

A Multi-MAV System for the Autonomous Elimination of Multiple Targets in the MBZIRC 2020 Competition

Yurii Stasinchuk¹, Matouš Vrba¹, Tomáš Báča¹, Vojtěch Spurný¹, Matěj Petrlík¹, Daniel Heřt¹, David Žaitlík and Martin Saska¹

Faculty of Electrical Engineering, Czech Technical University in Prague

Abstract: The Mohamed Bin Zayed International Robotics Challenge (MBZIRC) is a prestigious, biennial competition aimed at furthering the state of the art in the field of autonomous robotics. In this paper, we present our solution to one of the tasks in the MBZIRC 2020 competition, which design won second place in Challenge 1 and first place in the Grand Challenge of the competition. This paper focuses specifically on the popping task of multiple balloons by multiple Micro Aerial Vehicles (MAVs). This task required a rapid and robust performance to compete with systems from other expert robotic teams from around the world. In this task, a team of autonomous MAV's had to seek and attack several balloons positioned throughout the competition arena. The novel fast autonomous searching for multiple targets in 3D, their reliable detection, precise relative state estimation, and agile motion planning algorithms are presented in this paper, together with an application for general tasks of 3D target capturing. With a primary focus on reliability, the methods reported in this paper and the entire, complex multi-agent system were successfully verified in both extreme conditions of the desert and the MBZIRC competition. An evaluation of the proposed methods using data from the competition and additional separate datasets is presented. The relevant code of our implementation has been made publicly available for the robotics community.

Keywords: aerial robotics, perception

1. Introduction

Intruder MAV attacks have had a devastating effect on oil refineries in the Middle East. One such attack occurred in 2019¹, causing oil losses and significant economic issues throughout the region. Preventing such attacks was a primary motivation for Challenge 1 of the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2020 robotics competition. For safety and technical reasons, intruder MAVs cannot be countered with conventional ballistic defenses and, therefore,

¹ <https://www.cnn.com/2019/09/15/saudi-stock-market-dives-crude-to-jump-after-attack-on-oil-plants.html>

Received: 30 September 2020; revised: 26 February 2021; accepted: 7 May 2021; published: 4 August 2022.

Correspondence: Yurii Stasinchuk, Faculty of Electrical Engineering, Czech Technical University in Prague, Email: stasiyur@fel.cvut.cz

This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © 2022 Stasinchuk, Vrba, Báča, Spurný, Petrlík, Heřt, Žaitlík and Saska

DOI: <https://doi.org/10.55417/fr.2022052>

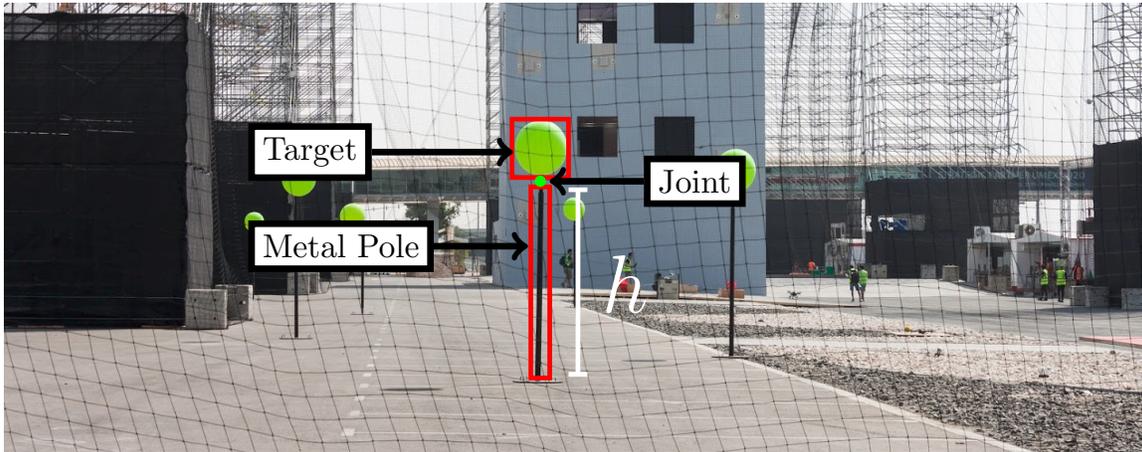


Figure 1. The Challenge 1 arena is shown with annotations of the target specifications. A balloon *Target* is connected to the rigid link *Metal pole* (with height h) via a *Joint*.

require novel solutions. Challenge 1 was intended to further research in the field of defense from attacks by small MAVs, focusing mainly on the design of systems for the cooperating MAV systems to counter multiple intruder MAVs. Such multi-agent defense systems are crucial for large infrastructure sites, such as airports, oil refineries, and power plants. In a recent example, an unknown swarm of MAVs circled over a nuclear power plant in the United States on two separate occasions². These situations with multiple MAVs present an even greater complication than the hunting of a single MAV, as they require management of target selection, visual perception, and communication systems. The utilization of these technologies was necessary in solving the MBZIRC 2020 Challenge 1, as well as required for the real MAV hunting scenario.

To emulate the multi-intruder MAV and multi-defender MAV scenario, Challenge 1 posed two tasks to be completed simultaneously by three MAVs. The first task was to catch a ball attached to the intruder MAV, which has been tackled in a separate publication [Vrba et al., 2020]. The second task was to locate and destroy static targets, represented by balloons, while avoiding obstacles in the area (see Figure 1). Each target was tethered to the ground via a metal pole. The Challenge objective, and main focus of our work, was to develop multi-agent strategy for efficient target localization and interaction, which is the main focus of the presented work. Although the main inspiration of the MBZIRC competition was drone safety³, the resulting system may also be utilized in other contexts, e.g. a scenario similar to the “Treasure hunt” task from MBZIRC 2017 [Spurny et al., 2019].

In commercial MAV hunting systems, two techniques are most often used: elimination by a shooting tool (i.e. shooting of a net or a missile) or elimination from direct impact. Elimination by shooting requires an additional onboard tool that could potentially cause dangerous system failures and would also require more dynamic and precise motion planning methods. However, it has the advantage of multiple shooting attempts, although the number of projectiles is usually limited (one for EAGLE.ONE⁴ and two for FORTEM⁵). In the case of MBZIRC 2020, shooting devices were not allowed in order to motivate research of direct physical elimination for cases where targets are too small or agile to be shot.

Although Challenge 1 of MBZIRC 2020 is only a physical simulation of real-world mission, it pushes the frontiers of such research. Hunting intruder MAVs with a multi-MAV defender team

² <https://www.forbes.com/sites/davidhambling/2020/07/30/MAV-swarm-invaded-palo-verde-nuclear-power-plant/>

³ <http://mbzirc.com/challenge/2020>

⁴ <http://eagle.one>

⁵ <https://fortemtech.com/>

proved to be a complicated scenario, drawing the participation of 20 top research teams, each offering a different solution to the problem.

1.1. Related work

The hunting of MAVs is a familiar task to the robotics community. The task is typically split into several subtasks that must be managed for the autonomous interception of an intruder MAV. Firstly, the intruder Micro Aerial Vehicle (MAV) must be automatically detected in a designated perimeter. Following detection of the target, visual perception methods are required for tracking. Lastly, an onboard mission planning system autonomously manages the following of the target and target interception. Moreover, the interception task can be solved by a multi-MAV system. This system is able to increase the overall system reliability thanks to redundancy and by decreasing the resulting elimination time if the information is shared between the agents. Since solving the complete task requires combining results from many robotics subfields, the entirety of the problem is rarely tackled in a single work. The MBZIRC 2020 competition aimed to push the frontiers of research by creating feasible testing conditions for this complex task.

The detection and visual tracking of intruder MAVs is a regularly evaluated problem for the robotics community. Numerous recent works have tackled autonomous visual detection of flying MAVs [Li et al., 2016, Rozantsev et al., 2015, Sapkota et al., 2016, Opromolla et al., 2018, Saqib et al., 2017, Schumann et al., 2017, Vrba and Saska, 2020] and have focused solely on the detection and visual tracking subproblems. A common approach for the marker-less detection of MAVs are Convolutional Neural Networks (CNNs). Due to the well-defined appearance of the target in question (dimension, color, shape), a much simpler color segmentation method has been proposed⁶ in this work. We have used a color-segmentation method, which is a common technique for object detection (similar to [Fitriana et al., 2016], which happens to also be for a competition, or to [Gai et al., 2020], which utilizes a combination of RGB segmentation with depth data as well). Together with the object detection algorithm, we provide a semi-automatic tool⁷ for color calibration which lends flexibility to the system. Since we don't rely on specific color, the detector can be recalibrated in a matter of minutes. Although a CNN could perform such detections, neural networks are impractical in this case as the high computational requirements necessary are difficult to meet onboard the small MAVs. Multiple computationally expensive and time-critical subsystems (planning, mapping, control, state estimation) must share a single dedicated onboard computer, which would restrict the use of CNNs in such situations. Moreover, a custom-built detection method can easily combine input data from multiple sensors (RGB camera and depth camera).

The use of external MAV detection systems is studied in [Drozdowicz et al., 2016, Liu et al., 2017, Nguyen et al., 2016]. In a practical realization of an aerial interception system, a ground-based detection station could be used to provide vital information to the onboard aerial system and improve detection. However, the competition did not allow such an installation. Therefore, the proposed system relies solely on onboard sensors. We expect that the real-world deployment of a MAV hunting system would benefit from using the proposed approach by increasing its overall reliability, adapting to changes in the environment, and providing redundancy.

A multi-robotic swarm approach is described in [Brust et al., 2017]. Additionally, multi-robot planning for interception and surveillance is studied in [Faigl et al., 2019]. Furthermore, there is existing research on planning and gathering information about targets, such as found in [Schlotfeldt et al., 2018, Capitan et al., 2016]. These works rely on apriori knowledge of the environment and focus mostly on surveillance and planning for MAVs.

Interception techniques have been studied in recent years [Yang and Quan, 2020]. Our previous work on detection, tracking, following, and interception of a target using a wide-baseline stereo

⁶ https://github.com/ctu-mrs/object_detect

⁷ https://github.com/ctu-mrs/color_picker

camera was proposed in [Vrba et al., 2019]. The system detailed in this paper first utilized a powerful sensor for detection and precise localization of an intruder MAV, then proceeded to automatically follow it. However, the proposed platform was relatively large in order to accommodate the sensors and computational power required, unlike the MAVs that was to be allowed in the MBZIRC 2020 competition.

