Automatic precision docking for autonomous mobile robot in
hospital logistics - case-study: battery charging

S Vongbunyong1*, K Thamrongaphichartkul, N Worrasittichai, and A Takutruea

1Innovation and Advanced Manufacturing Research Group (I AM)
Institute of Field Robotics, King Mongkut’s University of Technology Thonburi.
126 Pracha Uthit Rd, Bang Mot, Thung Khru, Bangkok, Thailand 10140

* Corresponding Author: supachai.von@kmutt.ac.th

Abstract
Autonomous Mobile Robots (AMRs) plays a vital role in various logistic applications, especially in the health industry. Precise localization is required for automated navigation of the robot throughout the operations. Docking is one of the key processes that the robot moves and physically attaches to the station. Higher precision in position and orientation is required for this process to ensure the secured connection. This article presents an implementation of precise docking maker technique for localization of the robot. The robot is equipped with affordable Lidar sensor while a geometrical marker is on the docking station. The proposed method should help the robot achieve better than ±20 mm error and 0.05 rad error in position and orientation.

Keywords: Autonomous Mobile Robots, Localization, Automatic Docking, Precision Marker, Hospital Logistics

1. Introduction - AMRs in hospital logistics during COVID-19
Logistics management in hospital is involved with effective and efficient processes to transport various items, e.g. food, medicine, medical devices, clothes, trash, and other items in hospitals [1]. In general, these items are manually transported by health workers. Some safety and security issues are involved, especially for restricted items, e.g., controlled medicine, infected items. Robotic technology can be concerned one of the possible solutions to improve the transportation process. Automated Guidance Vehicles (AGV) and Autonomous Mobile Robot (AMR) are concerned the possible solutions for automating transportation system. They convey items between specific locations automatically. They are widely used in industry for ages and recently applied in other working environments, e.g., hospitals.

During pandemic situation, especially COVID-19, avoiding physical contact between patients and health workers is one of the major practices to prevent infection. Many AGVs and AMRs are applied to help keeping distance between them in many activities, e.g., food, medical supplies, medicine delivery, tele-medicine, remoted symptom diagnosis, disinfection, general communication, etc. [2]. Apart from the level of accuracy that these robots need to meet, the contamination control of these machines remains a major challenge.

In general, “zone” is allocated precisely in COVID-19 patients wards. (a) “Dirty zone” refers to the area having chance of contamination where the patients are situated; while (b) “Clean zone” mostly refers to other working areas of the hospitals. The dirty zone includes the negative pressure patients’
room and the area nearby the entrance of room. The health workers normally need to put on personal protective equipment (PPE) suits before entering the room. The changeover of the equipment must be done outside the room in the dirty zone area before crossing to the clean zone. This is cumbersome for the health workers and reduces the efficiency of their operations.

Improving this activity, AMRs can be assigned to work in the dirty zone to reduce the needs of the health workers to enter those areas. AMRs must stay within the dirty zone at all times to prevent the contamination in the clean zone. Therefore, monitoring, control, and services of AMRs, must be able to achieve on-line.

In this research, an AMR system, named “CARVER” [3], is developed to transport items in hospitals automatically; and in this case, it is used in the dirty zone of COVID ward. Remote monitoring and control is enabled via an Internet-of-Things (IoT) platform [4]. However, physical operations such as battery charging and items transfer are still required to complete the tasks. Since the health workers cannot get into the dirty zone, those operation must be done automatically. In this case, the automatic docking ability needs to be developed as it will be used in many applications. Automatic docking for battery charging will be used as a case-study in this article.

2. Literature review

2.1 Applications of AGV and AMR in hospitals

AGV and AMR are used for transportation of items in various environment, e.g., hospital, office building, residential area, factory, etc. The AMRs can navigate through space by using data from various sensors to sense the internal and external factors that affect the motion; for example [5] combined detection and control architecture for detecting payload containers with horizontally mounted 2D laser scanner and a 3D camera for transport heavy payload. In hospitals, many systems are researched and developed. [6] provided technical specifications of an AGV designed for the healthcare facility. The simulation was used to evaluate the system from technical and economic points of view. The analysis on investment was done and the results showed that, currently, the AGV technology is economically feasible only for large scale facilities. [7] developed a path finder of AGV for transportation of material in hospital environments. On-site tests were done in a local hospital and the results confirmed that the robot can move along with the global path and reach the goal without colliding with static and moving objects.

