Medical Telerobotic: IRAPs SHaRE-aGIVeR

Noppadol Pudchuen[1], Jiraphan Inthiam[2], Wisanu Jitviriya[3]
Amornphun Phunopas[4], Chirdpong Deelertpaiboon[5]
Center of Innovative Robotics and Advanced Precision Systems: iRAPs,
Faculty of Engineering, King Mongkut's University of Technology North Bangkok,
1518 Pracharat 1 Road,Bangsue, Bangkok 10800, Thailand

Aran Blattler[6]
Department of Mechanical Information Science and Technology
Kyushu Institute of Technology,
680-4 Kawazu, Iizuka, Fukuoka 820-0067, Japan
E-mail: noppadol.p@eng.kmutnb.ac.th[1], jiraphan.i@eng.kmutnb.ac.th[2], wisanu.j@eng.kmutnb.ac.th[3],
amornphun.p@eng.kmutnb.ac.th[4], chirdpong.d@eng.kmutnb.ac.th[5], aran.blattler703@mail.kyutech.jp[6]
https://iraps.eng.kmutnb.ac.th

Abstract

The Coronavirus disease 2019 (COVID-19) pandemic has affected the global population. In particular, the medical personnel in direct contact with patients have been exposed to high risk. To reduce the spread of COVID-19 and protect health-care workers and patients, we would like to present the fully automated medical telerobot as the IRAPs SHaRE-aGIVeR robot. Our robot is capable of generating both 2D and 3D maps automatically, delivering medical supplies, food, or medical devices such as blood pressure monitors, pulse oximeters, and so on. In addition, the user interface system is also vital part. Users are able to connect and control the robot using a computer, a mobile device, or a tablet via the wireless network which is installed inside the robot. Currently, our robots are being operated on with medical personnel at hospitals, regarding the feedback from the formal caregivers that can prove our robot's efficiency in reducing the risk of COVID-19 spread.

Keywords: Medical Telerobotic, Perception, Planning, IRAPs SHaRE-aGIVeR.

1. Introduction

On 30th January 2020, the COVID-19 pandemic was declared as "Global Emergency" by The World Health Organization (WHO) because the virus had spread rapidly [1]. The COVID-19 pandemic has brought along several threats, including economic and social disruption. The group of people at high risk of catching COVID-19 are health-care workers, who spend much time up close with the patient. Therefore, the technologies involving medical robotics systems are mentioned to control and reduce infection spread to a large population [2].

Considering the current COVID-19 pandemic situation, robots are well suited for caring for the well-being of patients, sharing the workload of health-care workers in hospitals such as utilizing them for cleaning and food preparation jobs in infected hazardous areas. Robots are mainly classified with various applications in

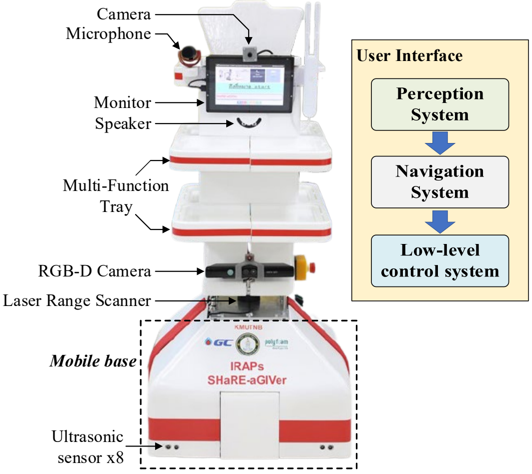

Fig.1 IRAPs SHaRE-aGIVeR robot

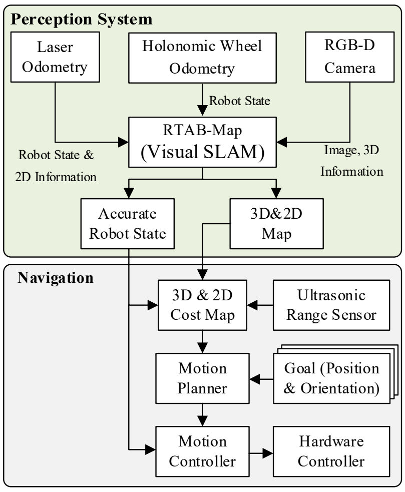

Fig.2 Perception Module and Navigation system

healthcare and related fields. **Receptionist robots** are used at a hospital's reception to give information about various sections of the hospital and guide visitors and patients [3]. **Nurse robots** are assigned to assist doctors in the hospital in the same manner as human nurses. These robots are commonly used in nursing and healthcare individuals undergo high stress and exhaustion due to patient load. For example, Robear: a robotics nurse to lift patients in Japan [4], Moxi: nursing robot placing medicines in bins [5], Paro robot and Pepper robot are used to assisting elderly patients in providing therapeutic assistance. **Serving robots** are used to deliver food and beverages, dispensing of drugs, or removing unclean laundry etc., inside the hospital [6],[7] and so on. For new alternative of the health-care robotics system, we would like to propose the fully automated medical telerobot as the IRAPs SHaRE-aGIVeR robot. Our robot is a holonomic movement platform, including the capability of generating both 2D and 3D maps automatically, path planning and navigation system, delivering medical supplies, food, or medical devices (e.g., blood pressure monitors, pulse oximeters) to patients.

