Thesis for the Degree of Master of Engineering

Object Following Controller Design for a Four Wheel Independent Steering Automatic Guided Vehicle Using Kinect Camera Sensor

by

Ding Xing Kun

Department of Interdisciplinary Program of

Mechatronics Engineering, The Graduate School,

Pukyong National University

February 2016

Object Following Controller Design for a Four Wheel Independent Steering Automatic Guided Vehicle Using Kinect Camera Sensor

키넥트 카메라 센서를 사용한4 륜 독립조정 무인차량용 물체추종
제어기 설계

by

Ding Xing Kun

Advisor: Professor Sang Bong Kim

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Engineering

**In the Department of Interdisciplinary Program of
Mechatronics Engineering,
The Graduate School,
Pukyong National University**

February 2016

Object Following Controller Design for a Four Wheel Independent Steering Automatic Guided Vehicle Using Kinect Camera Sensor

A thesis

by

Ding Xing Kun

Approved as to styles and contents by:

(Chairman) **Yeon Wook Choe**

Byun

(Member) **Gi Sik Byun**

김상봉

(Member) **Sang Bong Kim**

December, 2015

Acknowledgements

First of all, I would like to give my highest praise to God for all helps and guidance that have been given such that I could finish all my research and thesis in Master program very well. All achievement and success that have been granted to me only I would like to return to the God who owns them. In this right moment of time, I would like to express my deep gratitude to all people who helped me during my study and meaningful life in Republic of Korea.

Firstly, I would like to express my deep gratitude to my professor, Professor Sang Bong Kim, who has helped me significantly in completing my Master program. I really appreciate his kindness and helps that he has shown to me during my study in his excellent laboratory.

I would also like to express my sincere gratitude to the members of my thesis committee, Professor Yeon Wook Choe, Professor Gi Sik Byun and Professor Sang Bong Kim for their helpful comments and suggestions for my thesis.

Furthermore, I would like to express my profound gratitude to Professor Hak Kyeong Kim for his great help and advice such that I could finish my research and my master thesis very well.

I would also like to express my gratitude to all members of Computer Integrated Manufacturing Electronics Commerce (CIMEC) laboratory for their cooperation and for all the kindness and friendship, Dr. Pandu Sandi Pratama, Dr. Phuc Thinh Doan, Mr. Van Tu Duong, Mr. Nguyen Tron Hai, Mr. Huy Hung Nguyen as Doctor course students, Ms. Amruta Vinod Gulalkari as a master of engineering, Mr. Jin Wook Kim, Mr. Husam Hasan Khudhir El-dulaimi, Mr. Sheng Dongbo, Mr. Jae Hoon Jeong, Mr. Tian Shui Gao as Master course students, Mr. Jin Il Kim and Mr. Jeong Woong Lee as Undergraduate students.

Last but not least, I would like to express my deepest gratitude to my father, Jiahua Ding, my mother, Xiaojiao Yu, my old sister, Zhibo Ding, my old brother, Xing Ding and all my relatives for their love, endless prayers, encouragement and support for me not only in completing Master program but also in the whole of my life.

Pukyong National University, Busan, Korea

December, 2015

Ding Xing Kun

Contents

Object Following Controller Design for a Four Wheel Independent Steering Automatic Guided Vehicle Using Kinect Camera Sensor

Ding Xing Kun

Department of Interdisciplinary Program of Mechatronics Engineering, The Graduate School,
Pukyong National University

Abstract

This thesis presents a development of a vision-based object following four wheel independent steering automatic guided vehicle (4WIS-AGV) system using Kinect camera sensor. To do this task, the following steps are executed. Firstly, a 4WIS-AGV is designed and developed for experimental purpose. Each wheel has one driving DC motor and one steering DC motor. As a result, it can move in any direction in its environment. Secondly, a system kinematic modeling is described to understand the behavior of the 4WIS-AGV system and a moving candidate object. Thirdly, a candidate blue colored object is detected using Kinect camera sensor. A color-based object detection algorithm is used to obtain the center position coordinate of the moving candidate object inside the Kinect RGB data. Fourthly, the laser navigation system NAV-200 is used to obtain the global center position of the detected candidate object and the 4WIS-AGV. Industrial PC TANK-800 is used as the main controller for all the calculation of the controller and collecting data from positioning sensors. Fifthly, a Kalman filter algorithm is used to estimate the exact global center position coordinate of the detected moving object. Sixthly, a backstepping controller based on Lyapunov stability theory is designed for the 4WIS-AGV with Kinect camera sensor and laser navigation system NAV-200 to achieve an object following task with keeping a given constant safe distance between the moving candidate object and the 4WIS-AGV. Finally, the simulation and experimental are conducted to verify the effectiveness and performance of the proposed control algorithm for following the moving candidate object in both rotational and straight followings of the 4WIS-AGV. The results show that the proposed controller makes the 4WIS-AGV to follow the detected moving object well with keeping a given constant safe distance from the moving candidate object within an acceptable error.

Keywords: Four Wheel Independent Steering Automatic Guided Vehicle, Kinect Camera Sensor, Laser Navigation System NAV-200, Kalman Filter, Backstepping Method, Object Following.

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Background and motivation

Robotics has achieved its greatest success to manufacture industrial producs. When a robot arm is installed at its shoulder, it can move to a specific position in the assembly line with fast speed and high accuracy to perform repetitive tasks such as spot painting and welding as shown in Figs. 1.1 and 1.2, respectively.

