

Trajectory Tracking and Fault Detection Algorithm for Automatic Guided Vehicle Based on Multiple Positioning Modules

Pandu Sandi Pratama, Amruta Vinod Gulakari, Yuhanes Dedy Setiawan, Dae Hwan Kim, Hak Kyeong Kim, and Sang Bong Kim*

Abstract: This paper presents an implementation and experimental validation of trajectory tracking and fault detection algorithm for sensors and actuators of Automatic Guided Vehicle (AGV) system based on multiple positioning modules. Firstly, the system description and the mathematical modeling of the differential drive AGV system are described. Secondly, a trajectory tracking controller based on the backstepping method is proposed to track the given trajectory. Thirdly, a fault detection algorithm based on the multiple positioning modules is proposed. The AGV uses encoders, laser scanner, and laser navigation system to obtain the position information. To understand the characteristics of each positioning module, their modeling are explained. The fault detection method uses two or more positioning systems and compares them using Extended Kalman Filter (EKF) to detect an unexpected deviation effected by fault. The pairwise differences between the estimated positions obtained from the sensors are called as residue. When the faults occur, the residue value is greater than the threshold value. Fault isolation is obtained by examining the biggest residue. Finally, to demonstrate the capability of the proposed algorithm, it is applied to the differential drive AGV system. The simulation and experimental results show that the proposed algorithm successfully detects the faults when the faults occur.

Keywords: Automatic guided vehicle, extended Kalman filter, fault detection, multiple positioning.

NOMENCLATURE

(X_C, Y_C, θ_C) :	AGV pose in global coordinates frame	\mathbf{Q}_E :	process noise covariance for encoder
v_{cx}, v_{cy} :	linear velocities in local coordinate frame	k_r, k_l :	error constants related to encoders
ω_c :	angular velocity of AGV	(X_L, Y_L, θ_L) :	AGV position and orientation obtained from the laser position module
T :	sampling time	$\mathbf{u}_L = [\Delta x_L \ \Delta y_L \ \Delta \theta_L]^T$:	input for laser positioning model
(X_R, Y_R, θ_R) :	reference position and orientation	$\Delta x_L, \Delta y_L, \Delta \theta_L$:	changes in x and y direction and orientation
v_R, θ_R :	reference linear and angular velocities of AGV	\mathbf{Q}_L :	process noise covariance for laser positioning
e_1, e_2, e_3 :	tracking error in local coordinate of AGV	k_x, k_y, k_θ :	error constants related to laser positioning
k_1, k_2, k_3 :	control gains	(X_N, Y_N, θ_N) :	AGV position and orientation obtained from the NAV position module
$\mathbf{u}_c = [v_{xc} \ \omega_c]^T$:	output controller	n_x, n_y, n_θ :	error constants related to NAV positioning
r :	wheel radius	\mathbf{x}_{ik} :	state vector
b :	distance between the wheels and the geometric center of AGV	\mathbf{w}_{ik-1} :	process noise at previous time $k-1$
(X_E, Y_E, θ_E) :	AGV position and orientation obtained from the encoder position input for encoder model	\mathbf{v}_{ik} :	observation noises at current time k
$\mathbf{u}_E = [\Delta \phi_1 \ \Delta \phi_2]^T$:		$f_i(\cdot)$:	process nonlinear vector function
$\Delta \phi_1, \Delta \phi_2$:	changes of right and left wheel rotation angles		

Manuscript received July 23, 2014; revised April 20, 2015 and August 10, 2015; accepted October 5, 2015. Recommended by Editor Hyouk Ryeol Choi. This work was supported by a Research Grant of Pukyong National University (2014 year).

Pandu Sandi Pratama, Amruta Vinod Gulakari, Yuhanes Dedy Setiawan, Dae Hwan Kim, Hak Kyeong Kim, and Sang Bong Kim are with Department of Mechanical Design Engineering, Pukyong National University, Busan 48547, Korea (e-mails: {pandu.sandi, amrutagulakari, yuhanesdedy}@gmail.com, {kimdh2599, hakkyeong, kimsb}@pknu.ac.kr)

* Corresponding author.

$h_i(\cdot)$:	observation nonlinear vector function
\mathbf{z}_{ik} :	output vector at current time k
$\tilde{\mathbf{y}}_{ik}$:	measurement innovation at current time k
s_i :	residue
Th :	threshold value

1. INTRODUCTION

Automatic Guided Vehicles (AGVs) are the load carriers that travel along the given path with precisely controlled acceleration and deceleration on the floor of work environment without manual operator or driver. Typical application of AGVs is the transportation of raw material and finished goods in the industries to support the production lines and the storage. Since the AGVs work automatically, their reliability and safety are the crucial factors to be considered. Therefore, many fault detection algorithms were proposed to increase the safety and reliability of AGVs. The fault detection algorithm helps user to detect faults and prevents serious damage in the AGVs. The main purpose to develop the fault detection algorithm is to detect faults as quick as possible with the minimum false alarms [1].

There are several approaches developed to make the fault detection algorithm for different applications. As in [2], the fault detection algorithm based on local model neural networks was proposed to deal with the absence of a mathematical model. For the known mathematical modeling, [3, 4] proposed the fault detection algorithm based on unknown input PI observer. The adaptive fault detection algorithm for the stuck actuator was proposed by [5]. Robust fault detection filter based on H_∞ model matching was proposed by [6, 7]. However, these algorithms only worked for linear systems. For nonlinear systems, fault detection and isolation scheme were proposed by [8, 9]. In [9, 10], a particle filter was used for the fault detection and isolation. It could be used for nonlinear, non-Gaussian and multi-modal distributions. However, the computational costs for these algorithms were high. Moreover, a real-time model based sensor fault detection algorithm for unmanned ground vehicles was proposed by [11] which have used multiple sensors such as GPS, IMU sensor and encoders. However, this method was operated only on the sensor level.

Therefore, to deal with these problems, this paper proposes an improved fault detection algorithm for the AGV based on multiple positioning modules with known mathematical model and low computational cost. In this paper, the fault detection method uses two or more positioning systems and compares them using Extended Kalman Filter (EKF) to detect unexpected deviation effected by fault. The EKF is comparatively faster than the particle filter [12]. Residue is obtained from the pairwise differences between the estimated positions obtained from sen-

sors. When the faults occur, the residue value is greater than the threshold value. Fault isolation is obtained by examining the biggest residue.

2. SYSTEM DESCRIPTION

The system description of the AGV system is shown in Fig. 1. The AGV dimension is 60 x 100 x 190 cm. This system uses differential wheel drive system. Two driving wheels are mounted on the left and right sides of the AGV, and are driven by two BLDC motors. Two passive castor wheels are installed in front and back sides to support the AGV.

