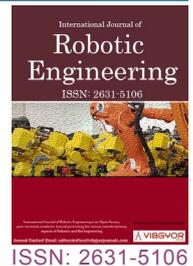


# Waiter Bots for Casual Restaurants



**Wan AYS<sup>1</sup>, Foo E<sup>2</sup>, Lai ZY<sup>2</sup>, Chen HH<sup>2</sup> and Lau WSM<sup>3\*</sup>**

<sup>1</sup>*Singapore Institute of Technology, Newcastle University, Singapore*

<sup>2</sup>*Nanyang Polytechnic, Ang Mo Kio, Singapore*

<sup>3</sup>*Newcastle Research & Innovation Institute (NewRIIS), Newcastle University, Singapore*

## Abstract

Waiter robots have been introduced in the food and beverage industry in the early 2000s in a number of countries, especially in Asia but with mixed receptions. Problem in designs appropriate for the environment, high costs and operation constraints are some reasons for the robots not being widely implemented on a larger scale. In this paper, it will discuss solutions that address these problems. To keep cost manageable, mechanical design is highly simplified, open source software algorithms for mobile robotics and integration with mobile Wi-Fi technologies currently in use in F&B settings are used. Wi-Fi apps for QR code allow for the tracking of customers' location and a secured local area Wi-Fi provides for the integration to business system. Using digital map and LiDAR and an array of short range IR sensors, the waiter robot is able to navigate autonomously from a serving point to a target point about 5 m away (repeated 10 times) to within 0.3 m (radius). It was next tested in an office environment and was ascertained to fulfil delivery tasks.

## Keywords

Waiter bots, Autonomous and systems integration, Mechatronics

## Introduction

Waiter robots in restaurants are both welcomed and rejected depending on what one read from press and business reviews. As early as 2006, restaurants in China started to replace waiters with robots [1] and India in 2017 [2] with some success and failures (e.g. in China [3]) where restaurants employing robots have closed and a number of robots unsold (Taobao [4]) online. Similar contradictory news is reported in

Singapore. Such reviews, not found in academic journals, are important because the success or failure on the use of bots in the service industry drives or retards future research and development. The main criticisms are cost and functionalities. In this paper, a redesigned concept is proposed to address the functionalities and the integration of waiter robotics in the food and beverage (F&B) industry. Certainly, waiter bots are not for all types of F&B: It will be appalling to have waiter bots in fine dining restaurants at this

**\*Corresponding author:** Lau WSM, Newcastle Research & Innovation Institute (NewRIIS), Newcastle University, 609607, Singapore

**Accepted:** October 21, 2019; **Published:** October 23, 2019

**Copyright:** © 2019 Wan AYS, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Wan et al. *Int J Robot Eng* 2019, 4:018

ISSN 2631-5106



9 772631 510005

junction. It has to be recognized that waiter bots are machines and should be deployed not to replace services provided by humans but to assist humans by relieving them of some manual tasks and allowing them to focus customers' relations services. The integration of robotics research and development with business operations and Wi-Fi technologies is an approach proposed here.

The paper begins with the examination of the efforts and experience of waiter bots in some F&B outlets in Singapore, the team's first waiter robot, the Beta-G, and the design and development of a 2<sup>nd</sup> generation lightweight general utility waiter bots for certain types of F&B outlets.

## System Concepts and Review of Current Deployment of Bots in F&B

### The current scene in Singapore

Robots are used in different roles in the service industry and the F&B industry. A seafood restaurant, in 2016, is believed to be the first in Singapore to use robot waiters (similar to the type shown in Figure 1) to serve food to customers as reported in AsiaOne [5]. At that time, each robot costs more than \$14,000 while waiters were hired for about \$30,000 per year. In food court setting robots shown in Figure 2 are used primarily for customers to deposit their trays after meals to allow faster turnaround and table availability. Drones had been tried for the delivery of food and beverages in one restaurant but were subsequently withdrawn. Bellhop and delivery robots are found in some hotels, for example, Techii Cobie robots and the Savioke R robot, Figure 3.



Figure 1: Waiter bot.

