Systems Article

Design and deployment of an autonomous unmanned ground vehicle for urban firefighting scenarios

Kshitij Jindal[1] , **Anthony Wang**[1], **Dinesh Thakur**[2] , **Alex Zhou**[2], **Vojtech Spurny**[3] ,
Viktor Walter[3] , **George Broughton**[3], **Tomas Krajnik**[3] , **Martin Saska**[3]
and Giuseppe Loianno[1,4]

[1]Department of Mechanical and Aerospace Engineering, New York University, New York, New York, USA
[2]Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania, Philadelphia, Pennsylvania, USA
[3]Faculty of Electrical Engineering, CTU in Prague, Prague, Czech Republic
[4]Department of Electrical and Computer Engineering, New York University, New York, New York, USA

Abstract: Autonomous mobile robots have the potential to execute missions that are either too complex or too dangerous for humans. In this paper, we address the design and deployment of an autonomous ground vehicle equipped with a robotic arm for urban firefighting scenarios. We describe hardware and algorithm designs for autonomous navigation, planning, fire source identification and abatement in unstructured urban scenarios. Our approach employs on-board sensors for autonomous navigation and thermal camera information for source identification. A custom electro–mechanical pump is responsible to eject water for fire abatement. The proposed approach is validated through several experiments, where we show the ability to identify and abate a simulated fire source in a building. The whole system was developed and deployed during the Mohamed Bin Zayed International Robotics Challenge (MBZIRC) 2020, for Challenge 3 – Fire Fighting Inside a High-Rise Building. Our approach was instrumental to win the first place in the MBZIRC Grand Challenge, which included the Challenge 3 as one its three tasks and it scored the highest number of points among all UGV solutions, while being the most compact one among all the teams.

Keywords: emergency response, navigation, GPS-denied operation

1. Introduction

Manipulator arms have proven their value in assisting humans and substantially speeding tasks in domains ranging from automotive (Banerjee et al., 2018; Lee et al., 2018; Rahman et al., 2019; Weng & Chen, 2017) to pharmaceutical (Chen et al., 2018; Saraee et al., 2017; Tamilselvi et al., 2018) while increasing human safety. Unmanned Ground Vehicles (UGVs) have been employed

to autonomously solve complex autonomous missions in dark or poorly-illuminated, confined, or cluttered environments (Ebadi et al., 2020; Thrun et al., 2003). A good overview and summary of various approaches investigated over the last 10 years is provided in Tardioli et al. (2019). Adding a robotic arm to a mobile base achieves a synergistic improvement: Arm flexibility allows the UGV to handle more complex tasks, while base mobility increases arm workspace and versatility. The complex configurations, achieved by the robotic arm, aid the mobile base immensely. For instance, by attaching a camera to the arm's end-effector and swinging the arm along its base joint, the camera gets a 360 degree view without requiring any UGV movement. The robotic arm holds the advantage of achieving complex configurations, which allows the mobile robot base to keep its movement efficient and simple. A mobile robotic arm provides additional mechanical redundancy easing the detection of objects in all the robot's directions.

In 2018, in the US alone, there were an estimated 379,600 residential building fires, resulting in an estimated \$8.1 billion loss (FEMA, 2020a), and 103,600 residential building fires, resulting in a \$2.6 billion loss (FEMA, 2020b). Reports (Ahrens, 2017; FEMA, 2020a, 2020b) indicate that systems capable of early fire suppression can reduce fire damage by more than 55% and life loss by 97%. In particular, automatic sprinkler systems can extinguish local fires by activating only a few nozzles, limiting the fire spread as well as the damage caused by the water. However, their installation is expensive and they require maintenance and regular checks. Moreover, traditional sprinkler heads typically spray a wide area, instead of delivering the water in a targeted way thus causing unnecessary damage, wasting water, and sometimes failing to reach the fire (Ahrens, 2017). These disadvantages can be mitigated by the use of small mobile robots, capable of early, autonomous, targeted fire suppression.

These challenges motivate the need to create autonomous robotic systems that can intervene early and assist humans in firefighting tasks typical in urban scenarios. Such missions constitute, as well, the main tasks in both Challenge 3 and the Grand Challenge of the Mohammed Bin Zayed International Robotics Challenge[1] (MBZIRC) 2020. The robot must detect and extinguish a simulated fire in an indoor environment. For this mission we designed and built an autonomous, mobile platform (Figure 1) incorporating multiple sensors and actuators that, working together, enable the robot to navigate to, detect, and "extinguish" simulated fires.

Robots designed for firefighting operations have been much discussed in recent years, and most currently deployed systems are teleoperated. The authors in Kim et al. (2009) introduce a lightweight portable robot that gathers information of the affected building and transmits it to a human controller. This strategy requires a human observer to control and guide the system within the environment, and furthermore, lacks the capability to extinguish a fire. In Khoon et al. (2012), the authors incorporate a flame sensor in their design. The major drawback in their method is that the mobile robot has to rotate 360° in order to scan the entire area. Moreover, the navigation system is based on line-following, which method does not scale well to real-world scenarios. In Uaday et al. (2019), a small and cost-effective firefighting solution based on Arduino Nano is presented. The approach does not address the autonomy challenges to operating in urban firefighting scenarios. The authors focus mainly on video–streaming capabilities, and the robot design is too small, having limited computational resources insufficient to operate in urban settings. The approach proposed in Hassanein et al. (2015) greatly summarizes the use of a fan as an extinguishing tool in many approaches. While this strategy may work well in small scale environments to extinguish candles in small-scale tests, it does not successfully scale to our envisioned scenario. Finally, multiple works (Altaf et al., 2007; Anantha Raj & Srivani, 2018; Maddukuri et al., 2016; Ye et al., 2019) present preliminary robot designs for firefighting scenarios. However, these solutions do not show any experimental validation or deployment in real–world settings to confirm the effectiveness of their proposed solutions.

In this paper, inspired by the MBZIRC 2020 Challenge 3 – Fire Fighting Inside a High-Rise Building as shown in Figure 2 which was also included as one of the key three tasks during the
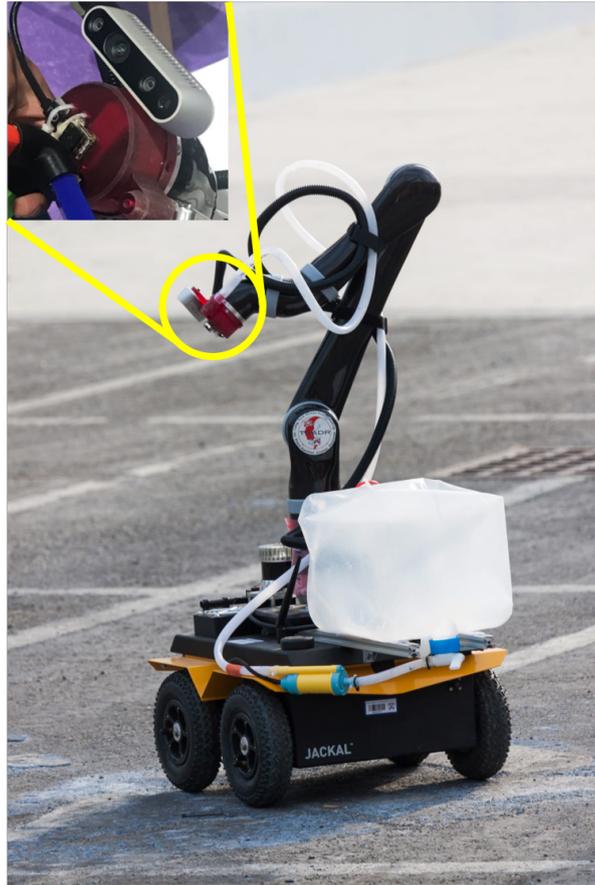
---

[1] http://mbzirc.com

Figure 1. Proposed UGV and manipulator system for firefighting while deployed during the MBZIRC 2020.

