NON-CONTACT SERVICE ROBOT DEVELOPMENT IN FAST-FOOD RESTAURANTS

CHIN S. CHEN1, (Member, IEEE), CHIA J. LIN12, and Chun C. Lai3

1Graduate Institute of Automation Technology, National Taipei University of Technology, Taipei 10608, Taiwan, R.O.C.
2TECO Group Research Institute, TECO Electric & Machinery Co., Ltd., 10F, No. 3-1, Park St., Nan-Kang, Taipei 11503, Taiwan, R.O.C.
3Department of Electronic Engineering, National Chin-Yi University of Technology, Taiping, Taichung, 411030, Taiwan, R.O.C.

Corresponding author: Chun C. Lai (e-mail: cclai@ncut.edu.tw)

This work was supported in part by the industrial technology development program of MOEA, R.O.C. under Grant 107-0102

ABSTRACT Given the prospect of low birth rate and aging population among developed countries, which further resulted in the shortage of workforces, service robots have been gradually applied to real-life from past academic research. This paper introduces the development of a service robot that is designed for food service in fast-food restaurants with the innovative improvement of mapping, localization, and navigation. Moreover, this research took the initiative of integrating 3D point cloud map and 2D occupancy grid map (OGM) in order to build a PC-OGM. In another word, using the sensory fusion method allows the service robot to adapt to a more complicated environment as well as enhance its positioning. In terms of its navigation function, the adaptive motion controller is refined so the service robot could navigate through narrow aisles smoothly. Finally, friendly contact-free food service robots were evaluated at fast-food restaurants in order to gain feedback from diners and waiters. Their feedback was broken down into 3 categories, availability, reliability, and satisfaction, for further analysis. As COVID-19 assails the world, we also look into future possibilities of food service robot deployment among restaurants to keep food and surface free from the virus.

INDEX TERMS Service Robot, Robotics and Automation, SLAM, Non-Contact Service, System Integration.

I. INTRODUCTION

DECLINING birth rate and aging population among developed countries has become a global concern. The above circumstance is especially attributable to the high cost and drop-out rate in the catering industry. As labor supply deteriorates significantly, the increased labor cost may encourage employers to adopt service robots. With the rapid development of artificial intelligence and space perception technology automation has gradually shifted into commercial services from traditional industrial applications. Therefore, the utilization of service robots is being presented to enter the field of practical life. The demand for service robot applications such as tour guides, entertainment purposes, and package delivery are being progressively accepted by the current market. The inquiry for automated digital conversion in the catering industry has led to a new revolution in business demand. Upon introducing service robots into the catering fields, the original labor-intensive work has been transformed into human-robot collaboration.

In this research, the design and implementation of a food service robot including software and hardware system architecture will be introduced respectively. Moreover, this implemented food service robot is specifically designed to meet the needs for deployment in fast-food restaurants. In order to improve the application of service robots, the estimation fusion method is used for incorporating 3D point cloud map into 2D occupancy grid map. Within the integration of the above two maps, PC-OGM is created for improving the spatial orientation of the robot as well as elevating positioning accuracy in complex environments. In terms of the navigation function of the service robot, the adaptive motion controller is refined so the service robot could navigate through narrow aisles smoothly.

The experimental results including the comparison of the default methods with long-term field testing have proven the robustness of the system operation. Finally, the benefits and opinions on applying this service robot in 15 actual fast-food restaurants in Taiwan are also being discussed. Moreover, the
possibility of service robot deployment during COVID-19 outbreak periods in order to maintain business while lowering infection rate was carried out for further discussion.

II. RELATED WORKS

A. LOCALIZATION AND MAPPING PROBLEM

An intelligent mobile robot is an artificial intelligence system that applies multiple sensors to perceive environmental status to complete the task of navigating in uncertain environments. One of the fundamental challenges of intelligent mobile robot development is simultaneous localization and mapping problem (SLAM) also known as concurrent mapping and localization (CML).

Cadena et al. [1] have defined three eras in the historical advancement of SLAM. The classical age (1986-2004) covered the essential formulations for SLAM characterized by Extended Kalman Filter (EKF) [2], Rao-Blackwellized particle filters (RBPF) [3] [4], and maximum likelihood estimation (MLE) [5] [6]. The posterior age (2004-2015) is expressed and analyzed via algorithmic analysis for examining the observability, convergence, and consistency properties of SLAM. At present, the robust-perception age is taking place in SLAM research and is delineated by four essential criteria: robust performance, high-level understanding, resource awareness, and task-driven perception. All of them are still unsolved challenges. To date, the most popular 2D laser-based SLAM is known as GMapping [4] [7] which is a highly efficient RBPF. The Cartographer [8] is derived from a graph optimized to support 2D and 3D mapping. Another one is the vision-based SLAM, which is also known as ORB-SLAM2 [9] [10] [11]. It has been established as the state-of-the-art algorithm from the camera.

In detail, the multi-sensor fusion method is frequently used on SLAM in order to minimize or even eliminate accumulated errors from sensors such as odometer and inertial measurement unit (IMU) [12] [13]. However, both 2D laser-based SLAM and vision-based SLAM all have blind spots and deficiencies when applied in the actual hybrid environment. For example, 2D laser-based SLAM is susceptible to the obstruction and interference of environmental dynamic objects, which leads to increased observation errors. Moreover, it is limited by the spatial information scanning range of the horizontal plane. As for vision-based SLAM, algorithm failure may occur in the case of fast camera movement due to insufficient light or lack of visual features. Therefore, this research specifically focuses on the fusion of laser-vision SLAM. We expanded the use of Multi-Sensor Estimation Fusion [14] [15] from heterogeneity methodologies, and successfully combined the simultaneous operation of 2D laser-based SLAM and vision-based SLAM. It has the advantages of complementary functions to enhance the complex environment’s synthetic graphics and strengthen positioning.

