



# Construction of an Environmental Map including Road Surface Classification Based on a Coaxial Two-Wheeled Robot

Minoru Sasaki<sup>1\*</sup>, Eita Kunii<sup>1</sup>, Tatsuya Uda<sup>1</sup>, Kojiro Matsushita<sup>1</sup>, Joseph K. Muguro<sup>1,2</sup>, Muhammad Syaiful Amri bin Suhaimi<sup>1</sup> and Waweru Njeri<sup>1,2</sup>

<sup>1</sup>*Gifu University (Department of Mechanical Engineering, Gifu University, 1-1 Yanagido, Gifu City, 501-1193)*

<sup>2</sup>*Dedan Kimathi University of Technology (Electrical and Electronics Department, Dedan Kimathi University of Technology, PRIVATE BAG, Nyeri, Kenya)*

*\*Corresponding Author - E-mail: sasaki@gifu-u.ac.jp*

**Abstract** This study details the construction of an environmental map as well as road surface identification based on a coaxial two-wheel robot. Two-wheeled robots are faced with the challenge of oscillatory motion making it hard to get accurate sensor data which is needed in mapping and localization. The proposed system utilizes Robot Operating System (ROS) for data fusion and a modified Simultaneous Localization and Mapping (SLAM) algorithm to reduce oscillatory motions as well as generate environmental map in real-time. Deep learning was used to perform image segmentation to classify road surface. From the results, posture control was verified as well as generation of indoor/outdoor environment maps from sensor data and image processing. The study reported a reduction of oscillatory motion from 40 to 10 degrees. Image segmentation reported a prediction confidence of 80% or more which was adequate for map generation.

**Keywords** Coaxial Two-Wheeled Robots, Environment Maps, SLAM, Deep Learning.

## 1. Introduction

In recent past, robots have been integrated in service industries such as cleaning, security, and others. The design of mobile robots follows the application areas which is informed by the dynamic environmental constraints like the walkable area, power requirements, size and others.

Traditionally, robots are human operated using inputs as is the case with motor vehicles. With the diversification of usage, human operation is being replaced with semi (preprogrammed actions) or full autonomy. Programmed usage require perfect knowledge of the working space with slight or no change in the environment or robot dynamics. This is challenging if any of the integral system is updated. The most feasible solution to the dynamics of

environment is to construct an autonomous robot: a robot that gets sensor data and decides on a probable action without referring to a programmed input [1]. This is achieved by integrating sensor data to an artificial intelligence model e.g. in object detection or classification. This way robot deployment can be generalized to any dynamic environment.

Mobility in robots is an important design consideration with either wheels or legs. The advantages of using driven wheels over legs are described in literature [2]. In wheeled robots, the number of wheels impacts motion stability, form-factor, energy efficiency, degrees of freedom and other parameters. In literature, varying number of wheel designs have been discussed ranging from one-wheel (Ball-shaped robots) [3], two-wheeled

robots [1], four-wheeled robots, six-wheeled robots [4].

The advantage with three or more-wheel robot is the static stability even in high speeds. The challenge with this configuration is the over-constrained system, requiring suspensions particularly in non-flat terrain. Analogous to a motor vehicle (4-wheeled robot), static stability leads to constrained maneuverability (forward/backward, turning motion) and form factor. The robot design with three or more wheels therefore have limited application in constrained spaces like indoor environment and uneven ground plane.

This paper considers the use of two-wheeled robot as applied in constrained spaces and indoor environment. The design of two-wheeled robot typically involves mounting two coaxial wheels on either side of the robot body. For this reason, the robot must actively stabilize themselves to avoid tipping.

Two-wheeled robot overcomes the problems of a typical four-wheeled robot by the ability to turn on-the-spot. This enables them to maneuver through corridors and tight corners with ease. Additionally, depending on the design, the robots can swing forward and backward to establish stability in slopes and inclined areas [2].

The swinging motion of the two-wheeled robot is advantageous in that it enables the robot to move in uneven ground planes. This however poses a challenge in stable sensor readings as the body moves back and forth.

To overcome the highlighted challenges of a two-wheeled robot, this paper utilized a posture controller for horizontal swings and a localization and mapping algorithm for robot position estimation. Further, deep learning algorithm is applied to detect and classify road surface.

A key consideration for any mobile robot is path planning and navigation [5]. In this case, path planning has to do with the initial and target positions as well as the route taken. Navigation in this context will be identification of objects and obstacles and recomputing the best route to avoid them. A combination of systems is necessary to inform the robot of its position relative to an obstacle.

Researchers approach path planning and navigation in various ways. A paper [2] identifies two robot navigation categories as either global or local navigation. Global navigation is viewed as position of all the present elements in the environment with respect to reference axis. In this category, prior information is required. In local navigation, the robot identifies the dynamic and positional relationship among the present elements. In other words, no prior information is required. Full review of two-wheel robot navigation is available in the referred

review paper [6]. This paper utilized Simultaneous Localization and Mapping (SLAM) algorithm as provided in Robot Operation System. Further details of the algorithm are described in section three.

## 2. Proposed system

As shown in Fig. 1, the proposed system consists of two-wheel servo motors, two horizontal angle control servo motors, notebook PC, inertial measurement unit (IMU) sensors, and image acquisition system (USB Camera). Table 1 shows the integration of all the sensing and controller components used in the proposed system. From the table, four tasks listed as posture control, localization and Mapping, surface identification and classification are mapped to the analysis tool employed and the output dependence. Posture control takes in IMU data (gyroscope and acceleration data) to control the horizontal position servo motors. Road surface identification task utilizes the same sensor data as posture control.

Localized mapping is performed using ROS software [7] from area sensor data (Laser rangefinder). Processed information informs the robot of its position to perform navigation. ROS software controls the movement (wheel servo motors) of the robot after integrating all the sensor and detection units. This is explained further in the next section. Fig. 2 shows Laser rangefinder (UTM-30LX) from Hokuyo® and its specifications.

In road surface identification, the surface angle of the road is determined using Madgwick filter attitude values. The filtered output is fed into the ROS for further analysis. Road type is classified based on trained Deep Learning algorithm [8]. The output of the surface classification is fed to ROS for accurate construction of the environmental map.

