Omni-directional Vehicle Platform Development for Visual SLAM Construction

P. I. Chang and Y. S. Shi

***Abstract*—This research demonstrated a visual SLAM (Simultaneous Localization And Mapping) construction with only encoder localization for an omni-directional vehicle platform, utilizing a 3D commercial Kinect device. The reconstructed SLAM is stitched from the local particle cloud by video tapping the surrounding area without a-priori knowledge, relying on only encoder position calculated with an on-board embedded system, and comparing the relative distance between object and camera. This vehicle is designed with PID feedback control to enhance the positional encoding capability, thus by minimizing the wheel's tracking error, this pilot study demonstrated that with full access to all the system data through ROS operation system, visual SLAM can be re-constructed with minimum position data, without need of GPS or gyroscope in the vehicle. This in-house omni-directional vehicle also includes Wifi remote control and data collection through Wifi in real-time, and reconstructs visual SLAM off-line.**

***Index Terms*—omni-directional vehicle, visual SLAM, Wifi remote sensing.**

I. Introduction

Omni-Directional Vehicle (ODV) varies in its design, from using different omni-directional wheels, such as Mecanum wheels (45^o subwheel/roller), or all-directional wheels (90^o subwheel/roller), or any other wheels that can direct itself in manuveuring the vehicle in all directions, for all x-y translation and θ rotation. This research developed an 90^o subwheel setup, with four individual driven wheels, to operate such a ODV platform, as shown in Fig. (2).

Other researchers have developed platforms with the above mentioned Mecanum wheeled mobile platform, and using similar controller embedded system in our case: with STM32, Jetson TX2, etc. Which in this case recorded the trajectory with help of IMU and UWB localization, and resulted in position error at around 5cm[1]. Recently researchers looked into adaptive sliding Mode for such four Mecanum wheel platform, comparing these advanced controllers to traditional SMC and PID in simulations[2].

As mentioned before, some researchers suspend their Mecanum wheel to make different prototypes and verified its steering with simple experiment[3]. While use of computer vision-based movement are also of interest recently, adding visual systems to the platform and researchers investigated in how to control a Mecanum ODV platform with machine vision. When considering its applied scenario, the mobile platform needs to move a target with known patterns. It is found that machine vision helps to find the relative position between robot and target, therefore only a simple control logic needed with finite predefined states[4].

In order to fully find the controlling aspect of a Mecanum ODV, especially its ability to maneuver in congested spaces, the kinematics of different wheel design, and considerations for wheel loading and traction is needed for full understanding using a state-space controller model. And researchers have found that with full state control, it is shown this omnidirectional capability greatly reduces the amount of area and time required for maneuvers, and the omni-directional wheel in particular reduces time because of the absence of singularities [5].

And since the kinematics and dynamics for such ODV vehicle motion do not require excessive computation, even in the case where they include compensation for wheel slip detection and correction, ODV is ideal for self-guided visual SLAM exploration and autonomous driving/manuveuring.

More recently, researchers proposed a stereo visual-inertial SLAM to improve performance of conventional visual SLAM in low texture or repeating texture. Since visual-inertial SLAM is introduced with growing computing ability, researchers investigated in the integration of data for large complexities[6].

Due to the autonomous nature of ODV application, there is need to explore the ODV models' friction and switching of rollers effect on the efficiency of the system[7], therefore it is needed to detect the health of the ODV, especially the state of each individual wheel, and actively monitor the performance of the ODV itself.

However, there is lack of 90^o omni-directional wheel platform experimental data, where most of the above research groups are focused on development and study of the more popular Mecanum, thus this team developed an in-house four wheel omni-directional wheel platform with visual SLAM capability, and demonstrated that with only encoder positioning data, it is still applicable in visual SLAM construction.

The paper is outlined as followed: in Sec. (II), the details of the components and driving dynamics and kinematics are developed, to fully demonstrate the control and remote design of the ODV. And in Sec. (III), the SLAM positioning and environmental exploration is demonstrated, and finally in Sec. (IV) conclusions are made for further studies.