Figure 2. The tall structure simulating a high-rise building (left). A CAD of the building obtained by photogrammetry during rehearsals (right).

Grand Challenge, we address the design of an autonomous ground vehicle equipped with a robotic arm, which is able to detect heat sources and abate the corresponding fire employing a water-pump system. The task outline suggested that one liter of water successfully hitting the heat source as the goal to extinguish the synthetic fire. Our solution scored the highest number of points among all UGV firefighting solutions proposed by the 17 teams admitted to the competition during the Grand

Challenge[2], while concurrently being the most compact one. It was instrumental to win the Grand Challenge scenario.

This work presents multiple contributions. First, we present a novel mobile platform composed of a mobile base and an articulated robotic arm. The system is equipped with a water tank used to extinguish the detected fire. Second, we describe how to achieve reliable state estimation, planning, and control that enable our mobile robot solution to autonomously navigate to, locate, and extinguish simulated fires. Third, this is the first time that design, estimation, planning, and control problems are addressed simultaneously in a such challenging scenario, wherein the robot must enter a building and solve a fire–extinguishing task. Finally, we release our current framework as an open source package to the community (Thakur et al., 2020).

This work represents a first step toward deploying autonomous robots in real-world, urban, fire-fighting scenarios. The ability to test our methodologies in the MBZIRC test–bed allows us to evaluate the ability to transition our approaches to more complex settings. To contribute and boost this transition process, as well as to facilitate new teams becoming part of the MBZIRC in the upcoming editions, we release our designs and algorithms in open-source form.

The paper is organized as follows. Section 2 introduces the hardware design and software overview of the system. Section 3 presents our methodology to control the UGV and the incorporated robotic arm. Section 4 explains our strategy to detect the heat source and then to align the UGV as closely to the identified target. Section 5 presents our experimental results, and Section 6 concludes the work and presents future opportunities.

2. System overview

2.1. Hardware overview

In order to fulfill the task of extinguishing fire in an indoor environment, a fire-fighting ground robot has been designed. Figure 3 visualizes all the sensors integrated in the system independently.
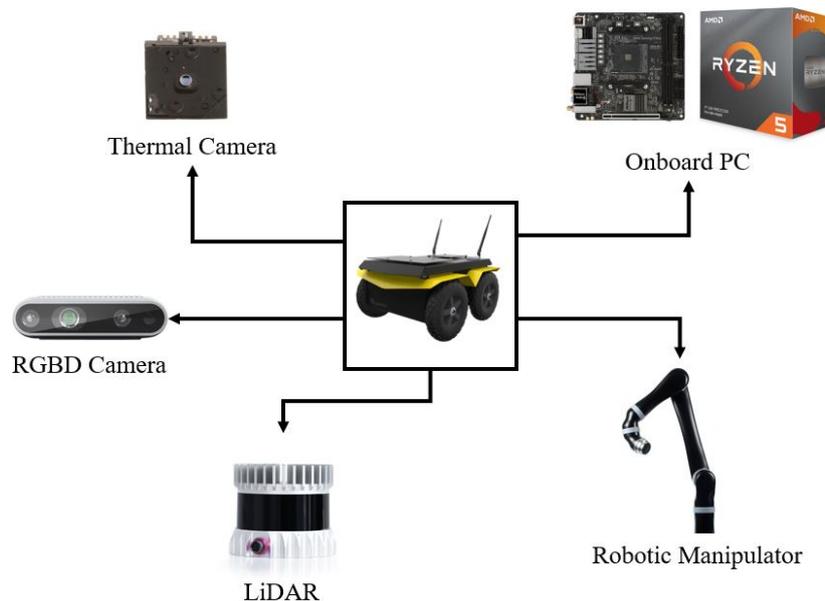

Figure 3. The proposed system architecture.

The system is made of an UGV Clearpath Jackal, a 32-beam Ouster LiDAR, a 6-DoF Kinova Gen2 robotic lightweight arm mounted on the top of the ground vehicle and equipped with an RGB camera, and a GSI outdoors 15 litre water container. The integrated robot is shown in Figure 1. The Ouster LiDAR is used to generate a point cloud of the environment for localization and mapping, along with the IMU integrated with Jackal UGV. A thermal camera is used for detecting high temperature hot–spots in the environment, and an RGB-D camera is utilized to measure its distance and position. A water-pump system is designed and manufactured to abate the detected fire.

The maximum speed of the UGV is 2.0 m/s, with a maximum payload of 20 kg. The UGV is also equipped with a 3DM-GX3-25[3] IMU, that has a data output rate of up to 1000 Hz. The Kinova Gen2 Ultra lightweight robotic arm is a 6 DoF curved wrist arm with a reach of 90 cm. It has a mid-range continuous maximum payload of 2.6 kg, with 2.2 kg full-reach peak/temporary maximum payload. It has a maximum linear arm speed of 20 cm/s. We designed a custom end-effector, which is equipped with an Intel RealSense D435[4] depth camera and a FLIR Lepton 3.5[5] LWIR (Long Wave Infrared) thermal camera. The Intel RealSense D435 depth camera is utilized on the system to estimate the depth of a heat source, if present. It has a maximum range of 10 m, with a Depth Field Of View (FOV) of 87°±3° × 58°±1° × 95°±3°. The RGB sensor FOV (H × V × D) is 69.4° × 42.5° × 77° (±3°). The RGB frame rate is 30 fps with the depth frame rate to be up to 90 Hz. The FLIR Lepton 3.5 LWIR thermal, attached to the designed end-effector of the arm, has a horizontal FOV of 57° and a diagonal FOV of 71°. The effective frame rate of the camera is 8.7 Hz with a thermal sensitivity of 0.05 °C. The Ouster OS-1 32 channel LiDAR is used in the system for autonomous navigation. It has a maximum range of 120 m with a precision of ±(1.5 cm to 5 cm), a vertical FOV of 45° with a vertical angular precision varying from 0.35° to 2.8°, and a horizontal FOV of 360°. An AMD Ryzen 5 3600 6-Core, 12-Thread[6] with 16 GB memory is used as onboard computer to run our algorithms.