Except for those robots serving in some hotels, the service robots in restaurants move along wires or tracks placed on the ground. These robots are constrained in their motions and tracks can be dirtied and require maintenance. In some places, the tray collecting robots have been withdrawn, and there is no significant increase in the delivery of robots' deployment in restaurants. The Relay and Techii robots, on the other hand, navigate autonomously and use mobile Wi-Fi and RFID technology to take unmanned lift rides and makes phone calls to the guests' rooms upon arrival to notify them of their delivery. The number of such bellhop robots is likely to increase, but the number in standalone



Figure 2: Tray collecting bot.



Figure 3: Service robot.

F&B operations is not likely to grow due to their limited contribution, higher costs and the diminishing novelty. Unlike hotels, family causal type restaurants [6] have a smaller area of operations, lower operating budget and hence require a slightly different approach. They will need bots to assist humans by reducing their manual work, allowing them to focus their attention on customers.

### Improving the situation

To reduce cost and to incentivize the greater deployment of waiter bots, the mechanical design should be a bare-bone lightweight low cost structure principally to deliver food to the right customer and to aid in the collection of used crockeries. To be profitably employed, they will be integrated with a suite of sensors and open source software for autonomous operations using matured mobile robot technologies such as SLAM (simultaneous localization and mapping) [7,8] and LiDAR (Light detecting and ranging) sensor that provides data for mapping, localization, path planning and obstacle avoidance. In addition, they will be integrated into the operations of the restaurants using a variety of computer technology and Wi-Fi services found in almost all outlets to improve operations. This class of bot is called the Serveur bots.

To manage software development cost, open source middleware ROS (robot operating system) [9] is used to develop the waiter bots' autonomous systems using SLAM and communications capabilities.

### Mechanical Design of the Waiter Bots Design

#### Design perspective

In our first-generation waiter bot, Beta-G [10] shown in Figure 4, the objective was to design a "workhorse" to carry 3 levels of trays, using a dumb waiter concept to move the required trays up and down to the level where they can be delivered from either side of the robot. It is autonomous, omnidirectional with a rectangular base and 4-Swedish 45° and is able to move sideways or at any angle without turning. Such design is highly maneuverable in tight space. However, it is heavy, has a big footprint and is not suitable for many smaller food outlets. As it has a rectangular frame, it must be properly docked parallel to tables in order to deliver the trays. However, its mechatronics design provides a working platform for the Serveur bots.



Figure 4: Beta-G (not comparable scaling).

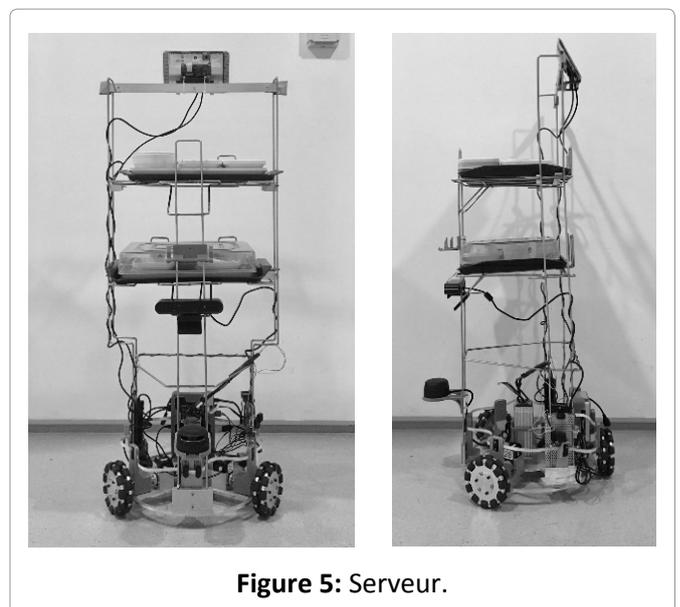
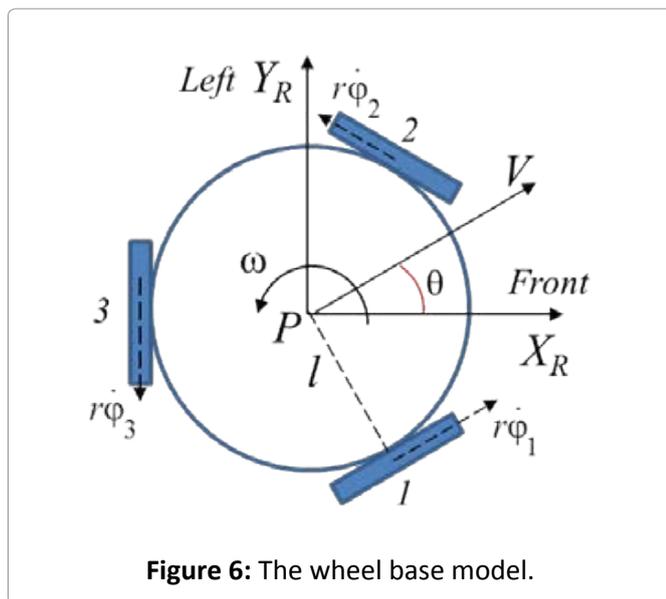