The electrical system description is shown in Fig. 2. The laser navigation system NAV-200 is used as a positioning sensor with the accuracy of ± 25 mm and is mounted on the top of the AGV. The laser measurement system LMS-151 is used to measure the distance between the robot and the landmark positions and is attached in front of the AGV. Two incremental encoders are attached on the left and right wheels to count the rotations of the

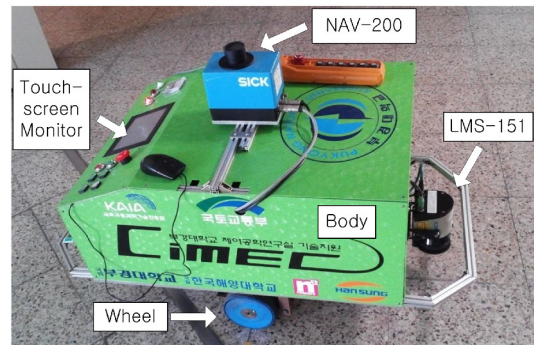


Fig. 1. Mechanical system description.

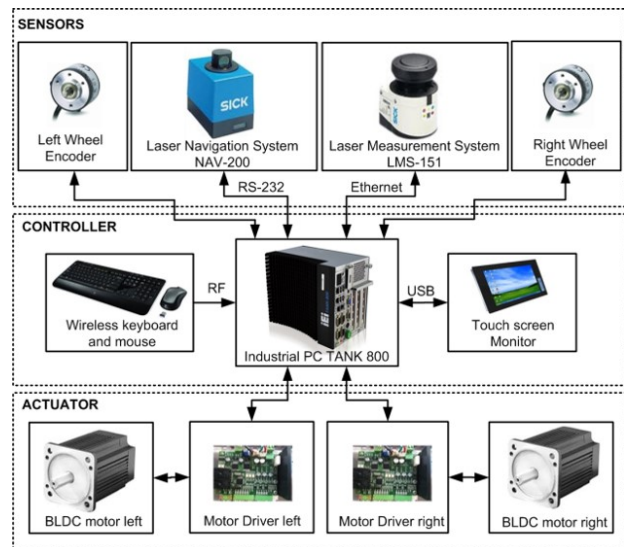


Fig. 2. Electrical system description.

wheels. Industrial PC as the main controller is placed inside the AGV platform. A touch screen monitor is used as an input and display device and is placed on the back side of AGV. The batteries for power supply are placed in the middle of AGV.

3. MATHEMATICAL MODEL

A system modeling for the AGV system is shown in Fig. 3. Kinematic equation of nonholonomic differential drive of AGV system as shown in Fig. 3 can be expressed as follows:

$$\begin{aligned} \mathbf{x}_C &= \int \dot{\mathbf{x}}_C dt, \\ \dot{\mathbf{x}}_C &= \begin{bmatrix} \dot{X}_C \\ \dot{Y}_C \\ \dot{\theta}_C \end{bmatrix} = \begin{bmatrix} \cos \theta_C & -\sin \theta_C & 0 \\ \sin \theta_C & \cos \theta_C & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{cx} \\ v_{cy} \\ \omega_c \end{bmatrix}, \end{aligned} \quad (1)$$

or in discrete type

$$\begin{aligned} \begin{bmatrix} X_C \\ Y_C \\ \theta_C \end{bmatrix}_k &= \begin{bmatrix} X_C \\ Y_C \\ \theta_C \end{bmatrix}_{k-1} \\ &+ \begin{bmatrix} \cos(\theta_C) & -\sin(\theta_C) & 0 \\ \sin(\theta_C) & \cos(\theta_C) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{cx}T \\ v_{cy}T \\ \omega_c T \end{bmatrix}_k, \end{aligned} \quad (2)$$

where \mathbf{x}_C is posture vector of AGV, (X_C, Y_C) is AGV position coordinate in global coordinates frame XOY , θ_C is AGV orientation, v_{cx} and v_{cy} are linear velocities in local coordinate frame, ω_c is angular velocity of AGV, T is sampling time. Normally, the value of v_{cy} is zero because the AGV can't move in perpendicular direction to the orientation of AGV under no slipping condition and pure rolling

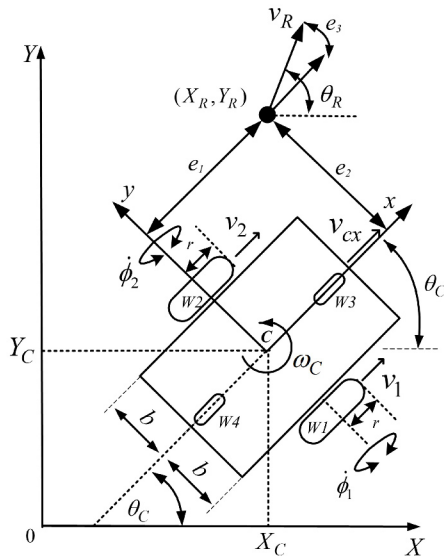


Fig. 3. System modeling.

condition. As a result, the second column of rotation matrix in (2) can be omitted.

4. CONTROLLER DESIGN

The purpose of this section is to design a trajectory tracking controller for the AGV to track the reference position $(X_R(t), Y_R(t))$ and the reference orientation $\theta_R(t)$ with the reference linear velocity $v_R(t)$ and the angular velocity $\omega_R(t)$. As shown in Fig. 3, the tracking error vector and its time derivative are defined as follows:

$$\begin{aligned} \mathbf{e}(t) &= \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} \\ &= \begin{bmatrix} \cos \theta_C & \sin \theta_C & 0 \\ -\sin \theta_C & \cos \theta_C & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_R - X_C \\ Y_R - Y_C \\ \theta_R - \theta_C \end{bmatrix}, \end{aligned} \quad (3)$$

$$\begin{aligned} \dot{\mathbf{e}}(t) &= \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \cos e_3 & 0 \\ \sin e_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_R \\ \omega_R \end{bmatrix} \\ &+ \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_{cx} \\ \omega_c \end{bmatrix}. \end{aligned} \quad (4)$$

To guarantee the stability of the system, Lyapunov function is chosen as:

$$V_0 = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 + \frac{1}{k_2}(1 - \cos e_3) \quad \text{for } k_2 > 0, \quad (5)$$

and its derivatives becomes

$$\begin{aligned} \dot{V}_0 &= e_1 \dot{e}_1 + e_2 \dot{e}_2 + \frac{1}{k_2}(\sin e_3) \dot{e}_3 \\ &= e_1(-v_{xc} + v_R \cos e_3) \\ &+ \frac{1}{k_2}(\sin e_3)(\omega_R - \omega_c + k_2 e_2 v_R). \end{aligned} \quad (6)$$

To achieve $\dot{V}_0 \leq 0$, the control law \mathbf{u}_c is chosen as follows:

$$\mathbf{u}_c = \begin{bmatrix} v_{xc} \\ \omega_c \end{bmatrix} = \begin{bmatrix} v_R \cos e_3 + k_1 e_1 \\ \omega_R + k_2 v_R e_2 + k_3 \sin e_3 \end{bmatrix}. \quad (7)$$

The schematics diagram of the proposed controller is shown in Fig. 4.

