Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 3492175, 12 pages
https://doi.org/10.1155/2022/3492175

Research Article

The Navigation of Mobile Robot in the Indoor Dynamic Unknown Environment Based on Decision Tree Algorithm

Yupei Yan,[1] Weimin Ma ,[1] Yangmin Li,[2] Sengfat Wong,[3] Ping He,[4] Shaoping Zhu,[1] and Xuemei Yin[1]

[1]*Department of Artificial Intelligence, Zhuhai City Polytechnic College, Zhuhai 519090, China*
[2]*Department of Industrial and Systems Engineering, HongKong Polytechnic University, HongKong 999077, Hong Kong*
[3]*Faculty of Science and Technology, University of Macau, Macau 999078, Macau*
[4]*School of Intelligent System Science and Engineering, Jinan University, Zhuhai 519072, China*

Correspondence should be addressed to Weimin Ma; yb47409@connect.um.edu.mo

Received 16 March 2022; Revised 4 April 2022; Accepted 5 April 2022; Published 20 June 2022

Academic Editor: Xin Ning

This study proposes an optimized algorithm for the navigation of the mobile robot in the indoor and dynamic unknown environment based on the decision tree algorithm. Firstly, the error of the yaw value outputted from IMU sensor fusion module is analyzed in the indoor environment; then, the adaptive FAST SLAM is proposed to optimize the yaw value from the odometer; in the next, a decision tree algorithm is applied which predicts the correct moving direction of the mobile robot through the outputted yaw value from the IMU sensor fusion module and adaptive FAST SLAM of the odometer data in the indoor and dynamic environment; the following is the navigation algorithm proposed for the mobile robot in the dynamic and unknown environment; finally, a real mobile robot is designed to verify the proposed algorithm.The final result shows the proposed algorithms are valid and effective.

1. Introduction

There is a wide and extensive application of the mobile robot in many fields; it can fulfill the repeated, heavy, accurate, and complex work in many special environments; the navigation is important for the mobile robot because it is the basic function for the other actions of the mobile robot; it should move freely to the destination without colliding with the obstacles, just like people need to avoid colliding with the other people on the road. There are many navigation algorithms proposed which can be applied in many fields to fulfill the different requirement for human, so the optimization of the navigation is necessary to make the mobile robot moving more intelligently and effectively to the destination [1–3].

Different from the outside static known environment, if the mobile robot is running in the indoor dynamic unknown environment, more conflicts and limitations will be considered; the following is a comparison between them.

The first difference is the location and orientation function; the self-driving car is taken for example; all the cars are running in the outside environment where the GPS sensor can receive the satellite signals properly, so GPS can show the accurate position of the car on the roads, but if the car is running in the indoor environment where the satellite signal is weak or shielded, the GPS function will be unavailable and the corresponding GPS positioning function will be limited. It can only search few or no satellites and it cannot get the accurate position. The IMU sensors have the same problem, so the mobile robot cannot get its location and orientation through the IMU sensors in the indoor environment [4, 5].

The second difference is all environments are known for self-driving car because the map of all roads is provided for the car, but sometimes the mobile robot needs to work in the unknown environment where no map is provided, such as the environment of 24-hour lobby filled with moving people where it is not easy to build the map with SLAM technology [6]; otherwise, the mobile robot may need to fulfilling burrow adventure tasks where it is a totally unknown environment, so the mobile robot cannot locate its position on the map if it is running in an unknown environment.

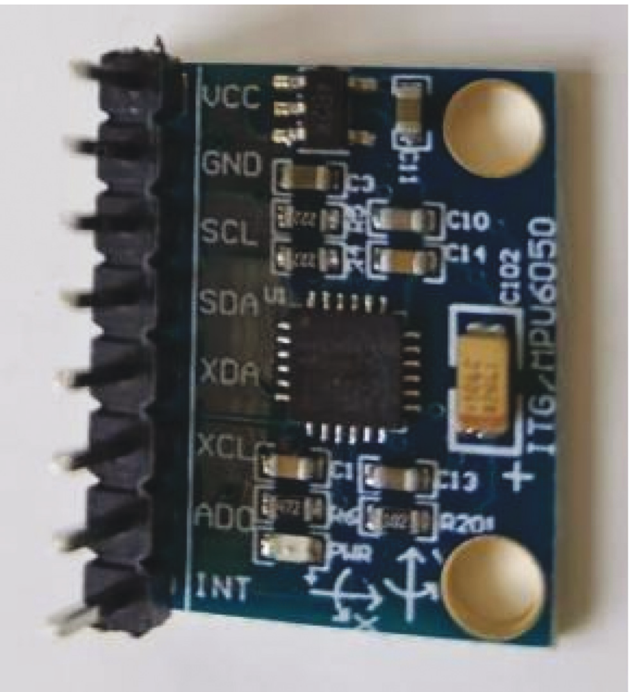


Figure 1: Gyroscope.

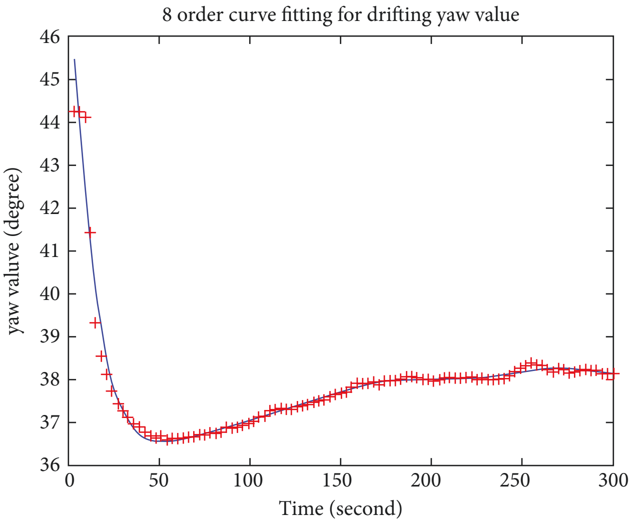


Figure 2: The 8-order curve fitting for the drifting yaw value.

The third difference is that sometimes mobile robot cannot be located in the dynamic environment; the home service mobile robot is taken for example; although it can build the map for the environment inside the house with SLAM, but people may move the position of the home appliance sometimes, so the mobile robot will recognize the wrong position through feature point matching whose data come from the laser sensor or the camera [7].

Based on the above comparisons, as well as considering the limited CPU and battery power for the mobile robot, the navigation algorithm in the indoor dynamic unknown environment should be improved.

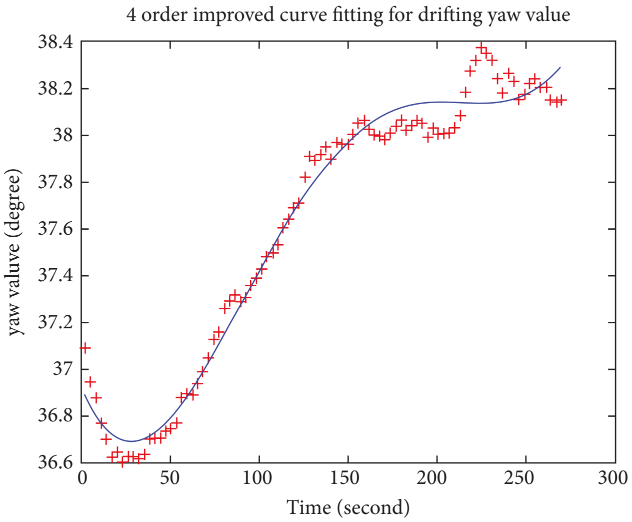


Figure 3: The 4-order improved curve fitting for the drifting yaw value.