B. THE DEVELOPMENT OF SERVICE ROBOT

In past decades, SLAM algorithms have been widely accepted and demonstrated among public areas. For example, museums [16], supermarkets [17] [18], and shopping malls [19], etc. They preliminarily integrated mapping, localization, navigation, interaction, and telepresence modules into a system. In the next decades, it is forecasted that the intelligent service robots’ market will grow rapidly around the world. However, the trend of using intelligent service robot varies between countries due to population structure and cultural awareness. For example, food service robots or robotic waiters deployed in leading chain restaurants in Asia have received positive results in terms of business growth.

Fig. 1 is the well-known MK Restaurant Robot [20] performing food service in a restaurant in Thailand. MK Company owns more than 450 restaurants in Thailand and the owner proposed the idea to use robots for service tasks in all of his restaurants. Though MK company’s robots have successfully increased its business by attracting more diners, limitations still exist. For example, the service robot navigates by following lines on the ground and requires a waiter in attendance at the table to pick up and serve the food to the diners. Therefore, the service robot deployed at MK’s restaurants only has little effect on lowering labor costs for the company.

C. THE CHALLENGES OF FOOD SERVICE

From the above reference and literature review, service robots are widely deployed for different purposes such as a tour guide, a shopping assistant, or waiters. Berezina et al. [33] have evaluated the intelligent service in the restaurant industry and Zemke, et al. [34] have investigated the diner perspectives by using robot technology in quick-service restaurant (QSR) operations. Among all the advantages and benefits gained from service robot deployment, there are still rooms for improvement as well as existing challenges. Below are the challenges for food service robots in fast-food restaurants which this research has overcome.

FIGURE 1. MK restaurant service robot (a) the staff places the food on the robot tray at robot station (b) the staff enters the table number to the robot (c) the robot moves to destination table using line tracing (d) the staff picks and places the food onto the table.
1) Environment Complexity
The 2D laser SLAM is unable to profile the overall environment map in a 2D grid map. As a result, the furniture and decoration layout in restaurants such as tables, chairs, wall cabinets, and screens, etc., will affect the robot’s navigation function during food service.

2) Crowds
Fast-food restaurants usually have busy traffic during peak hours. Therefore, crowds may block the view of the laser scanner or camera, which further causes the self-localization function to fail. As a result, increasing the robustness of robot self-localization is a critical problem to be solved during robot development.

3) Narrow Aisle Between Tables
The aisle between tables in a fast-food restaurant is about 70-90 centimeters, which greatly increases the challenge in robot design as the physical shape of the robot needs to be compacted with good motion control in order to stroll along the narrow aisle.

4) Friendly Service Etiquette
Traditionally, the waiter serves the food and greets the diners to make them feel warm and comfortable during their food service. In order to have the diners feel the same way, friendly interaction functions should be blended into the design of the service robot.

Considering the above challenges, the development trend of the autonomous fast-food service robot system is focused on improving environment map building precision, increasing the accuracy of self-localization, ensuring smooth navigating in the narrow aisle, and adding suitable interaction with diners.

III. FAST-FOOD SERVICE ROBOT DESIGN
In this section, the design concept of the food service robot will be elaborated on first. Then, an overview of the hardware architecture of the service robot is provided, and the software modules based on the Robot Operating System (ROS) [21] development which is divided into application and core layer will be introduced. Finally, detailed procedures of the food service process will be explained.

A. THE DESIGN CONCEPT
The design concept for fast-food service robots needs to take diners’ acceptance and sensitivity into consideration, as well as the practicality and convenience of the staff. Besides, it’s crucial for the robot’s design to adapt to the fast-food restaurant environment in order to improve food service efficiency. Furthermore, avoiding direct contact between operators and customers is a plus for pandemic prevention.

In terms of customer acceptance, the design is based on cartoons with a friendly appearance, creating an anthropomorphic look of a waiter substitute in order to provide courteous food service. The appearance of the robot is exquisite and centralized, and its height is limited so the diners will not feel oppressed. At the same time, when the food is served, it needs to be easy to pick up by the diners. At present, 15 food service robots have been deployed at a famous fast-food restaurant, MOS Burger, in Taiwan, providing daily food service. The deployment of food service robot is not only welcomed by children but also enhances the non-contact safety procedure between staff and customers as well as increases mutual trust.

B. HARDWARE ARCHITECTURE
The food service robot deployed at MOS Burger wears a white apron with an orange neck and arm shown in Fig. 2. The outer case, around 120 cm in height, is made of plastic materials that are certified for food application. The length and width of the food service robot are both 40 cm x 40 cm, which allows it to move freely along narrow aisles.

The face of the service robot is a LED display that can interact with diners via voice and text. The position of the neck is equipped with an RGBD camera to detect obstacles along its navigation path. To allow the service robot to interact with diners in a noisy environment, the robot is equipped with a 60 watt speaker at its lower body.

The bottom of the robot has a detachable mobile platform that is implemented with a set of differential motion system of 150 watt servo drives. The robot could continuously work for 8 hours in the service field with its detachable 24 v / 36 ah battery system.

As restaurants are normally densely populated during peak hours, a food service robot needs to achieve precise environmental perception. In order to do so, the robot is equipped with a 2D laser scanner, providing a 25 m scanning range at the bottom and an RGBD camera embedded in the neck for environmental detection. As to strengthening the safety of food service during peak hours, 6 ultrasound sensors are loaded at the bottom of the robot. Therefore, when it encounters an obstacle, it will stop and release a warning voice message and continue its route when the obstacle is out of its way. Furthermore, an emergency stop switch is built at the bottom of the robot to prevent sudden accidents which allow the robot to stop immediately upon pressing the switch.