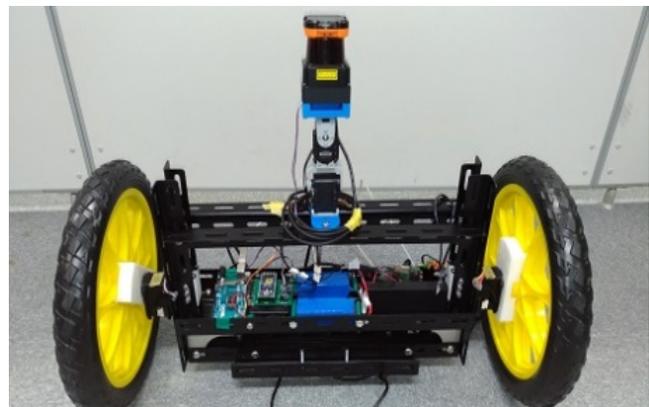


Fig. 1. A Two-wheeled robot



Table 1: Proposed architectural dependence

Task	Sensors	Analysis	Output
Horizontal Posture control	IMU Sensor (MPU-9250)	Microchip computer (Madgwick Filter)	Horizontal Servo Motors
Localization and Mapping	Area Sensor (Laser Rangefinder)	ROS SLAM Algorithm	Wheel Servo Motor
Road surface discrimination	IMU Sensor (MPU-9250)	Microchip computer (Madgwick Filter)	ROS Software
Road surface classification	USB Camera	Matlab Deep Learning	ROS Software



Voltage	DC12V
Range	0.1~30[m]
	270[deg]
Precision	0.1~10[m] : ±30[mm]
	10~30[m] : ±50[mm]
Resolution	0.25[deg]

Fig. 2. Area sensor (Laser rangefinder) with specifications

### 3. Environmental Map Construction by ROS

#### 3.1. Self-position estimation using SLAM and RBPF

For the robot to perform the mapping, it is necessary to perform self-location estimation. This way, the robot is aware of its own position relative to walls and other objects and performs map construction from sensor information at the same time. This approach is referred to as SLAM (Simultaneous Localization and Mapping), it is a subject that is actively studied to this day [9] [10]. In general, encoder and GPS are used for self-position estimation, and range sensors, cameras, etc. are used for wall surface estimation. Mathematical approaches are also applied separately or in conjunction with sensor estimations. Various methods are studied in the self-position estimation problem, one of which is the use of particle filter.

Particle filter approach assumes multiple positions (particles) of the robot on the map and estimates self-position based on the distribution of the assumed positions. Since this method expresses the probability distribution based on a plurality of sampling points and their weights, high estimation accuracy can be maintained even when applied to a map with a complicated form like in outdoor environment.

A challenge with this approach is that, when the dimension of the state vector increases, the number of

necessary particles increases thereby increasing the computational cost. As such, it is difficult to apply the particle filter as is and hence the paper adopted a modified particle filter called Rao-Blackwellized Particle Filter (RBPF) available in ROS wrapper of GMapping library as published in [11]. In GMapping, Rao-Blackwell method is used as a countermeasure to increased computational cost [9], [10]. The method reduces the dimension by limiting the particle distribution to the locus of the robot and distributing the landmark position analytically for each particle. Particle filter with Rao-Blackwell leads to RBPF [12]. The mathematical concepts of RBPF is as shown below. In the Rao-Blackwell method, the state vector  $x$  is first divided into two elements.

$$\mathbf{x} = \{\mathbf{z}, \boldsymbol{\theta}\} \quad (1)$$

The distribution  $p(x)$  of the state quantity is expressed in the following form.

$$\mathbf{p}(\mathbf{x}) = \mathbf{p}(\boldsymbol{\theta})\mathbf{p}(\mathbf{z}|\boldsymbol{\theta}) \quad (2)$$

In this case, if the distribution of the second term on the right side can be obtained analytically from  $\boldsymbol{\theta}$  as shown in (3), the dimension of the state space handled can be reduced by estimating particle filter for the first term shown in (4). This is the basic idea of the Rao-Blackwell method.

$$\begin{cases} \mathbf{z}_k = \mathbf{F}(\boldsymbol{\theta}_k)\mathbf{z}_{k-1} + \mathbf{G}(\boldsymbol{\theta}_k)\mathbf{v}_k, & \mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k) \\ \mathbf{y}_k = \mathbf{H}(\boldsymbol{\theta}_k)\mathbf{z}_k + \mathbf{w}_k, & \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k) \end{cases} \quad (3)$$

$$\boldsymbol{\theta}_k \sim f_{\boldsymbol{\theta}}(\boldsymbol{\theta}_k | \boldsymbol{\theta}_{k-1}) \quad (4)$$

The algorithm is as listed below.

1. Weighted particle group at time  $k-1$  is given.

$$\{(\boldsymbol{\theta}_{k-1}^{(i)}, \bar{\mathbf{z}}_{k-1|k-1}^{(i)}, \mathbf{V}_{k-1|k-1}^{(i)}, \mathbf{w}_{k-1}^{(i)})\}_{i=1}^M \quad (5)$$

In this case,  $\mathbf{V}_{k-1|k-1}^{(i)}$  is the covariance matrix,  $\bar{\mathbf{z}}_{k-1|k-1}^{(i)}$  is the median estimated by the Kalman filter.

2. Sampling particles according to proposed distribution

$$\begin{aligned} \tilde{\boldsymbol{\theta}}_k^{(i)} &\sim \mathbf{q}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{\theta}}_k^{(i)} | \boldsymbol{\theta}_{k-1}^{(i)}, \mathbf{y}_k), i \\ &= 1, 2, \dots, M \end{aligned} \quad (6)$$

3. The Kalman filter is applied to each particle of the particle series  $\tilde{\boldsymbol{\theta}}_{1:k}^{(i)}$

[3-1]. Find the mean and variance of the first-term prediction  $p(\mathbf{z}_k | \mathbf{y}_{1:k-1}, \tilde{\boldsymbol{\theta}}_{1:k}^{(i)})$

$$\begin{cases} \bar{\mathbf{z}}_{k|k-1}^{(i)} = \mathbf{F}(\tilde{\boldsymbol{\theta}}_k^{(i)})\bar{\mathbf{z}}_{k-1|k-1}^{(i)} \\ \mathbf{V}_{k|k-1}^{(i)} = \mathbf{F}(\tilde{\boldsymbol{\theta}}_k^{(i)})\mathbf{V}_{k-1|k-1}^{(i)}\mathbf{F}'(\tilde{\boldsymbol{\theta}}_k^{(i)}) + \mathbf{G}(\tilde{\boldsymbol{\theta}}_k^{(i)})\mathbf{Q}_k\mathbf{G}'(\tilde{\boldsymbol{\theta}}_k^{(i)}) \end{cases}$$



(7)

[3-2]. Find the mean and variance of the likelihood  $p(y_k | y_{1:k-1}, \bar{\theta}_{1:k}^{(i)})$

$$\begin{cases} \bar{y}_{k|k-1}^{(i)} = H(\bar{\theta}_k^{(i)}) \bar{z}_{k|k-1}^{(i)} \\ \Sigma_{k|k-1}^{(i)} = H(\bar{\theta}_k^{(i)}) V_{k|k-1}^{(i)} H'(\bar{\theta}_k^{(i)}) + R_k \end{cases} \quad (8)$$