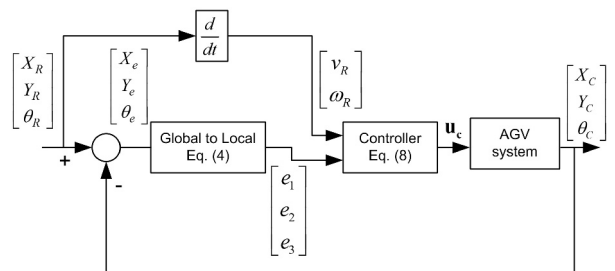


Fig. 4. Schematic diagram of the proposed controller.

5. POSITIONING MODULES

The term positioning module is used for the sensors or algorithms that provide position estimation. In this paper, three positioning modules are used for the AGV as follows: encoders, laser scanner LMS-151, and laser navigation system NAV-200. In the extended Kalman filter (EKF), process model, process noise covariance and measurement noise covariance are the important parameters. Therefore, in this section, modeling of each positioning module is presented, and its noise covariance is defined.

5.1. Encoder positioning module modeling

The encoder positioning for differential drive type of AGV is based on the accumulation of wheel rotation. Therefore, small measurement errors will cause drift after passing through the integration. The mathematical model of encoder positioning module can be expressed as

$$\begin{bmatrix} X_E \\ Y_E \\ \theta_E \end{bmatrix}_{k|k-1} = \begin{bmatrix} X_E \\ Y_E \\ \theta_E \end{bmatrix}_{k-1} + \begin{bmatrix} \cos(\theta_E) & -\sin(\theta_E) & 0 \\ \sin(\theta_E) & \cos(\theta_E) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_E \\ 0 \\ \Delta \theta_E \end{bmatrix}_k, \quad (9)$$

$$\begin{bmatrix} \Delta x_E \\ \Delta \theta_E \end{bmatrix} = \frac{r}{2} \begin{bmatrix} 1 & 1 \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix} \begin{bmatrix} \Delta \phi_1 \\ \Delta \phi_2 \end{bmatrix}, \quad (10)$$

where r is the wheel radius, b is the distance between the wheels and the geometric center of the AGV. The input for this model is chosen as $\mathbf{u}_E = [\Delta \phi_1 \quad \Delta \phi_2]^T$, where $\Delta \phi_1$ and $\Delta \phi_2$ are the changes of right and left wheel rotation angles.

The process noise covariance is given as

$$\mathbf{Q}_{E_{k-1}} = \begin{bmatrix} k_r |\Delta \phi_1| & 0 \\ 0 & k_l |\Delta \phi_2| \end{bmatrix}, \quad (11)$$

where k_r and k_l are error constants related with encoders.

The fault in encoder is caused by slip, calibration error, communication error, and damage on the sensor.

5.2. Laser scanner positioning module modeling

For the laser scanner positioning module, Simultaneous Localization and Mapping (SLAM) algorithm [13] is adopted based on location of the landmarks as shown in Fig. 5. In this paper, ‘metal rod’ landmarks are used for the experiment. Firstly, landmarks are detected as shown in Fig. 5(a). As the AGV moves, the encoder data change and the AGV new position is predicted by the EKF prediction step based on encoder data as shown in Fig. 5(b). Secondly, landmarks are detected from the AGV new position as in Fig. 5(c). The AGV then associates these landmarks with observations of landmarks that are previously observed. Re-observed landmarks are used to update the

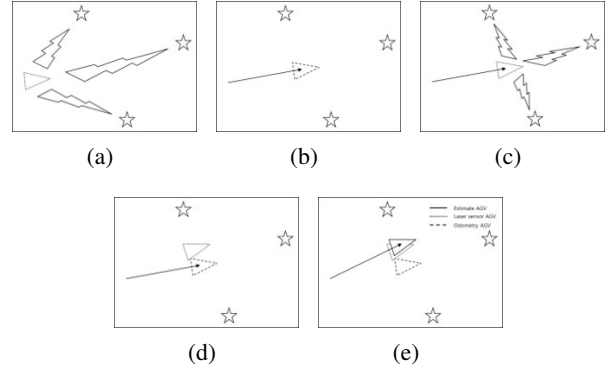


Fig. 5. Laser scanner positioning modeling.

AGV position using EKF update step as shown in Fig. 5(d). The real position, the encoder position and the estimated position of the AGV are shown in Fig. 5(e).

The position estimation will drift because of the integration, matching error, noise and numerical errors with accumulated overtime. The mathematical modeling of laser scanner positioning module can be given as

$$\begin{bmatrix} X_L \\ Y_L \\ \theta_L \end{bmatrix}_{k|k-1} = \begin{bmatrix} X_L \\ Y_L \\ \theta_L \end{bmatrix}_{k-1} + \begin{bmatrix} \cos(\theta_L) & -\sin(\theta_L) & 0 \\ \sin(\theta_L) & \cos(\theta_L) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_L \\ \Delta y_L \\ \Delta \theta_L \end{bmatrix}_k. \quad (12)$$

The input for this model is defined by $\mathbf{u}_L = [\Delta x_L \quad \Delta y_L \quad \Delta \theta_L]^T$.

The process noise covariance is

$$\mathbf{Q}_{L_{k-1}} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_\theta \end{bmatrix}, \quad (13)$$

where k_x , k_y and k_θ are error constants. Equation (12) is reasonable model because SLAM error is independent of the robot direction. The error depends on the difference between current scanning result and previous scanning result. Communication error, sensor damage and singularity are several example of faults related with laser scanner when the AGC moves on the long corridor or open space.

5.3. NAV-200 positioning module modeling

Using triangulation method, the NAV-200 positioning sensor obtains the AGV position. Reflectors are positioned around the environment and their locations are given to the sensor as shown in Fig. 6. The sensor then calculates its position in the environment from the observed positions of the reflectors [14].

