath, and V. Kumar, "Toward image based visual servoing for aerial grasping and perching," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 2113–2118.

[23] P. R. Soria, B. Arrue, and A. Ollero, "Detection, location and grasping objects using a stereo sensor on UAV in outdoor environments," *Sensors*, vol. 17, no. 12, p. 103, Jan. 2017. [Online]. Available: https://www.mdpi.com/1424-8220/17/1/103

[24] A. Suarez, P. R. Soria, G. Heredia, B. C. Arrue, and A. Ollero, "Anthropomorphic, compliant and lightweight dual arm system for aerial manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 992–997.

[25] M. Vrba and M. Saska, "Marker-less micro aerial vehicle detection and localization using convolutional neural networks," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2459–2466, Apr. 2020.

[26] M. Schwarz, H. Schulz, and S. Behnke, "RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 1329–1335.

[27] G. A. Korsah, A. Stentz, and M. B. Dias, "A comprehensive taxonomy for multi-robot task allocation," *Int. J. Robot. Res.*, vol. 32, no. 12, pp. 1495–1512, Oct. 2013.

[28] E. Nunes, M. Manner, H. Mitiche, and M. Gini, "A taxonomy for task allocation problems with temporal and ordering constraints," *Robot. Auto. Syst.*, vol. 90, pp. 55–70, Apr. 2017.
[29] L. E. Caraballo, J. M. Díaz-Báñez, I. Maza, and A. Ollero, "The block-information-sharing strategy for task allocation: A case study for structure assembly with aerial robots," *Eur. J. Oper. Res.*, vol. 260, no. 2, pp. 725–738, Jul. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0377221717300048
[30] E. Umili, M. Tognon, D. Sanalitro, G. Oriolo, and A. Franchi, "Communication-based and communication-less approaches for robust cooperative planning in construction with a team of UAVs," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Sep. 2020, pp. 279–288.
[31] M. Krizmancic, B. Arbanas, T. Petrovic, F. Petric, and S. Bogdan, "Cooperative aerial-ground multi-robot system for automated construction tasks," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 798–805, Apr. 2020.
[32] M. Schmittle, A. Lukina, L. Vacek, J. Das, C. P. Buskirk, S. Rees, J. Sztipanovits, R. Grosu, and V. Kumar, "OpenUAV: A UAV testbed for the CPS and robotics community," in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2018, pp. 130–139.
[33] K. Xiao, S. Tan, G. Wang, X. An, X. Wang, and X. Wang, "XTDrone: A customizable multi-rotor UAVs simulation platform," 2020, *arXiv:2003.09700*. [Online]. Available: http://arxiv.org/abs/2003.09700
[34] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 6235–6240.
[35] J. L. Sanchez-Lopez, M. Molina, H. Bavle, C. Sampedro, R. A. Suárez Fernández, and P. Campoy, "A multi-layered component-based approach for the development of aerial robotic systems: The aerostack framework," *J. Intell. Robotic Syst.*, vol. 88, nos. 2–4, pp. 683–709, Dec. 2017.
[36] T. Baca, M. Petrlik, M. Vrba, V. Spurny, R. Penicka, D. Hert, and M. Saska, "The MRS UAV system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles," 2020, *arXiv:2008.08050*. [Online]. Available: http://arxiv.org/abs/2008.08050
[37] F. Real, A. Torres-González, P. R. Soria, J. Capitán, and A. Ollero, "Unmanned aerial vehicle abstraction layer: An abstraction layer to operate unmanned aerial vehicles," *Int. J. Adv. Robotic Syst.*, vol. 17, no. 4, pp. 1–13, 2020, doi: 10.1177/1729881420925011.

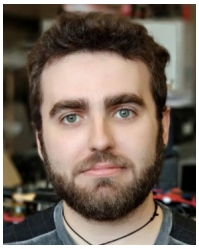
FRAN REAL received the M.Sc. degree in automation and robotics from the University of Seville, in 2009. He is currently a Researcher with the GRVC Robotics Laboratory, University of Seville, Spain. He has participated as a member of the Iberian Robotics Team in the Mohamed Bin Zayed International Robotic Challenge 3, in 2017 and 2020. He has participated in the more than 15 projects (eight international) and has developed and integrated software and hardware for several aerial and terrestrial robots, many of them with manipulation capabilities. His research interests include aerial robotics, reactive behaviors for autonomous navigation, and architectures for cooperative robotics. He and his team received the First prize from the Mohamed Bin Zayed International Robotic Challenge 3.

ÁNGEL R. CASTAÑO received the M.Eng. degree in telecommunications engineering and the Ph.D. degree in systems engineering and automation. He was a Visiting Researcher with the Center for Self-Organizing and Intelligent Systems, Utah State University, and the E. Piaggio Center, University of Pisa. He is currently an Associate Professor with the University of Seville. He has participated in more than 30 research projects (including 11 projects funded by the European Commission) related to unmanned vehicles and cooperating multivehicle systems. He was the Team Leader of the Iberian Robotics Team in the MBZIRC 2020 competition. He is the PI from the University of Seville in the EU projects GAUSS and DURABLE. He has also led 11 technology transfer projects with several companies, such as Airbus Military, Navantia, Iturri Group, or EMASESA. His research interests include multirobot systems and intelligent transportation systems. He and his team received the First prize from the MBZIRC 2020 Third Challenge.