Although the Challenge 1 task of the MBZIRC 2020 was unique, it was similar to the “Treasure Hunt” task introduced at the previous competition held in 2017 [Spurny et al., 2019, Beul et al., 2019]. The 2017 “Treasure Hunt” Challenge was a two-dimensional problem requiring the collection of colorful metal discs from the ground, whereas the 2020 task was elevated to a full, three-dimensional environment. The visual detection of target balloons viewed from an arbitrary angle is a complex task when compared to the visual detection of colored discs on the ground. Moreover, due to the presence of mounting poles for the balloons (see Figure 1), MAV collisions with obstacles in the operational area were anticipated in the 2020 challenge. Thus, a reliable system for task reallocation was necessary for minimizing the execution time when using multiple MAVs and to allow the full completion of the task in the event that one MAV team-member fails.

To our best knowledge, all existing state-of-the-art works focus on the subproblems of intruder localization and elimination. The system proposed in this paper goes beyond the current state of the art by designing all the necessary components for a complete system execution onboard the Micro Aerial Vehicle. We have proposed a complete methodology for executing multi-robotic scenarios reliably, from the initial takeoff phase to autonomous searching, elimination of targets, and the final landing. Moreover, the proposed solution had been extensively tested in various realistic outdoor environments and the time-constrained conditions of the MBZIRC 2020 competition.

1.2. Contributions

Based on the current MAV hunting state-of-the-art, the MBZIRC technical advisory committee⁸, composed of well-recognized professionals from the robotics community and industry, designed the MBZIRC 2020 Challenge 1 to advance technology towards the development of more autonomous, robust, and agile MAV hunters. The realization of a complete mission – from autonomous take-off, reliable interception of a fast-flying MAV, to a safe landing in a designated position – is rarely seen in existing solutions. Moreover, the balloon popping subtask introduced the important requirement for a system of multiple autonomous MAVs hunting multiple targets to need to solve situations where several, unauthorized UAVs penetrate a no-fly zone. This scenario has never been discussed in literature, and no practical demonstration of such a system is yet available. There are three main contributions addressed in this paper:

1. A methodology has been designed, proposed, and experimentally verified for solving the problem of multiple MAV agents hunting multiple targets in 3D space. The solution satisfied the requirements of the MBZIRC challenge, but has also been designed to provide a general approach for the control and coordination of multiple MAV hunters in real MAV hunting scenarios. The transition from the competition environment to real application is described in Section 6.
2. The paper has provided a scientific contribution in the field of precise control and state estimation by combining diverse sensors for increased robustness of the overall system. In the context of this application, the presented complete system of detection, filtration, control, planning, and multi-robot coordination is a novel design. This design proved to be robust to false positives and both software and hardware failures (including a total failure of some of the team-members), which are often overlooked points, making it suitable for real-world deployment in safety-critical applications.

⁸ <https://mbzirc.com/committee>

3. Lastly, it is a common approach of the MBZIRC community to share with other teams a detailed description of systems that have performed well, in order to facilitate progress to enable further progression beyond the current robotic state of the art in the next MBZIRC round. The system described in this paper, with all theoretical and technical details, produced one of the competition’s best performances. According to our knowledge, this was the only solution that managed to eliminate all targets in all official rounds of the competition (final rehearsal, Trials 1 and 2, and the Grand Challenge) without incurring penalty for using RTK-GNSS or manual control. Based on the careful study of available video records for all teams, we believe that the system also provided the fastest performance, which capability was essential for the MAV hunting task (104 s for the elimination of all six targets with two MAVs in Trial 2 and 129 s with one MAV in the Grand Challenge). As it is demonstrated in the extensive experimental evaluation before and after the competition, as well as through statistical analysis (Section 5), our system can to perform consistently under varied conditions. The presented solution and whole MAV system are also fully available as open-source⁹ for the community to facilitate furthering the progress for competitors joining in the next round next MBZIRC competition.

2. Problem Specification

Challenge 1 of the MBZIRC 2020 Competition consisted of performing two sub-tasks simultaneously in the same arena. The first task required catching a soft yellow ball (11 cm in diameter) that was attached via a magnet and a 1.45 m long wire to an intruder MAV. The intruder MAV flew on an eight-shaped trajectory with a speed of 8 m/s and height between 8–15 m. The ball-catching task is tackled in a separate publication [Vrba et al., 2020]. In this paper, we focus mainly on the second part of the problem.

The second task of Challenge 1 was the balloon hunting task. In total, five balloon targets were in the arena with their positions being changed during each trial. The balloons had a diameter of 50 cm when inflated and were green in color. Each balloon was attached to the ground via a rigid link - a metal pole with a fixed height of h , as shown in Figure 1. The maximum height of the poles was 5 meters. A joint connected the balloon to the pole, allowing the wind to affect movement of the balloon. The targets could not be eliminated via a shooting tool, only by direct contact.

The MBZIRC 2020 arena was shaped as a non-convex polygon with a length of 100 meters, width of 30 meters, and a 20 m ceiling. The arena borders were covered in net to prevent MAVs from escaping and potentially harming bystanders. The detailed description of the experimental setup is described in Subsection 5.3. A team of maximum three MAVs could be deployed in the arena at once. Each participant was required to start all MAVs from one particular zone assigned by the organizers. We decided to solve the balloon hunting task using a coordinated team of two elimination MAVs, with the third MAV searching for dynamic targets. Similarly for the Grand Challenge, a maximum number of three MAVs were permitted to operate at once. Therefore, to complete the balloon hunting task, only one MAV was used. The remaining two MAVs were completing the other tasks simultaneously.

3. Preliminaries

The MAVs deployed in the competition relied upon an underlying control and estimation system developed by the Multi-Robot Systems (MRS¹⁰) team at the Czech Technical University in Prague. The *MRS MAV System* [Baca et al., 2020] is an open-source pipeline providing advanced capabilities for real-world deployment of MAVs, such as state estimation, trajectory tracking, and control

⁹ https://github.com/ctu-mrs/mbzirc_2020_balloon_popping

¹⁰ <http://mrs.felk.cvut.cz>

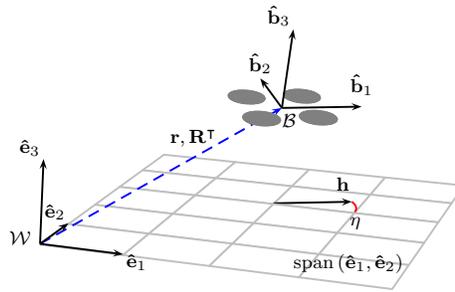


Figure 2. An illustration of the world frame \mathcal{W} in which the 3D position and orientation of the MAV body is expressed. The body frame \mathcal{B} relates to \mathcal{W} by the translation $\mathbf{r} = [x, y, z]^T$ and by rotation \mathbf{R}^T . The MAV heading vector \mathbf{h} , which is a projection of $\hat{\mathbf{b}}_1$ to the xy -plane, forms the heading angle $\eta = \text{atan2}(\mathbf{h}_{(2)}, \mathbf{h}_{(1)})$.

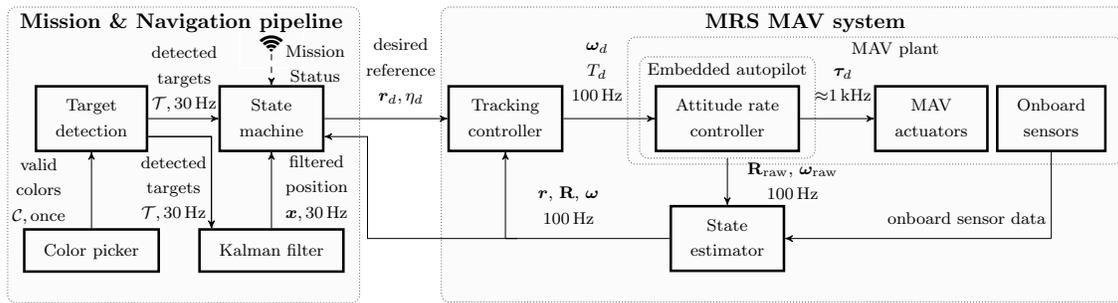


Figure 3. The onboard autonomous system consists of two distinct parts: the *MRS MAV system* [Baca et al., 2020] and the *Mission & Navigation pipeline* that is presented in this paper. The *MRS MAV System* provides MAV state estimation, feedback control, and feedforward tracking. The *Mission & Navigation pipeline* is responsible for balloon detection, balloon filtering, trajectory planning, and mission guidance for the competition. All modules shown in the figure are running on the MAV onboard computer. The MAVs share only information about their current position and mission status, as described in Subsection 4.4.

algorithms. The state estimate consists of the MAV position, orientation, and velocity (translational and angular), defining an MAV body frame \mathcal{B} within a world frame \mathcal{W} (the involved frames of reference are depicted in Figure 2). A Model Predictive Control (MPC) tracker receives desired trajectories from the user software and creates a feasible and smooth control reference with respect to the MAV dynamics. This control reference is tracked by a geometric tracking controller on $\text{SE}(3)$ [Lee et al., 2010]. The system is implemented in Robot Operating System (ROS) [Quigley et al., 2009] and is available as open source together with an extensible simulation environment¹¹.

The connection between the *MRS MAV system* and the proposed *Mission & Navigation pipeline* are shown in Figure 3. The *Mission & Navigation pipeline* commands the *MRS MAV system* by providing desired trajectory references $\{(\mathbf{r}_d, \eta_d)_1, (\mathbf{r}_d, \eta_d)_2, \dots, (\mathbf{r}_d, \eta_d)_k\}$ for the *Tracking controller*. A trajectory reference consists of a sequence of desired 3D positions \mathbf{r}_d and headings η_d of the MAV, sampled at regular intervals. The *Tracking controller* provides feedback control of the translational and rotational dynamics of the MAV. The controller outputs the desired attitude rate $\boldsymbol{\omega}_d$ and the desired collective thrust T_d for the embedded Pixhawk flight controller. The embedded flight controller generates desired thrusts for each motor τ_d and also outputs the estimated MAV orientation \mathbf{R}_{raw} and the estimated attitude rate $\boldsymbol{\omega}_{\text{raw}}$, which are fused with other onboard sensory data (such as Global Navigation Satellite System (GNSS) position) by the *State estimator*. The navigation pipeline also specifies a set of dynamics constraints that serve to limit the speed,

¹¹ https://github.com/ctu-mrs/mrs_uav_system

acceleration, jerk, and snap of the produced flight manoeuvres. The *MRS MAV system* ensures robust tracking of the desired trajectory (modified by the MPC tracker to respect the current dynamic constraints) and provides estimation of the MAV state.