Figure 5: Serveur.

The time required and the accuracy to orientate the bot during docking can be reduced by building the Serveur bot, Figure 5, on a circular 3-Swedish 90°-wheel base and using tiered open tray design. Its orientation vis-à-vis the diner's table is no longer critical unlike the rectangular base, which requires the Beta-G to dock sideways to deliver the trays. The trays on the Serveur are easily removed or inserted from any direction and so are assessable to the customers regardless of the orientation of the bot. With the use of Swedish wheels, the number of required motors and motor drivers is reduced from three to four (in Beta-G) but the degree of maneuverability remains the same as the Beta-G and the footprint is smaller. This new design is many times lighter

**Table 1:** The server bot specifications.

Items	Specifications
Sensors	Array of Infra-red, TeraBee and a LiDAR
Power	24 V@5A
Battery	Lithium 24 V self-charging (duration min 40 mins)
Motor Com	TTL serial
Wireless	Wifi IEEE 802.11 n
Processor	Intel i7 NUC
Wheels	Omni-direction ( 3 wheels) - 3 Robotis MX106T
Speed	Max forward speed: 0.5 m/s
Payload	10 kg
Tray	3 Tier (maximum)

**Figure 6:** The wheel base model.

than the Beta-G using as we are able to make use of Dynamixel motors instead of brushed DC permanent magnet motors. With an open structure of welded 5 mm rod to support the three trays, the weight is reduced to 7.5 to 8 kg, allowing it to function for longer operation hours and is capable of carrying load up to 10 kg.

### Bots specifications

The design objective is allowed users to customize to whichever configuration they would like for the upper torso. Table 1 list the specifications of the robot.

### Mathematical Model of the Wheel Base

#### The 3-omni wheel waiter bot kinematic model

A mathematical model of the bot wheels, shown in Figure 6, is needed for motion control of direction and speed. It is used to generate the estimated odometry data. Each wheel radius is  $r$  and the radius of the robot base is  $l$ . The rotational speed of each wheel is  $\dot{\phi}_i$  and the linear speed is  $r_i \dot{\phi}_i$ .

Assuming negligible slip at all wheel-ground contact points, the inverse kinematic equation given the forward (front) speed  $v_x$ , sideward speed  $v_y$ , and body rotational speed,  $\omega$  is given by:

$$\begin{bmatrix} r\dot{\phi}_1 \\ r\dot{\phi}_2 \\ r\dot{\phi}_3 \end{bmatrix} = \begin{bmatrix} \cos \frac{\pi}{6} & \sin \frac{\pi}{6} & l \\ -\cos \frac{\pi}{6} & \sin \frac{\pi}{6} & l \\ 0 & -1 & l \end{bmatrix} \times \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \quad (1)$$

Since the matrix is non-singular, the forward kinematics can be obtained by inverting the matrix and the inertial velocities can be obtained by pre-multiplying with the inverse of the rotation matrix.

