COVID-Bot: UV-C Based Autonomous Sanitizing Robotic Platform for COVID-19

Edgar C. Camacho * Nestor I. Ospina ** Juan M. Calderón ***

* Faculty of Electronic Engineering, Universidad Santo Tomás, Bogotá, Colombia (e-mail: edgarcamacho@usantotomas.edu.co)
** Department of Electrical and Electronic, Universidad Nacional de Colombia, Bogotá, Colombia (e-mail: niospinag@unal.edu.co)
*** Bethune-Cookman University, Daytona Beach, FL, USA (e-mail: calderonj@cookman.edu), Universidad Santo Tomás, Bogotá, Colombia (e-mail: juancalderon@usantotomas.edu.co)

Abstract:
This paper presents the design and implementation of COVID-Bot, an open-source robotic platform for sanitizing single plant environments such as offices, houses, apartments, among others. This development seeks to create a tool that contributes to the global fight against the COVID-19 pandemic, from a low-cost and easy-to-replicate robot, which disinfects surfaces through type C ultraviolet radiation. The platform is based on a differential robotic base, an RGB-D camera, a tracking camera, three UV-C lamps, and an embedded computer running the ROS-based control software. In addition, this paper presents the description of the hardware used, the software implemented, and the tests carried out to corroborate the operation of the integrated system. These tests demonstrated that the system is adequate to autonomously cover a one-floor apartment, based on the theoretical radiation distance of the used lamps.

Keywords: Mechatronic Systems and Robotics, Cost Oriented Automation (COA), Biomedical systems, Mobile Robots, 3D Mapping, Navigation, ROS.

1. INTRODUCTION

In December of 2019, the first case of a virus called Covid-19 was reported in Wuhan, the capital of the Chinese province called Hubei Sheng (Balkhair, 2020). After that, the virus quickly spread in Asia, Europe, America, and the rest of the world. After the rapid spread of the virus worldwide, the sanitary emergency was decreed by the World Health Organization, raising the COVID-19 label to the pandemic level in March of 2020 (Spinelli and Pellino, 2020). The virus is a disease caused by SARS-CoV-2 that can trigger a respiratory tract infection. It can affect the upper or the lower respiratory tract with main symptoms including fever, coughing, shortness of breath, trouble breathing, fatigue, Chills, sometimes shaking, body aches, headache, Sore throat, Congestion, runny nose, loss of smell or taste, nausea, and diarrhea (Struyf et al., 2020).

The virus can lead to pneumonia, respiratory failure, heart problems, liver problems, septic shock, and death. Many complications of COVID-19 can be caused by a condition known as cytokine release syndrome. It is when an infection causes the immune system to flood the bloodstream with inflammatory proteins called cytokines. These proteins can kill tissues and damage organs. In response to this virus, some governments decreed total and partial closures of commerce to avoid people crowds and decree social distancing, self-isolation, and travel restrictions to prevent the virus from spreading from person to person. The decisions taken by governments affected the economic

sector by reducing the workforce and caused many lost jobs as depicted by Bai et al. (2020). On the other hand, the education sector closed schools and migrated to online platforms keeping students at home and attending virtual classes (Daniel, 2020). The pandemic affected the socio-economic, education, finance, and health sectors, stopping the world from romping up.

2. RELATED WORK

Once the pandemic was decreed and the side effects of a prolonged quarantine were estimated, the world's scientific society focused on establishing act priorities following global needs. The scientific community faced different areas, such as studying and analyzing the virus for possible vaccine generation. In addition, the virus behavior study aims to create norms that mitigate its spread, relevant virus information dissemination, and improvement of self-care protocols. Likewise, the robotic scientific community spoke out, indicating four different areas where robotics can make a difference: Clinical care, logistics, reconnaissance, and work continuity with socioeconomic maintenance functions (Yang et al., 2020). During the Pandemic development, many robotics applications emerged given the interaction of the medical personnel, patients, civil people, and economy. Some of the areas where robots have been more frequently used are public safety, clinical care, continuity of work and education, quality of life, laboratory and supply chain automation, and non-hospital care (Murphy et al., 2020).