Unlike another positioning module, the NAV-200 is a position sensor. Therefore, the position data can be obtained directly through serial data communication without

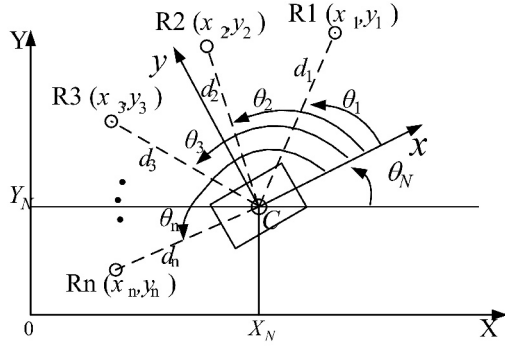


Fig. 6. NAV-200 positioning module modeling.

any further calculation. However, the position obtained from this sensor also have error caused by measurement noise. The mathematical modeling of the NAV-200 positioning module including measurement noise can be given as follows:

$$\begin{bmatrix} X_N \\ Y_N \\ \theta_N \end{bmatrix}_{k|k-1} = \begin{bmatrix} X_N \\ Y_N \\ \theta_N \end{bmatrix}_k + \begin{bmatrix} n_x \\ n_y \\ n_\theta \end{bmatrix}_k. \quad (14)$$

The fault related with the NAV-200 is caused by communication error, insufficient number of reflectors, singularity, etc.

6. FAULT DETECTION ALGORITHM

The flowchart of the fault detection algorithm is shown in Fig. 7. The EKF calculates the measurement probability distribution of the AGV position for nonlinear models driven by Gaussian noise. Using the probability distribution of innovation obtained from the EKF, it is possible to test whether the measured data are fit with the models or not. When the faults occur, the models will not be valid and the innovation will not be Gaussian and white. The residue calculation estimates the residue from pairwise differences between the estimated positions obtained from the positioning modules. Fault isolation is obtained by examining the biggest residue. Finally, from the Table 2 as shown in the next section, the fault position is known.

6.1. Extended Kalman Filter

Kalman filter theory provides a method to calculate measurement probability for linear model driven by Gaussian noise. Since the proposed system is nonlinear, the EKF is used.

In [15], a convergence analysis of the EKF as an observer for nonlinear deterministic discrete systems was done. It shows that the EKF converges locally for a broad class of nonlinear system. If the initial estimation error of the filter is not too large, the error goes to zero exponentially as time goes to infinity. In this paper, the initial estimation error is assumed to be zero.

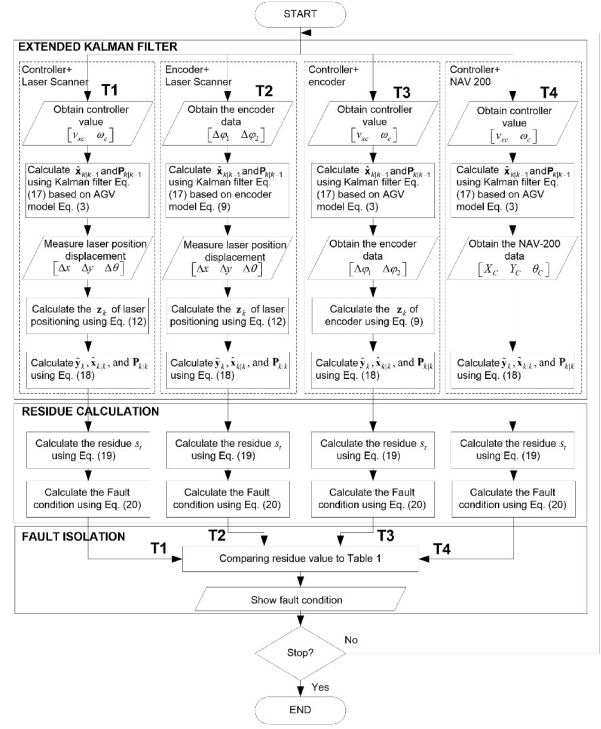


Fig. 7. Flowchart of the proposed fault detection algorithm.

In the EKF, the state transition model and the observation model are not linear functions of the state but may be differentiable functions as follows:

$$\begin{cases} \mathbf{x}_{ik} = f_i(\mathbf{x}_{ik-1}, \mathbf{u}_{ik-1}) + \mathbf{w}_{ik-1} \\ \mathbf{z}_{ik} = h_i(\mathbf{x}_{ik}) + \mathbf{v}_{ik}, \end{cases} \quad (15)$$

where \mathbf{x}_{ik} is the state vector at time k , \mathbf{w}_{ik-1} is the process noise at previous time $k-1$ and \mathbf{v}_{ik} is the observation noise at time k which are assumed to be zero mean multivariate Gaussian noises with covariance \mathbf{Q}_{ik} and \mathbf{R}_{ik} at time k , respectively. $f_i(\cdot)$ is the process nonlinear vector function, $h_i(\cdot)$ is the observation nonlinear vector function, and \mathbf{z}_{ik} is the output vector current time k .

In this paper, the state equation $f(\cdot)$ can be obtained from 3 cases such as: (9) for encoder positioning module, (12) for laser scanner positioning module and (14) for NAV-200 position module. For the measurement model, \mathbf{z}_{ik} is used as the following equation:

$$\mathbf{z}_{ik} = \mathbf{x}_{ik} + \mathbf{v}_{ik}. \quad (16)$$

The EKF consists of two steps as follows:

- Prediction step:

$$\begin{cases} \hat{\mathbf{x}}_{ik|k-1} = f_i(\hat{\mathbf{x}}_{ik-1|k-1}, \mathbf{u}_{ik-1}) \\ \mathbf{P}_{ik|k-1} = \mathbf{F}_{ik-1} \mathbf{P}_{ik-1|k-1} \mathbf{F}_{ik-1}^T + \mathbf{W}_{ik} \mathbf{Q}_{ik-1} \mathbf{W}_{ik}^T \end{cases}$$

Table 1. Extended Kalman Filter test condition.

Test	Estimation Input	State Equation	Measurement	EKF output
T1	Controller + LMS	(3)	X_L, Y_L, θ_L	$\tilde{\mathbf{y}}_{1k} \mathbf{S}_{1k}$
T2	Encoder + LMS	(9)	X_L, Y_L, θ_L	$\tilde{\mathbf{y}}_{2k} \mathbf{S}_{2k}$
T3	Controller + encoder	(3)	X_E, Y_E, θ_E	$\tilde{\mathbf{y}}_{3k} \mathbf{S}_{3k}$
T4	Controller + NAV	(3)	X_N, Y_N, θ_N	$\tilde{\mathbf{y}}_{4k} \mathbf{S}_{4k}$

$$\begin{cases} \mathbf{F}_{ik-1} = \frac{\partial f_i}{\partial \mathbf{x}_i} \Big|_{\hat{\mathbf{x}}_{i|k-1}, \mathbf{u}_{i|k-1}} \\ \mathbf{W}_{ik} = \frac{\partial f_i}{\partial \mathbf{u}_i} \Big|_{\hat{\mathbf{x}}_{i|k-1}, \mathbf{u}_{i|k-1}} \end{cases}, \quad (17)$$

where $\hat{\mathbf{x}}_{i|k-1}$ is the predicted state estimate at time k , and $\mathbf{P}_{i|k-1}$ is the predicted covariance matrix estimate at time k .