FIGURE 2. The hardware architecture of fast-food service robot.
C. SOFTWARE ARCHITECTURE

The architecture of the system software design is based on ROS communication and can be broken down into Drivers Layer, Control Core Layer, and Application Layer as shown in Fig. 3. The Driver Layer drives the relevant hardware, and the Control Core Layer is responsible for the robot’s mapping, localization, and navigation. The Application Layer is corresponding to customer service and the waiter’s operation.

The advantages of ROS are that it provides a set of robotic software libraries and a reliable communication mechanism for modular nodes. Therefore, all the programs can execute and communicate easily. It not only accelerates the speed of development, but also reduces the complexity of system maintenance.

1) Driver Layer
Each peripheral hardware of the service robot is corresponding with the driver for ROS communication such as an RGBD camera, a 2D laser scanner, and a battery management system (BMS) module which transmits power status periodically. On the other hand, the servo motor driver publishes the odometer topic and subscribes motion command, then the audio driver gets the playing music command during food service or issuing voice for prompts. For example, when the robot reaches the table: "enjoy your meal" or when the robot encounters people: "coming through".

2) Control Core Layer
The major core control layer includes mapping module, localization module, and navigation module. In the mapping module, the Extended Kalman Filter (EKF) method was amplified in order to fuse vision SLAM and 2D laser SLAM. Synchronously, the combination of 3D point cloud map, generated by vision SLAM, and the occupancy grid map (OGM), constructed by 2D laser scanner, were used to form a new fusion point cloud-occupancy grid map (PC-OGM). In the localization module, based on the above PC-OGM, EKF fusion method is used again, which further combines the pose estimation of vision SLAM and the positioning estimation based on 2D laser-based adaptive Monte Carlo localization (AMCL) [22] [29] to achieve robust and complementary positioning performance. For the navigation module, the application of an adaptive motion controller was proposed. By combining the commonly used dynamic window approach (DWA) [30] method and model-free adaptive control (MFAC) [31] model to achieve perfect follow-up control of the preset path. As a result of the above improvements, the robot is able to easily move through the narrow aisle in the restaurant.

3) Application Layer
The purpose of human-robot interaction (HRI) is to provide the operator with a convenient and safe interface. Fig. 4 (a) shows the design of a web-based interface operating on a tablet in order to control the service robot. Besides, to avoid contamination by frequently touching the tablet among operators, voice commands can also be used to control the robot as shown in Fig. 4 (b). The customer location module was used with the navigation script module. By using the RF beacon, the robot is able to identify the diners’ table and execute the corresponding navigation script. Finally, the scheduling module dispatches the service request and the operation status module keeps the robot status in a system log for daily review.

4) Food Service Process
The robot is able to interact with the diners by the text message shown on its LED display, the arm tray sensor, and the sound system. The service processes are as shown from
sequences 1 to 6 in Fig. 5. The diners will be given an RF tray beacon after placing their meal order shown in seq. 1. In seq. 2, when the meal is ready, the operator launches the robot via a tablet. As the robot delivers to the designated table, its arm tray will be extended while reminding the diner to take the meal with voice and facial messages shown in seq. 3, 4. The arm tray sensor detects whether the diner took the meal or not as shown in seq. 5. In seq. 6, the robot returns to its original position after completing this service and gets ready for the next one. If the service is not completed, the remote tablet will show a message to notify the current service status.

IV. SLAM IN COMPLEX ENVIRONMENT

In mobile robotics, occupancy grid map (OGM) is a classic and frequently used method to represent the robot’s surroundings. This representation, proposed by Moravec and Elfes [23] [24] ordinarily makes use of two or three juxtaposed dimensions to transform a real space into an array of cells, where each cell contains information of the occupancy probability and the spatial uncertainty of the environment. The smaller the spatial uncertainty, the lesser the number of cells occupied by any object [25]. Nowadays, 2D laser scanners are widely used for building occupancy grid map. However, using only one 2D laser scanner may be insufficient and less reliable for accomplishing motion tasks in restaurant environments. This is due to the fact that a physical 2D laser scanner only scans a horizontal plane which may be shaded by some dynamic obstacles during run time. On the other hand, the biggest problem of vision SLAM is when facing a single wall; it will not be able to recognize any visual features, causing SLAM to be interrupted. To solve the above problem, a PC-OGM map is developed by fusing vision SLAM and 2D laser SLAM in order to increase SLAM precision for mapping such as the PC-OGM map mentioned previously. The PC-OGM fusion flow diagram is shown in Fig. 6.

![PC-OGM flow diagram](image)

To add, PC-OGM has improved the spatial positioning accuracy of the original robot which only uses 3D vision SLAM or 2D laser SLAM. PC-OGM is able to provide a more refined environmental mapping in a complex dining environment. Besides, it can also avoid incorrect environmental mapping information which affects real-time positioning and results in robot drift away from the original route.

A. 2D LASER SLAM

Upon reviewing common SLAM issues, if an ideal odometer were applied to record and trace the robot pose, then the 2D laser can be directly transferred to the occupancy grid map. Fig. 7 shows the simulation environment by gazebo [26] and Fig. 7 (b) and (c) shows the 2D-OGM mapping result with the above-mentioned ideal odometer. The green path outlines the real robot trajectory.

![2D laser SLAM with ideal odometer](image)

However, in reality, the odometer is estimated via robot wheel encoder. One disadvantage of how the odometer is estimated is that the skidding effect between the wheel and ground will affect the accuracy of the odometer. In order to approximate the real situation, Gaussian noise is added to the ideal simulated odometer. Due to the noise recorded during odometer estimation and its cumulative error, the deviation between the robot and the actual position and direction angle will become larger and larger. If the information obtained by the odometer is directly transferred from the 2D laser to the OGM, the OGM map will show a very large deviation from the actual scenario as shown in Fig. 8. Therefore, this is also the problem needed to be solved by the SLAM algorithm.