II. Omni-Directional Vehicle System

An in-house 4-wheel omni-directional vehicle was designed and built for the purpose of this 3D camera SLAM experiment

This paper is submitted on March 5th, 2021.

Peter I. Chang is with the Mechanical Engineering Dept., National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan (+886-2-2737-6482; e-mail: itchang@mail.ntust.edu.tw).

Yun-Shou Shi was an undergraduate student at ME Dept., NTUST, and currently a master student at Mechanical Engineering, National Taiwan University.

[8].

The basic components are used as followed: four omni-directional wheel (all-direction), attached to four independently controlled servo-motor, individually driven by its motor controller board, which is governed by an embedded system development board, giving speed command to the four individual wheels. These components are grouped into four levels of control, as shown in Fig. 1.

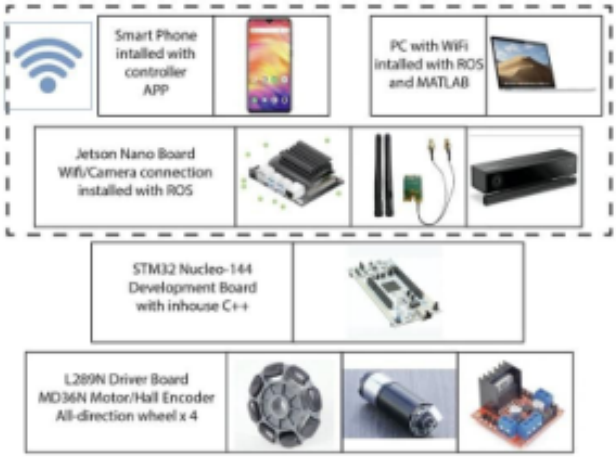

Figure 1. Omni-vehicle system architecture. The whole system has three structure: top/mid/bottom. Where the top consists of mapping and remote-control via WiFi. The mid structure joins the top/bottom for the vehicle's movement and location recording.

The omni-directional vehicle is designed with three major sub-systems [9], where the top part is the communicational subsystem, composed of local WiFi connection, smart phone acting as a remote control, and a PC for data collection. The Jetson Nano attached with a 3D camera (Microsoft Kinect)[10], [11] and WiFi, serve as the sensor/receiver subsystem, receiving high level left/right, forward/backward, as well as the vehicle clockwise/counter-clockwise turn command. A STM32 controller board is the middle part subsystem, where the high-level commands are translated into motor control command, according to the omni-vehicle dynamics [12], [13]. This embedded system provides direct motor speed command to the lower level sub-system, which compose of the four individually feedback controlled motor attached to all-directional wheels. The full detail of the components is outlined in the following Table I.

Given a four all-directional wheel configuration as Fig. 2, it is clear that the driving dynamics and kinematics are derived as follows:

Where, x, y and, θ are the vehicle's position (x,y), and its angle defined as the "front" facing direction Length d is the distance between the wheels and the geometric center of the vehicle. Velocity v_i with $i \in \{1, 2, 3, 4\}$, and angular velocity ω_i also with $i \in \{1, 2, 3, 4\}$ are the four independent wheel's controlled speed.

The kinematics of the vehicle can be defined as the four individual traction torque T_i, where $i \in 0, 1, 2, 3$ applied to the different wheels, but resulting with a vehicle linear velocity v and v_n, along with a vehicle linear rotational angular velocity ω.

