

# Sensor Fusion of Light Detection and Ranging and iBeacon to Enhance Accuracy of Autonomous Mobile Robot in Hard Disk Drive Clean Room Production Line

Sarucha Yanyong, Rattapoohm Parichatprecha, Punyavee Chaisiri,  
Somyot Kaitwanidvilai, and Poom Konghuayrob\*

School of Engineering, King Mongkut's Institute of Technology Ladkrabang,  
1 Chalongkrung, Ladkrabang, Bangkok 10520, Thailand

(Received September 30, 2022; accepted February 21, 2023)

**Keywords:** mobile robot, robot operating system, iBeacon, scan matching, timed elastic band local planner

In this paper, the adaptive Monte Carlo localization (AMCL) error in terms of similar data detected by light detection and ranging (LiDAR) in different locations is investigated. This localization causes a robot to move to the incorrect location temporarily. We propose the fusion of landmark-based localization using an iBeacon device combined with the AMCL algorithm. This technique can solve the probabilistic localization problem of the conventional techniques applied in mobile robots by fusing the timed elastic band (TEB) and scan-matching algorithms, which reduces the error from 7 cm to less than 3 cm. The proposed technique is implemented on a clean-room-type mobile robot with 100 kg payload certificated by the SOP39 standard.

## 1. Introduction

Many industries such as the automotive and electronics industries are using mobile robots to replace complex conveyors in their factories. Mobile robots can be categorized into automated guided vehicles (AGVs) and autonomous intelligent vehicles (AIVs). AGVs can only travel in accordance with guidelines, while AIVs can move anywhere without modifying the infrastructure. AIVs are one of the keys to realizing intelligent factories and have inspired many mobile robot research topics. Dudzik used the OptiTRACK motion capture system to localize a wheeled mobile robot using image data from an IR camera installed on a wall with the Hausdorff distance algorithm.<sup>(1)</sup> However, this technique is not suitable in an industrial environment. Okuyama *et al.* attempted to improve trajectory planning for differential-drive mobile robots based on a resilient propagation algorithm to minimize the trajectory time.<sup>(2)</sup> Rogne *et al.* focused on MEMS sensors and used an inertial measurement method to prevent the dead reckoning state.<sup>(3)</sup> An issue faced by factories is that it is difficult to align mobile robots with target machines, leading to the requirement of an additional system such as an articulated robot, a Cartesian robot, or an additional conveyor system for proper alignment. To overcome this

---

\*Corresponding author: e-mail: [poom.ko@kmitl.ac.th](mailto:poom.ko@kmitl.ac.th)  
<https://doi.org/10.18494/SAM4158>

problem, our paper presents a method of minimizing the error from localization and improving the parking accuracy for mobile robots. The primary objective of a mobile robot is to accurately transport production equipment from one location to another while navigating through a crowded environment filled with other machines. Achieving a low positioning error while carrying out this task is a significant challenge in mobile robot control. Consequently, much research has been conducted on enhancing the accuracy of localization. There have been many studies on accuracy improvement in localization. For example, Jung *et al.*<sup>(4)</sup> proposed the integration of an unscented Kalman filter (UKF) and a fuzzy inference system (FIS) to improve the positioning accuracy of AGVs. Yilmaz and Temeltas<sup>(5)</sup> proposed the self-adaptive Monte Carlo (SA-MCL) method for the indoor localization of AGVs. Their techniques focused on modifying the particle filter algorithm to improve the positioning accuracy of the robot. They obtained accuracies of 6.69 and 4.74 cm in the  $x$ - and  $y$ -directions, respectively.

The first sensor in the mobile robot system applied in our research is a 2D light detection and ranging (LiDAR) sensor. LiDAR is used to measure distances by illuminating a target with laser light and measuring the time for the reflected light to return to the receivers. Systems such as the Google self-driving car also use LiDAR sensors combined with high-accuracy GPS/INS systems to enable cars to drive long distances without user control. Three-dimensional LiDAR has also been applied in ecology to monitor animals.<sup>(6)</sup> Wolcott and Eustice proposed a method of applying 3D LiDAR with a monocular camera to autonomous driving.<sup>(7)</sup> In their research, a LiDAR sensor was used in a mobile robot to scan the environment, in which the information provided by the sensor was the distance to an obstacle. Scan matching is an algorithm that can be used with a LiDAR sensor.<sup>(8,9)</sup> This algorithm transforms the point cloud obtained by LiDAR into odometry information by iteratively solving an optimization problem by comparing the current scan of a LiDAR map with the previous scan to calculate the changes in the orientation and position of the robot. Scan matching significantly improves the accuracy of odometry. This algorithm has been employed in numerous studies.<sup>(10,11)</sup>

The adaptive Monte Carlo localization (AMCL) algorithm is employed in probabilistic localization systems to determine the position of a robot using information from a LiDAR sensor. In some cases, the approximation may not be precise, especially when the LiDAR data from two different rooms are comparable. To address this challenge, the AMCL algorithm incorporates landmark-based localization, which is part of our proposed technique. Additionally, Bluetooth Low Energy (BLE) technology has been widely adopted in the robotics field, particularly for indoor positioning applications. BLE technology in commercial radio transceivers has been developed with received signal strength indicator (RSSI)-based ranging techniques, making it suitable for indoor localization applications. Previous studies<sup>(12–14)</sup> have utilized BLE in smart homes, with one using the ring localization algorithm and BLE to achieve a high accuracy of 0.4 m in determining the position of a receiver device.<sup>(15)</sup> Trilateration has also been used as a technique to estimate the position of a device by utilizing the RSSI from at least three BLE beacons.<sup>(16)</sup> The distances from the beacons to the device are calculated and represented as overlapping circles to determine the device position.