For control purpose, the forward velocity is specified as  $V$  and at angle  $\theta$  (with respect to the robot  $X_R$  axis) instead of  $v_x$ , and  $v_y$ . The robot rotation about an axis through  $P$ , the centre, is  $V_\theta = l\omega$  rad/s. The commands equations are:

$$\begin{aligned} r\dot{\phi}_1 &= V \cos \left( \theta - \frac{\pi}{6} \right) + V_\theta \\ r\dot{\phi}_2 &= -V \cos \left( \theta + \frac{\pi}{6} \right) + V_\theta \\ r\dot{\phi}_3 &= -V \cos \left( \theta - \frac{\pi}{2} \right) + V_\theta \end{aligned} \quad (2)$$

For turning on the spot, set  $V = 0$ , and for forward motion,  $V_\theta = 0$ , and  $\theta = 0$ . Other directions can be obtained using the desired  $\theta$  and with  $V_\theta = 0$ . Encoders on each wheel provide for the physical odometry data.

### System Architecture and Software Design

#### The software architecture

The software architecture is shown in Figure 7. The intelligence and interaction portion are based on Windows OS.Net framework while the navigation portion is based on ROS running in Linux. The two OS communicate with each other using the

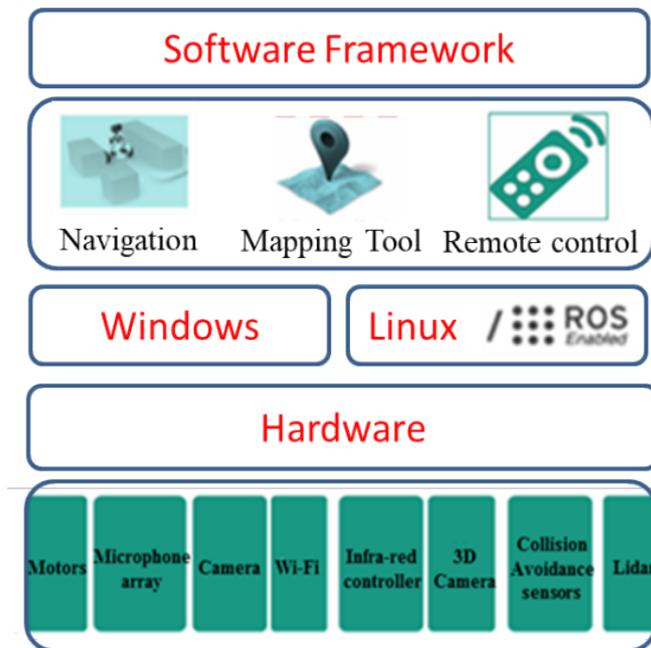


Figure 7: Waiter bot architecture.

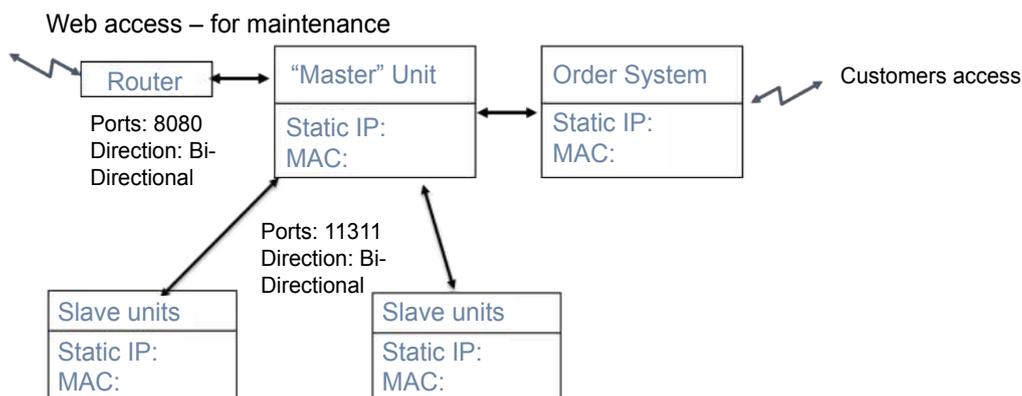


Figure 8: VLAN.

ROS bridge protocol sitting on top of the TCP/IP network framework.

