A Robotic Health-Care Assistant for the COVID-19 Emergency

A Proposed Solution for Logistics and Disinfection in a Hospital Environment

©SHUTTERSTOCK.COM/TOYTING

By Christian Tamantini, Francesco Scotto di Luzio, Francesca Cordella, Giuseppe Pascarella, Felice Eugenio Agrò, and Loredana Zollo

The COVID-19 pandemic and the related emergency have contributed to the push for innovative solutions applied to health care. In particular, robotics has shown huge potential for contributing to pandemic relief efforts and improving people's quality of life in several scenarios. In this article, a robotic system, characterized by interaction capabilities and autonomous navigation, is developed to be used in a COVID-19 health-care treatment center for logistics and disinfection purposes. The article describes the two-month use of the platform in the University Hospital Campus Bio-Medico (UCBM) COVID-19 treatment center in Rome, Italy, and presents experimental results for the robot's

Digital Object Identifier 10.1109/MRA.2020.3044953
Date of current version: 28 January 2021

navigation capabilities in unstructured environments and collaborative activities with health-care operators in the clinical setting.

Managing a Pandemic

On 31 December 2019, a pneumonia of unknown cause was detected in Wuhan, China. In a short time, the outbreak was declared a public health emergency of international concern. On 11 February 2020, the World Health Organization announced the name of the new disease: COVID-19. Severe acute respiratory syndrome coronavirus has affected people worldwide, causing more than 440,000 deaths [1].

Robotics has shown huge potential for contributing to pandemic relief efforts and for improving people's quality of life in several scenarios.

During the past several months, various solutions from the fields of prevention, diagnosis, and treatment have been considered for managing the pandemic. Although no specific treatment has been developed, the main solutions seem to be preventive measurements, i.e., quarantines, social distancing, and hygienic precautions [1]. The latter refers to both personal hygiene and surface disinfection, which are pivotally important to minimize the risk of contamination, especially in public environments, such as hospitals. Moreover, a fast diagnosis is crucial to limit the diffusion of the virus, especially by asymptomatic people, and the real-time polymerase chain reaction (RT-PCR) plays a fundamental role in the early identification of affected subjects. For this reason, to date, more than 4 million RT-PCR tests have been performed only in Italy.

However, these measures require more health workers, with a related risk of contagion in hospital settings, and this could result in a further concern: since an infected caregiver is a potential vehicle for virus dissemination, it becomes clear that avoiding the risk of infection among health-care providers is paramount and that every effort should be taken in this direction. Moreover, a sudden lack of medical staff due to illness would expose the health system to a critical overload. At the same time, robotics researchers across the world have been thinking about how they could help. The most relevant robotics contributions in this direction are summarized in [2]. They can be grouped into three main categories: 1) patient-care robots, both inside and outside clinical settings; 2) automation solutions, especially for logistics; and 3) environment disinfection platforms. Figure 1 shows examples of the developed robotics solutions.

Advances in digital technologies have promoted the development of solutions for remote patient monitoring. The COVID-19 pandemic is characterized by a very high contamination level; hence, it is becoming more and more important to break the transmission chain by avoiding physical contact with people. It is worth observing that health-care facilities could be sources of contamination and that face-to-face contact between clinicians and patients needs to be avoided. Therefore, examples of preliminary COVID-19 diagnoses that were conducted remotely can be found in recent literature [3].

Interactive robots have been introduced inside the hospital environment to assist health-care operators in different activities [4]. The Vici robot [5], for example, developed by InTouch Health, has a high-definition screen and camera that enable doctors and nurses to assist patients via videoconferencing services, without physical contact. Ginger, another robot, welcomes incoming patients and visitors, monitoring some of their vital signs, including taking people's temperature using the CloudMinds infrared thermometric system [6]. A challenging application of robotic platforms inside a real COVID treatment center is reported in [7]. In Wuhan, a ward was assisted by humanoid robots that delivered food to noncritical patients. Tommy [8] is a child-size, wheeled robot used to monitor patients in severe clinical conditions, measuring blood pressures and oxygen saturations when patients are on ventilators. Telepresence robots are becoming an effective tool for health-care operators treating highly infectious patients, such as those with COVID-19.

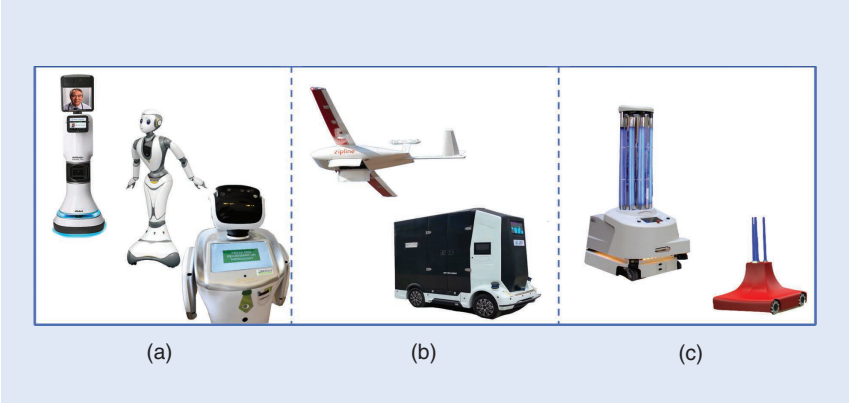


Figure 1. An overview of robotic platforms to assist with the COVID-19 response. (a) Patient care robots, including Vici [5], Ginger [6], and Tommy [8]. (b) Logistics robots, including the fixed-wing drone Zipling [9] and the Hercules autonomous vehicle [10]. (c) Disinfecting robots, including the one commercialized by UVD Robots [12] and the prototype equipped with ultraviolet-C lights in [13]. (Sources: [5]–[6], [8]–[10], [12]–[13]; used with permission.)

Automation and robotic solutions for delivery services and handling

contaminated material can also be found. Zipline is a company that tried to face the pandemic by providing drone delivery services to reduce human interaction in the supply chain, minimizing a vector for infection [9]. This approach addressed the issue of hospital overcrowding since nonurgent patients could receive care in local clinics that were closer to their homes. In particular, the fixed-wing delivery drone produced by Zipline provided blood to hospitals and clinics across Ghana, and an extension of the delivery service was planned in India and the United States. At the same time, some communities in Zibo, eastern China, received fresh vegetables that were transported and delivered by a robot. This autonomous vehicle, called Hercules [10], uses lidar, cameras, and deep learning algorithms to drive on streets, carrying up to 1,000 kg.