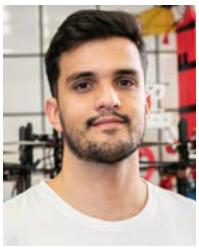
ARTURO TORRES-GONZÁLEZ received the Ph.D. degree from the University of Seville, in 2017. His Ph.D. thesis received the Best Iberian Thesis in robotics by SEIDROB (Spanish Robotics Society). During his Ph.D. study, he was a Visiting Researcher with the ACFR, The University of Sydney, Australia. He is currently a Postdoctoral Researcher with the GRVC Robotics Laboratory, University of Seville, Spain. He has participated in more than 12 projects (eight international) and also in the Iberian Robotics Team at the MBZIRC competition, in 2017 and 2020, in Abu Dhabi. He is author of more than 25 publications focusing on multi-robot cooperation, robot-sensor network cooperation and simultaneous localization and mapping. He and his team received the First prize from the MBZIRC 2020 Third Challenge.

JESÚS CAPITÁN received the Ph.D. degree in telecommunication engineering in 2011. He was a Postdoctoral Researcher with the Instituto Superior Tecnico, Lisboa, Portugal, and the University of Duisburg-Essen, Essen, Germany. During his Ph.D. study, he was a Visiting Researcher with the Robotics Institute, Carnegie Mellon University, Pittsburgh, USA, and the Instituto Superior Tecnico, Lisbon, Portugal. He is currently an Associate Professor with the University of Seville. He has participated in more than 20 projects (ten international). He was one of the PI from the University of Seville in the EU project MULTIDRONE and a participant of the USE team in the MBZIRC competition, in 2017 and 2020. He has authored more than 50 publications focusing on multi-robot cooperation, decision making, and multi-UAV conflict resolution. He was a Jury Member of the euRathlon Competition on Aerial Robotics, in 2015. He and his team received the First prize from the MBZIRC 2020 Third Challenge. He is an Associate Editor of the *International Journal of Advanced Robotics Systems*.

PEDRO J. SÁNCHEZ-CUEVAS (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in aerospace engineering from the GRVC Robotics Laboratory, University of Seville, Seville, Spain, in 2014, 2018, and 2020, respectively. He was a part of the Iberian Robotics Team presented at the MBZIRC 2020. His research interests include aerial robotics, mechatronics, and inspection robotics. He and his team received the First prize from the Third Challenge of the MBZIRC 2020.



MANUEL J. FERNÁNDEZ (Student Member, IEEE) received the B.Sc. degree in telecommunications engineering from the University of Seville, Spain, in 2020, where he is currently pursuing the master's degree in logic, computing and artificial intelligence. He is currently a Researcher with the GRVC Robotics Laboratory, University of Seville. He has participated in more than four research projects (three international) and has been a part of the Iberian Robotics Team. He has authored more than four publications about security and aerial manipulation. His research interests include aerial robotics, communication security, swarming, blockchain technologies, and robot manipulation. He and his team received the First prize from the Third Challenge of MBZIRC 2020. He was the Chair of the IEEE Student Branch from the University of Seville, from 2019 to 2020.



MANUEL VILLAR received the B.Sc. degree in engineering of industrial technologies, specialized in automation, from the University of Seville, Spain, where he is currently pursuing the M.Sc. degree. He is currently a Researcher with the GRVC Robotics Laboratory, since 2019. In his first project, he has been a part of the Iberian Robotics team in the MBZIRC 2020 and a Designer of the Challenge 2 pick-up mechanism. He and his team received the First prize from the Third Challenge of MBZIRC 2020.

ANÍBAL OLLERO (Fellow, IEEE) is currently a Full Professor and the Head of the GRVC Robotics Laboratory, University of Seville, and a Scientific Advisor of the Center for Advanced Aerospace Technologies, Seville, Spain. He is also a Full Professor with the University of Santiago, Spain, and the University of Malaga, Spain, a Researcher with the Robotics Institute, Carnegie Mellon University, Pittsburgh, USA, and LAAS-CNRS, Toulouse, France. He has been a supervisor of 45 Ph.D. theses and led more than 160 research projects, participating in more than 25 projects of the European Research Programs, a coordinator of seven and an associated coordinator of three, all of them dealing with unmanned aerial systems and aerial robots. Since 2018, he has been running the GRIFFIN ERC-Advanced Grant developing a new generation of aerial robots to glide, flap the wings, perch, and manipulate by maintaining the equilibrium. Since 2019, he has been coordinating the H2020-AERiAL-CORE project about aerial robotic manipulators for inspection and maintenance. He has transferred technologies to 20 companies. He authored more than 750 publications, including nine books, 200 journal articles, and 15 edited books. He was a Founder and the President of the Spanish Society for the Research and Development in Robotics, until 2017. He has also been elected between the three European innovators of the year being candidate to the European personalities in 2017 and the IEEE fellow for contributions to the development and deployment of aerial robots. He was a member of the Board of Directors of euRobotics, until 2019. He has received 23 international research and innovation awards, including the Overall Information and Communication Technologies Innovation Radar Prize 2017 of the European Commission and the recent Rei Jaume I in New Technologies, Spain. He is currently the Co-Chair of the IEEE Technical Committee on Aerial Robotics and Unmanned Aerial Vehicles, a Coordinator of the Aerial Robotics Topic Group de euRobotics. He has delivered plenaries and keynotes in more than 100 events, including the IEEE ICRA 2016 and IEEE IROS 2018.