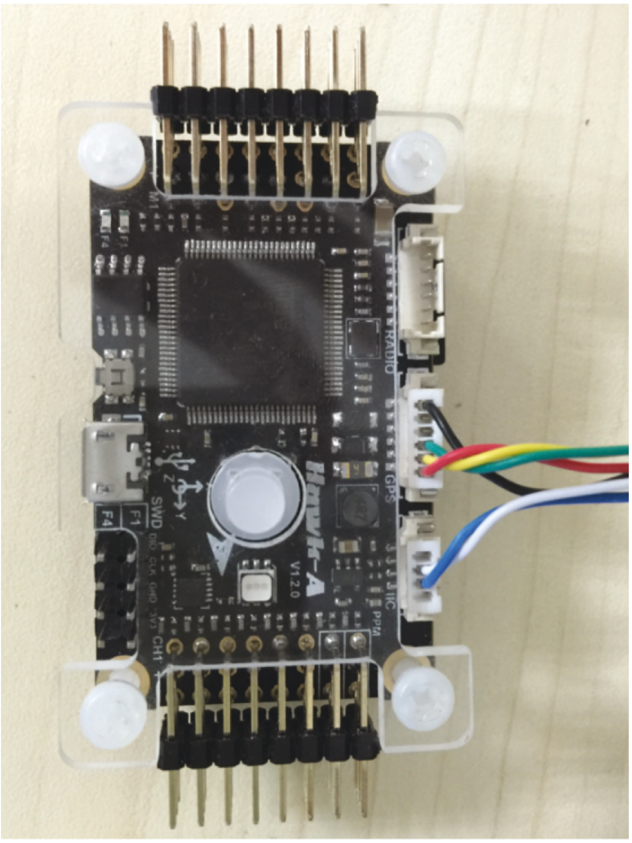


Figure 4: IMU sensor fusion module with GPS, gyroscope, electronic compass, magnetometer, and accelerometer.

2. Decision Tree Algorithm for the Yaw Value from IMU Sensor Fusion Module and Odometer

2.1. Yaw Value from IMU Sensor Fusion Module in the Indoor Environment. In the navigation process of the mobile robot,

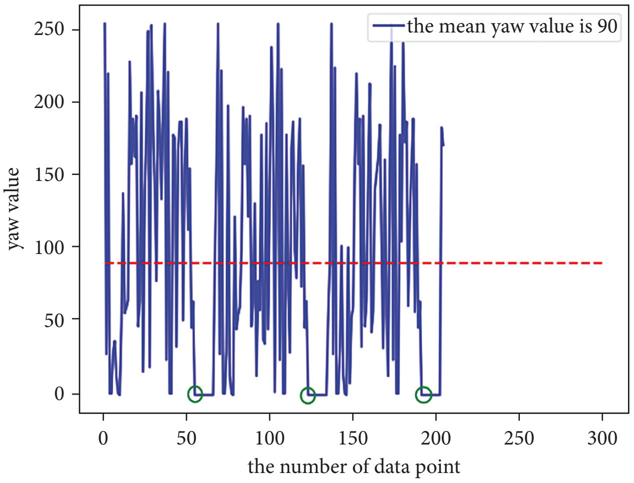


Figure 5: The output yaw value from multi-IMU sensor fusion module.

it is necessary to get the accurate current orientation of the mobile robot in every second, which can be obtained from gyroscope shown in Figure 1.

The output data of the gyroscope will be unavailable in the indoor environment; for example, when the mobile robot stops rotating and keeps static in the indoor environment, after the PID processing for gyroscope, the yaw value is collected and marked with the red cross as shown in Figure 2. It is obvious that the yaw value is still drifting and gradually maintains stability after 300 seconds (5 minutes); the blue line in Figure 2 shows the curve fitting for the yaw value during 300 seconds; the 1–7-order curve fitting is not perfect, but the 8 order can fit the curve more accurately as shown in Figure 2; the fitting function is

$$\begin{aligned} y = {} & 0.00000000000003 \times x^{8} - 0.000000000000086 \times x^{7} \\ & + 0.000000000049473 \times x^{6} \\ & - 0.000000015493482 \times x^{5} + 0.000002873185886 \times x^{4} \\ & + 0.021422402741956 \times x^{2} - 0.761774606869785 \times x \\ & + 47.571395698231385. \end{aligned} \tag{1}$$

Formula (1) is too complex and contains lots of double float data calculations which will consume lots of CPU resources and battery power, from the observation of Figure 2, that the data often drift dramatically in the first 30 seconds, which makes the curve complex and brings in lots of high-order term; if the data in the first 30 seconds can be deleted, it will simplify the curve fitting and polynomial formulas, compared with the other groups test data of yaw value; it is summarized that the yaw value will always drop 10%–20% in the first 30 seconds; the yaw value data at 30th second are often 80–90% of the initial yaw value, so it is often 85% of the initial yaw value for average.

Figure 3 shows the fourth-order improved curve fitting for drifting yaw value, and the yaw data in the first 30 seconds are deleted; the fitting formula is

$$\begin{aligned} y = {} & 0.000000003 \times x^{4} - 0.000002660 \times x^{3} + 0.000647 \times x^{2} \\ & - 0.0514723 \times x + Y_{i} \times 0.85. \end{aligned} \tag{2}$$

It is clear that the 4-order curve fitting is more simple; if x has an assigned value as 270, the stable yaw value can be computed according to formula (2), but it is still not accurate because the yaw value comes from only one gyroscope in the indoor environment; if the cost of the device for the mobile robot is not considered, the multisensor fusion algorithm for GPS, gyroscope, electronic compass, magnetometer, and accelerometer is better than data processing for only one gyroscope as Figure 4 shows.