2.2 Docking for autonomous mobile robot

AMRs and AGVs should be able to work automatically throughout the working process, such as scheduled tasks, power recharging, localization, path planning, etc. The docking process is one of the important processes that allow fully automatic operations.

To date, several studies have investigated the docking system by combining Lidars and various types of sensors, markers, and mechanisms to find position and orientation for the robot and the station for precisely docking. [8] proposed automatic docking system for recharging a surveillance robot. This docking method is based on self-localization of the robot and the infrared detectors of the docking station. The structure of the robots and docking station were designed based on the arc-shaped docking interface. [9] proposed a system combining the use of QR codes and infrared distance sensors. The QR code is used as a landmark therefore the robot can perceive its position to the dock. The experiments were done from different starting points and showed the robustness of the proposed solution. [10] proposed the use of a portable effector docking mechanism for path planning and localization of the robot. The portable effector was placed on the floor for correcting the alignment at the time of docking. In this research, the docking process is developed based on Lidars and a precision marker on the station to avoid further modification of the hospital floor.
3. Methodology

3.1 Concept of automatic docking

Having a physical connection to particular working stations, the AMR needs to be able to move to a specific location precisely. This current study employed a “battery recharging station” as a case study. The AMR needs to automatically dock at the station within ±20mm error to achieve the task. According to the hospital environment, extra installation, such as magnetic tape, is restricted in most cases due to many reasons; e.g., modification of the premise is prohibited; access for service and maintenance is limited, etc. Therefore, the precision marker located on the station is used in this research (Figure 1).

![Figure 1. System diagram of CARVER platform with Docking Section](image1)

The system architecture of AMR “CARVER” with IoT platform is shown in the diagram in Figure 2 where the further detail of the system is described in [4]. In this article, the extension from the system in [4] is the “Docking section”, consisting of a docking module and a charging station.

The AMR works in two navigation modes: (a) normal operation and (b) docking operation. The navigation scheme will be switched between these two navigation modes. To perform docking, the AMR will automatically navigate to the docking zone with normal operation scheme; and then switch to the docking operation scheme. The normal operation allows a larger position error which is in the range of ±80mm, while docking operation limits the error within ±20mm. An RPLidar sensor [11], a low-cost Lidar sensor for indoor localization, is used to detect the marker with a specific shape. The marker is simply an angle with two connected straight lines. The Lidar can detect data points and the lines will be recognized. As a result, the position of the docking station can be located relative to the marker position.

![Figure 2. System diagram of CARVER platform with docking Section](image2)
3.2 Algorithm for localization of the marker

The algorithm of the docking process is operated as a state machine running in the Robot Operation System (ROS). Several ROS packages [12] are used in the docking process and the system diagram is set out in Figure 3a. (a) “RPLidar” [11] connects the sensor and convert to sensor_msgs/LaserScan message format. (b) “Laser Line Extraction” [13] extracts the line from sensor_msgs/LaserScan message. The output of this package is to provide two endpoints of the line. (c) “Localization Marker” uses the output from Laser Line Extraction to find the lines which are used to identify the location of the marker according to Equations (1-3). (d) “Base Controller” controls the linear velocity and angular velocity of the robot. (e) “State Machine” is the algorithm to dock at the charger station. The docking procedure is run based on the pseudocode in Figure 3b. Further explanation can be seen in Section 3.3.

The output of the Laser Line Extraction package is \( l_n = [p_{sn}, p_{en}] \). Where \( l_n \) is line extraction; \( p_{sn} \) is the start point of line extraction where the position on xy-plane is \((x_{sn}, y_{sn})\); and \( p_{en} \) is the endpoint of line extraction where the position on xy-plane is \((x_{en}, y_{en})\).