[3-3]. Find the mean and variance of the filter distribution  $p(z_k | y_{1:k}, \bar{\theta}_{1:k}^{(i)})$

$$\begin{cases} \bar{z}_{k|k}^{(i)} = \bar{z}_{k|k-1}^{(i)} + K_k^{(i)} [y_k - \bar{y}_{k|k-1}^{(i)}] \\ \bar{V}_{k|k}^{(i)} = [I - K_k^{(i)} H(\bar{\theta}_k^{(i)})] V_{k|k-1}^{(i)} \end{cases} \quad (9)$$

In this case,  $K_k^{(i)}$  is the Kalman gain of the  $i^{\text{th}}$  particle.

$$K_k^{(i)} = \bar{V}_{k|k-1}^{(i)} H'(\bar{\theta}_k^{(i)}) [\Sigma_{k|k-1}^{(i)}]^{-1} \quad (10)$$

[3-4]. Update weights based on observed values

$$\tilde{w}_k^{(i)} \propto w_{k-1}^{(i)} \frac{f_{\theta}(\bar{\theta}_k^{(i)} | \theta_{k-1}^{(i)}) N(y_k; \bar{y}_{k|k-1}^{(i)}, \Sigma_{k|k-1}^{(i)})}{q_{\theta}(\bar{\theta}_k^{(i)} | \theta_{k-1}^{(i)}, y_k)} \quad (11)$$

Here,  $N(y_k; \bar{y}^{(i)}, \Sigma^{(i)})$  is the normal vector of the mean vector  $\bar{y}^{(i)}$ , the variance covariance matrix  $\Sigma^{(i)}$ , and  $y$  is substituted for the probability density function.

4. Resampling

When resampling, the particle  $\bar{\theta}_k^{(i)}$  is summarized, the mean vector  $\bar{z}_{k|k}^{(i)}$ , the variance covariance matrix  $\bar{V}_{k|k}^{(i)}$ , and extracts the  $M$  recovery source with a probability proportional to the weight  $\tilde{w}_k^{(i)}$ . The particle extracted by resampling is  $\theta_k^{(i)}$ , the average vector is  $\bar{z}_{k|k}^{(i)}$ , the variance covariance matrix is  $\bar{V}_{k|k}^{(i)}$ . The weights are equalized  $w_k^{(i)} = \frac{1}{M}$ . does not change if not complete.

$$\begin{aligned} \theta_k^{(i)} &:= \bar{\theta}_k^{(i)}, & \bar{z}_{k|k}^{(i)} &:= \bar{z}_{k|k}^{(i)}, & \bar{V}_{k|k}^{(i)} &:= \bar{V}_{k|k}^{(i)}, \\ w_k^{(i)} &:= \tilde{w}_k^{(i)}, & i &= 1, 2, \dots, M \dots \end{aligned} \quad (12)$$

By the above operation, a weighted particle group of filter distribution at time  $k$  is obtained. Estimation is performed by repeating steps 1 to 4 above. Parameter initialization was performed as per the GMapping algorithm described in [11] and [13]. In this paper, 50 particles were found to yield optimal performance.

3.2. Road surface detection by 6-axis motion sensor

In this research, posture controller was introduced to reduce noise in sensor measurements due to oscillations during robot motions. In order to perform attitude control, it is necessary to design a system that measures and corrects the current posture. Although it is common to use a motion sensor to determine the posture, it is difficult to use the obtained values because drift error increase with increase in measurement time. It is necessary to correct this error using a filter or other methods like precomputed error compensation method. In this system, Madgwick Filter is used as a drift error correction method.

Madgwick Filter was proposed by Madgwick as reported in [14]. By making the algorithm quaternion base, precision equivalent to the Kalman filter is obtained with reduced processing time. The expression of quaternion is as follows.

Quaternion expresses the rotation of three-dimensional space with four components of vector  $q_1$  and scalar ( $q_2, q_3, q_4$ ). At this point, the direction of the frame B with respect to the frame A is

$$\begin{aligned} {}^A_B \hat{q} &= [q_1, q_2, q_3, q_4] \\ &= [\cos \frac{\theta}{2}, -r_x \sin \frac{\theta}{2}, -r_y \sin \frac{\theta}{2}, -r_z \sin \frac{\theta}{2}] \end{aligned} \quad (13)$$

And a conjugate quaternion obtained by inverting the sign of the vector part

$${}^A_B \hat{q}^* = {}^B_A \hat{q} = [q_1, -q_2, -q_3, -q_4] \quad (14)$$

The direction from frame A to frame C is

$${}^A_C \hat{q} = {}^B_C \hat{q} \otimes {}^A_B \hat{q} \quad (15)$$

Quaternion estimation algorithm in Madgwick filter is described below. Fig. 3 is a block diagram of the algorithm.

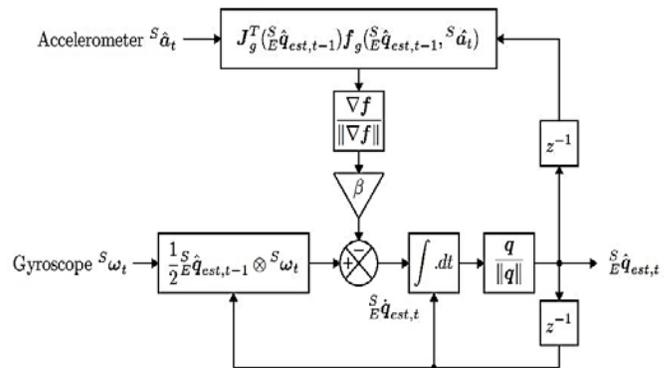


Fig. 3. Block diagram of Madgwick filter



First, using the quaternion for the angular velocity of the sensor frame is described as

$${}^S\omega = [0, \omega_x, \omega_y, \omega_z][rad/s] \quad (16)$$

The time differentiation of the rotation due to the quaternion from the ground frame to the sensor frame is as follows.

$${}^S\dot{q} = \frac{1}{2} {}^S\hat{q} \otimes {}^S\omega \quad (17)$$

Where  ${}^S\hat{q}_{\omega,t}$  is the pose at time t in discrete time,  ${}^S\hat{q}_{est,t-1}$  the estimated value of the pose at time t - 1,

$$\begin{cases} {}^S\dot{q}_{\omega,t} = \frac{1}{2} {}^S\hat{q}_{est,t-1} \otimes {}^S\omega_t \\ {}^S\hat{q}_{\omega,t} = \frac{1}{2} {}^S\hat{q}_{\omega,t} + {}^S\dot{q}_{\omega,t}\Delta t \end{cases} \quad (18)$$

Although the value of the acceleration sensor is originally the combination of the gravitational acceleration and the acceleration generated by the motion, Madgwick filter considers gravitational acceleration only since motion acceleration is what is being recovered.