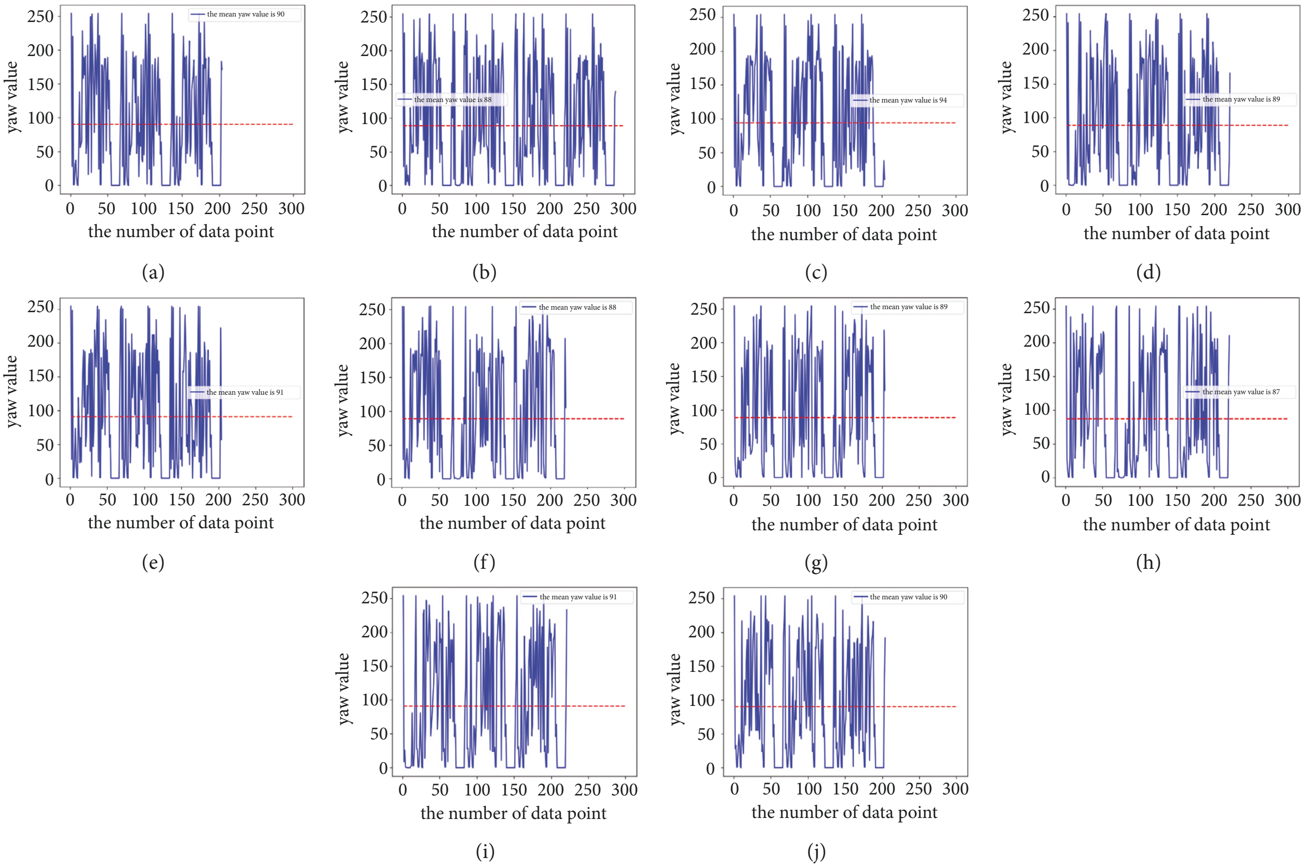
The multi-IMU sensor fusion module can be purchased on the market, which has downloaded the software of multi-IMU sensor fusion algorithm inside, so the output data of this module are more accurate [8]; when the mobile robot is rotating in the indoor environment and stops suddenly, the output yaw value is shown in Figure 5.

Figure 5 shows that the output yaw value is periodical which can be marked with green circle at the bottom of Figure 5; the baud rate is 115200; there are 90 data points in a period, and the output format is Byte, so the period is $10 * 90/115200 = 0.0078125$ second; in one period, the data still drift dramatically; it is needed to compute the mean yaw value in one period, which is 90 degree in Figure 5.

If the test is repeated for 10 times, when every time the mobile robot rotates and stops at a same direction, an object is placed to verify it is the same direction for 10 times' test, the result of 10 times test is shown in Figure 6.

From Figure 6, for the same direction in the indoor environment, the yaw value from one multi-IMU sensor fusion module is different every time and the range of value is between 87 to 94 in 10 times' test, so the yaw value from IMU sensors is not accurate in the indoor environment.

2.2. Odometer Data from the Processing of Adaptive FAST SLAM Algorithm. There is another method to obtain the yaw value, and the odometer can compute the yaw value [9]; generally, the reduction ratio of the gear motor is 1 : 50; the output pulses from the optical-electricity encoder is 500 when a DC motor rotate a circle; when gear motor connects to DC motor, the output pulse is 25000 when the wheel



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j)

Figure 6: The 10 times' test of the output yaw value from multi-IMU sensor fusion module.

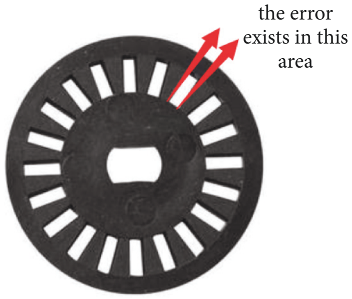


Figure 7: The error exists in each gap of grid of the optical-electricity encoder.

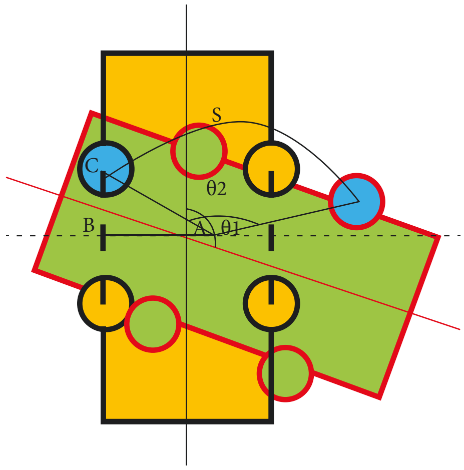


Figure 8: The yaw value of the mobile robot from the odometer.

rotates in a circle; although the yaw value can reach a high accuracy when there is 25 KHz pulse outputted, there is still some error exists as shown in Figure 7.

The error is about 360/25000 = 0.0144 degree for a grid of optical-electricity encoder; when the mobile robot moves for a long time, the error will be accumulated to a large value; at the same time, ground friction is another source of error, especially when the mobile robot collides with a heavy obstacle and the wheel is slipping in the dynamic environment; in this status, the optical-electricity encoder is still computing, but the yaw value does not change; usually, the sensor error and observation error all can be accumulated, which is accumulated error. Because the true yaw value can be measured from round circle ruler in the indoor environment, so the accumulated error can be computed.

If the accumulated error is $v(k)$, the diameter of wheel is D, and the pulse from optical-electricity encoder is P, through analyzing the status of the mobile robot in Figures 8 and 9, the systematic observation model can be obtained from

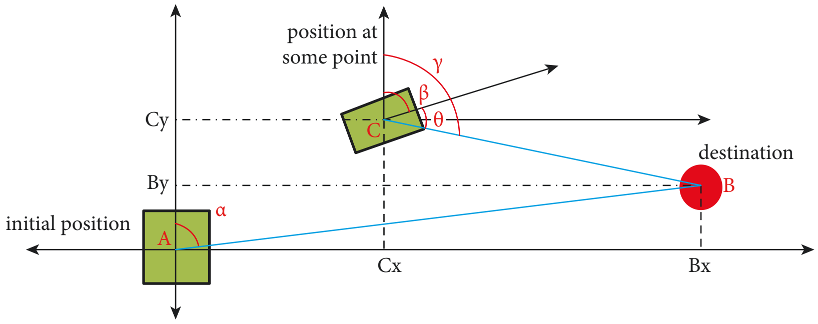


Figure 9: The simulation of the mobile robot moves in the indoor environment.

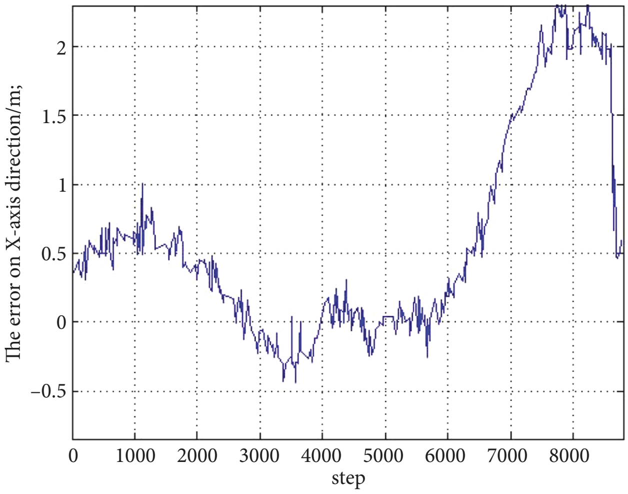


Figure 10: The error of EKF algorithm for the yaw value.

