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

Systems Article

A multipurpose mobile manipulator for autonomous firefighting and construction of outdoor structures

Meysam Basiri[1], João Gonçalves[1], José Rosa[1], Rui Bettencourt[1], Alberto Vale[2] and Pedro Lima[1]

[1]Institute for Systems and Robotics, Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal
[2]Instituto de Plasmas e Fusão Nuclear, Instituto Superior Técnico, University of Lisbon, Lisbon, Portugal

Abstract: This paper describes the implementation of a multipurpose, autonomous, mobile manipulator for building outdoor structures and for firefighting. Such a system finds applications in different industrial automation, manufacturing, and civil-construction scenarios as well as in search-and-rescue operations. This system was developed for the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2020, showcasing once again the role of major scientific competitions in advancing the state of the art and exploring solutions to open problems. The paper presents in detail the hardware and software architectures of the developed mobile manipulator, while proposing methods for: a) Building outdoor structures consisting of heterogeneous brick patterns, and b) Entering buildings to locate and extinguish fires. Solutions were successfully deployed in the near-realistic arenas of the MBZIRC 2020 competition and resulted in the first-place award for the firefighting scenario.

Keywords: mobile manipulation, emergency response, construction automation, wheeled robots, firefighting robot, MBZIRC

1. Introduction

Scientific competitions provide a common framework for the comparison of intelligent robots and autonomous systems in real world conditions (Basiri et al., 2019) and play the role of scientific experiments that appeal to both researchers and to the general public. Currently several major Grand Challenges exist in robotics around the world e.g., the Mohamed Bin Zayed International Robotics Challenge (MBZIRC), DARPA Grand Challenge, CYBATHLON, or EuRoC challenges, where the aim is to push the state of the art toward solving open problems (Dias et al., 2016). Robot competitions such as RoboCup or the European Robotics League encourage integrating different sub-disciplines of AI and robotics into complete robot systems that solve complex tasks in realistic

Received: 30 September 2020; revised: 30 March 2021; accepted: 05 April 2021; published: 29 October 2021.

Correspondence: Meysam Basiri, Institute for Systems and Robotics, Instituto Superior Técnico, University of Lisbon, 11049-001 Lisbon, Portugal, Email: meysam.basiri@tecnico.ulisboa.pt

environments, benchmarking the solutions against a shared problem (Basiri et al., 2019; Ventura et al., 2016).

A mobile manipulator is composed of a robotic manipulator mounted on a mobile base, offering both mobility and dexterity. Although the mobile base significantly extends the workspace of a robotic manipulator, it also introduces additional challenges due to the large number of degrees of freedom and the need to operate inside unstructured or semi-structured environments of the real world. Many different professional and consumer-service applications can be envisioned for mobile manipulators, such as manufacturing assistance (Hvilshøj & Bøgh, 2011), autonomous construction (Helm et al., 2012), domestic-service tasks, and human assistance (Caselli et al., 2003; Smarr et al., 2014; Stückler et al., 2012; Ventura et al., 2016), autonomous transportation of goods inside warehouses and stores (Cosma et al., 2004), and tasks involving hazardous environments (Kang et al., 2003).

This paper describes our efforts in developing an autonomous, mobile manipulator for the second and third challenges of the MBZIRC 2020 robot competition[1]. We addressed two challenging tasks: a) Locate and manipulate a set of differently sized bricks and construct a wall of predefined pattern in an outdoor environment, b) Reach and enter a building and detect and extinguish the fires inside the building. These two scenarios were among the three challenges that the MBZIRC 2020 competition posed, each to be solved by a team of ground and aerial robots, thus encouraging solutions that can potentially be employed in real-world applications such as construction, autonomous, robotic, 3D printing of civil structures, and search-and-rescue (SAR) operations. Functionalities developed for these scenario can also be employed in many other purposes and applications. The mobile manipulator we developed for this competition is currently being extended for use in inspecting and maintaining solar farms under the DURABLE[2] project.

The outline of this paper is as follows: Section 2 describes the related work on autonomous mobile manipulators and their applications for real-world problems. Section 3 describes the second and third challenges of the MBZIRC 2020 competition. Section 4 introduces the developed mobile manipulator and provides a description of the general hardware and software components. Section 5 presents the proposed approach for solving Challenge 2, and describes in detail the main functionalities developed for autonomous pick and placing of brick-shaped objects to construct a wall. Section 6 describes the method and functionalities for Challenge 3 to locate and extinguish fires inside a building. Section 7 describes the evaluation of the entire system in performing the two scenarios. Section 8 presents the outcomes and lessons learned from the competition. Finally, Section 9 concludes the paper by providing the conclusion and future work.

2. Related work

The use of autonomous mobile manipulators for executing real-world missions has received a great deal of attention over the past decade, and designs with different format ranging from wheeled-mobile robots (A. H. Korayem et al., 2019; M. H. Korayem et al., 2010) to humanoids (Bouyarmane et al., 2019), legged robots (Rehman et al., 2016) and even aerial robots (Jimenez-Cano et al., 2013) have been demonstrated. Many service applications can benefit from the use of mobile manipulators, such as manufacturing assistance in industrial environments (Meng et al., 2021; Sprunk et al., 2017), domestic service tasks (Stückler et al., 2012; Ventura et al., 2016) and human assistance (Caselli et al., 2003; Smarr et al., 2014), logistic tasks (Cosma et al., 2004; Dömel et al., 2017) and transporting objects in hazardous environments, for SAR operations (Petrlík et al., 2020; Rouček et al., 2020) and in areas with radioactive or toxic debris (West et al., 2019).

Mobile robots have become an essential tool to help humans in SAR operations (Basiri et al., 2018; Lima et al., 2003). Initially, SAR robots mainly focused on passive observation, displaying mission information while providing assistance to the operators. Recently, operations have become

[1] http://mbzirc.com/challenge/2020
[2] https://www.durableproject.eu

more active, where robots operate autonomously or semi-autonomously providing intervention on the scenario. Beyond the usage of mobile robots equipped with high sensing technologies to operate in cluttered scenarios, the designs of hybrid robots, fitted with a single manipulator (Ben-Tzvi et al., 2007) or dual manipulators (Park et al., 2017) are significant. Multiple examples can be found in the literature about SAR-like operations with robots in scenarios of natural disasters, such as earthquakes, (Kruijff et al., 2012; Matsuno & Tadokoro, 2004), or floods (Ozkan et al., 2019). During the last three decades, the number of nuclear power stations has been increasing, and serious nuclear accidents have occurred, such as the three well known events at Three Mile Island, Chernobyl, and Fukushima. Only in the latter two disasters were SAR robots employed, and there with no success, given the high levels of radiation and the limited technology readiness level (TRL) (Nagatani et al., 2013). However, the usage of nuclear technologies is still increasing and new incidents are imminent (Tsitsimpelis et al., 2019). Defence against chemical, biological, radiological, and nuclear (CBRN) threats is another on demand issue due to wars, terrorist attacks, disasters or or simple human negligence. Robots play an important role not only during the protection against these threats, but particularly during the SAR like operations after the accidents (Baums, 2017; Guzman et al., 2016; Marques et al., 2017). Fires, whether natural or man-made, deliberate or accidental, constitute another important type of disasters. This type of accidents can start in small fire flares inside of a building and, if detected and extinguished quickly, can save lives and money. Multi-rotor UAVs can easily fly close to the hotspots to extinguish them, (Aydin et al., 2019). Teams of heterogeneous robots can cooperate in detecting and combating fires in urban and forest scenarios (Sherstjuk et al., 2019; Viguria et al., 2010). Different solutions with high TRL, such as thermal cameras and portable fire extinguishers, can be used on ground robots or even on UAV multi-rotors (Aydin et al., 2019). A particular example is presented in this paper.

Similarly, the use of robots for civil applications and autonomous construction of structures can alleviate the rising demand of workforce and safety regulations and reduce the cost and time of constructions. Robot fabrication is commonly known as stationary industrial robots that operate under fixed conditions with a limited working area (Helm et al., 2014). Such solutions consist of large stationary robots with specific and pre-determined roles while constraining the work-piece size that they can act upon. The use of versatile multi-purpose mobile manipulators can allow cooperative operation directly on construction sites and to build complex structures with simple building components. This insight has led to innovative efforts to develop multi-purpose, ground-based, mobile manipulators (Gawel et al., 2019; Lussi et al., 2018; Sandy et al., 2016) and aerial robots (Mirjan et al., 2016; Willmann et al., 2012) for autonomous construction of structures.

To encourage research and development on advanced mobile manipulators for challenging real-world problems, multiple robotic Grand Challenges such as DARPA (Krotkov et al., 2017) and MBZIRC (Dias et al., 2019) competitions have defined complex scenarios requiring robots to perform a series of challenging tasks and combining perception, localization, navigation, manipulation, and even multi-robot cooperation. In Carius et al. (2018) a custom mobile manipulator for operating a valve stem was developed under the MBZIRC 2017 challenge. The same competition encouraged development of innovative airborne robot solutions to locate and pick-and-place small ground objects (Loianno et al., 2018; Spurný et al., 2019). In Schwarz et al. (2018) a mobile manipulation robot was developed for the DARPA grand challenge to execute several challenging manipulation tasks in a disaster environment.

3. MBZIRC challenges

This section provides an overview of the second and third challenges of the MBZIRC 2020 competition, while focusing more on the assignments for the unmanned ground vehicle (UGV) that was the aim of this work. The first challenge only focused on Unmanned aerial vehicles (UAVs) which is out of the scope of this paper. More details about the rules and procedure for the challenges can be obtained from the MBZIRC 2020 website.