At this point, if the true value is known in the base frame, the value measured by the sensor frame can be inferred from the quaternion, and the difference between the two vectors can be obtained. The posture quaternion  ${}^S\hat{q}$  whose difference is smallest is the most appropriate posture.

$$\min({}^S\hat{q} \in R^4) f({}^S\hat{q}, {}^E\hat{d}, {}^S\hat{s}) \quad (19)$$

$$f({}^S\hat{q}, {}^E\hat{d}, {}^S\hat{s}) = \frac{{}^S\hat{q}^* \otimes {}^E\hat{d} \otimes {}^S\hat{q}}{-{}^S\hat{s}} \quad (20)$$

$${}^S\hat{q} = [q_1, q_2, q_3, q_4] \quad (21)$$

$${}^E\hat{d} = [0, d_x, d_y, d_z] \quad (22)$$

$${}^S\hat{s} = [0, s_x, s_y, s_z] \quad (23)$$

Normally, the optimum value is obtained by Hessian matrix, but because the calculation load increases, steepest descent method was used.

$${}^S\hat{q}_{k+1} = {}^S\hat{q}_k - \mu \frac{\nabla f({}^S\hat{q}, {}^E\hat{d}, {}^S\hat{s})}{\|\nabla f({}^S\hat{q}, {}^E\hat{d}, {}^S\hat{s})\|} \quad (24)$$

$$k = 0, 1, 2, \dots, n$$

$$\begin{aligned} \nabla f({}^S\hat{q}, {}^E\hat{d}, {}^S\hat{s}) \\ = J^T({}^S\hat{q}, {}^E\hat{d}) f({}^S\hat{q}, {}^E\hat{d}, {}^S\hat{s}) \end{aligned} \quad (25)$$

$$\begin{aligned} f({}^S\hat{q}, {}^E\hat{d}, {}^S\hat{s}) = & \\ & \begin{bmatrix} 2d_x \left( \frac{1}{2} - q_3^2 - q_4^2 \right) + 2d_y(q_1q_4 + q_2q_3) + \dots \\ 2d_x(q_2q_3 - q_1q_4) + 2d_y \left( \frac{1}{2} - q_2^2 - q_4^2 \right) + \dots \\ 2d_x(q_1q_3 + q_2q_4) + 2d_y(q_3q_4 - q_1q_2) + \dots \end{bmatrix} \end{aligned}$$

$$\begin{bmatrix} 2d_z(q_2q_4 - q_1q_3) - s_x \\ 2d_z(q_1q_2 + q_3q_4) - s_y \\ 2d_z \left( \frac{1}{2} - q_2^2 - q_3^2 \right) - s_z \end{bmatrix} \quad (26)$$

$$\begin{aligned} J^T({}^S\hat{q}, {}^E\hat{d}) = & \\ & \begin{bmatrix} 2d_yq_4 - 2d_zq_3 & 2d_yq_3 + 2d_zq_4 \\ -2d_xq_4 + 2d_zq_2 & 2d_xq_3 - 2d_yq_2 + 2d_zq_1 \\ 2d_zq_3 + 2d_yq_2 & 2d_xq_4 - 2d_yq_1 + 2d_zq_2 \\ -4d_xq_32d_yq_2 - 2d_zq_1 & -4d_xq_4 + 2d_yq_1 + 2d_zq_2 \\ 2d_xq_2 + 2d_zq_4 & -2d_xq_1 - 4d_yq_4 + 2d_zq_3 \\ 2d_xq_1 + 2d_yq_4 - 4d_zq_3 & 2d_xq_2 + 2d_yq_3 \end{bmatrix} \end{aligned} \quad (27)$$

Next, the measurement value of the acceleration sensor is normalized by dividing it into motion and those based on acceleration of gravity

$${}^E\hat{g} = [0, 0, 0, 1] \quad (28)$$

$${}^E\hat{a} = [0, a_x, a_y, a_z] \quad (29)$$

Substituting this into equation (3-26) - (3-27) yields:

$$f({}^S\hat{q}, {}^S\hat{a}) = \begin{bmatrix} 2(q_2q_4 - q_1q_3) - a_x \\ 2(q_1q_2 + q_3q_4) - a_y \\ 2 \left( \frac{1}{2} - q_2^2 - q_3^2 \right) - a_z \end{bmatrix} \quad (30)$$

$$f({}^S\hat{q}, {}^S\hat{a}) = \begin{bmatrix} 2q_3 & 2q_4 & -2q_1 & 2q_2 \\ 2q_2 & 2q_1 & 2q_4 & 2q_3 \\ 0 & 2q_2 & -4q_3 & 0 \end{bmatrix} \quad (31)$$

Therefore, the update formula is

$${}^S\hat{q}_{\nabla,t} = {}^S\hat{q}_{est,t-1} - \mu_t \frac{\nabla f}{\|\nabla f\|} \quad (32)$$

Here, the optimum value of  $\mu_t$  can be defined as the convergence rate of  ${}^S\hat{q}_{\nabla,t}$  to avoid overshoot caused by unnecessarily large step size. Therefore,  $\mu_t$  can be obtained by the following equation.

$$\mu_t = \alpha \left\| \frac{{}^S\dot{q}_{\omega,t}}{\|\dot{q}_{\omega,t}\|} \right\| \Delta t, \quad \alpha > 1 \quad (33)$$

Finally, altitude  ${}^S\hat{q}_{\nabla,t}$  calculated from the acceleration using the above method and the altitude  ${}^S\hat{q}_{\omega,t}$  calculated from the angular velocity are fused together with filter fusion algorithm as shown. To calculate the final estimated value as  ${}^S\hat{q}_{est,t}$  and the weight of each pose quaternion  $\gamma_t$  is as follows:

$${}^S\hat{q}_{est,t} = \gamma_t {}^S\hat{q}_{\nabla,t} + (1 - \gamma_t) {}^S\hat{q}_{\omega,t}, \quad 0 \leq \gamma_t \leq 1 \quad (34)$$

Here, the optimum value of  $\gamma_t$  can be defined by the fact that the divergence of the weight of  ${}^S\hat{q}_{\omega,t}$  is equal to the convergence of the weight of  ${}^S\hat{q}_{\nabla,t}$ . It is expressed in form.



$$(1 - \gamma_t)\beta = \gamma_t \frac{\mu_t}{\Delta t} \quad (35)$$

$$\gamma_t = \frac{\beta}{\frac{\mu_t}{\Delta t} + \beta} \quad (36)$$

Here,  $\frac{\mu_t}{\Delta t}$  is the convergence rate of  ${}^S_E \mathbf{q}_{\nabla,t}$  and  $\beta$  is the divergence rate of  ${}^S_E \mathbf{q}_{\omega,t}$ .