4. Autonomous Multi-MAV System

A general structure of the proposed *Mission & Navigation pipeline* is shown in [Figure 3](#) and consists of the following blocks:

- *Target detection*: is a visual perception module that detects the balloons and estimates the positions of these detected targets (see [Subsection 4.1](#)),
- *Color picker*: is a semi-automatic color selection tool that provides the *Target detection* subsystem with a set of valid target colors that are to be detected (see [Subsection 4.1](#)),
- *Kalman filter*: this module provides a filtered position estimate of the selected target during the elimination phase of the state machine to reduce false positives and filter out detection noise (see [Subsection 4.2](#)),
- *State machine*: controls the mission execution, selects the current target to be eliminated, and plans the search, approach, and elimination trajectories (see [Subsection 4.3](#)).

4.1. Target detection and position estimation

A crucial element of an autonomous robotic system designed for elimination of intruder MAVs is the target position and state estimation method. This method must be precise, reliable, and fast. For general MAV hunting tasks, neural networks may be effectively used if relying on computer vision, as we have demonstrated in a single MAV hunting scenario [[Vrba and Saska, 2020](#)]. For the the efficient multi-target elimination in the MBZIRC 2020 scenario, the presented color segmentation technique achieved faster and more precise results while maintaining sufficient reliability. Nevertheless, any detection technique may be used with the proposed system if required.

The balloons are detected from an RGB camera image using a binary color segmentation algorithm. The 3D position of the detected balloons relative to the camera is estimated from their apparent size in the image and optionally utilizes the aligned depth image for better distance estimation and false-positive rejection (see [Figure 4](#) for an example of the detection). The steps of the detection and position estimation algorithm are:

1. The RGB image is segmented pixel-wise using a Look-Up Table (LUT), which maps the RGB color space to a binary value that indicates validity of each color. The set of valid colors \mathcal{C} may be selected manually, but we have used a novel color selection tool to be described further. The result of the segmentation is a binary image where each pixel's value represents whether its original color is a valid color of the target.

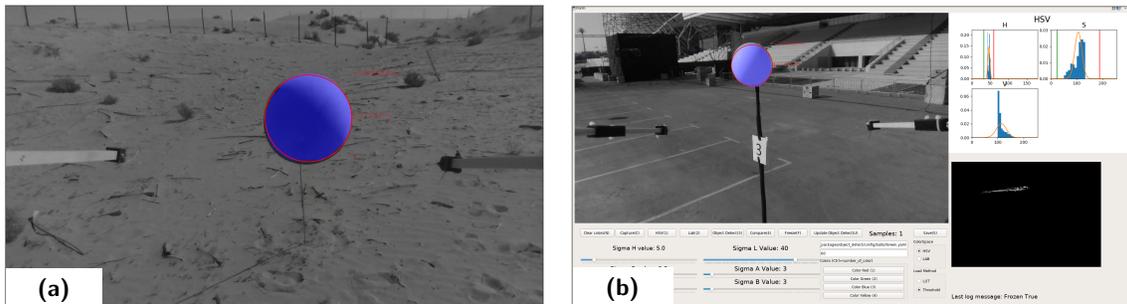


Figure 4. In figure (a), the result of the target detection algorithm is shown. The pixels with valid colors (result of the color segmentation) are highlighted blue and the detected target is marked with a red circle. In figure (b), GUI of the *Color picker* tool is shown.

2. The algorithm presented in [Suzuki and Abe, 1985] is used to extract contours from this binary image.
3. The extracted contours are filtered based on several shape descriptors, specifically for area, circularity, convexity, and inertia. The contours conforming to the empirically determined bounds of the shape descriptors are considered detection candidates.
4. The distance of each detection candidate from the camera is estimated using its known physical dimensions relative to its apparent size in the image and by using a calibrated mathematical model of the camera.
5. If available, the median distance for a contoured target is calculated from the depth image. This distance estimate may not be available when, for example, the target is beyond the limited range of the depth camera. This distance is compared to the distance estimate from the previous step. If the difference is too great, this indicates that the apparent size of the detection candidate in the image does not correspond with the known dimensions and measured depth of the target. Therefore, the detection candidate is discarded as a false-positive. Otherwise, the distance estimate from the previous step is substituted by the median distance supplied from the depth image. This is due to the fact that the depth measurement (if available) is more precise than the estimated distance based on the object’s apparent size.
6. A directional vector from the camera origin to the center of the contour is calculated using a calibrated camera model for each detection candidate.
7. Relative 3D positions of the detected balloons are calculated using the corresponding directional vectors and estimated distances from the camera. This set of detected target positions \mathcal{T} is the final output of the detection and position estimation algorithm that is repeated for each input RGB image.

The LUT is generated based on the selected set of valid colors \mathcal{C} and is indexed using red, green, and blue channels. For the 24-bit color representation that was output from the RealSense D435 stereo camera used in this paper, this indexing pattern translates to a LUT size of $2^8 \times 2^8 \times 2^8 = 256 \times 256 \times 256 = 16777216 \approx 16.8 \cdot 10^6$ elements. Each LUT element represents validity (value of 1) or invalidity (value of 0) of the corresponding indexed color and is saved as one byte. This implementation provided very good segmentation speed (approx. 28 Hz for an image size 1920 px \times 1080 px on our onboard i7-8559U CPU and 100 Hz with hardware acceleration using the CPU’s integrated Iris Plus 655 GPU) at a cost of <17 MiB of RAM space, which can be considered a negligible memory requirement for modern systems.

This high segmentation speed using the LUT method enabled our system to process each image from the input image stream with very low latency and without consuming too much of the limited onboard processing power. The low latency and high refresh rate of the balloon position estimates are crucial for accurate feedback when popping the balloons (which must be as fast as possible in order to be competitive). This is in contrast to the methods utilizing Convolutional Neural Networks (CNNs) which were used by most other teams in the competition. We have tested several CNN-based methods, but found them to be too slow, too computationally demanding (approx. 4 Hz for the tiny-YOLO [Redmon and Farhadi, 2017] and for the CenterNet [Duan et al., 2019] CNN architecture with HW acceleration), and unnecessarily complex for this task.

In a typical scenario, the set of valid colors for the color segmentation is selected manually by the user. Based on our previous experience (MBZIRC 2017 [Spurny et al., 2019]), this is a tedious process that takes up valuable time and must be done whenever the lighting conditions are changed. Therefore, we have created the *Color picker* tool to serve as a semi-automatic tool that selects an area in an image and generates a LUT of the colors represented in the area. The valid colors may be selected interactively using several methods in the HSV [Joblove and Greenberg, 1978] or L*a*b* [International Color Consortium and others, 2004]. The resulting LUT may be directly used by the *Target detection* package. An illustration of the color selection process and the resulting segmentation is shown in Figure 4.

The *Color picker* reduced the color selection duration to less than 40 seconds during the competition. The selection was able to be done from a significant distance, as was crucial for fast re-calibration of the colors before each trial and for enabling the reliable detection of the targets. The set of valid colors can be selected from any camera connected with ROS. Offline selection from data logged using the rosbag tool¹² is also possible. We have provided the *Color picker* for open-source use at https://github.com/ctu-mrs/color_picker to enable faster development and deployment of segmentation-based detection methods for the robotics community.

4.2. Balloon position filtering

To stabilize the output of the balloon detector and to discard false-positive detections, a position filtering algorithm based on the Kalman filter is applied. For the competition, it is assumed that the balloons are stationary. However, this assumption was not true in practice due to the effect of wind on the movement of the balloons as they oscillated around their attachment point. These balloon perturbations were filtered out to better stabilize the popping manoeuvres. The presented method was also tested with dynamically floating balloons tethered to the ground via a rope. Even under such conditions of movement, this method proved to be robust enough to enable reliable elimination (as is demonstrated in Section 5).

A discrete linear stochastic system model is used in the form

$$\mathbf{x}_{[k]} = \mathbf{A}\mathbf{x}_{[k-1]} + \mathbf{v}_{[k]}, \quad \mathbf{v}_{[k]} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}), \quad (1)$$

$$\mathbf{z}_{[k]} = \mathbf{H}\mathbf{x}_{[k]} + \mathbf{w}_{[k]}, \quad \mathbf{w}_{[k]} \sim \mathcal{N}(\mathbf{0}, \mathbf{S}_{[k]}), \quad (2)$$

where $\mathbf{x}_{[k]}$ is a state vector estimate at time-step k , \mathbf{A} is a state transition matrix, \mathbf{v} is process noise, \mathbf{Q} is a covariance matrix of the process noise, $\mathbf{z}_{[k]}$ is a measurement vector, \mathbf{H} is a state to measurement-mapping matrix, \mathbf{w} is measurement noise, and \mathbf{S} is a measurement noise covariance matrix. The state vector $\mathbf{x} \in \mathbb{R}^3$ is an estimate of the filtered balloon's 3D position in the static global coordinate frame \mathcal{W} . The measurement vector $\mathbf{z} \in \mathbb{R}^3$ is its position, as detected by the method described in Subsection 4.1, transformed to \mathcal{W} . The matrices are defined as

$$\mathbf{A} = \mathbf{I}_{3 \times 3}, \quad \mathbf{Q} = \sigma_v \mathbf{I}_{3 \times 3}, \quad (3)$$

$$\mathbf{H} = \mathbf{I}_{3 \times 3}, \quad \mathbf{S}_{[k]} = \mathbf{R}_{[k]} \begin{bmatrix} \sigma_{xy} & 0 & 0 \\ 0 & \sigma_{xy} & 0 \\ 0 & 0 & \sigma_{z[k]} \end{bmatrix} \mathbf{R}_{[k]}^T, \quad (4)$$

where $\mathbf{I}_{3 \times 3}$ is a 3×3 identity matrix, and σ_v and σ_{xy} are empirically determined parameters. The value of $\sigma_{z[k]}$ is scaled according to the distance of the detection as

$$\sigma_{z[k]} = \max \left\{ \sigma_{z,\min}, \sigma_{z,\text{coeff}} \cdot \|\mathbf{z}_{[k]} - \mathbf{r}_{[k]}\|^2 \right\}, \quad (5)$$

where $\mathbf{r}_{[k]}$ is the current MAV position in the static global coordinate system, with parameters of $\sigma_{z,\min}$ and $\sigma_{z,\text{coeff}}$. The matrix $\mathbf{R}_{[k]}$ is the rotation matrix which aligns a unit vector $[0, 0, 1]^T$ to the vector $\mathbf{z}_{[k]} - \mathbf{r}_{[k]}$ (direction of the detection from the camera). The scaling and the rotation of the measurement noise covariance matrix $\mathbf{S}_{[k]}$ expresses the higher uncertainty of the distance estimation when the target is far away, as is important for the improved convergence of the filter.