A 15 liter water cube[7] is located at the rear of the UGV to supply the water pump. It is controlled by our system to pressurize the water through the piping system and exits from the nozzle. The direction of the nozzle can be altered by the movement of the robot arm and the UGV platform to guide the water stream to the desired location. The water is ejected from the nozzle, housed inside the end-effector supplied from a water pump. An overview of the water projecting mechanism is demonstrated in Figure 4.

The constraints and performance requirements that our fire extinguishing system needs to achieve are the water stream reach and flow rate. To prevent the water from reflecting back on to the robot once hitting a hard surface, the water should reach at least 1.5 meters. Therefore, after sizing the appropriate motor, piping system, and nozzle design, a 24 V DC inline pump that is capable of delivering 1 bar of pressure has been employed. The piping of the system routes the pressurized water stream from the base of the robot arm to the tip of the end-effector. The design of the end-effector is shown in the top-left corner of Figure 1.

The end-effector is designed to house three components, a RGB-D camera, a thermal camera, and a nozzle. In Figure 5, the flange extended on the top is used for housing the RGB-D camera, specifically designed for the Intel RealSense D435. The thermal camera is designed to mount right on the center of the circular surface and aligning with the axis of rotation of joint 6. Lastly, the nozzle is extended from the side of the end-effector and has a male receptor for connecting the piping and an opening on the other end for the jet stream to exit from. The nozzle is 0.15 inches in diameter which yields an exit velocity that meets our requirement. The water pump is controlled using an Arduino Nano[8]. The output of the Arduino is connected to a transistor which acts as an electronically activated switch to supply 24 V DC to the water pump. A diode is also placed into circuit to prevent

---

[3] https://www.microstrain.com/inertial/3dm-gx3-25
[4] https://www.intelrealsense.com/depth-camera-d435
[5] https://lepton.flir.com/news-and-updates/new-lepton-3-5-with-radiometry
[6] https://www.amd.com/en/products/cpu/amd-ryzen-5-3600
[7] https://gsioutdoors.com/15-l-water-cube.html
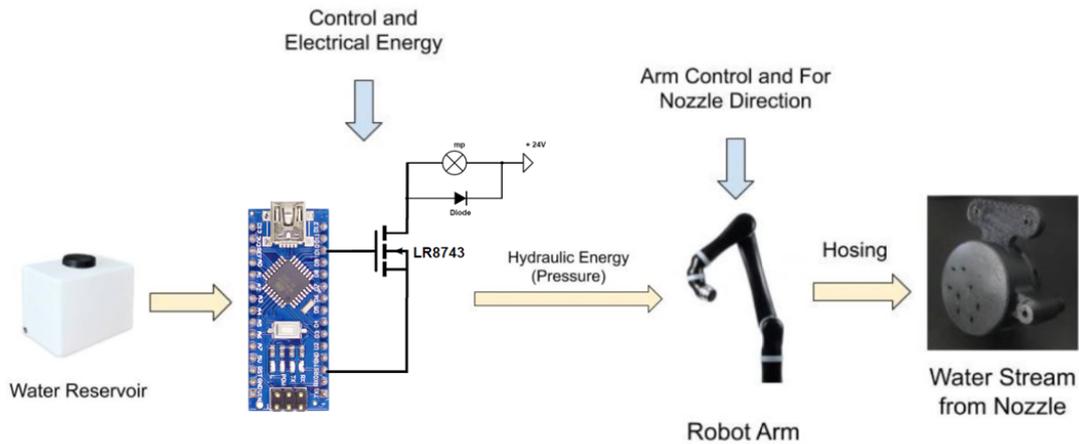[8] https://store.arduino.cc/usa/arduino-nano

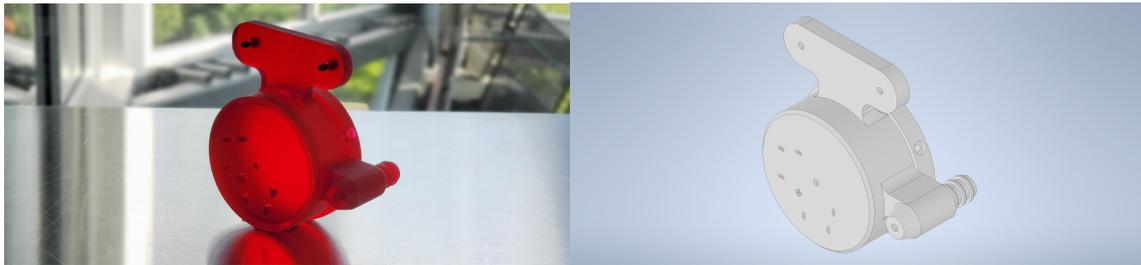**Figure 4.** Flowchart of the water-pump system.

Figure 5. Designed CAD end-effector (left) and corresponding 3D printed prototype (right).

back EMF from damaging the controller. Finally, the actuation of the water pump is controlled using PWM signal. The Arduino Nano uses ATmega328 USB to TTL serial chip to communicate with the on–board PC. Figure 4 presents the circuit diagram to control our water pump system.

2.2. Software overview

To achieve the autonomous fire detection task, the proposed system needs to accomplish different intermediate tasks such as autonomous navigation through the environment, the robot arm control, the detection of the heat sources, and the deployment of the water pump system. Using the hardware design described in the previous section, we provide an overview of the software system architecture. The implementation of all the functions relies on Robotic Operating System (ROS) (Quigley et al., 2009), which is an open-source development tool widely used nowadays in the robotics community. Since this is a mobile system, a navigation system is needed to drive the UGV from initial state to target locations. The combination of LiDAR along with the default IMU of the jackal UGV are the incorporated sensors which help the system in achieving this goal. Further, a heat detection algorithm is required. The thermal camera is the key sensor which is used to detect the heat source. Based on the heat source information, the RGB-D camera is used to locate it with respect to the robot arm. The water pump is then activated to achieve the final goal of extinguishing the fire at identified target locations. A control and planning policy is also needed for the robotic manipulator. Finally, we do require a state–machine that is able to activate and coordinate all the different sub–tasks in a sequential or concurrent manner.

3. Autonomous navigation and control

The ROS navigation stack move_base[9] is employed for navigating the Jackal UGV. The *move_base* package utilizes costmap_2d[10] approach to accomplish navigation tasks. Cost map, which aids in determining the areas in the map where the robot's motion is permitted, is calculated from an occupancy grid built from the sensory data. The generated cost map allows the UGV to plan its own path based on the goal location, obstacle configuration and actual robot position.

For estimating the UGV location, the system must have an established localization algorithm. For this purpose, we use Lightweight Ground-Optimized LiDAR Odometry and Mapping (LeGO-LOAM) (Shan & Englot, 2018) on the point cloud generated by the LiDAR. LeGO-LOAM is an extended version of LiDAR Odometry and Mapping (LOAM) (Zhang & Singh, 2014), which extracts edge and surface point features in a LiDAR point cloud based on the roughness of the local surface. Then the features are re-projected on the next scan based on a motion model. Finally, the 3D motion is estimated recursively by minimizing the sum of overall distances between point correspondences. LeGO-LOAM introduces a procedure to determine the ground plane and distinguishes between features detected on ground plane and on obstacles. The algorithm presents better accuracy when the Jackal IMU is used since it exploits the sensor fusion among the two sensors. Combining the localization algorithm with the move_base navigation stack, resulted in autonomous navigation of the robot in the tested environment.