### Integration with food ordering system and locating customers

In [10], Cheong, et al. used a coaster paging system to track customers' location: A coaster pager reads the RFID tag on the table which enables the bot to deliver the order to the correct table. However, someone has to issue the coaster pager when the order is taken, and the customer has to bring and place the pager to the table. To provide for a more complete digital solution, the waiter bot system can tap on the Wi-Fi capabilities provided in most, if not all, F&B outlets. A mobile phone QR app can be used instead to facilitate both ordering

and locating the customers. On each table, a QR code will be located. A snapshot of the QR code with a mobile phone will direct the customer's mobile app to the business URL where a menu will be displayed for both ordering and inputting the customer's table ID. For F&B outlets that use tablets for customers to place their orders, this problem is already solved. The table IDs can be made available to the bots.

### VLAN - virtual local area network - integration with business system

To integrate with customers placing orders function and locating customers' tables function, a VLAN is designed and implemented (Figure 8). It provides for some degree of security for the

business system. The waiter bots operate within this VLAN and receive data from the Order System, and are separate from the Order/Kitchen System and any other business software system. Data exchanges include table locations and feedbacks from the bots that they have completed their respective tasks. The VLAN also allows for access, using a router, to ROS to enable designers to rebuild, to upgrade bot's software or to introduce new applications when so required.

### Development in ROS (Robot Operating System) - software modules

**Digital mapping:** All the appropriate software for control and communication between the waiter bots and the master control can be found in the ROS environment. The wheel base mathematical model developed above is coded in base motion module, and navigation module requires a digital map of the area of operation for the bots to plan its path to the tables' location as indicated on the map and locate itself in the environment vis-à-vis the map. A digital map can be created using a variety of SLAM algorithms [7] or by digitalizing an architecture plan.

SLAM maps are typically obtained by driving a robot or by a robot moving autonomously around in the restaurant and as the bot moves, range data from a LiDAR (Light detection and ranging) sensor is collected and used by a SLAM algorithm to produce a map. This can be time consuming and inefficient. Instead, a mobile map maker housing a LiDAR and an onboard processor was used. It can be easily be moved around by an employee to areas where the robot will operate. The point clouds obtained are automatically uploaded to the onboard computer and a map of the area (example Figure 9) is created using HECTOR SLAM [11]. This SLAM algorithm does not need odometry data and has smaller error compared to Gmapping [12]. The map is used in ROS for navigation and localization. Example of a SLAM produced map of the test site 2 (discussed below) is shown in Figure 9.

**Navigation system using ROS:** With the digital map uploaded to the ROS, software development for the waiter bots to operate in the restaurant based on the digital map, and to communicate, via the ROS Bridge, with other systems using Windows OS can be done efficiently. Navigation and control

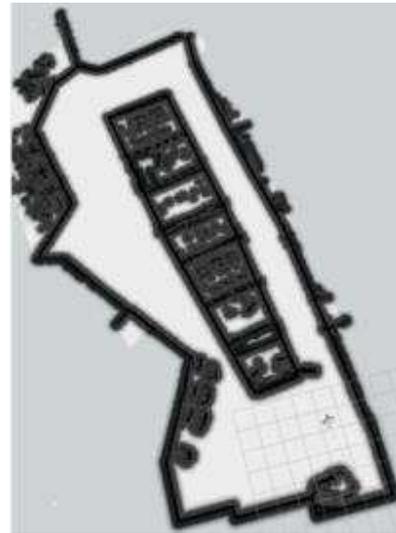


Figure 9: Hector SLAM map of test site 2.

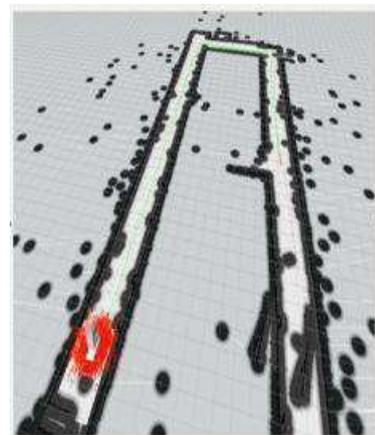


Figure 10: Cost map of test site 1.

of the bot wheel base are done using the Navigation module in ROS. Given the start and the endpoint, a path is automatically planned using the ROS provided path planning tool. As the bot moves along the planned path, it locates itself in the environment using data from its onboard LiDAR and the wheel odometry and the ROS provided AMCL (adaptive Monte Carlo localization) algorithm [13]. The AMCL uses a probabilistic localization technique and particle filters to track the pose of the robot as it moves along the path.