- Update step:

$$\begin{cases} \tilde{\mathbf{y}}_{ik} = \mathbf{z}_{ik} - h(\hat{\mathbf{x}}_{i|k-1}) \\ \mathbf{S}_{ik} = \mathbf{H}_{ik} \mathbf{P}_{i|k-1} \mathbf{H}_{ik}^T + \mathbf{R}_{ik} \\ \mathbf{K}_{ik} = \mathbf{P}_{i|k-1} \mathbf{H}_{ik}^T (\mathbf{S}_{ik})^{-1} \\ \hat{\mathbf{x}}_{i|k} = \hat{\mathbf{x}}_{i|k-1} + \mathbf{K}_{ik} \tilde{\mathbf{y}}_{ik} \quad \tilde{\mathbf{y}}_{ik} = \mathbf{z}_{ik} - h_i(\hat{\mathbf{x}}_{i|k-1}) \\ \mathbf{P}_{i|k} = (\mathbf{I} - \mathbf{K}_{ik} \mathbf{H}_{ik}) \mathbf{P}_{i|k-1} \\ \mathbf{H}_{ik} = \frac{\partial h_i}{\partial \mathbf{x}_i} \Big|_{\hat{\mathbf{x}}_{i|k-1}} \end{cases}, \quad (18)$$

where $\tilde{\mathbf{y}}_{ik}$ is the measurement innovation at time k , \mathbf{z}_{ik} is the output vector from estimation at time k , $h_i(\hat{\mathbf{x}}_{i|k-1})$ is the measurement result, \mathbf{S}_{ik} is the innovation covariance at time k , \mathbf{R}_{ik} is the measurement noise covariance at time k , \mathbf{K}_{ik} is the Kalman gain at time k , $\hat{\mathbf{x}}_{i|k}$ is the updated state estimation at time k , and $\mathbf{P}_{i|k}$ is the updated covariance matrix estimation at time k .

This paper implements multiple EKFs with multiple independent state vectors. Pairwise of positioning modules, as shown in Table 1, are tested using Kalman filter.

6.2. Residue calculation

From the EKF, the measurement innovation $\tilde{\mathbf{y}}_{ik}$ and the innovation covariance \mathbf{S}_{ik} can be obtained. To check the residue measurement, Mahalanobis distance [16] is used. The residue $(s_i)_k$ at time k can be calculated as follows:

$$(s_i)_k = (\tilde{\mathbf{y}}_i^T \mathbf{S}_i^{-1} \tilde{\mathbf{y}}_i)_k \text{ for } i = 1, 2, 3, 4. \quad (19)$$

Table 2. Fault isolation with respect to residue.

Test	Fault					
	N	F1	F2	F3	F4	
	<i>Normal</i>	<i>Encoder</i>	<i>BLDC motor</i>	<i>Laser scanner</i>	<i>NAV</i>	
T1	Controller + laser scanner (LMS)	0	0	X	X	0
T2	Encoder + laser scanner (LMS)	0	X	0	X	0
T3	Controller + encoder	0	X	X	0	0
T4	Controller + NAV	0	0	0	0	X

* 0 means normal, X means fault.

If the value of s_i is larger than threshold value Th , one of the pairwise sensors is fault condition as follows [17]:

$$\text{Fault condition} \begin{cases} \text{normal} & 0 & \text{if } s_i < Th \\ \text{fault} & X & \text{if } s_i > Th. \end{cases} \quad (20)$$

The threshold value determines how system is sensitive to the fault. The existence of model uncertainty and disturbances such as noise can cause a false alarm. Therefore, it is important to design a threshold to accommodate uncertainties in the model that would help in minimizing false alarms and missed detections. In this paper, the threshold value is predetermined and constant based on empirical data.

6.3. Fault isolation

Finally, to isolate the fault, 4 test conditions have been setup as shown in Table 2. The residue values above the threshold value are examined, and then matched with the Table 2 showing the relation between residues and faults. Several fault conditions are considered as follows:

- In normal condition (N), all positioning modules provide correct position. Therefore, all tests are normal since all residues are less than the threshold value.
- When the encoder is broken (F1), all tests related with encoder such as test T2 and test T3 have the fault results since the positioning from encoder is false and the resulting residue from both tests T2 and T3 are greater than the threshold value.

- When the BLDC motor is broken (F2), all tests related with controller such as test T1 and test T3 have the fault results since the positioning from controller is false and the resulting residue from both tests T1 and T3 are greater than the threshold value.
- When laser scanner is broken (F3), all tests related with laser scanner such as test T1 and test T2 have the fault result since positioning from laser scanner is false. The resulting residue from both tests T1 and T3 are greater than the threshold value.
- When NAV-200 is broken (F4), the NAV would not provide the correct position. Therefore, the residue from test T4 is greater than the threshold value.

7. EXPERIMENTAL RESULT

To verify the effectiveness of the proposed fault detection algorithm, several experiments has been conducted. Table 3 shows the parameters and their initial values for the experiment.

During the experiments, the AGV follows the given trajectory. The experimental environment is a corridor surrounded by walls and windows. The algorithm was run in realtime on industrial computer installed inside the AGV. In the experiment, the sensor outputs are observed at 10Hz. The fault data and any other information are saved inside industrial computer during experiment and can be copied after experiment is finished. The proposed algorithm is tested during several conditions T1, T2, T3, and T4. During the experiment, the trajectories from each test such as controller + laser scanner (LMS), Encoder + laser scanner (LMS), Encoder + controller, and controller + NAV 200 can be observed. The residue of each test can also be obtained and it is compared with the threshold value Th . In this experiment, the threshold value Th is chosen as 500.

7.1. Normal condition

The experimental results for normal condition are shown in Fig. 8 and Fig. 9. Fig. 8 shows that trajectory tracking results in several conditions T1, T2, T3, and T4 obtained from each positioning module fused by EKF are almost similar since fault condition doesn't exist. Therefore, the residue values shown in Fig. 9 are less than the threshold value Th .

Table 3. Parameters and initial values.

Parameter	Value	Parameter	Value
r	0.07 m	(k_1, k_2, k_3)	(0.5, 20, 0.1)
b	0.60 m	$(X_C, Y_C, \theta_C)_{t=0}$	(0m, 1.5m, 0°)
k_r, k_l	(0.02, 0.02)	k_x, k_y, k_θ	(0.01, 0.01, 0.001)

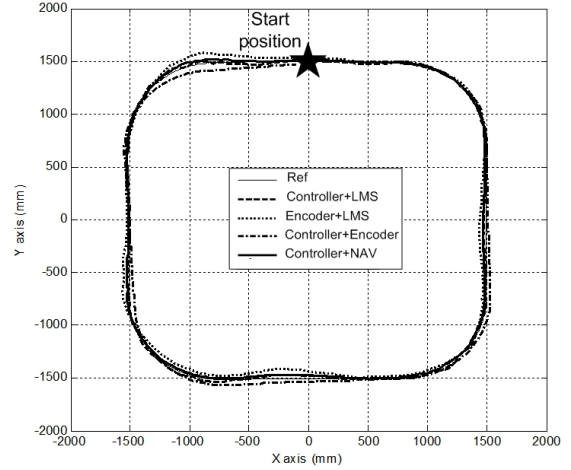


Fig. 8. Trajectory tracking in normal condition.

