

# Intelligent mobile robot for contagious disease treatments in hospitals ☆☆☆



Hoang T. Tran<sup>a,b</sup>, Dong TL. Tran<sup>a,b</sup>, Minh T. Nguyen<sup>c,d,\*</sup>, Thang C. Vu<sup>e</sup>

<sup>a</sup> Center of Electrical Engineering, Duy Tan University, Da Nang, 550000, Viet Nam

<sup>b</sup> Faculty of Electrical-Electronic Engineering, Duy Tan University, Da Nang, 550000, Viet Nam

<sup>c</sup> Department of Electrical Engineering, Thai Nguyen University of Technology, Thai Nguyen, 240000, Viet Nam

<sup>d</sup> Thai Nguyen University, Thai Nguyen, 240000, Viet Nam

<sup>e</sup> Thai Nguyen University of Information and Communication Technology, Thai Nguyen, 240000, Viet Nam

## ARTICLE INFO

### Method name:

Design an Intelligent Mobile Robot for Contagious Disease Treatments in Hospitals

### Keywords:

Environmental disinfection  
Extended Kalman filter  
Localization  
Intelligent mobile robot  
Disinfection robot

## ABSTRACT

This paper provides a novel and applicable work that builds a real system for disinfecting the air and surfaces of the environment in a hospital room, with a non-contact measurement system for supporting contagious disease treatments in hospitals. The system is built on an intelligent mobile robot system that operates autonomously in a simulated real treatment room. The research team uses a new positioning algorithm. It is a combination of data from the Lidar sensor, encoder, and Extended Kalman filter. The program that applies segmentation and image feature extraction algorithms is developed to meet requirements of real-time environment mapping in the room. Control algorithms for moving and avoiding obstacles are also proposed. Next, techniques for collecting health data including patient identification, body temperature, and blood oxygen index via wireless sensor network are also mentioned in the article. Analysis and experimental results show qualified outcomes and promise. The main contribution of the paper can be listed as follows.

- Design and build a new CEE-IMR, an intelligent mobile robot that can recognize patients, guide and lead them walking in hospitals, especially keep a safe distance avoiding contagious diseases.
- A novel framework for controlling the robot is proposed. The robot can move flexible, avoid obstacles, etc. based on advanced control algorithms. A new control mechanism is also proposed.
- Methods of collecting data and processing medical data to support either patients or doctors to improve the efficiency in hospitals in contagious disease management.

☆ **Related research article:** None

☆☆ **For a published article:** None

\* Corresponding author.

*E-mail address:* [nguyentuanminh@tnut.edu.vn](mailto:nguyentuanminh@tnut.edu.vn) (M.T. Nguyen).

<https://doi.org/10.1016/j.mex.2024.102941>

Received 14 July 2024; Accepted 28 August 2024

Available online 4 September 2024

2215-0161/© 2024 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC license

(<http://creativecommons.org/licenses/by-nc/4.0/>)

## Specifications table

This table provides general information on your method.

Subject area:	Engineering
More specific subject area:	<i>Autonomous robot (AR), control, AR assisted medical treatment, sensing systems</i>
Name of your method:	<b><i>Design an Intelligent Mobile Robot for Contagious Disease Treatments in Hospitals</i></b>
Name and reference of original method:	<i>All the reference papers are provided in the Ref. Section.</i>
Resource availability:	<i>All data of this work (e.g., equipment, data, software, hardware) can be shared upon request.</i>

## Background

Mobile robots provide a wide range of applications in both civil and military areas [1,2]. Mobile robots may detect, monitor, collect and analyze data, navigate among other things, depending on their intended use. Since the outbreak of the coronavirus pandemic 2019 (COVID-19) occur in 2019, new approaches for robots in a contagious world are exploited rapidly to support either hospitals or in public areas. The role of robots in the following COVID-related applications: disinfection, checking human temperature, monitoring public places, delivering food and other items, food preparation and personal interactions by distanced contacts [3–5]. The effective control and prevention of this epidemic remain major challenges for the international community. In addition, contagious surfaces and environments are important sources of infection that promote the spread of the virus. Therefore, environmental disinfection in healthcare facilities is important to reduce transmission, prevent hospital-acquired infections, and control disease outbreaks [6,7].

Currently, the global socio-economic development and health system reform have enabled the medical industry to move towards true "intelligence". The combination of artificial intelligence (AI) techniques, 5 G and 6 G technologies, Internet of Things (IoTs), manufacturing robots, cyber-physical (CPS), and other new high technologies enrich the medical profession [8]. The advanced technologies can fully support the equipments to fulfill their tasks in hospitals or in other environments to protect human from the virus.

Prevention and control of hospital-acquired infections is a cornerstone of health safety and a core mission in improving the quality of health services. The combination of engineering technology and infection management in hospitals has led to breakthroughs in infection control technology [9–14]. Healthcare organizations around the world have been optimizing hospital intelligence by gathering and organizing information from multiple perspectives to build innovative models in hospital development. Some results related to medical services avoiding contagious diseases have been shown separately due to specific missions. Robots are preferably deployed at a hospital's front desk to spread information about the hospital's various units/sections and to guide patients and visitors [9]. Nursing robots in hospital and at home for elderly care are investigated in [10]. Robots also can provide immediate medical aid after an accident is critical in order to prevent intensification of trauma [11]. Robots can help doctors far from patients to take all the physiological parameters and diagnoses a disease using audiovisual aids [12]. Delivery services inside hospitals can be carried out by mobile robots that serving food and beverages, dispensing of drugs, removing of unclean laundry, delivery of fresh bed linen, and transportation of regular and contaminated waste etc. [13]. Cleaning robots for medical surroundings appear to be capable of delivering the innovation that non-industrial robot creators predicted years ago [14]. These models help to reduce costs associated with the prevention and control of hospital sterilization and only deal with specific functions.

Recently published research works [15–18] have solved the problems of disinfecting and cleaning the environments, but have not mentioned the problems of positioning and guiding or navigating robots that can operate correctly in disinfected environments. This is a critical issue that the robot can come into direct contact with the patient to replace the human that has not been interested by the authors.

In this work, many functions are integrated into one mobile robot, called CEE-IMR. The mobile robot shows a maneuverability to move in an infected area to collect data from patients to be sent to doctors or a data processing center. The hardware and software are designed significantly to be able to adapt to hospital environments. The robot receives orders and work independently. Technical designs are different from existing work that can support effectively to either patients or doctors.

As shown in Fig. 1, we present the indirect control of blood oxygen levels (SpO<sub>2</sub> is an important indicator to monitor and evaluate the health of COVID-19 patients in the recommended treatment regimen [19] and patient body temperature through Thermal Image Processing or Infrared (IR) spectral detection method [20]. Through these two parameters, the doctor does not need to have direct contact with the patient, but can still monitor and evaluate the appropriate treatment regimen.

The disinfection is performed by a system including spraying, mopping the floor surface, and UV-C lamp described in the design of the robot body. In addition, controlling the robot to follow the trajectory to the desired locations and some theories related to this test will be presented in the next sections. For the task of avoiding close obstacles in the local area, we have applied the VFH+ method [21–23] with a system of ultrasonic sensors and developed a control program that allows avoiding obstacles in the local area from 0.3 to 4meters.

The Lidar system allows the detection of features of the robot's environment to combine with encoder sensor data, which allows the robot to move precisely to the target locations. The measurement and experimental results are presented in section 3.1. With these results, it is possible to open up the possibilities of applying intelligent robots for contagious disease treatments in practice.

The rest of this work is organized as follows. The system models that combine all the functions in the robot is addressed. The proposed methods including algorithms, calculations are provided. Next, experimental results including control problems, data collecting, processing and analyzing are addressed. Finally, conclusions and future developments are provided.

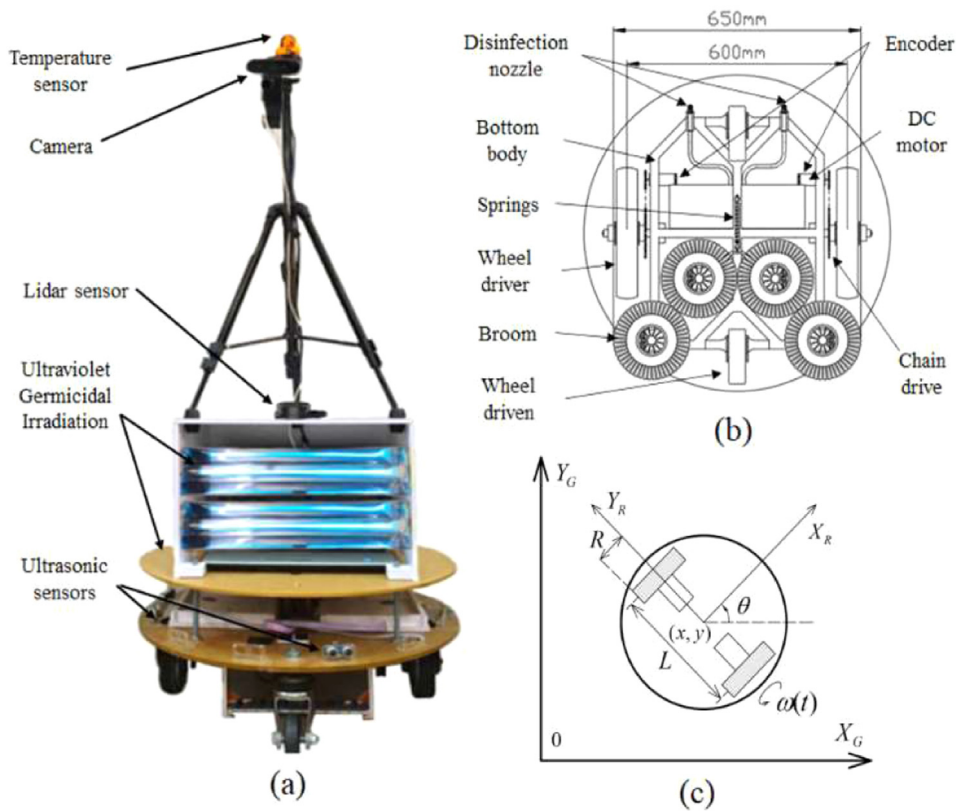


Fig. 1. (a) Image of robot model in the experiment, (b) Detailed mechanical drawing of the base of the robot, (c) Kinetic model of mobile robot.

**Method details**

- + A method of designing and building a new intelligent mobile robot, named CEE-IMR, that can recognize patients, guide and lead them walking in hospitals, especially keep a safe distance avoiding contagious diseases.
- + A method for controlling the robot is proposed. The robot can move flexible, avoid obstacles, etc. based on advanced control algorithms. A new control mechanism is also proposed.
- + A Method of collecting medical data and processing the data is proposed to support either patients or doctors to improve the efficiency in hospitals in contagious disease management.

**The proposed system models to support contagious disease treatments**

Fig. 2 is a block diagram of a mobile robotic system model for contagious disease treatment support. This model has 2 main systems: The sensor system consists of turquoise blocks (Fig. 2) presented in Section 2.1; The control system consists of orange and blue blocks (Fig. 2) presented in Section 1.2.

*Sensor systems*

*Encoder*

The optical encoder has an LED light source, a light detector, a “code” disc/wheel mounted on the shaft, and output signal processor, Fig. 3. The disc has alternating opaque and transparent segments and is placed between the LED and photodetector. As the encoder shaft rotates, the light beam from the LED is interrupted by the opaque lines on the “code” disk before being picked up by the photodetector. This produces a pulse signal: light = on; no light = off. The signal is sent to the counter or controller, which will then send the signal to produce the desired function.

In mobile robotics, the encoder is used to measure the movement (direction and speed) of each of the wheels of the robot. Determining the position of the robot by this encoder is a popular method in the world called the Odometry method.