Robots are increasingly present inside hospitals and clinics to help health-care providers with their work activities [11]. Special attention has been paid to the disinfection of hospital environments by using ultraviolet (UV) light. Infected materials can play a central role in contamination by transmitting pathogens to people who touch them. UV-C radiation emits enough energy to destroy the deoxyribonucleic acid and ribonucleic acid of any microorganism exposed to the light. Mobile robots, both aerial and wheeled, are equipped with UV-C lights and steered through an environment to be sanitized [12], [13]. They scan the area using lidar and annotate points of interest (PoIs) on a map, and they can autonomously reach rooms and other environments to perform disinfection tasks.

This article describes a two-month experiment using a service robot, TIAGo (PAL Robotics, Barcelona, Spain), at the UCBM COVID treatment center, in Rome, Italy. The article analyzes the effectiveness and reliability of the proposed solution for logistics and disinfection. To the best of our knowledge, the UCBM platform is the first reported robotic solution to be introduced inside a COVID treatment center and to be capable of performing medical deliveries and disinfection while requiring only minor hardware modifications to switch from one task to another. The modular nature and ease of use of the proposed system make the robot versatile and effective for applications inside a real clinical environment. Nonexpert operators can easily interact and work with the robot to carry out fundamental activities. In addition, the ability to work autonomously enables the robotic system to successfully complete tasks under safe conditions for the health-care staff who collaborate with the device. Safety criteria are met by using obstacle-avoidance and dynamic path-planning algorithms.

Materials and Methods

The Collaborative UCBM COVID Robot

The UCBM COVID robotic platform relies on TIAGo [14], which appears in Figure 2(a). The robot has a head, with pan and tilt degrees of freedom (DoF), that supports a red–green–blue depth sensor (RGB-D) camera; an anthropomorphic arm with 7 DoF; a lifting torso; and a wheeled mobile base, whose footprint has a diameter of

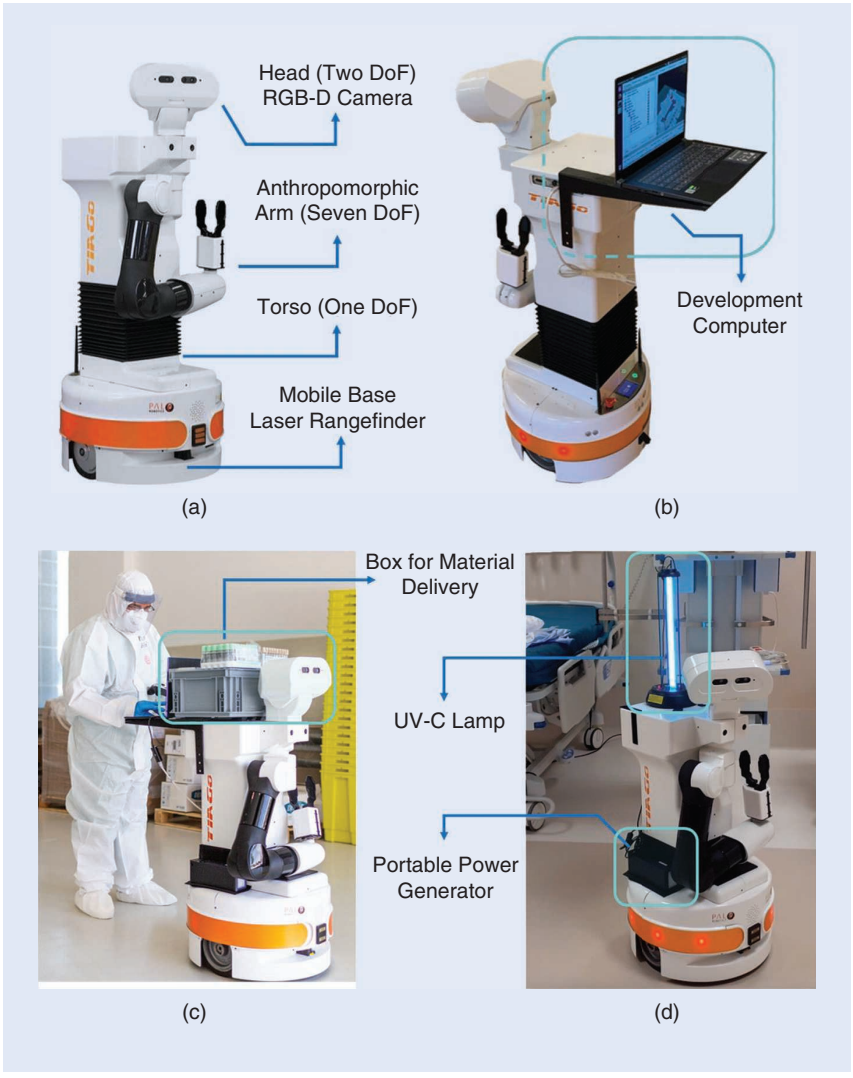


Figure 2. (a) TIAGo, the service robot adapted for this case study. (b) The development computer mounted on TIAGo by custom metal supports. (c) and (d) The robot inside the UCBM COVID treatment center. (c) The logistics scenario. (Source: Luigi Avantaggiato; used with permission.) (d) The disinfection scenario.

0.54 m. The robot is equipped with a frontal laser rangefinder and three rear sonars that are used to scan the environment. The rangefinder has a π-rad field of view and an angular step of 0.006 rad, with a visible range of 0.05–10 m, operating at a rate of 15 Hz. The robot autonomously navigates its environment, thanks to the embedded sensors. To address the two intended scenarios of logistics and disinfection with the same platform, a modular system was developed that could be adapted as needed:

- For the logistics scenario, the UCBM COVID robotic system was equipped with a box behind its head [Figure 2(c)] and was able to deliver materials weighing up to 5 kg.
- For disinfecting the hospital environment, it was equipped with a UV-C lamp [Figure 2(d)].

Fundamental TIAGo capabilities, e.g., chassis motion and autonomous navigation, were exploited to promptly address the pandemic through available technology. On the other hand, TIAGo is a sophisticated robotic system capable of interacting with human users in multiple modalities and performing reaching and grasping tasks. This opens opportunities for future developments in hospital scenarios (which we are working on), where all the robot's functionalities can be fully exploited.

Two ad hoc, easy-to-use web graphical user interfaces (GUIs) were developed in HTML to select commands for each of the scenarios (the GUIs are described in the "Scenario 1: Logistics" and "Scenario 2: Disinfection" sections). They ran on a laptop mounted to the back of TIAGo [Figure 2(b)]. The communication between the robot and the computer was established via an Ethernet port to avoid data packet losses and data slowdowns from wireless communication. The GUIs could be easily launched from the computer through any browser and operating system. Once a command was selected by a user, the robot autonomously accomplished the desired task. The energy autonomy of the UCBM COVID platform runs out in 4 h. Hence, the system needs to be recharged multiple times per day. The time required for a full recharge is approximately 2 h. The platform was equipped with a 23.4-Ah portable generator providing power (220 V) to the UV-C lamp. The external power supply is shown in Figure 2(d).