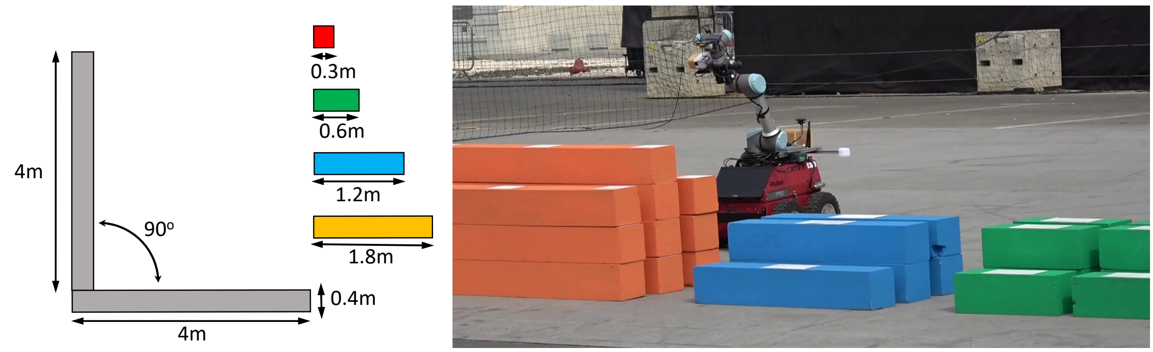


Figure 1. Left: Top view diagram illustrating the shape and dimensions of the wall and bricks to be built by the UGV in the second challenge of the MBZIRC 2020. Right: Photo from an actual run of the MBZIRC2020 trial illustrating the brick piles.

3.1. Challenge 2: Constructing a structure using bricks

In the second challenge of the MBZIRC 2020 competition, a team of UAVs and a UGV were required to work autonomously together in an outdoor environment to locate, pick, transport, and assemble a set of brick-shaped objects and to construct a pre-defined structure composed of multiple walls. Two separate walls were expected to be built by the robots, an L shaped wall to be assembled completely by the UGV and a wall with U shaped channels to be completed by the UAVs.

Four types of bricks with equal cross-section of ($0.2\,\text{m} \times 0.2\,\text{m}$) but different lengths and weights were available in the arena. Figure 1 illustrates a diagram of the bricks and the wall that was to be built by the UGV. A ferromagnetic plate ($\sim$0.3 m long and 0.6 mm thick) was attached on top center of all bricks allowing robots to grab the bricks using a magnetic gripper. The color of the ferromagnetic plates were white and different from the base color of the brick surfaces.

The arena size for the challenge was $50\,\text{m} \times 60\,\text{m}$. Initially bricks were stacked according to their colors in different piles. The robots were required to locate, approach and pick the bricks from the piles and to deliver and place them on top of the wall base to construct a multi-layer wall. Each layer comprised a randomly generated pattern of differently-sized bricks and was communicated to the robots at the beginning of the trial. The location of the wall base was unknown and the robots needed to locate it during the trial. The maximum duration of a trial was 30 minutes.

3.2. Challenge 3: Indoors fire detection and extinguishing

The third challenge of the MBZIRC 2020 competition aimed at developing solutions for extinguishing fires inside high-rise buildings. It required a team of up to 3 UAVs and a UGV to work together to locate, reach and extinguish a set of fires inside and outside a building. Some of the fires could only be reached by the UAVs, such as fires on the higher floors, while the UGV was expected to explore and extinguish fires on the ground floor.

The Challenge 3 arena consisted of an area of $50\,\text{m} \times 60\,\text{m}$ and contained a building with height of 20 meters, as show in Figure 2. Fires were simulated at various locations at ground level in the arena (indoor and outdoor), and at different heights of the building. The maximum duration of a trial was 20 minutes. The robots were deployed from a starting location. The UGV task was to enter the building's ground floor through an open door while the UAVs had to enter higher floors through open windows. The facade fires and the fires inside the building were to be extinguished by pouring water on to the fire, while the two outdoor ground fires were to be covered with fire blankets. A score was computed based on the number of fires extinguished and the total amount of water delivered to all fires.

Figure 2. Pictures from the Challenge 3 arena that consisted of both indoor and outdoor areas.

Figure 3. Left: Indoor area of the Challenge 3 arena. Right: Simulated indoor fire containing a heated plate and a moving red fabric, and a glass container to collect and measure the water deposited by the robots.

Although it was not defined beforehand, the set of potential fire locations were communicated at the time of the competition. For the UGV, a fire was randomly switched on at two possible high-risk locations on the ground floor. This fire was simulated by a moving red fabric and a heated plate ($6\,\text{cm} \times 3.6\,\text{cm}$), as shown in Figure 3.

4. Multipurpose Mobile Manipulator

To address the MBZIRC challenges, we developed a custom mobile manipulator by refurbishing an existing, four-wheeled, differential-drive, mobile base (iRobot ATRV-Jr) and integrating it with a six degrees of freedom robotic arm (Universal robotics UR5e) and a suite of other sensors and actuators.

The ATRV-Jr mobile base was used in the past in the Portuguese RESCUE project (Lima et al., 2003), to foster research on multi-agent robotic systems for SAR operations, and later in the INFRANET project (Vale & Gomes-Mota, 2007) to test robotic solutions for power line inspection. The mobile base has 55 cm of height, 78 cm of length, is equipped with four wheels connected to 2 high torque DC servo motors, weights about 50 kg, and can transport a 25 kg payload.

The ATRV-Jr robot was fully upgraded to be used for the challenges. Besides replacing many of the old power circuits and control boards, we fitted the UR5e control boards inside the robot, along with an Intel NUC7i7BNH as the new on-board computer. The UR5e arm was assembled at the top front of the robot with all the control circuits fitted inside the mobile base. For this purpose the base structure of the robot was reinforced to support the arm.

Other sensors and actuators were added to the robot for the intended applications, as illustrated in Figure 4. A Pixhawk 2.1 IMU with a Here+ GPS antenna positioned at the top of the robot was

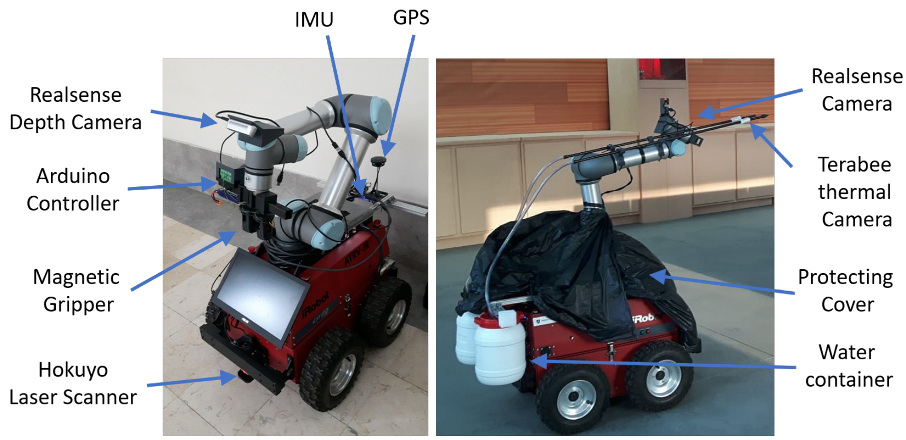


Figure 4. Final hardware configuration used at the MBZIRC competition indicating the hardware components. Left: Setup for Challenge 2. Right: Setup for Challenge 3.

installed for the purpose of robot localization and navigation. A Hokuyo UTM-30LX laser scanner was also installed at the front of the robot, which provides two dimensional laser scans from 0.1 m to 30 m within a field of view of 180°. Additional sensors and actuators such as a RealSense D435, a Terabee thermal sensor and a magnetic gripper were also added to the robot. The Intel RealSense D435 camera that incorporates a full-HD, RGB-plus-depth sensor was installed on the manipulator and near the end-effector, allowing the environment to be actively perceived through the movement of the manipulator.

For the magnetic gripper, required for Challenge 2, we initially tested a commercial magnetic gripper (Zimmer HEM1080) and found it too weak in holding thin metal plates. Hence, a custom-made gripper was used instead, which consisted of Magswitch magnets and servo motors to rotate the magnets to pick and release objects. The gripper was controlled by a simple Arduino and was designed to accommodate a variable number of magnets according to the required capabilities.

Specifically for Challenge 3, we installed two 6-liter water containers at the back of the robot. Each container carried a pump to eject water through a hose installed on the UR5e arm. The pumps were activated and deactivated by an Arduino control board. Additionally, a thermal camera was installed between the two hoses to detect fires based on temperature. The whole robot system can be powered either using two LifePO4 24 V 20 Ah batteries or using two Pb 12 V 35 Ah batteries, providing an autonomy of more than 5 hours of continuous operation.

The Robot Operating System (ROS) was used as a common framework for integrating different software components and interacting with the robot hardware. The UGV operates by autonomously switching between a set of predefined states that are triggered autonomously by sensor inputs and the current status of the mission. A state-machine was built for each of the two challenges using the SMACH library (Bohren & Cousins, 2010) allowing the definition of robot states and state transitions and executing the mission plan. Details of the software components and functionalities for each of the two challenges are provided in the following subsections.

5. Challenge 2: Autonomous construction

The proposed method for solving Challenge 2 is summarized as follows: Once the input wall pattern is communicated to the team by the operator, the UAVs take off and search for the brick piles and the wall base. Upon identifying these targets, the mobile manipulator is then guided to the vicinity of the desired pile of bricks specified by the input wall pattern. This is achieved through a GPS-based localization system and a waypoint navigation algorithm (described in Section 5.1)

Algorithm 1. Challenge 2 routine

```
Result: Wall built
input: brick_pattern, brick_pile_loc, wall_loc
begin
    S = generate_brick_sequence(brick_pattern);
    forall bricks b ∈ S do
        move_to(brick_pile_loc(b));
        align_to_pile(b);
        detect_brick(b);
        pick_brick(b);
        move_to(wall_loc);
        align_to_wall;
        if b is the first brick of wall then
            detect_wall_base;
            place_brick(b);
        else
            detect_reference_brick(b);
            place_brick(b);
        end
    end
end
```

that includes obstacle detection and avoidance based on a laser scanner sensor. Once the robot is in the vicinity of a desired pile, a laser-based brick detection and alignment method is then employed (described in Section 5.2) allowing the robot to move precisely towards the center of the desired brick. A vision based algorithm (described in Section 5.3) that uses the color-depth camera attached near the end-effector of the robotic manipulator then confirms the desired brick color and estimates the brick pose (position and orientation) as well as the center point of the brick. The robotic manipulator is then activated to make the magnetic gripper approach the center point of the brick, terminating with force feedback when the brick is reached. Once the brick is attached to the gripper, it is picked up by the manipulator. The waypoint navigation algorithm is then employed to drive the mobile robot to the vicinity of the wall. Upon arrival, the laser rangefinder and the vision-based detection methods are employed once again to precisely adjust the robot and its end-effector and to position the brick on the desired location. If available, an already placed adjacent brick on the wall is used as the reference to position the new brick precisely. The brick is finally released by deactivating the magnetic gripper and the robot moves towards the next brick in the pile.