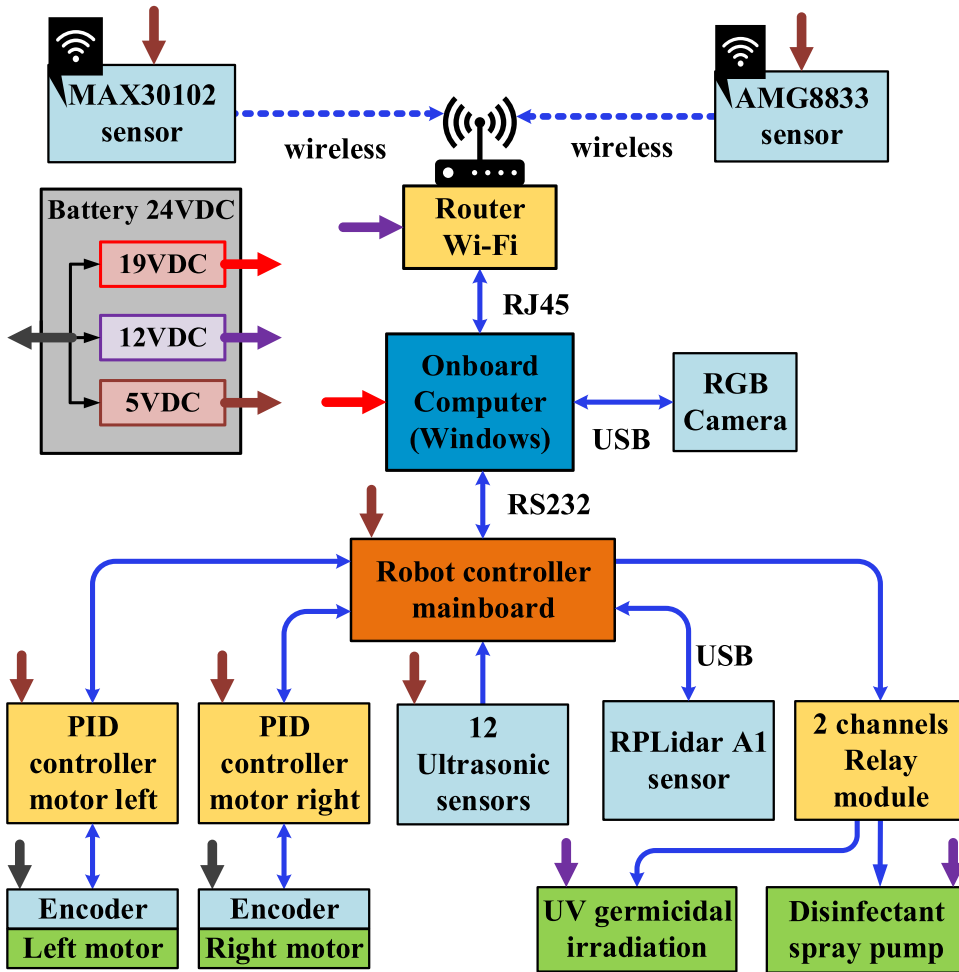


Fig. 2. Block diagram of CEE-IMR system.

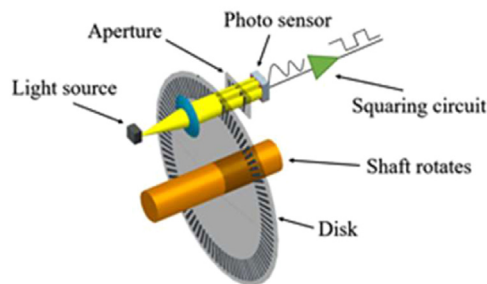


Fig. 3. Structure of the rotary encoder.

*Lidar (Light detection and ranging)*

To increase the efficiency of positioning we use the Lidar sensor. Lidar is a method for determining ranges (variable distance) by targeting an object with a laser and measuring the time for the reflected light to return to the receiver. In this research, as shown in Fig. 4, the research team used the lidar method to locate the robot in the hospital using the RPLIDAR A1 sensor.

The RPLIDAR A1 is a low-cost 360° 2D laser scanner (LIDAR) solution developed by SLAMTEC. The system can perform a 360° scan within the 12 m range. The produced 2D point cloud data can be used in mapping, localization, and object/environment modelling.

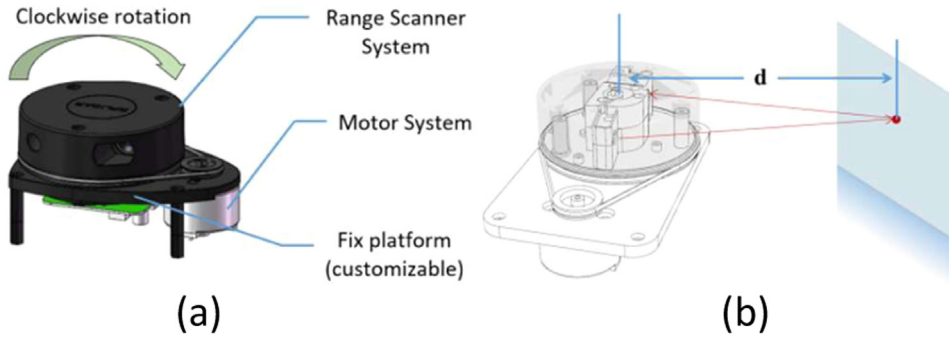


Fig. 4. (a) RPLIDAR A1 System Composition, (b) The RPLIDAR A1 Working Schematic.

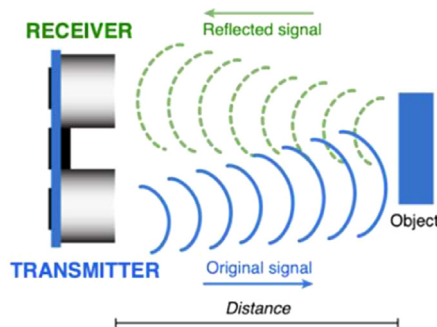


Fig. 5. Ultrasonic sensor.



Fig. 6. Logitech BRIO 4 K.

*Ultrasonic sensor*

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

In a hospital environment, there are many people and obstacles on the way, so the robot must be aware to avoid that obstacle. The ultrasonic sensors are used primarily as proximity sensors. They can be found in automobile self-parking technology and anticollision safety systems. Therefore, we use ultrasonic sensor SRF05 for the robot so that the robot can avoid obstacles, as illustrated in Fig. 5.

*RGB camera*

The robot is designed in addition to the functions mentioned above, it also has the function of guiding the patient to move in the hospital. Accordingly, the robot both moves and tracks the patient’s face to maintain a distance from the patient during movement. If the patient stops or slows down, the robot will stop to wait, which is made possible by a method that predicts the distance from the image sensor to the person’s face using the monocular camera. We use the Logitech BRIO 4 K Camera, as shown in Fig. 6, installed on the robot to capture the patient’s face image.

*Infrared thermal image sensor and SpO2 sensor*

The temperature sensor used in this research is Panasonic’s AMG8833, as shown in Fig. 7. An infrared thermal image sensor with 64 thermal pixels distributed into a 2D 8 × 8 matrix, can measure the temperature of an object in the range of from 0 °C to 80 °C.

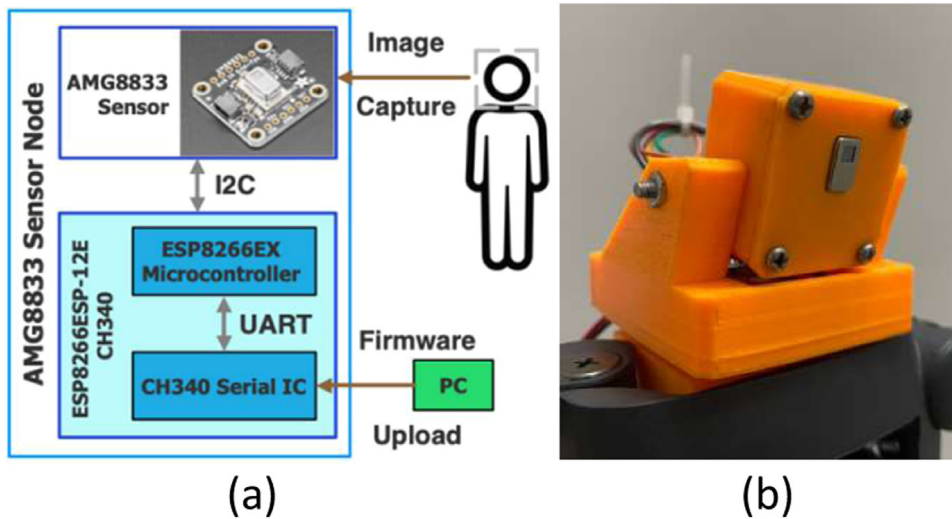


Fig. 7. (a) Infrared thermal image sensor – AMG8833 sensor Node, (b) Case for the AMG8833 sensor Node.

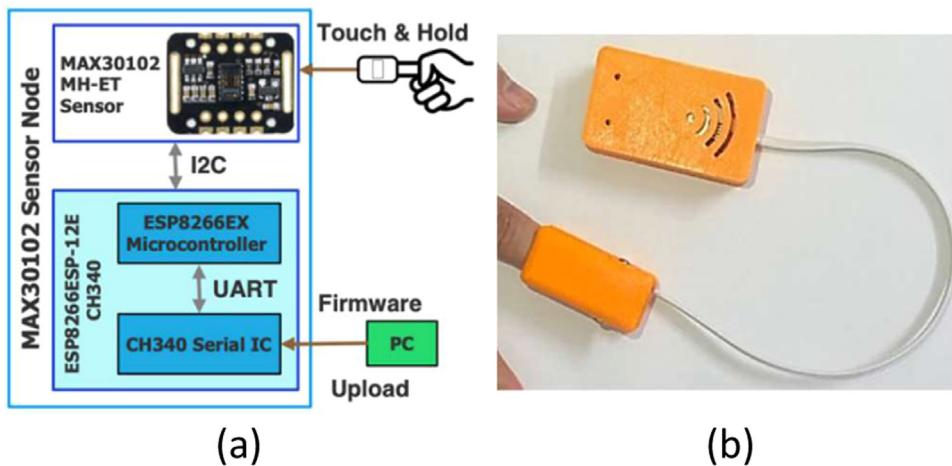


Fig. 8. (a) Infrared thermal image sensor – MAX30102 sensor Node, (b) Case for the MAX30102 sensor Node.

Non-contact infrared sensor with measuring distance up to 7 m under ideal conditions. Sensor measuring angle up to 60° in both horizontal and vertical. The sampling rate is 10 fps (frame per second).

The robot is set to move at a speed of 0,2 m/s to 0,6 m/s. Therefore, under ideal conditions, it is possible to collect temperature data even while the robot is moving.

The SpO2 sensor, as shown in Fig. 8, is the MAX30102 MH-ET LIVE module that integrates the MAX30102 heart rate and SpO2 sensor. The MAX30102 sensor contains two light-emitting diodes (LEDs), one infrared (peak wavelength 880 nm) and the other red (peak wavelength 660 nm); along with a photodiode that is specific for wavelengths between 600 and 900 nm. The values measured by the sensor are transmitted with a checksum, such that the validity of the data is verified when it is received by the supporting microcontroller (3). According to Fig. 8, this sensor is worn on the patient’s finger.

*The control system*

The sensory information network of the robot is shown in Fig. 2, in which the computer is the central control unit. Due to the characteristics of different information rates between sensors (the information here is point-to-multipoint), a special design for the communication network in the robot has been implemented. As shown in Fig. 2, the ultrasonic sensors, RPLidar and robot actuators (motors) are connected to the robot’s control unit (serial communication RS-232C); The RGB camera sensor is connected to the control unit via a high-speed communication channel; Other path (Modules Relay, UV germicidal irradiation, disinfectant spray pump). These blocks connect to the control board (via RS-485 communication standard) with a point-to-multipoint line according to the Modbus/RTU polling protocol.