Our paper is organized as follows: Chapter 2 explains the IRAPs SHaRE-aGIVeR robot overview system, which has three systems. In Chapter 3, our results are shown how our robot can operate with medical personnel at hospitals. In the final section, we conclude the paper and describe our future works.

2. Overview system of IRAPs SHaRE-aGIVeR

The robotic platform consists of the aluminum profile construction weighing over 35 kg and capable of a maximum speed of 0.8 m/s. The Mecanum wheeled drive is used with the robot's mobile base, which provides good properties and is popular in modern robotic designs. It can move sideways or rotate around its axis. The dimensions of the robot are 80 x 70 x 150 cm (length, width, height), it has a total weight of 60 kg and can carry an additional 80 kg. The appearance of IRAPs SHaRE-aGIVeR robot is shown in Fig.1.

2.1. *Perception System*

The first main core of our system is the perception system. The perception system provides a common state such as position, orientation velocity, and acceleration. Besides, it also provides the overall environment in the form of a 3D map. The system overview of our perception system is shown in Fig. 2. Each sub-system described as follows:

2.1.1. *Holonomic Wheel Odometry*

The first essential key for the perception-system is Holonomic wheel odometry. Therefore, we use the by-products of our driving system by employing the encoder wheel information was attached to the driving system with an algorithm in [8] to estimate the robot state such as position, orientation, and velocity. However, because of our drive system, which we use to be our odometry has four wheels. The odometry matrix that shows in [8] noninvertible due to the equation system is an overdetermined system because we have four odometry wheels, but we have only three state variables to solve due to our robot laying on the plane. In order to solve this problem, we permutate the configuration of the odometry wheel such as (1,2,3), (2,3,4), (3,4,1), (4,1,2). Therefore, the odometry matrix turns to be invertible and has a configuration as expressed in Eq. (1).

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}_{a,b,c} = \begin{bmatrix} \cos(\theta_a) & \sin(\theta_a) & L \\ \cos(\theta_b) & \sin(\theta_b) & L \\ \cos(\theta_c) & \sin(\theta_c) & L \end{bmatrix}^{-1} \begin{bmatrix} \dot{\phi}_a \\ \dot{\phi}_b \\ \dot{\phi}_c \end{bmatrix} \qquad (1)$$

When the left side vector is a state of the robot. θ is the angle of velocity vector for each wheel reference to robot local coordinate. $\dot{\phi}$ is the angular velocity of odometry wheels, and (a, b, c) is odometry wheel configuration. However, the robot state vector received from the wheel odometry still has uncertainty due to it does not have an error correction system. So, we will employ the following algorithm to correct the accumulative error.

2.1.2. *Lidar Odometry and Visual Odometry*

As mentioned before, wheel odometry has uncertainty such as friction force and wheel slippage. In order to deal with the uncertainty of an environment, we employ the lidar odometry techniques because the core algorithm of lidar odometry relies on external features information such as plane and corner. The core concept for our lidar odometry includes the following steps:

- Point Cloud Filtering: The input point cloud is downsampled by voxel grid filter and statistical outlier removal filter [9] in order to reduce bandwidth, computational power and remove the outlier which comes from an environment such a highly reflective surface and the reflection of sunlight
- ICP Registration: In order to estimate the state vector of the robot, we employ iterative-closest-point (ICP) [10] implemented in libpointmatcher [11] by registering the new point cloud to the last keyframe, which called scan to scan method (S2S).

For now, we received the state vector of the robot from lidar odometry. Nonetheless, it still has uncertainty like wheel odometry. The lidar odometry cannot deal with a long corridor or large flat plane environment. Due to the system does not have enough features to track between two frames.