The widely known 2D SLAM algorithm such as GMapping is a highly efficient Rao-Blackwellized particle filter [4]
generating the OGM from laser range data. As a result, the noise odometer and 2D laser scanner were both applied on GMapping. Fig. 9 (a) shows the GMapping OGM result; the green and yellow trajectories represent the ground truth and GMapping estimation. Fig. 9 (b) shows the distance error between GMapping and the ground truth. The maximum distance error is over 0.2 m when GMapping is running at the 490 time index.

Fig. 10 (a) shows the ORB vision features from the camera image and Fig. 10 (b) shows the ORB-SLAM2 3D point cloud mapping result; the green and blue trajectories represent ground truth and ORB-SLAM2 estimation respectively.

Fig. 10 (c) and (d) show that the robot is facing a wall without the ORB vision features near the end loop. When the robot is moving to the starting position and facing a wall without ORB vision features, the pose estimation will be interrupted. Fig. 10 (e) shows the maximum trajectory error is less than 0.08 m when ORB-SLAM2 continuously obtains the vision features or else the trajectory error will increase due to the ORB-SLAM failing to generate output while the robot is still moving.

From above simulation, the ORB-SLAM2 is able to obtain a more precise SLAM result than GMapping. But, if image sequences are not rich in texture information or the camera rotates at a large angle, the ORB-SLAM2 system will fail to generate output.

C. CONCURRENT LASER AND VISION SLAM

Intuitively by executing laser GMapping and vision ORB-SLAM2 at the same time, a 2D-OGM and 3D point cloud map can be obtained respectively. However, even if these two algorithms are executed at the same starting point, the positioning and construction of the two are not guaranteed for consistent mapping results during the process. The results were shown in Fig. 11 (a) and (b). In the middle part of the trajectory shown in Fig. 11 (b), the estimated position...
The overall objective is to estimate the full 3D pose and orientation of a mobile robot over time. It can be defined as a nonlinear dynamic system as follows:

\[ x_k = f(x_{k-1}) + w_{k-1} \]

where \( x_k \) is the robot’s system state (i.e., 3D pose) at time \( k \), \( f \) is a nonlinear state transition function, and \( w_{k-1} \) is the process noise, which is assumed to be normally distributed. The twelve-dimensional state vector, \( x \), involves the robot’s 3D pose, 3D orientation, and their respective velocities. Rotational values are expressed as Euler angles. Also, the measurements present the following form:

\[ z_k = h(x_k) + v_k \]

where \( z_k \) is the measurement at time \( k \), \( h \) is a nonlinear sensor model that maps the state into measurement space, and \( v_k \) is the customarily distributed measurement noise. The first stage in the algorithm is to carry out a prediction step that projects the current state estimation and error covariance forward in time. This process is shown in (3) and (4).

\[ \hat{x}_k = f(x_{k-1}) \]  
\[ \hat{P}_k = F\hat{P}_{k-1}F^T + Q \]

In robot motion, \( f \) is a standard 3D kinematic model derived from Newtonian mechanics. The estimated error covariance, \( P \), is projected via \( F \), the Jacobian of \( f \), and then perturbed by \( Q \), the process noise covariance. The next stage in the algorithm is to perform a correction step, which is summarized from the equations:

\[ K = \hat{P}_kH^T(H\hat{P}_kH^T + R)^{-1} \]
\[ x_k = \hat{x}_k + K(z - H\hat{x}_k) \]
\[ P_k = (I - KH)\hat{P}_k(I - KH)^T + KRK^T \]

The Kalman gain is computed using the observation matrix, \( H \), the measurement covariance, \( R \), and \( P_k \). The gain is utilized to update the state vector and covariance matrix. Besides, the Joseph form covariance is applied to update (7) to promote filter stability by ensuring that \( P_k \) remains positive semi-definite. The standard EKF formulation specifies that \( H \) should be a Jacobian matrix of the observation model function \( h \). To support an extensive array of sensors, it is assumed that each sensor produces measurements of the estimated state variables. A core feature of EKF Localization [27] is that it allows partial updates of the state vector and recognizes the integration of 15 states:

1. Position: \( x, y, z \)
2. Orientation: \( \text{yaw}(\phi), \text{pitch}(\theta), \text{roll}(\psi) \)
3. Linear Velocity: \( x', y', z' \)
4. Angular Velocity: \( \text{yaw}'(\phi'), \text{pitch}'(\theta'), \text{roll}'(\psi') \)
5. Linear Acceleration: \( x'', y'', z'' \)

All the trajectories and mapping results are shown in Fig. 12 (a). On the other hand, Fig. 12 (b) is the trajectory error compared between the 2D laser GMapping, vision ORB-SLAM2, and EKF-Fusion result. The EKF-Fusion from 2D laser SALM and 3D vision SLAM was able to generate a more precise trajectory output compared with the ground truth trajectory i.e. less trajectory error. Based on the EKF-Fusion, the 2D-OGM creator module can directly project the laser sensing to the ground level as an EKF-Fusion 2D mapping. As the trajectories produced by both vision ORB-SLAM2 and EKF-Fusion are closer to the ground truth. We were able to obtain higher consistency PC-OGM from 3D vision SLAM and 2D EKF-Fusion mapping.
FIGURE 12. EKF fusion of laser and vision SLAM (a) trajectories of robot ground truth (green), GMapping (yellow), ORB-SLAM2 (blue) and EKF-Fusion (red) (b) trajectories error with respect to ground truth for GMapping, ORB-SLAM2 and EKF-Fusion.