Given these application areas where robotics can help counteract the Covid-19 effects, the two main areas that have presented the most relevant development are Public Safety and Clinical Care. Public Safety has presented the most significant number of robotic applications around the world. The Public Safety application is based on the use of mobile robots in public areas. Those robots collaborate with the authorities to maintain social distancing through audible communication with general people (Chen et al., 2021). These robots are also used to help in body temperature sensing activities, although this activity is a bit controversial due to the inaccuracy of the measurement system. Finally, mobile robots for public safety are even used to sanitize open areas through the sprinkling processes. Both aerial and terrestrial robots have been used in this application area (Somaldo et al., 2020), although mainly terrestrial. The second most common application area has been Clinical Care, where robots are used mainly for telepresence tasks (Isabet et al., 2021), transport of medicines and meals in hospital facilities, patient care, surveillance, and sanitizing hospital areas. In order to carry out this type of application, it is necessary to have access to an excellent wireless internet network, adequate spaces to carry out these tasks. Additionally, these mobile robots need navigation systems that allow the robot to move from one place to another specific place without colliding obstacles. Sanitizing tasks in hospital areas are carried out mainly through the use of UVC lamps and with navigation systems based on sensors with a good autonomy range. This paper proposes the design of a sanitizing mobile robot for indoor areas affected by Covid-19. The sanitizing task is performed through the use of UVC lamps.

We present the design and implementation of COVID-Bot for sanitizing indoor environments as houses, apartments, and offices, among others. With this development, we are presenting four different contributions: a low-cost design of a multi-application mobile robot, a totally replicable robot design and development, the integration of a low-cost navigation system based on commercial visual sensors and locomotion base, and a sanitizing UV-C based system for use in multiple indoor places.

3. ROBOTIC PLATFORM DESIGN

The designed robotic platform can be represented by the block diagram presented in figure 1. An embedded computer manages the operation of the complete system. It receives visual information and localization awareness from two Intel Realsense Cameras: one D435 and one T265, through a USB 3.1 interface. It also sends movement commands to an iRobot Create two robotic bases via USB 2.0. In addition, it generates activation signals for the UV-C radiation system.

3.1 Multi-levels platform

The mechanical structure could be based on acrylic, 3D printing, wood, or aluminum platform levels fixed to the mobile base through aluminum spacers. The platforms' material depends on the design requirements' durability, performance, weight, and low-cost development. Each level has enough space to install the components that make
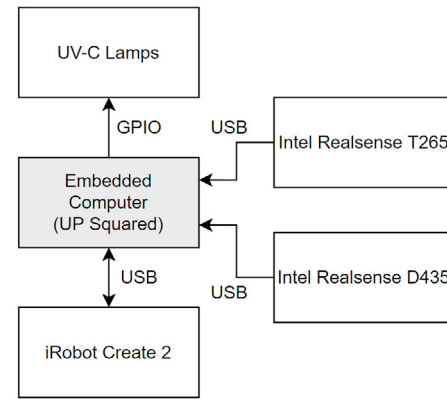

Fig. 1. General block diagram.