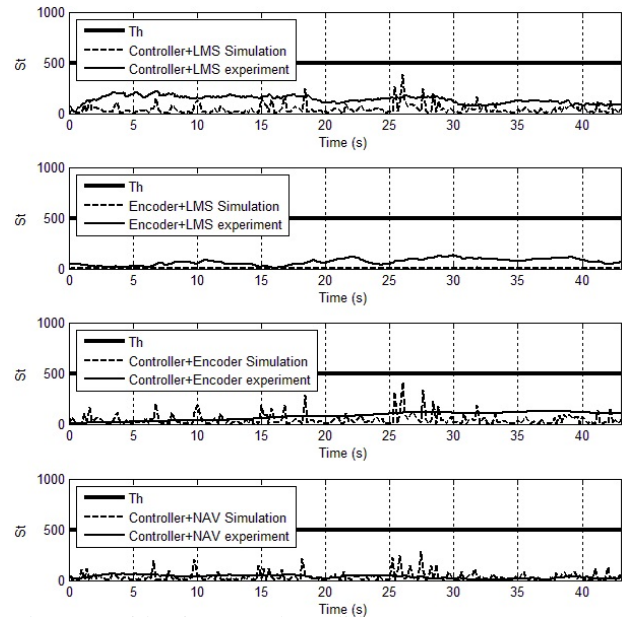


Fig. 9. Residue in normal condition.

7.2. Encoder fault

The experimental results for the encoder fault are shown in Fig. 10 and Fig. 11. To resemble the encoder fault condition, at $t = 20$ s, the controller stops receiving the data from encoders. Therefore, the encoder positioning modules doesn't provide the positioning data anymore. Fig. 10 shows the trajectory tracking when the encoder fault happens. In Fig. 11, the residue values of encoder + laser scanner (LMS) and controller + encoder are increased over the threshold value at $t = 21.6$ s. From the Table 2, it can easily be observed that the encoder fault happens.

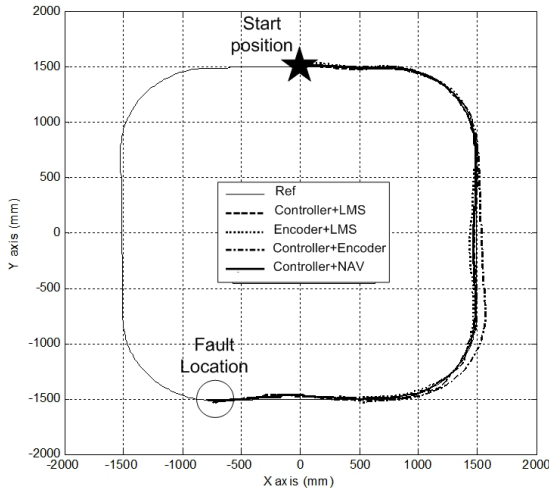


Fig. 10. Trajectory tracking at encoder fault condition.

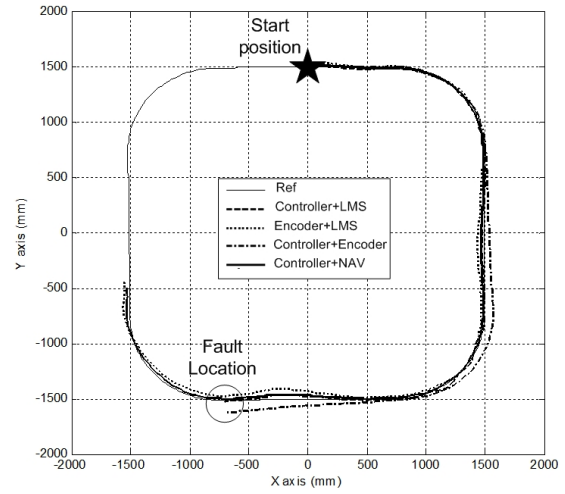


Fig. 12. Trajectory tracking at BLDC motor fault.

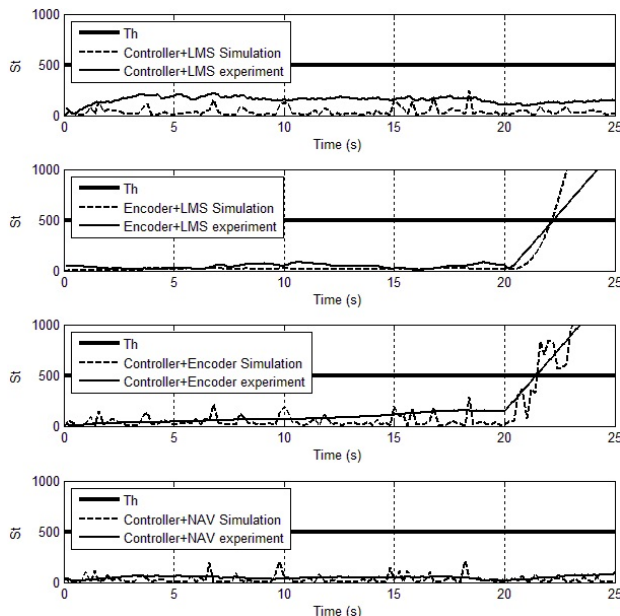


Fig. 11. Residue at encoder fault condition.

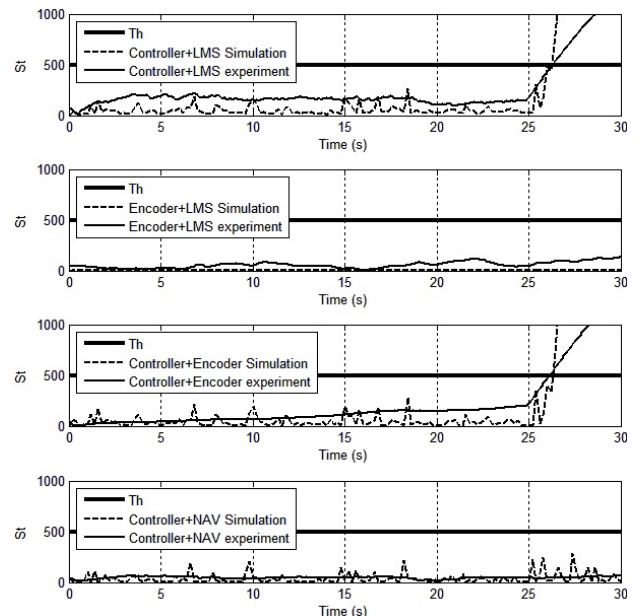


Fig. 13. Residue at BLDC motor fault.