In this study, we propose a mobile robot that modifies the navigation stack from the original robot operating system (ROS) `move_base` package. The navigation process in the `move_base`

package starts from the `map_server` package, which provides data for the occupancy grid map created from simultaneous localization and mapping (SLAM) using the Hector SLAM method. The first step in the navigation stack is to generate a global path planner from the occupancy grid map information. The robot then follows this path to reach its destination. The sets of information from laser scanning, AMCL, and odometry are combined to estimate the local path. In this paper, the original algorithm is also replaced with the timed elastic band (TEB) local planner<sup>(17)</sup> to generate a smooth trajectory and reduce the travel time. Additionally, we apply landmark-based localization to estimate the final position, which also helps to prevent the instability of the localization in similar environments.

## 2. System Architecture

In our research, the AIV is the main part to be improved. The localization and navigation require data from the sensor and send the control signal to the motor driver. This section describes all linkages of the robot components.

As shown in Figs. 1 and 2, the robot navigation obtains information from the LiDAR sensors to measure the distance between the robot and the surrounding environment. The large amount of data obtained from LiDAR is transferred to the main processor via Gigabit Ethernet in the ROS as a laser scan topic (`/scan`). The laser scan is used in many packages in the ROS, such as the localization, navigation, and visible odometry packages. After that, the localization package calculates the position of the robot on a virtual map based on the information from the laser scan and odometry. The odometry is handled by the ROS topic named `/odom`, which is derived from the forward kinematics as shown in Fig. 3. The odometry provides the linear velocity and angular velocity to the ROS. As shown in Fig. 4, these parameters are calculated from the speeds of the robot wheels as follows:

$$v_r = \frac{\omega_1 + \omega_2}{2}, \quad (1)$$

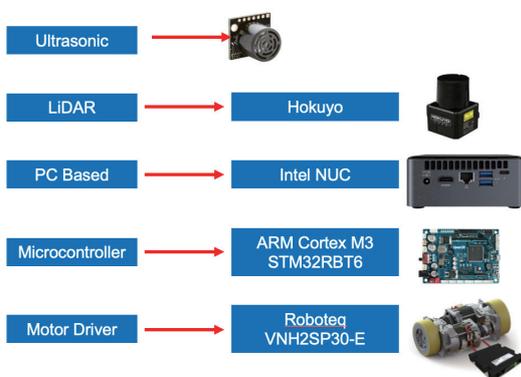


Fig. 1. (Color online) Hardware components.



Fig. 2. (Color online) Differential-drive mobile robot.

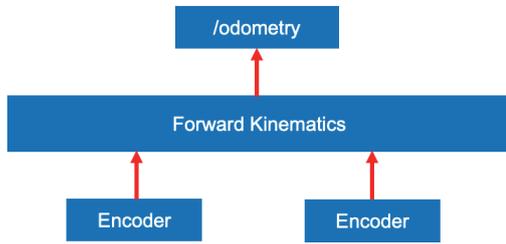


Fig. 3. (Color online) Diagram illustrating the construction of the odometry module.

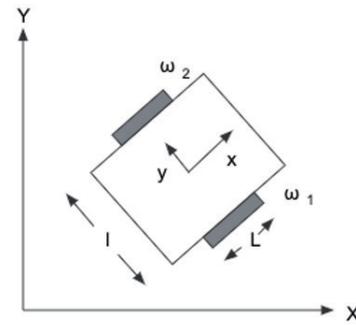


Fig. 4. Differential drive kinematics.

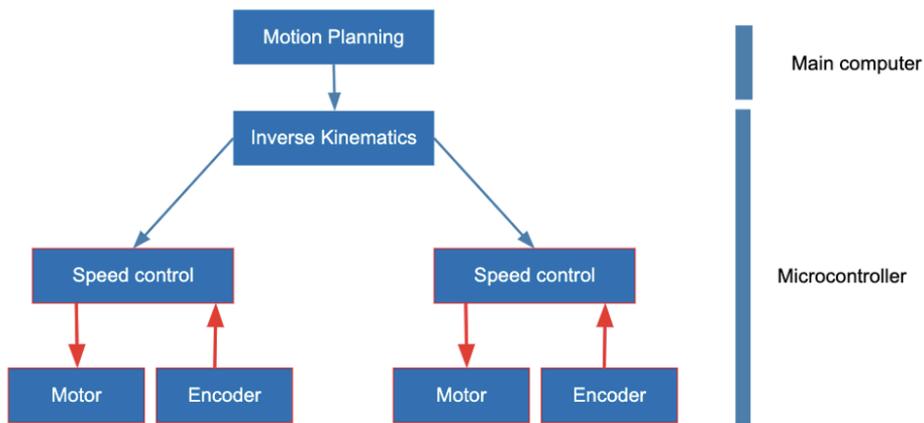


Fig. 5. (Color online) Diagram of the motor control from the motion planning module.

$$\omega_r = \frac{\omega_1 - \omega_2}{l}, \tag{2}$$

where  $v_r$  and  $\omega_r$  are the linear and angular velocities of the robot, respectively,  $\omega_1$  and  $\omega_2$  are the speeds of the two wheels, and  $l$  is the distance between the wheels.

As shown in Fig. 5, to control the module, the inverse kinematics algorithm and a speed control mechanism such as proportional–integral–derivative (PID) control are utilized. The motion planning produces output in the form of linear and angular velocities, which are transferred to the microcontroller for further computation. To control the motor speed, the linear and angular velocities of the robot must be converted to the wheel speeds using the inverse kinematics algorithm,<sup>(18)</sup> as expressed by

$$\begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix} = \begin{bmatrix} L/2 & L/2 \\ L/l & -L/l \end{bmatrix}^T \begin{bmatrix} v_r \\ \omega_r \end{bmatrix}, \tag{3}$$

where  $L$  is the wheel diameter.