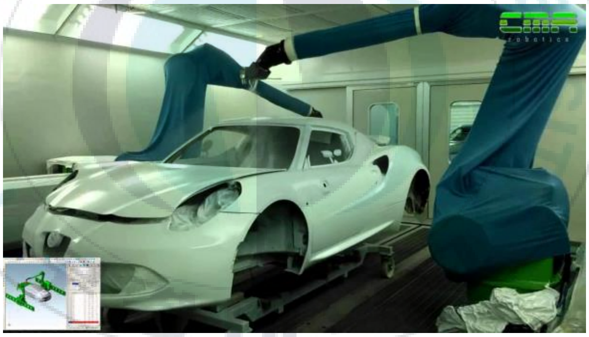


Fig. 1.1 Painting

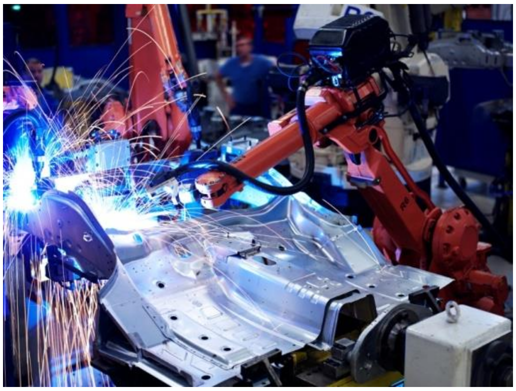
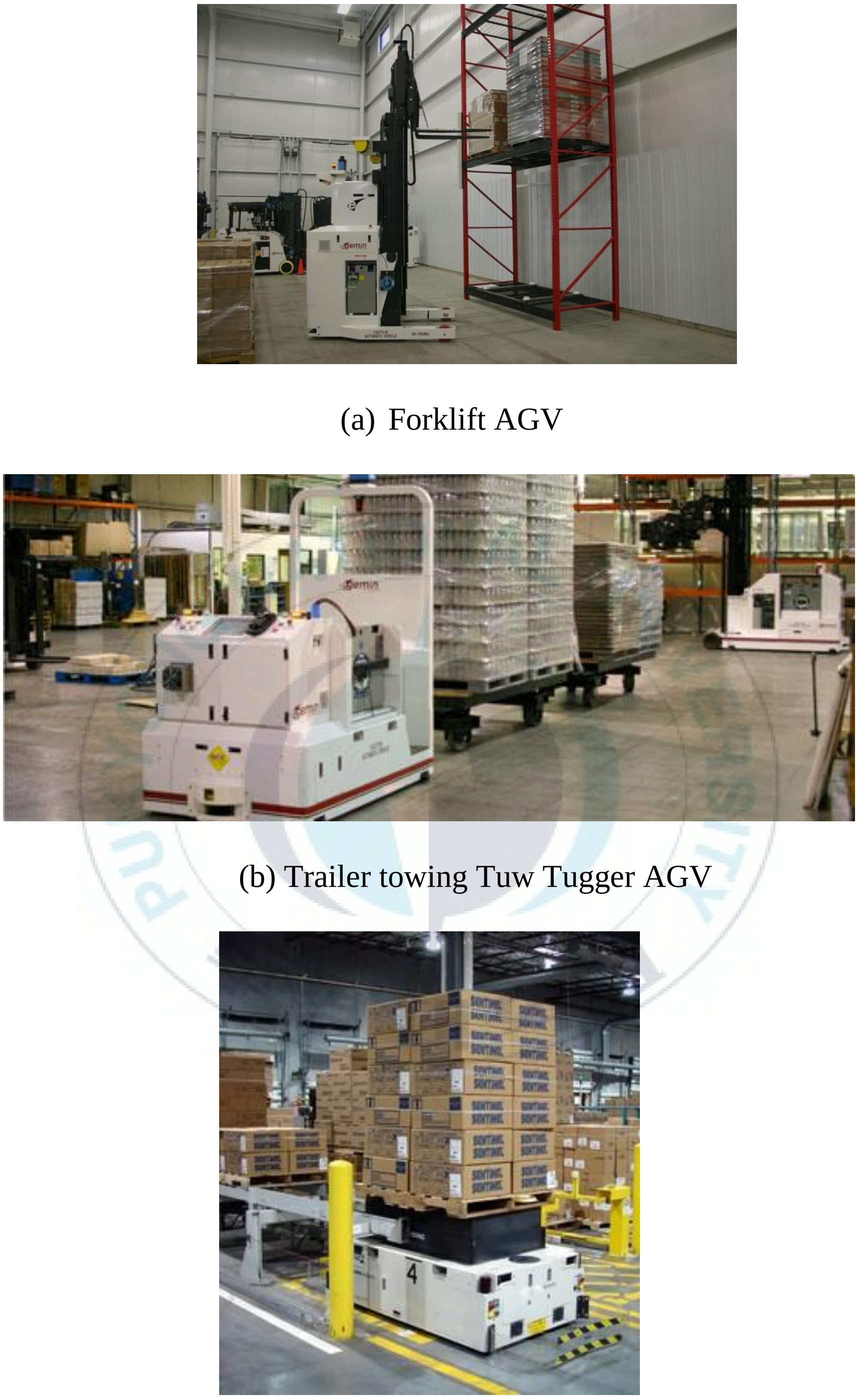


Fig. 1.2 Welding

However, these commercial robots suffer from a fundamental disadvantage: lack of mobility. In contrast, an Automated Guided Vehicle (AGV) can move to anywhere of the manufacturing plant. It is a transportation vehicle automatically traveling on a manufacturing facility or a warehouse.