TABLE 1. Particle Filter Algorithm for Mobile Robot Localization.

| Input: | $X_{t-1}, u_t, z_t, m$ |
| Output: | $X_t$ |
| 1: | $\bar{X}_t = X_t = \phi$ |
| 2: | for $m = 1$ to $M$ do |
| 3: | $x_t^{[m]} = \text{sampl motion model}(u_t, x_{t-1}^{[m]})$ |
| 4: | $w_t^{[m]} = \text{measurement model}(z_t, x_t^{[m]}, m)$ |
| 5: | $\bar{X}_t = X_t + \text{weighted sample} <x_t^{[m]}, w_t^{[m]}>$ |
| end for |
| 6: | for $m = 1$ to $M$ do |
| 7: | draw $i$ with probability $\propto w_t^{[i]}$ |
| 8: | add $x_t^{[i]}$ to $X_t$ |
| 9: | end for |
| 10: | end for |
| 11: | return $X_t$ |

command, $z_t$ is robot measurement, and $M$ is the particle number.

However, in the real application, the 2D laser scanner on the robot may be obstructed by the crowds in the restaurant. The obstruction may cause the mobile robot to fail and become dangerous as it can only rely on the dead reckoning of the noise odometer. Fig. 13 (a) shows the AMCL simulation based on the ideal odometer, the trajectory is perfectly overlapping the ideal odometer. However, if AMCL were to receive the noise odometer, the trajectory error will gradually increase as shown in Fig. 13 (b). If we re-sprinkled the AMCL particles, the position error will converge rapidly. This simulation is relatively close to the AMCL applied in real robot localization.

**B. VISUAL ORB-SLAM2 IN LOCALIZATION MODE**

ORB-SLAM2 incorporates a "Localization Mode" which is useful for lightweight long-term localization among well-mapped areas, as long as no significant changes are made in the environment. In this mode, the local mapping and loop closing threads are deactivated and the camera is continuously localized by the tracking within re-localization if needed. Moreover, the tracking leverages visual odometer matches and matches to map points. Visual odometer matches are matches between ORB in the current frame and 3D points created in the previous frame from the stereo/depth information. These matches make the localization robust to unmapped regions, but drift can be accumulated. As a result, map point matches ensure drift-free localization to the existing map.

**C. EKF POSE FUSION**

For practical application, the visual ORB-SLAM2 is allowed to be executed in localization mode with the point cloud map of the previous PC-OGM. Also, the AMCL is applied for global pose estimation with 2D laser and 2D OGM of previous PC-OGM. In this research, the global pose estimation was fused from the visual ORB-SLAM2 in localization mode; the particle-based Monte Carlo Localization. Fig. 14 shows the global localization flow is improved after fusing the ORB-SLAM2 pose, AMCL pose.

The EKF-Fusion localization simulation is shown in Fig. 15. In the left of Fig. 15 (a), the blue points show the ORB-SLAM2 vision features are matched in localization mode.
The right side of Fig. 15 (a) shows the trajectories of ground truth in green, AMCL in yellow, ORB-SLAM2 in blue and EKF-fusion result in red. Fig. 15 (b) shows the re-sprinkle of AMCL particles in order to converge the odometer error and Fig. 15 (c) shows the robot is moving closer to the starting point. Moreover, the trajectory error compared with the ground truth is shown in Fig. 15. The EKF-Fusion trajectory is relatively stable to approximate the ground truth whenever ORB-SLAM2 failed or the error of AMCL is higher.

FIGURE 14. EKF localization fusion flow.

VI. ADAPTIVE MOTION CONTROLLER

A. NAVIGATION OF SERVICE ROBOT

This section aims to explain the design and improvement of the robot’s navigation control in this research, which allows the robot being deployed to navigate and deliver meals in a complex environment successfully. In the design of navigation, the robot needs to obtain mapping information to plan its path and set the corresponding target point in the generated grid map for navigating purpose. The design of the path takes the global path and the local path planning as the main axis. In global path planning, the global path between the robot itself to the target is calculated, and the best path is generated based on the evaluation of the grid map. Among the classification algorithms, Dijkstra and A-Star are widely used to find the best path from a grid map. For local path planning, the global path is taken into consideration before planning a suitable and safe route. Sensors are used to sense the robot’s surroundings in order to make the robot navigate smoothly through obstacles and ensure the navigation follows the global path. The current mainstream, the local planner of robot, is mainly based on the dynamic window approach (DWA) [30]. DWA is a velocity response controller adapted to the robot’s local obstacle avoidance. The cross combination of angular velocity (linear velocity \( v \) and angular velocity \( w \)) matches the posture of the robot to calculate multiple sets of multiple trajectories, represented by a series of arcs with different curvatures, and obtain the best solution as shown in Fig. 16.

FIGURE 15. EKF-Fusion localization simulation result (a) ground truth (green) AMCL (yellow), ORB-SLAM2 (blue) and EKF-Fusion result (red) (b) re-sprinkle the AMCL particles (c) return to the starting point (d) the trajectory error with respect to the ground truth.

FIGURE 16. DWA trajectories simulation.

The sampled trajectories are evaluated by an objective function \( O(v, w) \), which is consisted with the goal heading, obstacle clearance, and velocity as the following listed criteria:

\[
O(v, w) = \sigma(\alpha \times \text{heading}(v, w)) + \beta \times \text{dist}(v, w) + \gamma \times \text{velocity}(v, w) \tag{8}
\]

The \( \text{heading}(v, w) \) is the angle difference between current head orientation of the robot and to the direction of the goal, the \( \text{dist}(v, w) \) is the distance to the closest obstacle, and the \( \text{velocity}(v, w) \) presents the forward velocity of the robot with kinematic constraints.
B. NAVIGATION IN A COMPLEX ENVIRONMENT

In the operation of automatic navigation, the robot will generally use the A-Star or Dijkstra algorithm to generate the shortest path planning and DWA to achieve global path following and local obstacle avoidance. It has the characteristics of good flexibility and strong adaptability to the static environment in a large open field. However, in a narrow dining environment, as shown in Fig. 17, accompanied by environmental features such as the prominent dining table A-Star shortest path planning, it will cause the robot to easily collide with the corner of the table when navigating, causing delivering failure. In view of this, in the navigation module, an adaptive motion controller is proposed to allow the robot to run stably and follow the established food delivery path.