up the general system. The first level corresponds to the mobile base. The second level contains the power supply components (batteries, regulators, etc.). The third level includes the central computing unit and the RGB-D camera. The fourth level holds the tracking camera and the AC-DC inverter. The last level is used as the support of the UV-C lamps. This distribution is shown in figure 2.
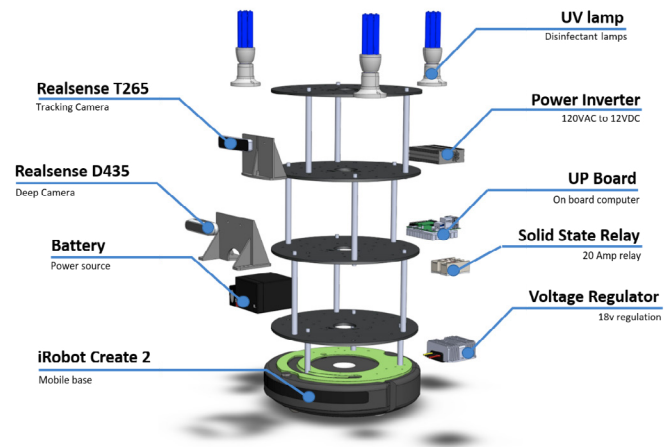

Fig. 2. Components distribution.

Places like offices, houses, and apartments are considered target environments for the operation of this robot. Starting from this requirement, a robot dimension was selected that facilitated navigation across doors, halls, rooms, etc. General robot dimensions are presented in figure 3.

3.2 Mobile Base

The motion functionality is based on the iRobot Create-2 low-cost educational robotic platform, chosen because of its easy integration and versatility for modular systems. The main features of this device are:

- Embedded velocity controller.
- Maximum wheel linear speed: 0.5m/s.
- Telemetry of position and angular velocity of wheels.
- Serial-USB interface.
- Integrated battery.

Fig. 3. Robot dimensions (millimeters).
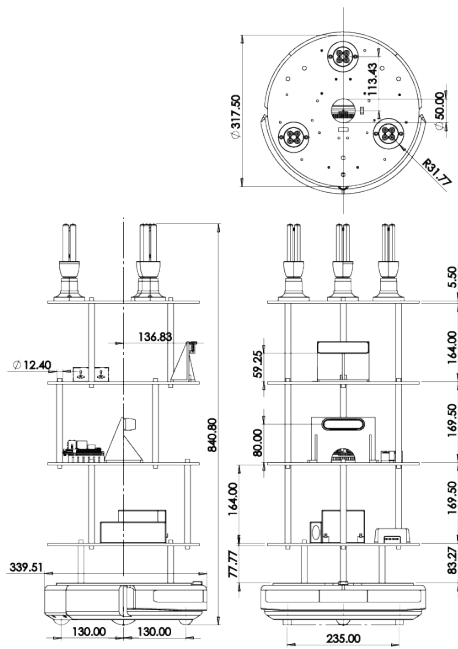
The Create 2 traction is based on a differential system with two wheels, allowing Z-axis rotation, forward and backward movement, and their combination. Kinematic model is presented in equations 1, from the axis presented in 4 (according to Malu and Majumdar (2014)).

$$\omega = \frac{v_l - v_r}{d}$$
$$v = \frac{v_l + v_r}{2}$$

(1)

Fig. 4. Mobile base's kinematic.

Where ω_r and ω_l are wheels angular velocities, v_r and v_l are linear velocities, w is robot angular velocity, v is robot linear velocity, and d is distance between wheels.

Create-2 embeds of the kinematic model computation to estimate odometry, measuring the velocity and position of the wheels (using optical encoders). The embedded computer obtains this estimation through the included USB interface.

3.3 Vision and Tracking Modules

The proposed robot has an RGB-D Intel Realsense D435 camera, which provides the ability of tridimensional perception. The depth sensor is based on Active IR Stereo and has a range of 0.3 meters to 3.0 meters and an accuracy of less than 2% at 2 meters. It also has a field of view (FOV) of 86° × 57° (±3°), resolution of 1280x720 pixels, and up to 90 frames per second.

In addition, the robotic platform includes a tracking camera Intel Realsense T265, that performs real-time correction of the odometry previously computed from the wheels telemetry. This device concatenates the internal IMU measurement with visual information from two 163±5° FOV fisheye cameras. Additionally, It concatemates the odometry calculated from the wheels telemetry. That data feeds a proprietary V-SLAM algorithm executed in an embedded Intel Movidius Myriad Vision Processing Unit (VPU).