AGV has a great contribution for the modern industry such as reduction in the number of worker, improvement of productivity and quality, improvement of working environment and safety, less damage on transporting goods, real-time control of material flow, improved management on product and easy communication with other automatic devices such as automatic door and elevator.

Fig. 1.3 shows some examples of industrial AGV. Fig. 1.3(a) shows Forklift AGV picking loads from rack. Fig. 1.3(b) shows Tugger AGV towing multiple trailers in a warehouse environment. Fig. 1.3(c) shows Unit Load Carriers AGV called as a “top carrier” for cargo handling.



(a) Forklift AGV

(b) Trailer towing Tuw Tugger AGV

(c) Unit Load Carriers AGV

Fig. 1.3 Examples of industrial AGV

Automatic Guided Vehicles (AGVs) have been operated in most industries due to their high efficiency for material handling [1-8]. Therefore, many AGV researches have been conducted for development of laser [9-15], development of vision [16-21], localization [22-31] and navigation [32-44] for the AGV. One of the techniques to improve the performance of AGVs is to improve AGVs' maneuverability and flexibility because the higher characteristics such as maneuverability and flexibility are, the better the AGVs' performance is. Several kinds of wheel configurations for AGVs have been developed. Pratama et al. [45] developed a differential drive AGV that could work autonomously using laser navigation measurement system. Obstacle avoidance algorithm was also applied in this system. The disadvantage of this system was that the AGV had only two driving wheels without independent steering. When the vehicle wants to change its direction, the vehicle orientation must be changed by controlling velocities of two differential driving wheels. On the other hand, Doan et al. [46] developed a tricycle wheeled AGV. In this vehicle, they proposed a path tracking control using camera sensor. The disadvantage of the system was that such wheeled configurations had their less flexibility and limited maneuverability. To get higher maneuverability and flexibility, four wheel independent steering configuration has been proposed for development of AGVs, hereafter referred to as 4WIS-AGVs.

Four wheel independent steering configuration is a wheel configuration that each wheel orientation can be adjusted separately. Because of this reason, 4WIS-AGVs can move to any direction in their work environment. The main advantage of this wheel configuration has small vehicle sideslip angle and better dynamic response characteristics [47]. Nevertheless, controlling the configurations of four wheels independently to make the 4WIS-AGV move to certain directions is challenging because the orientation of each wheel has to be adjusted with particular formation such that the vehicle can move in the expected direction.

In the past few decades, object following has been one of the interesting research topics in the field of autonomous robotics. A mobile robot equipped with camera sensor can analyze its situation well to perform an object following task. There have been numerous researches carried out related to object detection and following using various camera sensors as follows [48]. The problem for mono-vision camera was to have the complexity of extracting necessary depth and color information from the received image data. The problem for time-of-flight cameras was an inefficient and complex camera [49].

The problem for stereo cameras was to increase the hardware complexity of the system because of the system with more than one camera using stereo vision method. In some research works, cameras were installed on different kinds of robotic platforms to do an object following task. G. Xing proposed people-following system design for mobile robots using Kinect camera sensor [51]. The disadvantage was that the camera control law was too complex. N. Mir-Nasiri proposed a camera-based 3D object tracking and following mobile robot [52]. The disadvantage was that it was complex and easy to lose object because the camera performs tracking of moving objects by changing the orientation of camera eyes towards the object.

Moreover, the selection of an appropriate controller is also an important issue in such kind of systems. There were several control algorithms available for four wheel independent steering vehicles such as an optimal control [53], multi-mode control method based on Fuzzy selector [54], nonlinear decoupling control with an observer [55], etc. However, their disadvantage was that it was very difficult to find their control law and their controllers were not stable enough because their simulation and experiment results had the big errors and took long time go to zero. Thus, a better controller is necessary for 4WIS-AGVs.

1.2 Problem statements

In order to ensure that the 4WIS-AGV system follows a moving candidate object keeping the reference distance (1.25 m) between the object and the 4WIS-AGV with an acceptable small error using Kinect camera sensor and laser navigation system NAV-200, a new controller must be developed.

The problem statements to design such a controller are described as follows:

- To design and develop a 4WIS-AGV system suitable for experiment.
- To introduce modeling for a 4WIS-AGV system.
- To measure the 4WIS-AGV position using laser navigation system NAV-200 and to get the position of a moving object by image processing method from its images obtained from the Kinect camera sensor, respectively.
- To get position and velocity information of the target moving candidate object more exactly.
- To design a controller to track a reference point with keeping a constant safe distance between the moving candidate object and 4WIS-AGV.

- To perform the simulation and experiment for showing the effectiveness and the applicability of the proposed controller.