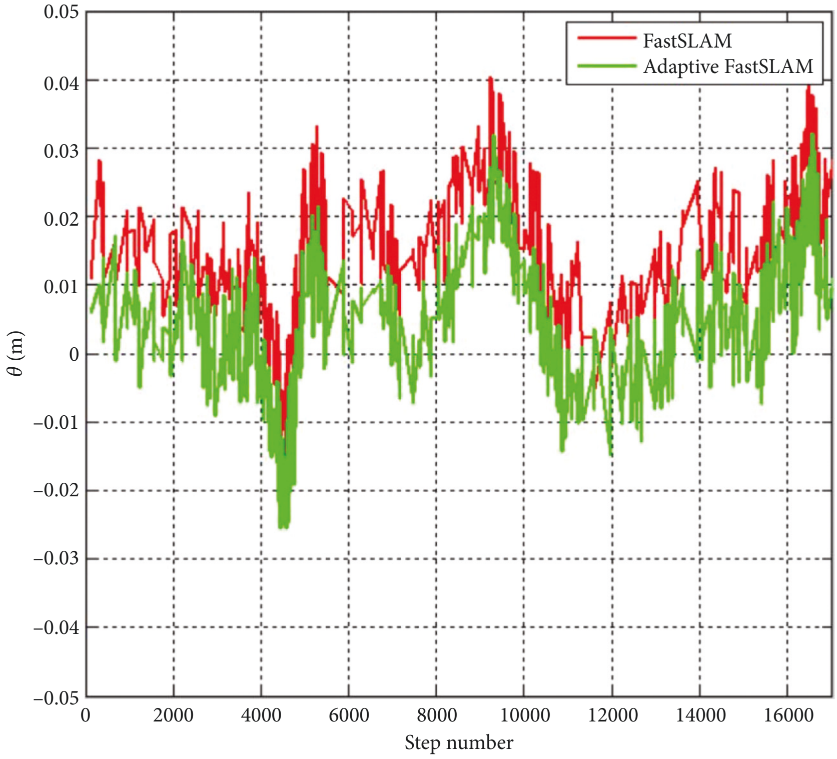


Figure 11: The comparison of yaw value error between Fast SLAM and Adaptive Fast SLAM.

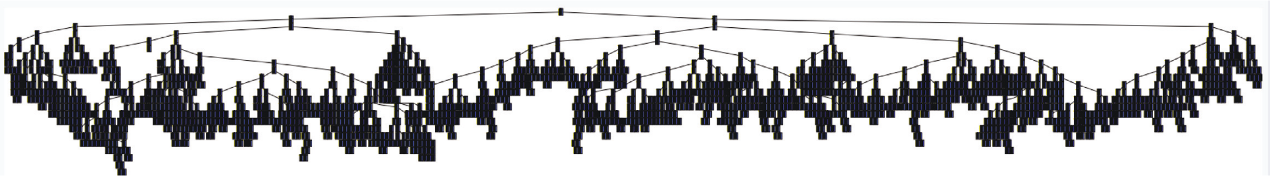


Figure 12: The 360 classifications of the decision tree.

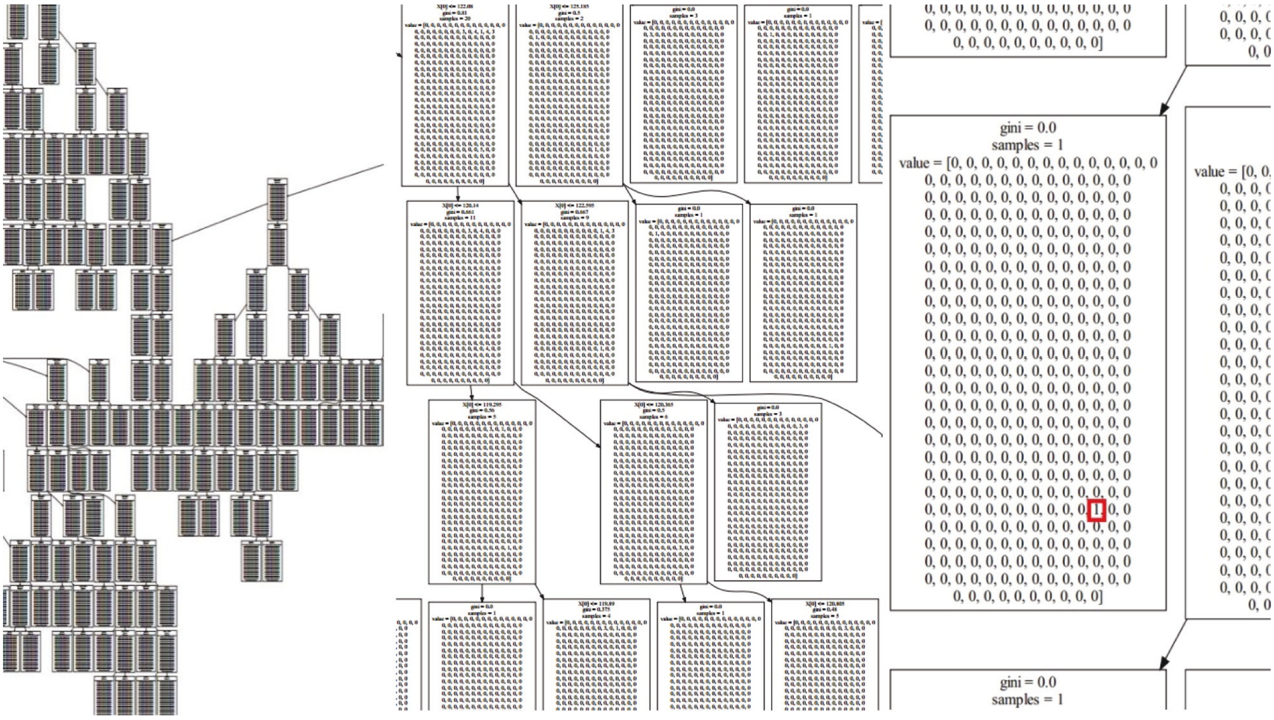


Figure 13: The detailed information of the decision tree.

$$Z(k)=\begin{bmatrix} r(k) \\ \phi(k) \end{bmatrix}+v(k) = \begin{bmatrix} BC \\ \gamma-\beta \end{bmatrix}+v(k) = \begin{bmatrix} \left((Bx-Cx)^2+(By-Cy)^2\right)^{1/2} \\ \tan^{-1}\left(\left(C_y-B_y\right)/\left(C_x-B_x\right)\right)+\pi/2- \\ \sum_{i=0}^{k} D\times P\times t_i/F\times\sqrt{AB^2+BC^2} \end{bmatrix}+v(k). \qquad (3)$$