Both cameras are arranged so that they do not capture the very structure of the robot, ensuring that the obtained information is entirely usable for mapping, location, and navigation purposes.

3.4 Embedded Computer

The system executes a centralized control in an embedded computer, managing the peripherals, sensors, and actuators through multiple ROS (Robot Operating System) nodes. The computer corresponds to a UP-Squared with the features presented in table 1.

Processor	Intel Atom x5-E3940 1.8HGz 4 Cores
RAM	4 GB
Storage	32 GB
Communication	WIFI 5GHz

Table 1. Embedded Computer Technical Specifications.

The embedded computer is in charge of the following tasks:

- High-level trajectories generation.
- Capturing and processing of RGB-D images.
- Reading of odometry information from the mobile base and the tracking camera.
- 3D Mapping, Localization, and Navigation.
- Activation of UV-C radiation system.

3.5 UV-C Radiation System

The robotic platform has three UV-C lamps of 30W and 120VAC each. This type of radiation was selected due to several studies showing satisfactory results in treating different types of bacteria and viruses (Yang et al. (2019); Fleming et al. (2018)).

The lamps are powered by a 12V/4Ah lead-acid battery and a DC-AC inverter that raises the voltage to 120VAC. Its operation is controlled through a solid-state relay directly activated by a GPIO of the UP Squared (galvanically isolated).

4. CONTROL SOFTWARE

The general control of the system is executed on the embedded computer, and the software runs mainly on ROS Kinetic (Quigley et al. (2009)) running on Ubuntu 18.04 LTS. ROS's decentralized philosophy allows the different blocks of the system to be developed and tested in independently controlled environments to be successfully integrated later. Additionally, the visualization and debugging tools offered by ROS speed up the system development and implementation.

Figure 5 presents the general block diagram of the control software, which contains the main ROS nodes that act in the system[1]. The information that is transmitted between nodes through topics is specified below:

A Velocity command generated by the navigation stack.
B Smoothed velocity command.
C Wheels velocities and positions.
D Odometry estimated from the wheels telemetry.
E Odometry corrected by the Realsense T265.
F RGB-D streaming.
G Environment map.
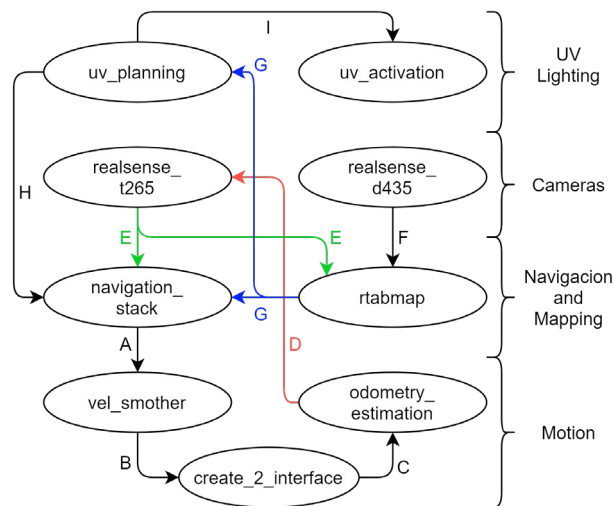H Navigation targets.
I UV-C activation signal.

Fig. 5. Software Block Diagram (ROS Nodes)

4.1 Motion Nodes

The *create_2_interface* node is the official bridge between Create 2 and ROS. It receives a linear and angular velocity command, calculates the kinematics, and sets the appropriate speed for each robot wheel. Also, it reads the speed and position of the wheels from the robot and publishes them as ROS topics. The *vel_smother* node is a decelerator that smoothes the speed commands to ensure the mechanical stability of the device (considering its height versus base area ratio). Finally, the *odometry_estimation* node computes the estimated position of the robot in the environment (using the numerical approximation of the integral of the speeds).