The wheel diameter and robot width, equal to the distance between the wheels, are used in the calculation. The resulting wheel speeds serve as reference values for PID control, which are continually compared with the current speed measured by the encoder sensor. This approach enables the high-precision control of the robot.

### 3. Improvement of Localization

Generally, the default method of robot localization provided by the `stack move_base` inside the ROS is AMCL,<sup>(19)</sup> which is a probabilistic localization system for a robot moving in 2D space. Owing to its probabilistic concept, AMCL can recorrect the position of a robot better than only encoder-based odometry. If the wheels of the robot slip as the robot moves on a surface, this will increase the total error on the encoder for localization. However, this problem can be solved by applying the visual odometry algorithm called scan matching.<sup>(20)</sup> Scan matching is applied to the ROS by replacing the odometry information with the information received from the wheels. The iterative closest/nearest point (ICP) algorithm, as shown in Fig. 6 and given in Eq. (4), matches the surface  $S^{ref}$  and a set of points  $\{p_i\}$  using the point-to-line metric concept. The minimization of the distance between points  $p_i$  and surface  $q$  that is projected on surface  $S^{ref}$  is the matching point on the surface used to estimate the position of the moving robot. In industrial mobile robot localization, the point on surface  $q$  is the point cloud received from the LiDAR. Therefore, the matching of the surface can be used to track the traveling distance of the robot applied in the ROS odometry node.

$$\min_q \sum_i \|p_i \oplus q - \prod\{S^q, p_i \oplus q\}^2\| \quad (4)$$

In Fig. 7, the scan-matching node is replaced by the position obtained from odometry before it is input to the AMCL algorithm to estimate the position of the robot.

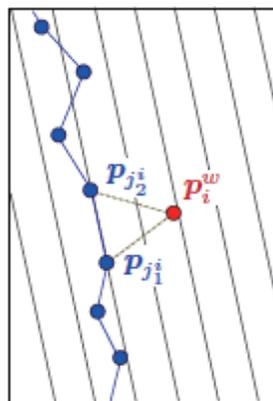


Fig. 6. (Color online) Point-to-line metric approximating distance to surface.

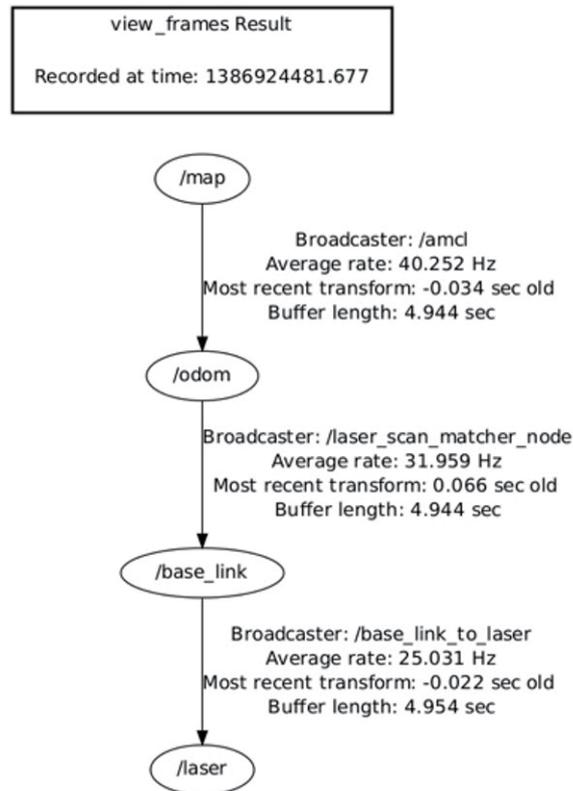


Fig. 7. Replacement of scan-matching node.

#### 4. Improvement of Navigation

The relationships in the ROS navigation stack are schematically shown in Fig. 8. The navigation stack is a built-in Dynamics Window Approach (DWA) local planner.

The TEB technique replaces the original configuration of the navigation stack. The proposed TEB planner and the DWA are both able to find a collision-free path to the goal despite the obstruction caused by dynamic obstacles. The DWA often reduces the robot speed to avoid imminent collisions. In contrast, the TEB technique can consider alternative future evolutions of a scenario, enabling a robot to navigate toward the goal at a speed close to its maximum speed.

As shown in Fig. 9, the global objective function  $V^*(b)$ , which is formulated in terms of the cost function, is to be minimized. It represents a summary of the total time intervals  $\Delta T$  between two consecutive configurations in Eq. (5). This ensures that the TEB technique optimizes the traveling time of the robot.

$$V^*(b) = \sum_{k=1}^{n-1} \Delta T_k^2 \quad (5)$$

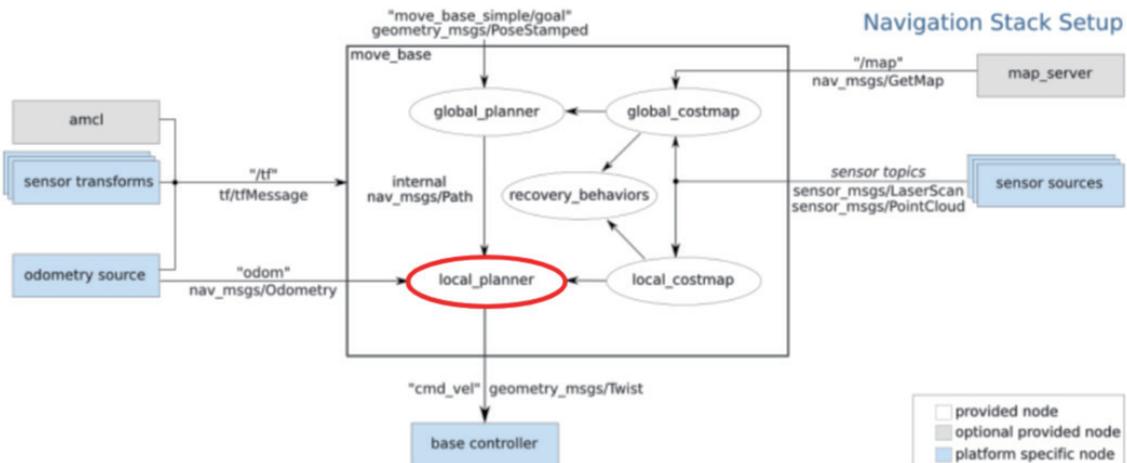
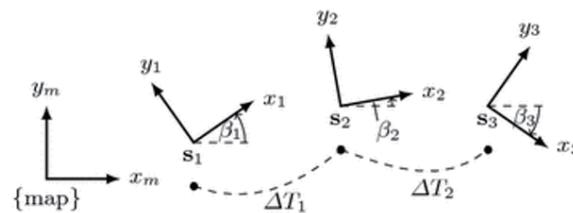
Fig. 8. (Color online) ROS navigation stack.<sup>(21)</sup>