Table I
4-WHEEL OMNI-DIRECTIONAL VEHICLE PLATFORM COMPONENTS

Components	System Setup		
	System Level	**Name**	**Note**
All-directional wheel	4X Basic component	N/A	90 degree orthogonal two-axis wheel
Servo motor with encoder and driver	Low level feedback actuation	MD36	Stepper motor with speed encoding
32 bit embedded system	Mid level signal I/O system	STM 32 Nucleo-144 Dev-Board	In-house C/C++ code
32 bit ROS embedded system	High level I/O signal for vehicle command	Jetson Nano	ROS installed
3D camera	High level I/O signal for srounding survey	Microsoft Kinect 2	
WiFi	High level I/O signal for data communication	N/A	

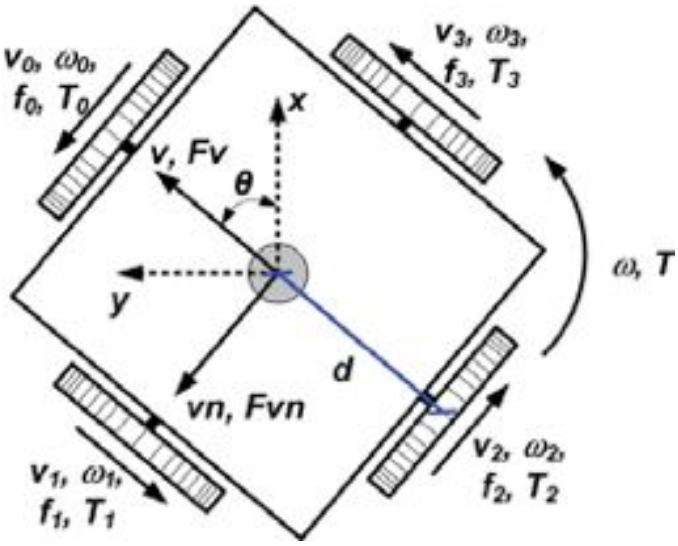

Figure 2. A four all-directional wheel Omni-vehicle system dynamics orientation setup [14].

The dynamics is best described with this resulting force along the v and v_n direction (usually aligned with x and y direction), shown here as F_v and F_{vn}. And the combined vehicle torque T that is the summation of the four individual: $T = \sum_{i=0}^{3} T_i$.

A. $x - y$ Kinematics Description

Since for an omni-directional vehicle/robot $v_x(t) = \frac{dx(t)}{dt}$, $v_y(t) = \frac{dy(t)}{dt}$ and $\omega(t) = \frac{d\theta(t)}{dt}$. It is clear that one can transform the absolute $x-y$ coordinate into the vehicle's front/side coordinate as the following equation. Where one defines the two different state variables: $X_R = [v(t)\ v_n(t)\ \omega(t)]^T, X_0 =$

$[v_x(t)\ v_y(t)\ \omega(t)]^T$ provided,

$$X_R = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} X_0. \tag{1}$$

It is clear that for this 4-wheeled vehicle, the kinematics that relates the center robot movement can be derived to the individual wheels as Eq. (2).

$$\begin{bmatrix} v_0(t) \\ v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & d \\ -1 & 0 & d \\ 0 & -1 & d \\ 1 & 0 & d \end{bmatrix} \begin{bmatrix} v(t) \\ v_n(t) \\ \omega(t) \end{bmatrix} \tag{2}$$

Where Eq. 2 is an overdetermined system, given a redundant wheel (three is enough for driving such system). However, it can be found that it follows the equation set.

$$v(t) = 1/2\,(v_3(t) - v_1(t)) \tag{3a}$$
$$v_n(t) = 1/2\,(v_0(t) - v_2(t)) \tag{3b}$$
$$\omega(t) = 1/(4d)\,(v_0(t) + v_1(t) + v_2(t) + v_3(t)) \tag{3c}$$

Thus for any given time, it can be seen that if one controls the velocity of the wheel v_i, the full front facing forward/backward velocity v, sideways left/right velocity v_n, and the vehicle rotational angular velocity making clockwise/counter-clockwise turn ω, are found at any instance.