The control program for the system to collect patient data and disinfect the floor during robot movement can be summarized as follows:

Starting from the hardware configuration shown above, the software that collects information from the sensors can be divided into modules with the following specific tasks:

- The program for collecting and processing data from the sensor RPLidar is a special type developed specifically with the C++ language;
- Camera image information is collected directly into the computer and processed through an image processing program, which is developed in Visual C++ environment with Intel's OpenCV tool;
- Control program for two optical encoder measuring circuits (mounted on motor shafts) connected directly to a microcomputer (via RS-232C);
- The remaining information is transmitted and received between 14 network nodes and control nodes from the computer (2 nodes for channels Relay module, 12 node for 12 ultrasonic sensors) via transmission line RS-485 using Modbus/RTU protocol;

Fig. 9 shows an overview of the navigation process for a mobile robot, which can be summarized as follows.

1. Robot collects data through sensors;
2. Determine the position of the robot;
3. Mapping (if necessary or import existing global maps);
4. plan the way;
5. Control the moving robot to follow the trajectory to the destination and avoid obstacles (if any).

#### Collection of patient health data for medical treatment

As mentioned above, SpO2 index and body temperature are two important measurements in monitoring and treating covid 19 patients. Our robot will collect these two kinds of data automatically. Data collection is done automatically. The robot moves close to the patient's bed within a radius of 1 meter, the sensor nodes will automatically connect to the Wi-Fi network emitted by a router placed on the robot. And those nodes actively send measured data to the central computer according to the MQTT protocol [24]. A typical MQTT is a TCP-based protocol uses a topic-based publish-subscribe architecture, as illustrated in Fig. 10. It's used to establish communication between multiple devices. This communication protocol is suitable for transferring data between resource-constrained devices with low bandwidth and low power requirements. Therefore, this messaging protocol is widely used for communication in IoT Framework.

As shown in Fig. 11, there are, 1- the publishers, which are IoT devices, usually sensors that share data; 2- the subscribers, which are IoT devices that receive shared data by the publishers; 3- the brokers, which are entities that manage topics and message exchange, and can be implemented within any device in the network, including the publishers and subscribers. When new data is generated, publishers send messages to the broker, in order to update a given topic. Every time a topic is updated, the broker broadcasts the new data to every subscriber of that topic. Fig. 11 represents a sequence diagram regarding the message exchange in the MQTT protocol. With the above technical characteristics, MQTT can work in unstable transmission conditions. Hence it is suitable for our robot model.

In Fig. 11, the wireless sensor data collection system is set up with 3 clients in which AMG8833 and MAX30102 are 2 sensor nodes (called MQTT clients) that collect data independently of each other and with the robot. Every time the sensor nodes "find" the Wi-Fi network emitted by the robot, they connect to a server (which is an open-source Mosquitto broker application that runs in the background inside the robot's computer-<https://mosquitto.org/>). All collected sensor data is stored in the software on the computer to provide medical treatment data to the doctor.

## The proposed methods

### Program to merge data of sensors for mobile robot positioning

#### Positioning for robots using extended Kalman filters

As shown in Fig. 1(c) the coordinate system and parameter symbols of the mobile robot are designed to move on the flat floor in the room, where  $(X_G, Y_G)$  are the axes of the global coordinate system of the room,  $(X_R, Y_R)$  are local coordinate system axes that match the robot's center. Let the angular velocities of the right and left wheels be  $\omega_R$  and  $\omega_L$  respectively, and the sampling interval  $\Delta t$  is sufficiently short. So that the input control signals that are the displacement increments applied to these wheels are:

$$\begin{aligned}\Delta s_R &= \Delta t R \omega_R \\ \Delta s_L &= \Delta t R \omega_L\end{aligned}\tag{1}$$

From here, the distance increment of the robot center  $\Delta s$  is:

$$\Delta s = \frac{\Delta s_R + \Delta s_L}{2}\tag{2}$$

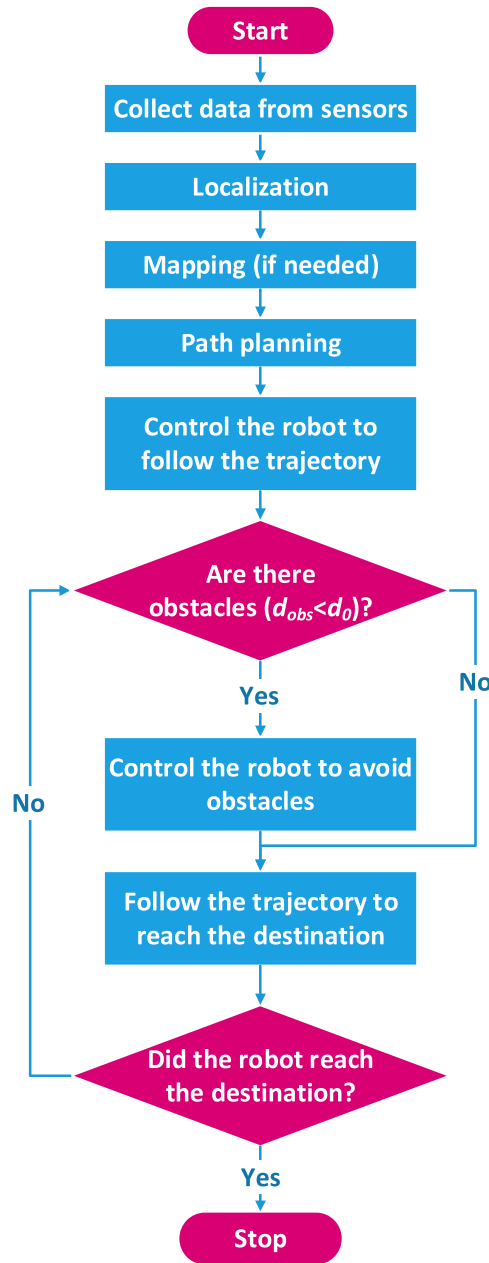


Fig. 9. Flowchart of navigation process for mobile robot.

And the angular increment in which the robot rotates  $\Delta\theta$  is:

$$\Delta\theta = \frac{\Delta s_R - \Delta s_L}{L} \tag{3}$$

So the equation of state of the robot at time  $k$  in the global coordinate system updated from time  $k-1$  is as follows:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \Delta s_k \cos(\theta_{k-1} + \Delta\theta_k/2) \\ \Delta s_k \sin(\theta_{k-1} + \Delta\theta_k/2) \\ \Delta\theta_k \end{bmatrix} \tag{4}$$

The displacement increments  $\Delta s_R$  and  $\Delta s_L$  (proportional to the set speeds of the 2 wheels  $\omega_R$  and  $\omega_L$  as measured by the crankshaft encoder sensors) are input variables affected by errors systematic and random errors.

In practice, the system (4) encounters systematic errors (such as dimensional errors of mechanical parts, shaft offsets, encoder resolution limits, ...) and non-systematic errors. In this paper, the errors are proportional to the two-wheel displacement increments



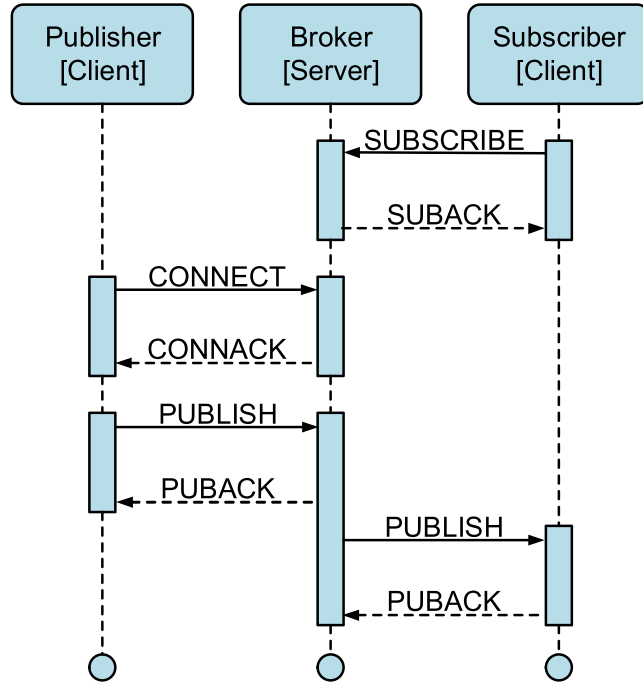


Fig. 10. Message exchange in the MQTT protocol.

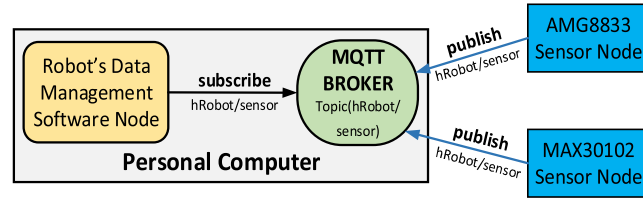


Fig. 11. Diagram of sensor data exchange using MQTT protocol.

$\Delta s_R$  and  $\Delta s_L$ , these increments when subjected to noise can be expressed as the sum of the two components including the value identification and interference:

$$\begin{aligned} \Delta s_R &= \Delta s_{R0} + \varepsilon_R \\ \Delta s_L &= \Delta s_{L0} + \varepsilon_L \end{aligned} \tag{5}$$

where  $\Delta s_{R0}$  and  $\Delta s_{L0}$  are the nominal values of the input signal;  $\varepsilon_R$  and  $\varepsilon_L$  are the values of the white noise process, which are independent, have zero mean and are normally distributed.

The equation of state (4) can be written in more detail as follows:

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{(\Delta s_{R0(k)} + \varepsilon_{R(k)}) + (\Delta s_{L0(k)} + \varepsilon_{L(k)})}{2} \cos \left( \theta_{k-1} + \frac{(\Delta s_{R0(k)} + \varepsilon_{R(k)}) - (\Delta s_{L0(k)} + \varepsilon_{L(k)})}{2L} \right) \\ \frac{(\Delta s_{R0(k)} + \varepsilon_{R(k)}) + (\Delta s_{L0(k)} + \varepsilon_{L(k)})}{2} \sin \left( \theta_{k-1} + \frac{(\Delta s_{R0(k)} + \varepsilon_{R(k)}) - (\Delta s_{L0(k)} + \varepsilon_{L(k)})}{2L} \right) \\ \frac{(\Delta s_{R0(k)} + \varepsilon_{R(k)}) - (\Delta s_{L0(k)} + \varepsilon_{L(k)})}{L} \end{bmatrix} \tag{6}$$

Due to the cumulative nature of systematic errors, the accuracy of the position estimation will be worse as the robot moves. Many studies have tried to increase the reliability of this positioning, including the use of the Kalman filter. We also applied sensor fusion using the Kalman filtering algorithm for positioning to the robot and obtained results that improved the quality of these measurements significantly [25,26]. It can be summarized as follows:

If the state vector  $\mathbf{x} \in \mathfrak{R}^n$  of a controlled process is represented by Eq. (12). This nonlinear equation has the form:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}, \mathbf{w}_{k-1}) \tag{7}$$

where  $f$  is a system function with  $\mathbf{x} = [x \ y \ \theta]^T$  is the state vector,  $\mathbf{u} = [\Delta s_R \ \Delta s_L]^T$  is the input control vector and  $\mathbf{w} = [\varepsilon_R \ \varepsilon_L]$  is the process noise.

This state is observed by several measurements that make up the output vector  $\mathbf{z}$ :

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k) \quad (8)$$

where  $h$  is the measurement function and  $\mathbf{v} = [\varepsilon_R \ \varepsilon_L]$  is the measurement noise affecting the readings of displacement increments from the rotary axis encoders.