7.3. BLDC motor fault

The experimental results for the BLDC motor fault are shown in Fig. 12 and Fig. 13. The fault condition can be obtained by turning off the motors when the AGV moves. Fig. 13 shows that since the motor stops at $t = 25$ s, the residue values of controller + laser scanner (LMS) and controller + encoder are increased over the threshold value at $t = 26.2$ s. From the Table 2, it can easily be observed that the BLDC motor fault happens.

7.4. Laser scanner fault

The experimental results for the laser scanner (LMS) fault are shown in Fig. 14 and Fig. 15. The fault condition can be obtained by pulling the sensor's power connection

when the AGV moves at $t = 30$ s.

Since the laser scanner doesn't provide any data, the SLAM positioning cannot provide new position data. Therefore, as shown in Fig. 15, the residue values of controller + laser scanner (LMS) and Encoder + laser scanner (LMS) are over the threshold value at $t = 31.2$ s.

7.5. NAV fault

The experimental results for NAV fault are shown in Fig. 16 and Fig. 17. The fault condition can be obtained by pulling of the power connection of the sensor when the AGV moves at $t = 35$ s. Since the NAV sensor doesn't provide any data, as shown in Fig. 17, the residue value of controller + NAV is increased and over the threshold value

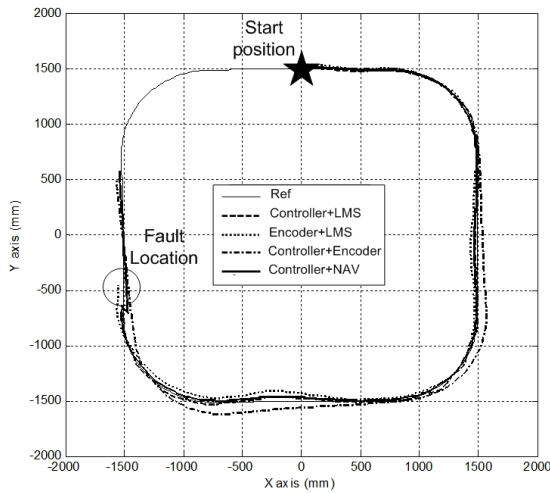


Fig. 14. Trajectory tracking at laser scanner fault.

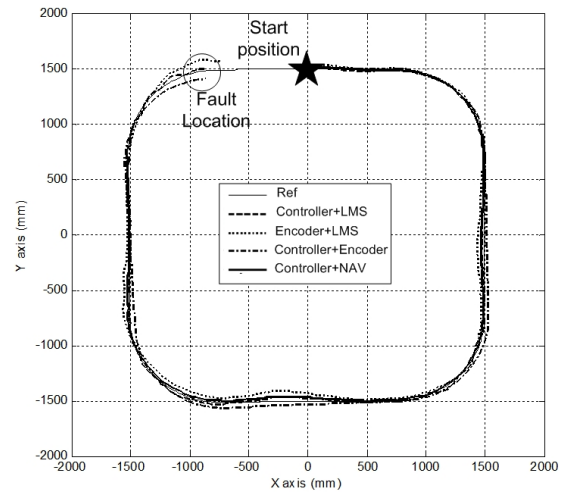


Fig. 16. Trajectory tracking at NAV-200 fault.

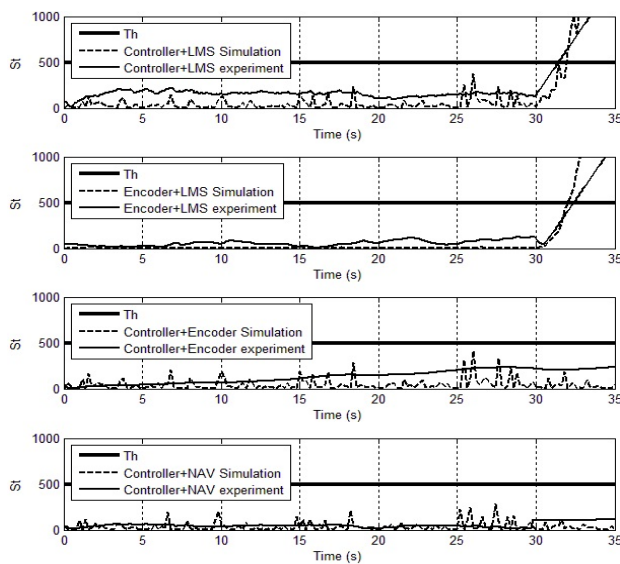


Fig. 15. Residue when laser scanner fault.

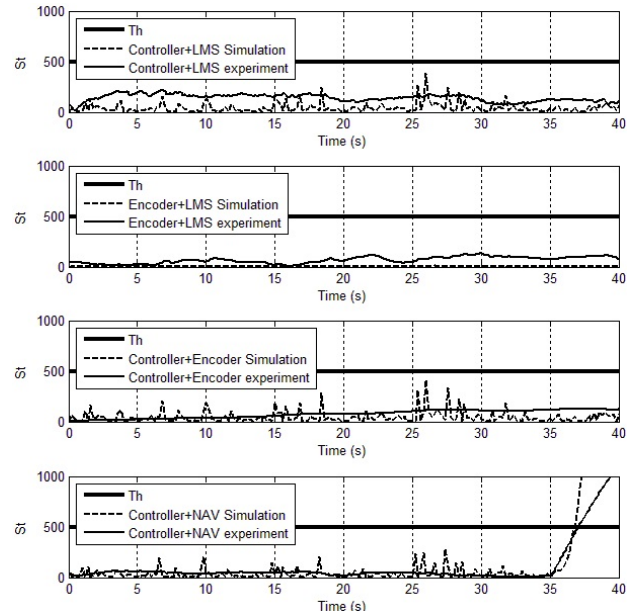


Fig. 17. Residue at NAV-200 fault.

at $t = 36.1$ s.

8. CONCLUSION

This paper proposed an implementation and experimental validation of fault detection algorithm for the Automatic Guided Vehicle (AGV) system based on multiple positioning modules using EKF. The experimental results in normal condition and fault condition showed that the proposed algorithm successfully detected the fault conditions with the computational time of 125 ms. The algorithm calculated the residue of each pairs positioning modules based on EKF. Depending on residue values, the fault conditions were detected. In the future, this system

can be improved so that it can detect more than one fault at a time.