B. Lagrangian Description

It is useful to derive our omni-directional vehicle using the Lagrangian dynamic description, when considering our omni-wheel, its dynamic not considering height potential if the vehicle is driven on a flat surface, the Lagrange equation is:

$$\frac{d}{dt}(\frac{\partial K}{\partial \omega_i}) - \frac{\partial K}{\partial \theta_i} + \frac{\partial D_i}{\partial \omega_i} = T. \tag{4}$$

Where K is the vehicle's kinetic energy, associated to its translational and rotational movement, as well as all four wheel's rotation θ_i and angular velocity ω_i. D_i is each wheel's nominal friction, combining all the ground, small sub-wheel axial, and motor axial friction. The kinetic engergy can be found as:

$$K = \frac{1}{2}m(v^2 + v_n^2) + \frac{1}{2}J\omega^2 + \frac{1}{2}J_w(\omega_0^2 + \omega_1^2 + \omega_2^2 + \omega_3^2). \tag{5}$$

Where m and J are the vehicle's mass and moment of inertia, respectively. J_w is a single wheel's moment of inertia. It is clear that the individual wheel's velocity $v_i = r_w\omega_i$ is proportional to it's wheel radius r_w, replacing all v, v_n, and ω, with Eq. (3) and replacing all the v_i with $r_w\omega_i$, it can be shown that:

$$T = \mathbf{M}\dot{\boldsymbol{\omega}} + D_\theta\boldsymbol{\omega}. \tag{6}$$

Where $\boldsymbol{\omega} = [\omega_0\ \omega_1\ \omega_2\ \omega_3]^T$, and $D_\theta = diag\,[d_0\ d_1\ d_2\ d_3]$ is the diagonal of the nominal friction. And M can be found re-arranging the vehicle's mass m, moment of inertia J, and each omni-directional wheel's moment of inertia J_W, along with the vehicle's half width d, and the wheel's radius r_w.

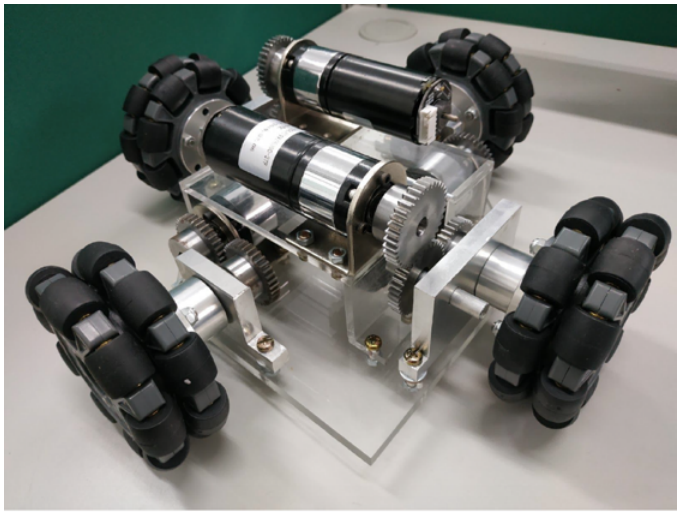

Figure 3. 4-wheel all-directional vehicle system chassis.

$$M = \begin{bmatrix} a+b+J_W & b & -a & b \\ b & a+b+J_W & b & -a \\ -a & b & a+b+J_W & b \\ b & -a & b & a+b+J_W \end{bmatrix},$$

where $a = \frac{m}{4}r_w^2$, $b = \frac{J}{16d^2}r_w^2$.

It is trivial to re-arrange Eq. (6) into a full state control form:

$$\dot{\omega} = -\mathbf{M}^{-1}D_{theta}\omega + \mathbf{M}^{-1}T,$$

therefore if we define $\mathbb{A} = -\mathbf{M}^{-1}D_{theta}$, $\mathbb{B} = \mathbf{M}^{-1}$, and $\mathbb{C} = \mathbb{I}$ a 4 by 4 identity matrix, the full dynamics equation is now in its standard form:

$$\dot{\omega} = \mathbb{A}\omega + \mathbb{B}T$$
$$\mathbf{y} = \mathbb{C}\omega$$

C. Localization Tracking

In our system design, since the individual wheels are feedback controlled, it is desired that the individual v_i are faithfully implemented by the given command, thus there is no need to add real time self-adjustment, but only high-level SLAM correction after a long period of operation.