$\mathbf{w}_k$  and  $\mathbf{v}_k$  are independent, are white noise, have a normal probability distribution with covariance matrices  $\mathbf{Q}$  and  $\mathbf{R}$  respectively:

$$P(\mathbf{w}_k) \sim N(0, \mathbf{Q}_k); P(\mathbf{v}_k) \sim N(0, \mathbf{R}_k); E(\mathbf{w}_i \mathbf{v}_j^T) = 0 \quad (9)$$

*Define matrices in sensor fusion steps*

The speed and distance of the robot are shown in Fig. 1(c) is measured by the Odometry method based on the function  $f$  in the system Eq. (7).

- **Starting phase:** From step  $k = 0$ , the robot starts from a known initial position with a posterior state estimate  $\hat{\mathbf{x}}_0$  and the covariance of the error estimate  $\mathbf{P}_0$ .
- **Prediction phase:** At each time step  $k$ , the filter propagates the state  $\mathbf{x}$  and covariance  $\mathbf{P}$  of the system in the previous step to the current step using the time update equation:

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_{k-1}, \mathbf{0}) \quad (10)$$

$$\mathbf{P}_k^- = \mathbf{A}_k \mathbf{P}_{k-1} \mathbf{A}_k^T + \mathbf{W}_k \mathbf{Q}_{k-1} \mathbf{W}_k^T \quad (11)$$

Here, input control signals  $\mathbf{u} = [\Delta s_R \ \Delta s_L]^T$  measured by the encoder sensor.

To get the filter's matrix  $\mathbf{P}$ , the matrices  $\mathbf{Q}$  and  $\mathbf{A}$  are defined as follows:

- $\mathbf{Q}_k$  is the input noise covariance matrix with size  $[2 \times 2]$ , we have:

$$\mathbf{Q}_k = \text{covar}(\Delta s_{R(k)}, \Delta s_{L(k)}) = \begin{bmatrix} \delta_R |\Delta s_{R(k)}| & 0 \\ 0 & \delta_L |\Delta s_{L(k)}| \end{bmatrix} \quad (12)$$

Here,  $\delta_R$  and  $\delta_L$  are error constants representing the uncertainty parameters of the motor control circuit and the interaction of the wheel with the floor, and they are independent of each other. Thus, the variance of the process noise is proportional to the absolute value of the increments distance travelled by each wheel  $\Delta s_R$  and  $\Delta s_L$ . These scale constants depend on the structure of the robot and the environment. They are determined experimentally as  $\delta_R = \delta_L = \delta = 0,0003$ .

- $\mathbf{A}_k$  is a matrix of size  $[3 \times 3]$ , calculated by:

$$\mathbf{A}_k = \begin{bmatrix} \frac{\partial f_x}{\partial x} & \frac{\partial f_x}{\partial y} & \frac{\partial f_x}{\partial \theta} \\ \frac{\partial f_y}{\partial x} & \frac{\partial f_y}{\partial y} & \frac{\partial f_y}{\partial \theta} \\ \frac{\partial f_\theta}{\partial x} & \frac{\partial f_\theta}{\partial y} & \frac{\partial f_\theta}{\partial \theta} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\Delta s_k \sin\left(\theta_{k-1} + \frac{\Delta \theta_k}{2}\right) \\ 0 & 1 & \Delta s_k \cos\left(\theta_{k-1} + \frac{\Delta \theta_k}{2}\right) \\ 0 & 0 & 1 \end{bmatrix} \quad (13)$$

- $\mathbf{W}_k$  is a matrix of size  $[3 \times 2]$ , calculated by:

$$\mathbf{W}_k = \begin{bmatrix} \frac{1}{2} \cos\left(\theta_{k-1} + \frac{\Delta \theta_k}{2}\right) + \frac{\Delta s_k}{2L} \sin\left(\theta_{k-1} + \frac{\Delta \theta_k}{2}\right) & \frac{1}{2} \cos\left(\theta_{k-1} + \frac{\Delta \theta_k}{2}\right) + \frac{\Delta s_k}{2L} \sin\left(\theta_{k-1} + \frac{\Delta \theta_k}{2}\right) \\ \frac{1}{2} \sin\left(\theta_{k-1} + \frac{\Delta \theta_k}{2}\right) + \frac{\Delta s_k}{2L} \cos\left(\theta_{k-1} + \frac{\Delta \theta_k}{2}\right) & \frac{1}{2} \cos\left(\theta_{k-1} + \frac{\Delta \theta_k}{2}\right) - \frac{\Delta s_k}{2L} \sin\left(\theta_{k-1} + \frac{\Delta \theta_k}{2}\right) \\ & \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \quad (14)$$

- **Correction phase:** The filter corrects the a priori state estimate with  $\mathbf{z}_k$  measurements by the following measure update equations:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (15)$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{z}_k - h(\hat{\mathbf{x}}_k^-)) \quad (16)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (17)$$

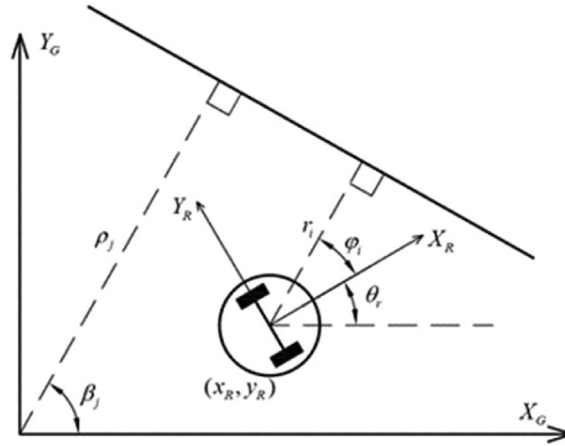


Fig. 12. Lidar sensor measures 2 parameters of a straight line in the environment.

In this case we use an absolute measurement using the RPLIDAR sensor. This sensor allows the detection of features of the environment around the robot, which are lines detected in the environment (for example, a line that cuts across a flat wall in a room).

By a transformation such as the Hough transform, we have two parameters of distance  $r$  from the line to the origin and angle  $\psi$  between the line as shown in Fig. 12. The global map of the built environment consists of a set of Line segments are described by the parameters  $\beta_j$  and  $\rho_j$ . The equation of the straight line in canonical form is:

$$x_G \cos \beta_j + y_G \sin \beta_j = \rho_j \tag{18}$$

As the robot moves, a local map of the environment is built with coordinates mounted on the robot. It also includes a set of line segments described by the equation:

$$x_R \cos \psi_i + y_R \sin \psi_i = r_i \tag{19}$$

where  $\psi_i$  and  $r_i$  are the parameters of the  $i^{th}$  line among  $N$ .

Line segments of the local and global environment will be detected as matches. Line segments that are matched to the local map will be collected into vector  $z_k$ , used as input for the EKF editing step:

$$z_k = [r_{1k}, \psi_{1k}, \dots, r_{Nk}, \psi_{Nk}]^T \tag{20}$$

Calculate the matrices  $R$ ,  $h$  and  $V$

From the coordinates and direction of the robot estimated by the odometry method, the parameters  $\beta_j$  and  $\rho_j$  of the  $j^{th}$  line segment in the global map are converted to parameters  $r_i$  and  $\psi_i$  according to the local coordinate system of the robot. They are calculated by:

$$\begin{bmatrix} r_i \\ \psi_i \end{bmatrix} = \begin{bmatrix} |C_j| \\ \beta_j - \theta + \pi(-0.5 \text{sign}(C_j) + 0.5) \end{bmatrix}; \tag{21}$$

$$C_j = \rho_j - x \cos \beta_j - y \sin \beta_j$$

$\psi_i$  and  $r_i$  are the measurements that are directly affected by the measurement noises of the RPLIDAR sensor. Call these measurement disturbances  $\epsilon_r$  and  $\epsilon_\psi$ , then the measured component vector from the laser sensor when including the interferences will be:

$$\begin{bmatrix} r_i \\ \psi_i \end{bmatrix} = \begin{bmatrix} |C_j| \\ \beta_j - \theta_r + (-0.5 \text{sign}(C_j) + 0.5)\pi \end{bmatrix} + \begin{bmatrix} \epsilon_r \\ \epsilon_\psi \end{bmatrix} \tag{22}$$

Thus, when adding a Lidar sensor, the measurement function  $h$  is added 2 N components  $r_i$  and  $\psi_i$ . When  $k$  is hidden, we have the following matrices:

- The matrix  $H_{odl}$  has size  $[(2N) \times 3]$  and is equal to:

$$H_{odl} = \begin{bmatrix} \frac{\partial r_1}{\partial x} & \frac{\partial r_1}{\partial y} & \frac{\partial r_1}{\partial \theta} \\ \frac{\partial \psi_1}{\partial x} & \frac{\partial \psi_1}{\partial y} & \frac{\partial \psi_1}{\partial \theta} \\ \vdots & \vdots & \vdots \\ \frac{\partial r_N}{\partial x} & \frac{\partial r_N}{\partial y} & \frac{\partial r_N}{\partial \theta} \\ \frac{\partial \psi_N}{\partial x} & \frac{\partial \psi_N}{\partial y} & \frac{\partial \psi_N}{\partial \theta} \end{bmatrix} = \begin{bmatrix} -c_1 \cos \beta_1 & -c_1 \sin \beta_1 & 0 \\ 0 & 0 & -1 \\ \vdots & \vdots & \vdots \\ -c_1 \cos \beta_N & -c_1 \sin \beta_N & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{23}$$

where  $c_i$  is the sign function  $sign(C_i)$ .

- The matrix  $V_{odl}$  has size  $[(2 N) \times 3]$  and is equal to:

$$V_{odl} = \begin{bmatrix} \frac{\partial r_1}{\partial \varepsilon_\varphi} & \frac{\partial r_1}{\partial \varepsilon_r} & \frac{\partial r_1}{\partial \varepsilon_\psi} \\ \frac{\partial \psi_1}{\partial \varepsilon_\varphi} & \frac{\partial \psi_1}{\partial \varepsilon_r} & \frac{\partial \psi_1}{\partial \varepsilon_\psi} \\ \vdots & \vdots & \vdots \\ \frac{\partial r_N}{\partial \varepsilon_\varphi} & \frac{\partial r_N}{\partial \varepsilon_r} & \frac{\partial r_N}{\partial \varepsilon_\psi} \\ \frac{\partial \psi_N}{\partial \varepsilon_\varphi} & \frac{\partial \psi_N}{\partial \varepsilon_r} & \frac{\partial \psi_N}{\partial \varepsilon_\psi} \\ \frac{\partial \varepsilon_\varphi}{\partial \varepsilon_\varphi} & \frac{\partial \varepsilon_r}{\partial \varepsilon_\varphi} & \frac{\partial \varepsilon_\psi}{\partial \varepsilon_\varphi} \\ \frac{\partial \varepsilon_\varphi}{\partial \varepsilon_r} & \frac{\partial \varepsilon_r}{\partial \varepsilon_r} & \frac{\partial \varepsilon_\psi}{\partial \varepsilon_r} \\ \frac{\partial \varepsilon_\varphi}{\partial \varepsilon_\psi} & \frac{\partial \varepsilon_r}{\partial \varepsilon_\psi} & \frac{\partial \varepsilon_\psi}{\partial \varepsilon_\psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ \vdots & \vdots & \vdots \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (24)$$

- The matrix  $R_{odl}$  has size  $[(2 N) \times (2 N)]$  and is equal to:

$$R_{odl} = \begin{bmatrix} var(r_1) & 0 & 0 & \dots & 0 & 0 \\ 0 & var(\psi_1) & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & var(r_N) & 0 \\ 0 & 0 & \dots & 0 & 0 & var(\psi_N) \end{bmatrix} \quad (25)$$

### Identification of Kalman filter's parameters from the Lidar

This section describes the algorithm to extract line segments from a set of points based on the least square quadratic (LSQ) method. The point cloud was obtained from a scanner with an  $360^\circ$  field of view and a  $1^\circ$  angular resolution. The data collected by the RPLIDAR was first converted into Cartesian coordination and stored in an array according to the equation:

$$x = r \cos\left(\varphi - \frac{\pi}{180}\right); y = r \sin\left(\varphi - \frac{\pi}{180}\right) \quad (26)$$

where  $r$  and  $\varphi$  are the range and bearing from robot position to obstacles respectively.