Fig. 9. TEB trajectory representation.

## 5. Landmark-based Localization

AMCL is a probabilistic localization system based on LiDAR and odometry information. Owing to the probabilistic nature of AMCL, it incorrectly estimates the robot location when the shapes of the room and factory production line are similar. To overcome this problem, we propose landmark-based localization based on iBeacon, a BLE device, which is incorporated in AMCL and used to block incorrect positions in the new iterations of AMCL location estimation using the condition obtained from landmark-based localization.

As shown in Table 1, the MCL algorithm is typically implemented in three steps. First, a set of particles is generated. The belief  $X_t$  represents the possible position and orientation (pose) of the robot in the known map. Second, the particles are updated on the basis of the robot control inputs and sensor measurements. This is done by applying the control inputs to each particle to predict how it will move in the environment, then comparing the predicted pose of each particle with those obtained from the actual sensor. In the third step, the updated set of particles is used to estimate the robot pose. The particle distribution is used to calculate the new position of the robot. Subsequently, as the proposed method, the robot pose is fused with the pose estimated from iBeacon. In the fusion method, the robot pose value is not merged with the location estimated from iBeacon, but the truth is reconsidered using iBeacon instead. Therefore, the new robot position might not be published if it is not in the estimated location of iBeacon.

Table 1  
MLC algorithm.<sup>(22)</sup>

---

0: **Function MCL** ( $X_{t-1}, u_t, z_t$ ):

1:  $\bar{X}_t = X_t = \emptyset$

2: for  $m = 1$  to  $M$ :

3:  $x_t^{[m]} = motion\_update(u_t, x_{t-1}^{[m]})$

4:  $w_t^{[m]} = sensor\_update(z_t, x_t^{[m]})$

5:  $\bar{X}_t = \bar{X}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$

6: end for

7: for  $m = 1$  to  $M$ :

8: draw  $x_t^{[m]}$  from  $\bar{X}_t$  with probability  $\propto w_t^{[m]}$

9:  $X_t = X_t + x_t^{[m]}$

10: end for

11: return  $X_t$

---

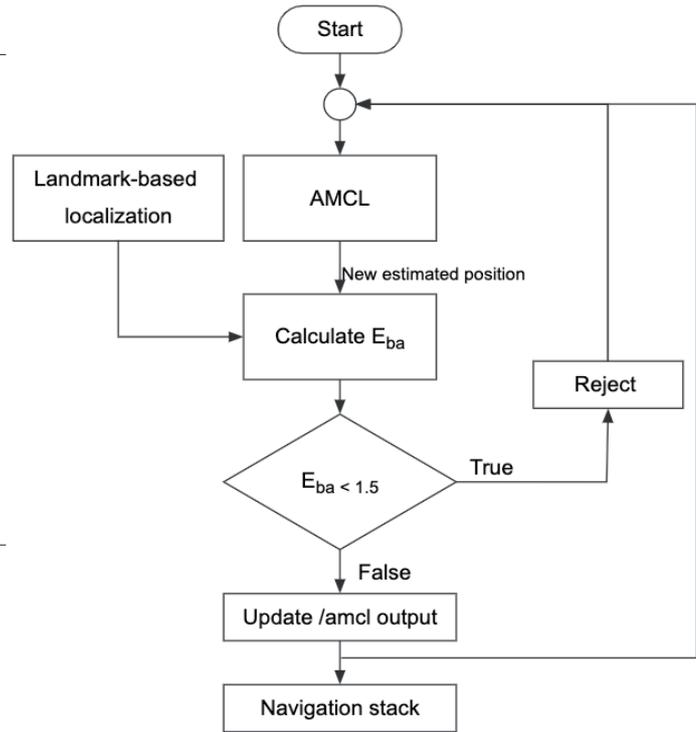


Fig. 10. AMCL estimation blocking algorithm.

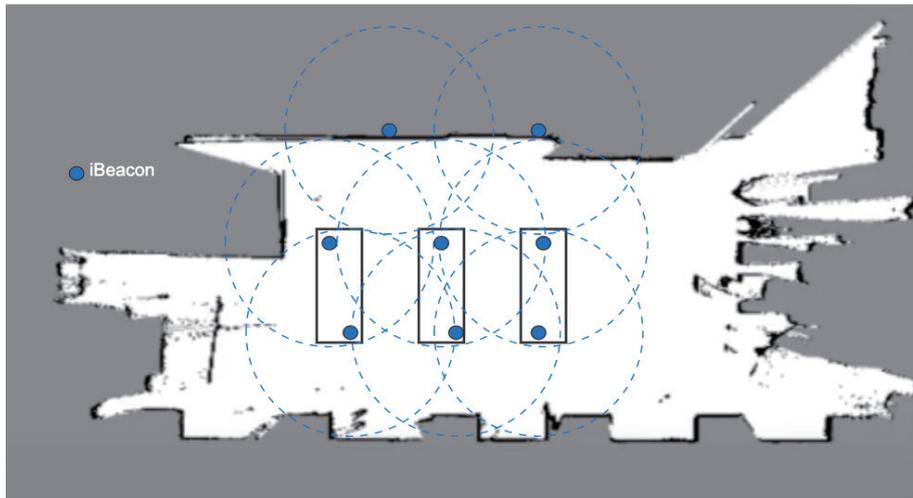


Fig. 11. (Color online) iBeacon installation and coverage area.