The start and endpoints are used to create a linear equation. Two conditions are used for identifying the marker (see Figure 4(c)). The first condition is the delta angle (\( \Delta \theta_{23} \)) between two lines which is between \( \pm5 \) degree. The second condition is the distance (\( \Delta d_{23} \)) between end point (\( p_{en} \)) and start point (\( p_{sn+1} \)) which is less than 0.05 mm. If both conditions are satisfied, the intersection between two lines, \((x_{intersect}, y_{intersect})\) can be identified as the position of the maker by using equations (1-3).

\[
\begin{align*}
y_n &= m_n x_n + c_n \\
x_{intersect} &= \frac{(c_{n+1} - c_n)}{(m_{n+1} - m_n)} \\
y_{intersect} &= m_n x_{intersect} + c_n
\end{align*}
\]
As can be seen in Figure 4d, the angle of the marker frame with base link frame (the reference coordinate frame at the base of the robot) can be obtained with the geometric approach as the three following steps:

1. Create a circle equation at the point $p_{\text{intersect}}$ with radian $r$ mm;
2. Find intersection point $p_{\text{cl}}$ and $p_{\text{cr}}$ between the line equation and circle equation; and,
3. Find the angle of marker $\theta_{\text{marker}}$ by $\text{Atan2}(p_{\text{cl}}, p_{\text{cr}})$.

![Figure 4. Identification of the markers according to the data from lidar](image)

Figure 4. Identification of the markers according to the data from lidar

![Figure 5. Docking states](image)

Figure 5. Docking states
(a) Go to pre-contact point (b) Align with pre-contact (c) Go to contact point (d) Final state

3.3 Algorithm for docking
After the marker location $\mathbf{p}_{\text{intersect}}$ is obtained, this position is hereby called the position of charger station ($\mathbf{p}_{\text{charger\_station}}$). Transformation matrices are applied to find other reference points for docking (see Figure 5), namely, contact charger station point ($\mathbf{p}_{\text{contact\_charger\_station}}$) and pre-contact charger station point ($\mathbf{p}_{\text{pre\_contact\_charger\_station}}$).

The docking process uses $\mathbf{p}_{\text{base\_link}}$ and $\mathbf{p}_{\text{pre\_contact\_charger\_station}}$ as references for the positions of robot and charger, respectively. From the basic docking algorithm provided in Figure 3b, the robot operates in five states:

1. “Initial state”: the robot locates the marker and knows to obtain the distance between them;
2. “Go to pre-contact point state”: the robot uses the base link to move to the pre-contact charger station;
3. “Align with pre-contact state”: the robot adjusts itself to align with the pre-contact charger station;
4. “Go to contact point state”: the robot moves to the contact charger station point; and
5. “Final state”: the robot is at the charger station position and recharges the battery.

4. Implementation and experiment

4.1 Experiment method

The purpose of the experiment is to test the performance of the proposed docking procedure. The test-bed is set up in the lab environment to perform automatic docking at the battery recharging station (see Figure 6a). The experiments were done in 2 cases with 10 times each. Case-1 represents a basic docking operation since the path is straight in front of the station. Case-2 represents actual movement when the starting point can be anywhere in the space (see Figure 6b).

![Figure 6. Experiment setup (a) test-bed in lab environment (b) Experiment cases](image)

4.2 Experiment results

The experiment result of the position and orientation of the robot at the final state are summarized in Table 1. There was an insignificant difference between both cases. The average position errors were within 13.0 mm on the x-axis and 10.9 mm on the y-axis, while the orientation is represented by heading where the error was within 0.01 rad. Based on the battery charging application, this proposed docking process was able to accomplish the task with compliant design at the charging contact points which allow this level of position and orientation error.

| Case | Position Errors (mm) | Orientation Error (rad) |
|------|----------------------|------------------------|
| Case-1 | 13.0 | 0.01 |
| Case-2 | 10.9 | 0.01 |