Points that lie too close to robot position ( $<40$  cm) were treated as noise and were eliminated. Because the RPLIDAR scans from right to left, neighboring points have high chance to lie on a same landmark such as on the wall. The LSQ was applied to a group of  $N_1$  consecutive laser readings starting from the first points to find the best fit line of these points. A distance threshold  $disT$  was predefined and compared with the distance from each point in the group to the best fit line. If there were more than  $N_2$  of the points in the group (in our research,  $N_2 = 0,75N_1$ ), the distance of which is fewer than the  $disT$ , a line was detected and a new best fit line was calculated based on these points. The point with the smallest index would be saved as one end of the line segment. The slope  $m$  and intercept  $k$  of the line were computed using the formula:

$$m = \frac{\sum xy - \frac{(\sum x)(\sum y)}{n}}{\sum x^2 - \frac{(\sum x)^2}{n}}; k = \bar{y} - m\bar{x} \quad (27)$$

where  $n$  is the total number of data points,  $\bar{x}$  and  $\bar{y}$  are the mean of the  $x$ - and  $y$ - coordinates of the data points respectively. The algorithm would then be used to find the remaining points of the cloud to add more points to the line segment. Points that lie too far from their previous neighbors in the array would be ignored. The last point that met the distance threshold was then treated as the other end of the found line segment. The algorithm was repeated with the new group until all elements of the collected point cloud were checked.

Nevertheless, a fixed threshold  $disT$  did not work for all scans. Experiment results show that in about 10% of the scans, the threshold of the distance from one point to the best fit line should be larger in order to detect lines from data points. However, if the threshold applied for all scans was too large, extracted line segments would be not smooth and could affect the matching result. In this paper, a simple dynamic threshold was developed to solve this problem. A maximal threshold  $disT_{max}$  was defined as the largest distance to determine if a point belongs to a line. For each group of points, instead of comparing the distance from each point to the best fit line with fixed  $disT$ , all distances would be stored in an array and sorted in an ascending order. If the  $N_2$ th distance was smaller than the maximal threshold  $disT_{max}$ , it would be chosen as the distance threshold for this line segment. Therefore, the number of distance threshold in each scan would be the same as the number of detected line segments.

### Local and global maps matching

The extracted line segments of same landmarks from local and global maps are then matched together using a straight-forward algorithm. The extracted line segments from global map  $G_j$  and local map  $L_j$  are described as follows:

$$G_j = [x_{1G,j} \ x_{2G,j} \ y_{1G,j} \ x_{2G,j}] \ j = 1 \dots n_G$$

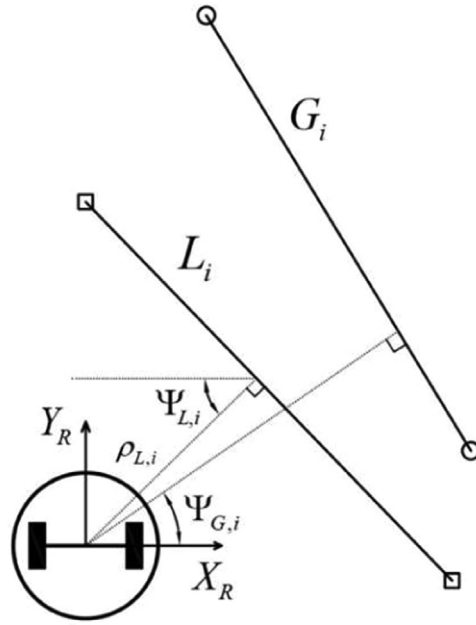


Fig. 13. The range and bearing of global and local line segments to robot's position in robot's coordinate.

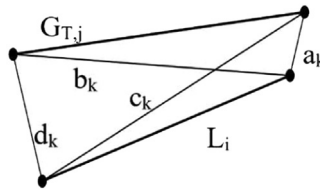


Fig. 14. The overlapping parameters between global and local line segments.

$$L_i = [x_{1L,i} \ x_{2L,i} \ y_{1L,i} \ y_{2L,i}] \quad i = 1 \dots n_L \tag{28}$$

where  $(x_1, y_1)$  and  $(x_2, y_2)$  are Cartesian coordinate of two end points of extracted line segments respectively,  $n_G$  and  $n_L$  are number of extracted global and local line segments.

The detected global line segments are first transformed into local map with the equation:

$$G_{T,j} = G \left( \begin{bmatrix} x_{1G,j} & y_{1G,j} \\ x_{2G,j} & y_{2G,j} \end{bmatrix} - \begin{bmatrix} x_0 & y_0 \\ x_0 & y_0 \end{bmatrix} \right) \tag{29}$$

where  $G = \begin{pmatrix} \cos \varphi_0 & -\sin \varphi_0 \\ \sin \varphi_0 & \cos \varphi_0 \end{pmatrix}$  is the conversion matrix,  $[x_0 \ y_0 \ \theta_0]^T$  are indicators of robot's position in global map estimated by Odometry. Each local line segment  $L_i$  is compared with all transformed global line segments  $G_{T,j}$ , and two line segments are considered as "matched" if their range and bearing to robot position are approximately the same and the overlapping rate between them is less than previous defined threshold [27].

The range and bearing of global and local line segments to robot position in robot's coordinate are presented as  $(\rho_{G,j}, \psi_{G,j})$  and  $(\rho_{L,i}, \psi_{L,i})$ , respectively as shown in Fig. 13.

The overlapping rate between the local line segment  $L_i$  and the transformed global line segment  $G_{T,j}$  is defined as follows:

$$\begin{aligned} O_k(a_k, b_k) &= |a_k + b_k - \bar{G}_{T,j}| \\ O_k(c_k, d_k) &= |c_k + d_k - \bar{G}_{T,j}| \\ k &= (j - 1)n_L + i \\ k &= 1, 2, \dots, n_L n_G - 1, n_L n_G \quad k = 1, 2, \dots, n_L n_G - 1, n_L n_G \end{aligned} \tag{30}$$

where  $G_{T,j}$  is the length of transformed global line segment;  $a_k, b_k, c_k$  and  $d_k$  are Euclidean distances between the end points of the line segments  $L_i$  and  $G_{T,j}$  as shown in Fig. 14.

The inequations below represent conditions for matching local and transformed global line segments in robot's coordinate. Two line segments  $L_i$  and  $G_{T,j}$  are matched if all following conditions are met:

$$\begin{cases} O_k(a_k, b_k) < T \\ O_k(c_k, d_k) < T \\ (\rho_{L,i} - \rho_{G,j})^2 < T_\rho \\ (\psi_{L,i} - \psi_{G,j})^2 < T_\psi \end{cases} \tag{31}$$

where  $T, T_\rho, T_\psi$ , are predefined threshold.

*Estimation of line parameter's covariances*

In order to compute the measurement covariance for the EKF, each extracted line segment could be presented as  $(\rho, \psi)$  parameters where  $\rho$  stands for the perpendicular distance from robot's position to the line and  $\psi$  is the line orientation:

$$\rho = \frac{|-k|}{\sqrt{m^2 + 1}}; \psi = \arctan 2\left(\frac{k}{1 + m^2}, \frac{-mk}{1 + m^2}\right) \tag{32}$$

where  $(m,k)$  are slope and intercept of detected line which were computed in (33).

From the solution give by Derich [28], the measurement covariance matrix,  $R$ , can be calculated with the assumption that each data point has the same Cartesian uncertainty:

$$R_i = \frac{a_i\sigma_{yy}^2 - b_i\sigma_{xy}^2 + c_i\sigma_{xx}^2}{(a_i - c_i)^2 + b_i^2} \begin{pmatrix} 1 & -e \\ -e & e^2 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \frac{\sigma_{yy}^2 \cos^2 \varphi_i + \sigma_{xx}^2 \sin^2 \varphi_i - 2\sigma_{xy}^2 \sin \varphi_i \cos \varphi_i}{n} \end{pmatrix} \cong \begin{bmatrix} \text{var}(r_i) & 0 \\ 0 & \text{var}(\psi_i) \end{bmatrix}$$

where :  $e_i = \bar{y} \cos \psi_i - \bar{x} \sin \psi_i$ ;  $\varphi_i = \left(\psi_i + \frac{\pi}{\bar{x}}\right)$ ;  $\bar{x} = \frac{1}{n}x_j$ ;  $\bar{y} = \frac{1}{n}y_j$ ;  $a_i = \sum (x_j - \bar{x})^2$ ;  $b_i = 2 \sum (x_j - \bar{x})(y_j - \bar{y})$ ;  
 $c_i = \sum (y_j - \bar{y})^2$  \tag{33}

*Control the robot to track the trajectory*

The main goal is to control the mobile robot to track a certain trajectory. A different trajectory with a path with time constraints added to it, which makes the control target not only minimize the distance between the robot and the path, but also to ensure the travel time. We define the actual robot state as:  $X = [x \ y \ \theta]^T$  and according to the pattern trajectory is:  $X_r = [x_r \ y_r \ \theta_r]^T$

When the robot moves, the error will appear:

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \tag{34}$$

From the kinetic and derivative model (40), we get the error model as follows:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \cos e_3 & 0 \\ \sin e_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{35}$$

where  $v_r, \omega_r$  are the linear and angular velocity of the robot according to the trajectory.

The controller for the robot is built as follows:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r \cos e_3 \\ \omega_r \end{bmatrix} + \begin{bmatrix} v_{fb} \\ \omega_{fb} \end{bmatrix} \tag{36}$$

where  $v_{fb}, \omega_{fb}$  are the feedback signal of the controller, are selected as follows:

$$\begin{aligned} v_{fb} &= k_1 e_1 \\ \omega_{fb} &= k_2 v_r \frac{\sin e_3}{e_3} e_2 + k_3 e_3 \end{aligned} \tag{37}$$

where  $k_1 > 0, k_2 > 0$  and  $k_3$  are the coefficients.

Then the control law for tracing the trajectory will be rewritten as:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 v_r \frac{\sin e_3}{e_3} e_2 + k_3 e_3 \end{bmatrix} \tag{38}$$

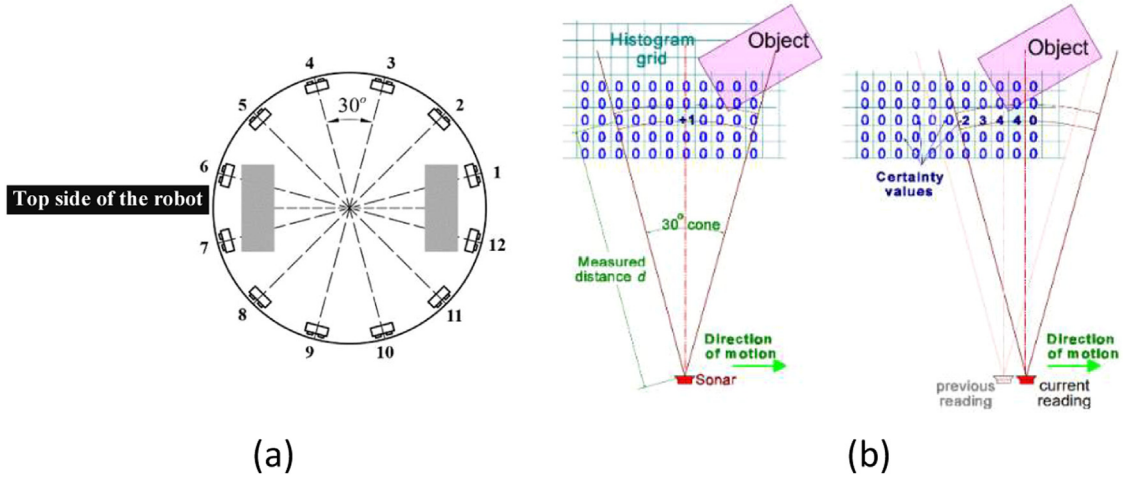


Fig. 15. (a). Layout diagram of ultrasonic sensors; (b) Update the Histogram Grid by continuously sampling the ultrasonic sensors while moving.