Localization, Mapping, and Path Planning

To perform autonomous navigation, the mobile robot is equipped with a simultaneous localization and mapping (SLAM) algorithm. It requires data from the exteroceptive sensors (i.e., lidars, sonars, and cameras) as well as odometry information provided by wheel encoders and inertial measurement units. The SLAM algorithms estimate the robot pose x_t inside known and unknown environments and build a map m_t. This is performed by collecting and fusing sensor measurements z_t and odometry readings u_t, with all these quantities indexed by a time step t.

The Gmapping [15] strategy was implemented to solve the SLAM problem. Such a planar technique relies on robot odometry as well as measurements coming from exteroceptive sensors (e.g., sonars and lasers) to estimate the robot's pose and the map where the robot is operating. The map is represented as a planar grid (the occupancy grid) consisting of a set of square cells, each having a probability value of the navigability of the cell itself. A Bayesian approach enables fusing pose transitions and laser scans in a probabilistic way. The algorithm solves the SLAM by factorizing the posterior probability with a Rao–Blackwellized particle filter (RBPF) [15]. Such a probability [16] is expressed as

$$p(x_t, m \mid z_t, u_t) = p(x_t \mid z_t, u_t)\, p(m_t \mid x_t, z_t), \qquad (1)$$

where a complexity reduction has been performed through the RBPF factorization. Here, $p(x_t \mid z_t, u_t)$ is about having pose x_t, given measurements z_t and odometry u_t, i.e., a pure localization problem; $p(m_t \mid x_t, z_t)$ represents the map m_t, given robot pose x_t and scan measurements z_t, i.e., a pure mapping problem.

The robot's pose x_t and the map of the environment m_t are stored inside a set of particles through a particle filter approach. The ith particle has the history of previous robot poses x_i, the map computed from such a history m_i, and a weight w_i. The weight is the probability that the particle represents both the pose and the map. Gmapping estimates this probability based on robot odometry and scan matching between consecutive laser scans and generates new particles. At the same time, the number of particles N is kept constant to reduce the computational burden. The complexity of the adopted approach is $O(N \times M)$, where M is the size of the generated grid map, as described in [15].

To address autonomous navigation, the robot is able to generate a reference trajectory through a path-planning module. This module is based on a Robot Operating System navigation package called *global_planner*. It needs the map to preplan the path to be followed and a local planner based on the global dynamic window approach, described in [17], that recalculates the trajectory according to the instantaneous local condition along the path. This is an obstacle-avoidance approach that evaluates the robot's kinematic and dynamic constraints and facilitates identifying the desired velocities. The desired velocities are chosen to maximize the alignment of the robot with the target and minimize the length of the trajectory in the absence of any obstacle. In this way, it is possible to combine path planning and obstacle avoidance to enable the robot to navigate safely in an unstructured environment. The navigation parameters adopted in this work are given in Table 1. They are used to plan the global path, adjust the local trajectory, and adapt the behavior of the platform in the presence of obstacles.

The Gmapping SLAM algorithm was chosen because of two features, i.e., the ease of implementation and

robustness. These make the system effectively cope with highly dynamic indoor environments, such as hospital wards, with few estimation errors and a low computational burden [18]. More sophisticated SLAM approaches can be found in the literature [19]; however, they suffer from drawbacks that make them unsuitable for this application. For example, visual SLAM methods may suffer from the invalidity of the static world assumption, which limits the application of RGB-D SLAM in dynamic environments [20].

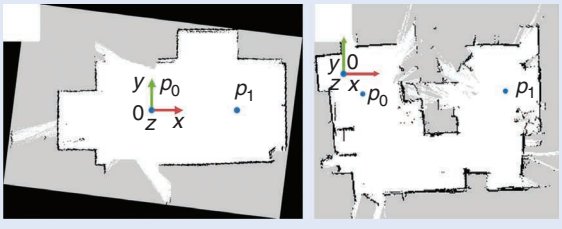
Preliminary Autonomous Navigation Evaluation

The autonomous navigation system was preliminarily validated in laboratory settings that reproduced the health-care ward. This phase facilitated measuring the system's performance and assessing the reliability and robustness of the navigation system, especially in the presence of static and dynamic obstacles. Two controlled environments were mapped, and the obstacle-avoidance robustness was tested. The first environment, shown in Figure 3(a), represents an open space where the robot can navigate autonomously. The latter, reported in Figure 3(b), is a more complex environment, consisting of a narrow passage between two areas. The robot could reach one area by crossing a 1.2-m-wide corridor resembling the COVID treatment center where the robot was intended to operate. The tests were conducted under three conditions: the first aimed at testing the repeatability of the performed trajectory in an undisturbed condition, the second assessed how the implemented autonomous navigation handled the presence of static obstacles, and the third included the presence of dynamic obstacles (e.g., a health-care provider crossing in front of the robot).

The robot was directed to autonomously navigate from the starting pose $\mathbf{X}_0$ toward the target point $\mathbf{X}_1$ and vice versa. The reference fixed frames are reported in Figure 3(a) and (b) for the two environments used for the platform testing. Position and orientation are expressed by means of Cartesian coordinates and quaternions, respectively. Poses $\mathbf{X}_0$ and $\mathbf{X}_1$ for each environment are presented in Table 2. The robot performed each navigation task 16 times in each proposed environment. To validate the algorithm in the presence of static obstacles, a box with dimensions of $0.3 \times 0.21 \times 0.5$ m was placed in p_{obstacle}, expressed in the reference frame $O - xyz$, specifically, $p^{E1}_{\text{obstacle}} = [1.5, 0, 0]^T$ m in environment E1 and $p^{E2}_{\text{obstacle}} = [1.45, 2.25, 0]^T$ m in E2. The robot was asked to

Table 1. The navigation parameters implemented on the UCBM COVID robotic platform.

Global Planner Parameters	Value
Update frequency of the global planned path	1 Hz
Neutral cost	50
Cost factor	3
Lethal cost	253
Tolerance at the goal point	0.1 m
Local Planner Parameters	**Value**
Time to forward-simulate trajectories	1.7 s
Update frequency of the local planned path	1 Hz
Controller frequency	10 Hz
Explored samples in the *x* velocity space	10
Explored samples in the *y* velocity space	0
Explored samples in the theta velocity space	20
Step size to take between points on a given trajectory	0.025 m
Weight for how much the controller should stay close to the path	32
Weight for how much the controller should attempt to reach its local goal	24
Weight for how much the controller should attempt to avoid obstacles	0.01
Tolerance in reaching goal position	0.2 m
Tolerance in reaching goal orientation	0.2 rad
Maximum transnational velocity	1.5 m/s
Maximum rotational velocity	2 rad/s
Cost Map Parameters	**Value**
Global and local cost map update frequency	10 Hz
Local cost map obstacle range	3.5 m
Local cost map ray trace range	4 m
Local cost map width and height	5 m
Local cost map resolution	0.025 m



(a) (b)

Figure 3. Maps built by the robot in the laboratory setting. The fixed reference frame $O - xyz$ and the target points p_0 and p_1 used to validate the robot's navigation capability are highlighted for the two tested environments. (a) The open-space scenario. (b) The more challenging indoor scenario.