The waypoint generation by the move_base is achieved by specifying the goal locations from the current location. For motion planning, the stack maintains two planners: a global planner that generates a plan for the robot to navigate avoiding obstacles and a local planner that generates velocity commands to track the global plan. The global planner goal location can either be obtained from the user or from the planning system itself. Once a goal location is specified, a cost map is generated such to guide the UGV from current location to the goal location. Cost map generation is dynamic, which makes it possible for the planner to update its path based on real-time occupancy.

To control the robot arm using ROS, we need to build and setup four different modules (Deng et al., 2017). First ROS needs to understand the modeling properties of our robot arm as well as the kinematic relationships between joints. This is conventionally defined in robot_description on the parameter server. Next, we need to utilize a motion planning algorithm to compute the optimal path for the robot to reach given goals. We determined that RRT-Connect (Kuffner & LaValle, 2000) in the Open Motion Planning Library (OMPL) is a suitable planner for our application as it presents a simple and efficient randomized algorithm for generating paths in high-dimensional configuration spaces. Third, we need to construct an environment model, either through pre-defined user settings or interpreted by the sensors in real time. This helps the robot arm to avoid collisions and understand where to place the end-effector to complete the mission. Lastly, our computer needs a method to communicate with the robot arm to convert all the generated trajectories into physical motions, both in simulation and in real world.

The four modules are conveniently taken care of by Moveit! (Chitta et al., 2012), a motion planning library which integrates the different aspects of robot control and ensures the information consistency between each module. By configuring it using the setup assistant GUI, we are able to load the robot properties in the form of Universal Robot Description Files (URDF). We also configure RRT-Connect planner to be compatible with our Kinova arm so the user can provide a goal in either Cartesian or joint space, and the planner generates a trajectory for the robot arm to reach the desired location. Additionally, communication to the robot controllers is setup to execute the planned trajectories. The Kinova low–level driver publishes joint state information and provides a routine to execute trajectory tracking. We can achieve the following objectives

- Integrate the kinematics information of our robot arm to the motion planning library

---

[9] http://wiki.ros.org/move_base
[10] http://wiki.ros.org/costmap_2d

- Collision checking of robot arm with the UGV, sensors and peripherals mounted on the UGV, and the environment
- Take into account the physical hardware limits of the specific robot arm we are using while planning and executing trajectories for task completion
- Dynamically update the location of the robot arm and environment
- Notify user when a specific motion execution is completed

All of this is done by one script file, which is launched when starting up the system. The library contains many useful APIs we can directly or indirectly utilize to achieve the firefighting tasks such as setting goals and executing trajectories. However, considering the re-usability of code and convenience of operation, it is desired to create a task level planner that generalizes and wraps all the useful APIs into one single function. Therefore, we created an interface that aggregates all the necessary APIs provided by Moveit! into a single action server and standardized the action goal. Rather than writing multiple routines for path planning, users can now simply call the task-level planner using a unified message to activate the path planning pipeline. Therefore, when controlling the robot arm in state machines, or stand-alone calls, the user can easily interface with the robot arm without having to write a separate node.

4. Heat source detection and localization

To extinguish the simulated fire, the water stream has to pass through a relatively narrow aperture. While the localization drift of the used navigation system was low, the overall positional error after travelling several meters would prevent the water stream to reach the heat source even if the heat source position would be known a priori. Therefore, we implemented a module, which would locate the heat source and aim the end effector precisely enough to reach the heat source.

The artificial fire sources comprised anodized aluminium heating elements of dimensions $3.5\,cm \times 6.0\,cm$ mounted on plexiglass backplates, and were covered 15 cm in the front by another plexiglass plate with a circular hole 15 cm in diameter, see Figure 7. We detected these heating elements using FLIR Lepton 3.5 LWIR thermal camera that outputs an image where the values of pixels correspond to the estimated temperature in the given region. The heating elements were set to maintain a temperature of 120 °C, but due to the lower emissivity of anodized aluminium of 0.55 (Minkina & Dudzik, 2009) compared to the reference value of 1 appear to be of lower temperature, approximately 70 °C. See Figure 6 for an example of the view of the heat source from the thermal camera during the competition.
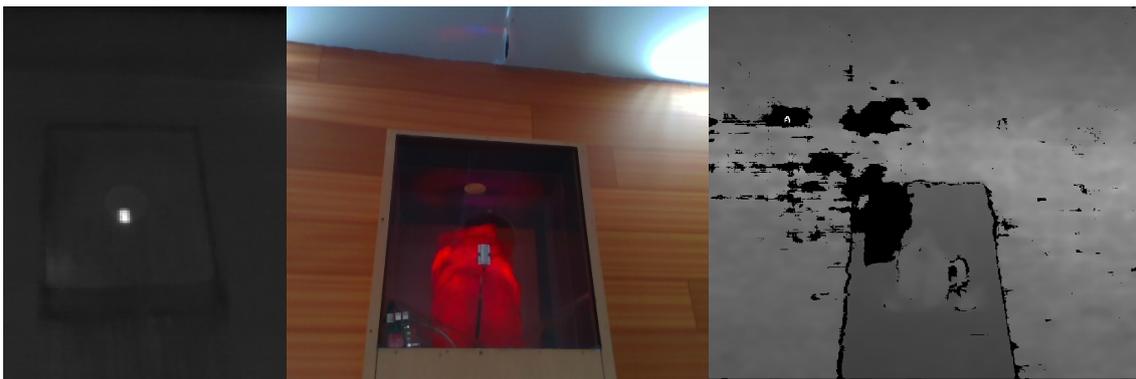

Figure 6.   The white region represents the heat source present in thermal camera feed (left). Corresponding RGB and depth images of the heat target (middle, right).
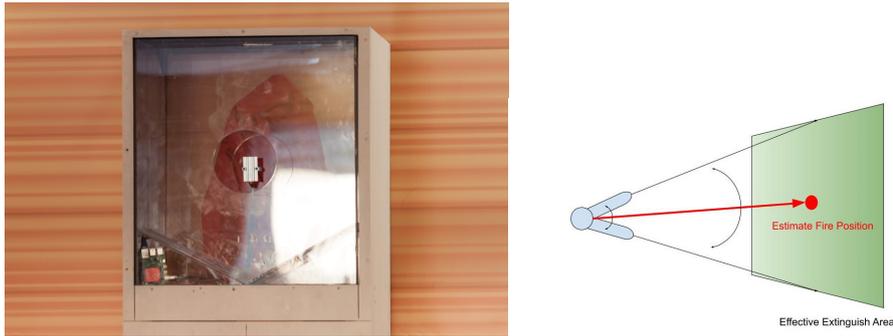
Figure 7. Image of the target utilised in the MBZIRC 2020 challenge (left) and image visualizing area covered by the water spray with the added movement (right).