The state estimation of the mobile robot is analyzed to get the method to eliminate the accumulated error; the purpose is to obtain the maximum probability of the state of the mobile robot:

$$\widehat{x}=\arg\max_{x} p(x|v,y). \qquad (4)$$

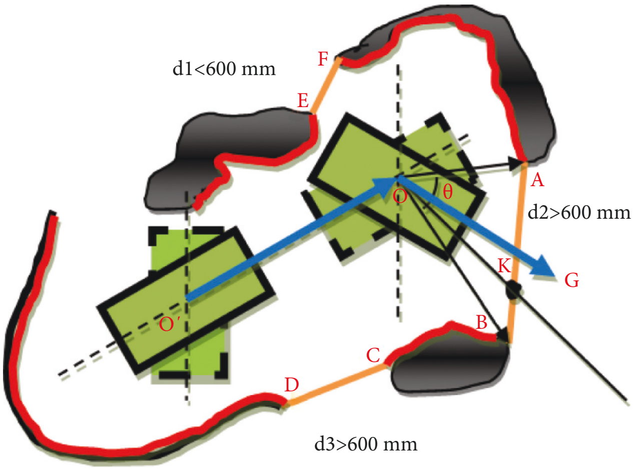


Figure 14: The movement of the mobile robot in the dynamic unknown environment.

Formula (4) can be decomposed by Bayes formulas, so the probability of pose under the condition of obtaining sensor value and observation value can be decomposed as the following formula:

Table 1: The relationship between angle of destination and turning direction.

The angle of destination compared the current orientation	Turning direction
If stop_forward = 0 and angle of destination is between 340° ~ 20°	Forward
If stop_forward > 0 and angle of destination is between 340° ~ 20°	Right
If angle of destination is between 20° ~ "160°"	Right
If stop_forward = 0 and angle of destination is between 160° ~ 200°	Backward
If stop_forward > 0 and angle of destination is between 160° ~ 200°	Right
If angle of destination is between 200° ~ "340°"	Left

$$\hat{x} = \arg\max_{x} \prod_{k=0}^{K} p\left(y_k|x_k\right)\cdot\left(p\left(x_0|\check{x}_0\right)\prod_{k=0}^{K} p\left(x_k|x_{k-1},v_k\right)\right). \tag{5}$$

Usually, the distribution of the probability of the input value and observed value obeys Gauss distribution; in order to simplify the expression and computation, the maximum likelihood estimation is computed; for example, prior probability before the observation in each step can be expressed as the following:

$$\ln p\left(x_k|x_{k-1},v_k\right) = -\frac{1}{2}\left(x_k - A_{k-1}x_{k-1} - v_k\right)^T Q_k^{-1}\left(x_k - A_{k-1}x_{k-1} - v_k\right) - \frac{1}{2}\ln\left(\left(2\pi\right)^M \det Q_k\right)^T, \tag{6}$$

where Q_k is the noise in the moving process and A_{k-1} is the transfer matrix at time k-1 because the first part of the formula (6) is relative to the states x, which can be expressed as

$$J_{(v,k)}(x) = -\frac{1}{2}\left(x_k - A_{k-1}x_{k-1} - v_k\right)^T Q_k^{-1}\left(x_k - A_{k-1}x_{k-1} - v_k\right), \tag{7}$$

where $J_{(v,k)}(x)$ stands for the error from the real value to the estimated value, so it is necessary to get the minimum value of $J_{(v,k)}(x)$ as follows:

$$\frac{\partial J_{(v,k)}(x)}{\partial x^T} = 0, \tag{8}$$

where it means that when $J_{(v,k)}(x)$ get the minimum value, x can get the suitable estimated value whose probability is the maximum value; in this way, the EKF function can be deduced.

From Figure 10, it is obvious that, after the processing of EKF algorithm, the error of the yaw value is still large. EKF algorithm only considers the linear system; for the assessment of the nonlinear system, EKF algorithm will not be effective because it ignores Taylor's high-order term, which makes the big error of the status assessment. Because the particle filter algorithm has no limit for the system noise, it is taken as the good solution for the nonlinear Gaussian system [10–12].

However, as the number of particles increases, the computational complexity will also increase. In addition, due to the sequential importance sampling particle filter, the samples with bigger weight are chosen for many times, resulting in a decline in the diversity of the particle collection of samples, and thus, there will be sample depletion problems [13–15].

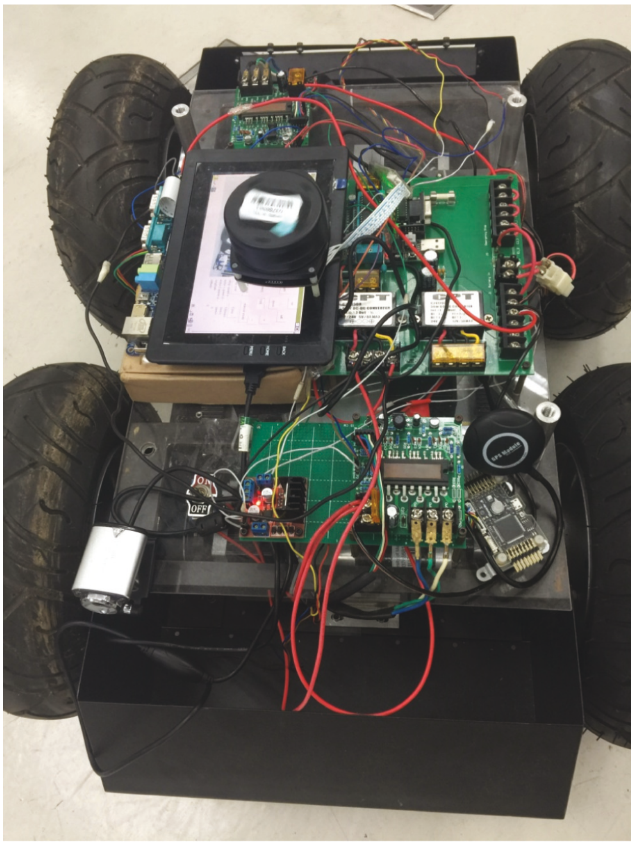
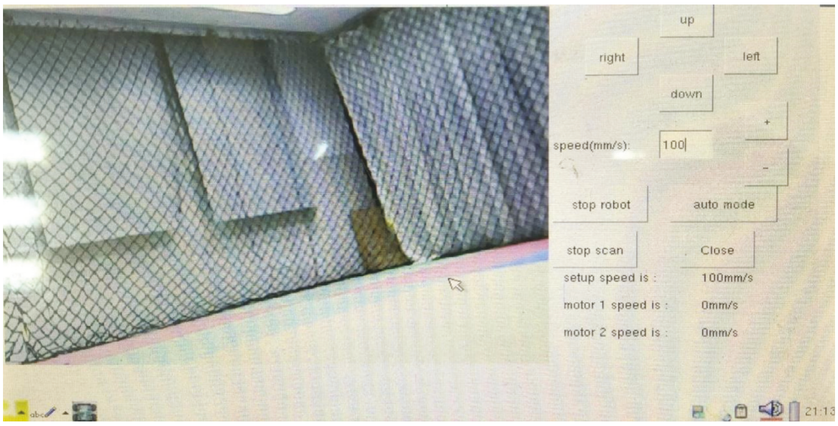


Figure 15: The designed mobile robot.