2.1.3. *Visual SLAM*

Our last perception system is the visual SLAM system. The two systems mentioned before still have problems such as accumulative error and cannot deal with a low geometry feature frame. Both problems can be solved by SLAM (Simultaneous Localization and Mapping) or, in other words, it is odometry with a close loop. At present, the state-of-the-art SLAM has both sensors: lidar sensor and camera. For our robot, we choose RTAB-Map [12] that is visual or camera SLAM. Because it relies on visual features, so it does not affect by low geometry feature frames like lidar odometry. Besides that, it is also combined with loop closure detection and non-linear optimization, which is used to deal with the accumulative error. Moreover, RTAB-Map supported combining the other types of odometry: wheel odometry and lidar odometry, to reduce or eliminate accumulative error and uncertainty around an environment. RTAB-Map has two working modes. The first one is the mapping mode. We used it to create a consistent map as the memory of the robot. Another one is localization mode, which fuses all information from wheel odometry, lidar odometry, and visual odometry to estimate the robot's accurate state vector references to the created map.

2.2. *Navigation System*

The next system is the navigation system, which plans the trajectory path and controls the robot movement. The system overview of the navigation system is shown in Fig.2. Each sub-system described as follows:

2.2.1. *3D and 2D Cost mapping*

The first sub-module of the navigation system is 3D and 2D cost mapping used to transform the 2D occupancy grid map provided from RTAB-Map into costmap used for planning route. The structure of our cost mapping module references from [13]. It was also implemented inside the ROS [14] costmap_2d module. However, we also developed a new range sensor layer to deal with noisy ultrasonics using a probabilistic model. The overall system of costmap is divided into two parts: global and

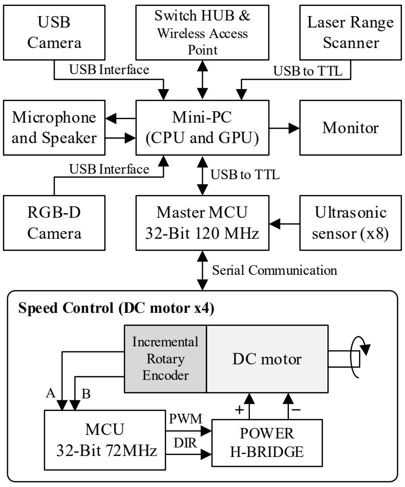

Fig.3 Low-level Control System

local costmaps, which both parts will be used in the next sub-module separately.

2.2.2. *Motion Planner and Motion Controller*

The next sub-module inside the navigation system is the motion planner or trajectory planner module. The motion planner consisted of two parts. The first one is the global planner. It is used for planning a trajectory to reach the goal in the long term. It is called as the global path. Another one is a local planner. It used to deal with immediate situations by trying to reach the global path. We called it the local path. If there is no obstacle the local and global paths will overlap. The first time we build the robot we apply the Dijkstra's/A*global planner together with the dynamic window approach (DWA) [15], [16] local planner which both implemented inside ROS navigation stack. Then, we found that the DWA local planner was not suited for our robot. Because the robot uses a holonomic drive system which increases the complexity for the DWA local planner by "n" times. When "n" is a number of samples used to exploring the velocity space for each axis. For example, the differential drive robot has sample space for all velocity space: O(n2) because it has two of freedom. Nevertheless, for a holonomic robot, the complexity increases to be O(n3) because it has three degrees of freedom. Besides, it also has unpredictable behavior because its algorithm is a sampling base. The user may be confused about its behavior and feel uncomfortable. Therefore, we decided to develop a new local planner which we refer to elastic band planner [17]. Because it has lower complexity, suitable for a holonomic drive system, and also has a predictable behavior. The user can get acquainted in a short time. In summary, we used Dijkstra's/A* for the global planner and elastic band for the local planner. The final step for the Motion Planner and also Navigation system is Motion Controller. We apply a simple PD controller to control both the position and orientation of the robot for each time step.

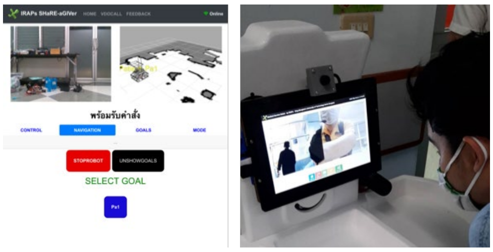

Fig.4 (Left) The medical staff can save position and select goal for the robot. (Right) Telemedicine using peerCalls localhost.

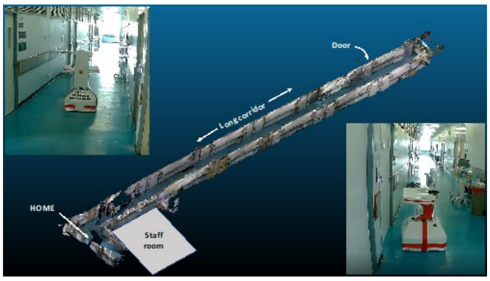

Fig.5 The robot's working area in memory of the mixed map and image reconstruction in 3D

2.3. *Low-level Control System*

The robot contains a powerful Mini-PC (CPU and GPU units) that can handle harder computational tasks. For the module interface of Mini-PC that can communicate with all the sensors and other modules. A detailed scheme of the low-level control system is shown in Fig. 3. The drive subsystem, four DC motors are equipped with incremental rotary encoders (resolution 12 bits) for rotational speed measurement. PID controller is implemented in the control loop of the speed controller of each DC motor.