The position filter is initialized by the state machine in the *Detection confirmation* state (see Subsection 4.3) with an initial balloon detection $\mathbf{z}_{[0]}$. Each following measurement $\mathbf{z}_{[k]}$ is selected as the detection from $\mathcal{T}_{[k]}$, which is closest to $\mathbf{z}_{[k-1]}$ and lies within a gating distance d_{gate} . If no such detection is available, the filter is not updated. If the filter is not updated for a duration of t_{FP} , the filter is reset and the initial balloon detection is marked as a false positive. Such false positives are handled by the state machine, as is explained in the next section.

¹² <http://wiki.ros.org/rosbag>

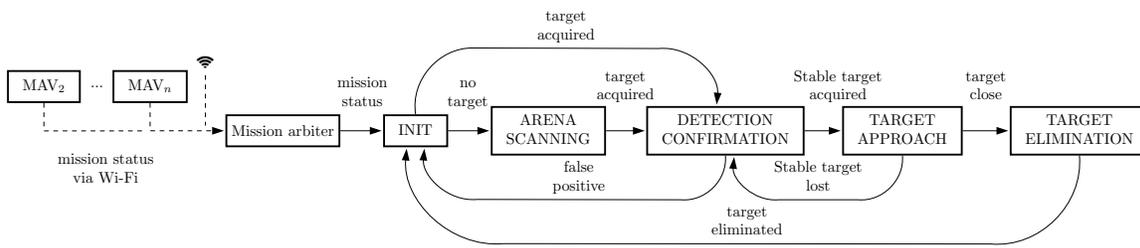


Figure 5. A diagram of the multi robot state machine for MAV_1 . The Mission arbiter receives information about the mission status of the colleague MAVs via Wi-Fi.

4.3. State Machine with coordination of multiple agents

The state machine controls the target elimination mission based on data from the various system modules (see Section 3). Output of the state machine is a setpoint trajectory for the MAV control pipeline.

The state machine (presented in Figure 5) has 5 states:

- *Init* - the initial state of the state machine. During this state, the mission status is processed and the arena is selected.
- *Arena scanning* - in this state, the MAV flies through the arena while scanning for suitable targets.
- *Detection confirmation* - this state is activated after locating a target in the *Arena scanning*. The purpose of this state is to reject potential false positive detections. Once the target is confirmed as a true positive, the *Target approach* state is then activated.
- *Target approach* - after passing the checks in the *Detection confirmation* state, the target is approached up to a distance d_b .
- *Target elimination* - this state controls the final step of the system: destroying the target.

Although no *a priori* information about target positions was available in the MBZIRC 2020 competition, this task may be solved more efficiently in a real MAV-hunting scenario by integrating information from an external MAV detection system. Such systems provide a rough location of detected MAVs, which 3D position may be used as an initial estimate of the position of a target that is to be eliminated.

The transitions between different states of the state machine are visualized in Figure 5. The state machine executes the search-and-destroy plan, while also dealing with possible false positive detections, hardware issues, and emergencies. Each state can switch back to *Init* or to a previous state to recover from a false positive detection or in the event of a module failure (e.g. balloon position estimation or camera failure). The *Mission arbiter* is described in Subsection 4.4.

To improve robustness and prevent a deadlock of the state machine in the event of a hardware failure or a software glitch, the states have a maximal execution time. These states are namely *Detection confirmation*, *Target approach*, and *Target elimination*. Every time one of the states exceeds the respective limit, the currently selected target b_c is marked as a forbidden zone and the state is switched back to *Init*. The detections that are classified as false positives by the Kalman filter subsystem (see Subsection 4.2) are also marked as forbidden zones. A forbidden zone is defined as a sphere with a center x_f , a radius r_f , and an expiration duration t_f . Any detection inside a forbidden zone is ignored. After time t_f , the forbidden zone is deleted and the State Machine can select this target again. This lends flexibility to the state machine so that the MAV is able to skip problematic targets (e.g., when it took too much time to eliminate such a target or it was a false positive detection) and continue searching and focusing on others.

All together, these components of the state machine helped to counter deadlocks and failures of the state machine itself. These changes have been applied incrementally. A further description of the lessons learned from this process are shown in Section 7.

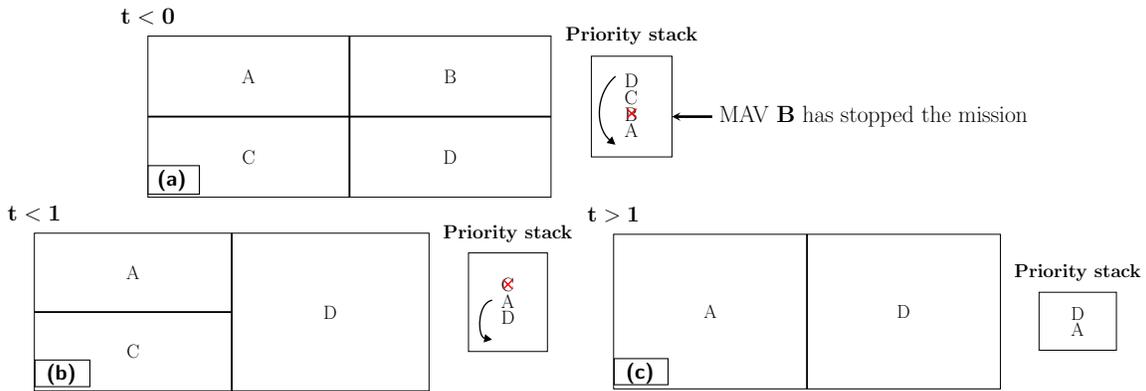


Figure 6. An illustration of the proposed arena-splitting redundancy approach for multiple robots. **A, B, C, D** are the MAVs authorized to work in the respective zones. **Priority stack** shows the current priority levels of active MAVs, ordered from top to bottom (highest priority to lowest). The initial split of the arena is seen in image (a). At time $t=0$, the MAV **B** has stopped mission execution. Its neighbor with the highest priority (**D**) overtakes its arena, resulting in the split displayed in (b). After overtaking the arena of its neighbor, the MAV **D** is moved to the bottom of the priority stack. The same procedure is repeated again during $t=1$ when the MAV **C** fails, resulting in the arena split displayed in (c).

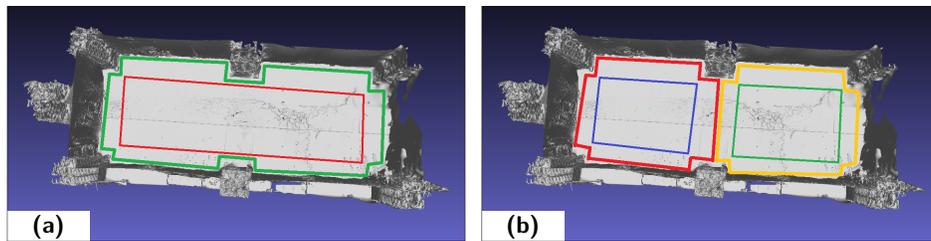
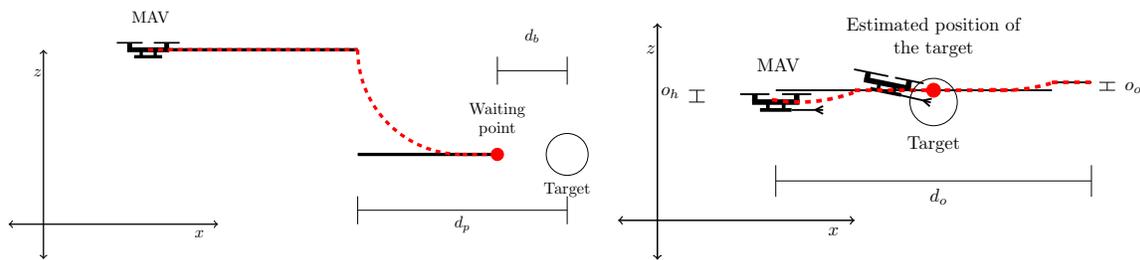


Figure 7. Illustration of the different areas highlighted on a 3D model of the arena. Separation of the arena for a single MAV is shown in (a). The green polygon is the *Operational area* of the MAV and the red rectangle represents the *Scanning area*. An arena split for two MAVs is shown in (b), where the yellow and red polygons are the *Operational areas* for the respective MAVs, and the blue and green rectangles outline the *Scanning areas*.

4.4. Coordination of multiple agents

Having been inspired by the requirements of end-users of real multi-target and multi-hunters systems, we decided to split the arena into equal parts based on the number of MAVs in the team. To ensure high robustness of the system, redundancy of the MAV team is leveraged to compensate potential failure of some of the agents. The MAVs use wireless communication to periodically report their status to other members of the team. When a MAV must prematurely end its mission- be it due to a sensor failure, expended battery, or any other reason- it notifies the other team members. Its operational area is then taken over by the neighbor with the highest priority. Similarly, when no message is received from an MAV for a duration t_s , it is assumed that its onboard computer is no longer operational and the same method is applied to compensate for its failure. The algorithm is illustrated in Figure 6.

During the MBZIRC 2020 competition, the team consisted of two MAVs which were communicating over Wi-Fi. The MBZIRC arena was laid out as a polygon with its sides covered in netting for security purposes. The arena was split into two *Operational areas* (one for each MAV), as illustrated in Figure 7. An *Operational area* encompassed the polygonal-shaped area where the respective MAV could perform target elimination and was also separated from *Operational areas* of other MAVs. In order to simplify the scanning trajectory generation, each *Operational area* contained a sub-area called the *Scanning area* in which the scanning is planned. The *Scanning area* has the shape of



(a) An example of a *Target approach* trajectory to a distance d_b from the target. The descent to the height of the target begins when the MAV is closer to the target than d_p . The waiting point is where the transition between the *Target approach* and *Target elimination* states occurs. The MAV waits here while the elimination trajectory is planned using the stabilized position of the target.

(b) An illustration of *Target elimination* trajectory planning. The red dot is the point of interaction with the target, offset in height by o_h from the target's center. An overshoot offset o_o is added to the last point of the trajectory to avoid remnants of the eliminated target. d_o is the total length of the elimination trajectory.

Figure 8. Illustration of the *Target approach* trajectory (8a) and the *Target elimination* trajectory (8b). The solid black line represents the planned trajectory and the red dashed line represents the reference trajectory after reshaping by the MPC. The trajectory parameters were empirically tuned to provide good estimation of the target and to position the end-effectors directly at the center of the target in order to increase the damage on impact and to avoid entanglement of the popped balloon in the propellers.

the largest axis-aligned inscribed rectangle inside the respective *Operational area*. The presented redundancy approach ensured that even if one of the team members failed in some of the trials, the mission was successfully finished with all targets eliminated every time, as is discussed in Section 5. In the rare cases when multiple MAVs attempt elimination of the same target at the common edge of their *Operational areas*, collision avoidance is ensured by our previously published collision avoidance mechanism [Baca et al., 2018].