Also, assuming a large value of  $\alpha$  relative to  $\mu_t$ , equation (32) and with  $\gamma_t \approx 0$  (35) can be approximated as shown below.

$${}^S_E \mathbf{q}_{\nabla,t} \approx -\mu_t \frac{\nabla f}{\|\nabla f\|} \quad (37)$$

$$\gamma_t \approx \frac{\beta \Delta t}{\mu_t} \quad (38)$$

Equations (18), (32) and (33) are substituted into (34) to obtain (39) below.

$${}^S_E \mathbf{q}_{est,t} = \frac{\beta \Delta t}{\mu_t} \left( -\mu_t \frac{\nabla f}{\|\nabla f\|} + (1 - 0) ({}^S_E \hat{\mathbf{q}}_{est,t-1} + {}^S_E \dot{\mathbf{q}}_{\omega,t} \Delta t) \right) \quad (39)$$

This equation can be simplified to form (40), and a filter update equation is obtained.

$${}^S_E \mathbf{q}_{est,t} = {}^S_E \hat{\mathbf{q}}_{est,t-1} + {}^S_E \dot{\mathbf{q}}_{est,t} \Delta t \quad (40)$$

Where  ${}^S_E \dot{\mathbf{q}}_{est,t}$  and  $\beta \hat{\mathbf{q}}_{\varepsilon,t}$  are as shown in (41) and (42).

$${}^S_E \dot{\mathbf{q}}_{est,t} = {}^S_E \dot{\mathbf{q}}_{\omega,t} - \beta \hat{\mathbf{q}}_{\varepsilon,t} \quad (41)$$

$$\beta \hat{\mathbf{q}}_{\varepsilon,t} = \frac{\nabla f}{\|\nabla f\|} \quad (42)$$

#### 4. Verification of Basic Performance

##### 4.1. Horizontal control of pedestal

First, horizontal control function was verified by measuring the posture of the upper part of the sensor base during translational motion. The posture measurement is realized by applying Madgwick filter to the measured values of acceleration / angular velocity from the 6-axis motion sensor. The results in Fig. 4(a) outlines the effects of applying horizontal controller while Fig. 4(b) shows the measurements when no controller is applied. When the horizontal control is not applied, the amplitude of the pitch angle indicating the forward / backward oscillation is 10 to 40 degrees around the center of 20 degrees, while it is 5 to 10 degrees around the center of 0 degree when applied. From these results, suppression of the forward and backward oscillations is confirmed.

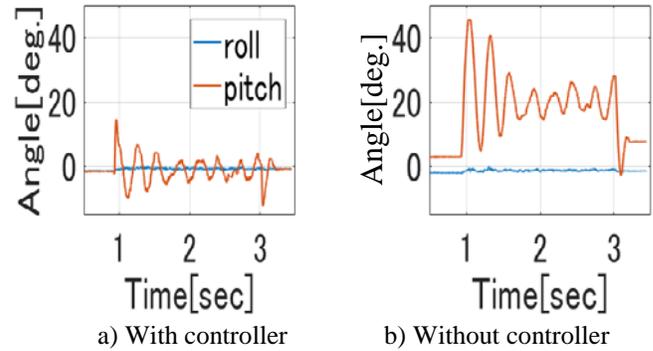
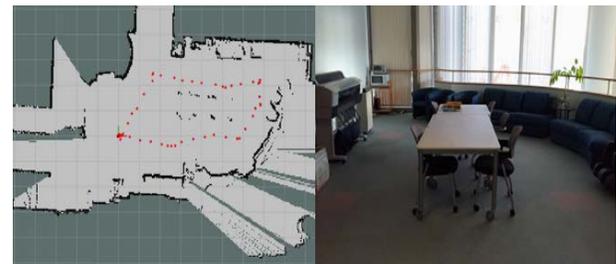


Fig.4 Performance of the horizontal controller.

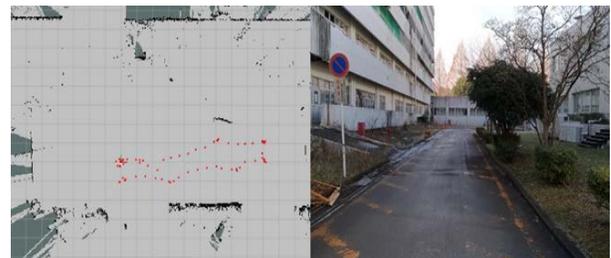
##### 4.2. Construction of environment map using attitude control

Next, map construction in multiple environments using the mobile robot equipped with horizontal controllers was performed. For the environment, indoor spaces and varying multiple outdoor spaces were selected for evaluation. The outdoor environment comprises of asphalt pavement, concrete pavement and interlocking pavement for representative sample of road surfaces. Further, two types of sloping roads, paved with non-flat surface and unpaved surface, were also evaluated. The map construction results for all the evaluated environments are shown in Fig. 5.

Fig. 5(a) shows an indoor environment (office setup). The black dots on the map indicate that an object has been detected, and the red dot shows the travel position of the robot. Fig. 5 (b)-(f) shows varying outdoor environments and corresponding construct of the map.



a) Indoor environment (Office)



b) Outdoor environment (Asphalt)

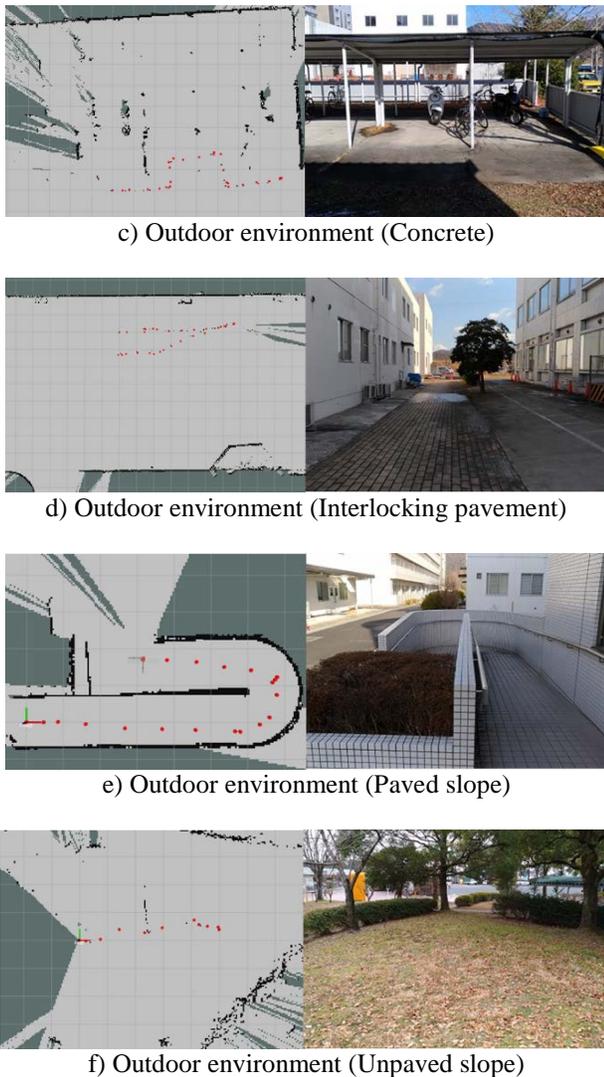


Fig. 5 Environmental map construction for different surfaces

As described above, an environmental map with coaxial two-wheeled robot both indoor and outdoor environment was constructed. The constructed maps help the robot determine a path, avoiding obstacles and walls. However, it was difficult to detect small objects such as desk legs with accuracy.