The in-house vehicle was designed to center the movement at the chassis base. It can be clearly shown at Fig. 3.

With full feedback control, and assuming a non-slip condition for the vehicle's operation, the controller command kinematic equation can be state as followed in the x-y coordinate:

$$r_w \begin{bmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} \sin\theta & -\cos\theta & -d \\ \sin(\theta+\pi/2) & -\cos(\theta+\pi/2) & -d \\ \sin(\theta+\pi) & -\cos(\theta+\pi) & -d \\ \sin(\theta+3\pi/2) & -\cos(\theta+3\pi/2) & -d \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \tag{7}$$

Where r_w is the radius of an individual wheel, and thus given the individual ω_i command, the robot vehicle will take the according $\dot{x}$, $\dot{y}$, and $\dot{\theta}$ movement.

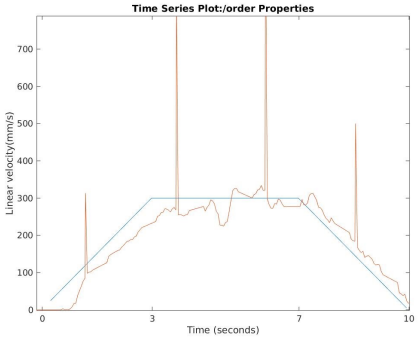

Figure 4. x-direction linear velocity (mm/s) measurements.

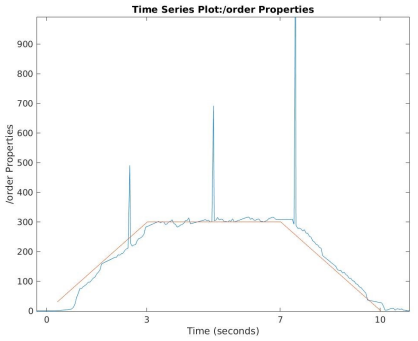

Figure 5. y-direction linear (mm/s) measurements.

III. SLAM Measurement and Results

From the above equation, it is clear that the overall x and y position, as well as the rotation angle θ, can be found by integrating the above equation.

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ \theta_0 \end{bmatrix} + r_w \int_0^t \begin{bmatrix} \sin\theta & -\cos\theta & -d \\ \sin(\theta+\pi/2) & -\cos(\theta+\pi/2) & -d \\ \sin(\theta+\pi) & -\cos(\theta+\pi) & -d \\ \sin(\theta+3\pi/2) & -\cos(\theta+3\pi/2) & -d \end{bmatrix}^{-1} \begin{bmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} d\tau \tag{8}$$

While one monitors all the instantaneous ω_i and can then find the estimated location of the vehicle.

As mentioned, the typical way to trace out the current position and angular facing angle (x, y, θ), is to integrate the above equation. Fig. 4 and 5 show such a linear velocity profile, tracked during a 10 second period.

It is clear due to the hardware limitation, as well as the numerical limitation, it is clear that there are jumping spikes that occur during the x velocity estimation. This also occurred for the y linear velocity case.

The robot vehicle is remote controlled to navigate throughout the environment, and taking video images along the exploration.

Figure 6. Environment testing for the all directional vehicle, where a door way is located at the corner of room, along with a table to the right.

Figure 7. SLAM reconstruction of the environment, showing the door corner as well as the table fixture at the right-hand side. A small bob can be seen at near the left wall, indicating the door stop.

The SLAM algorithm automatically stiches the continuous image from the video taken, and maps out using the 3D camera's distancing capability [14], [15].

However, since the distance is estimated from the video image, there are slight differences along a straight wall when considering the object detection. This is not surprising considering the stitching of the video needs many reliable localization methods.

In our case of local SLAM, the robot is slowly maneuvered and rotated along the environment, such that the different angles are spotted and identified, with use of only the 3D camera.