We assumed that the detection in this sub-task was only performed indoors where the effects of direct sunlight can be neglected and that no other objects of such temperatures will be present. We could therefore detect the targets by simple value threshold. Their image centers were transformed to direction vectors based on the presumed thermal camera matrix, derived from the FOV and pixel resolution of the IR camera, assuming its pinhole model. This gave us the information on the direction of the target, which would be sufficient if the heating element was close enough to consider the water stream straight.

To reach far targets, we acquire the information on their distance by an Intel RealSense D435 depth camera attached close to the thermal camera itself with known relative transformation between the two (5 cm offset). Assuming pinhole camera model for the thermal camera the relative distance between the robot and the target is estimated using this ray and the depth obtained from the depth camera by re-projecting the the direction vectors of the IR camera on the depth camera.

The precision of the distance estimate varied based on the incidence angle with respect to the front plexiglass of the target. Since the plexiglass was transparent, sometimes the retrieved distance corresponded to the distance to the back plate instead of the front plate, or due to averaging over larger area, the distance estimate sometimes came out in between the two plates. However, this difference was not significant for this application, since the distance of the back plate to the front one was much smaller compared to the shooting distance. Additionally, having a line of sight to the heating element meant that we were also aiming inside of the front opening, since the plexiglass is opaque for the IR radiation band of our thermal camera.

The final output of the heating element localization module is a 3D coordinate of the heating element in the coordinate frame of the camera mounted on the end effector along with the nozzle. This position was recalculated at the IR camera FPS, i.e. at 8 Hz, which allowed to perform the final nozzle alignment in a closed-loop manner. To perform both horizontal and vertical alignment we implemented two separate feedback loops controlling concurrently the robot and arm motion.

The movement of the mobile base was used to align the robot in horizontal plane. We used the laser measurements to align the robot with the furniture or wall with the artificial fire source and keep a distance of 1.5 m. Once aligned to the wall, the robot can compensate the horizontal displacement of the simulated fire source simply by moving forwards and backwards. In this way, the water spray faces the artificial fire source almost perpendicularly, which maximises the cross section of the aperture it has to pass through in order to extinguish the simulated fire. The vertical alignment of the nozzle is performed simply by moving the third joint of the robot manipulator, which changes the nozzle pitch. Once the vertical and horizontal alignment is finished, the water pump is turned on.

In Figure 7 (left), it can be observed that the open area of the artificial fire is rather small. Thus, even a small misalignment or small pressure drop can cause the water stream to miss the target. Such a situation is difficult to detect, because the water stream can obstruct the view of the artificial fire.

To avoid this, we increased the coverage of the water spray by introducing small, cyclic movements of two arm joints around the aligned position. This results in the final spray area to form a squared section which overlaps the simulated fire aperture. This squared area is effectively the tolerance zone of our water spray aiming. The covered area obtained by adding this movement is visualized in Figure 7 (right). In a real system, it is worth to point out that these movements would correspond to a situation, where the water is spread not only on the simulated fire, but also around it to prevent possible fire spreading.

The different steps of the heat source detection and localization process are handled by the system state machine described in Section 5.1 and depicted in Figure 9.

5. Experimental results

In this section, we present the results and performances of our system at the MBZIRC 2020 Challenge 3 – Fire Fighting Inside a High-Rise Building shown in Figure 2 (right). The key objective of this work is to navigate and localize the UGV, identify the heat source, and extinguish the heat source in the ground area of the building as shown in Figure 8. Since the test were performed on synthetic heat, by setting heat sources. The objective of this task was to shoot 1 liter of water at the heat source target. We developed a high-level execution policy to manage all the tasks described in the previous sections, using ROS. The execution policy, from here on referred as state machine, was developed to progress from one function to another based on the outcome of the current function. The state machine was developed leveraging the behavior engine FlexBE (Schillinger et al., 2016). It provides an extensive user interface to combine different states and execute them. The autonomy level for the transitions can be set by the user to control the system. Data can also pass from one state to the other.

Each state in FlexBE corresponds to an action executed by the robotic system. It is a good platform for navigating a robot by providing waypoints, as well as for controlling a robotic manipulator. To

Figure 8. UGV navigating towards the target (left), performing heat source detection (center), and abating the detected heat (right).
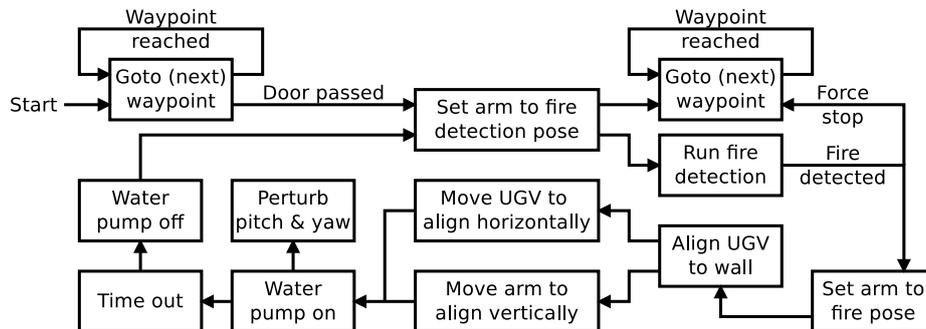

Figure 9. Overview of the state machine. The top part of the pipeline regards navigation and the bottom part corresponds to the fire extinguishing activity.

exploit the multiple features of our system such as activating and deactivating the water pump system, detecting the heat source using the thermal camera, and for aligning the robot to the target, we developed appropriate customized functions and algorithms using ROS. The system transition from one state to another depends on its current state and corresponding action outcome. Leveraging this information, the system progress from one state to another of the state machine. All the transitions were fully autonomous. Figure 9 represents the high-level state machine running on our system and responsible for the varied behavior execution policies.

5.1. State machine

The state machine, shown in Figure 9, combines waypoint-based navigation with a fire detection and suppression procedure. There are two sets of waypoints: outdoor and indoor waypoints. When navigating outdoors, the robot has its arm in a folded position, so it can accelerate and turn rapidly without the risk of slipping wheels or tripping over. Indoor navigation is performed slowly with the arm in a raised position, so that the camera can inspect the wall surfaces completely.

During the competition the start location with respect to building was roughly known and the building was within scanning range of the LiDAR. To avoid lengthy exploration of the entire arena, we exploited this a priori knowledge and set the navigation waypoints manually. In particular, we processed the 3D LiDAR scans and extracted the vertical planes of the building facade to obtain a 2D layout of the building. This 2D layout was subsequently used to determine the positions of the waypoints the robot navigated through. The projected 2D layout also allows doors/gaps to be extracted easily.