Table 2. The target poses used in the navigation capability test phase.

	Pose	Position (m)	Orientation ( )
E1	$\mathbf{X}_0$	$p_0 = [0, 0, 0]^T$	$q_0 = [0, 0, 0, 1]^T$
	$\mathbf{X}_1$	$p_1 = [2.5, 0, 0]^T$	$q_1 = [0, 0, 0, 1]^T$
E2	$\mathbf{X}_0$	$p_0 = [0.53, -0.42, 0]^T$	$q_0 = [0, 0, -0.70, 0.71]^T$
	$\mathbf{X}_1$	$p_1 = [3.92, -0.42, 0]^T$	$q_1 = [0, 0, -0.77, 0.64]^T$

repeat the same navigation task 16 times in each environment. Finally, for validation with dynamic obstacles, a volunteer was invited to cross the path of the robot when it was 1 m from the starting point. Here, too, the robot traveled 16 times in each environment.

The robot's navigation capabilities were measured according to the following performance indicators:

- *Completion time (CT)*: This is the time needed to reach the end position from the starting point, and it facilitates estimating the robot's autonomous navigation replanning capability.
- *Path length (PL)*: This is the effective distance traveled by the robot and provides a measure of how far along the path the robot traveled during different trials. It is computed as

$$PL = \int_{t_0}^{t_f} \dot{p}(t)\, dt, \tag{2}$$

where $\dot{p}$ is the robot's linear speed and t_0, t_f, and t are the initial, final, and current time of each navigation task, respectively.

- *Distance error (DE)*: This is the error made in reaching the assigned target position at the end of the movement. It provides a measure of the accuracy in reaching the assigned target. It is computed as

$$DE = \| p_{\text{goal}} - p(t_f) \|, \tag{3}$$

where p_{goal} is one of the aforementioned goal positions $\{p_0, p_1\}$ and $p(t_f)$ is the position of the robot at the end of the movement.

- *Success rate (SR)*: This is defined as

$$SR = \frac{N_{\text{succ}}}{N_{\text{tot}}} \cdot 100, \tag{4}$$

where N_{succ} is the number of completed navigation tasks satisfying both the constraints on the DE, i.e., $DE \leq 0.1$ m, and the absence of collisions and N_{tot} is the total number of performed motions ($N_{\text{tot}} = 16$).

Scenario 1: Logistics

A crucial aspect of health-care providers' activities within a COVID treatment center involves continuous interaction with patients. In addition, health-care providers have to move around the hospital to deliver drugs and medical devices and to collect samples intended for the laboratory. Moreover, practitioners working inside intensive care units (ICUs) cannot leave because they must monitor their patients. This limits their ability to move from one environment to another. The introduction of a robot to transport useful materials can parallelize activities, help the caregivers, and optimize logistics.

From this perspective, for logistics purposes, a box was mounted on the robot's laptop tray and loaded with several materials—drugs, blood products, and small instruments—to be transported along the hallway of the ward. The robot had to navigate autonomously from one starting point to a target one, selected by an operator who could choose the target through the GUI. The robot planned the shortest path to reach the target and started navigating. The target areas identified inside the UCBM COVID treatment center, highlighted in Figure 4, are explained briefly in the following:

- *Charging station*: This was a very important point because the robot needed to be recharged after every 4 h of full working. This system automatically selected the target point whenever the battery level was lower than 15% to avoid having the robot stop in the hallway. The charging station was also the starting point for the mapping phase; thus, it was the origin of the global fixed reference frame.
- *Hot cell*: This was the environment where ambulances accessed the hospital to drop off incoming patients, as reported in Figure 4. Patients were admitted in this zone to avoid contact with other environments and operators.
- *Red area*: This included the ICU, which held a maximum of nine beds for critical patients. Each bed was provided with a monitoring station and a mechanical ventilator.
- *Intensive short observation area*: This was a sub-ICU equipped with five bed stations.
- *Holding area*: This area contained a medical ward with 26 beds.

The GUI is described in Figure 5. It enabled an operator to check for a successful connection with the robot

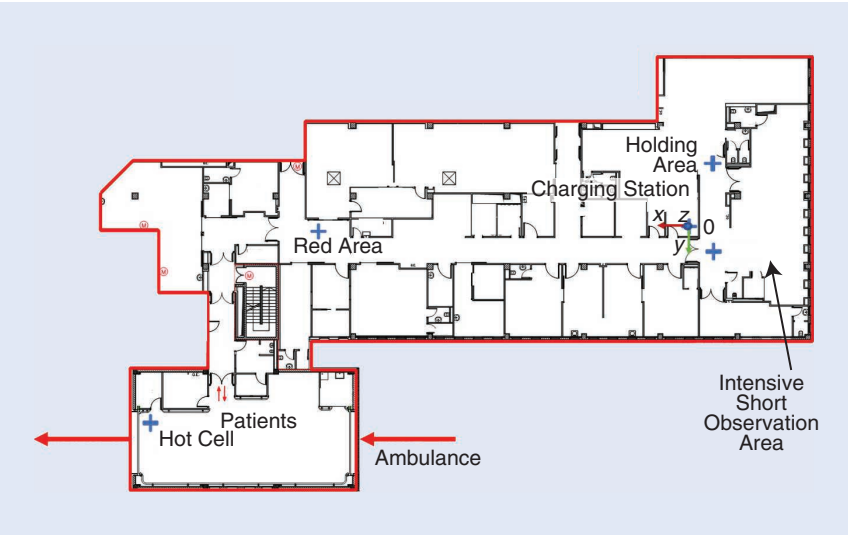


Figure 4. The UCBM COVID treatment center floor plan. Blue crosses represent target points that the robot can autonomously reach. The charging station is the origin of the fixed reference frame $O - xyz$ used for the navigation algorithm.

and to select one of the aforementioned target locations. After one of those buttons was pressed, the robot started its autonomous navigation toward the assigned target. Once the robot reached the desired position, a voice message, reproduced by the robot, informed the operator about its arrival and the delivered materials. The robot performed its logistics activity during the day, transporting drugs, tubes, blood components, and small medical devices from the hot cell toward the red area, intensive short observation area, and holding area and vice versa. Figure 2(c), shows a health-care operator interfacing with the GUI. The robot's impact on logistics tasks was assessed through a set of indicators: the day of use, mean time spent delivering materials in a day, number of recharges required per day, average traveled distance in a day, and number of health-care personnel required to use the robot.