Fig. 8 shows the image stitching, as the video is sent to the

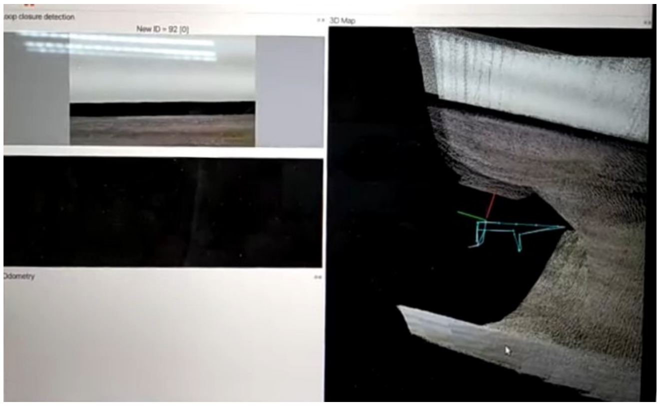

Figure 8. SLAM reconstruction from the video image stitching, showing local co-ordinate tracing at the bottom of the floor, and the different angular images stitched with the black line matching at different frames.

remote PC for SLAM reconstruction.

It is clear from the stitching process, the local velocity is traced up to its standard, and the stitching process can almost exactly find the distance between the object and the vehicle, with satisfactory results.

IV. Conclusion and Future Work

The result of using only 3D camera for SLAM reconstruction shows promising results, given that even without other localization sensor, such as GPS or Gyro for positioning [16], the relative distance from object and local positioning, is enough for SLAM reconstruction. Given the vehicle is maneuvered slowly.

It is needed to test further about the speed limit, as well as the video stitching capability, to understand if the process can be done in real-time, as now the bottleneck for this algorithm is the video transmission time. The image stitching is now found due to the slow movement speed of the vehicle, but limitations to this should be quantified as well.

Other sensors should be added for comparison purposes, such that the error of distance estimation and SLAM miss judgement can be quantified.

Acknowledgement

The authors would like to thank the NTUST and National Taiwan Normal University joint undergraduate research project competition committee, as well as the National Taiwan University System for providing this project fund.

References

[1] Z.Yuan, Y. Tian, Y. Yin, S. Wang, J. Liu, and L. Wu, “Trajectory tracking control of a four mecanum wheeled mobile platform: An extended state observer-based sliding mode approach,” *IET Control Theory & Applications*, vol. 14, no. 3, pp. 415–426, 2021/03/10 2020. DOI: https://doi.org/10.1049/iet-cta.2018.6127. [Online]. Available: https://doi.org/10.1049/iet- cta. 2018.6127.

[2] X.Lu, X. Zhang, G. Zhang, and S. Jia, “Design of adaptive sliding mode controller for four-mecanum wheel mobile robot,” in *2018 37th Chinese Control Conference (CCC)*, 2018, pp. 3983–3987, ISBN: 1934-1768. DOI: 10.23919/ChiCC.2018.8483388.

[3] J. S. Keek, S. L. Loh, and S. H. Chong, “Comprehensive development and control of a path-trackable mecanumwheeled robot,” *IEEE Access*, vol. 7, pp. 18 368–18 381, 2019. DOI: 10.1109/ACCESS. 2019.2897013.

[4] M.Y. Naing, A. S. Oo, I. Nilkhamhang, and T. Than, “Development of computer vision-based movement controlling in mecanum wheel robotic car,” in *2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICASYMP)*, 2019, pp. 45–48. DOI: 10.1109/ICA- SYMP. 2019.8646254.

[5] S. L. Dickerson and B. D. Lapin, “Control of an omnidirectional robotic vehicle with mecanum wheels,” in *NTC ’91 - National Telesystems Conference Proceedings*, 1991, pp. 323–328. DOI: 10 . 1109 / NTC . 1991 . 148039.