The proposed algorithm is expressed by Eq. (6) and shown in Fig. 10. The iBeacon installation and coverage area are shown in Fig. 11.

$$E_{ba} = \sqrt{(X_b - X_a)^2 + (Y_b - Y_a)^2} \tag{6}$$

Here,  $X_b$  and  $Y_b$  are the coordinates  $(x, y)$  of the robot obtained from landmark-based localization.  $X_a$  and  $Y_a$  are the coordinates  $(x, y)$  of the robot obtained from AMCL.

The position obtained from landmark-based localization can be calculated from the trilateration method. This method determines the location by measuring the received signal strength (RSS) between the robot and iBeacon. The distance between the robot and iBeacon is expressed using the RSS as Eq. (7). The position of the robot can be calculated by trilateration using Eqs. (8)–(10). iBeacon must be installed in the overlap area in any location on the map. At least three iBeacon signal devices are required for sufficient signal strength for reliable detection.

$$d_k^i = 10^{\frac{R^0 - R_k^i}{10\gamma}} \quad (7)$$

$R^0$  : RSS of iBeacon

$R_k^i$  : reference RSS value at 1 m distance

$\gamma$  : path loss exponent

$d_k^i$  : distance between the device and iBeacon

$$(X_b - X_{b1})^2 + (Y_b - Y_{b1})^2 = d_{b1}^2 \quad (8)$$

$$(X_b - X_{b2})^2 + (Y_b - Y_{b2})^2 = d_{b2}^2 \quad (9)$$

$$(X_b - X_{b\Box})^2 + (Y_b - Y_b)^2 = d_b^2 \quad (10)$$

## 6. Testing

We propose a testing technique for use in collaborative hard disk production lines, in which the accuracy measurement method must be considered. Visual inspection using a monocular camera is performed in various localization applications such as underwater vehicle positioning.<sup>(23)</sup> The test results in this paper are obtained by visual inspection techniques to measure the robot position at any target point. This technique can also eliminate human error from measurement. The accuracy of the visual inspection testing system is verified by taking 40 shots of a checkerboard adhered on the top of the stationary robot, as shown in Fig. 12. The captured images are then processed using template matching to determine the robot position. The testing system must be verified to meet the hard disk drive factory standards by proofing the measurement unit with the Gage Repeatability and Reproducibility (Gage R&R) methodology. The Minitab program processes the measured position<sup>(24)</sup> to obtain a result that is accurate within the acceptable value of 10%.

After the testing system is verified, the repeatability and accuracy are tested by commanding the robot to move from the initial position to the destination position 70 times. Figures 13(a)–13(c) show the result of testing robot parking. The robot has the ability to park along the  $X$ -axis

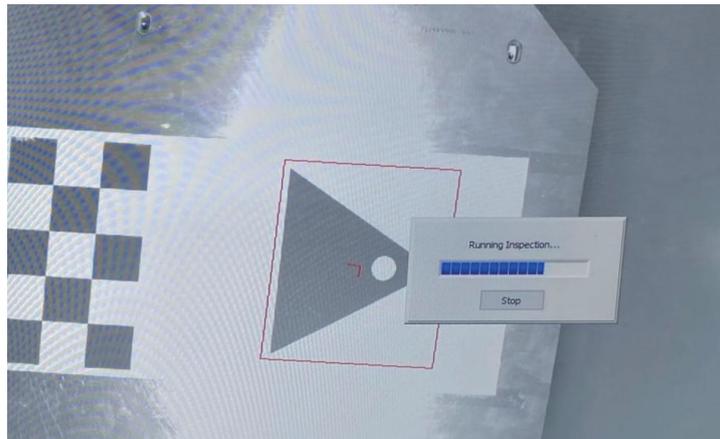


Fig. 12. (Color online) Accuracy and repeatability test using image processing.