4.5. Motion planning for target elimination

The vision-based detection pipeline produces estimates of the target positions, which are then filtered by the state machine to select a primary target. The primary target is selected based on the Euclidean distance between its estimated position and the current position of the MAV. The closest target is selected and the approach step follows.

During the *Target approach* state (see Subsection 4.3), the MAV approaches the balloon to the distance d_b . The approach trajectory is visualized in Figure 8a. The delayed descent to the target height is implemented in order to reduce the risk of collisions during the approach. This solution was sufficient for the MBZIRC 2020 Competition where the only obstacles were the metal poles to which the targets were tethered (see Figure 1). These poles were also guaranteed to not be closer to the targets than d_p . In an environment with denser obstacles, a more sophisticated planning method would have to be employed, such as the method found in [Usenko et al., 2017] which could be used with an onboard stereo camera. To avoid a collision with the tether pole of the target, the trajectory during the *Target elimination* state also changes height based on the MAV's distance from the target (see Figure 8b). During the experimental phase, we experienced an issue when the MAV eliminated a target at a speed of 3.5 m/s. The remnants of the balloon were then propelled in the same direction as the MAV, entangling into its rotors and nearly causing a crash. To avoid such problems, we have added a trajectory height offset following the elimination of the target, to ensure that the MAV avoids collisions with the remnants of the balloons.

4.5.1. Position reference deadband

The position of the target estimated by the *Target detection* subsystem is noisy in practice. For example, there were ≈ 0.2 m jumps in the estimated position of a single target under the conditions

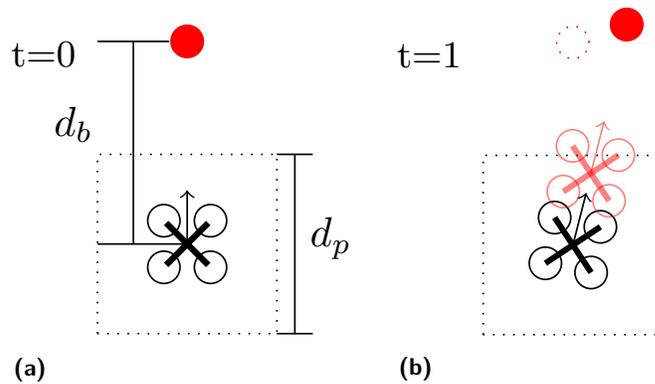


Figure 9. An illustration of the proposed deadband scheme. The deadband area with size d_p is marked with a dashed line, and distance to the target is d_b . At time $t = 0$ (a), the MAV is commanded to be at a distance d from the target (the red filled circle) before transitioning to the *Target elimination* state. At time $t = 1$ (b), the estimated position of the target (represented by the red filled circle) has moved relative to its previous estimated position (the red dotted circle). This can shift the MAV position reference and potentially cause oscillations. However, since the new MAV position (red) is inside the deadband area, it is ignored and only the heading is changed (the black MAV position in (b)). This approach effectively keeps the target in sight while preventing oscillations in the MAV position, and in the position estimation of the target due to noise.

of the competition. Because the estimated position of the target is used for planning the MAV’s trajectory, this noise may lead to positive-feedback oscillation between the MAV position together with the estimated object reference, as was the case during the competition. Due to the overall complexity of the system, we were unable to determine the exact cause of the positive feedback causing the oscillations, but they remained present to some degree with various settings and timings. A suspected culprit is an unknown non-linearity in the target distance estimation algorithm described in [Subsection 4.1](#), possibly related to switching between distance estimation based on the depth image and based on the apparent size of the object’s projection.

To prevent the MAV from oscillating, deadbanding of the position reference [[Johnson, 1999](#)] was used. With deadbanding, a neutral zone around the current MAV position is defined. If the newly calculated reference position is inside this zone, only the heading of the MAV is changed and not its position. When size of the neutral zone is chosen appropriately, this approach completely mitigates the oscillations and prevents destabilization of the system. The proposed deadband scheme is illustrated in [Figure 9](#).

4.6. Arena scanning

To provide a complete description of the MBZIRC 2020 system, let us also briefly detail the arena scanning method. Note, that this process is replaced by an external localization system in the autonomous MAV hunting missions we are targeting for the future development of this system (see [Section 6](#)). The goal of the scanning trajectory is to access the entire arena and to provide the camera as much good viewing coverage as possible. With respect to this goal, a planning approach was designed:

1. Go to one edge of the arena (with the lowest Y-coordinate).
2. Move s_{scan} distance along the X-axis.
3. Go to the other edge of the arena (with the highest Y-coordinate).
4. Move s_{scan} distance along the X-axis.
5. If the end of the arena is reached, stop the plan - else, go to step 1.

An illustration of this plan is shown in [Figure 10](#). The arrows represent heading of the MAV in the corresponding step of the plan. The MAV faces in the direction of the flight throughout the

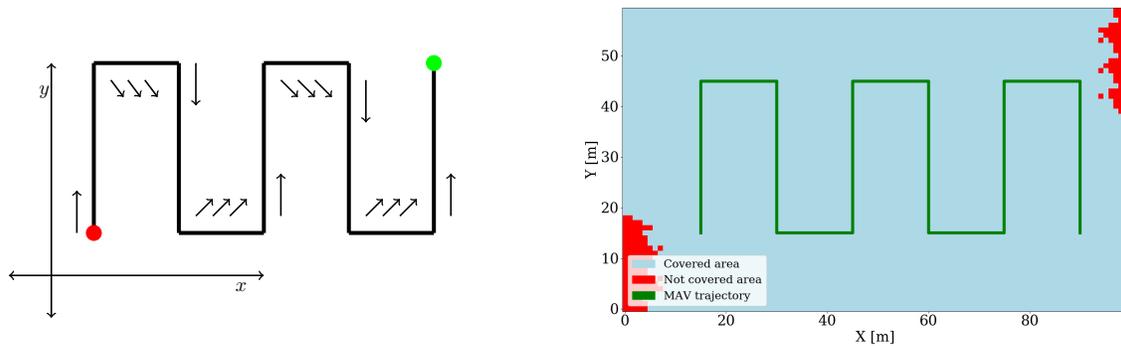


Figure 10. An illustration of the proposed scanning plan is shown on the left. The red dot is the starting point and the green dot is the finishing point of the planned trajectory. The arrows represent heading of the MAV at different positions during the trajectory. The resulting coverage (assuming a 30 m detection range and 69.4° Field Of View (FOV) according to the detector’s parameters) is shown on the right. The areas not covered by the first scan (marked red) are covered when the MAV starts the scan trajectory again (in the opposite direction).

trajectory except for the short turns, where it faces inwards. This is to give the camera a better view of the inner part of the arena where the targets are most likely to be located. To reduce the tilt of the vehicle while turning, the speed near the corners of the arena is decreased.

Since the arena is a non-convex polygon, its width differs at each point. The scanning trajectory is planned in a rectangle-shaped *Scanning area* inside of the *Operational area*, as shown in Figure 7. The algorithm produces a trajectory from the current position of the MAV to the furthest side of the arena. The trajectory is re-planned once per second to accommodate for GNSS drift and changes of the *Scanning area*, such as when the area of another team-member has to be taken over due to a failure.

5. Evaluation

The system presented in this paper was extensively tested in different environments, conditions, and with a variety of colors of the targets (yellow, red, and green balloons) as shown in Figure 17. The system proved to be robust to environmental changes, different lighting conditions, and varied wind and terrain conditions. These were key aspects for the competition and for real MAV hunting systems as well. In both cases, the system must be ready for an immediate deployment whenever it is triggered to start- either by the start of the competition trial or by an alert from the external MAV detection system. It is not possible to wait for certain conditions, such as favorable weather or light conditions, and thus reliability of the system is the most important factor.

5.1. The MAV platform

The MAV platform designed for the given task consists of off-the-shelf components that are mounted together using 3D printed parts. The base of the MAV is a Tarot T650 quadcopter frame with Tarot motors that are connected to a Pixhawk 4 flight controller unit¹³ and an Intel NUC¹⁴ onboard computer with an Intel i7-8559U CPU, 8 GB of RAM, and an onboard integrated GPU Intel Iris Plus Graphics 655 (see Figure 11). The computer runs the Ubuntu 18.04 LTS operating system.

The Pixhawk 4 contains several integrated sensors, including a GNSS receiver with a magnetometer, gyroscope, an IMU, and a barometer which are used for basic stabilization and self-localization of the vehicle. The Pixhawk is connected to the motors through ESCs which drive the brushless motors

¹³ https://docs.px4.io/v1.9.0/en/flight_controller/pixhawk4.html

¹⁴ <https://www.intel.com/content/www/us/en/products/boards-kits/nuc/kits/nuc8i7beh.html>

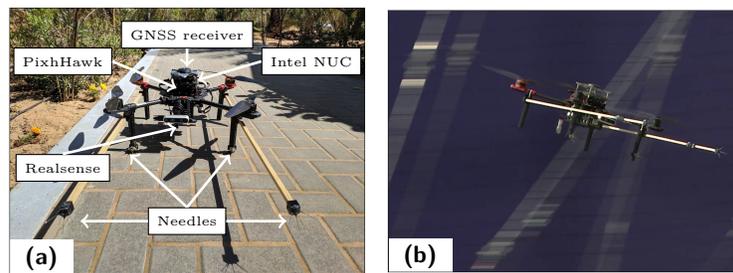


Figure 11. The MAV platform developed and used by the Multi Robot Systems (MRS) Group for the MBZIRC 2020 Competition. Figure (a) depicts the main parts of the platform and (b) shows its deployment during the competition.

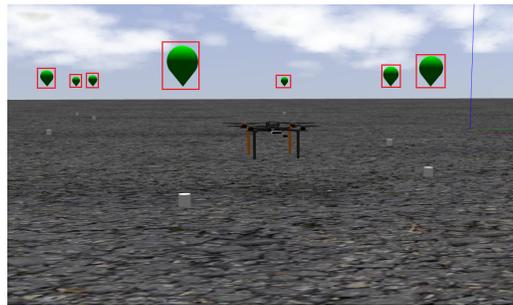


Figure 12. A model of the T650 MAV that was used for experiments and competition in the Gazebo simulator. The targets are highlighted with red rectangles.

according to the received attitude commands from the control pipeline which runs on the onboard computer. Intel RealSense D435¹⁵ stereo camera is used to provide color and depth images for detection of the targets. The color images are with a 1920×1080 pixel resolution and a $69.4^\circ \times 42.5^\circ$ horizontal \times vertical FOV. The color image is supplied at a maximum of 30 Frames Per Second (FPS) in this configuration. The depth images have a $87.4^\circ \times 58^\circ$ FOV and a 848×480 resolution at 30 FPS.