Inside the building, the vertical planes were assumed as the most appropriate structure to help guiding the robot. In particular, the plane left on the robot was assumed and the waypoints were set so that the robot would move in clockwise fashion around the central piedestal of the building. To ensure that the robot would inspect all of the wall surfaces, we extended the arm to position the end effector as high as possible.

The distance from the walls was determined so that considering the FOV of the thermal camera, all of the inspected surfaces would be seen at least once. Given the diagonal FOV of the thermal camera and distance from the wall we can assume that a fire will be easily visible if the camera height is kept fixed. This simplifies the search problem as the arm can be held in a fixed position and the robot just navigates alongside the wall. Hence we choose to set the waypoints to emulate a simple wall following strategy while searching the thermal source. The two likely candidates for fire location were known a priori: the outer walls of the room and a central pedestal. This requires two possible loops inside the room, one with thermal camera pointing inwards while the other pointing thermal camera outwards. For the first loop inside the room we chose to point the camera inwards and follow the walls.

While following the wall, we monitor the thermal camera output and if a temperature spot above certain threshold is reported, we stop the navigation. As the heat source is detected, we execute the fire extinguish steps, which are shown in the bottom part of Figure 9. The arm is first set to fire abating position and the robot aligns itself to the wall. The final aiming is then realised by moving the robot forward and backward and the nozzle up and down as described in Section 4. The pump is then set on and the nozzle yaw and pitch is perturbed in circular fashion to increase the water delivery area. After 20 seconds, the pump is switched off and we check if the fire is still present. If the heat source is detected, we run the fire extinguish steps again. If not, the robot proceeds to the next waypoint.

5.2. Results

Four experiments were performed as part of the MBZIRC 2020 Challenge 3 and the Grand Challenge. While one can argue that the number of experiments is not sufficient to obtain statistically significant results, their aim is not to measure precision or accuracy, but to demonstrate reliability and robustness

of the solution when deployed in a near-real scenario. Moreover, the experiments are evaluated by an international jury, that does not allow to rule out cases, where the system failed for technical difficulties and of course numerous preliminary experiments of the system components as well as the entire system were done in the testing phase prior to the competition. The evaluation of the experiments is based purely on the ability to deliver water through the narrow aperture of the fire targets. The key questions of the experiments were: Is the proposed design made by off-the shelf components sufficient to accomplish the proposed task? More specifically, is the SLAM-based navigation and thermal source location pipeline precise enough to realize the fire abating by itself or do we need a priori knowledge and specialized fire detection modules?

In all of the experiments, we let the robot reach the target locations with the arm extended in a way that in a case of precise localization, it would hit the simulated fire. In the first run, we did not use an a priori known map build by the robot 3D scanner and we simply set the waypoints using hand-measured positions. In the subsequent runs, the waypoints were set using the map created by the 3D scanning system. In the first and second experimental run, we did not use the final nozzle alignment specified in Section 4. In the third and fourth run, we used the fire detection modules to perform the final nozzle alignments. During first run, the hand-set waypoints were not precise enough to allow the path planning system to reliably pass through the door and the robot did not even enter the building. The second run used refined waypoints, and the robot passed through the door and navigated the building interiors without problems. However, the localization drift prevented the robot to deliver water to the a priori known artificial fire spots. In the third and fourth runs, performed during the Grand Challenge round, the robot performed the final nozzle alignment using the methods specified in Section 4. In both of these runs, the artificial fire was detected correctly and water was delivered to it with ease. The average time to accomplish the overall mission was around 3 minutes. The attached multimedia material[11] further shows the successful performance of our system.

Figure 10 shows the trajectory of the UGV during the experiment. The trajectory is shown in two different views, to visualize the arena in which the tests were performed. The multicolored trail express the trajectory of the robot in the images. To avoid tipping over backwards while carrying the heavy water payload the UGV was operated at maximum velocity of 0.6 m/s. Loop-closure was not performed during the competition, but during manual tests the localization was sufficiently precise. We empirically verified, due to the absence of ground truth, using the environment dimensions that similar localization errors are obtained compared to the one reported in original paper (Shan & Englot, 2018). The localization and mapping algorithm runs at 10 Hz and consumes an average of 40% of the overall CPU.
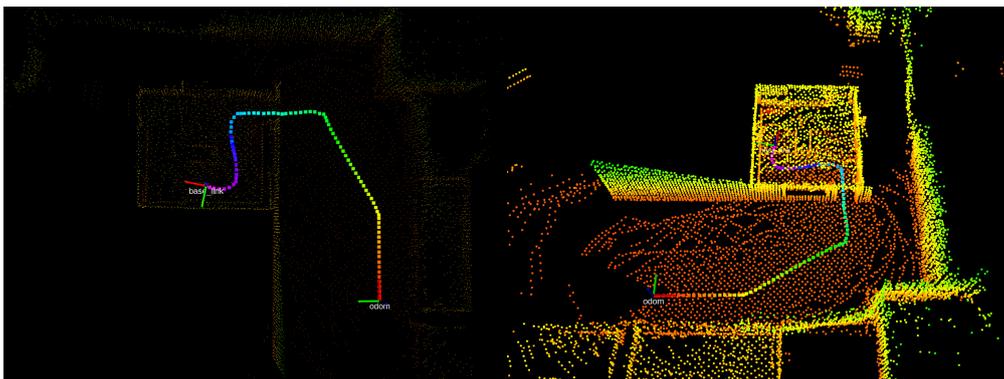

Figure 10. Trajectory of the UGV represented from two angles.

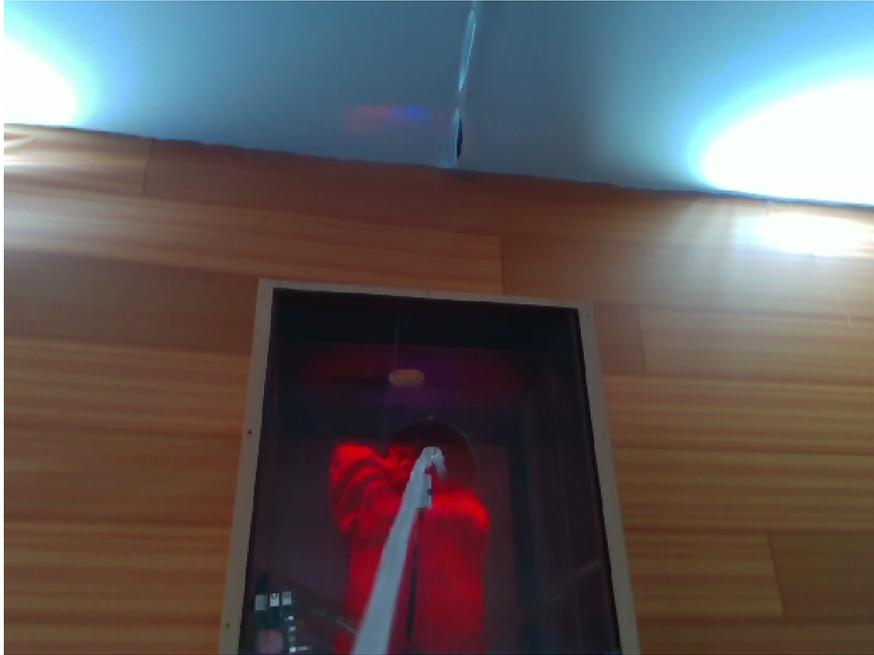[11] https://youtu.be/HCqQMEmkiN8

Figure 11. RGBD camera feed showing water being ejected to the heat source.