*Controlling the robot to avoid obstacles with VFH+ using ultrasonic sensors*

Along the way, the robot must be able to detect and avoid unexpected obstacles. For these cases, a proximity sensor system is used. That is 12 ultrasonic distance sensors installed on the robot as shown in Fig. 15(a), allowing to detect obstacles in front and two sides of the robot.

The VFH+ method uses a histogram grid to map the environment around the robot. This map is continuously updated with distance to obstacle data obtained from the ultrasonic sensors mounted on the robot as shown in Fig. 15(b). The method will find the optimal direction of movement when encountering obstacles, and appropriate velocity control for the robot (linear velocity, angular velocity).

*Histogram grid*

In this step, a two-dimensional Cartesian histogram grid (C) is generated containing the information transmitted from the ultrasonic sensors (the selected C has dimensions of 81 × 81 and a resolution of 0.1 m/cell). Each grid cell C[i, j] contains a value c<sub>i,j</sub> representing the reliability of the existence of an obstacle at the coordinate position (i, j). The grid is filled using distances measured by ultrasonic sensors, each of which updates the value of only one cell at a time.

*Polar histogram*

In the second step, a window C\* has a fixed orientation, whose center is attached to the robot and moves along its motion (the selected C\* is 33 × 33). It is called the active window and it overlays the histogram grid C at the robot’s current position. The content of each active cell in the map grid is treated as an obstacle vector, its direction β being towards the center of the active region (robot center point (RCP)):

$$\beta_{i,j} = \tan^{-1} \left( \frac{y_i - y_o}{x_i - x_o} \right) \tag{39}$$

The vector magnitude of an active cell C[i, j] is given by:

$$m_{i,j} = c_{i,j}^2 (a - bd_{i,j}^2) \tag{40}$$

where a and b are positive constants, c<sub>i,j</sub> is the certainty value of an active cell (i, j), d<sub>i,j</sub> distance from each cell to the center of the active area (RCP), m<sub>i,j</sub> is the magnitude of the obstacle vector at the cell (i, j), (x<sub>0</sub>, y<sub>0</sub>) is the coordinate of the robot center, (x<sub>i</sub>, y<sub>i</sub>) is the coordinate of each cell (i, j).

Based on the obstacle vectors, the Primary Polar Histogram H<sup>p</sup> is built. H<sup>p</sup> has an arbitrary angular resolution α so that n = 360°/α is an integer. Each angular sector k corresponds to a discrete angle kα.

For each obstacle cell, the enlargement angle γ<sub>i,j</sub> is defined by:

$$\gamma_{i,j} = \arcsin \frac{r}{d_{i,j}} \tag{41}$$

where r is the radius of the robot’s safe zone.

For each sector k, the polar obstacle density is then calculated by:

$$H_k^p = \sum m_{i,j} h'_{i,j} \tag{42}$$

with:

$$h'_{i,j} = \begin{cases} 1 & \text{if } k\alpha \in [\beta_{i,j} - \gamma_{i,j}, \beta_{i,j} + \gamma_{i,j}] \\ 0 & \text{otherwise} \end{cases} \tag{43}$$

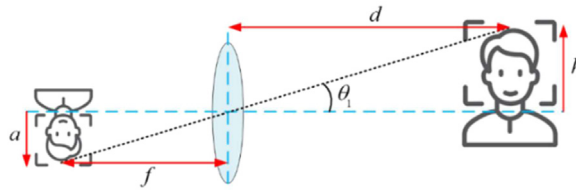


Fig. 16. The method of determining the focal length of the camera.

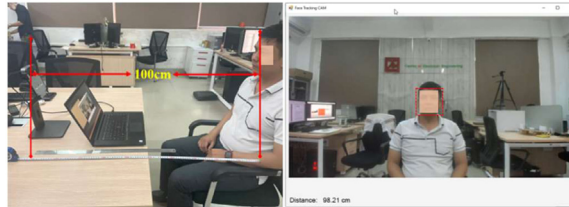


Fig. 17. Determine the distance using the triangle similarity method.

From the Polar Histogram we can find out which direction is free so that the robot can move in that direction without encountering obstacles. The polar chart is updated according to the following rules:

$$H_{k,i}^b = \begin{cases} 1 & \text{if } H_k^p > \tau_{high} \\ 0 & \text{if } H_k^p < \tau_{low} \\ H_{k,i-1}^b & \text{otherwise} \end{cases} \quad (44)$$

where  $\tau_{high}$  and  $\tau_{low}$  are high thresholds and low thresholds are chosen to ensure that obstacles too far away from the robot do not affect the robot's movement, even though the obstacle is in front of the moving direction.

### Patient identification

To guide the patient by maintaining the distance between the robot and the patient during the robot movement, we apply the face detection technique [29]. From there, determine the height of the face in the image and calculate the distance from the image sensor to the face of the person standing in front of the camera, applying the principle of similarity of 2 triangles as shown in Figs. 16 and 17.

Determine the distance  $d$  from the formula  $a/f = \tan\theta_1 = h/d$ . We can calculate the distance of  $d$  as:

$$d = h \times \frac{f}{a} \quad (45)$$

where  $a$  is the height of the human face in the image,  $f$  is the focal length,  $h$  is the height of the actual human face,  $d$  is the distance from the person to the camera.

### Disinfection and sterilization

The sterilization and disinfection components of the robot are described in Fig. 1(a) and 1(b). In which the UV lamp is equipped at the back for the purpose of sterilization. UV lamp with a power of 36 W, wavelength 253.7 nm, the glass shell of the lamp has a filter function of 185 nm wavelength to create ozone in the air.

Disinfection is carried out as follows: in front of the robot body, there is a container of disinfectant solution with a capacity of 20l Disinfectant solution containing 0.1% active Chlorine (as recommended [30]). And will be evenly sprayed on the surface of the disinfected special treatment room with a dosage of 0.3 – 0.5l/m<sup>2</sup> by two nozzles that have been arranged under the front chassis of the robot. With the equipment arrangement as shown in Fig. 1(b), the disinfectant solution will be sprayed forward and diffused lower than the patient's lying position as recommended by medical authorities. With 1 meter of movement of the robot, the amount of sprayed solution will be equivalent to 1 ml (at each nozzle). Thus, with the carrying of 20l of the solution, the robot will disinfect an entire room with a charge of 60m<sup>2</sup> over 150 times.

In addition to disinfection, the robot is also specially equipped with a floor cleaning function according to the robot's movement diagram. Below the robot is arranged 4 rotating brushes with a diameter of 170 mm. The brush rotates passively through the drive force from the wheel, and the mopping mechanism is pressed against the floor by a spring, to ensure that there are no gaps or gaps between the floor and the brush surface. The brush can be simply removed for cleaning or replacement.



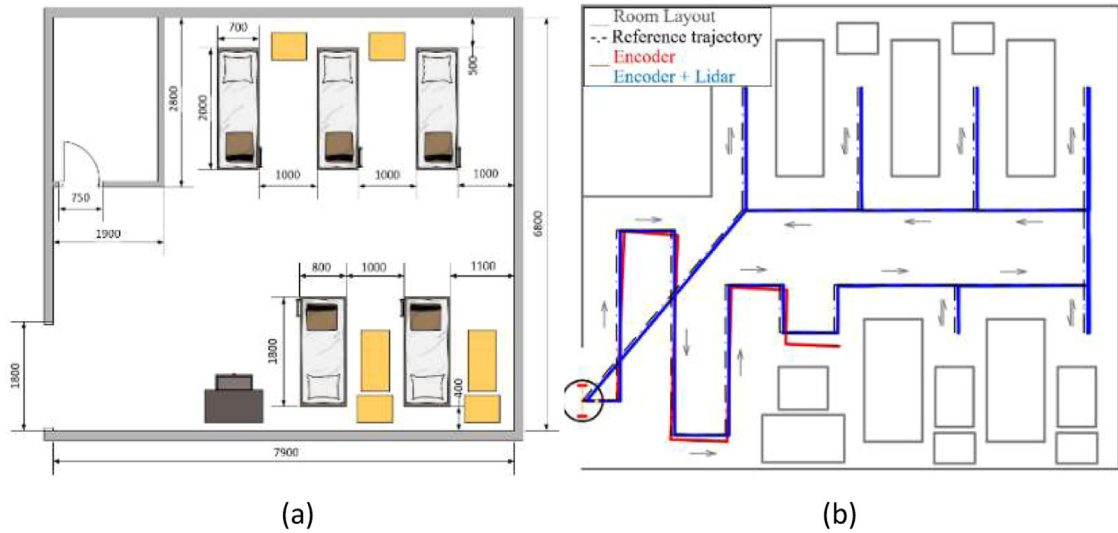


Fig. 18. (a) Medical room plan; (b) The path of the robot in different configurations when following the trajectory.

*SpO2 sensor and AMG8833 sensor*

First of all, blood oxygen saturation (SpO2) was calculated using a formula supplied in the Maxim Integrated™ sample code [24]. The AC and DC

Component of the pulsative waveform was calculated for both the red and infrared channels and stored in integer variables ( $AC_{Red}$ ,  $AC_{IR}$ ,  $DC_{Red}$ ,  $DC_{IR}$ ) as a mean of 5 consecutive peaks/valleys. A ratio (R) of the AC and DC was then calculated from the mean AC and DC values using Eq. (46), and the SpO2 value was calculated by the Eq. (47):

$$R = \frac{AC_{Red} \div DC_{Red}}{AC_{IR} \div DC_{IR}} \tag{46}$$

$$SpO2 = -45,060R^2 + 30,345R + 94,845 \tag{47}$$

Next, the AMG8833 sensor measures the temperature of an object based on the thermal radiation emitted by that object. The Stefan-Boltzmann law describes the power emitted by an object using the formula (48):

$$P = \epsilon A \sigma T^4 \tag{48}$$

where  $P$  is the power radiated from an object;  $T$  is the temperature of the object;  $A$  is surface area;  $\sigma$  is Stefan–Boltzmann constant,  $\epsilon$  is the emissivity of the object (it gets values from 0 to <1 for objects which do not absorb all incident radiation - grey object; if it is an absolute black object then equal 1).

Applying Moghaddam’s formula [31], we calculate the temperature of a grey body as follows:

$$V_{out} = k(T_o^4 - T_s^4) \tag{49}$$

where  $V_{out}$  is the output voltage of the sensor,  $T_o$  is the surface temperature of the object being measured,  $T_s$  is the temperature of the sensor’s thermistor,  $k$  is the empirical constant representing the parameters  $A$ ,  $\epsilon$ ,  $\sigma$  as well as the electronic noise that may exist during the measurement (calibrating the sensors).

Based on (49), the surface temperature of the object will be calculated by the formula (50):

$$T_o = (V_{out}/k + T_s^4)^{1/4} \tag{50}$$

**Experimental results**

*Experimenting with a robot that follows the trajectory and avoids obstacles*

With the shape and size of the hospital room as shown in Fig. 18(a), we can draw a map and set the trajectory for the robot as shown in Fig. 18(b), the arrow direction is the direction of movement. The robot will both follow the trajectory and disinfect (the disinfecting road surface is 0,65 m wide), when approaching the hospital beds, the robot will collect data without contacting the patient. The robot is tested with sensor fusion cases without lidar and when lidar is present. In order to avoid obstacles appearing unexpectedly on the robot’s trajectory, the Vector Field Histogram+ (VFH+) method [21–23] is applied. The real path of the robot is actually measured.