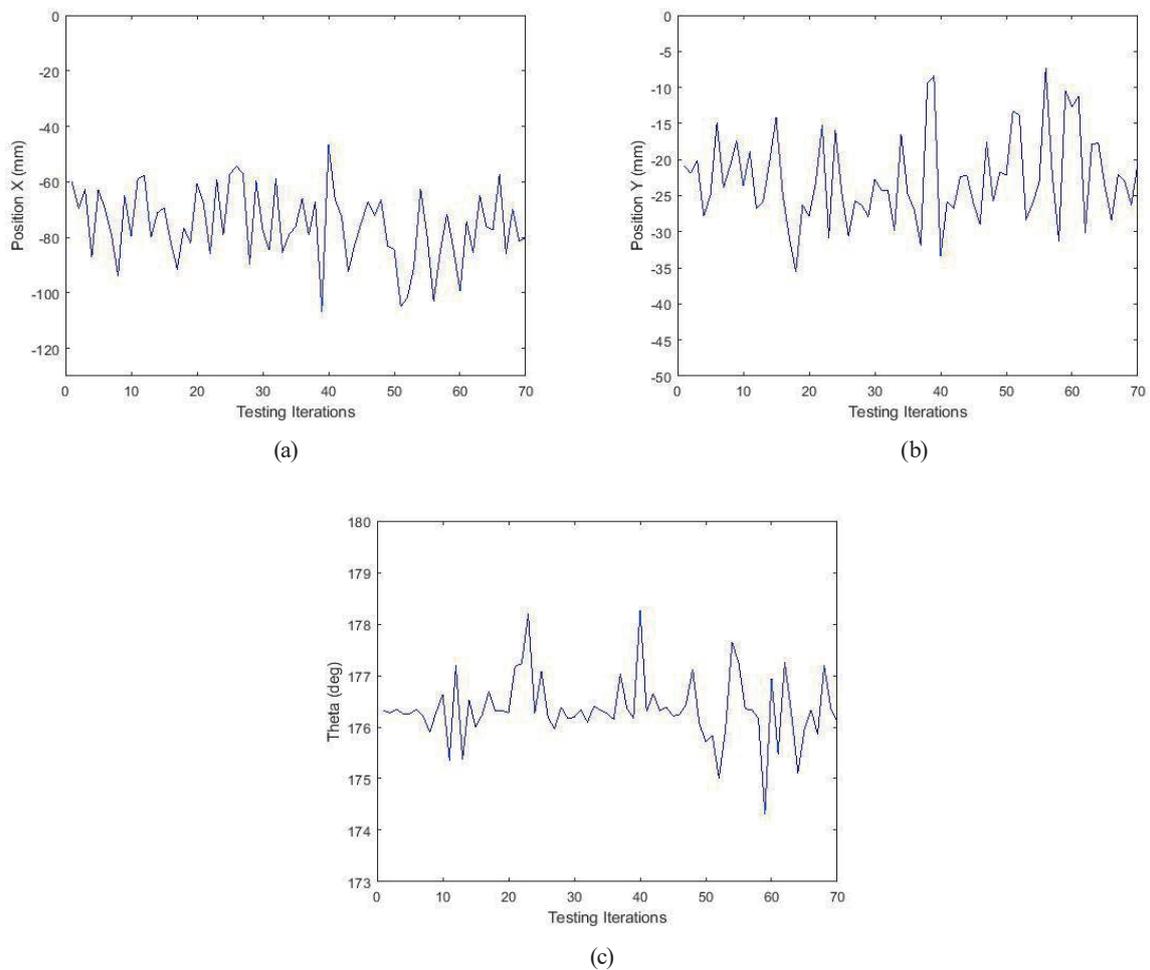


Fig. 13. Parking positions in the (a)  $X$ - and (b)  $Y$ -directions, and (c) orientation, where the expected values are  $-75.5$  mm,  $-24.0$  mm, and  $176.5^\circ$ , respectively.

aligned with the target machine with an accuracy of  $\pm 29$  mm. The gap from the machine, which was measured in the  $Y$ -direction, is  $\pm 14$  mm and the parking orientation accuracy is  $\pm 1.9^\circ$ . This parking accuracy indicates that the proposed method combining the TEB technique and scan matching in the AMCL algorithm can achieve an acceptable position error of less than 3 cm in the  $X$ - and  $Y$ -directions with low orientational error.

Although the parking accuracy is improved, the localization problem of the probabilistic approach still exists. The robot is tested by navigating from the center of the testing area to the target room. In this test, there are two rooms (Room A and Room B) having the same data detected by LiDAR as shown in Fig. 14(a). From the navigation testing, we found that the localization algorithm of the robot resulted in the mislocation of around 10%. Such mislocation occurred because of the similar point cloud data received from the LiDAR of Room A and Room B. Since conventional AMCL uses statistical data, the covariance obtained from matching the robot map and real-time data of LiDAR is less than zero in both rooms. Therefore, these results indicate that conventional AMCL cannot classify the locations of these two different rooms.

Figure 15 shows a comparison of the performance of the proposed technique for blocking incorrect position estimates with that of the conventional AMCL method. The comparison shows the robot positions in the  $X$ - and  $Y$ -directions, and the orientation. At 2.3 s, the conventional AMCL method causes the robot position to swap to another room with similar point cloud data. However, the proposed technique prevents this position swap by blocking incorrect estimates and keeping the robot stable in its destination area, resulting in improved performance, as shown by the red line. The proposed technique works by blocking position estimates that are more than 2 m from the position estimated by the landmark-based localization (iBeacon) method, and instead it uses the previous estimate as the current robot position.

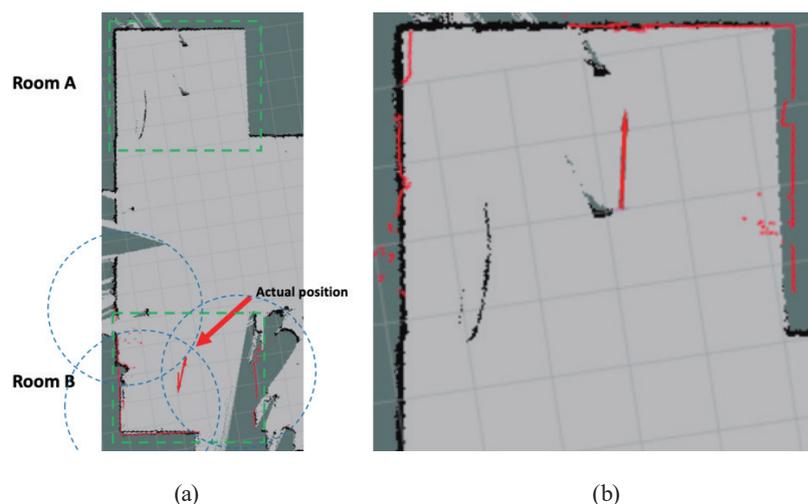


Fig. 14. (Color online) Robot in the (a) testing area and (b) a similar room.