5.2. Simulation

We have utilized the ROS-compatible *Gazebo* simulator to provide a realistic physics simulation. Each phase of the described work was tested in simulation. The MAV platform was modeled according to the real-world model (see Figure 12), including properties of the motors, mass of the mechanical elements, etc. Oscillations of the balloon positions due to wind were simulated using the Gazebo physics engine. The shape of the simulated arena copied the competition arena (see Figure 13) and the targets positions were changed randomly in each run of the simulation.

Combining these elements provided a realistic simulation of the experiments in the sense that the performance of the MAVs in the simulation represents the real-world tests well (as it is indicated by the similar results obtained in simulations and in the real-world deployments). Results of the simulations are presented in Figure 14 and Figure 15. The results show that a larger number of team-members leads to a decrease in the average mission time, supporting the multi-robot approach to the problem. The elimination times for a two-member team with one MAV failing during the simulation correspond well with the system performance observed in the real-world deployments (compare Figure 14 with Figure 23).

¹⁵ <https://www.intelrealsense.com/depth-camera-d435/>

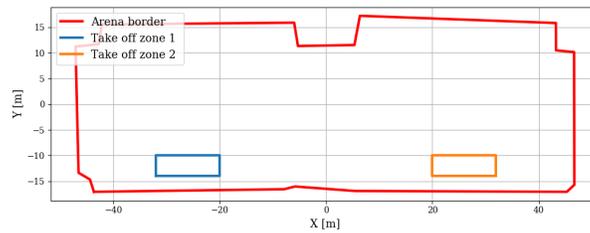


Figure 13. Top-down view of the MBZIRC 2020 Challenge 1 arena map.

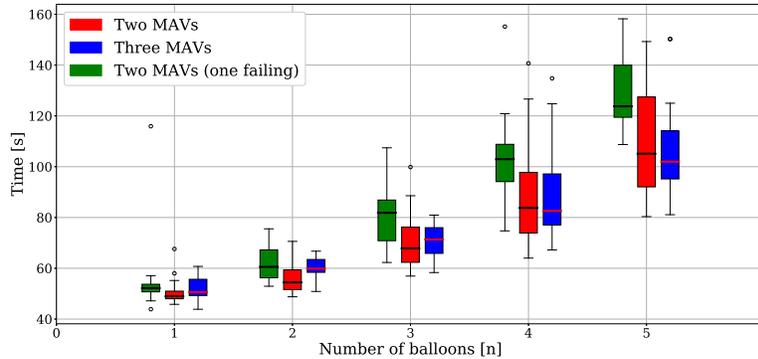


Figure 14. Average time of elimination of the n -th target since mission start for different team configurations (results from 20 runs of the simulation).

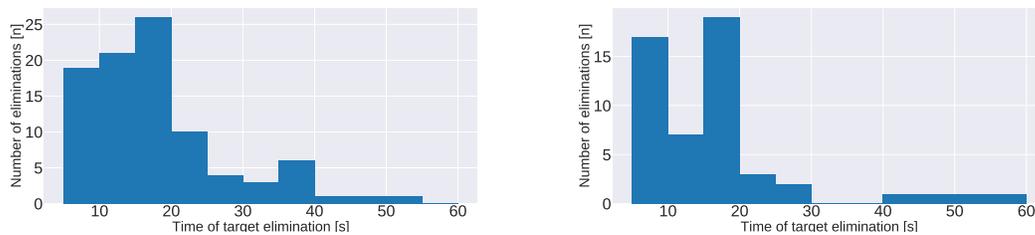


Figure 15. Histograms of duration between consecutive target eliminations - time of finding and eliminating any target (results from the same simulation runs with two (left histogram) and with three MAVs (histogram on the right) as [Figure 14](#)).

5.3. Experimental setup

The competition trials lasted approximately 15 minutes and our preliminary experiments lasted about the same time as the competition. As previously stated, the arena was a non-convex polygon (the top-down view of the arena during the competition is shown in [Figure 13](#)). Balloons of different colors (green, yellow, and red) were used for tests performed in our custom experimental arenas in the desert of the United Arab Emirates, and in the Czech Republic (see [Figure 16](#) and [Figure 17](#)). In each of the tests, five or six balloons were randomly placed in the arena (five were placed during the competition).

The preliminary experiment included many tests under various conditions, all with the aim to identify potential problems and incrementally develop an overall system to prevent or overcome them (specific examples and their solutions are listed in [Section 7](#)). The MBZIRC 2020 Competition included 3 test runs prior to the competition trials, which we utilized for specific purposes as follows:

1. *Systems test*: Two targets were set up, but the autonomous system was not deployed as this test was focused on verifying the equipment, Wi-Fi communication, and measuring the arena.



Figure 16. Images from the onboard cameras of the MAVs during the real-world experiments. *From left to right:* field in the Czech Republic, desert in the United Arab Emirates, the MBZIRC competition site in Abu Dhabi.



Figure 17. Photos of the experimental sites (a) in the Czech Republic, (b) in a desert near Abu Dhabi (b), and during the MBZIRC 2020 Competition (c).

2. *First autonomy test:* Two targets were set up and the autonomous elimination system was deployed. However, we had to test our solution with only one MAV as the other had experienced a mechanical hardware issue.
3. *Second autonomy test:* All five targets were set up and the autonomous elimination system was deployed as a full team of two MAVs. During the test, one of the MAVs encountered a software error and had to land, requiring the other MAV to overtake its arena section, enabling us to evaluate the multi-robot redundancy scheme.

After the test runs, there were two competition trials of Challenge 1 and one for the Grand Challenge. Results of the experiments are presented and discussed in the following sections.

5.4. Performance of the MAV control subsystem

During the tests, the MAV control system was extensively tested under extreme conditions and demonstrated resistance to unexpected situations. Such unexpected events included strong sudden winds and an instance where the target balloon failed to pop, producing a large control error (other situations are discussed also in [Section 7](#)). The top two graphs in [Figure 18](#) show how the planned trajectory was executed by the MAV in real-world deployment (compared with [Figure 8a](#)). [Figure 19](#) presents photos from the competition corresponding to manoeuvre in these graphs. The bottom two graphs in [Figure 18](#) present the same manoeuvre during one of the simulated experiments, demonstrating high fidelity of the simulation setup as discussed in a previous section.

The arena scanning method utilized different constraints of the MAV dynamics to provide 97.05% arena coverage. These constraints are limitations enforced on MAV speed and acceleration in the horizontal and vertical planes, as well as limitations on how fast the heading may change. In [Table 1](#), the constraints that were used are shown. The **Sweeping** constraints enable a fast heading change during the turns, but without significant acceleration to avoid tilt of the MAV. This is done to ensure that the camera of the MAV is always looking forward and searching for targets. The **Attack** constraints limit the speed and acceleration less to enable agile maneuvering during approach and elimination of the target. The resulting MAV trajectories and the arena separations for the multi-robot cooperation during the competition are shown in [Figure 20](#). This approach proved to be

Table 1. The constraints of MAV dynamics used by the MPC tracker.

constraint	horizontal		ascent		descent		heading	
	speed	acc.	speed	acc.	speed	acc.	speed	acc.
Sweeping	5 m s^{-1}	1.5 m s^{-2}	2 m s^{-1}	2 m s^{-2}	2 m s^{-1}	3 m s^{-2}	3.14 rad s^{-1}	5 rad s^{-2}
Attack	8.3 m s^{-1}	3 m s^{-2}	2 m s^{-1}	2 m s^{-2}	2 m s^{-1}	3 m s^{-2}	3.14 rad s^{-1}	5 rad s^{-2}

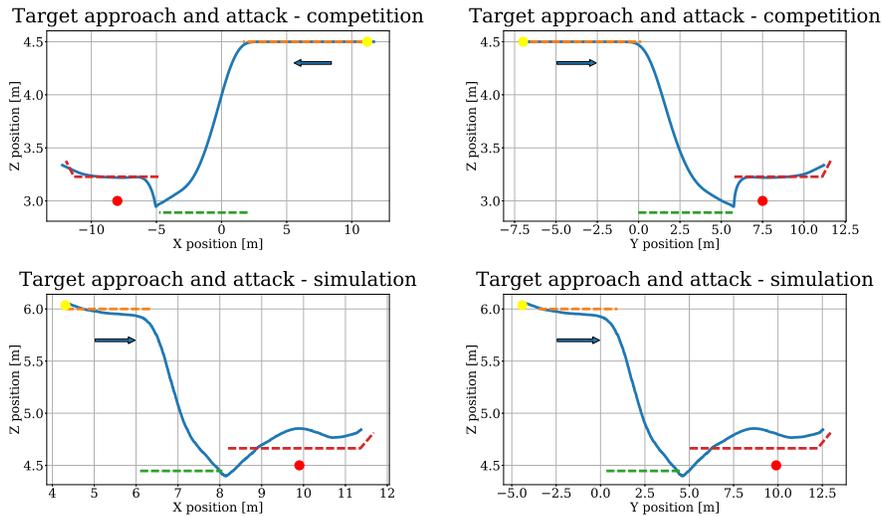


Figure 18. Graphs of the MAV's position during the *Target approach* and *Target elimination* states. The dashed lines represent the desired trajectories output from the presented *Mission & Navigation pipeline* (before reshaping with respect to MAV dynamics by the MPC tracker). The solid blue lines represent the executed MAV trajectory. The blue arrows show the direction of flight. The red dot indicates the position of the target and the yellow dot indicates the position where the MAV has selected the target for elimination. The MAV is following a trajectory pattern that was introduced in [Subsection 4.5](#).

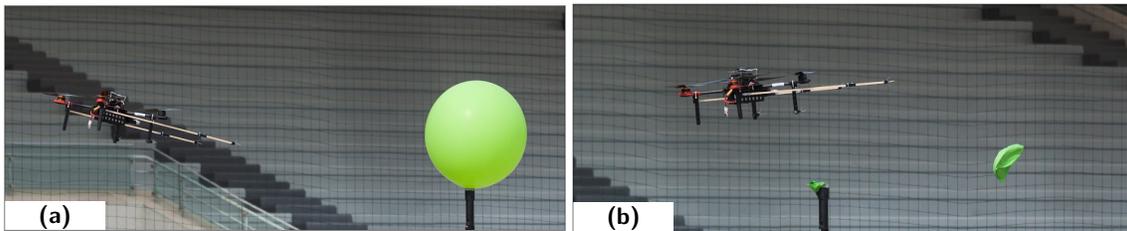


Figure 19. The MAV eliminates the balloon during the first trial of Challenge 1.