![FIGURE 17. The narrow dining environment.](image)

The adaptive motion controller design architecture diagram is shown in Fig. 18. The original idea is to modify the automatic generation of the original global path and load the preset path points. Because the preset path points are pre-defined by the user, the design automatically adjusts to the appropriate path points and feeds them to the DWA in order to achieve a smooth path, allowing the robot to navigate in a narrow aisle on the scheduled meal delivery path.

![FIGURE 18. Adaptive motion controller framework.](image)

1) Pre-defined Path Loader
Loading the pre-defined position points as the robot’s travel path is able to effectively avoid the salient problem by the 3D shape of the space object profile; its format is defined as the point vector or array.

2) Path Point Regulator
Filtering function of the points of the pre-defined path. The main purpose is to clear inappropriate points or encounter obstacles in the pre-defined path. Automatically selecting the path points to be suitable for smooth following. The function is designed as follows:

Abnormal waypoint adjustment
\( \vec{v}_1 \) represents the vector as from the center of the robot to the current target point \( P_1 \), \( \vec{v}_2 \) represents the vector as from the target point \( P_1 \) to the second target point \( P_2 \), when the angle \( \Theta \) in (9) between the two vectors is greater than \( \pi/2 \) rad, it means that \( P_1 \) is the appropriate point for robot tracing which is shown in Fig. 19 (a). On the contrary, when the angle \( \Theta \) between the two vectors is smaller than \( \pi/2 \) rad as shown in Fig. 19 (b), it means that \( P_1 \) is an inappropriate way point. \( P_1 \) will be canceled and labeled it as \( P_{last} \).

\[
\Theta = \cos^{-1} \frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1||\vec{v}_2|} \quad (9)
\]

Obstacle point adjustment
In Fig. 19 (c) \( |\vec{v}_1| \) represents the distance from the center of the robot to the target point \( P_1 \), and \( |\vec{v}_l| \) represents the distance from the target point \( P_{last} \) to the target point \( P_1 \), although when \( |\vec{v}_1| > |\vec{v}_l| \) The robot should turn to the target point \( P_1 \), but because the path point \( P_1 \) has detected an obstacle, the robot cannot turn to pursue the target point \( P_1 \). In this case, it means that \( P_1 \) is an inappropriate way point to follow. Then point \( P_1 \) will be eliminated and marked as \( P_{last} \).

![FIGURE 19. Path point regulator.](image)

3) Path Point Feed Controller
In order to enable the robot to travel in a narrow aisle, this work introduces the method of Model Free Adaptive Control (MFAC) [31] [32], using the concept of pseudo gradient vector and order, it adopts a series of dynamic linear time-varying models in the robot path for replacing the general discrete-time nonlinear system. The distance between the
robot pose and the travel path is used as the observation and the pseudo-gradient vector is used as the selecting between the path points.

\begin{align}
  u(k) &= u(k-1) + \frac{\rho \phi(k)}{\lambda + \phi^2(k)} (y(k+1) - y(k)) \\
  \phi(k) &= \phi(k-1) + \frac{\eta \Delta u(k-1)}{u + \Delta u^2(k-1)} (\Delta y(k) - \phi(k-1) \Delta u(k-1)) \\
  \phi(k) &= \phi(1), \text{ if } \phi(k) \leq \epsilon, \text{ or } |\Delta u(k-1)| \leq \epsilon
\end{align}

From Fig. 19 (d), it can be seen that the distance \( H \) between the robot center and the line segment of path as in (13) and (14), which can be used as the performance evaluation of the robot following the path. The vector distance between the previous target point \( P_{last} \) and the target point \( P_1 \) is \( |\vec{v}_a| \), \( |\vec{v}_b| \) is the vector distance between the center point of the robot and the previous target point \( P_{last} \).

\begin{align}
  \vec{v}_a \times \vec{v}_b &= |\vec{v}_a||\vec{v}_b| \sin \phi \\
  H &= \frac{|\vec{v}_a \times \vec{v}_b|}{|\vec{v}_a|} \\
\end{align}

Via the MFAC model in (10), (11) and (12), we can regard \( H \) as \( y(k) \), and the corresponding \( u(k) \) can be obtained as the way point selecting for the DWA target path input.

4) DWA Path Planner

According to the Path Point Feed Controller, the corresponding target way point is matched and selected, and it will pass to DWA as in (8). DWA will convert and generate the best output velocity command for the motor servo drive. It will control the robot to move smoothly to the corresponding target point.

C. PATH FOLLOWING SIMULATION

The adaptive motion controller is simulated for path following result verification. Fig. 20 (a) shows a pre-defined circle path with a diameter of 2 m. Fig. 20 (b) shows the path following error with the time index. The maximum distance error of following the circle is about 0.0562 m and the mean distance error is about 0.0238 m.

VII. EXPERIMENTS AND FIELD TRIALS

A. PC-OGM MAPPING WITH EKF-FUSION

Fig. 21 shows the PC-OGM map of a fast-food restaurant that was created by the 2D-OGM creator with the 3D point cloud that was created by the ORB-SLAM2. Creating the map is crucial for the well-function of the localization system.

B. ROBUST LOCALIZATION WITH EKF-FUSION

The following images allow us to see the final pose after moving into the environment using the robot’s localization package. Fig. 22 (a) shows the matching ORB vision features and Fig. 22 (b) shows robust localization in the fast-food restaurant. When either of the AMCL localization based on laser scanner failed temporarily or the ORB-SLAM2 obtains fewer vision features from the camera, the EKF-Fusion robot pose is able to stay correctly on the map.