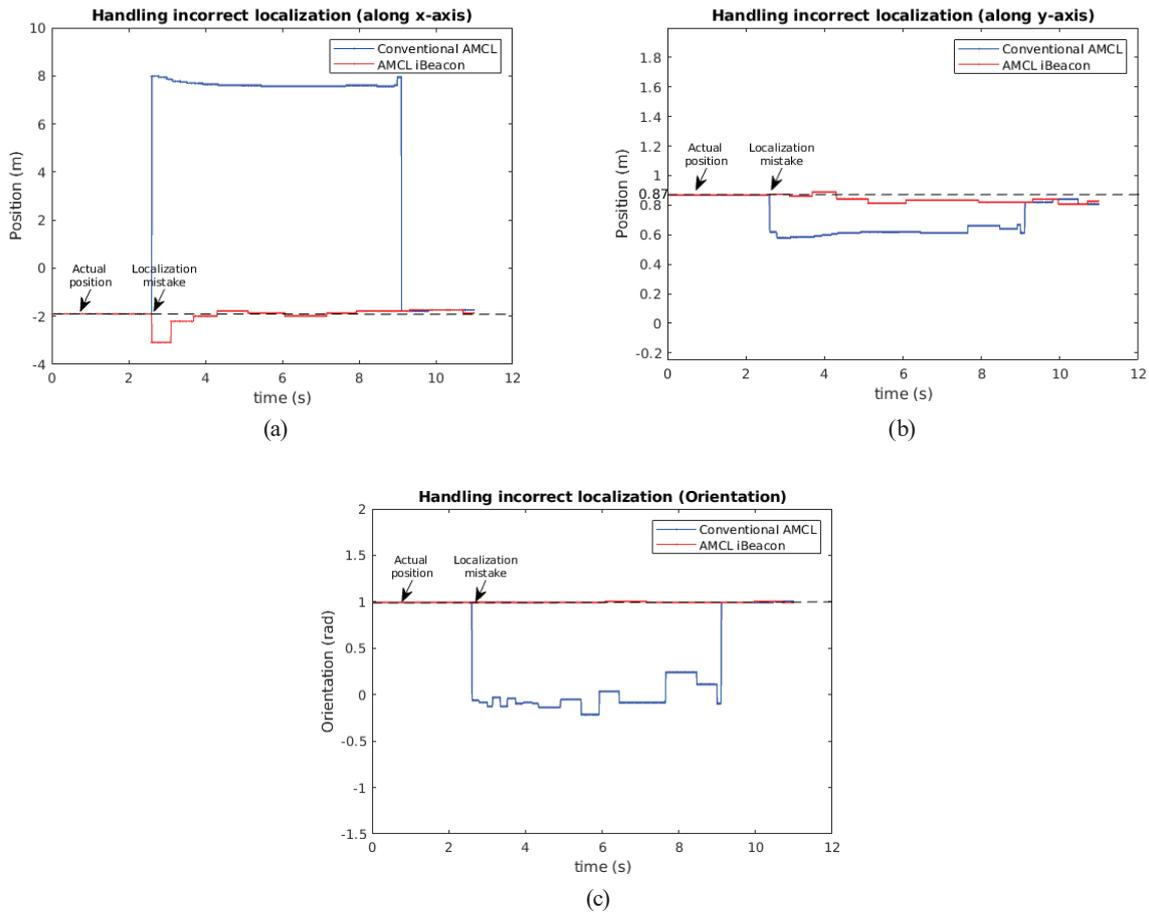


Fig. 15. (Color online) Comparison of results of conventional localization and proposed techniques in terms of (a)  $X$ - and (b)  $Y$ -directions, and (c) robot orientation.

## 7. Conclusion

We have presented a solution to improve the location accuracy and solve the mislocation problem in mobile robots. The proposed technique integrates LiDAR and landmark-based localization using iBeacons to help overcome mislocation problems and effectively prevent positional problems. The integration of the TEB and scan-matching techniques increases the robot accuracy by utilizing LiDAR data. A testing technique is developed to meet factory standards and ensure repeatability. By utilizing computer vision and Gage R&R methodology, the technique reduces human error. The robot completed a repetitive test 70 times, and the results showed an average error of 29 mm in the alignment from the machine ( $X$ -direction), meeting the requirement of  $\pm 5$  cm, and a gap of 14 mm ( $Y$ -axis) between the machine and the robot with a maximum error of 20 cm, as well as a low average orientation error of  $1.9^\circ$ . These results are expected to lead to more reliable robot localization and have the potential to significantly improve the performance of mobile robots in real-world applications.

## Acknowledgments

This work was supported by the Thai Research Fund and Seagate Technology (Thailand) Co., Ltd.