An important tool in ROS is the visualization tool, RVIZ. It provides two functions. The first function shows the robot's current pose as it moves along the planned path; the degree of uncertainty of its current position and orientation are represented by (red) particles. Figure 10 shows the bot on the map of the simulated restaurant when it is about

to move. Initially, the particles are widely dispersed because the bot is equally likely to be at any of the pose shown. When the bot moves, the AMCL algorithm will improve on the estimate of the state based on the data obtained by the sensors and the probabilistic outcomes of the particle filters. The closer the particles are, the more reliable is the estimate of the new state. The greater number of particles used improves the certainty. With a bigger environment, there is a need for a larger number of particles [14]. However, a larger number of particles require more CPU processing. The improvement in state estimation is shown by the particles converging towards the actual position of the robot.

The second function provides for the fine tuning of a bot operating parameters, for example, the constraint on the speeds and accelerations in the surge (X) and sway (Y) directions, and the safe distance to avoid collisions. Appropriate tuning of these values will get the bot to move in the desired manner and at the desired speed in an omnidirectional manner. A correct setting on the X and Y speeds will allow the bot to move to the target at the desired speed using an appropriate set of acceleration values. A differential drive wheel base will require a different tuning parameters to get the best result.

Using the RVIZ, it is possible to position the bot to less than 0.5 m on average, from their intended endpoint - see results presented later.

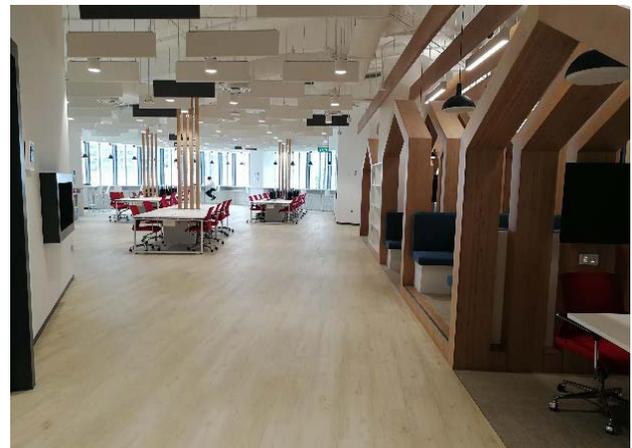
## Testing and Evaluation

Two levels of tests were conducted to evaluate the basic operation capabilities: Point A to Point B, about 5 m apart, test to determine the accuracy and repeatability, and the other to test the bot in an office space that mimics a food outlet.

In the point to point test, the target locations were published with x and y coordinates with respect to the map. Ten runs were made with two trays, containing water and some “food”, at different tiers, and the resulting values were tabulated. The radius of error is given by  $R = \sqrt{x^2 + y^2}$ . This uses the ROS navigation stack. The (DWA) local planner of the “move base” node will manage the velocity control (including the omnidirectional control), and global planner of the “move base” node will publish the (including the omnidirectional control), and global planner of the “move base” node

**Table 2:** Results of point to point tests [15].

S\N	Docked Pose wrt Goal Pose		Docking Radius (m)
	X (m)	Y (m)	
1	-0.125	0.025	0.127
2	0.22	0.035	0.223
3	0.63	0.115	0.640
4	0.2	0.005	0.2
5	0.655	0.06	0.658
6	-0.02	0.06	0.063
7	-0.25	0.113	0.274
8	0.165	0.05	0.172
9	-0.285	0.09	0.299
10	0.56	0.105	0.570
Average	0.175	0.066	0.323



**Figure 11:** The office workspace with cubicles.

will publish the goal point and the waypoints. The results are given in Table 2 [15].

The 2<sup>nd</sup> test was done on the site shown in figure 12.