Fast SLAM is the universal SLAM technology, which can run the particle filter with few particles; similarly, low-dimensional KF is used to filter the surrounding features with known absolute position and consistency. Adaptive resampling algorithm can improve Fast SLAM algorithm; it sets the threshold number of resampling particles as 0.6 times of particles, which can take effective resampling operation by real time, computing effective particle number and particle degradation degree, to deal with the problem "sample degeneration" and "sample impoverishment" which is result in the frequent resampling [16].

The test between Fast SLAM and adaptive Fast SLAM is conducted on the mobile robot; in Figure 11, it is shown that when the mobile robot moves for a longer time, the yaw value error in adaptive FAST SLAM is much lower than it in FAST SLAM, so the yaw value which comes from odometer will be more accurate if they are processed by adaptive FAST SLAM [17].



speed(mm/s): 100

setup speed is : 100mm/s

motor 1 speed is : 0mm/s

motor 2 speed is : 0mm/s

Figure 16: The LCD and touchscreen with camera video display in the embedded Linux system.

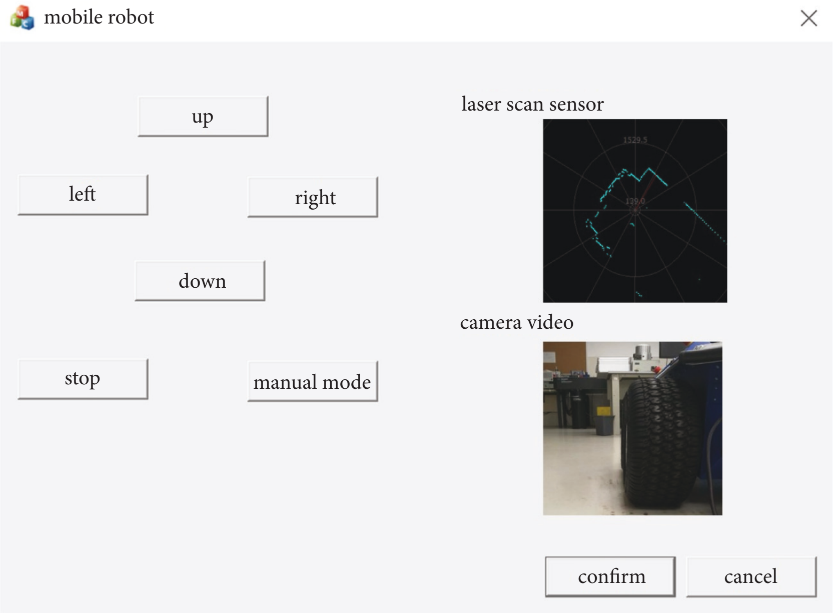


Figure 17: The software on the PC.

2.3. Decision Tree Algorithm.

2.3. Decision Tree Algorithm. Lots of the data are obtained from the IMU sensor fusion module and the odometer every second, which can be analyzed through machine learning algorithms; considering the limited CPU and battery power of the mobile robot, the complex machine learning algorithms are not suitable for running on the mobile robot. For example, SVM algorithm can convert the data from low dimension to high dimension to classify the data into categories, but it will consume lots of hardware resource and take long time [18], so decision tree algorithm is suitable to run on the mobile robot to predict the yaw value every second because it runs fast and save CPU computation compared with the other machine learning algorithms.

Usually, decision tree algorithm can be used on the classification, but not prediction [19]; if the yaw value can be divided into 360 classifications, every classification stands for one degree; then, the classification can be converted into prediction.

The true yaw value can be measured by the round ruler, and there are only two attributes data, one is yaw value comes from the IMU sensor module and the other is the yaw value which comes from the odometer which is processed by adaptive FAST SLAM. The large number data can be divided into training set, test set, and validation set, with the K fold cross-validation method; then, the data can be trained to get a decision tree model [20]. It should be noted that the data should contain the yaw value when the mobile robot is in some special status, such as the wheel of the mobile robot slips when mobile robot collides with a heavy obstacle in the dynamic environment, or the mobile robot in the dense cluster of buildings where the Earth magnetic field is very weak and the outputted data from IMU sensor fusion module are vibrating dramatically; in this way, the overfitting ability of the model will be intensified.

In Figure 12, the yaw value can be classified into 360 classifications through the processing of decision tree algorithm, and each classification stands for the predicted yaw value in each second for the mobile robot; if Figure 12 is zoomed larger, some details can be visualized as shown in Figure 13.

In Figure 13, the each classification is distributed on the end point of the tree, the value in each end point is a matrix that contains 360 integers, in which one is 1 and others are 0, the position of integer 1 in the matrix stands for the corresponding yaw value. The outputted accuracy of the yaw value from the decision tree can reach 99%; it is very near the true yaw value which is measured by the round ruler, which provides a method to predict the true yaw value of the mobile robot in the indoor environment.

After the accurate yaw value is obtained in the indoor dynamic environment, the following is to judge the next step movement when mobile robot is in the dynamic unknown environment, as shown in Figure 14; the black object in Figure 14 is the dynamic moving object; the red line is the scanned environment from the laser scanning sensor. If the length of the mobile robot is 600 mm, it will go from the gap between moving objects whose length is larger than 600 mm while whose orientation is most near the angle of the destination. The distance and angle of the obstacles can be obtained from the laser scanning sensor, the location of the mobile robot can be computed from the odometer through adaptive FAST SLAM algorithm, and the orientation of the mobile robot can be computed through the decision tree algorithm. If there is an object appearing in the “320°” ~“40°” of the mobile robot, the mobile robot will stop move forward and start rotate left or right to move bypass the object; in this situation, the bit stop_forware is 1, else it is 0; based on the above analysis, the next step movement can be judged according to Table 1.

According to the above analysis, the mobile robot can move to the destination without colliding with moving obstacles in the indoor dynamic unknown environment.

3. Experiment

A mobile robot is designed by us; the CPU is S5PV210 CPU with ARM structure and OS is embedded Linux system, which is shown in Figure 15. On each side of the mobile robot, the belt connects 2 wheels that are driven by the DC motor and 1 : 50 reduction ratio gear motor, so the mobile robot can bear heavy object on it.

For hardware of the mobile robot, the first is the main board with “S5PV210” CPU and various connectors, which connecting the following IMU sensor fusion module, USB camera, LCD display, and touch screen operation, Serial port to Bluetooth module, USB to wifi module, and laser scan sensor. The second is the MCU board; it can send the speed of the motors to the serial port of mainboard and can receive the motor control command from the mainboard through the designed protocol. Because the Linux system is not the real-time operating system, the ARM board cannot read the high frequent pulse accurately, so the MCU board takes the task to read the 25 KHz motor speed pulse. The third part is the power board with SCM6716 chipset to supply the large current and power, and the peak current can reach 1.8 A, which is assurance that the mobile robot can carry the heavy objects.