#### 4.3. Road surface detection by 6 axis Motion Sensor

Further, road surface angle detection was examined for paved slopes and unpaved slopes. From this, roughness characteristics of the road surface is targeted. Fig.6 shows the roll and pitch angle of the two sloping surfaces under evaluation. The graph in Fig. 6(a) is the paved surface and Fig. 6(b) is the unpaved surface. The pitching angle indicates forward and backward swinging/oscillations and as shown, the unpaved surface has a larger oscillation than the paved surface. These results confirmed that it is possible to characterize road surfaces from the motion sensors. However, it is challenging to accurately

quantitatively detect the road surface angle from pitch because the forward/backward movement of two-wheeled robot is a nonholonomic system with one-point contact with the road surface. Roll angle compared to pitch angle is a more appealing approach and more stable as shown in Fig. 6. This is because roll angle which indicates the lateral swing is two-point contact between the left and right wheels. For this reason, the road surface angle is more stably detected by considering the both roll and pitch angle.

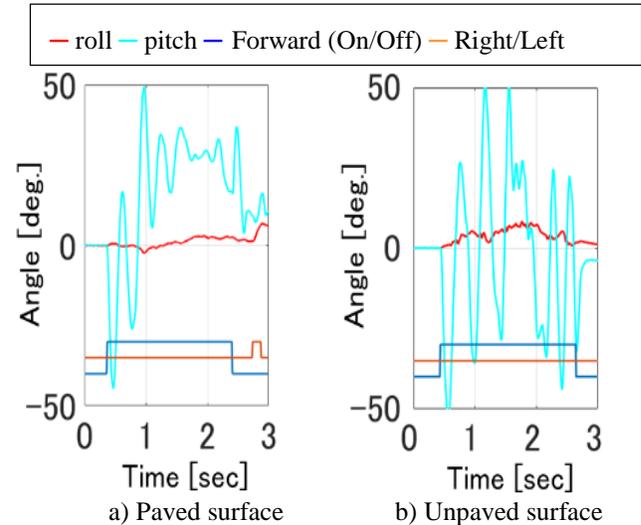


Fig. 6 Robots' roll and pitch angle during motion.

#### 4.4. Road surface type identification using deep learning

The Deep learning algorithm used in this paper is an image segmentation methodology called SegNet as reported in [15], [16] and [17]. It is one type of techniques for segmenting an image on pixel-by-pixel basis called Fully Convolutional Networks (FCN). By preparing a plurality of labelled training images to SegNet, the network can learn to classify each pixel of a general naïve image presented to it.

The configuration of SegNet consists of a series of nonlinear processing layers (encoders) and a corresponding set of decoders followed by a pixel-wise classifier. Each encoder comprises a convolutional layer with batch normalization and rectified linear unit (ReLU) activation with non-overlapping MaxPooling and subsampling. Further network settings and configurations are available in [18]. The mechanism of SegNet is shown in Fig. 7. As shown, the network detects all the objects and classify all the pixels to a unique pixel color. In the end, a silhouette of presented image is outputted.

In this paper, transfer learning is applied from the preexisting trained network to customize it to robot



classification as opposed to training the network from the beginning. This approach requires less training data and lower computation power but still achieves a comparable accuracy [19]. The network was trained with multiple images of three target road surface; asphalt pavement roads, concrete pavement roads, and interlocking paved roads which are relatively flat with less irregularities. During training, the network returns a prediction of each pixel and compares it with the label. This provides a basis for confidence value which in data analytics is referred to as global accuracy of the network. This is the total number of accurately classified pixels in the dataset. The process of training is continued until the accuracy of prediction is acceptable.

During testing, the network returns the pixel classification and an average confidence value of all predictions. Lower confidence imply uncertainty in the prediction and vice versa.

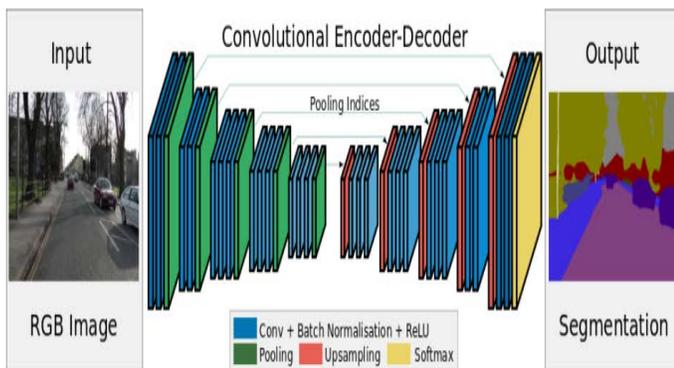


Fig.7 Typical Image segmentation using SegNet. Source: [18]

#### 4.5. Performance of Deep Learning in Road surface classification

Because of the flatness of the surfaces, it is difficult to distinguish the roads based on motion sensor only hence the need to result to other forms of characterization and or identification method. Image data is captured by camera and classified by the network. The results of the three surfaces under consideration are shown in Fig. 8 with the original image and the classification.

As seen in Fig. 8, three types of pavement roads were identified successfully with confidence of 80% or more. Some misclassifications are found in the identification of asphalt and concrete. This is thought to be caused by the difficulty to distinguish between these two colors depending on dirt on the surface and light conditions. This is predominantly shown in Fig. 8(a) where the identified image has traces of two colors. On the interlocking pavement in Fig. 8(c), the gap of the block can be

recognized as a pattern and therefore no erroneous classification occurred (100% confidence).