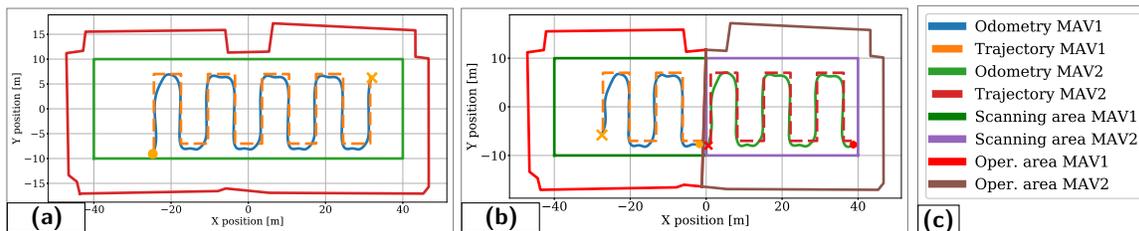


Figure 20. A top-down view of the MAVs scanning the arena during the Grand Challenge of MBZIRC 2020 (a) and with two MAVs during the second trial of Challenge 1 (b). The start of the scan is marked with a circle and finished with a cross. The plot legend is shown on (c).

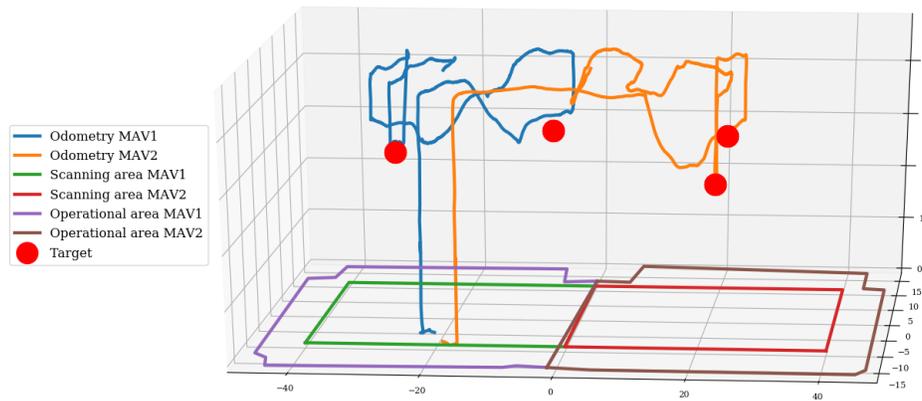


Figure 21. A plot showing trajectories of two MAVs performing their missions simultaneously during the second trial of Challenge 1 of the MBZIRC 2020. Two targets were eliminated by MAV1 (the two leftmost targets inside the violet operational zone) and the other two were eliminated by MAV2 (the two rightmost targets inside the brown operational zone). There were five targets in total for this trial, but the last target was eliminated solely by MAV1 after a reset which was triggered by a mechanical issue of MAV2.

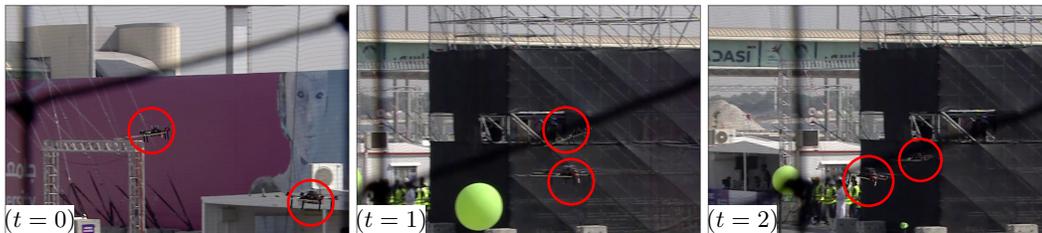


Figure 22. Two MAVs cooperatively search for and interact with targets at the MBZIRC 2020 Challenge 1 arena during the second trial. The plot of their movements is shown in Figure 19.

effective in practice as all targets were identified (and eliminated) in all the competition trials, as well as in the simulated experiments.

5.5. MBZIRC 2020 Competition performance

Numerous simulations and experiments prior to the competition evaluated the robustness of the state machine in the event of various failure scenarios. During all trials of the competition, there was no such case observed. The resulting algorithm worked as planned state-by-state thanks to its robust design and built-in fail-safe mechanisms. A video of one of the experiments is available online¹⁶.

The trajectories of the two MAVs executing the mission cooperatively during Challenge 1 are plotted in Figure 21. The corresponding photos from this run are shown in Figure 22. The plot shows the process of scanning the arena, spotting the target, executing approaching trajectory, and then destroying the target as performed repeatedly by two MAVs simultaneously. In consideration of our performance for the task being discussed in this paper, the two main criteria to be evaluated are reliability and total mission time. The proposed system was deployed four times in the MBZIRC 2020 Competition with all targets having been destroyed in all trials. This produced the best score among all 22 teams selected for the MBZIRC 2020 Final from the 180 registered teams. This performance and reliability significantly contributed to achieving second place in the Challenge 1 of MBZIRC 2020 and first place in the MBZIRC 2020 Grand Challenge.

¹⁶ <http://mrs.felk.cvut.cz/mbzirc-2020-balloon-popping>

Table 2. The results of the multi-robot team in the different experiments. During the rehearsal and first trial, one MAV crashed due to a hardware failure of one of the components. Time is calculated only when at least one MAV was in the air and until the last balloon was popped. Initial number of MAVs is written in brackets.

Trial	MAVs	Time	Note
Rehearsal 3	1(2)	163 s	One MAV crashed due to a disarm
Trial 1	1(2)	135 s	One MAV crashed due to a disarm
Trial 2	2	104 s	-
Grand Challenge	1	129 s	Only one MAV was used for balloon popping
Desert	2	106 s	Test in a custom arena with 4 balloons instead of 5

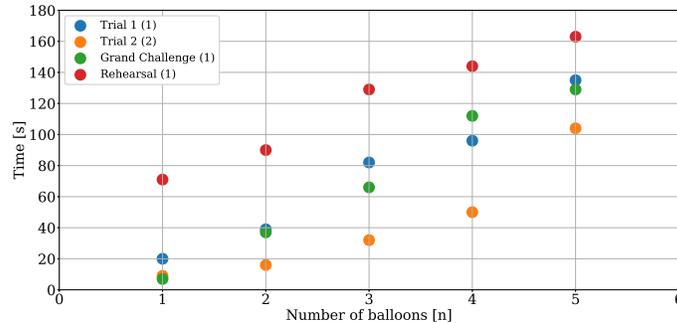


Figure 23. Elimination times of the n -th balloon in different competition trials. The number of active MAVs during each trial is shown in the legend brackets.

Table 2 presents the times during all MBZIRC 2020 trials and the number of MAVs that started and finished in each particular trial. In Rehearsal 3 and Trial 1, one of the two MAVs finished its mission prematurely due to a hardware failure. The remaining MAV was able to take over its tasks to finish the mission to completion. This verifies the multi-agent approach proposed in Subsection 4.4 under real conditions.

Figure 23 compares the performance elimination times of individual balloons during the different trials. With both MAVs functional throughout the entire trial, the five balloons are eliminated up to 20% faster than was the case with only one MAV performing the task.

6. Going Beyond MBZIRC

One of the primary motivations of the MBZIRC 2020 was to advance the research in MAV safety and autonomous multi-robot interception. Our team has expertise in this area because of our participation in the Eagle.One project, in which we have developed a working prototype of an Autonomous Aerial Interception System (AAIS). The MBZIRC 2020 Competition has given us the hands-on experience of real field deployments of a multi-robot solution for the MAV interception scenario. This practice inspired us for further development of the AAIS. The approaches presented in this paper and the experiences gained will be integrated with our current AAIS solution by implementing a cooperative, multi-robot, target-elimination system. We expect the extended system to improve the robustness of the autonomous interception similar to what has been demonstrated in the experimental evaluation of the methods presented in this paper.

7. Lessons Learned

Our system’s performance and robustness are due, in large part, to extensive preliminary testing that enabled solving numerous potential problems before real deployment in the competition. In this section we describe the most important issues we have encountered and their solutions. The number and variety of the problems encountered and the performance of the resulting system in the

competition highlights the importance of sufficient testing of robotic systems which are aimed to be used in practice under various conditions.

7.1. Mechanical and electronic issues

The main focus of this work is on real-world deployment of MAVs in diverse conditions. This effort has involved extensive use of the MAV platforms and sensors in a variety of seasons and climates. In particular, the MAVs were deployed during the spring, summer, and autumn (10-15 degrees Celsius) of a moderate climate and in desert conditions of intense sunlight and temperatures of up to 35 degrees Celsius. The final version of the system that is presented in this paper is a result of this thorough testing process and incremental modification and improvement of the design.

The most critical mechanical issue we encountered related to sensor connectivity. Due to the high number of tests (around 40-50 for each of the MAVs) and extreme conditions, the electronic connectors tended to gather a lot of sand and dust. This conditions, electrical connectors sometimes caused spontaneous disconnection of some sensors and failure of the respective MAV team-members. The stereo camera sensor used for the detection (the Realsense D435) also suffered from overheating because of dust, sand and extreme ambient temperatures. Even though the disconnection problems were partially mitigated by utilizing locking connectors and by mechanically fastening them in the desired position, these issues were the main reason that inspired us to implement the failsafe methods, described in Sections 4.3 and 4.4.

Additionally, the electromagnetic noise from USB3 connectors with high bandwidth (such as that of the Realsense D435 camera) proved to be problematic, deteriorating signal quality of the onboard GNSS receiver. Since the MAVs rely on a GNSS for navigation, additional electromagnetic shielding had to be employed to ensure sufficient precision of the navigation.

The multi-robot cooperation scheme described in Section 4.4 proved to be robust in handling less common and unexpected hardware failures as well, such as:

- On completing one balloon attack, part of the popped target became entangled in one of the MAV propellers, causing it to crash.
- A defective battery caused one MAV to terminate its mission prematurely due to a lowered energy capacity and thus a shorter flight time.
- Due to drift in the GNSS-measured MAV position, a part of the MAV collided with one of the metal poles holding the balloons (see Figure 1) and the MAV crashed as a result.

In all of these cases, another team-member was able to finish the mission, highlighting the importance of the multi-robot approach.

7.2. State Machine

Based on the experience gained from the numerous real-world deployments of the system, we have identified the false-positive target filtering, the multi-robot cooperation, and the state-deadlock avoidance as the most critical state-machine elements regarding safety and robustness. During testing, the system encountered false-positive detections outside the arena (artificial, placed by us, and natural when we used green balloons as targets and there were bushes around the area with a similar color). The final designed approach is robust and managed to successfully complete the mission in these situations.