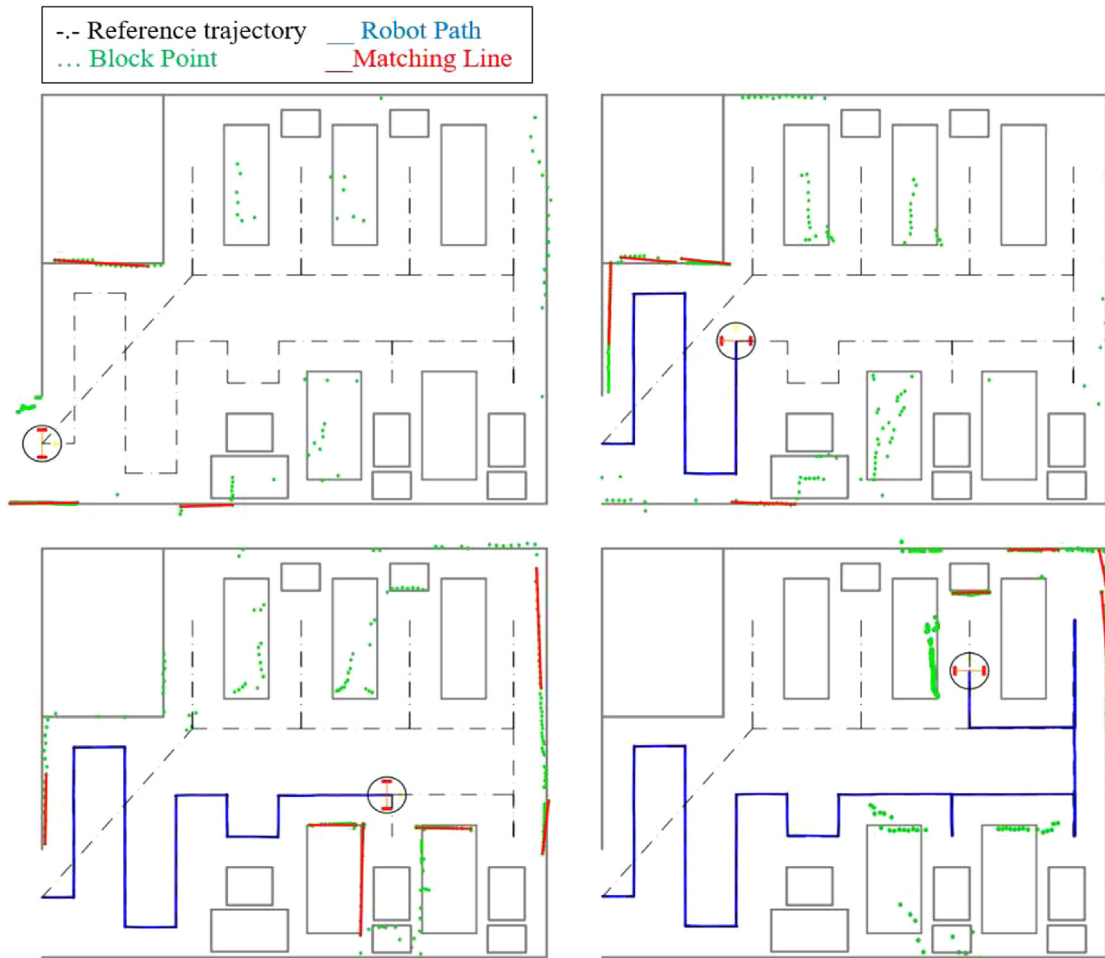


Fig. 19. The data is obtained from the Lidar and the line segments are matched to the map.

In Fig. 19, the result showing the robot’s path in 3 cases: the green line is the real trajectory, the red line is obtained when using only Encoder and the blue line is when adding the Lidar sensor. When using only Encoder, the robot moves away from the real trajectory when the trajectory is longer due to the accumulation of accumulated error noise, leading to the robot colliding with equipment in the room before going all the way and having to end the journey. When fusing both Encoder and Lidar sensors, the result for the robot’s path is closest to the true trajectory by the positioning method based on the Kalman filter. This method is based on the data obtained from the Lidar which is matched to the global map.

Fig. 19 shows the data obtained from the lidar sensor along the trajectory, the red lines are the lines extracted from the points and matched to the map. The Kalman filter will use the matched segments to compute the values of the matrices for the filter to locate.

Experimental results show that the effectiveness of the Kalman filter when adding more sensors, the position estimation results are closer to the real line values.

Fig. 20 provides the result of moving when the robot encounters an obstacle blocking the trajectory. The robot uses the VFH+ method to avoid obstacles based on data from ultrasonic sensors mounted on the robot. The results show that the robot avoids obstacles effectively and keeps track of its trajectory after avoiding obstacles.

*Collect patient temperature and SpO2 data*

Based on the robot’s movement map, stop points are established along the way for the robot to stop and collect sensor data as shown in Fig. 21. We arranged the tester to lie on the bed corresponding to the stop points that will perform sensor data collection.

The data collection process follows the following steps: Every time the robot moves to a stop, it will perform 2 to 3 rotations at the center.

- First, at stop position 1, the robot will rotate to the right at an angle of 45° relative to the direction of the path following the red arrow. This is the direction that is determined from the center of the image sensor to the face of the patient lying on the bed (the

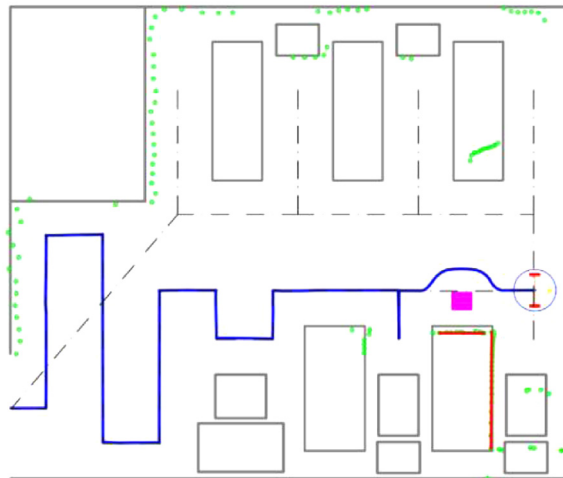


Fig. 20. Robot avoids obstacles by the VFH+ method.

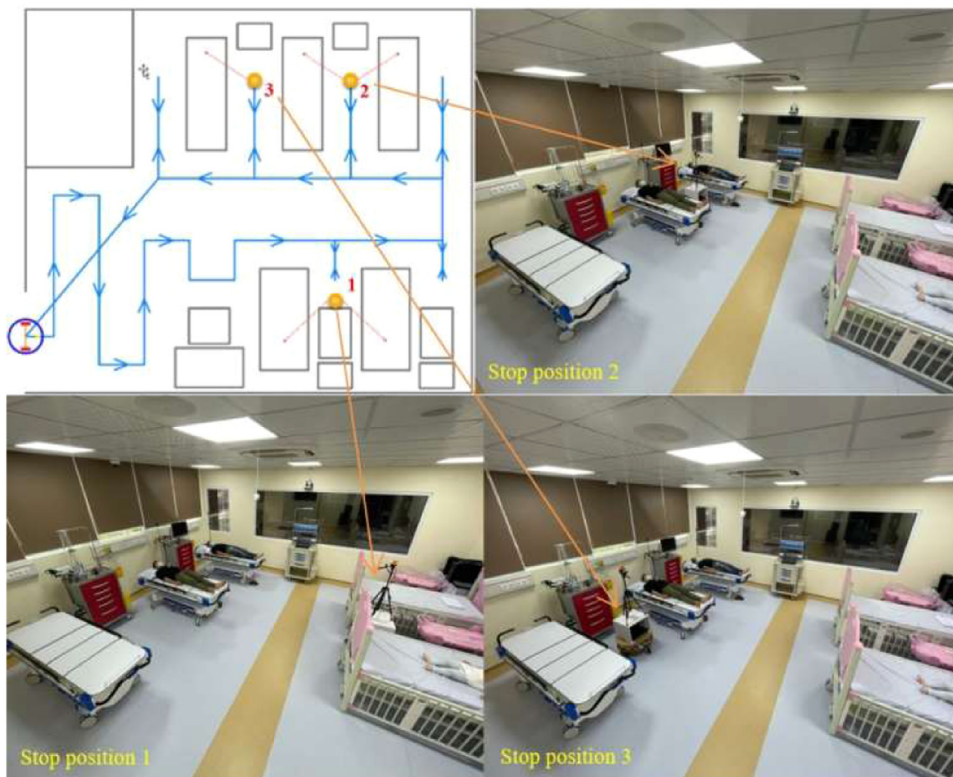


Fig. 21. Map of breakpoints for robot data collection; The yellow point marks the stop position; Directional arrow for AMG8833 sensor node that collects body temperature data.

patient’s lying position are supine, upright, and facing the ceiling). At each rotation, the robot stops for 3 s (determined through actual testing) so that the AMG8833 sensor reads the temperature data of the right patient. The next turn to the left 90° to collect the temperature of the patient on the left. In the third rotation, the robot continues to rotate left by an angle of 135° to come out of the stop and continue the route.

- At stop position 2, the robot performs the same action as stop position 1 to collect data of 2 patients on both sides.
- At stop position 3, the robot only performed 2 left rotations at an angle of 45° for the first time and an angle of 135° for the second time to collect data of the left patient.

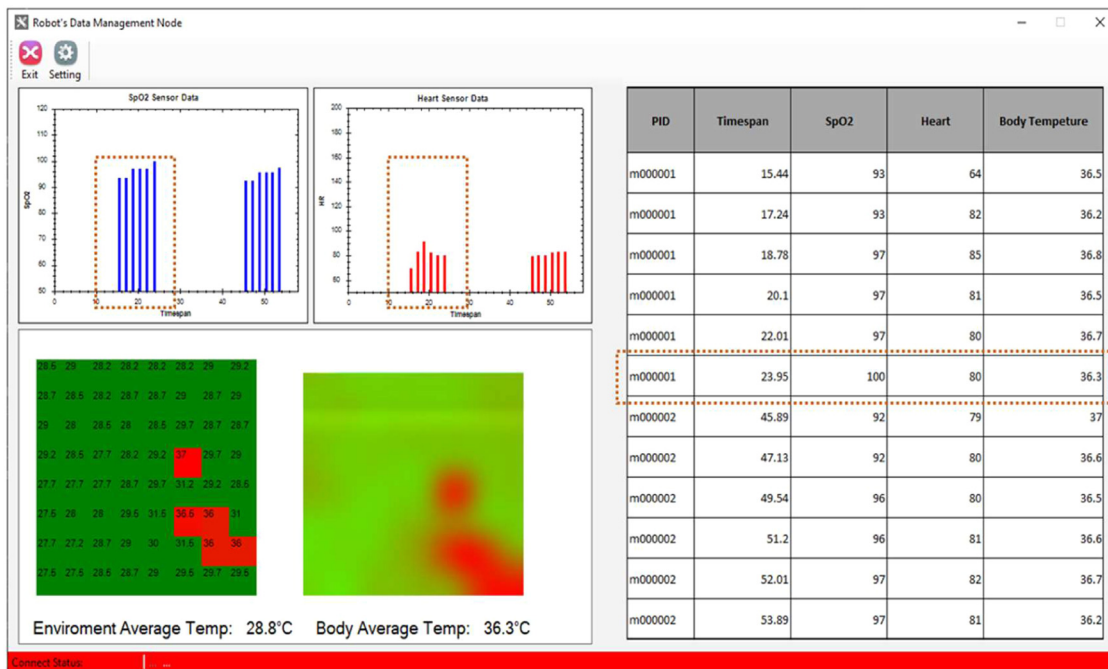


Fig. 22. Final data sample of test participants in bed 2.