The sequence of expected events is described in Algorithm 1. Details about each state and the corresponding actions are provided in the following sections.

5.1. Outdoor robot localization and waypoint navigation

A main requirement for a robot competing in both challenges is to be able to navigate reliably in outdoor environments. To obtain an accurate robot navigation functionality, two main system components were implemented:

- A robot localization system that estimates the position and orientation of the mobile base with respect to the arena frame.
- A waypoint path planner, which computes feasible paths for the robot to reach specific points in the environment.

The localization system (shown in Figure 5) uses the Odometry information from the wheel encoders of the robot, the GPS and the IMU sensors to estimate the accurate pose of the robot. This

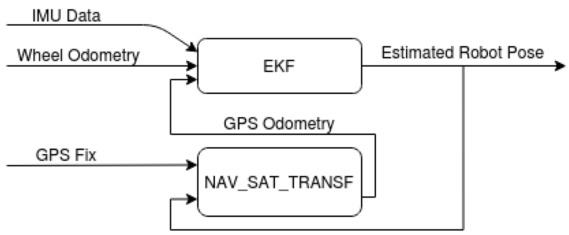


Figure 5. Diagram illustrating the EKF-based sensor fusion technique. The filter is composed of two nodes. The EKF node uses Odometry and IMU data and fuses them together based on a predefined weight. The NAV-SAT-TRANSF node transforms the GPS message into a message consistent with the robot's world frame.

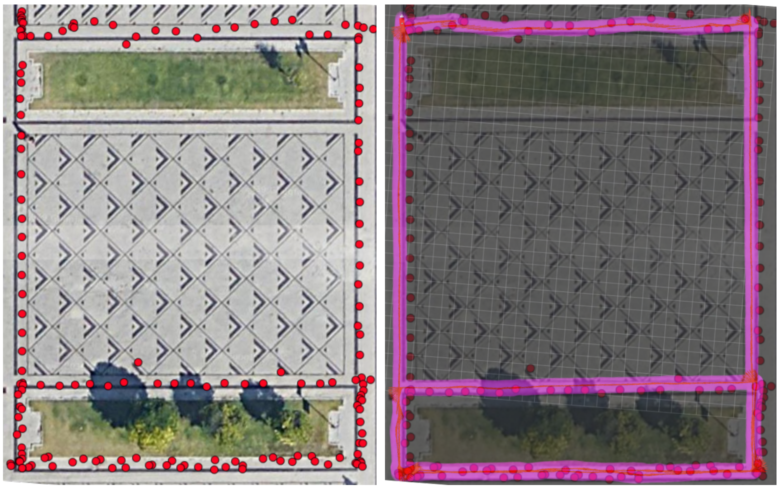


Figure 6. An outdoor localization experiment. Left: Google Maps view of raw GPS coordinates. Right: EKF-based estimation (red line) using Odometry, IMU, GPS, and the estimation covariance (pink).

is achieved using an Extended Kalman Filter (EKF) (Moore & Stouch, 2015) that fuses the input information together and outputs an estimate of the pose.

Inputs are associated with covariances, enabling the filter to weight the information from the sensors according to its expected accuracy. The localization system allows the robot to estimate its pose at every time instance with respect to a fixed reference frame and to compute its pose relative to other fixed frames in the arena (for example the starting pose, the wall and the piles of bricks). Figure 6 shows the localization results from an experiment performed at the Instituto Superior Técnico where the robot was driven on a path of several hundreds of meters consisting of visual markers used as the ground truth.

For the waypoint navigation, a global planner and a local planner work together to calculate the best path from the robot's current location to the desired waypoint in the arena (Marder-Eppstein et al., 2010). For this purpose a costmap[3] representing all obstacles and an occupancy grid representing the arena are considered. Obstacles are represented in the costmap as occupied cells. In their vicinity, cells are assigned a value representing the probability of that cell also being occupied (as shown in Figure 7). This probability decreases with the distance to the obstacle allowing the planner to compute a smooth path while avoiding collisions between the occupied cells and the cells inside the robot's footprint.

[3] http://wiki.ros.org/navigation

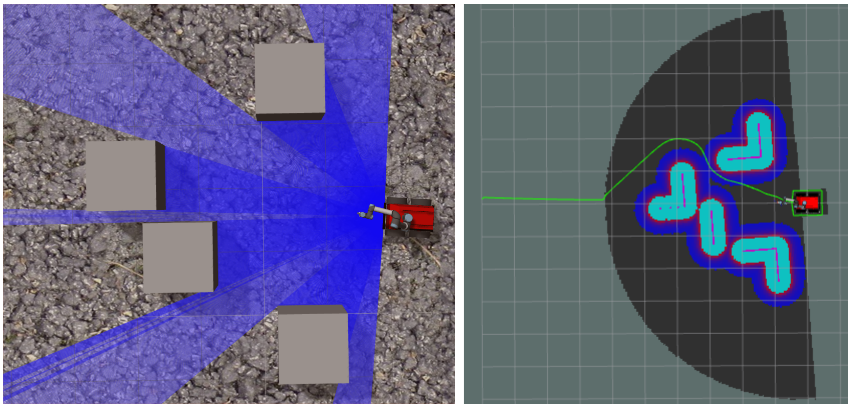


Figure 7. Left: An example of the path planning solution in Gazebo simulation; Right: The corresponding costmap containing both the known (darker) and the unknown spaces (lighter) of the environment and the occupation probability of the cells are used by the planner to compute the path of the robot (right).

For the Global Planner, which is based on the Dijkstra's Algorithm (Dijkstra et al., 1959), a modified version of the `Navfn` planner[4] was used for computing the global path. However, the planner is modified to allow creation of plans that traverse unknown spaces in the global costmap. This is because the map of the outdoor environment is not available in advance and its not possible to produce a map before the mission as the layout of the arena is expected to be frequently changing. The known space is only considered as the area that is perceived by the laser scanner.

A Dynamic Window Approach (DWA) Local Planner (Fox et al., 1997) is used along with a local costmap of $10\,\text{m} \times 10\,\text{m}$ to compute circular arc motions (expressed by their linear and angular velocity combinations) towards the local goal pose, taking into consideration the kinematic characteristics of the robot and the obstacles in the scene. By scoring each of the options, the best motion for the robot at each step is then selected. This is regularly repeated throughout the global path provided by the Global Planner until the final goal is reached.

5.2. Laser-based alignment

The robot localization and navigation algorithm described before allows the robot to move autonomously and safely in the arena and to reach waypoints of interest such as near the wall or the pile of bricks. However, for approaching objects to interact with them, a more precise navigation method is required. To address this situation, a relative localization and navigation system based only on the measurements from the laser scanner is employed to position the robot precisely close to the pile of bricks or the wall, hence allowing the robotic arm to reach, grasp and manipulate bricks.

This algorithm starts by detecting and segmenting lines (presumed to correspond to edges) in the 2D point cloud obtained from the laser scanner. The method presented in Pfister et al. (2003) was employed to extract a list of all lines visible in the point cloud. Algorithm 2 describes the brick alignment procedure. Once a line that corresponds to a desired brick (or the wall in the case of placing a brick) is detected, the relative angle and distance of the brick is computed and is used to guide the robot to reach within a given distance (30 cm in our examples) of the brick's center. This computation is updated with a frequency of 10 Hz whenever a new laser scan is available. Once the robot reaches the desired brick, it rotates to face the brick. Figure 8 illustrates an example of the brick aligning procedure, where the robot approaches and positions itself within manipulation range from the brick.

[4] http://www.ros.org/wiki/navfn

Algorithm 2. Brick Alignment

```
Result: Robot aligned to brick
input : line_segments, brick_length
begin
    for Segment s in line_segments do
        if distance(robot, s) < 2 m and length(s) == brick_length then
            compute_angle(robot, s);
            compute_trajectory;
            while distance(robot, s) > 0.3 m do
                move_forward;
            end
            rotate_to_align;
        else
            continue;
        end
    end
end
```

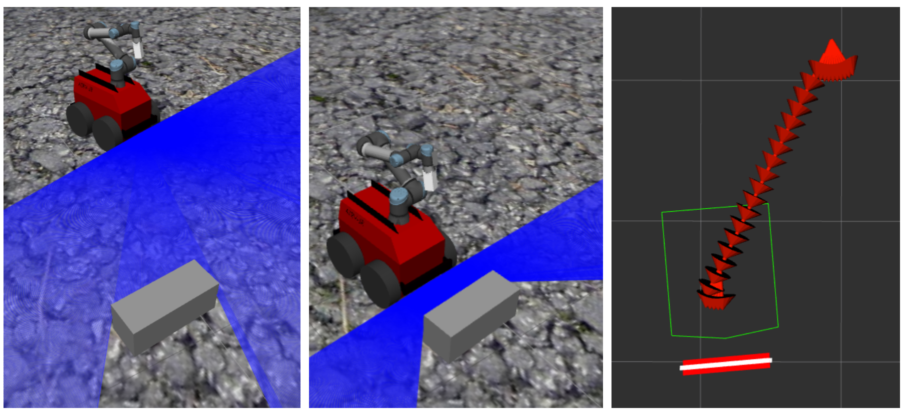


Figure 8. Left: robot starting position; Center: robot final position (aligned with the brick surface); Right: the path calculated to perform the laser-based alignment.

5.3. Brick picking and placement

For picking and placing bricks, we developed a vision-based algorithm that allowed the robot to precisely locate the top surface of the bricks and to guide the robotic manipulator to pick the brick from the center. Furthermore, the vision-based brick detection also allows the robot to notice bricks stacked atop others, a situation undetectable by the laser scanner. The RealSense D345 depth camera near the end-effector of the robotic manipulator was used for this purpose.