A mapmaker using Hector SLAM was used to map the area of operations. It was manually pushed around the area where the robot is to operate. As the map maker does not have velocity data from a wheel, only the scan data is available. The map obtained in this way may not be accurate, but it is enough to use it as the initial map for testing. As discussed in the VLAN section above, a network is set up to allow the Hector SLAM maker to communicate with the embedded controller (Figure 11 and Figure 12).

The test showed that the bot is able to move from the pantry to a desired office cubicle. The ori-



**Figure 12:** Bot docking at target.

entation is not controlled. The speed of the robot is crucial. If the bot were to move off too quickly or come to a halt too quickly, water will be split due to sudden change of acceleration. The acceleration has to be fine-tuned to a suitable value, and the jerk has to be small. It is difficult to design a single speed profile for transporting liquid and non-liquid loads - with non-liquid, the bot should be travelling faster. Presently, the use of finite state machines [16] or SMACH in ROS to manage speed profiles along the trajectory is being explored.

The local planner provides for good obstacle avoidance as it can avoid all obstacles detected by the LiDAR. This local plan is also good for encountering dynamic obstacles (e.g. humans) and has good velocity control. However, with local planner, the robot can only perform small-ranges of omnidirectional motion as the LiDAR, by virtue of its current mounting location is not able to provide a full 360-degree scan [17]. In this case, the robot is in a differential drive state for most of the time. If the LiDAR scan is a full 360° view, the local planner is able to provide full omnidirectional movement. This will involve some redesign or relocation of the LiDAR mounting location.

## Conclusion

This article presents an improved design of a

waiter robot that is able to navigate in tight confine space which is premium in many food and beverages outlet in Singapore, and possibly in many countries too. Using off-the-shelf ROS navigation stacks and in-house designed nodes, a low-cost utility waiter bot is produced and presented. Unlike other commercially available robots in the service sector, the waiter bot is very basic and economical in design and importantly, functional. It can carry two tiers of food/drink trays (and one tier of electronics) and is able to navigate from point to point without the need to dock in a fixed orientation. The bot has been tested for accuracy and for functionality. Currently, the speed profiles are not ideal for conveying liquid. As such, the team is working on the use of ROS SMACH to manage the speed profiles. Preliminary tests had shown that tea and coffee can be conveyed without spilling.

## References

1. Clinton Nyugen (2016) Restaurants in China are replacing waiters with robots.
2. (2017) Restaurant in India replaces waiters with robots.
3. Cooper D (2016) Hiring a robot waiter can cost you your business. Engadget.
4. Custer C (2015) There are almost 170 robot waiters on Taobao, and not a single one has sold.
5. (2018) Singapore restaurant 'hires' robot waiters. ASIAONE.
6. (2019) Types of service and table settings in waiter and waitress training. Universal Class.
7. Durrant-Whyte H, Bailey T (2006) SLAM Part 1: The essential algorithms.
8. Cadena C, Carlone L, Carrillo H, Latif Y, Sacramuzza D, et al. (2016) Past, Present and Future of SLAM: Toward Robust-Perception Age. IEEE Transactions on Robotics 32: 1309-1332.
9. <http://wiki.ros.org/navigation>
10. Cheong A, Lau MWS, Foo E, Ju WB, Hedley J (2016) Development of a robotic waiter system. IFAC Paper-sOnLine 49: 681-686.
11. Kohlbrecher S, Meyer J, Graber T, Petersen K, von Stryk O (2013) Hector open source modules for autonomous mapping and navigation with rescue robots. Proceedings of 17th RoboCup international symposium.
12. Santos J, Portugal D, Rocha R (2013) An evaluation of 2D SLAM techniques available in robot operating system.

13. Fox D, Burgard W, Thrun S (1998) Active markov localization for mobile robots. *Robotics and Autonomous Systems* 25: 195-207.
14. Zhang L, Zapata R, Lepinay P (2009) Self-adaptive monte carlo localization for mobile robots using range sensors. *IEEE/RSJ International Conference on Intelligent Robots and Systems*.
15. Wan Ash YS (2019) Velocity profiling design and testing. Technical Report Sep 2019 No 1.
16. Ben-Ari M, Mondala F (2017) *Elements of robotics*.
17. Lai ZY (2019) NUIS internship technical report.

ISSN 2631-5106



9 772631 510005