## References

- 1 S. Dudzik: *Energies* **13** (2020) 6437. <https://doi.org/10.3390/en13236437>
- 2 I. F. Okuyama, M. Maximo, and R. Afonso: *J. Control Autom. Electr. Syst.* **32** (2021) 120. <https://doi.org/10.1007/s40313-020-00657-x>
- 3 R. H. Rogne, T. H. Bryne, T. I. Fossen, and T. A. Johansen: *IFAC-PapersOnLine* **49** (2016) 139. <https://doi.org/10.1016/j.ifacol.2016.10.334>
- 4 K. Jung, J. Kim, J. Kim, E. Jung, and S. Kim: *Rob. Auton. Syst.* **62** (2014) 1241. <https://doi.org/10.1016/j.robot.2014.03.016>
- 5 A. Yilmaz and H. Temeltas: *Rob. Auton. Syst.* **122** (2019) 103285. <https://doi.org/10.1016/j.robot.2019.103285>
- 6 A. B. Davies and G. P. Asner: *Trends Ecol. Evol.* **29** (2014) 681. <https://doi.org/10.1016/j.tree.2014.10.005>
- 7 R. W. Wolcott and R. M. Eustice: *Proc. 2014 IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IEEE, 2014)* 176–183. <https://doi.org/10.1109/IROS.2014.6942558>
- 8 Canonical Scan Matcher (CSM): <http://censi.science/software/csm/> (accessed April 2021).
- 9 A. Censi: *Proc. 2008 IEEE Int. Conf. Robotics and Automation (IEEE, 2008)* 19–25. <https://doi.org/10.1109/ROBOT.2008.4543181>
- 10 J. Li, R. Zhong, Q. Hu, and M. Ai: *Sensors* **16** (2016) 1265. <https://doi.org/10.3390/s16081265>
- 11 R. Miyamoto, M. Adachi, H. Ishida, T. Watanabe, K. Matsutani, H. Komatsuzaki, S. Sakata, R. Yokota, and S. Kobayashi: *J. Robot. Mechatron.* **32** (2020) 1137. <https://doi.org/10.20965/jrm.2020.p1137>
- 12 R. Dalce, A. V. Bossche, and T. Val: *Proc. 2014 IEEE Int. Conf. Ultra-WideBand (IEEE, 2014)* 681–691. <https://doi.org/10.1109/ICUWB.2014.6958942>
- 13 R. Dalce, A. V. Bossche, and T. Val: *Proc. 2013 IEEE Int. Conf. Indoor Positioning and Indoor Navigation (IEEE, 2013)* 1–6. <https://doi.org/10.1109/IPIN.2013.6817852>
- 14 P. Spachos and K. N. Plataniotis: *IEEE Syst. J.* **14** (2020) 3483. <https://doi.org/10.1109/JSYST.2020.2969088>
- 15 A. Thaljaoui, T. Val, N. Nasri, and D. Brulin: *Proc. 2015 IEEE Int. Conf. Industrial Technology (IEEE, 2015)* 2178–2183. <https://doi.org/10.1109/ICIT.2015.7125418>
- 16 T. Dinh, N. Duong, and K. Sandrasegaran: *IEEE Sens. J.* **20** (2020) 10283. <https://doi.org/10.1109/JSEN.2020.2989411>
- 17 C. Rösmann, F. Hoffmann, and T. Bertram: *Rob. Auton. Syst.* **88** (2017) 142. <https://doi.org/10.1016/j.robot.2016.11.007>
- 18 F. Ke, Z. Li, and C. Yang: *IEEE Trans. Ind. Electron.* **65** (2017) 3437. <https://doi.org/10.1109/TIE.2017.2756595>
- 19 F. R. Q. Ani, A. N. Jati, and U. Sunarya: *Proc. 2016 IEEE Int. Conf. Information Technology Systems and Innovation (IEEE, 2016)* 1–6. <https://doi.org/10.1109/ICITSI.2016.7858235>
- 20 A. Censi: *Proc. 2006 IEEE Int. Conf. Robotics and Automation (IEEE, 2006)* 2291–2296. <https://doi.org/10.1109/ROBOT.2006.1642044>
- 21 Move base: [http://wiki.ros.org/move\\_base](http://wiki.ros.org/move_base) (accessed April 2021).
- 22 S. Thrun, W. Burgard, and D. Fox: *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)* (The MIT Press, 2005) 1st ed.
- 23 C. Masuzaki, I. Yamamoto, A. Morinaga, K. Sadano, Y. Kai, T. Nakano, and Y. Kato: *Sens. Mater.* **33** (2021) 907. <https://doi.org/10.18494/SAM.2021.3223>
- 24 R. M. Khan: *Problem Solving and Data Analysis Using Minitab a Clear and Easy Guide to Six Sigma Methodology* (Wiley, New York, 2013) 1st ed., Chap. 6.

## About the Authors



**Sarucha Yanyong** received his B.Eng degree in mechatronics engineering and his M.Eng degree in electrical engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand, in 2012 and 2014, respectively. He is currently working towards a Ph.D. degree in electrical engineering at KMITL. His research interests are in the areas of mobile robotics, machine learning, control systems, and artificial intelligence. ([sarucha.y@gmail.com](mailto:sarucha.y@gmail.com))



**Rattapoohm Parichatprecha** received his B.Eng in civil engineering from Mahanakorn University of Technology, Thailand, in 1997 and his M.Eng and Ph.D. degrees in structure engineering from Asian Institute of Technology, Thailand, in 1999 and 2008, respectively. Currently, he works at King Mongkut's Institute of Technology Ladkrabang in the Department of Civil Engineering. His research interests include the monitoring and safety evaluation of railway tracks and vehicles, railway infrastructure, and smart railway maintenance systems. ([rattapoohm.pa@kmitl.ac.th](mailto:rattapoohm.pa@kmitl.ac.th))



**Punyavee Chaisiri** received his B.Eng., M.Eng., and Ph.D. degrees from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand in 2008, 2011, and 2020, respectively. He has been working at KMITL's Department of Electrical Engineering since 2012, and his research interests include high-voltage engineering applications and sensors. ([punyavee.ch@kmitl.ac.th](mailto:punyavee.ch@kmitl.ac.th))



**Somyot Kaitwanidvilai** received his B.Eng and M.Eng degrees in electrical engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Thailand, in 1996 and 2000, respectively. He received his Ph.D. degree in mechatronics engineering from Asian Institute of Technology in 2005. Currently, he works at KMITL in the Department of Electrical Engineering. His research interests include artificial intelligence in power and energy systems, control systems, mechatronics, power electronics, and robust and adaptive control in power systems. ([somyot.ka@kmitl.ac.th](mailto:somyot.ka@kmitl.ac.th))



**Poom Konghuayrob** received his B.Eng, M.Eng, and Ph.D. degrees in electrical engineering from King Mongkut's Institute of Technology Ladkrabang (KMITL), Bangkok, Thailand, in 2012, 2013, and 2018, respectively. Currently, he works at KMITL in the Department of Electrical Engineering. His research interests are in the areas of robust control, DC/AC inverters, solar cell fields, and artificial intelligence. ([poom.ko@kmitl.ac.th](mailto:poom.ko@kmitl.ac.th))