Algorithm 3 describes the picking procedure. Once the mobile base is in place and ready to pick a brick, the arm is initially extended out to a predefined position and orientation to have the camera above the brick and parallel to the ground (as shown by the camera image illustrated in Figure 9). The top-view image of the camera is then filtered using a depth segmentation to remove the ground planes or points below the surface of the brick. Afterwards, white color segmentation and rectangle fitting is performed to detect the brick's steel plate.

Once the metallic plate of the brick is identified, the center and the tilt angle of the fitted rectangle is computed. In addition, the average depth of a set of pixels around the center point of the rectangle is used to compute the relative distance between the camera and the brick. This information is used

Algorithm 3. Brick picking

```
Result: Brick attached to robot arm gripper
input : brick_color, point_cloud
begin
    move_arm(top_view);
    magnetize_gripper;
    while point_cloud is empty do
        acquire_new_image;
        remove_ground(point_cloud);
        segment_color(point_cloud, brick_color);
    end
    rectangle=fit_rectangle(point_cloud);
    pixel_center=find_center(rectangle);
    depth_center=extract_depth(pixel_center,point_cloud);
    desired_pose=compute_pose(robot, pixel_center, depth_center);
    arm_pose=compute_arm_ikinematics(desired_pose);
    move_arm(arm_pose);
    while brick not attached do
        lower_end_effector(−0.2 cm);
        sleep(0.1 s);
    end
    move_arm(carry_position)
end
```

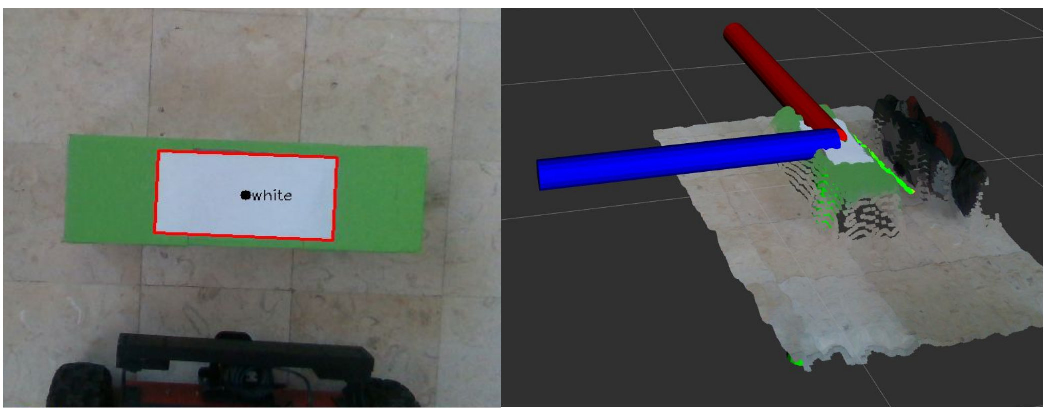


Figure 9. Illustration of the vision based algorithm used for picking a brick. Left: A top-view image of the brick is used to perform white-color segmentation and to fit a rectangle to the brick's metallic surface. Right: The center pose with axis aligned to the brick is computed and communicated to the manipulator planer. (x,y) axis indicated by (red, blue) respectively.

along with the camera's intrinsic parameters to obtain the 3D coordinate of the center point and the orientation of the brick with respect to the robot. The manipulation planner then computes the inverse kinematic solution as well as the trajectory to have the gripper positioned about 5 cm above the brick's center point. The gripper is then gradually lowered until it grasps the brick as detected by the manipulator's force-feedback sensors. The motion planner previously reported in (Gonçalves & Lima, 2019) was used for computing an efficient inverse kinematics solution and to plan a collision free trajectory for the arm.

Algorithm 4. Brick placing

```
Result: Brick placed in the correct position
input : brick_attached, reference_brick, point_cloud
begin
    move_arm(top_view);
    while point_cloud is empty do
        new_image;
        remove_ground(point_cloud);
        segment_color(point_cloud, reference_brick_color);
    end
    rectangle=fit_rectangle(point_cloud);
    edges=compute_rectangle_edges(rectangle);
    pixel_edge_point=select_point(edges, point_cloud);
    depth_edge_point=extract_depth(pixel_edge_point);
    desired_pose=compute_pose(robot, pixel_edge_point, depth_edge_point);
    arm_pose=compute_arm_ikinematics(desired_pose);
    move_arm(arm_pose);
    while force ≥ (initial_force × 0.8) do
        lower_end_effector(−0.2 cm);
        sleep(0.1 s);
    end
    demagnetize_gripper;
    move_arm(default_position)
end
```

The brick placement algorithm (Algorithm 4) follows a similar principle as the brick picking. However, the algorithm also takes into account a priori information on the current wall status and selects a brick on the wall as the reference brick to place the new brick with respect to this reference, i.e. to place it on the right, left or above the reference brick depending on the input wall pattern.

After picking a brick, the robot initially navigates to the wall location, using the waypoint navigation method described in Section 5.1, and then approaches the desired region of the wall containing the reference brick and positions itself using the method described in Section 5.2. Once in position, the robot extends the arm out to have the camera above the reference brick similar to the picking algorithm described in Section 5.2. Note that the position and viewing angle of the camera is selected in such a way that the picked brick does not interrupt the camera's field of view.

Once a top-view image is acquired, the reference brick is then identified in the image using color segmentation and a rectangle is fitted to the top surface of the brick, as shown in Figure 10. The rectangle parameters are then used to compute the edge coordinates of the reference brick, allowing the picked brick to be placed relative to this reference edge. For this purpose, a set of points near the edge of this brick is selected and averaged to compute a reliable depth measurement for the edge. This tactic avoids noisy depth measurements that often appear near the object edges. The edge coordinate in the camera frame is then transformed to the robot's frame and is used to compute the placement coordinate for the picked brick. Finally, the manipulator moves the brick above the desired location and gradually lowers the arm until the manipulator's force-feedback indicates that the brick is in place. Figure 10 illustrates an experiment where the target position is computed based on the top-view image and the pose provided to the manipulator planner.

6. Challenge 3: Firefighting

The proposed method for solving Challenge 3 is summarized as follows: Once the start signal is received, the robot autonomously navigates to the entrance of the building using the outdoor

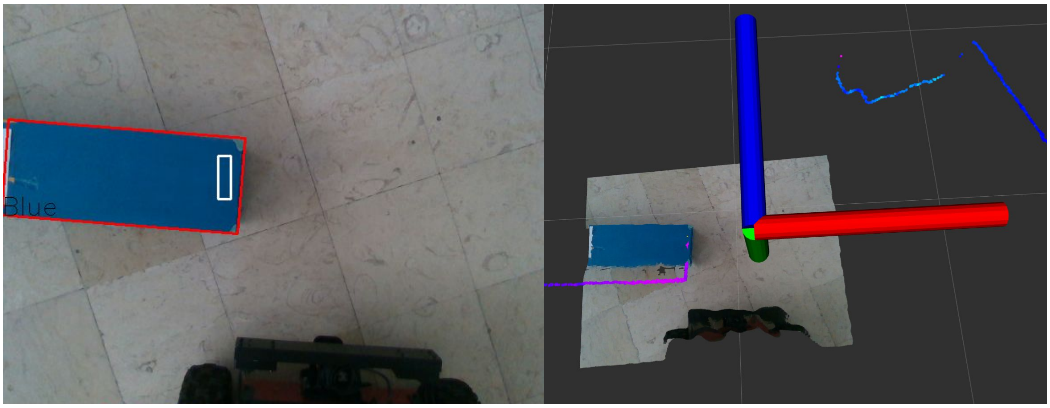


Figure 10. Illustration of the vision based algorithm used for placing a picked brick. Left: Color segmentation with rectangle fitting and averaging samples at the edge is used to compute a reliable position for the edge of the reference brick. Right: A placement pose is computed relative to the reference edge that is communicated to the manipulator planer. (x,y,z) axis indicated by red, blue, green respectively.

localization system and waypoint navigation algorithm (described in Section 5.1). The robot then switches to an indoor localization and navigation system (described in Section 6.1) that relies on the laser scanner and is suitable for entering and navigating inside the building where GPS signal is not available. The robot then searches for fires using its visual and thermal sensors and extinguishes them by pointing two water tubes towards the fire and pouring water directly on to the fire (described in Section 6.2).

The states of the robot for this challenge is divided into five main states:

1. Waypoint Navigation;
2. Switch Localization;
3. Fire Detection;
4. Laser and Thermal image based alignment;
5. Shoot Water.

Algorithm 5 describes how the firefighting procedure is executed, where the robot starts from a known starting position, approaches and enters a building to check two known high-risk locations for possible fires.

6.1. Indoor localization and navigation

For Challenge 3, the robot was required to enter and navigate inside a building to locate and extinguish fires (Figure 2). Since GPS signals were not available indoors, a different localization strategy was employed. For this purpose, Adaptive Monte Carlo Localization (AMCL) (Dellaert et al., 1999) was used that took into account the laser scans, wheel odometry measurements, and a previously constructed map of the environment to localize the robot.

Since different localization approaches are used for the outdoor and indoor environments, the navigation system must be capable of switching between localization strategies based on the robot's current status. The way the system dealt with this was by having a switch node that received both the outdoor localization, described in Section 5.1, and the indoor AMCL localization and to select one of them given the current robot status. Upon a transition, the last reliable pose estimate from the previous method is used as the initial estimate for the new method. This switch node also automatically updates the transforms between the map and world (robot start position) frames with

Algorithm 5. Challenge 3 routine

```
   Result: Fire detected and put out by the mobile UGV
   input : map, waypoint_list[], initial_position
1  begin
2      switch_localization('outdoors', None);
3      move_to(waypoint_list["Door"]);
4      switch_localization('indoors', initial_position);
5      move_to(waypoint_list["Check Fire 1"]);
6      if detect_fire(timeout = 5) is false then
7          move_to(waypoint_list["Check Fire 2"]);
8          if detect_fire(timeout = 5) is false then
9              Go back to line 5;
10         end
11     end
12     align();
13     shoot_water();
14     move_to(waypoint_list["Door"]);
15     switch_localization('outdoors', None);
16     move_to(waypoint_list["Origin"]);
17 end
```

a fixed transformation while in outdoors mode, or allows the AMCL node to continuously calculate this transformation while in indoors mode.