7.2.1. False-positive filtering

While our false-positive rejection approach (Section 4.3) is not directly related to safety or robustness of the mission, it significantly improves balloon-elimination performance. During the preliminary experiments, we have encountered problems with false-positive detections of objects visually similar to the targets (e.g. green bushes), which were present in the operational area of the MAV team.

Therefore, we were able to identify vulnerabilities in the state machine and vision algorithm early and adapt our system accordingly. Specifically, the false detections caused that a team-member would waste time trying to eliminate a non-existent target, which was completely mitigated by the false-positive rejection method.

7.2.2. Multi-robot cooperation

The initial design of the multi-robot cooperation scheme, described in Section 4.3, worked slightly differently. Each team-member sent a single “**failure**” message when it prematurely terminated the mission and required another team-member to take over its part of the operational area. However, when the onboard computer of one of the MAVs failed (e.g. due to a fatal crash), the “**failure**” message was not sent and the take-over did not happen, causing some of the targets to not be eliminated. This inspired us to devise the final cooperative-redundancy scheme, which utilizes continuous stream of “**operational**” messages and when the “**operational**” message is not received from one of the team-members, another the take-over is activated. This strategy proved crucial to system robustness, such as failure of the onboard PC due to another fatal crash occurred during the competition.

7.2.3. State-deadlock avoidance

Due to drift of the GNSS-measured position of the MAV or due to false-positive detections, the MAV observed a target at the edge of its current operational area, whose position observed, at the edge of its current operational area, a target whose position appeared to oscillate in and out of the arena. . This error sometimes caused the MAV to get stuck in the *Detection confirmation* state (see Section 4.3), infinitely waiting to obtain a stable target. A similar situation sometimes happened with the other states as well. Adding conditions for maximum state-duration and forbidden zones mitigated this problem and, in combination with the rest of the safety features mentioned in this section ensured overall, multi-robot, system reliability.

8. Conclusion

This paper had presented a detailed description and evaluation of our solution to the MBZIRC 2020 Challenge 1 balloon-hunting task has been provided in this paper. MBZIRC tasks are designed to push development of novel, vision-based detection and control algorithms and, most importantly, to expand research in multi-robot cooperation. After analyzing the balloon hunting problem in simulations and preliminary experiments, we developed a multi-robot approach employing cooperative search and elimination of targets in a designated area. The system has repeatedly proven itself to be robust as it has performed reliably in different environments of varied terrain and weather conditions and has demonstrated the viability of a multi-robot scheme for this task. This system allowed us to achieve second place in Challenge 1 among 22 other competitor teams and first place in the final Grand Challenge. We have also developed tools and approaches that can facilitate further research in cooperative aerial interception. We continue actively contributing to this research and developing, thus accomplishing the primary purpose of the competition. The open-source software will be released on https://github.com/ctu-mrs/mbzirc_2020_balloon_popping.

Acknowledgements

We thank the University of Pennsylvania and the New York University for the collaboration as the members of the CTU-UPenn-NYU team. This work would not have been possible without the tireless dedication of all the team members and support and patience from their families. Also, special thanks go to the Faculty of Applied Sciences of Ukrainian Catholic University without their help and support, it would not be possible. The presented work has been supported by Khalifa University.

ORCID

Yurii Stasinchuk  <https://orcid.org/0000-0002-2197-1442>

Matouš Vrba  <https://orcid.org/0000-0002-4823-8291>

Tomáš Báča  <https://orcid.org/0000-0001-9649-8277>

Vojtěch Spurný  <https://orcid.org/0000-0002-9019-1634>

Matěj Petrлік  <https://orcid.org/0000-0002-5337-9558>

Daniel Heřt  <https://orcid.org/0000-0003-1637-6806>

Martin Saska  <https://orcid.org/0000-0001-7106-3816>

References

- Baca, T., Hert, D., Loiano, G., Saska, M., and Kumar, V. (2018). Model Predictive Trajectory Tracking and Collision Avoidance for Reliable Outdoor Deployment of Unmanned Aerial Vehicles. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE.
- Baca, T., Petrлік, M., Vrba, M., Spurny, V., Penicka, R., Hert, D., and Saska, M. (2020). The MRS UAV system: Pushing the frontiers of reproducible research, real-world deployment, and education with autonomous unmanned aerial vehicles. arXiv preprint, arXiv:2008.08050.
- Beul, M., Nieuwenhuisen, M., Quenzel, J., Rosu, R. A., Horn, J., Pavlichenko, D., Houben, S., and Behnke, S. (2019). Team NimbRo at MBZIRC 2017: Fast landing on a moving target and treasure hunting with a team of micro aerial vehicles. *Journal of Field Robotics*, 36(1):204–229.
- Brust, M. R., Danoy, G., Bouvry, P., Gashi, D., Pathak, H., and Gonçalves, M. P. (2017). Defending against intrusion of malicious UAVs with networked UAV defense swarms. In *2017 IEEE Conference on Local Computer Networks (LCN)*, pages 103–111.
- Capitan, J., Merino, L., and Ollero, A. (2016). Cooperative decision-making under uncertainties for multi-target surveillance with multiples UAVs. *Journal of Intelligent & Robotic Systems*, 84(1-4): 371–386.
- Drozdowicz, J., Wielgo, M., Samczynski, P., Kulpa, K., Krzonkalla, J., Mordzonek, M., Bryl, M., and Jakielaszek, Z. (2016). 35 GHz FMCW drone detection system. In *2016 International Radar Symposium (IRS)*, pages 1–4.
- Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., and Tian, Q. (2019). Centernet: Keypoint triplets for object detection. In *IEEE International Conference on Computer Vision*, pages 6569–6578.
- Faigl, J., Vána, P., Pěnička, R., and Saska, M. (2019). Unsupervised learning-based flexible framework for surveillance planning with aerial vehicles. *Journal of Field Robotics*, 36(1):270–301.
- Fitriana, A. N., Mutijarsa, K., and Adiprawita, W. (2016). Color-based segmentation and feature detection for ball and goal post on mobile soccer robot game field. In *2016 International Conference on Information Technology Systems and Innovation (ICITSI)*, pages 1–4.
- Gai, J., Tang, L., and Steward, B. L. (2020). Automated crop plant detection based on the fusion of color and depth images for robotic weed control. *Journal of Field Robotics*, 37(1):35–52.
- International Color Consortium and others (2004). Image technology colour management-architecture, profile format, and data structure. *Specification ICC. 1: 2004-10 (Profile version 4.2. 0.0)*.
- Joblove, G. H. and Greenberg, D. (1978). Color spaces for computer graphics. In *5th annual conference on Computer graphics and interactive techniques*, pages 20–25.
- Johnson, C. D. (1999). *Process control instrumentation technology*. Prentice Hall PTR.
- Lee, T., Leok, M., and McClamroch, N. H. (2010). Geometric tracking control of a quadrotor UAV on SE(3). In *IEEE conference on decision and control (CDC)*, pages 5420–5425. IEEE.
- Li, J., Ye, D. H., Chung, T., Kolsch, M., Wachs, J., and Bouman, C. (2016). Multi-target detection and tracking from a single camera in Unmanned Aerial Vehicles (UAVs). In *2016 IEEE/RSJ IROS*.
- Liu, Y., Wan, X., Tang, H., Yi, J., Cheng, Y., and Zhang, X. (2017). Digital television based passive bistatic radar system for drone detection. In *2017 IEEE Radar Conference*, pages 1493–1497.
- Nguyen, P., Ravindranatha, M., Nguyen, A., Han, R., and Vu, T. (2016). Investigating cost-effective RF-based detection of drones. In *Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pages 17–22.
- Opromolla, R., Fasano, G., and Accardo, D. (2018). A vision-based approach to UAV detection and tracking in cooperative applications. *Sensors*, 18(10):3391–3417.

- Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). ROS: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan.
- Redmon, J. and Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In *IEEE CVPR*, pages 6517–6525.
- Rozantsev, A., Lepetit, V., and Fua, P. (2015). Flying objects detection from a single moving camera. In *IEEE CVPR*, pages 4128–4136.
- Sapkota, K. R., Roelofsen, S., Rozantsev, A., Lepetit, V., Gillet, D., Fua, P., and Martinoli, A. (2016). Vision-based unmanned aerial vehicle detection and tracking for sense and avoid systems. In *2016 IEEE/RSJ IROS*, pages 1556–1561.
- Saqib, M., Khan, S. D., Sharma, N., and Blumenstein, M. (2017). A study on detecting drones using deep convolutional neural networks. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*.
- Schlotfeldt, B., Thakur, D., Atanasov, N., Kumar, V., and Pappas, G. J. (2018). Anytime planning for decentralized multirobot active information gathering. *IEEE Robotics and Automation Letters*, 3(2):1025–1032.
- Schumann, A., Sommer, L., Klatte, J., Schuchert, T., and Beyerer, J. (2017). Deep cross-domain flying object classification for robust UAV detection. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6.
- Spurny, V., Baca, T., Saska, M., Penicka, R., Krajnik, T., Thomas, J., Thakur, D., Loianno, G., and Kumar, V. (2019). Cooperative autonomous search, grasping and delivering in a treasure hunt scenario by a team of UAVs. *Journal of Field Robotics*, 36(1):125–148.
- Suzuki, S. and Abe, K. (1985). Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46.
- Usenko, V., von Stumberg, L., Pangercic, A., and Cremers, D. (2017). Real-time trajectory replanning for MAVs using uniform B-splines and a 3D circular buffer. In *2017 IEEE/RSJ IROS*, pages 215–222.
- Vrba, M., Heřt, D., and Saska, M. (2019). Onboard marker-less detection and localization of non-cooperating drones for their safe interception by an autonomous aerial system. *IEEE Robotics and Automation Letters*, 4(4):3402–3409.
- Vrba, M. and Saska, M. (2020). Marker-less micro aerial vehicle detection and localization using convolutional neural networks. *IEEE Robotics and Automation Letters*, 5(2):2459–2466.
- Vrba, M., Stasinchuk, Y., Bába, T., Spurný, V., Petrlík, M., Heřt, D., Žaitlík, D., and Saska, M. (2020). Autonomous capturing of agile flying objects using MAVs: The MBZIRC 2020 challenge. Submitted to 2020 IEEE Transactions on Systems, Man and Cybernetics: Systems.
- Yang, K. and Quan, Q. (2020). An autonomous intercept drone with image-based visual servo. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2230–2236.

How to cite this article: Stasinchuk, Y., Vrba, M., Bába, T., Spurný, V., Petrlík, M., Heřt, D., Žaitlík, D., & Saska, M. (2022). A multi-MAV system for the autonomous elimination of multiple targets in the MBZIRC 2020 competition. *Field Robotics*, 2, 1697–1720.

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