Scenario 2: Disinfection

In the second implemented scenario, displayed in Figure 2(d), the robot was equipped with a UV-C lamp to sanitize crucial environments, such as isolation rooms and areas dedicated to positive patients, i.e., the main hallway. Usually, certain health-care workers have the task of sanitizing hospital environments, including floors and surfaces, multiple times per day, using alcohol- and sodium hypochlorite-based solutions. However, this approach requires maximum human attention and precision, consuming time and requiring adequate staff.

For disinfection, a lamp emitting UV light at certain wavelengths (i.e., in the range of 200–280 nm) can be used. Such a radiant light has germicidal properties and can inactivate viral agents. The OnReal UV-C lamp was used; the mounting is depicted in Figure 2(d). This lamp was 0.49 m tall and had a UV quartz tube powered by the main voltage, i.e., the 220 V supplied by the portable generator. Such a device can output 36 W, efficiently treating 10 m^2 every 15 min, with a maximum operating time of 60 min, after which the lamp automatically turns off. Given the dimensions of the ward hallway, the disinfection procedure was divided into two consecutive paths—orange and blue—that received 2 h of cleaning at night. In particular, the hallway was divided into five areas, each identified as a PoI. As shown in Figure 6, during the orange and blue disinfection procedures, the robot reached and stopped at PoIs $\{1a, 2a, 3a\}$ and $\{1b, 2b\}$, respectively. By dividing the activity into two sequential steps, it was possible to treat the corridor from the red area to the holding area.

The orange-area disinfection required the robot to stop for at least 20 min at each PoI. The robot began at the charging station and reached three PoIs identified on the map. Using a remote control, which functioned at a safety distance of 20 m, a human operator turned on the lamp when the robot stopped at the first PoI ($1a$). As an additional safety measure, the lamp did not begin disinfecting for 10 s after it was turned on, enabling the user to leave the corridor. At the end of the orange-area disinfection, the lamp turned off, and the robot returned to the charging station. Afterward, the blue-area disinfection could be launched. The lamp was turned on again, and the robot ran through the two PoIs of the blue disinfection path. At PoIs $\{1b, 2b\}$, the robot stayed for 30 min until the procedure was completed.

Health-care operators could boot the disinfection procedure by using the second implemented GUI, reported in Figure 6. This interface showed the state of the connection with the robot and enabled starting the disinfection procedure by means of dedicated buttons. To better distinguish the two consecutive phases, orange and blue were used to highlight the stations where the robot stopped during each treatment. A picture of the environment to be sanitized was shown in the GUI to ease the selection of the desired path. The first disinfection phase was marked with orange dots and the second one with blue dots. In addition, these points were numbered to indicate the order in which the robot would travel to the stations. As for logistics, certain measures were computed to quantify the robot's impact during the disinfection scenario: the

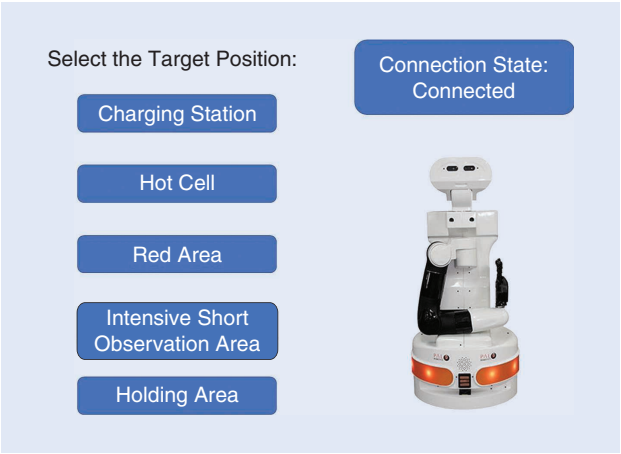


Figure 5. The GUI developed for the logistics scenario. It enables operators to check for the correct connection between the development computer and the robot and to select the target position to be reached.

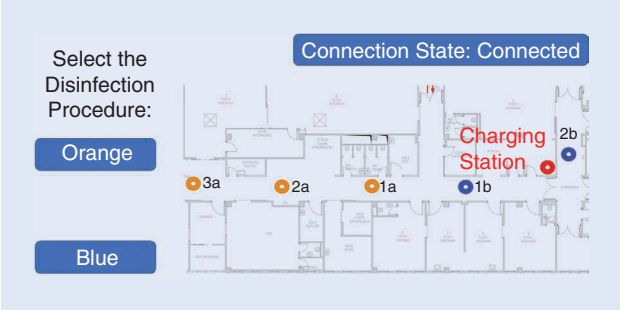


Figure 6. The GUI developed for the disinfection scenario. It enables operators to check for the correct connection between the development computer and the robot and to select a disinfection procedure.