C. PATH FOLLOWER IN FAST-FOOD RESTAURANT

The actual deployment of service robot navigation in a fast-food restaurant is shown in Fig. 23 (a). In the fast-food restaurant, the pre-defined path is established to move around the restaurant as the green segment in Fig. 23 (b) to (e). The service robot was able to follow the path well and reach the pre-defined goal at last.

D. FEEDBACK OF SERVICE ROBOT DEPLOYMENT

The food service robot research and development project was launched in 2018 and continually tested and updated in 2019 and 2020. Since June 2019, the foodservice robots have been
FIGURE 23. Robot navigation in real fast-food restaurant for non-contact food service (a) service robot working in a real fast-food restaurant (b) robot starts to follow a long path (c) robot turn in left-up corner (d) robot turn in right-down corner (e) robot will reach final goal.

TABLE 2. Reasons Not to Use Service Robot.

| Reason                        | Percentage |
|-------------------------------|------------|
| Long time for food served     | 56%        |
| Staff without training        | 15%        |
| Staff insufficient training   | 7%         |
| Staff unfamiliar with operation | 9%       |
| Pre-fixed table requirement   | 14%        |
| Without sales activity        | 12%        |
| Customer refuse               | 10%        |

1) Availability
Compared with well-trained waiters, the service robot takes more time to serve food, which is the key reason (56%) that restaurant managers refused to deploy service robots as shown in Table 2. Moreover, the time needed for waiters to serve food is about one minute, while it takes 2 to 3 minutes for service robots per round trip. The waiter’s walking speed is about 70 cm/s, while the robot’s velocity is about 40–60 cm/s. On the other hand, some waiters are amateur employees or part-time students; 15% of them lack training and 9% are worried about the failure of robot service operation since they were not as familiar with the setting as the full-time waiters. However, from diners’ perspective, they have no problem waiting a few minutes longer for service robots to serve their food. In another word, diners not only are willing to wait longer for service robots to serve food they also found it interesting.

2) Reliability
The reasons for the failure of robot food service are listed in Table 3. According to the survey, the main reason (44%) for service robots to fail is the uncertainty of table and chair location. When a robot serves the food, the system will compare environmental characteristics with the original map. If the tables got rearranged from its’ default location, the robot couldn’t complete the food service. Besides, the existence of obstacles along a planned path for food delivery, such as backpacks, strollers, shopping bags, etc., will also interrupt food service by 18%. As a result, under the above circumstances, the robot will remind diners to pick up their food even when robots failed to navigate to its’ designated table. Furthermore, 14% of failure was caused by network interruption. When the surface of the 2D laser scanner is dirty, it will cause 9% of the error of sensor detection. Therefore, the scanner needs to be cleaned regularly.

3) Satisfaction
Table 4 indicates the satisfaction of staff and diners’ feedback for using service robot and is divided into four degrees. From the perspective of a waiter, about 11% are very satisfied, and deployed successively in 15 fast-food restaurants across Taiwan. On average, a deployed service robot regularly performs 44 food services per working day. Specifically, around 35 times during weekdays and 52 times during weekends. To improve system quality and service performance, a survey was conducted between the waiters and diners. The result of the survey is then presented in three different fields, availability, reliability, and satisfaction, respectively.
42% are satisfied. However, 30% of the waiter in the survey refused to deploy a service robot. As some store owners are concerned about the high cost of purchasing service robots and its’ maintenance fee which caused their staff to refuse using the robot.

The diners’ satisfaction is significantly higher than that of staffs’. Most customers felt fresh and pleasantly surprised by the service robot, about 57% are very satisfied and 35% are satisfied. Overall, customer feedbacks are positive, but the design of robots in human-machine interfaces still needs improvement such as easy-to-use guidelines, more friendly tones, and even a built-in understanding of common sign language. After the outbreak of COVID-19, service robots have become more popular. More than 90% of diners and waiters recognized the deployment of robots for non-contact food delivery in order to lower infection risks of the virus.

**TABLE 3.** Robot Food Service Failure.

**TABLE 4.** The Satisfaction Investigation of Staff and Customer.

VIII. CONCLUSIONS

In this research, practicability, and robustness of the service robot, deployed in fast-food restaurants, are the two major areas being focused on during the design and implementation phase of the research. To improve the application of the service robots, the estimation fusion to fuse the heterogeneous SLAM methodologies was expanded and has successfully combined the vision-based SLAM and 2D laser-based SLAM. The generation of combining 3D point cloud and 2D occupancy grid has increased the consistency, and it has the advantage of complementary abilities to enhance the robot’s self-localization in a complex environment. When either of the 2D laser-based localization temporarily fails or the vision SLAM obtains fewer features from the camera, the robot pose is able to maintain the accuracy with respect to the map. Besides, in terms of the navigation function of the service robot, the adaptive motion controller is refined with the dynamic window approach so that the service robot could navigate through narrow aisles smoothly on the scheduled meal delivery path.

The result of staff and diners’ feedbacks from this research has shown that people hold positive thoughts towards the deployment of service robots in restaurants. Furthermore, with the innovation and improvement done in this research, the possibility of increasing the number of service robots utilized in multiple service areas is high. Especially within COVID-19 pandemic periods, service robot deployment can help to maintain business while lowering infection risks. With more trial deployments of service robots in different service areas, this can be a potential solution for the global labor shortage and social distancing.

ACKNOWLEDGMENT

This study was sponsored by TECO Electric & Machinery Co., Ltd., and the field demonstration was executed in 15 MOS Burger Fast-food Restaurants in Taiwan.

REFERENCES

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Towards the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] T. Bailey, J. Nieto, J. Guivant, M. Stevens and E. Nebot, "Consistency of the EKF-SLAM Algorithm," *IEEE International Conference on Intelligent Robots and Systems*, Beijing, pp. 3562–3568, 2006.