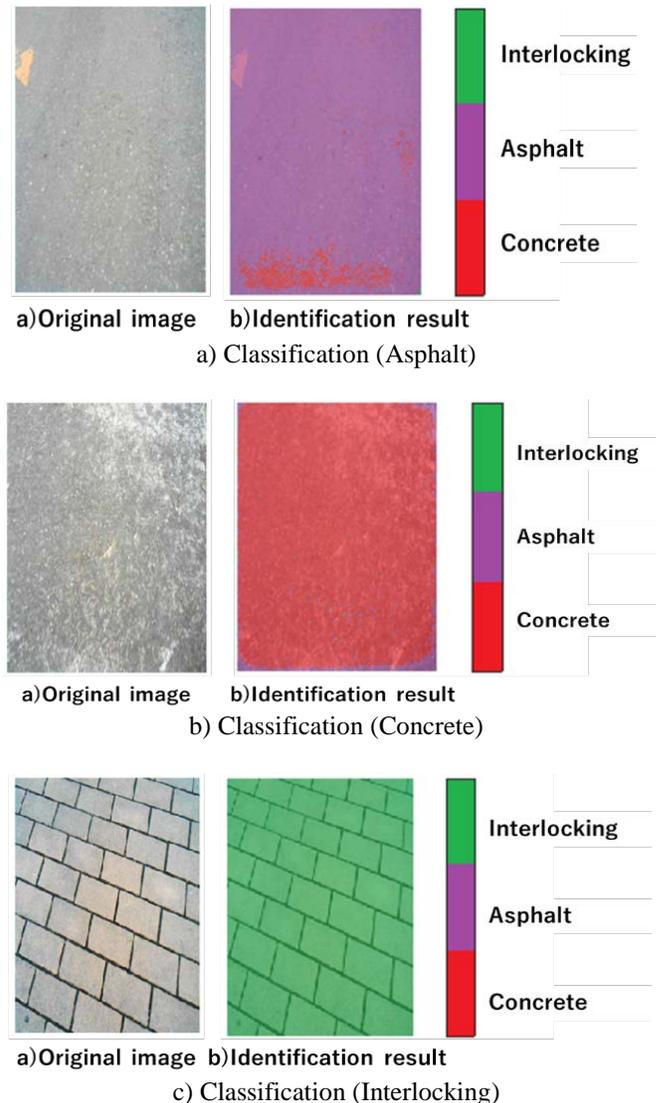


Fig 8. Classification results of road surfaces

In conclusion, map generation with range sensor data and SLAM GMapping algorithm was successful for different environments (outdoor asphalt environment, concrete environment, interlocking environment, pavement slope road and non-paved slope road). Discrimination of the plane / slope was confirmed from the magnitude of the pitch angle by the measurement of the attitude based on the 6 axis motion sensor, and it was further confirmed that the size of the road surface roughness can be identified from the size of the roll angle. The results also confirm discrimination of representative road surface (asphalt, concrete, and interlocking pavements) with confidence of more than 80%. Considering the nature of misclassified pixels, the value is acceptable and would not impact the overall performance of map generation. This conclusion



is arrived by observing the nature of the errored pixel. They are either engulfed in a sea of accurately classified pixels or are at the edges of the images.

5. Construction of road surface environment map by autonomous movement

After confirming the effectiveness of the proposed system by the above four kinds of verification experiments, the overall performance of the system was evaluated in a dynamic environment described below. Real-time road surface environment map was constructed by autonomous movement of the robot for 30 minutes in selected environment. The experiment was done in-door (office) environment with carpets of three different colors used as roads to be classified. Fig 9(a) shows a schematic diagram of the experimental environment to be reconstructed in robotic soft map. Fig.9(b) and (c) shows the experiment setup, reconstructed map and the road surface classification result respectively.

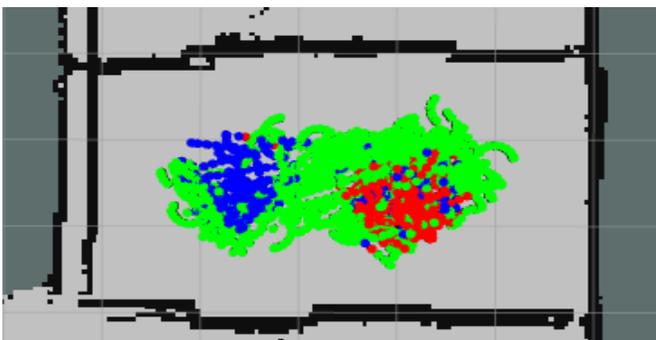
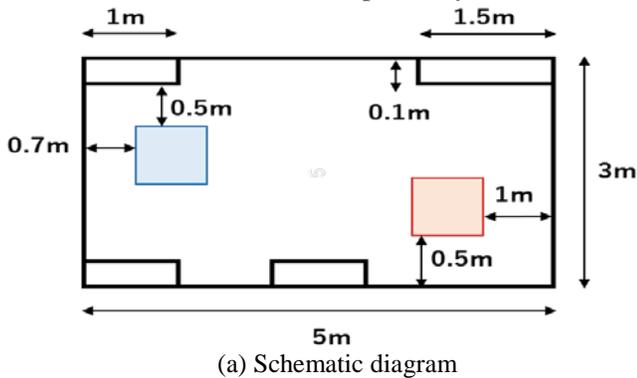


Fig. 9 Experiment setup and the generated environment map

In Fig.9(c), the classification result is displayed in color on the constructed map 20 cm ahead from the position of the robot. The green points on the map is gray on the setup area, the blue point on the constructed map correspond to the blue, and red to red areas. In this way, the objective of constructing the road surface environment map using the proposed system is achieved. Also, from Fig. 10, the roll angle did not change much implying the surface were uniformly flat. This makes surface characterization using roll and pitch difficult and hence justifying the use of classification by machine learning.

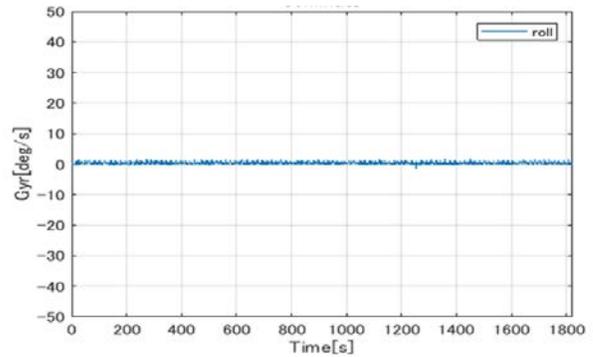
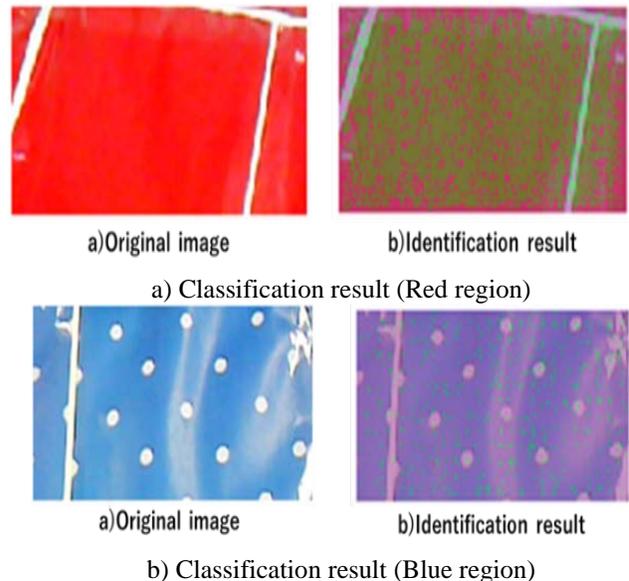


Fig.10 Altitude estimation.