[1] Project source code:
https://github.com/edgarcamilocamacho/create_xl

4.2 Realsenses Nodes

Official nodes by Intel Realsense allow interface between the cameras and the ROS graph and manage their configuration. The *realsense_d435* node publishes the RGB image captured by its camera, as well as the point cloud captured by its infrared stereo system. The *realsense_t265* node subscribes to the wheels odometry and delivers it to the camera to correct it through its visual information. As a result, it publishes new and more accurate odometry.

4.3 Mapping and Navigation

Real-Time Appearance-Based Mapping (RTAB-Map) uses RGB-D images and Lidar Graph-Based SLAM approach to implement an incremental appearance-based loop closure detector. During the mapping process, RTAB-Map creates a graph-based map in which each node corresponds to the pose of the robot and the visual information obtained at one specific moment. During the rest of the mapping and localization process, the algorithm determines the level of similarity between the new visual information and previously stored information and creates the arcs corresponding to neighborhoods, similarities, or matching between the detections (Labbé and Michaud (2018)). Finally, during the localization process, the system loads the created graph and publishes the odometry and the estimated pose of the robot and the 2D projection of the map.

ROS 2D navigation stack is a toolbox that allows processing odometry information and sensor streams to generate a trajectory or movement sequence that makes the robot reach the desired destination (Marder-Eppstein et al. (2010)).

The system allows robot parameters configuration such as maximum speeds, movement restrictions, update frequencies, collision tolerance, among others. The navigation process concatenates a global map (generated by RTAB-Map) and a local map (dynamically created with the information obtained from the sensors). Then, it runs the planner to create a dynamic route (it is dynamic because it is modified when sensors find obstacles not on the global map). Different planners can be used for the specific case of this robot Path Optimization by Elastic Band (Quinlan and Khatib (1993)) was selected.

4.4 UV-C Radiation Nodes

"UV_Planning" node generates equidistant points in the available areas of the environment map, generates the commands to navigate to these points, and sends commands to turn on lamps and rotate when each objective is reached. The distance between the established targets is 1.5 m, and the rotation time is 15 seconds. However, these parameters can be adjusted according to the selected lamp's radiation intensity per area unit.

The "UV_activation" node is the interface with the hardware UV-C radiation hardware (solid-state relay). This node receives the Boolean value corresponding to the desired state of the lamps (ON or OFF).

4.5 Transformations Tree

This tree, whose transformations are static (except the odometry transformation), allows the different sensors and actuators to be located in a three-dimensional space to compute the appropriate geometric transformations to the points acquired by the sensors (in this case, the point cloud delivered by depth camera). The tree is presented in figure 6.
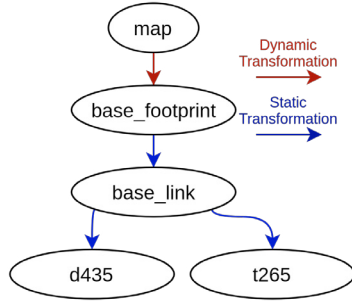

Fig. 6. Transformations tree

The transformation from frame *map* to frame *base_footprint* corresponds to the odometry calculated by the Create 2 wheels and corrected by the Realsense T265. The other transformations correspond to the geometry of the robot and the arrangement of the different elements that make up the system.

5. EXPERIMENTAL RESULTS

The final result of the robot implementation is presented in the figure 7. In order to corroborate the correct working of the system and start the robot operation, two stages are necessary: mapping and sanitizing (involves navigation).

Fig. 7. Robotics Platform.

Tests were carried out in a $38m^2$ one-plant loft with the following sections: kitchen, living room, and bedroom.