[3] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99–110, June 2006.

[4] G. Grisetti, C. Stachniss and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb. 2007.

[5] S. Thrun, W. Burgard and D. Fox, *Probabilistic Robotics*, MIT Press, 2005.

[6] G. Grisetti, R. Küblerle, C. Stachniss, W. Burgard, "A tutorial on graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[7] ROS gmapping Package. [Online]. Available: http://wiki.ros.org/gmapping

[8] M. Filipenková and T. Afanasyev, “Comparison of Various SLAM Systems for Mobile Robot in an Indoor Environment,” *International Conference on Intelligent Systems*, pp. 400–407, 2018.

[9] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[10] B. M. da Silva, R. S. Xavier, T. P. do Nascimento and L. M. Gonçalves, “Experimental evaluation of ROS compatible SLAM algorithms for RGB-D sensors,” *Latin American Robotics Symposium (LARS), Curitiba, Brazil*, 2017.

[11] Z. Chen, W. Sheng, G. Yang, Z. Su and L. Baojuan, “Comparison and analysis of feature method and direct method in visual SLAM technology for social robots,” *World Congress on Intelligent Control and Automation*, Changsha, China, 2018.
[12] J. Lin, et al., "ORB-SLAM, IMU and Wheel Odometry Fusion for Indoor Mobile Robot Localization and Navigation," Academic Journal of Com- munication & Information Science, vol. 3(1), 2020.

[13] Y. Song, S. Nuske, S. Scherer, "A Multi-Sensor Fusion MAV State Es- timation from Long-Range Stereo, IMU, GPS and Barometric Sensors," Sensors, vol. 3(1), 2017.

[14] C. Debeunne and V. Damien, "A review of visual-LiDAR fusion based simultaneous localization and mapping," Sensors, vol. 20(7), 2020.

[15] F. Hu and G. Wu, "Distributed Error Correction of EKF Algorithm in Multi-Sensor Fusion Localization Model," IEEE Access, vol. 8, pp. 93211–93218, 2020.

[16] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lake meyer, D. Schulz, W. Steiner and S. Thrun, "The interactive museum tour-guide robot," AAAI, pp. 11–18, July 1998.

[17] H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, and R. Nakatsu, "Robovie: A robot generates episode chains in our daily life," Int. Sym- posium on Robotics, pp. 1356–1361, 2001.

[18] V. Kulyukin, C. Gharpure and J. Nicholson, "RoboCart: toward robot-assisted navigation of grocery stores by the visually impaired," IEEE International Conference on Intelligent Robots and Systems, Edmonton, Alta., pp. 2845–2850, 2005.

[19] Ying F. Chen, F. Wu, W. Shuai, "Robots serve humans in public places—KeJia robot as a shopping assistant," International Journal of Advanced Robotic Systems, vol. 14(3), May 2017.

[20] A. Eksiri and T. Kimura. "Restaurant service robots development in Thailand and their real environment evaluation," Journal of Robotics and Mechatronics, vol. 27(1), pp. 91–102, 2015.

[21] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, and A. Y. Ng, "ROS: an open-source Robot Operating System," ICRA workshop on open source software, vol. 3. no. 3.2, 2009.

[22] F. R. Q. Aini, A. N. Jati and U. Sunarya, "A study of Monte Carlo localiza- tion on robot operating system," International Conference on Information Technology Systems and Innovation, Bandung, pp. 1–6, 2016.

[23] H. P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 1985.

[24] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," Computer, vol. 22, no. 6, pp. 46–57, 1989.

[25] John. J. Leonard, Hung F. Durrant-Whyte and Ingemar J. Cox, "Dynamic map building for autonomous mobile robot," IEEE International Workshop on Intelligent Robots and Systems, Ibaraki, Japan, 1990.

[26] B. Abhayavos, R. Lavrenov, A. Zakiev, K. Yakovlev, M. Sv inin and E. Magid, "Automatic tool for Gazebo world construction: from a grayscale image to a 3D solid model," IEEE International Conference on Robotics and Automation, pp. 7226–7232, 2020.

[27] T. Moore and D. Stouch, "A generalized extended kalman filter implemen- tation for the robot operating system," Intelligent Autonomous Systems 13, pp. 335-348, Springer, Cham, 2016.

[28] M. Isard and A. Blake, "Condensation - conditional density propagation for visual tracking," International Journal of Computer Vision, vol. 29, no. 1, pp. 5–28, 1998.

[29] Zheng, Kaiyu. "Ros navigation tuning guide," Robot Operating System (ROS), pp.197-226, Springer, Cham, 2021.

[30] D. Fox, W. Burgard and S. Thrun, "The dynamic window approach to collision avoidance," IEEE Robotics & Automation Magazine, vol. 4, no. 1, pp. 23–33, 1997.

[31] Z. S. Hou and W. Huang, "The model-free learning adaptive control of a class of SISO nonlinear systems," American Control Conference, vol. 1 pp. 343–344, 1997.

[32] Chia J. Lin, Chun C. Lai, Kuo-H. Hsia and Chin S. Chen, "Apply Model- Free Adaptive Control Approach for Mobile Robot Path Following," Journal of Robotics, Networking and Artificial Life vol.7(3), pp.190–193, 2020.

[33] K. Berechka, O. Ciftci, and C. Cobanoglu. "Robots, artificial intelligence, and service automation in restaurants." Robots, artificial intelligence, and service automation in travel, tourism and hospitality, pp. 185–219, Emerald Publishing Limited, 2019.

[34] Dina Marie V. Zemke, et al., "How to build a better robot... for quick- service restaurants." Journal of Hospitality & Tourism Research, vol. 44, no. 8, pp. 1235–1269, 2020.