3. Verification

The robot was tested in the Covid-19 state quarantine from June to September 2020 at the Queen Sirikit Naval Hospital, Thailand. The medical staff could autonomously send the robot to deliver food and things to the front of infected people's rooms. When the robot had reached the goal, the medical team could make a video call from the staff's room to communicate with the patients without personal contact, as shown in Fig. 4. The robot moved along the long corridor to stop at the selected room on the left or right side in Fig. 5. The robot has logged data for four months to a google sheet, and we can summarize the operational performance. The robot gets the active command 5,448 times in 94 days to go to the rooms. The robot has an average operating time of 12 hours and 58 orders per day. It can reach the goals of 94% and 6% missing because there is something on the goal's position. The robot waits for clearing another object at the destination until a timeout and cancel the order by itself. The robot can continue working from 6 to 8 hours when starting with a full charge, depending on the number of orders.

4. Conclusions

The robot is a success to use in a real situation of state quarantine. The medical staff can look after the patients from a distance using the robot for delivery and communication. The robot can go to the destination automatically with obstacle avoidance and is easy to control through the web application by any device. However, the team is developing more functions to deploy to the robot by the doctor's advice.

Acknowledgements

This work is supported by King Mongkut’s University of Technology North Bangkok, Queen Sirikit Naval Hospital, Navamindradhiraj University, PTTGC and Polyfoam Manufacturing Co., LTD.

References

1. WHO. Report of the WHO-China Joint Mission on Coronavirus Disease 2019 (COVID-19); WHO: Geneva, Switzerland, 2020.
2. Yang, G. Z. et al., "Combating COVID-19—The role of robotics in managing public health and infectious diseases," ed: Science Robotics, 2020.
3. France-Presse, A. Robot Receptionists Introduced at Hospitals in Belgium. The Guardian. Available online: https://www.theguardian.com/technology/2016/jun/14/robot-receptionists-hospitals-belgiu m-pepper humanoid.
4. Hurst, D. Japan Lays Groundwork for Boom in Robot Carers. The Guardian. Available online: https://www.theguardian.com/world/2018/feb/06/japan-robots-will-care-for-80-of-elderly-by-2020
5. Care is a Team Effort. Diligentrobots. Available online: https://diligentrobots.com.
6. Panasonic Autonomous Delivery Robots—HOSPI—Aid Hospital Operations at Changi General Hospital.
7. TUG-Change Healthcare. Aethon. Available online: https://aethon.com/mobile-robots-for-healthcare
8. J. Inthiam and C. Deelertpaiboon, "Self-localization and navigation of holonomic mobile robot using omni-directional wheel odometry," TENCON 2014 - 2014 IEEE Region 10 Conference, Bangkok, 2014, pp. 1-5, doi: 10.1109/TENCON.2014.7022281.
9. R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," 2011 IEEE International Conference on Robotics and Automation, Shanghai, 2011, pp. 1-4, doi: 10.1109/ICRA.2011.5980567.
10. P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239-256, Feb. 1992, doi: 10.1109/34.121791.
11. Pomerleau, F., Colas, F., Siegwart, R. et al. Comparing ICP variants on real-world data sets. Autonomous Robots 34, 133–148 (2013). https://doi.org/10.1007/s10514-013-9327-2
12. M. Labbé and F. Michaud, “RTAB-Map as an Open-Source Lidar and Visual SLAM Library for Large-Scale and Long-Term Online Operation,” in Journal of Field Robotics, vol. 36, no. 2, pp. 416–446, 2019.
13. D. V. Lu, D. Hershberger and W. D. Smart, "Layered costmaps for context-sensitive navigation," 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, 2014, pp. 709-715, doi: 10.1109/IROS.2014.6942636.
14. Quigley, Morgan, et al. "ROS: an open-source Robot Operating System." ICRA workshop on opensource software. Vol. 3. No. 3.2. 2009.
15. O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C), Detroit, MI, USA, 1999, pp. 341-346 vol.1, doi: 10.1109/ROBOT.1999.770002.
16. D. Fox, W. Burgard and S. Thrun, "The dynamic window approach to collision avoidance," in IEEE Robotics & Automation Magazine, vol. 4, no. 1, pp. 23-33, March 1997, doi: 10.1109/100.580977.
17. S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," [1993] Proceedings IEEE International Conference on Robotics and Automation, Atlanta, GA, USA, 1993, pp. 802-807 vol.2, doi: 10.1109/ROBOT.1993.291936.