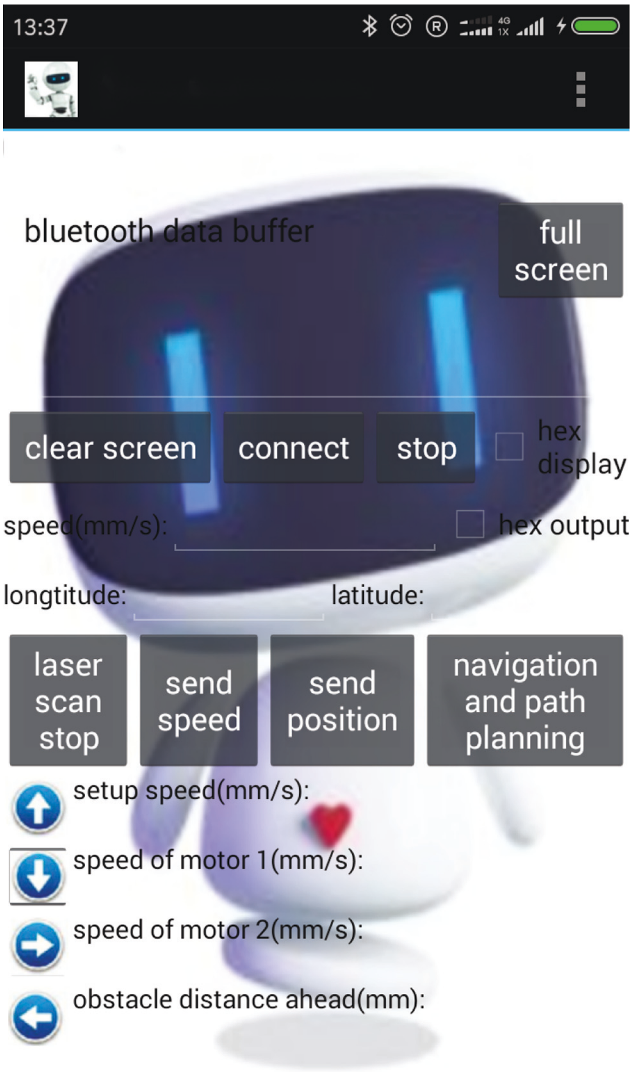


Figure 18: Android APP.

Various widgets on the LCD is designed with the embedded QT in the embedded Linux system, which is shown in Figure 16; through the signal and slot principle of QT, it can fulfill the LCD and touch screen functions. The software on the PC is designed to control the mobile robot and also to show the camera video and laser scanning environment, as shown in Figure 17. The Android APP is designed to communicate with the mainboard with Bluetooth protocol, which can send the value and command of the mobile robot to the Serial-Bluetooth module on the mainboard, as shown in Figure 18. The mjpeg-streamer is a universal tool for the remote monitoring in the B/S structure through the TCP/IP protocol, which can run on the mobile robot in the embedded Linux system; the camera video can be shown on Firefox browser in the Windows system with mjpeg-streamer as shown in Figure 19.

The map of the laboratory environment is not stored in the mobile robot; in order to test the navigation algorithm in the real world environment, the mobile robot is running in our laboratory environment which is the indoor dynamic

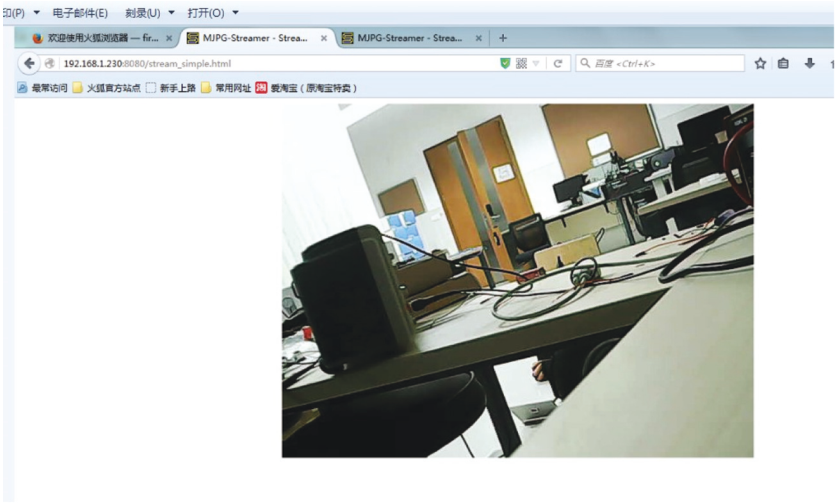


Figure 19: Remote monitoring on Firefox browser.

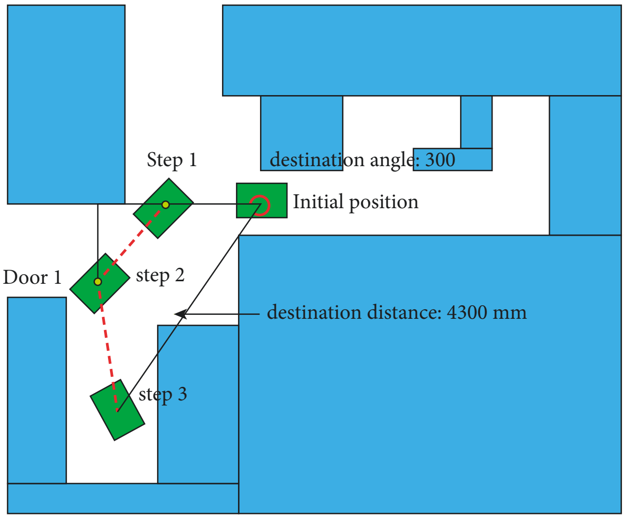


Figure 20: Navigation test of the mobile robot in the laboratory environment.

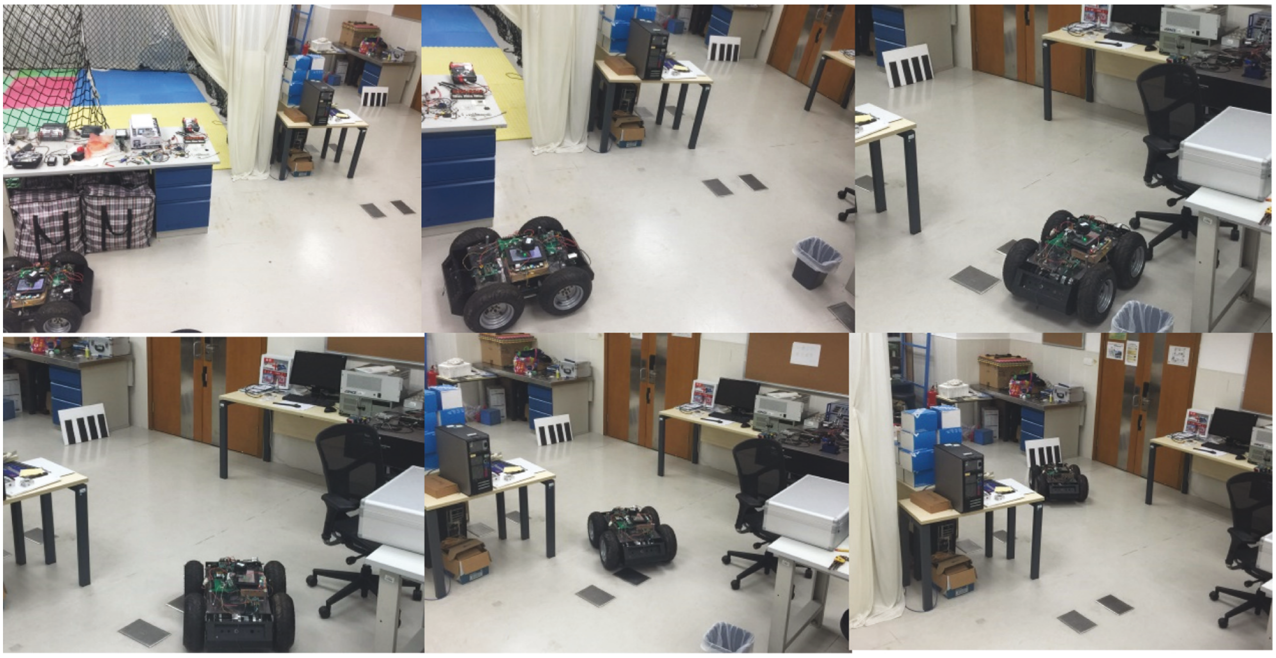


Figure 21: The navigation test on the designed mobile robot in the laboratory environment.

unknown environment just as shown in Figures 20 and 21; compared with initial position, the angle of destination is 300° and the distance of destination is 4300 mm which can be inputted in the Android APP; the orientation of the mobile robot can be computed according to the decision tree algorithm; the turning direction is judged according to Table 1 in every second; the angle of destination and distance is calculated and updated again in every second; finally, it can move to the destination without colliding with the obstacles on the path.