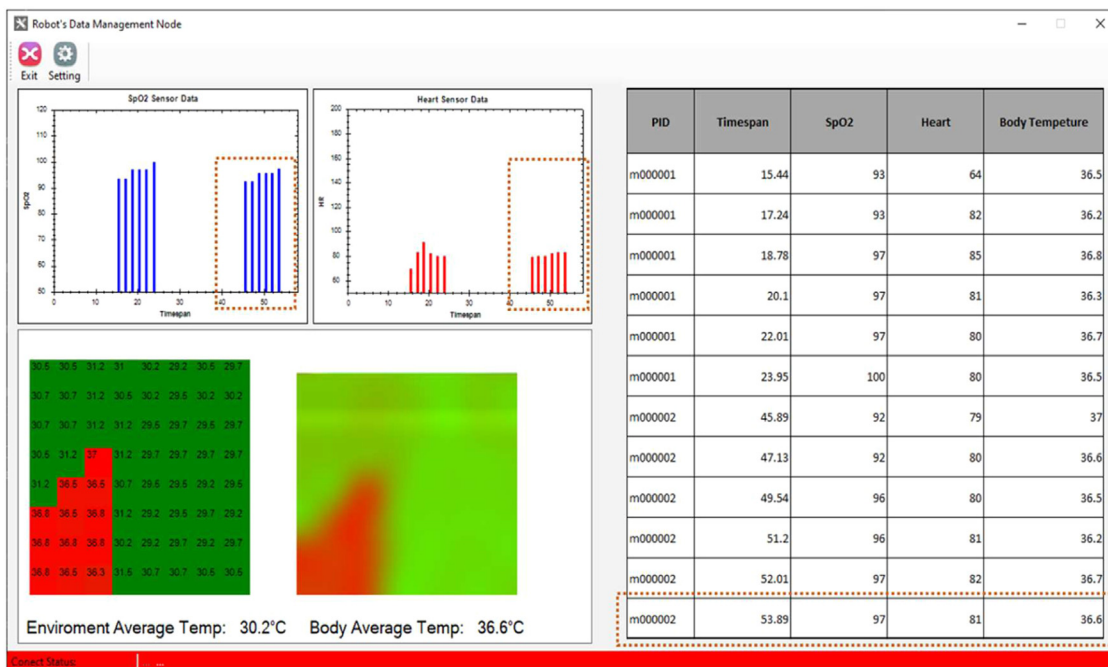


Fig. 23. Final data sample of test participants in bed 3.

- After each rotation, the sensor data will be sent to the computer. As shown in Figs. 22 and 23, we can see that every time the robot stops at a stop, the sensor data will be sent to the computer every 0,5 s. In there:
- For the temperature graph, the red pixels represent temperatures above 35 °C, which is also the location of the tester. The group of red pixels is skewed to the right for the test participant in bed 2 and to the left for the test participant in bed 3.
- For the SpO2 chart, every 3 s, there will be 06 data samples sent to the computer. The space between the 2 groups of data on the SpO2 and Heart Rate charts shows the time it took the robot to rotate 45° from bed 2 direction to bed 3 direction in 06 s.

## Conclusions and future developments

This article presents a novel design of a mobile robot, called CEE-IMR. The research provides a smart disinfection system and indirect measurement system for mobile robots supporting Covid-19 or contagious disease treatments in hospitals. Smart disinfection robots can improve the shortcomings of existing sterilization methods, improve sterilization quality, and reduce the probability of infection. The robot model uses intelligent photo recognition, independent sensors to monitor the sterilization process in real-time, and intelligently plan moves independently. The evaluation of the results proves the effectiveness of the proposed method. Automated air and surface disinfection and non-human patient monitoring have reduced unnecessary direct contact between infected people and healthcare workers. The results can be developed for real applications in hospitals and show promise.

In the future developments, we will deploy more AI techniques and security methods into the systems. The system can operate more independently and adapt with sudden changes in flexible working environments. More facilities could be embedded in the system to provide necessary services for either patients and doctors in hospitals.

### Method validation

All Results and Discussion are **provided** in the above Section, Method details. The results are from real robots running in hospitals that can validate the methods. The corresponding discussions are provided at every steps to validate the methods.

### Limitations

The current limitation of this work is that the robot has not been trained with AI techniques to be able to adapt with either the environments or the people around. This can lead to more functions for the robot and be more effective while working in contagious disease environments.

### Ethics statements

None

### CRedit author statement

Hoang T. Tran, Dong TL. Tran: Conceptualization, Methodology, Software. Minh T. Nguyen, Thang C. Vu: Validity tests, Data curation, Writing- Original draft preparation. Hoang T. Tran: Visualization, Investigation. Hoang T. Tran, Thang C. Vu: Supervision. Dong TL. Tran: Software, Validation. Minh T. Nguyen, Hoang T. Tran: Writing- Reviewing and Editing.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Data availability

Data will be made available on request.

### Acknowledgments

The authors would like to thank Thai Nguyen University of Technology, Thai Nguyen University, Ministry of Education and Training (Project [B2023-TNA-16](#)), Viet Nam for the support.

### References

- [1] M.T. Nguyen, H.M. La, K.A. Teague, Collaborative and compressed mobile sensing for data collection in distributed robotic networks, *IEEE Trans. Control Netw. Syst.* 5 (4) (2017) 1729–1740.
- [2] M.T. Nguyen, H.R. Boveiri, Energy-efficient sensing in robotic networks, *Measurement* 158 (2020) 107708.
- [3] Bogue, R. (2020). Robots in a contagious world. *Industrial Robot: the international journal of robotics research and application*.
- [4] N. Zhu, D. Zhang, W. Wang, X. Li, B. Yang, J. Song, W. Tan, A novel coronavirus from patients with pneumonia in China, 2019, *N. Engl. J. Med.* (2020).
- [5] Q. Li, X. Guan, P. Wu, X. Wang, L. Zhou, Y. Tong, Z. Feng, Early transmission dynamics in Wuhan, China, of novel coronavirus-infected pneumonia, *N. Engl. J. Med.* (2020).
- [6] C. Wang, P.W. Horby, F.G. Hayden, G.F. Gao, A novel coronavirus outbreak of global health concern, *Lancet* 395 (10223) (2020) 470–473.
- [7] D.S. Hui, E.I. Azhar, T.A. Madani, F. Ntoumi, R. Kock, O. Dar, E. Petersen, The continuing 2019-nCoV epidemic threat of novel coronaviruses to global health—the latest 2019 novel coronavirus outbreak in Wuhan, China, *Int. J. Infect. Dis.* 91 (2020) 264–266.
- [8] C.W. Park, S.W. Seo, N. Kang, B. Ko, B.W. Choi, C.M. Park, H.J. Yoon, Artificial intelligence in health care: current applications and issues, *J. Korean Med. Sci.* 35 (42) (2020).
- [9] I. Karabegović, V. Doleček, The role of service robots and robotic systems in the treatment of patients in medical institutions, in: *Advanced Technologies, Systems, and Applications*, 2017, pp. 9–25.

- [10] B. Kumar, L. Sharma, S.L. Wu, Job allocation schemes for mobile service robots in hospitals, in: 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, 2018, pp. 1323–1326.
- [11] H. Samani, R. Zhu, Robotic automated external defibrillator ambulance for emergency medical service in smart cities, *IEEE Access* 4 (2016) 268–283.
- [12] N. Koceska, S. Koceski, P. Beomonte Zobel, V. Trajkovik, N. Garcia, A telemedicine robot system for assisted and independent living, *Sensors* 19 (4) (2019) 834.
- [13] T. Mettler, M. Sprenger, R. Winter, Service robots in hospitals: new perspectives on niche evolution and technology affordances, *Eur. J. Inform. Syst.* 26 (5) (2017) 451–468.
- [14] E. Prassler, A. Ritter, C. Schaeffer, P. Fiorini, A short history of cleaning robots, *Autonom. Robot.* 9 (3) (2000) 211–226.
- [15] F. Astrid, Z. Beata, E. Julia, P. Elisabeth, D.E. Magda, The use of a UV-C disinfection robot in the routine cleaning process: a field study in an Academic hospital, *Antimicrob. Resist. Infect. Control* 10 (1) (2021) 1–10.
- [16] K. Nosirov, S. Begmatov, M. Arabboev, K. Medetova, Design of a model for disinfection robot system, in: 2020 International Conference on Information Science and Communications Technologies (ICISCT), IEEE, 2020, pp. 1–4.
- [17] A. Vyshnavi, A. Manasa, C. Hamsika, P. Shalini, UV disinfection robot with automatic switching on human detection, *Eai Endors. Transact. IoT* 6 (23) (2020).
- [18] N. Mikhailovskiy, A. Sedunin, S. Perminov, I. Kalinov, D. Tsetserukou, Ultrabot: autonomous mobile robot for indoor uv-c disinfection with non-trivial shape of disinfection zone, in: 2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, 2021, pp. 1–7.
- [19] Ministry of Health: Decision on the issuance of Guidelines for diagnosis and treatment of COVID-19 caused by a new strain of Corona virus (SARS-CoV-2), In: 3416/QD-BYT 14/7/2021.
- [20] S. Shaikh, N. Akhter, R. Manza, Current trends in the application of thermal imaging in medical condition analysis, *Int. J. Innov. Technol. Explor. Eng* 8 (8) (2019) 2708–2712.
- [21] I. Ulrich, J. Borenstein, VFH+: reliable obstacle avoidance for fast mobile robots, in: In Proceedings. 1998 IEEE international conference on robotics and automation (Cat. No. 98CH36146), 2, IEEE, 1998, pp. 1572–1577.
- [22] T.T. Hoang, D.T. Hiep, P.M. Duong, N.T.T. Van, B.G. Duong, T.Q. Vinh, Proposal of algorithms for navigation and obstacles avoidance of autonomous mobile robot, in: 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA), IEEE, 2013, pp. 1308–1313.
- [23] D. Díaz, L. Marín, VFH+ D: an improvement on the VFH+ algorithm for dynamic obstacle avoidance and local planning, *IFAC-PapersOnLine* 53 (2) (2020) 9590–9595.
- [24] MAXREFDES# 117, INTEGRATED, M. (2020): Heart-Rate and Pulse-Oximetry Monitor. Disponivel em: <https://www.maximintegrated.com/en/design/reference-designcenter/systemboard/6300.html>. Acesso em, 13.
- [25] K. Goyal, K. Agarwal, R. Kumar, Face detection and tracking: using OpenCV, in: In 2017 International conference of Electronics, Communication and Aerospace Technology (ICECA), 1, IEEE, 2017, pp. 474–478.
- [26] S. Moghaddam, J. Lawler, J. Currano, J. Kim, Novel method for measurement of total hemispherical emissivity, *J. Thermophys. Heat. Trans.* 21 (1) (2007) 128–133.
- [27] Ministry of Health: Decision on promulgating Guidance on SARS-CoV-2 infection prevention in medical examination and treatment establishments, In: 5188/QD-BYT 14/12/2020.
- [28] T.T. Hoang, P.M. Duong, N.T.T. Van, D.A. Viet, T.Q. Vinh, Multi-sensor perceptual system for mobile robot and sensor fusion-based localization, in: 2012 International Conference on Control, Automation and Information Sciences (ICCAIS), IEEE, 2012, pp. 259–264.
- [29] T.T. Hoang, P.M. Duong, N.T.T. Van, D.A. Viet, T.Q. Vinh, Development of an EKF-based localization algorithm using compass sensor and LRF, in: 2012 12th International Conference on Control Automation Robotics & Vision (ICARCV), IEEE, 2012, pp. 341–346.
- [30] L. Teslić, I. Škrjanc, G. Klančar, EKF-based localization of a wheeled mobile robot in structured environments, *J. Intell. Robot. Syst.* 62 (2) (2011) 187–203.
- [31] R. Deriche, R. Vaillant, O.D. Faugeras, From noisy edge points to 3D reconstruction of a scene: a robust approach and its uncertainty analysis, in: *Theory And Applications Of Image Analysis: Selected Papers from the 7th Scandinavian Conference on Image Analysis*, 1992, pp. 71–78.