5.1 Mapping

The mapping procedure was performed by manually commanding the robot and covering as much of the area as possible. The 3D map is shown in figure 8. Likewise, its 2D projection is presented in the figure 9 [2]. Obtained maps correspond in shape and dimensions to the mapped loft.
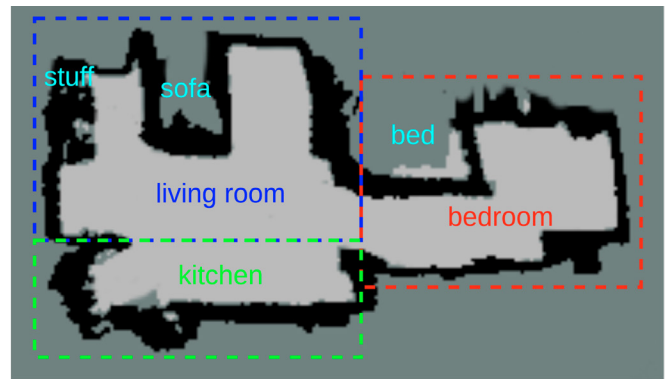

Fig. 8. 3D map.

Fig. 9. 2D map.

As well, the localization process was based on the RTAB-Map algorithm, which extracts characteristics from the captured images and compares them with those stored in the database to estimate the pose of the robot [3].

5.2 Navigation and UV-C Radiation

In order to sanitize as much area as possible in the working environment, the system generates equidistant target navigation points at 1.5m above the free spaces on the apartment map. The disinfection sequence corresponds to get located in each of these points and rotate for 15 seconds.

Figure 10 shows the apartment map and a grid of equally spaced target points. Only points that are located on a free area on the map are considered for the navigation route. The resulting route is simple due to the proximity of the selected targets. An orange shade represents the theoretical disinfection coverage, assumed as 1.5m radially.

During the route execution, 17 stops are made, in which the robot rotates on its axis to perform local disinfection.

[2] Mapping video:
https://www.youtube.com/watch?v=Y8m5YcCxWNw
[3] Localization video:
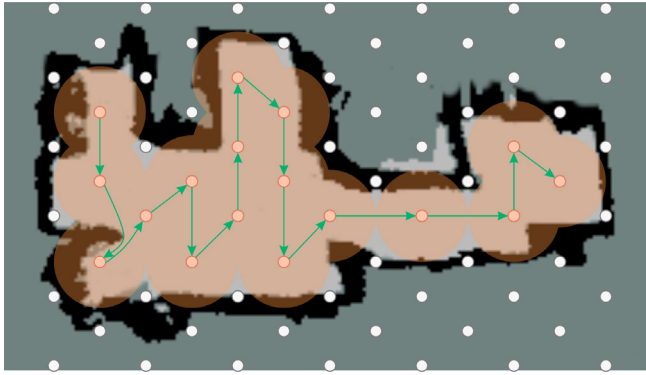https://www.youtube.com/watch?v=MziAE37uJtM

Fig. 10. Navigation paths, goal points, and sanitizing area.

The duration of the apartment sanitizing is approximately 8 minutes, as long as no obstacles are found.

6. CONCLUSIONS

The design and implementation of a robotic platform for a single plant indoor environment disinfection, such as offices, apartments, houses, etc., was presented. The easy-to-build and low-cost platform represents an additional tool to combat COVID-19, such as viruses and bacteria, which have been shown to be eradicated from type C ultraviolet radiation.

The system's effectiveness focuses on the possibility of autonomous disinfection without risk to cleaning staff, workers, or family members. COVID-Bot can disinfect the environment in which it works without direct supervision and covering the maximum possible surface in a single plant. In addition, the possibility to configure disinfection parameters such as the distance between the navigation targets and the disinfection time allows it to be functional and easy to replicate.

In addition, the implementation from standard open source technologies provides system scalability since additional technologies can be applied to increase the robot's functionalities. Therefore, this feature allows the constant growth and updating of the platform by the scientific community and its implementation and use in other applications, for instance: catering service, medication delivery in medical centers, groceries distribution in offices, and more.