1.3 Objective and research method

To solve the problems mentioned in section 1.2, this thesis proposes an object following controller design for a four wheel independent steering automatic guided vehicle (4WIS-AGV) by using a Kinect camera sensor to follow a moving candidate object with keeping a constant safe distance. To do this task, a system description and modeling, an image processing algorithm, a tracking algorithm, a control algorithm design, and simulation and experiment to evaluate the controller are presented as follows:

Firstly, a 4WIS-AGV is designed and developed for the experimental purpose. The 4WIS-AGV has its body and four independent steering wheels. The 4WIS-AGV can move in any direction in its environment by the 4 independent steering wheels. Therefore, a wheel configuration has been chosen for this thesis. Each wheel of the developed 4WIS-AGV has one driving DC motor and one steering DC motor. A Kinect camera sensor is installed on the top front of the robot for image processing. This low cost RGB-D camera provides depth data directly and thus, reduces the complexity of the image processing algorithm. The 4WIS-AGV is controlled by the industrial PC Tank 800 and the microcontroller AVR ATmega128. The microcontroller AVR ATmega128 is used to control the driving DC motor and the steering DC motor of wheel configuration read encoders and receives feedback data from DC motors. The industrial PC Tank 800 calculates the image processing algorithm and the control algorithm, and sends the velocity and rotational angle signals to the microcontroller. A Graphical User Interface (GUI) is designed using Visual studio C# software to send control commands to the 4WIS-AGV and to display the results to users.

Secondly, a system kinematic modeling is described to understand the behavior of the 4WIS-AGV system and a moving candidate object as follows. For the kinematic modeling of the 4WIS-AGV, control inputs such as linear velocities and angular velocity in local coordinate are defined for the object following control. And a candidate blue colored object is selected to do the object following experiment.

Thirdly, a color-based object detection algorithm is developed using AForge.NET C# framework to detect the position coordinate of a blue colored candidate object. This image processing algorithm is applied to RGB image frames obtained from the Kinect camera sensor. Primarily, the pixel coordinate of the center of the candidate object inside the RGB image frames is obtained. Then, using simple trigonometry and Kinect depth information, a real world coordinate of the candidate object is obtained in mm. Kalman filter algorithm based on the object motion model is adopted to estimate the position and velocity of the moving candidate object. The Kalman filter is a recursive estimator that estimates the state from the previous time step and the current measurement efficiently.

Fourthly, a controller for the 4WIS-AGV to follow a moving candidate object is designed based on the backstepping method using Lyapunov function. System stability is checked based on Lyapunov stability theory.

Finally, simulations and experiments are conducted to verify the effectiveness and performance of the proposed control algorithm for following the moving candidate object in both rotational and straight followings of the 4WIS-AGV. The results show that the proposed controller makes the 4WIS-AGV follow the moving candidate object very well with keeping a given constant safe distance from the moving candidate object within an acceptable error.

1.4 Outline of thesis and summary of contributions

Chapter 1: Introduction

In this chapter, firstly, the background and the motivation of this thesis are addressed. Secondly, the problem statements of the thesis are described. Thirdly, the objectives and research method of this thesis are described. Finally, an outline of the thesis and its contributions are described.

Chapter 2: System Description and Modeling

This chapter, firstly, describes the basic terminologies and equations for a four wheel steering vehicle. Secondly, the four wheel steering maneuvers and zero-sideslip maneuver are described. Finally, the kinematic modeling of the 4WIS-AGV system is described.

Chapter 3: Hardware Structure of the 4WIS-AGV

This chapter describes the hardware structure of the 4WIS-AGV used to implement the designed controller. The hardware structure of the 4WIS-AGV consists of mechanical design and electrical design such as DC motors, encoders, industrial PC Tank 800, battery, microcontroller AVR ATmega128, motor drivers, monitor, Graphical Unit Interface (GUI) software and Kinect camera sensor of the 4WIS-AGV.

Chapter 4: 4WIS-AGV and Object Position Measurement

This chapter describes laser navigation system NAV-200 basic principle to measure the position of 4WIS-AGV, and Kinect camera sensor basic principle to measure the real position of a moving candidate object.

Chapter 5: Object Tracking Using Kalman Filter

This chapter describes the Kalman filter to estimate and track a moving candidate object exactly. Since the Kinect camera sensor has some noise and sometimes it detects unwanted objects due to the different real-time condition, it is difficult to keep tracking the position coordinates of the moving candidate objects. And the Kalman filter to estimate and track the moving candidate object well is described.

Chapter 6: Controller Design

This chapter describes a controller design for the 4WIS-AGV to makes the 4WIS-AGV follow a moving candidate object with keeping a given constant safe distance between the moving candidate object and the 4WIS-AGV using Backstepping method based on the Lyapunov stability theory.

Chapter 7: Simulation and Experimental Results

Simulation and experimental results are shown for the 4WIS-AGV with Kinect camera sensor and laser navigation system NAV-

200 to follow the moving candidate object successfully with keeping a given constant safe distance between the moving candidate object and the 4WIS-AGV using the proposed controller within an acceptable small error.

Chapter 8: Conclusions and Future Works

Conclusions of this thesis and several suggestions for future works are presented.

Chapter 2: System Description and Modeling

2.1 Basic terminologies and equations

The schematic of four wheel steering vehicles is shown in Fig. 2.1.