Regarding road surface identification, each carpet color and the boundary surface are correctly classified. However, on the map, some parts are misidentified in blue, especially at the location of the red carpet as seen in Fig. 9(c). This is an erroneous discrimination caused by the unstable motions in the camera due to the forward and backward swing of the robot. The classification result is as shown in Fig. 11.



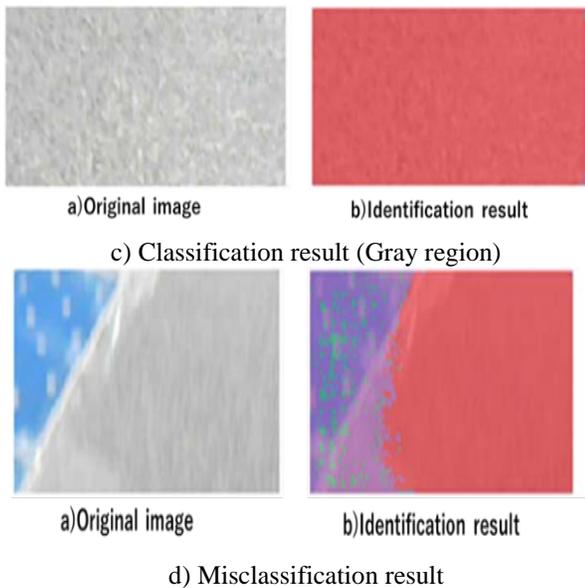


Fig. 11 Classification results of different region (colors)

From the results, both map construction and road surface classification were performed proving the viability of the proposed method with the setup. A challenge with the proposal was on the buildup of self-position estimation error emanating from long operation time. A maximum object positioning error of about 30 cm was noted.

With respect to the carpet color, an erroneous identification of red carpet occurred. This was attributed to image quality and motion blurs. This will be addressed in a later study by optimal positioning of the camera or any other approach. Similarly, a larger training data and scenes will further be incorporated to increase accuracy of classification.

## 6. Conclusion

The objective of this study was to develop an autonomous mobile robot that can operate in a human living environment. This is done by constructing a road surface environment map with a two-wheeled robot in real-time to enable accurate navigation of the robot in constrained spaces. Coaxial two-wheel mechanism has a disadvantage of causing forward and backward swinging during motion and as such, the proposed system using SLAM algorithm with altitude controller on the sensor base for stabilization.

The effectiveness of each function of the system constructed in this experiment was verified. The contents of the verification were divided into four: (1) horizontal maintenance by the altitude controller, (2) map construction by the sensor, (3) road sensing by the IMU sensor, and (4) road surface classification from image and

sensor data.

From the results, tilt due to the forward and backward swinging during motion was reduced from 40 degrees to 10 degrees by the proposed posture controller. An environment map for various environments was generated using image data and IMU sensor data. The errored classification of the map is attributed to the light conditions and image quality. Further research will be undertaken to improve the confidence of prediction which was reported to range from 80%.

## Acknowledgement

This work is partially supported by Grants-in-aid for Promotion of Regional Industry-University-Government Collaboration from Cabinet Office, Japan.

## References

- [1] I. Satoshi, N. Kosuke, S. Shoya and M. Ryosuke, "An autonomous mobile robot with passive wheels propelled by a single motor," *Robotics and Autonomous Systems*, vol. 122, 2019.
- [2] P. M. C. Ronald, A. S. Karl and C. H. Roger, "Review of modelling and control of two-wheeled robots," *Annual Reviews in Control*, vol. 37, no. 1, pp. 89-103, 2013.
- [3] J. Y. Tomi, J. H. Aarne and J. F. Pekka, "Dynamic modeling and obstacle-crossing capability of flexible pendulum-driven ball-shaped robots," *Robotics and Autonomous Systems*, vol. 87, p. 269–280, 2017.
- [4] L. Weihua, L. Zhencai, L. Yiqun, D. Liang, W. Jianfeng, G. Haibo and D. Zongquan, "Semi-autonomous bilateral teleoperation of six-wheeled mobile robot on soft terrains," *Mechanical Systems and Signal Processing*, vol. 133, 2019.
- [5] I. Hiroki, N. Katsuki, M. Ryutaro and N. Haruo, "Path Generation with Clothoid Curve Using Image Processing for Two-Wheel-Drive Autonomous Mobile Robots," in *IFAC Proceedings Volumes*, 2009.
- [6] B. Patle, B. Ganesh, P. Anish, D. Parhi and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technology*, vol. 15, no. 4, pp. 582-606, 2019.
- [7] ROS, "Robot Operating System (ROS)," [Online]. Available: <http://wiki.ros.org/>. [Accessed January 2019].
- [8] "Deep Learning," Mathworks Inc., [Online]. Available: <https://www.mathworks.com/discovery/deep-learning.html>. [Accessed March 2019].
- [9] G. Giorgio, S. Cyrill and B. Wolfram, "Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling;," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2005.
- [10] G. Giorgio, S. Cyrill and B. Wolfram, "Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters;," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34-36, 2007.
- [11] G. Giorgio, S. Cyrill and B. Wolfram, "GMmapping," openSLAM, 2017. [Online]. Available: <https://openslam-org.github.io/gmapping.html>. [Accessed March 2019].



- [12] H. Hara, T. Tsubouchi and A. Oshima, "Rao-Blackwellized Particle Filter SLAM considering the past with stochastically accumulated scan geometry," *Journal of the Japan Society of Mechanical Engineers*, vol. 82, no. 834, 2016.
- [13] Myrobotics, "ROS GMAPPING," 31 7 2018. [Online]. Available: <https://sy-base.com/myrobotics/ros/gmapping/>. [Accessed 3 2019].
- [14] S. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," *Report x-io and University of Bristol (UK)*, vol. 25, 2010.
- [15] K. Alex, B. Vijay and C. Roberto, "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding," *arXiv preprint arXiv:1511.02680*, 2015.
- [16] B. Vijay, K. Alex and C. Roberto, " SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," in *PAMI*, 2017.
- [17] B. Vijay, H. Ankur and C. Roberto, " SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling;," *arXiv preprint arXiv:1505.07293*, 2015.
- [18] K. Alex, B. Vijay and C. Roberto, "SegNet," University of Cambridge, [Online]. Available: <http://mi.eng.cam.ac.uk/projects/segnet/>. [Accessed March 2019].
- [19] T. Chuanqi, S. Fuchun, K. Tao, Z. Wenchang, Y. Chao and L. Chunfang, "A Survey on Deep Transfer Learning," *Cornell University Press*, 2018.