The results obtained from the experimental tests showed that the robot could cover most of the area of a single-plant apartment, having previously known about the theoretical radiation range of the used lamps.

REFERENCES

Bai, H.M., Zaid, A., Catrin, S., Ahmed, K., and Ahmed, A. (2020). The socio-economic implications of the coronavirus pandemic (covid-19): A review. *Int. J. Surg.*, 8(4), 8–17.

Balkhair, A.A. (2020). Covid-19 pandemic: a new chapter in the history of infectious diseases. *Oman medical journal*, 35(2), e123.

Chen, Z., Fan, T., Zhao, X., Liang, J., Shen, C., Chen, H., Manocha, D., Pan, J., and Zhang, W. (2021). Autonomous social distancing in urban environments using a quadruped robot. *IEEE Access*, 9, 8392–8403.

Daniel, J. (2020). Education and the covid-19 pandemic. *Prospects*, 49(1), 91–96.

Fleming, M., Patrick, A., Gryskevicz, M., Masroor, N., Hassmer, L., Shimp, K., Cooper, K., Doll, M., Stevens, M., and Bearman, G. (2018). Deployment of a touchless ultraviolet light robot for terminal room disinfection: the importance of audit and feedback. *American journal of infection control*, 46(2), 241–243.

Isabet, B., Pino, M., Lewis, M., Benveniste, S., and Rigaud, A.S. (2021). Social telepresence robots: A narrative review of experiments involving older adults before and during the covid-19 pandemic. *International Journal of Environmental Research and Public Health*, 18(7), 3597.

Labbé, M. and Michaud, F. (2018). Rtab-map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation: LabbÉ and michaud. *Journal of Field Robotics*, 36. doi:10.1002/rob.21831.

Malu, S.K. and Majumdar, J. (2014). Kinematics, localization and control of differential drive mobile robot. *Global Journal of Research In Engineering*.

Marder-Eppstein, E., Berger, E., Foote, T., Gerkey, B., and Konolige, K. (2010). The office marathon: Robust navigation in an indoor office environment. In *International Conference on Robotics and Automation*.

Murphy, R.R., Gandudi, V.B.M., and Adams, J. (2020). Applications of robots for covid-19 response. *arXiv preprint arXiv:2008.06976*.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. (2009). Ros: an open-source robot operating system. volume 3.

Quinlan, S. and Khatib, O. (1993). Elastic bands: connecting path planning and control. In *[1993] Proceedings IEEE International Conference on Robotics and Automation*, 802–807 vol.2. doi:10.1109/ROBOT.1993. 291936.

Somaldo, P., Ferdiansyah, F.A., Jati, G., and Jatmiko, W. (2020). Developing smart covid-19 social distancing surveillance drone using yolo implemented in robot operating system simulation environment. In *2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC)*, 1–6. IEEE.

Spinelli, A. and Pellino, G. (2020). Covid-19 pandemic: perspectives on an unfolding crisis. *Journal of British Surgery*, 107(7), 785–787.

Struyf, T., Deeks, J.J., Dinnes, J., Takwoingi, Y., Davenport, C., Leeflang, M.M., Spijker, R., Hooft, L., Emperador, D., Dittrich, S., et al. (2020). Signs and symptoms to determine if a patient presenting in primary care or hospital outpatient settings has covid-19 disease. *Cochrane Database of Systematic Reviews*, (7).

Yang, G.Z., Nelson, B.J., Murphy, R.R., Choset, H., Christensen, H., Collins, S.H., Dario, P., Goldberg, K., Ikuta, K., Jacobstein, N., et al. (2020). Combating covid-19—the role of robotics in managing public health and infectious diseases.

Yang, J.H., Wu, U.I., Tai, H.M., and Sheng, W.H. (2019). Effectiveness of an ultraviolet-c disinfection system for reduction of healthcare-associated pathogens. *Journal of Microbiology, Immunology and Infection*, 52(3), 487–493.