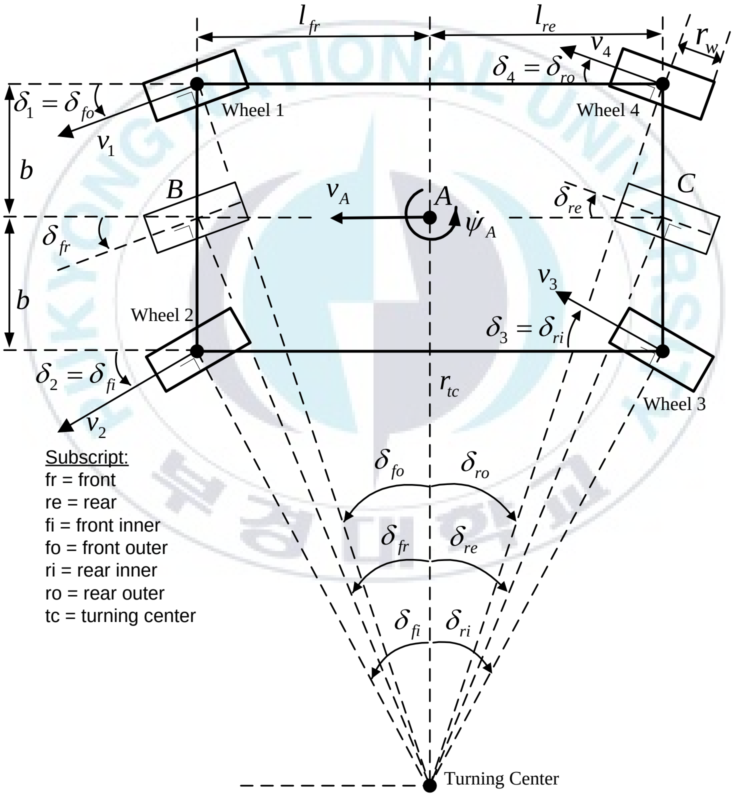


Fig. 2.1 Schematic of the four wheel steering vehicles

where

δ_{fi} , δ_{fo} = Front inner and front outer steering angles (rad), respectively

δ_{ri}, δ_{ro} = Rear inner and rear outer steering angle (rad), respectively

δ_{fr}, δ_{re} = Front and rear steering angles (rad), respectively

v_A = Vehicle linear velocity of the 4WIS-AGV (m/s)

v_i = Each wheel linear velocity, for $i = 1, \cdots, 4$ (m/s)

$\dot{\psi}_A$ = Vehicle angular velocity of the 4WIS-AGV (rad/s)

l_{fr} , l_{re} = Longitudinal distances from the center of gravity of the vehicle to the front and rear wheels (m), respectively

r_{tc} = Turning center radius of the 4WIS-AGV (m)

b = Distance from vehicle center of gravity (CG) to right or left wheel (m)

r_w = Wheel radius (m)

Four wheel steering vehicles have four wheels that are adjusted separately. Thus, each wheel has each wheel steering angle δ_i, each wheel linear velocity v_i, and wheel angular velocity ω_i ($v_i = \omega_i r_w$). The wheel names are determined by defining the wheel when the vehicle moves in curved trajectories as shown in Fig. 2.1. In this condition, there are two wheels which are nearer to the turning center and therefore defined as the inner wheels. The rest of the wheels which are further to the turning center are called outer wheels. The wheel numbers are started from the front outer wheel as wheel 1, going to the front inner wheel as wheel 2, then going to the rear inner wheel as wheel 3 and the last wheel is the rear outer wheel as wheel 4.

Subsequently, angles of each wheel can be calculated as follows [56, 58]:

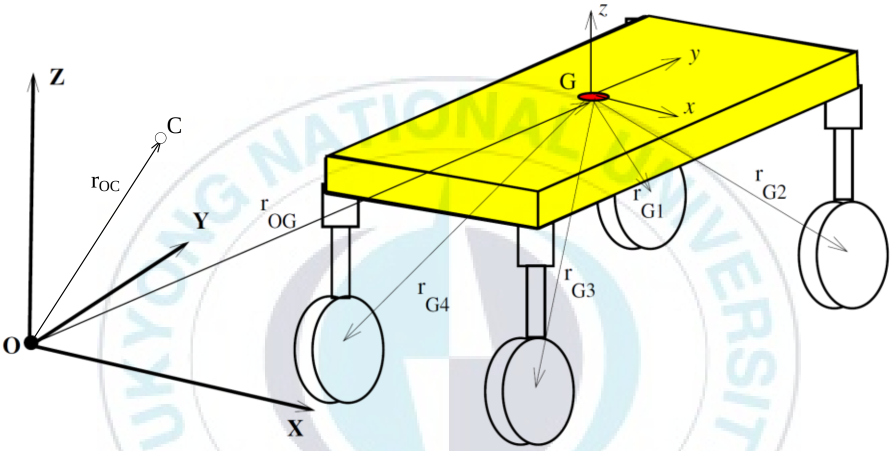


Fig. 2.2 Configuration of the coordinate system for a robotic vehicle

The velocities at the wheel centers in Fig. 2.2 and Fig. 2.3 can be expressed as [56, 58]

$$\vec{V}_i = \vec{\Omega} \times \left[\vec{r}_{OC} - \left(\vec{r}_{OG} + \vec{r}_{Gi} \right) \right], \; i = 1, \cdots, 4 \qquad (2.1)$$

where

$\vec{r}_{OG}$: the relative position of the vehicle’s center of mass in the XYZ inertial frame from its origin,

$\vec{r}_{OC}$: the single point C in the inertial frame serving as its instantaneous center of zero velocity,