4. Conclusion

The testing result shows the proposed navigation algorithm is effective and real time responding in the indoor dynamic unknown environment. Compared with other algorithms, the mobile robot can obtain the accurate orientation through the decision tree algorithm in the indoor environment; it can obtain its location through the odometer data which are processed by adaptive FAST SLAM algorithm in the dynamic and unknown environment; it can scan the distance and angle of moving obstacles through the outputted data from the laser scanning sensor, which provide the data to judge the next step movement as shown in Table 1 in the dynamic and unknown environment.

There are some challenges in the navigation algorithm and realization of the mobile robot; one challenge is how to obtain the accurate orientation of the mobile robot in the indoor environment; through the decision tree algorithm, the predicted yaw value is much accurate especially when there is serious slipping of the wheels. The other challenge is that how to make the synchronization of the two threads; there is always errors when reading the outputted data from IMU sensor fusion module; finally, the problem detected that another thread is processing this data while this thread is reading this data; after adding the mutex lock on two threads, the none synchronous problem of two threads is solved and the reading and processing of the data from IMU sensor fusion module becomes normal.

The advantage of the proposed navigation algorithm is that it can be applied in the indoor dynamic unknown environment where there is the working environment for the home service robot; especially, this algorithm can save lots of CPU computation and battery power for the mobile robot which runs on the embedded platform. The next step work is to design and apply the deep learning algorithm on the mobile robot, which can recognize the human face and the appearance of home appliances, making the home service robot more stable and intelligent.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Featured Innovation Project of Guangdong Province in 2018 (2018GKTSCX056). Key Project of Guangdong Provincial Department of Education in 2019 (2019GZDXM014). The research project of Zhuhai City Polytechnic College in 2021 (KY2021Y01Z). Project of Ministry of Education of Higher Education in 2020 (2020ITA03008).

References

[1] R. Cui, Y. Li, and W. Yan, “Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 993–1004, 2016.

[2] Y. Wang, P He, P. Shi, and H. Zhang, “Fault detection for systems with model uncertainty and disturbance via coprime factorization and gap metric,” *IEEE Transactions on Cybernetics*, vol. 2, no. 8, pp. 1–11, 2021.

[3] P. He, “Consensus of uncertain parabolic PDE agents via adaptive unit-vector control scheme,” *IET Control Theory & Applications*, vol. 12, no. 18, pp. 2488–2494, 2018.

[4] Y. Li and Y. Liu, “Real-time tip-over prevention and path following control for redundant nonholonomic mobile modular manipulators via fuzzy and neural-fuzzy approaches,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 128, no. 4, pp. 753–764, 2006.

[5] Y. P. Yan and S. F. Wong, “A Navigation Algorithm of the Mobile Robot in the Indoor and Dynamic Environment Based on the PF-SLAM Algorithm,” *Cluster Computing*, vol. 22, pp. 1–12.

[6] Y. P. Yan and Y. M. Li, “Mobile robot autonomous path planning based on fuzzy logic and filter smoothing in dynamic environment,” in *Proceedings of the 12th World Congress on Intelligent Control and Automation*, pp. 1479–1484, Guilin, China, June 2016.

[7] P. He, Y. Li, and J. H. Park, “Noise tolerance leader-following of high-order nonlinear dynamical multi-agent systems with switching topology and communication delay,” *Journal of the Franklin Institute*, vol. 353, no. 1, pp. 108–143, 2016.

[8] Y. G. Yue and P. He, “A comprehensive survey on the reliability of mobile wireless sensor networks: taxonomy, challenges, and future directions,” *Information Fusion*, vol. 44, pp. 188–204, 2018.

[9] Y. M. Li and Y. Liu, “Fuzzy logic self-motion planning and robust adaptive control for tip-over avoidance of redundant mobile modular manipulators,” in *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics(AIM)*, pp. 1281–1286, Monterey, CA, U.S.A, July 2005.

[10] S. Zhu and L. Xia, “Human action recognition based on fusion features extraction of adaptive background subtraction and optical flow model,” *Mathematical Problems in Engineering*, vol. 2015, pp. 1–11, Article ID 387464, 2015.

[11] Y. P. Yan and Y. M. Li, “Autonomous path planning and navigation of a mobile robot with multi-sensors based on fuzzy logic in dynamic environment,” *International Journal of Intelligent Systems and Applications in Robotics*, vol. 7, no. 1, pp. 1–14, 2016.

[12] K. Zhao and Y. M. Li, “Path Planning in Large-Scale Indoor Environment Using RRT,” in *Proceedings of the 32nd Chinese Control Conference (CCC)*, pp. 5993–5998, Xian, China, July 2013.

[13] T. Ohki, K. Nagatani, and K. Yoshida, “Local path planner for mobile robot in dynamic environment based on distance time transform method,” *Advanced Robotics*, vol. 26, no. 14, pp. 1623–1647, 2012.

[14] E. Keshavarz and E. Khorram, “A fuzzy shortest path with the highest reliability,” *Journal of Computational and Applied Mathematics*, vol. 230, no. 1, pp. 204–212, 2009.

[15] S. Zhu and L. Xia, “Pain expression recognition based on pLSA model,” *The Scientific World Journal*, vol. 2014, pp. 1–8, Article ID 736106, 2014.

[16] Y. P. Yan and S. F. Wong, “A Navigation Algorithm for the Mobile Robot in the Indoor and Dynamic Environment Based on the Image Recognition Algorithm,” *The Hong Kong Institution of Engineers*, vol. 2, no. 1, pp. 1–12, 2018.

[17] X. N Shen, Y. Guo, and W. L. Hu, “Mobile robot path planning based on multi-objective evolutionary programming,” in *Proceedings of the 6th World Congress on Intelligent Control and Automation*, pp. 3114–3118, Dalian, China, June 2006.

[18] X. H. Liu and S. F. Wong, “Mechanism using taguchi-based differential evolution algorithm to optimize the bicycle industry in China,” *Industrial Engineering & Management Systems*, vol. 17, no. 1, pp. 72–81, 2018.

[19] S. Zhu, “Automatic recognition of facial expression based on computer vision,” *International Journal On Smart Sensing And Intelligent Systems*, vol. 8, no. 3, pp. 1464–1483, 2015.

[20] X. Chen and Y. Li, “Smooth path planning of a mobile robot using stochastic particle swarm optimization,” in *Proceedings of the 2006 International Conference on Mechatronics and Automation*, pp. 1722–1727, Luoyang, China, June 2006.