A scheme of this localization switch node is shown in Figure 11. Three reference frames are used: the origin of the map, the start position of the robot (world) and the estimated position of the robot (base). The outdoor localization system updates the reference of the base in relation to the world, while the relation between the world and the map remains unchanged. The fixed transformation used for the outdoors navigation represents the transformation between the starting position of the robot and the position where the mapping began. An easy way to determine this transformation is by starting the mapping process at the start position, making it a null transform. For the indoors localization mode, the transformation between the map and world is calculated by the AMCL while the transformation between world and the base is estimated from the accumulated odometry. As seen in the scheme, the transition between algorithms is operated in the *localization_switch* node and controlled by a trigger message sent when a change in localization is required.

6.2. Fire detection and extinguishing

The fire detection was achieved with a TinyYOLO neural network (Redmon & Farhadi, 2017). The training dataset was composed of RGB images of real fires and images of simulated fires (using a Mini Silkflame device). The dataset consisted of 700 RGB images, which amount was below optimal, but since only one class ("fire") was being trained it proved sufficient. Training results were satisfactory, and the robot could even track fires while moving. After detecting a fire, the image depth is computed by integrating information from the detected location in the RGB image with the depth image obtained from the RealSense camera, similarly to the algorithm used in the brick center calculation. Additionally, a Terabee thermal camera was used to detect fires based on the radiated heat. However, the camera has a low resolution (36 pixels $\times$ 36 pixels) and does not provide depth information. An image of both the RGB image and the thermal image obtained from the Terabee thermal camera is illustrated in Figure 12.

The algorithm to search for fires inside the building can consider two cases of A) search in known, high-risk locations such as the kitchen stove or oven, or B) to search the entire floor. We considered

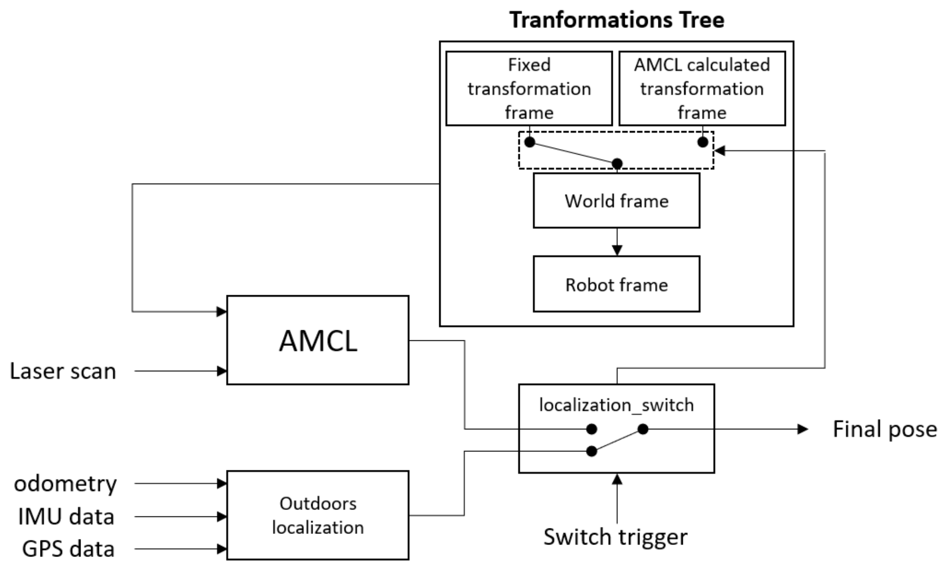


Figure 11. Scheme of the localization switch for switching between indoor and outdoor localization.

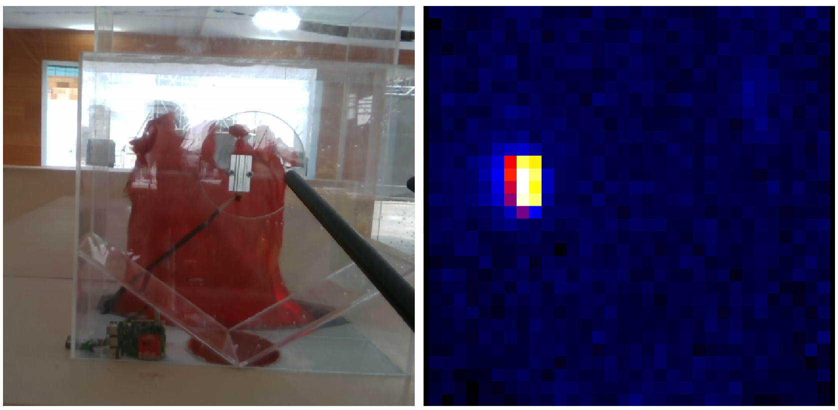


Figure 12. An instance from a competition trials showing RGB and thermal images. (Left: RGB image, Right: Thermal image), where the blue pixels indicate a temperature of around 25 Celsius degrees (room temperature) and the white/yellow pixels have a temperature of about 90 Celsius degrees corresponding to the heating plate.

both possibilities, since no information about the fire locations was available before the competition. Case A was used at the competition.

Algorithm 6 describes the fire detection algorithm. When high-risk locations are not specified beforehand, the robot drives around the building, visiting a set of distributed waypoints, and checks the entire area for fires. When a positive detection is obtained, the fire location is estimated and used to calculate the position and pose the robot should reach to achieve appropriate distance to the fire and shoot water. An example of this process is illustrated in Figure 13.

When given a set of high-risk locations, the system defines a fixed waypoint directly in front of each candidate location. The robot then navigates to each waypoint and checks for active fires using both image and thermal detection. Figure 14 shows a picture from the competition where the robot is checking one of the two existing high-risk locations. To shoot water, the robot first uses its laser

Algorithm 6. Fire detection algorithm

```
Result: Fire detection and robot in a shooting position.
input : rgb_image, thermal_image, waypoint_list[], possible_fire_locations[]
begin
    if possible_fire_locations[] is empty then
        for waypoint in waypoint_list[] do
            move_to(waypoint);
            fire_location = detect_fire_yolo(rgb_image);
            if fire_location is not None then
                pose_close_to_fire = calculate_pose(fire_location);
                move_to(pose_close_to_fire);
                adjust_pose();
                return Successful_Fire_Detection;
            end
        end
    else
        for location in possible_fire_locations[] do
            move_to(location);
            if detect_fire(thermal_image) is true then
                adjust_pose();
                return Successful_Fire_Detection;
            end
        end
    end
    return Failed_Fire_Detection;
end
```

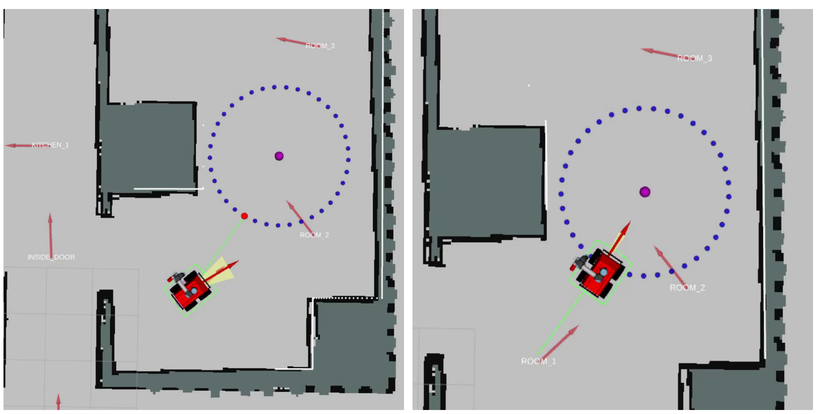


Figure 13. Robot approaching a detected fire in simulation. Left: Robot choosing the best pose to shoot water from. The purple sphere shows the location of the detected fire, the blue dots show the possible positions that the robot can select from to have the desired distance to the fire, red dot shows the pose selected by the robot. Right: Robot reaches the chosen pose.

measurements to move to the right distance to the burning structure. Once in position, the thermal camera located between the two tubes is then used to perfectly align the heading of the robot to have the water tubes directly pointing to the fire. For this purpose, a simple closed loop controller is implemented to modify the heading of the robot until the pixel with the highest temperature is in the center of the thermal image.

Figure 14. Robot detecting a fire in the arena built for MBZIRC 2020 Challenge 3.

Finally, once the robot is at the correct distance and is aligned to the fire, the arm is moved in to a calibrated shooting position. The robot then starts shooting water for a predetermined time, while slightly moving the arm up and down and rotating the robot base right and left to increase coverage and account for alignment errors and deviations.

7. System evaluation

Extensive experiments performed in both simulation and on the real robot demonstrated the success of the control algorithms and the overall system in executing both challenges. Highlights from some of the experiments and the competition runs are compiled in a video[5].

7.1. Challenge 2

A simulated world was initially developed in Gazebo (Koenig & Howard, 2004) to simulate the robot and the scenario. Models of the four types of bricks were added to the world along with a L-shaped frame where the wall should be built on. Since Gazebo uses a physics engine it provides realistic rendering of the robot and the environment including gravity, inertial properties, collisions, lights, shadows and textures with a good approximation.

Although simulations cannot completely capture all the details of the real world, they contributed significantly in developing and testing our algorithms, especially regarding the design and testing of the state machines and execution of the mission routines. Figure 15 shows an example run from the simulator where the robot builds a 2 layer wall composed of 6 bricks. The time to complete this scenario was measured through 5 runs and averaged 9 minutes and 12 seconds.

We also tested our designs with the physical robot, which experiments demonstrated the success of the methods in accomplishing the challenge. For the real experiments, we used multiple bricks with dimensions and weights as to the ones used in the actual competition. The robot was then instructed to complete the full pick and place cycle described without any human intervention. This cycle included: navigating to an area containing a brick, detecting the brick to be picked, aligning with the brick, grasping and delivering the brick to the wall location, and correctly placing the brick with respect to a reference brick on the wall. Figure 16 illustrate a run of the experiment where the robot continuously performs the full pick and place operation.