$\vec{r}_{Gi}$: the position vector of wheel i from the vehicle’s mass center

$\vec{\Omega}$: the angular velocity vector of the frame xyz in frame XYZ

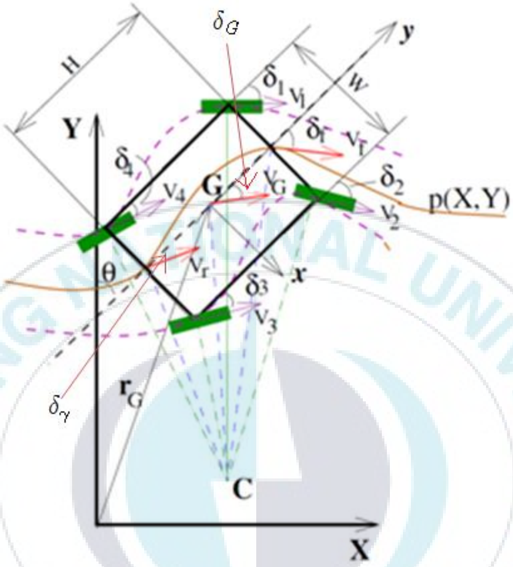


Fig. 2.3 Typical wheel configuration for a four wheel independent steering vehicle on a flat surface

From Fig. 2.3, if v_f , v_G and v_r respectively are the front, center and rear linear velocities of the vehicle along the longitudinal centerline with their respective steering angles δ_f , δ_G and δ_r then kinematic and rigid body constraints on the centerline satisfy $v_f \cos\delta_f = v_G \cos\delta_G = v_r \cos\delta_r$. And at the front and real axles, these constraints satisfy $v_1 \sin\delta_1 = v_f \sin\delta_f = v_2 \sin\delta_2$ and $v_3 \sin\delta_3 =$ $v_r \sin\delta_r = v_4 \sin\delta_4$. Therefore, the triangular relationships of these equations are shown in Fig. 2.4.

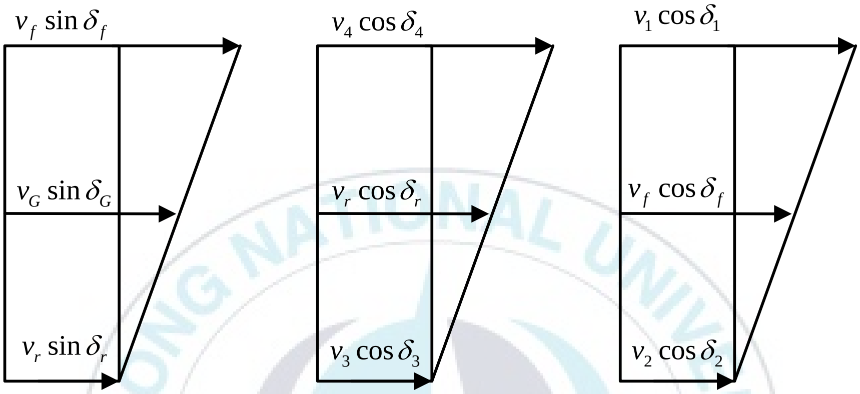


Fig. 2.4 Triangular relationships

From the triangular relationships shown in Fig. 2.4, the following equations can be obtained.

$$v_r \sin\delta_r = \frac{v_f \sin\delta_f - v_r \sin\delta_r}{2} + v_G \sin\delta_G \tag{2.2}$$

$$v_f \cos\delta_f = \frac{v_1 \cos\delta_1 - v_2 \cos\delta_2}{2} + v_2 \cos\delta_2 \tag{2.3}$$

$$v_r \cos\delta_r = \frac{v_4 \cos\delta_4 - v_3 \cos\delta_3}{2} + v_3 \cos\delta_3 \tag{2.4}$$

Then by some simplification, these equations become as follows:

$$\nu_f \sin\delta_f - \nu_r \sin\delta_r = 2\left(\nu_G \sin\delta_G - \nu_r \sin\delta_r\right) = H\omega \quad (2.5)$$

$$\nu_1 \cos\delta_1 - \nu_2 \cos\delta_2 = 2\left(\nu_f \cos\delta_f - \nu_2 \cos\delta_2\right) = W\omega \quad (2.6)$$

$$\nu_4 \cos\delta_4 - \nu_3 \cos\delta_3 = 2\left(\nu_r \cos\delta_r - \nu_3 \cos\delta_3\right) = W\omega \quad (2.7)$$

where $\omega = \dot{\theta}$.

If all angles $\delta_i, i = f, G, r, 1, \cdots, 4$ are such that $\left|\delta_i\right| < \dfrac{\pi}{2}$, then by combining Eqs. (2.5) - (2.7) with the instantaneous center, C, Eq. (2.1) leads to the wheel linear velocities as follows:

$$\nu_i = \begin{cases} \dfrac{\nu_A \tan\delta_{fr} \csc\delta_i}{\sqrt{1 + \dfrac{1}{4}\left(\tan\delta_{fr} + \tan\delta_{re}\right)^2}}, i = 1, 2; \\ \dfrac{\nu_A \tan\delta_{re} \csc\delta_i}{\sqrt{1 + \dfrac{1}{4}\left(\tan\delta_{fr} + \tan\delta_{re}\right)^2}}, i = 3, 4; \end{cases} \quad (2.8)$$

and wheel steering angles are

$$\delta_1 = \delta_{fo} = \cot^{-1}\left(\cot\delta_{fr} + \frac{2b}{2(l_{fr} + l_{re})}\cot\delta_{fr}\left[\tan\delta_{fr} - \tan\delta_{re}\right]\right) \quad (2.9)$$

$$\delta_2 = \delta_{fi} = \cot^{-1}\left(\cot\delta_{fr} - \frac{2b}{2(l_{fr} + l_{re})}\cot\delta_{fr}\left[\tan\delta_{fr} - \tan\delta_{re}\right]\right) \quad (2.10)$$

$$\delta_3 = \delta_{ri} = \cot^{-1}\left(\cot\delta_{re} - \frac{2b}{2(l_{fr} + l_{re})} \cot\delta_{re} \left[\tan\delta_{fr} - \tan\delta_{re} \right] \right) \quad (2.11)$$

$$\delta_4 = \delta_{ro} = \cot^{-1}\left(\cot\delta_{re} + \frac{2b}{2(l_{fr} + l_{re})} \cot\delta_{re} \left[\tan\delta_{fr} - \tan\delta_{re} \right] \right) \quad (2.12)$$