REFERENCES

- [1] F. Baghernezhad and K. Khorasani, "A robust fault detection scheme with an application to mobile robots by using adaptive thresholds generated with locally linear models," *Proc. of IEEE Conf. on Computational Intelligence in Control and Automation (CICA)*, pp. 9-16, 2013. [click]
- [2] E. N. Skundrianos and S. G. Tzafestas, "Fault diagnosis on the wheels of a mobile robot using local model neural networks," *IEEE Robotics and Automation Magazine*, pp. 83-90, September 2004. [click]

- [3] W. Y. Jeong, P. S. Pratama, B. H. Jun, and S. B. Kim, "Fault detection for underwater hexapod robot based on PI observer," *Proc. of Conf. Korea Marine Science and Technology*, pp. 1772-1777, 2013.
- [4] J. H. Min, W. Y. Jeong, P. S. Pratama, Y. D. Setiawan, H. K. Kim, and S. B. Kim, "Fault diagnosis system of caterpillar wheel type pipeline inspection robot," *Proc. of International symposium on Mechatronics and Robotics*, pp.74-78, 2013.
- [5] X. J. Li and G. H. Yang, "Adaptive fault detection and isolation approach for actuator stuck faults in closed-loop system," *International Journal of Control, Automation and Systems*, vol. 10, no. 4, pp. 830-834, August 2012. [click]
- [6] T. Li, L. Y. Wu, and X. J. Wei, "Robust fault detection filter design for uncertain LTI systems based on new bounded real lemma," *International Journal of Control, Automation and Systems*, vol. 7, no. 4, pp. 644-650, August 2009. [click]
- [7] S. C. Jee, H. J. Lee, and Y. H. Joo, " H_2/H_∞ sensor fault detection and isolation in linear time-invariant systems," *International Journal of Control, Automation and Systems*, vol. 10, no. 4, pp. 830-834, August 2012.
- [8] Z. Q. Zheng, Y. X. Liu, H. W. Yuan, and Q. S. Wang, "Distributed fault detection and isolation scheme for abrupt and incipient faults in a class of nonlinear systems," *International Journal of Control, Automation and Systems*, vol. 10, no. 3, pp. 623-631, June 2012. [click]
- [9] M. Sami and R. J. Patton, "Active fault tolerant control for nonlinear systems with simultaneous actuator and sensor faults," *International Journal of Control, Automation and Systems*, vol. 11, no. 6, pp. 1149-1161, December 2013. [click]
- [10] Z. X. Cai, Z. H. Duan, J. F. Cai, X. B. Zou, and J. X. Yu, "A multiple particle filters method for fault diagnosis of mobile robot dead-reckoning system," *Proc. of IEEE Conf. on Intelligent Robots and Systems*, pp. 481-486, 2005. [click]
- [11] Z. H. Duan, Z. X. Cai, and J. X. Yu, "Adaptive particle filter for unknown fault detection of wheeled mobile robots," *Proc. of IEEE Conf. on Intelligent Robot and System*, pp. 1312-1315, 2006. [click]
- [12] A. Monteriu, P. Asthana, K. Valavanis, and S. Longhi, "Experimental validation of a real-time model-based sensor fault detection and isolation system for unmanned ground vehicles," *Proc. of IEEE Conf. on Control and Automation*, pp. 1-8, 2006. [click]
- [13] P. S. Pratama, T. L. Bui, J. G. Kim, and S. B. Kim, "Trajectory tracking of automatic guided vehicle based on simultaneous localization and mapping method," *Proc. of Conf. Korean Society for Logistic Science and Technology*, 2013.
- [14] H. H. Lin and C. C. Tsai, "Laser pose estimation and tracking using fuzzy extended information filtering for an autonomous mobile robot," *Journal of Intelligent Robotic System*, vol. 53, no. 2, pp. 119-143, 2008. [click]
- [15] A. J. Krener, "The convergence of the extended Kalman filter," *Lecture Notes in Control and Information Sciences*, vol. 286, 2003, pp. 173-182. [click]
- [16] P. C. Mahalanobis, "On the generalised distance in statistics," *Proc. of the National Institute of Sciences of India*, pp. 49-55, 1936.
- [17] C. Hajiyev and F. Caliska, *Fault Diagnosis and Reconfiguration in Flight Control Systems*, Kluwer Academic Publishers, Springer, 2003. [click]



Pandu Sandi Pratama was born in Indonesia on November 1, 1986. He received the B.S. degree in Electrical Engineering Dept. of Diponegoro University, Indonesia in 2011. He received the M.S. degree from the Interdisciplinary Program of Mechatronics Engineering Dept., Pukyong National University, Busan, South Korea in 2013. He received his Ph.D. degree from the Dept. of Mechanical Engineering, Pukyong National University, Busan, Korea in 2015. His research fields of interest are computer science, robotic and mobile robot.



Amruta Vinod Gulalkari was born in India on November, 1988. She received the B.S. degree from Dept. of Electronics and Telecommunication, SSGB Amravati University, Amravati, India in 2010. She then received the M.S. degree from the Dept. of Interdisciplinary Program of Mechatronics Engineering, Pukyong National University, Busan, Korea in 2015. Her research fields of interest are legged robots, mobile robot control and image processing.



Yuhanes Dedy Setiawan was born in Indonesia on December 28, 1989. He received the B.S. degree in Mechanical Engineering from Diponegoro University, Indonesia, in 2012. He is received his Master degree from the Dept. of Mechanical Design Engineering in Pukyong National University, Busan, Korea in 2014. His research fields of interest are nonlinear control, adaptive control, path planning algorithm, AGV control and SLAM.



Dae Hwan Kim was born in Korea on March, 1982. He received the B.S. degree in Electrical Engineering from Chosun University, Kwangju, Korea in 2008. He received the M.S. and Ph.D. degrees in Mechanical engineering from the Pukyong National University, Busan, Korea, in 2009 and 2015, respectively. His fields of interests are robust control, combustion engineering control, and mobile robot control.



Hak Kyeong Kim was born in Korea on November 11, 1958. He received the B.S. and M.S. degrees from Dept. of Mechanical Engineering, Pusan National University, Korea in 1983 and 1985. He received his Ph.D. degree from the Dept. of Mechatronics Engineering, Pukyong National University, Busan, Korea in February, 2002. His fields of interest are robust

control, biomechanical control, mobile robot control, and image processing control.



Sang Bong Kim was born in Korea on August 6, 1955. He received the B.S. and M.S. degrees from National Fisheries University of Busan, Korea, in 1978 and 1980. He received his Ph.D. degree from Tokyo Institute of Technology, Japan in 1988. After then, he is a Professor of the Dept. of Mechanical Engineering, Pukyong National University, Busan, Korea. His re-

search has been on robust control, biomechanical control, and mobile robot control.