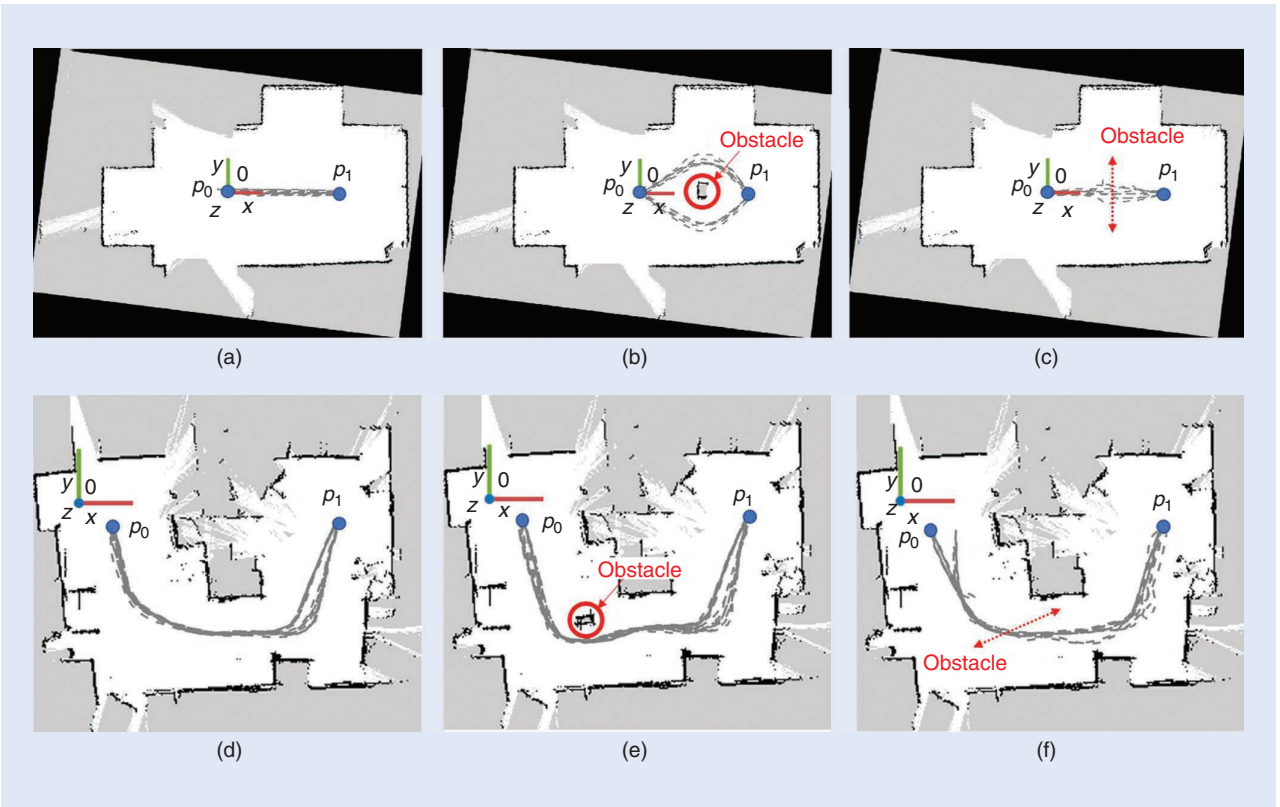


Figure 7. The preliminary tests of the autonomous navigation performed in laboratory settings. In (a)–(c) and (d)–(f), gray dashed lines indicate the trajectories the robot followed in environments E1 and E2, respectively. In (a) and (d), the robot moved toward the goal point without facing any obstacle. In (b) and (e), the robot avoided a fixed obstacle, circled in red. In (c) and (f), a person crossed the robot's path while it was navigating.

number of procedures performed, mean time spent disinfecting the corridor, average traveled distance during the procedure, and number of personnel required to boot the disinfection scenario.

Results and Discussion

Experimental results showing the performance of the autonomous navigation algorithm in the laboratory settings are reported in Figure 7(a)–(c) and Figure 7(d)–(f) for environments E1 and E2, respectively. The figure includes the maps built with the laser rangefinder, the fixed reference frames, and the trajectories executed by the robot in all the experimental conditions. Three different conditions were tested: the movement of the robot without obstacles in its path [Figure 7(a) and (d)], the avoidance of a static obstacle [Figure 7(b) and (e)], and the avoidance of a dynamic obstacle [Figure 7(c) and (f)]. As expected, the robot deviated from the unperturbed path once obstacles were introduced. Quantitative metrics, defined in the "Preliminary Autonomous Navigation Evaluation" section, appear in Table 3.

Concerning the first environment (E1), the linear path the robot followed [Figure 3(a)] constitutes the simplest condition, and it can be used as ground truth to assess the other two situations. This experimental condition is in Figure 7(a). The mean CT required to reach the target position was 6.41 ± 1.04 s, and the traveled path was 2.58 ± 0.13 m. The error at the end of the movement was 0.05 ± 0.03 m. The introduction of a static obstacle led the robot to deviate from the straight line and plan an alternative path. This is highlighted in Figure 7(b). The robot took more time to reach the target—more than 10 s—and traveled a longer path (3.83 ± 0.47 m). The DE was 0.04 ± 0.05 m. In the third condition, a human subject crossed the ward while the robot was navigating. The robot stopped until the dynamic obstacle disappeared. This resulted in an increase in the required time to reach the target, keeping the PL comparable with the first case.

Table 3. The robot's quantitative performance in a laboratory setting.

	CT (s)	PL (m)	DE (m)	SR (%)
Without obstacle E1	6.41 ± 1.04	2.58 ± 0.13	0.05 ± 0.03	100
Static obstacle E1	10.81 ± 1.74	3.83 ± 0.47	0.04 ± 0.05	93.75
Dynamic obstacle E1	11.2 ± 2.51	2.82 ± 0.77	0.05 ± 0.06	68.75
Without obstacle E2	21.32 ± 1.02	7.79 ± 0.19	0.04 ± 0.02	100
Static obstacle E2	25.56 ± 2	8.35 ± 0.1	0.05 ± 0.02	100
Dynamic obstacle E2	39.06 ± 14.64	8.68 ± 2.24	0.06 ± 0.03	87.5

The deviation with respect to the linear path was very small and can be seen in Figure 7(c).

The second scenario, illustrated in Figure 3(b), was also used to test the robot's navigation capability. The robot's paths are reported in Figure 7(d)–(f). In the unperturbed condition, the robot was able to reach the target position in 21.32 ± 1.02 s along a path 7.79 ± 0.19 m long. The target was reached in all the trials with a DE of 0.04 ± 0.02 m. The introduction of the static obstacle led the robot to take a longer path to reach the assigned position. The CT consequently increased. The presence of the static obstacle did not affect the capability of the robot to reach the target, with a small residual error. In fact, the resulting DE was 0.05 ± 0.02 m. During the third condition, i.e., a person crossing the robot's path, the robot's replanning capabilities were assessed. As expected, the robot stopped when a dynamic obstacle appeared in front of the mobile base. Therefore, the mean CT was much greater than those measured in the other experimental conditions (39.06 ± 14.64 s). The PL computed in this case was 8.68 ± 2.24 m. The high variability shown in these trials results from the fact that the robot, in reaching the target in position p_1, searched for different paths when the dynamic obstacle obstructed the corridor, thus lengthening the PL. On the other hand, the robot was capable of reaching the assigned target with a small DE (0.06 ± 0.03 m).

All the tasks performed by the robot obtained a high SR. The robot was always able to avoid collisions with the obstacles and to reach the assigned target position with a small DE. The SR was less than 100% due to some DEs greater than 0.1 m that occurred in a few trials, i.e., 8%, corresponding to eight of the 96 trials. The results demonstrate that the proposed system was able to complete the required tasks by safely managing (both for the robot and for the user) unexpected events, such as the presence of static and dynamic obstacles.

After this validation in a laboratory setting, the robot was installed at the UCBM COVID center at the beginning of May 2020. The main results collected during two months of use are reported and discussed in this section. Figure 8(a) conveys the map generated by the robot during the mapping phase. The red dots represent the PoIs listed in the previous section, while the blue dashed lines are the trajectories the robot executed during a daily logistics function. Quantitative results for the logistics scenario are summarized in Table 4. Mean values, computed during the total number of days of use, are reported with their standard deviation.