2.2 Four wheel steering maneuvers

There are two kinds of special maneuver for four wheel steering vehicle, i.e. parallel steering maneuver and zero-sideslip maneuver [57]. These maneuvers take advantage of the unique kinematic characteristics of four wheel steering vehicles. In this thesis, the experimental 4WIS-AGV will be tested on its capability for these two special maneuvers.

2.2.1 Parallel steering maneuver

This maneuver is done when both the front and rear wheels are steered with the same angle, direction and velocity as shown in Fig. 2.5.

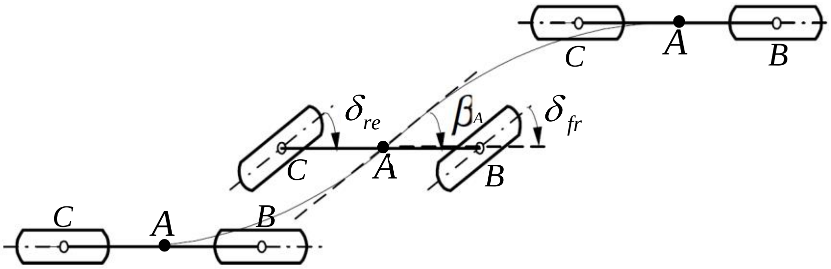


Fig. 2.5 Parallel steering maneuve

Therefore, in this condition, the front and rear steering angles are set as follows

$$\delta_{fr} = \delta_{re} \tag{2.13}$$

and

$$\beta_A = \delta_{fr} = \delta_{re} \tag{2.14}$$

This condition causes the turning radius of the vehicle motion, r_{tc} , always equal to infinity along the path.

$$r_{tc} = \infty , \qquad s : s_0 \rightarrow s_n \tag{2.15}$$

where s represents the path length travelled by the reference point of vehicle from the starting point, s_0 , and s_n represents the ending point.

Furthermore, in this condition, the vehicle moves without changing its orientation during the motion, and mathematically this condition can be written as follows:

$$\theta(s) = \theta_0 , \qquad s : 0 \rightarrow s_n \tag{2.16}$$

where θ_0 is the initial orientation of the vehicle.

This maneuver is very practical in vehicle lane-changing and obstacle avoidance. Furthermore, this maneuver is very useful for AGVs when the AGVs have to move in small working space on which ordinary AGVs have limitation to move.

2.2.2 Zero-sideslip maneuver

This maneuver is done when the sideslip angle of the vehicle is kept to be zero while the vehicle moves. This maneuver is illustrated with Fig. 2.6.

At a point A on the path, the incremental change, ds, can be expressed as:

$$ds = \sqrt{dX^2 + dY^2} \tag{2.17}$$

where dX and dY are the incremental changes in X and Y directions, respectively. The tangential angle θ_A at point A can be expressed as:

$$\theta_A = \tan^{-1}\frac{dY}{dX} \tag{2.18}$$

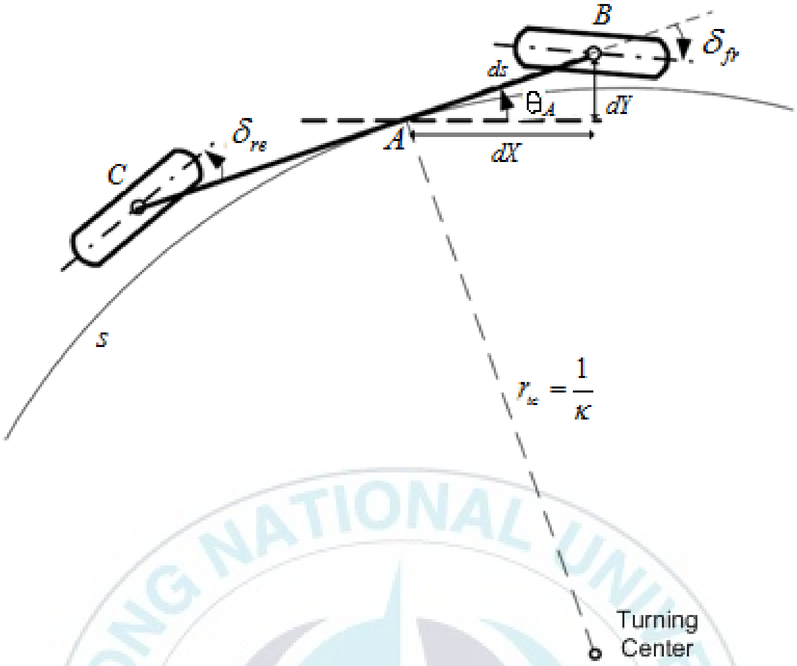


Fig. 2.6 Zero-sideslip maneuver

Subsequently, the path curvature can be obtained by deriving function $\theta_A(s)$ with respect to s as follows:

$$\kappa = \frac{d\theta_A}{ds} = \frac{d^2Y/dX^2}{\left(1+\left(dY/dX\right)^2\right)^{3/2}} \tag{2.19}$$

When the vehicle moves along the path s, the sideslip angle can be written as follows:

$$\beta_A(s) = 0, \quad s : s_0 \to s_n \tag{2.20}$$

The orientation of the vehicle $\theta_A(s)$ is set to match the tangential angle of the desired path $\theta_d(s)$.

$$\theta_A(s) = \theta_d(s), \qquad s : 0 \to s_n \tag{2.21}$$

Because the vehicle body is always tangent to the path, this maneuver is desirable for the vehicle motion.