[6] C.Chen, H. Zhu, L. Wang, and Y. Liu, “A stereo visual-inertial slam approach for indoor mobile robots in unknown environments without occlusions,” *IEEE Access*, vol. 7, pp. 185 408–185 421, 2019. DOI: 10 . 1109/ ACCESS.2019.2961266.

[7] B. I. Adamov and G. R. Saypulaev, “A study of the dynamics of an omnidirectional platform, taking into account the design of mecanum wheels and multicomponent contact friction,” in *2020 International Conference Nonlinearity, Information and Robotics (NIR)*, 2020, pp. 1–6. DOI: 10.1109/ NIR50484.2020.9290193.

[8] ... M. C¸ elik, and ¨O. Pektas¸, “Design of an autonomous mobile robot based on ROS,” in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, 2017, pp. 1–5. DOI: 10.1109/ IDAP. 2017.8090199.

[9] Z. Xuexi, L. Guokun, F. Genping, X. Dongliang, and L. Shiliu, “Slam algorithm analysis of mobile robot based on lidar,” in *2019 Chinese Control Conference (CCC)*, 2019, pp. 4739–4745. DOI: 10 . 23919 / ChiCC . 2019 . 8866200.

[10] D. S. O. Correa, D. F. Sciotti, M. G. Prado, D. O. Sales, D. F. Wolf, and F. S. Osorio, “Mobile robots navigation in indoor environments using kinect sensor,” in *2012 Second Brazilian Conference on Critical Embedded Systems*, 2012, pp. 36–41. DOI: 10.1109/CBSEC.2012. 18.

[11] Y. Deng, Y. Shan, Z. Gong, and L. Chen, “Large-scale navigation method for autonomous mobile robot based on fusion of gps and lidar slam,” in *2018 Chinese Automation Congress (CAC)*, 2018, pp. 3145–3148. DOI: 10.1109/CAC.2018.8623646.

[12] P. Beˇno, F. Duchoˇn, M. T¨olgyessy, P. Hubinsk´y, and M. Kajan, “3d map reconstruction with sensor kinect: Searching for solution applicable to small mobile robots,”

Robotics in Alpe-Adria-Danube Region (RAAD), 2014, pp. 1–6. DOI: 10.1109/RAAD.2014.7002252.

[13] S.Lee, J. Jung, S. Kim, I. Kim, and H. Myung, "Dv-slam (dual-sensor-based vector-field slam) and observability analysis," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 2, pp. 1101–1112, 2015. DOI: 10.1109/TIE.2014.2341595.

[14] J.Li, M. Lee, S. Park, and J. Kim, "Range sonar array based slam for p-suro auv in a partially known environment," in *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2012, pp. 353–354. DOI: 10.1109/URAI.2012.6463014.

[15] D.Y. Kim, J. Kim, I. Kim, and S. Jun, "Artificial landmark for vision-based slam of water pipe rehabilitation robot," in *2015 12th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2015, pp. 444–446. DOI: 10.1109/URAI.2015.7358900.

[16] HyunChul Roh, Chang Hun Sung, Min Tae Kang, and Myung Jin Chung, "Fast slam using polar scan matching and particle weight based occupancy grid map for mobile robot," in *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2011, pp. 756–757. DOI: 10.1109/URAI.2011.6146004.

Peter I-Tsyuen Chang received his B.S. and M.S. from National Taiwan University, and his Ph.D from Boston University, and is currently an Assistant Professor at the Mechanical Engineering Department, National Taiwan University of Science and Technology. Before teaching and researching at NTUST, Peter spent time at Automation and Control Institute, Vienna University of Technology and Center of Applied Science, Academia Sinica. Peter is involved in the field of robotics, mechatronics, as well as smart sensing and control.

Yun-Shou Shi graduated from National Taiwan University of Science and Technology with a B.S. in Mechanical Engineering in 2020, and is currently pursuing his M.S. at the Department of Mechanical Engineering, National Taiwan University.