The robot worked in the logistics scenario for 43 days, for roughly 7 h per day, from 8 a.m. to approximately 8 p.m. During the daily activities, at least two recharge phases were required. The average traveled distance inside the ward, where the robot transported drugs, tubes, blood components, and medical devices, was roughly 8,000 m per day. The number of human operators required for the logistics scenario was three. The operators were inside the hot cell, the red area, and the intensive short observation and holding areas. Each user filled and emptied the robot's cargo box according to their needs and determined the robot's target location. In this way, the operators did not have to leave their work station while the robot transported materials.

Figure 8(b) provides a UCBM COVID treatment center map with the trajectories executed by the robot during a disinfection procedure. As explained in the previous

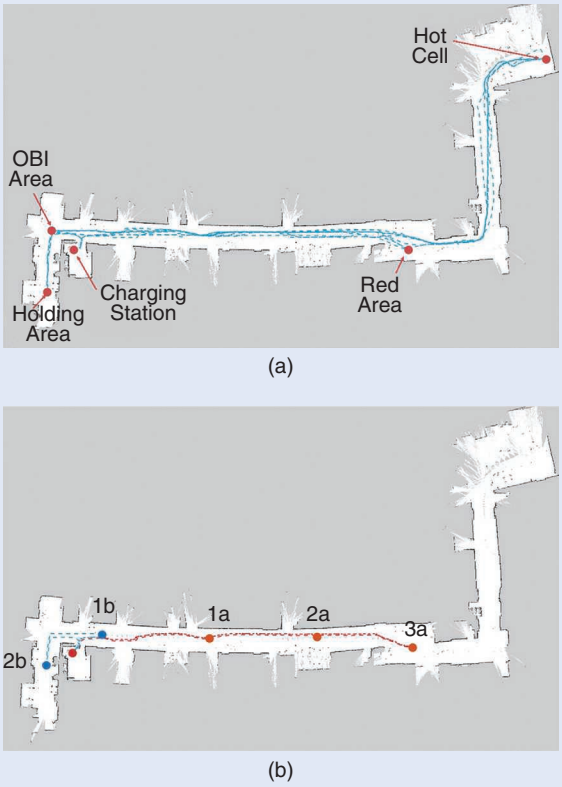


Figure 8. A map of the UCBM COVID treatment center that the robot built. In (a), red points represent the targets the robot had to reach for the logistics scenario. The dashed blue lines indicate trajectories the robot performed during a day. In (b), the dots represent targets the robot had to reach for the disinfection scenario. The two dotted lines, in orange and blue, represent the robot's paths to treat the entire hallway. In particular, the points where the robot stopped to treat the environment are highlighted with large dots.

Table 4. The results of the logistics scenario.

Scenario 1: Logistics	
Days of use	43 days
Mean time delivering materials	437.7 ± 28.2 min
Number of recharges per day	2.13 ± 0.74
Mean traveled distance	7,971.2 ± 44.1 m
Number of health-care operators	3

Usage indexes of the proposed robotic platform are reported as the mean value and the standard deviation.

section, the disinfection scenario was composed of two sequential treatments. The first was composed of three PoIs, shown in Figure 8(b) and represented by an "a" marked by orange dots. The latter had only two PoIs, which are indicated by a "b" marked by blue dots. The increasing number represents the order in which the robot traveled to the stations. The executed trajectories are reported with orange and blue dashed lines, respectively, and the disinfection stations are indicated by large orange and blue dots. From Figure 8(b), it is possible to observe that the robot reached all the PoIs and, at the end of each procedure, returned to the charging station.

The metrics computed for the disinfection scenario are presented in Table 5. For the scenario, the estimated working time was roughly 2 h. The disinfection of the corridor was carried out at night, generally from 2 a.m. to 4 a.m. for safety reasons, and it was performed 35 times in two months. During the procedure, the robot traveled 85 m for the first phase and 30 m for the second one. The disinfection scenario was easier to be booted: only one health-care operator was involved. He or she had only to select the desired disinfection procedure and turn on the UV-C light. At the end of each phase, the robot returned to the charging station.

The presence of the robot had significant psychological benefits for patients forced to stay in the treatment center and away from their families.

Introducing the robot into highly biohazardous environments had multiple advantages in both scenarios. First, in the logistics scenario, the robot reduced the number of movements required by medical personnel among different environments. This led to a reduction in the biological risk to which the operators were subjected. In addition, the presence of the robot had significant psychological benefits for patients forced to stay in the treatment center and away from their families. The robot did not aim to replace the warmth and moral support of health-care professionals but enhanced their role and increased their benefits.

Table 5. The results of the disinfection scenario.

Scenario 2: Disinfection	
Number of performed procedures	35
Mean time disinfecting	125.6 ± 3.4 min
Mean traveled distance	116.4 ± 7.2 m
Number of health-care operators	1

Usage indexes of the proposed robotic platform are reported as the mean value and the standard deviation.

Despite this, using the robot had uncovered weaknesses. The platform runs on battery power and needs to be periodically recharged. Medical personnel must be adequately trained to properly work with the robot, which is not easy during a global pandemic. Finally, it is better to define the map before using the navigation algorithms. So, during the first installation, expert robotics users have to go through all relevant areas with the robot to record the map and manage the navigation and obstacle-avoidance algorithms.

Conclusions

In this article, a two-month experiment studying the use of the UCBM COVID robotic system was described. The robot was used to assist health-care operators during their daily activities, in particular, for logistics and disinfection procedures. To accomplish such tasks, the robot had autonomous navigation capability. The robot was equipped with the Gmapping SLAM algorithm to provide robustness against a dynamically changing environment and facilitate the ease of implementation. A list of the parameters used to implement the navigation capabilities was provided as a guideline for further developments of autonomous robots for similar scenarios.

Preliminary tests conducted in laboratory settings were carried out to assess the robot's autonomous navigation capabilities. The good results obtained in a controlled environment enabled the robot to access the UCBM COVID treatment center. To address two scenarios with the same platform, a modular system was developed: the robot could be equipped with a box for delivering materials and with a UV-C lamp for disinfection. Furthermore, two web GUIs were developed so that users could easily and intuitively select commands for each of the two scenarios.

The introduction of the robotic system facilitated the parallelization of material transport inside the ward, and the proven effectiveness and ease of use led operators to exploit the robot throughout the work day. Users did not have to be expert to interact with the robot, thanks to the GUIs. During the night shift, the robot was exploited for disinfection purposes. The absence of health-care workers in the corridors made it possible to carry out this activity under safe conditions. The automation of the procedure enabled treating the entire area of interest without human intervention.

Future efforts will aim at increasing safety during the disinfection procedure and improving the interaction modalities between the health-care operators and the robot. In particular, software and hardware improvements will be implemented to automatically detect people entering an area that is being disinfected and turn off the UV-C lamp. Moreover, voice commands and control are being developed to improve the interaction modalities and communication between robot and health-care personnel.