Figure 6 shows the feed from the thermal camera. The white blob in the image represents the heat source, which is successfully detected. The pixel location of the detected heat source in the thermal camera image is transformed to the RGB-D camera frame, using the approach explained in Section 4. Once the transformation is performed, the heat source is localized, the UGV is aligned using the two step approach previously described, and the water is ejected on the heat source. Figure 11 shows the image acquired from the RGB-D camera while the water is dispensed by the system to the detected heat source. Once the heat source is detected by thermal camera, and localized by the RGB-D camera, the water pump is turned on. The added cyclic motion of the arm increases the likelihood to shoot the target, while also covering greater area, which is beneficial in actual scenarios. The aim of the competition was to shoot 1 liter of water towards the heat source. This goal was achieved successfully, but due to the size of the UGV, the water carried was not sufficient to extinguish the fire in one trip. The water container was refilled to extinguish the fire completely. Upon the refill, the robot performed all the tasks again successfully, with full autonomy.

5.3. Lesson learned

We discuss the research and technological challenges related to our approach and how the proposed solution can be transitioned from the MBZIRC test–bed to more complex real world urban firefighting scenarios.

First, the firefighting task definitely showed the importance to engage different researchers with cross-disciplinary competences in complementary research domains, such as localization, autonomous navigation, robotic arm control, fire detection, and hardware design to integrate the different robots parts and to obtain an end effector to house a thermal camera, stereo camera, and hose. The proposed solution can definitely inspire early roboticists to understand how the modules in the system are interconnected and work together to create an autonomous mobile robot solution for search and rescue scenarios.

Second, the system is not a monolithic solution to the firefighting problem. Rather, it can be considered as a first step towards an efficient, rapid response system capable of early fire suppression. The main problem of such highly task specialized systems – the cost is actually mitigated by the versatile and modular design of our robot. A mobile base with a manipulator could perform a variety of other tasks, such as delivery or facility inspection and the fire suppression could be achieved simply by exchanging the end effector (Kong et al., 2004). The presented system demonstrated that an off-the-shelf robotic platform, employed with customized state-of-the-art navigation, image processing and control methods can perform the fire suppression task reliably and efficiently. Several customized parts needed were rapidly prototyped and 3D printed within few hours. These aspects indicate that the field of mobile robotics is mature enough to deploy robots capable of autonomous, targeted fire suppression. Thus, mobile robots might soon compete with traditional sprinkler systems.

Finally, while the experiments indicate that the technology is ready to be deployed in buildings or small residential clusters, complex urban scenarios require more advanced, socially-aware navigation (Kucner et al., 2017; Vintr et al., 2020), capable to deal with low visibility (Broughton et al., 2021). These capabilities have not been addressed in the proposed approach and will require further future investigation. On the other hand, the proposed approach for robot localization provides direct feedback control to the robot event transitioning from outdoor to indoor settings. The water tank and automated water pump designs for fire abatement are simple, inexpensive, and effective. Similar mechanical and circuitry designs and concepts can be used by private or government organization as firefighting solution. To demonstrate the versatility of the approach, we open source our mechanical and electrical designs and navigation solution (Thakur et al., 2020).

6. Conclusion

In this work, we developed a mobile manipulator equipped with a specialized end–effector to detect "simulated fires" in indoor environments and extinguish them. Challenges included autonomous navigation of the mobile platform, designing for heat source identification, and control of a water pump system to achieve fire abatement. The presented results show the effectiveness of our approach and its applicability in real firefighting scenarios. We believe that the proposed solution can be considered a first step toward the transitioning ground robotic solutions to realistic, rapid, fire suppression systems and search and rescue scenarios. The system has been designed as part of a solution in the MBZIRC 2020 for the Challenge 3 -– Fire Fighting Inside a High-Rise Building and for the Grand Challenge. The proposed solution was instrumental and one of the key elements to win the Grand Challenge in the MBZIRC competition as captured in Figure 12. It was the smallest UGV solution proposed among the competing teams and scored during the Grand Challenge the highest number of points among all UGV firefighting solutions presented by the 17 teams admitted to the final competition.

Future works will include the ability to increase autonomy and localization by detecting semantic cues in the environment, tracking heat sources while simultaneously moving the UGV and the arm, and incorporating exchangeable end-effectors with varied capabilities. Although developed for the specific needs of the competition, the system has shown to be a promising solution to be employed in future complex, real-world, urban firefighting scenarios and offers commercial potential, as well.

Acknowledgments

Figure 12. Our team during the ceremony celebrating the first place in the Grand Challenge.

ORCID

Kshitij Jindal ⬤ https://orcid.org/0000-0001-5408-3612
Dinesh Thakur ⬤ https://orcid.org/0000-0001-5046-8160
Vojtech Spurny ⬤ https://orcid.org/0000-0002-9019-1634
Viktor Walter ⬤ https://orcid.org/0000-0001-8693-6261
George Broughton ⬤ https://orcid.org/0000-0003-0071-5834
Tomas Krajnik ⬤ https://orcid.org/0000-0002-4408-7916
Giuseppe Loianno ⬤ https://orcid.org/0000-0002-3263-5401

References

Ahrens, M. (2017, July). *US experience with sprinklers* (tech. rep.). https://www.nfpa.org//-/media/Files/News-and-Research/Fire-statistics-and-reports/Suppression/ossprinklers.pdf

Altaf, K., Akbar, A., & Ijaz, B. (2007). Design and construction of an autonomous fire fighting robot. *2007 International Conference on Information and Emerging Technologies*, 1–5. https://doi.org/10.1109/iciet.2007.4381341

Anantha Raj, P., & Srivani, M. (2018). Internet of robotic things based autonomous fire fighting mobile robot. *2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 1–4. https://doi.org/10.1109/iccic.2018.8782369

Banerjee, D., Yu, K., & Aggarwal, G. (2018). Image rectification software test automation using a robotic arm. *IEEE Access*, *6*, 34075–34085. https://doi.org/10.1109/access.2018.2846761

Broughton, G., Majer, F., Rouček, T., Ruichek, Y., Yan, Z., & Krajník, T. (2021). Learning to see through the haze: Multi-sensor learning-fusion system for vulnerable traffic participant detection in fog. *Robotics and Autonomous Systems*, *136*, 103687. https://doi.org/10.1016/j.robot.2020.103687

Chen, X., Zhao, B., & Gao, X. (2018). Noninvasive brain-computer interface based high-level control of a robotic arm for pick and place tasks. *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 1193–1197. https://doi.org/10.1109/fskd.2018.8686979