Table 1. Results of the summary – errors in the position and the orientation at the final state
5. Discussion and conclusion

AMRs have been currently used in many applications. The second aim of this study was to investigate the effects of AMR operation in hospital applications. To enable fully autonomous ability, automatic docking is one of the important functions as it allows the AMRs to physically interact with other items and equipment. Battery recharging is one of the main features to ensure that the AMR can be continuously operated without the aid of human operators. In this article, automatic docking is developed based on the precision marker and RPLidar. A geometrical approach is used to locate and identify the precision marker. The robot can plan the path and navigate to the station accurately. From the experiment results, the position and orientation errors at the final state are in the allowable range and the AMR was able to achieve the recharging task.

However, from observation, there was some deviation of heading direction along the way between the starting point and contact position. More advanced controller and trajectory control will be investigated and developed to increase docking precision in the future. In a few cases that the deviation was over the allowable range, the size of the charger contact is considered to enlarge to prevent the operation failure.

Acknowledgement

This project is partially funded and supported by (a) Project “Mod Boriraksa” FACO: FIBO Against COVID-19, (b) NXPO Office of Higher Education Science Research and Innovation Policy Council as part of the project “Promotion of Robotics for All & Development of Robotics Innovators, Researchers, Engineers, and Startups”, and (c) “Petchra Pra Jom Klao” Research Scholarship from King Mongkut’s University of Technology Thonburi, Thailand.

References

[1] Gunjan M and Pankaj L 2016 Global Opportunity Analysis and Industry Forecast 2014-2022
[2] Mišekis J Caroni P Duchamp P Gasser A Marko R Mišekienė N Zwilling F de Castelbajac C Eicher L Früh M and Früh H 2020 IEEE Robot. Autom. Lett. 5(4) 5339-5346
[3] Vongbunyong S Worrasittichai N Tripathi S P Takutruae A Thamrongaphichartkul K and Prayongrak T 2020 Int. Jt. Symp. Artif. Intell. Nat. Lang. Process. and Int. Conf. Artif. Intell. Internet Things
[4] Thamrongaphichartkul K Worrasittichai N Prayongrak T and Vongbunyong S 2020 Proc. Int. Jt. Symp. Artif. Intell. Nat. Lang. Process. and Int. Conf. Artif. Intell. Internet Things
[5] Buck S Hanten R Bohlmann K and Zell A 2017 Proc. - IEEE Int. Conf. Robot. Autom. 718-723
[6] Pedan M Gregor M and Plinta D 2017 Procedia Eng. 192 665-670
[7] Bacik J Durovsky F Biros M Kyslan K Perdukova D and Padmanaban S 2017 IEEE Access 5 26892-26900
[8] Song G Wang H Zhang J and Meng T 2011 IEEE Transactions on Consumer Electronics 57(2) 428-435
[9] Quilez R Zeeman A Mitton N and Vandaele J 2015 Proc. EAI Int. Conf. Testbeds Res. Infrastructures Dev. Networks Communities
[10] Takeuchi E and Tsubouchi T 2006 Proc. - IEEE Int. Conf. Robot. Autom. 2006(May) 3380-3386
[11] GitHub, Inc. 2020 RPLIDAR ROS package, github.com/robopeak/rplidar_ros, accessed on 1 December 2020
[12] Robot Operating System, www.ros.org, accessed on 1 December 2020
[13] Laser Line Extraction, github.com/kam3k/laser_line_extraction

| (n = 10) | x (mm) | y (mm) | heading (rad) | x (mm) | y (mm) | heading (rad) |
|---------|--------|--------|---------------|--------|--------|---------------|
| Min     | 0.5    | 1.4    | 0.005         | 2.8    | 6.6    | 0.001         |
| Max     | 26.2   | 19.1   | 0.015         | 17.2   | 16.7   | 0.009         |
| Average | 13.0   | 10.9   | 0.010         | 9.3    | 12.8   | 0.004         |
| S.D.    | 9.0    | 6.3    | 0.003         | 5.0    | 3.0    | 0.002         |