[5] https://www.youtube.com/watch?v=7Vq-pK2xvEw

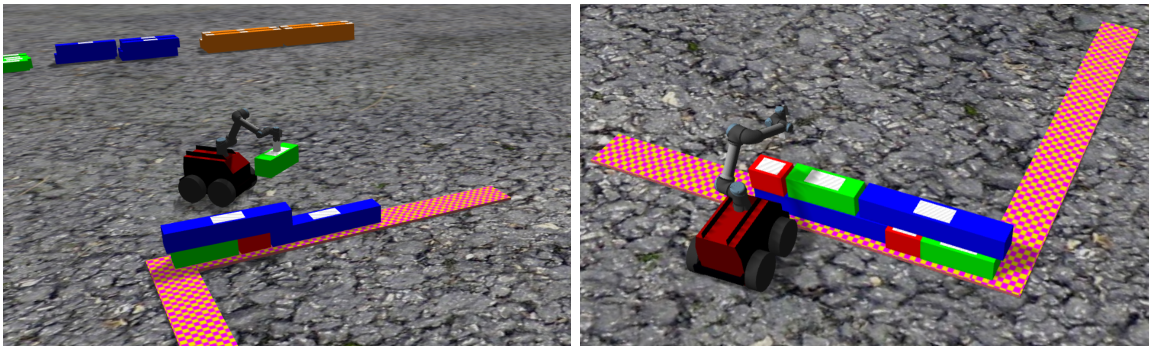


Figure 15. Robot building a 6-brick wall in simulation.

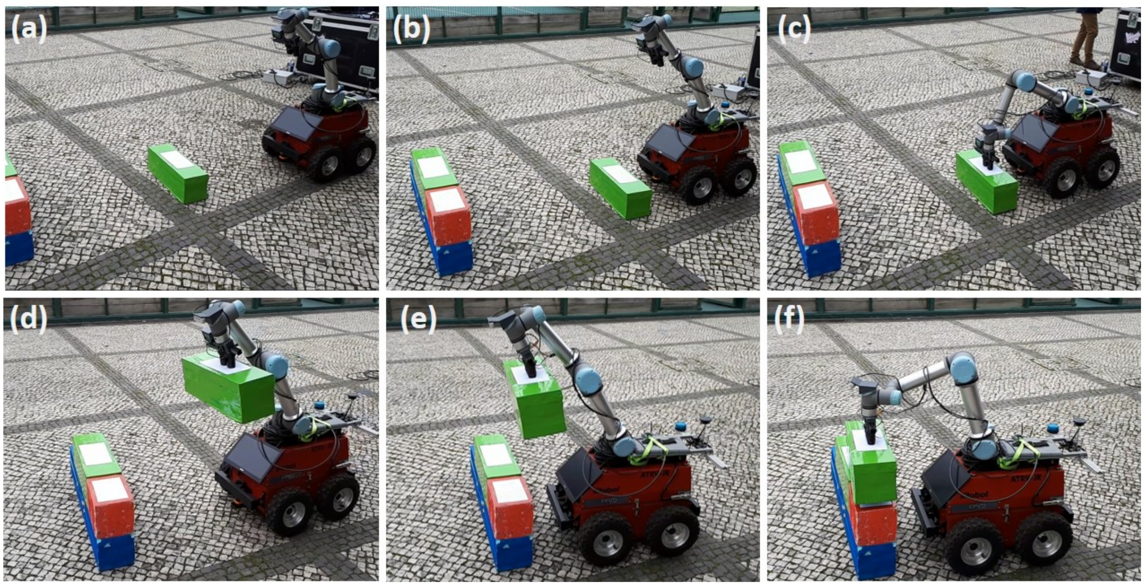
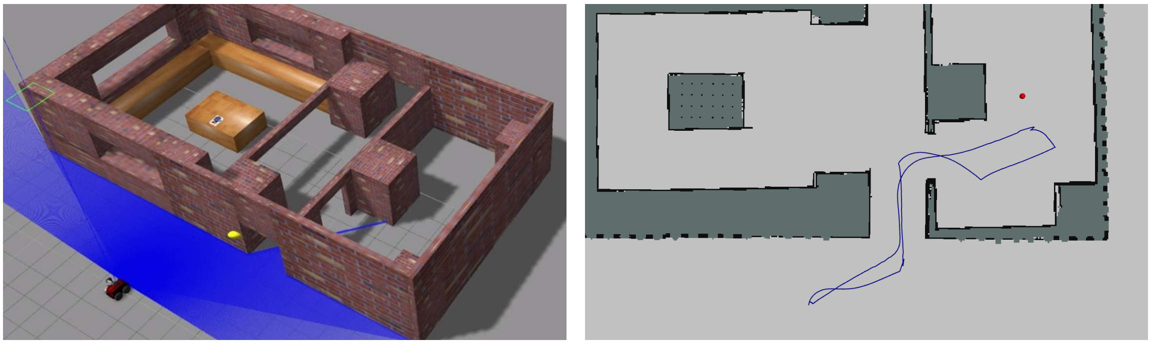


Figure 16. Snapshots from an outdoor experiment where the full pick and place cycle is performed autonomously. (a) The robot searching for a brick with laser scanning (b) Robot detects the brick and aligns itself to the brick (c) The brick is visually detected and grasped by the manipulator from the center (d) The robot carries the brick to the wall location (e) Robot aligns itself with the wall using feedback from the laser scanner and uses the camera at the manipulator to localize the red reference brick on the wall (f) The robot places the brick on the third layer while using the edge of the reference brick to calculate the placement position.

7.2. Challenge 3

Similarly, to help design and evaluate our methods at both the module and system levels, we conducted simulations in a Gazebo world. Figure 17a shows the world created in Gazebo that simulates the building of the competition containing the fires. In particular, simulations were used to test and tune the localization and navigation algorithms, such as the outdoor navigation, indoor AMCL and the transition between the two methods. The robot was capable of successfully entering and navigating through the building corridors and doors as shown in Figure 17b, where the robot's path to achieve its goal is depicted as well as the fire location.



(a) (b)

Figure 17. Simulation environment for Challenge 3. (a) The simulated environment. (b) The path of the robot (blue line) during the execution of Challenge 3 and location of the fire that was detected as a red circle.

(a) (b) (c) (d) (e) (f)

Figure 18. Snapshots of an experiment consisting of indoor navigation and detection of fire at the ISR test-bed. (a) The robot starts its navigation (b) The robot reaches the waypoint in front of the door (c) The robot goes to the first waypoint inside the room and performs a fire detection, that fails since there are no fires in its field of view (d) The robot moves to the second waypoint inside the room and detects the fire (e) The robot computes and moves to a pose relative to the fire suitable for depositing water (f) The robot moves outside the test room.

For our physical experiments, we used the ISRobo- Net@Home testbed[6] and a Silkflame fire simulator to perform the challenge tasks and to verify the the complete system, including the state machine, the localization transition, the indoor localization and navigation, and the fire detection and localization method. Figure 18 shows the steps of one of the trials of the real-world experiments. In these experiments, the robot always started some distance away from the testbed and was able to correctly navigate to a waypoint near the entrance of the testbed using odometry (as GPS signals were not available inside the facility). Once at the waypoint, the robot smoothly changed localization method, moved inside the apartment and searched for the fire simulator by visiting several waypoints and continuously looking for the fire. When a fire was detected the robot calculated a position to shoot water and moved to it. Since it was not possible to shoot water inside our testbed, the robot simply waited for five seconds and published a message to inform that it was in shooting mode. Then the robot exited the room and terminated the trial.

[6] http://welcome.isr.tecnico.ulisboa.pt/isrobonet/

8. Competition results and lessons learned

8.1. Challenge 2

Despite the many successful attempts in executing the full pick and place cycle before the competition and while backstage at the competition, multiple unexpected challenges prevented the system from performing as well as expected when inside the competition arena. Almost all of these challenges were related to the introduction of an unfamiliar environment and the limited number of trials available for teams to adapt their solutions to this new environment.

The first problem we observed in the trials was that the GPS readings in the arena proved unreliable, potentially due to the tall structures around the arena or other interferences, which resulted in poor performance of the outdoor localization system. Fortunately, the estimation with only the odometry and IMU was good enough and was employed in rest of the trials, by always starting the robot at the same starting location

The second problem encountered was that, opposed to our expectations, the competition arena was not flat but included a large, wooden ramp at the center, connecting two surfaces of different heights. In our trials it was observed that the ramp was detected as an obstacle by the laser sensor and prevented the robot from moving on to the ramp. To tackle this problem, a blind navigation strategy was used, where the robot simply ignored obstacles at the edge of the ramp to move onto the ramp.

The third problem was also related to the arena. The wooden ramp comprised most of the area was neither completely rigid nor flat. Hence it consisted of multiple curves along the way or it deformed when the robot moved over the ramp, causing the laser scanner, mounted close to the ground, to detect regions of the ramp as obstacles. This situation resulted in the robot getting stuck or continuously steering to avoid obstacles that did not exist. To deal with this problem the costmap parameters were tuned to only consider obstacles within 3 meters of the robot for computing a smooth and efficient path.

In the last available trial, the robot navigation behaved as expected and the robot always managed to reach the piles of bricks and to pick up the desired, orange brick, the largest and heaviest. However, in this trial the robot always failed to pick at the brick's center and hence dropped it while moving up the ramp. Logs showed that, due to the time of the trial and the configuration of the orange pile, there was a clear shadow line exactly in the middle of the white plate and hence the color segmentation failed to correctly determine the center point of the plate from the very high contrast image of the plate.

Challenge 2 required a wide variety of complex tasks to be performed together. Despite the system being fully functional before the event, we failed to achieve the same performance in this new environment. However, positive progress among the few available trials was clearly observed and better results would have been obtained if more trials were available.

8.2. Challenge 3

The performance of the system in Challenge 3 was much better and contributed to the winning of the team in this challenge. However, we also encountered multiple, unexpected problems during the practice trials, again mainly due to the unfamiliarity of the teams with the new environment and the competition setup.