Acknowledgments

This work was supported, in part, by Campus Bio-Medico University of Rome Strategic Projects (call 2018) with the SAFE-MOVER project; Regione Lazio with the HeAL9000 project (grant B84I20001880002); the Italian Institute for Labor Accidents, with the PPR AS 1/3 (grant E57B16000160005) and the SENSE-RISC project (grant B56C18004200005). Felice Eugenio Agro and Loredana Zollo contributed equally to this work.

References

[1] G. Pascarella et al., "COVID-19 diagnosis and management: A comprehensive review," *J. Internal Med.*, vol. 288, no. 2, pp. 192–206, 2020. doi: 10.1111/joim.13091.

[2] G.-Z. Yang et al., "Combating COVID-19—The role of robotics in managing public health and infectious diseases," *Sci. Robot.*, vol. 5, no. 40, p. eabb5589, 2020. doi: 10.1126/scirobotics.abb5589.

[3] C. Eccleston et al., "Managing patients with chronic pain during the COVID-19 outbreak: Considerations for the rapid introduction of remotely supported (ehealth) pain management services," *Pain*, vol. 161, no. 5, pp. 889–893, 2020. doi: 10.1097/j.pain.0000000000001885.

[4] L. Zollo, C. Laschi, G. Teti, B. Siciliano, and P. Dario, "Functional compliance in the control of a personal robot," in *Proc. 2001 IEEE/RSJ Int. Conf. Intell. Robots Syst., Expanding Societal Role Robot. Next Millennium (Cat. No. 01CH37180)*, vol. 4, pp. 2221–2226. doi: 10.1109/IROS.2001.976400.

[5] A. Matyus. "Meet the robot helping doctors treat coronavirus patients." Digital Trends. https://www.digitaltrends.com/news/meet-the-robot-helping-doctors-treat-coronavirus-patients/ (accessed Oct. 7, 2020).

[6] C. Clifford, "Look inside the hospital in china where coronavirus patients were treated by robots," CNBC, Englewood Cliffs, NJ. Accessed: Oct. 7, 2020. [Online]. Available: https://www.cnbc.com/2020/03/23/video-hospital-in-china-where-covid-19-patients-treated-by-robots.html

[7] S. O'Meara, "Hospital ward run by robots to spare staff from catching virus," *New Sci. (1971)*, vol. 245, no. 3273, p. 11, 2020. doi: 10.1016/S0262-4079(20)30526-1.

[8] M. E. Romero. "Tommy the robot nurse helps Italian doctors care for COVID-19 patients." The World. https://www.pri.org/stories/2020-04-08/tommy-robot-nurse-helps-italian-doctors-care-covid-19-patients (accessed Oct. 7, 2020).

[9] E. Ackerman. "Zipline wants to bring medical drone delivery to U.S. to fight COVID-19." IEEE Spectrum. https://spectrum.ieee.org/automaton/robotics/drones/zipline-medical-drone-delivery-covid19 (accessed Oct. 7, 2020).

[10] E. Guizzo. "Robot vehicles make contactless deliveries amid coronavirus quarantine." IEEE Spectrum. https://spectrum.ieee.org/automaton/transportation/self-driving/robot-vehicles-make-contactless-deliveries-amid-coronavirus-quarantine (accessed Oct. 7, 2020).

[11] C. Lauretti, F. Cordella, E. Guglielmelli, and L. Zollo, "Learning by demonstration for planning activities of daily living in rehabilitation and assistive robotics," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1375–1382, 2017. doi: 10.1109/LRA.2017.2669369.

[12] E. Ackerman. "Autonomous robots are helping kill coronavirus in hospitals." IEEE Spectrum. https://spectrum.ieee.org/automaton/robotics/medical-robots/autonomous-robots-are-helping-kill-coronavirus-in-hospitals (accessed Oct. 7, 2020).

[13] S. S. of Advanced Studies Press Office. "Un robot mobile per la disinfezione di ambienti sanitari, da settembre a Pisa." gonews.it. https://www.gonews.it/2020/04/28/un-robot-mobile-per-la-disinfezione-di-ambienti-sanitari-da-settembre-a-pisa/ (accessed Oct. 7, 2020).

[14] J. Pages, L. Marchionni, and F. Ferro, "Tiago: The modular robot that adapts to different research needs," in *Proc. Int. Workshop Robot Modularity, IROS*, 2016, pp. 1–4.

[15] G. Grisetti et al., "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Trans. Robot.*, vol. 23, no. 1, pp. 34–46, 2007. doi: 10.1109/TRO.2006.889486.

[16] S. Thrun et al., *Probabilistic Robotics* (Intelligent Robotics and Autonomous Agents Series). Cambridge, MA: MIT Press, 2005.

[17] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Proc. 1999 IEEE Int. Conf. Robot. Autom.*, vol. 1, pp. 341–346. doi: 10.1109/ROBOT.1999.770002.

[18] J. M. Santos, D. Portugal, and R. P. Rocha, "An evaluation of 2D SLAM techniques available in robot operating system," in *Proc. 2013 IEEE Int. Symp. Safety, Security, Rescue Robot. (SSRR)*, pp. 1–6. doi: 10.1109/SSRR.2013.6719348.

[19] C. Cadena et al., "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, 2016. doi: 10.1109/TRO.2016.2624754.

[20] Y. Sun, M. Liu, and M. Q.-H. Heng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robot. Auton. Syst.*, vol. 89, pp. 110–122, Mar. 2017. doi: 10.1016/j.robot.2016.11.012.

Christian Tamantini, Research Unit of Advanced Robotics and Human-Centred Technologies, Università Campus Bio-Medico di Roma, Rome, 00128, Italy. Email: c.tamantini@unicampus.it.

Francesco Scotto di Luzio, Research Unit of Advanced Robotics and Human-Centred Technologies, Università Campus Bio-Medico di Roma, Rome, 00128, Italy. Email: f.scottodiluzio@unicampus.it.

Francesca Cordella, Research Unit of Advanced Robotics and Human-Centred Technologies, Università Campus Bio-Medico di Roma, Rome, 00128, Italy. Email: f.cordella@unicampus.it.

Giuseppe Pascarella, Unit of Anaesthesia, Intensive Care, and Pain Management, Department of Medicine, Università Campus Bio-Medico di Roma, Rome, 00128, Italy. Email: g.pascarella@unicampus.it.

Felice Eugenio Agrò, Unit of Anaesthesia, Intensive Care, and Pain Management, Department of Medicine, Università Campus Bio-Medico di Roma, Rome, 00128, Italy. Email: f.agro@unicampus.it.

Loredana Zollo, Research Unit of Advanced Robotics and Human-Centred Technologies, Università Campus Bio-Medico di Roma, Rome, 00128, Italy. Email: l.zollo@unicampus.it.