Chitta, S., Sucan, I., & Cousins, S. (2012). MoveIt! [ROS topics]. *IEEE Robotics & Automation Magazine*, *19*(1), 18–19. https://doi.org/10.1109/mra.2011.2181749

Deng, H., Xiong, J., & Xia, Z. (2017). Mobile manipulation task simulation using ROS with MoveIt. *2017 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, 612–616. https://doi.org/10.1109/rcar.2017.8311930

Ebadi, K., Chang, Y., Palieri, M., Stephens, A., Hatteland, A., Heiden, E., Thakur, A., Funabiki, N., Morrell, B., Wood, S., Carlone, L., & Agha-mohammadi, A.-a. (2020). LAMP: Large-scale autonomous mapping and positioning for exploration of perceptually-degraded subterranean environments. *2020 IEEE International Conference on Robotics and Automation (ICRA)*. https://doi.org/10.1109/icra40945.2020.9197082

FEMA. (2020a, March). *Fire estimate summary*. https://www.usfa.fema.gov/downloads/pdf/statistics/res_bldg_fire_estimates.pdf

FEMA. (2020b, March). *Fire estimate summary*. https://www.usfa.fema.gov/downloads/pdf/statistics/nonres_bldg_fire_estimates.pdf

Hassanein, A., Elhawary, M., Jaber, N., & El-Abd, M. (2015). An autonomous firefighting robot. *2015 International Conference on Advanced Robotics (ICAR)*, 530–535. https://doi.org/10.1109/icar.2015.7251507

Khoon, T. N., Sebastian, P., & Saman, A. B. S. (2012). Autonomous fire fighting mobile platform [International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012)]. *Procedia Engineering*, *41*, 1145–1153. https://doi.org/10.1016/j.proeng.2012.07.294

Kim, Y.-D., Kim, Y.-G., Lee, S.-H., Kang, J.-H., & An, J. (2009). Portable fire evacuation guide robot system. *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2789–2794. https://doi.org/10.1109/iros.2009.5353970

Kong, F.-k., Diao, Y.-f., & Yang, E.-x. (2004). Research on automatic robot manipulator exchange. *Mechanical Engineer*, *5*.

Kucner, T. P., Magnusson, M., Schaffernicht, E., Bennetts, V. H., & Lilienthal, A. J. (2017). Enabling flow awareness for mobile robots in partially observable environments. *IEEE Robotics and Automation Letters*, *2*(2), 1093–1100. https://doi.org/10.1109/lra.2017.2660060

Kuffner, J. J., & LaValle, S. M. (2000). RRT-connect: An efficient approach to single-query path planning. *2000 IEEE International Conference on Robotics and Automation (ICRA)*, *2*, 995–1001. https://doi.org/10.1109/robot.2000.844730

Lee, J.-D., Li, W.-C., Shen, J.-H., & Chuang, C.-W. (2018). Multi-robotic arms automated production line. *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*, 26–30. https://doi.org/10.1109/iccar.2018.8384639

Maddukuri, S. V. P. K., Renduchintala, U. K., Visvakumar, A., Pang, C., & Mittapally, S. K. (2016). A low cost sensor based autonomous and semi-autonomous fire-fighting squad robot. *2016 Sixth International Symposium on Embedded Computing and System Design (ISED)*, 279–283. https://doi.org/10.1109/ised.2016.7977097

Minkina, W., & Dudzik, S. (2009). *Infrared thermography: Errors and uncertainties*. John Wiley & Sons.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. Y. (2009). ROS: An open-source Robot Operating System. *ICRA Workshop on Open Source Software*, *3*(2).

Rahman, R., Rahman, M. S., & Bhuiyan, J. R. (2019). Joystick controlled industrial robotic system with robotic arm. *2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON)*, 31–34. https://doi.org/10.1109/raaicon48939.2019.18

Saraee, E., Joshi, A., & Betke, M. (2017). A therapeutic robotic system for the upper body based on the Proficio robotic arm. *2017 International Conference on Virtual Rehabilitation (ICVR)*, 1–2. https://doi.org/10.1109/icvr.2017.8007498

Schillinger, P., Kohlbrecher, S., & von Stryk, O. (2016). Human-robot collaborative high-level control with application to rescue robotics. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2796–2802. https://doi.org/10.1109/icra.2016.7487442

Shan, T., & Englot, B. (2018). LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4758–4765. https://doi.org/10.1109/iros.2018.8594299

Tamilselvi, R., Merline, A., Beham, M. P., Anand, R. V., Karthik, M. S., & Uthayakumar, R. H. (2018). EMG activated robotic arm for amputees. *2018 2nd International Conference on Inventive Systems and Control (ICISC)*, 456–461. https://doi.org/10.1109/icisc.2018.8399114

Tardioli, D., Riazuelo, L., Sicignano, D., Rizzo, C., Lera, F., Villarroel, J. L., & Montano, L. (2019). Ground robotics in tunnels: Keys and lessons learned after 10 years of research and experiments. *Journal of Field Robotics*, *36*(6), 1074–1101. https://doi.org/10.1002/rob.21871

Thakur, D., Loianno, G., Štibinger, P., & Krajník, T. (2020). *MBZIRC firefighting challenge UPENN-CTU-NYU solution* [Computer software]. GitHub. https://github.com/gestom/mbzirc_upenn

Thrun, S., Hahnel, D., Ferguson, D., Montemerlo, M., Triebel, R., Burgard, W., Baker, C., Omohundro, Z., Thayer, S., & Whittaker, W. (2003). A system for volumetric robotic mapping of abandoned mines. *2003 IEEE International Conference on Robotics and Automation (ICRA)*, *3*, 4270–4275. https://doi.org/10.1109/robot.2003.1242260

Uaday, M. A., Nazmul Islam Shuzan, M., Shanewaze, S., Rakib, R. I., & Zaman, H. U. (2019). The design of a novel multi-purpose fire fighting robot with video streaming capability. *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, 1–5. https://doi.org/10.1109/i2ct45611.2019.9033931

Vintr, T., Yan, Z., Eyisoy, K., Kubis, F., Blaha, J., Ulrich, J., Swaminathan, C. S., Molina, S., Kucner, T. P., Magnusson, M., Cielniak, G., Faigl, J., Duckett, T., Lilienthal, A. J., & Krajnik, T. (2020). Natural criteria for comparison of pedestrian flow forecasting models. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 11197–11204. https://doi.org/10.1109/iros45743.2020.9341672

Weng, C.-Y., & Chen, I.-M. (2017). The task-level evaluation model for a flexible assembly task with an industrial dual-arm robot. *2017 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM)*, 356–360. https://doi.org/10.1109/iccis.2017.8274801

Ye, Z., Su, F., Zhang, Q., & Wan, L. (2019). Intelligent fire-fighting robot based on STM32. *2019 Chinese Automation Congress (CAC)*, 3369–3373. https://doi.org/10.1109/cac48633.2019.8996761

Zhang, J., & Singh, S. (2014). LOAM: Lidar odometry and mapping in real-time. *Robotics: Science and Systems X*. https://doi.org/10.15607/rss.2014.x.007