During the competition it was quickly observed that the developed vision-based fire detection could not be utilized. This was because the indoor fires were not emulated with the Silkflame device described in the rulebook. Each competition fire was simply a moving red fabric and did not resemble a real fire, hence they could not be detected with our TinyYOLO fire detector. An approach to solve the problem was to train the network again with new images, but this was time consuming considering the short amount of time that was available. Hence, we could only use the thermal detection solution with the Terabee camera, which method did reliably detect the heated plates. However, there was an instance of 1 false positive which resulted in the water being shot at the wrong target. We believe this was simply due to the heating element of the inactive fire still being hot from the last trial as there was a very tight schedule in place. If time had allowed, the best solution would have been to employ both thermal and color detection to obtain a more robust solution.

Another problem that was encountered in this challenge was with the indoor navigation system. In one of the attempts, the water splashes on the laser sensor, along with the dusty competition environment, resulted in poor laser readings. We quickly identified and resolved this. Furthermore, due to the unreliable GPS readings, the outdoor robot navigation to reach the entrance of the building was not consistent and hence caused different initial positions at the entrance for the AMCL localization, which sometimes resulted in an unsuccessful transition between the indoor and the outdoor localization methods. This problem was mitigated by avoiding the use of GPS for outdoor navigation and always starting the robot in the same starting position at the begining of the trial.

9. Conclusion

This paper described a versatile, multi-purpose, mobile, service robot, capable of smooth adaptation to different functions and applications, developed through participation in a scientific robot competition, highlighting once again the role of competitions in contributing to the development of solutions for real-world problems. The paper presents in detail the system description of an autonomous mobile manipulator, including both hardware and software architectures, for solving the second and third challenges of the MBZIRC 2020 robot competition. The paper described the algorithms for detecting, localizing, picking and placing heterogeneous building blocks to form a wall-like structure (Challenge 2), and algorithms for detecting and extinguishing fires inside a building, requiring both indoor and outdoor navigation and smooth transition between the two environments (Challenge 3). Results obtained from the experiments and the competition trials illustrated the success of the system in accomplishing the main goals of the challenges. The team received the first place award for Challenge 3. In addition, the set of problems encountered and the lessons learned from employment of the solutions in the near-realistic conditions of the Competition was described. The versatile and multipurpose system can potentially be advantageous in many different applications such as in search-and-rescue operations, robot-based 3D printing of structures and transporting objects in hazardous environments. The developed mobile manipulator and the solutions presented in this paper are currently being adapted to be employed for the innovative application of UGV-based inspection and maintenance of solar farms under the DURABLE[7] project.

Acknowledgments

This work was supported by a grant from the Mohamed Bin Zayed International Robotics Challenge (MBZIRC 2020) and partially funded by the INTERREG ATLANTIC EAPA_986/2018 DURABLE project and LARSyS - FCT Plurianual funding 2020-2023. The authors would like to thank all members of the Iberian Robotics team including the GRVC group of University of Seville and FADA-CATEC for their help and support.

ORCID

Meysam Basiri https://orcid.org/0000-0002-8456-6284
José Rosa https://orcid.org/0000-0002-8585-9791
Rui Bettencourt https://orcid.org/0000-0002-0917-5232
Alberto Vale https://orcid.org/0000-0003-3423-3905
Pedro Lima https://orcid.org/0000-0002-8962-8050

[7] https://www.durableproject.eu

References

Aydin, B., Selvi, E., Tao, J., & Starek, M. J. (2019). Use of fire-extinguishing balls for a conceptual system of drone-assisted wildfire fighting. *Drones*, *3*(1), 17. https://doi.org/10.3390/drones3010017

Basiri, M., Piazza, E., Matteucci, M., & Lima, P. (2019). Benchmarking functionalities of domestic service robots through scientific competitions. *KI-Künstliche Intelligenz*, *33*(4), 357–367. https://doi.org/10.1007/s13218-019-00619-9

Basiri, M., Schill, F., U.Lima, P., & Floreano, D. (2018). Localization of emergency acoustic sources by micro aerial vehicles. *Journal of Field Robotics*, *35*(2), 187–201. https://doi.org/10.1002/rob.21733

Baums, A. (2017). Response to CBRNE and human-caused accidents by using land and air robots. *Automatic Control and Computer Sciences*, *51*(6), 410–416. https://doi.org/10.3103/s0146411617060025

Ben-Tzvi, P., Goldenberg, A. A., & Zu, J. W. (2007). Implementation of sensors and control paradigm for a hybrid mobile robot manipulator for search and rescue operations. *2007 International Workshop on Robotic and Sensors Environments*, 1–6. https://doi.org/10.1109/rose.2007.4373974

Bohren, J., & Cousins, S. (2010). The SMACH high-level executive [ROS news]. *IEEE Robotics and Automation Magazine*, *17*(4), 18–20. https://doi.org/10.1109/mra.2010.938836

Bouyarmane, K., Chappellet, K., Vaillant, J., & Kheddar, A. (2019). Quadratic programming for multirobot and task-space force control. *IEEE Transactions on Robotics*, *35*(1), 64–77. https://doi.org/10.1109/tro.2018.2876782

Carius, J., Wermelinger, M., Rajasekaran, B., Holtmann, K., & Hutter, M. (2018). Deployment of an autonomous mobile manipulator at MBZIRC. *Journal of Field Robotics*, *35*(8), 1342–1357. https://doi.org/10.1002/rob.21825

Caselli, S., Fantini, E., Monica, F., Occhi, P., & Reggiani, M. (2003). Toward a mobile manipulator service robot for human assistance. *1st Robocare Workshop*.

Cosma, C., Confente, M., Governo, M., & Fiorini, R. (2004). An autonomous robot for indoor light logistics. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, *3*, 3003–3008. https://doi.org/10.1109/iros.2004.1389866

Dellaert, F., Fox, D., Burgard, W., & Thrun, S. (1999). Monte Carlo localization for mobile robots. *1999 IEEE International Conference on Robotics and Automation (ICRA)*, *2*, 1322–1328. https://doi.org/10.1109/robot.1999.772544

Dias, J., Althoefer, K., & Lima, P. U. (2016). Robot competitions: What did we learn? *IEEE Robotics & Automation Magazine*, *23*(1), 16–18. https://doi.org/10.1109/mra.2015.2511678

Dias, J., Lima, P. U., Seneviratne, L., Khatib, O., Tadokoro, S., & Dario, P. (2019). Journal of Field Robotics special issue on MBZIRC 2017 challenges in autonomous field robotics. *Journal of Field Robotics*, *36*(1), 3–5. https://doi.org/10.1002/rob.21851

Dijkstra, E. W. et al. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, *1*(1), 269–271. https://doi.org/10.1007/bf01386390

Dömel, A., Kriegel, S., Kaßecker, M., Brucker, M., Bodenmüller, T., & Suppa, M. (2017). Toward fully autonomous mobile manipulation for industrial environments. *International Journal of Advanced Robotic Systems*, *14*(4), 172988141771858. https://doi.org/10.1177/1729881417718588

Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics Automation Magazine*, *4*(1), 23–33. https://doi.org/10.1109/100.580977

Gawel, A., Blum, H., Pankert, J., Krämer, K., Bartolomei, L., Ercan, S., Farshidian, F., Chli, M., Gramazio, F., Siegwart, R., Hutter, M., & Sandy, T. (2019). A fully-integrated sensing and control system for high-accuracy mobile robotic building construction. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2300–2307. https://doi.org/10.1109/iros40897.2019.8967733

Gonçalves, J., & Lima, P. (2019). Grasp planning with incomplete knowledge about the object to be grasped. *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 1–6. https://doi.org/10.1109/icarsc.2019.8733615

Guzman, R., Navarro, R., Ferre, J., & Moreno, M. (2016). RESCUER: Development of a modular chemical, biological, radiological, and nuclear robot for intervention, sampling, and situation awareness. *Journal of Field Robotics*, *33*(7), 931–945. https://doi.org/10.1002/rob.21588

Helm, V., Ercan, S., Gramazio, F., & Kohler, M. (2012). Mobile robotic fabrication on construction sites: DimRob. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (ICRA)*, 4335–4341. https://doi.org/10.1109/iros.2012.6385617

Helm, V., Willmann, J., Gramazio, F., & Kohler, M. (2014). In-situ robotic fabrication: Advanced digital manufacturing beyond the laboratory. In F. Röhrbein, G. Veiga, & C. Natale (Eds.), *Springer Tracts in*

Advanced Robotics: Vol. 94. Gearing up and accelerating cross-fertilization between academic and industrial robotics research in Europe: (pp. 63–83). Springer. https://doi.org/10.1007/978-3-319-02934-4_4

Hvilshøj, M., & Bøgh, S. (2011). "Little Helper" – an autonomous industrial mobile manipulator concept. *International Journal of Advanced Robotic Systems*, *8*(2), 15. https://doi.org/10.5772/10579

Jimenez-Cano, A. E., Martin, J., Heredia, G., Ollero, A., & Cano, R. (2013). Control of an aerial robot with multi-link arm for assembly tasks. *2013 IEEE International Conference on Robotics and Automation (ICRA)*, 4916–4921. https://doi.org/10.1109/icra.2013.6631279

Kang, S., Cho, C., Lee, J., Ryu, D., Park, C., Sin, K.-C., & Kim, M. (2003). ROBHAZ-DT2: Design and integration of passive double tracked mobile manipulator system for explosive ordnance disposal. *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, *3*, 2624–2629. https://doi.org/10.1109/iros.2003.1249266

Koenig, N., & Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator. *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, *3*, 2149–2154. https://doi.org/10.1109/iros.2004.1389727

Korayem, A. H., Nekoo, S. R., & Korayem, M. H. (2019). Sliding mode control design based on the state-dependent Riccati equation: Theoretical and experimental implementation. *International Journal of Control*, *92*(9), 2136–2149. https://doi.org/10.1080/00207179.2018.1428769

Korayem, M. H., Azimirad, V., Nikoobin, A., & Boroujeni, Z. (2010). Maximum load-carrying capacity of autonomous mobile manipulator in an environment with obstacle considering tip over stability. *The International Journal of Advanced Manufacturing Technology*, *46*(5-8), 811–829.

Krotkov, E., Hackett, D., Jackel, L., Perschbacher, M., Pippine, J., Strauss, J., Pratt, G., & Orlowski, C. (2017). The DARPA Robotics Challenge Finals: Results and perspectives. *Journal of Field Robotics*, *34*(2), 229–240. https://doi.org/10.1007/978-3-319-74666-1_1

Kruijff, G.-J. M., Pirri, F., Gianni, M., Papadakis, P., Pizzoli, M., Sinha, A., Tretyakov, V., Linder, T., Pianese, E., Corrao, S., Priori, F., Febrini, S., & Angeletti, S. (2012). Rescue robots at earthquake-hit Mirandola, Italy: A field report. *2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 1–8. https://doi.org/10.1109/ssrr.2012.6523866

Lima, P., Custódio, L., Ribeiro, I., & SantosVictor, J. (2003). The RESCUE project: Cooperative navigation for rescue robots. *Proceedings do 1st International Workshop on Advances in Service Robotics*, 94–101.

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

Lussi, M., Sandy, T., Dörfler, K., Hack, N., Gramazio, F., Kohler, M., & Buchli, J. (2018). Accurate and adaptive in situ fabrication of an undulated wall using an on-board visual sensing system. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 3532–3539. https://doi.org/10.1109/icra.2018.8460480

Marder-Eppstein, E., Berger, E., Foote, T., Gerkey, B., & Konolige, K. (2010). The office marathon: Robust navigation in an indoor office environment. *2010 IEEE International Conference on Robotics and Automation (ICRA)*, 300–307. https://doi.org/10.1109/robot.2010.5509725

Marques, M. M., Carapau, R. S., Rodrigues, A. V., Lobo, V., Gouveia-Carvalho, J., Antunes, W., Gonçalves, T., Duarte, F., & Verissimo, B. (2017). GammaEx project: A solution for CBRN remote sensing using unmanned aerial vehicles in maritime environments. *OCEANS 2017 - Anchorage*, 1–6.

Matsuno, F., & Tadokoro, S. (2004). Rescue robots and systems in Japan. *2004 IEEE International Conference on Robotics and Biomimetics*, 12–20. https://doi.org/10.1109/robio.2004.1521744

Meng, J., Wang, S., Li, G., Jiang, L., Zhang, X., Liu, C., & Xie, Y. (2021). Iterative-learning error compensation for autonomous parking of mobile manipulator in harsh industrial environment. *Robotics and Computer-Integrated Manufacturing*, *68*, 102077. https://doi.org/10.1016/j.rcim.2020.102077

Mirjan, A., Augugliaro, F., D'Andrea, R., Gramazio, F., & Kohler, M. (2016). Building a bridge with flying robots. In D. Reinhardt, R. Saunders, & J. Burry (Eds.), *Robotic fabrication in architecture, art and design* (pp. 34–47). Springer. https://doi.org/10.1007/978-3-319-26378-6_3

Moore, T., & Stouch, D. (2015, July). A generalized extended Kalman filter implementation for the Robot Operating System. In E. Menegatti, N. Michael, K. Berns, H. Yamaguchi, E. Menegatti, N. Michael, K. Berns, & H. Yamaguchi (Eds.), *Advances in Intelligent Systems and Computing: Vol. 302. Intelligent Autonomous Systems 13* (pp. 335–348). Springer. https://doi.org/10.1007/978-3-319-08338-4_25

Nagatani, K., Kiribayashi, S., Okada, Y., Otake, K., Yoshida, K., Tadokoro, S., Nishimura, T., Yoshida, T., Koyanagi, E., Fukushima, M., & Kawatsuma, S. (2013). Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots. *Journal of Field Robotics*, *30*(1), 44–63. https://doi.org/10.1002/rob.21439

Ozkan, M. F., Carrillo, L. R. G., & King, S. A. (2019). Rescue boat path planning in flooded urban environments. *2019 IEEE International Symposium on Measurement and Control in Robotics (ISMCR)*, B2-2-1–B2-2-9. https://doi.org/10.1109/ismcr47492.2019.8955663

Park, D. I., Kim, H., Park, C., & Kim, D. (2017). Design and analysis of the dual arm manipulator for rescue robot. *2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, 608–612. https://doi.org/10.1109/aim.2017.8014084

Petrlík, M., Báča, T., Heřt, D., Vrba, M., Krajník, T., & Saska, M. (2020). A robust UAV system for operations in a constrained environment. *IEEE Robotics and Automation Letters*, *5*(2), 2169–2176. https://doi.org/10.1109/lra.2020.2970980

Pfister, S. T., Roumeliotis, S. I., & Burdick, J. W. (2003). Weighted line fitting algorithms for mobile robot map building and efficient data representation. *2003 IEEE International Conference on Robotics and Automation (ICRA)*, *1*, 1304–1311. https://doi.org/10.1109/robot.2003.1241772

Redmon, J., & Farhadi, A. (2017). YOLO9000: Better, faster, stronger. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6517–6525. https://doi.org/10.1109/cvpr.2017.690

Rehman, B. U., Focchi, M., Lee, J., Dallali, H., Caldwell, D. G., & Semini, C. (2016). Towards a multi-legged mobile manipulator. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 3618–3624. https://doi.org/10.1109/icra.2016.7487545

Rouček, T., Pecka, M., Čížek, P., Petříček, T., Bayer, J., Šalanský, V., Heřt, D., Petrlík, M., Báča, T., Spurný, V., Pomerleau, F., Kubelka, V., Faigl, J., Zimmermann, K., Saska, M., Svoboda, T., & Krajník, T. (2020). DARPA subterranean challenge: Multi-robotic exploration of underground environments. In J. Mazal, A. Fagiolini, & P. Vasik (Eds.), *Lecture Notes in Computer Science: Vol. 11995. Modelling and simulation for autonomous systems* (pp. 274–290). Springer. https://doi.org/10.1007/978-3-030-43890-6_22

Sandy, T., Giftthaler, M., Dörfler, K., Kohler, M., & Buchli, J. (2016). Autonomous repositioning and localization of an in situ fabricator. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2852–2858. https://doi.org/10.1109/icra.2016.7487449

Schwarz, M., Beul, M., Droeschel, D., Klamt, T., Lenz, C., Pavlichenko, D., Rodehutskors, T., Schreiber, M., Araslanov, N., Ivanov, I., Razlaw, J., Schüller, S., Schwarz, D., Topalidou-Kyniazopoulou, A., & Behnke, S. (2018). DRC team NimbRo rescue: Perception and control for centaur-like mobile manipulation robot Momaro. In M. Spenko, S. Buerger, & K. Iagnemma (Eds.), *Springer Tracts in Advanced Robotics: Vol. 121. The DARPA Robotics Challenge Finals: Humanoid robots to the rescue* (pp. 145–190). Springer. https://doi.org/10.1007/978-3-319-74666-1_5

Sherstjuk, V., Zharikova, M., & Sokol, I. (2019). Forest fire fighting using heterogeneous ensemble of unmanned aerial vehicles. *2019 IEEE 5th International Conference Actual Problems of Unmanned Aerial Vehicles Developments (APUAVD)*, 218–223. https://doi.org/10.1109/apuavd47061.2019.8943826

Smarr, C.-A., Mitzner, T. L., Beer, J. M., Prakash, A., Chen, T. L., Kemp, C. C., & Rogers, W. A. (2014). Domestic robots for older adults: Attitudes, preferences, and potential. *International Journal of Social Robotics*, *6*(2), 229–247. https://doi.org/10.1007/s12369-013-0220-0

Sprunk, C., Lau, B., Pfaff, P., & Burgard, W. (2017). An accurate and efficient navigation system for omnidirectional robots in industrial environments. *Autonomous Robots*, *41*(2), 473–493. https://doi.org/10.1007/s10514-016-9557-1

Spurný, V., Báča, T., Saska, M., Pěnička, R., Krajník, T., Thomas, J., Thakur, D., Loianno, G., & Kumar, V. (2019). Cooperative autonomous search, grasping, and delivering in a treasure hunt scenario by a team of unmanned aerial vehicles. *Journal of Field Robotics*, *36*(1), 125–148. https://doi.org/10.1002/rob.21816

Stückler, J., Holz, D., & Behnke, S. (2012). Demonstrating everyday manipulation skills in RoboCup@Home. *Robotics and Automation Magazine*, *19*(2), 34–42. https://doi.org/10.1109/mra.2012.2191993

Tsitsimpelis, I., Taylor, C. J., Lennox, B., & Joyce, M. J. (2019). A review of ground-based robotic systems for the characterization of nuclear environments. *Progress in Nuclear Energy*, *111*, 109–124. https://doi.org/10.1016/j.pnucene.2018.10.023

Vale, A., & Gomes-Mota, J. (2007). LIDAR data segmentation for track clearance anomaly detection on over-head power lines. *Proc. IFAC Workshop*.

Ventura, R., Basiri, M., Mateus, A., Garcia, J., Miraldo, P., Santos, P., & Lima, P. (2016). A domestic assistive robot developed through robot competitions. *IJCAI 2016 workshop on autonomous mobile service robots*.

Viguria, A., Maza, I., & Ollero, A. (2010). Distributed service-based cooperation in aerial/ground robot teams applied to fire detection and extinguishing missions. *Advanced Robotics*, *24*(1-2), 1–23. https://doi.org/10.1163/016918609x12585524300339

West, C., Arvin, F., Cheah, W., West, A., Watson, S., Giuliani, M., & Lennox, B. (2019). A debris clearance robot for extreme environments. In K. Althoefer, J. Konstantinova, & K. Zhang (Eds.), *Lecture Notes in Computer Science: Vol. 11649. Towards autonomous robotic systems* (pp. 148–159). Springer. https://doi.org/10.1007/978-3-030-23807-0_13

Willmann, J., Augugliaro, F., Cadalbert, T., D'Andrea, R., Gramazio, F., & Kohler, M. (2012). Aerial robotic construction towards a new field of architectural research. *International Journal of Architectural Computing*, *10*(3), 439–459. https://doi.org/10.1260/1478-0771.10.3.439

How to cite this article: Basiri, M., Gonçalves, J., Rosa, J., Bettencourt, R., Vale, A., & Lima, P. (2021). A multipurpose mobile manipulator for autonomous firefighting and construction of outdoor structures. *Field Robotics*, *1*, 102–126.