2.3 Kinematic modeling of the 4WIS-AGV system

Fig. 2.7 shows the configuration of the 4WIS-AGV system for kinematic modeling. The coordinate Axy represents local coordinate, whereas the coordinate OXY represents global coordinate.

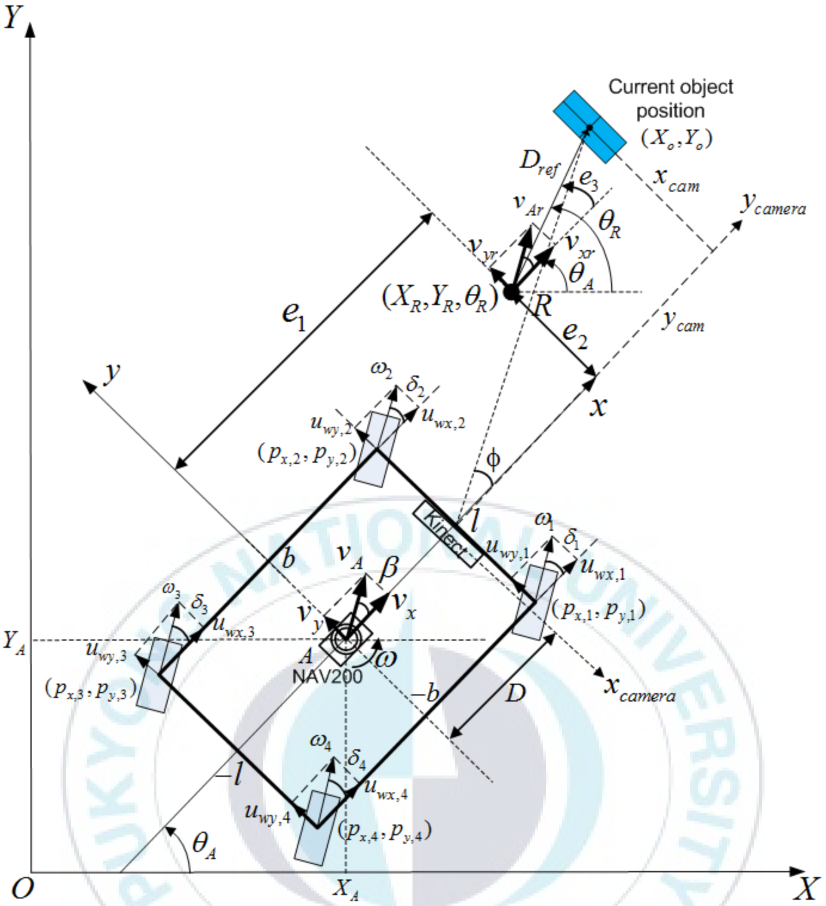


Fig. 2.7 Configuration of the 4WIS-AGV system

The kinematic equation for the proposed 4WIS-AGV is given by Eq. (2.22).

$$\begin{bmatrix} \dot{X}_A \\ \dot{Y}_A \\ \dot{\theta}_A \end{bmatrix} = \begin{bmatrix} \cos\theta_A & -\sin\theta_A & 0 \\ \sin\theta_A & \cos\theta_A & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \qquad \cdots (2.22)$$

where $\dot{X}_A$ is the vehicle linear velocity in X axis of global coordinate,

$\dot{Y}_A$ is the vehicle linear velocity in Y axis of global coordinate, and $\dot{\theta}_A$ is the vehicle angular velocity in global coordinate. v_x is the vehicle linear velocity in x axis of local coordinate, v_y is the vehicle linear velocity in y axis of local coordinate, and ω is the vehicle angular velocity in local coordinate. (X_A, Y_A) is the current position of the 4WIS-AGV in global coordinate. (X_R, Y_R) is reference position and θ_R is reference orientation in global coordinate. δ_i and ω_i $(i = 1, 2, 3, 4)$ are the i^{th} wheel steering angle and the i^{th} wheel angular velocity, respectively.

For the given total linear velocity of 4WIS-AGV in local coordinate v_A and vehicle sideslip angle β, the vehicle linear velocities v_x and v_y in the local coordinate can be obtained by Eq. (2.23).

$$\begin{cases} v_x = v_A \cos\beta \\ v_y = v_A \sin\beta \end{cases} \qquad \cdots (2.23)$$

The vehicle sideslip angle β is defined as the angle between the movement directions of 4WIS-AGV and x axis of 4WIS-AGV by Eq. (2.24).

$$\beta = \operatorname{atan2}(v_y, v_x) = \tan^{-1}\left(v_y / v_x\right) \quad \cdots (2.24)$$

Target velocities in x and y directions of the i^{th} wheel in local coordinate of 4WIS-AGV $u_{wx,i}$ and $u_{wy,i}$ can be obtained by Eq. (2.25).

$$\begin{cases} u_{wx,i} = v_x - p_{y,i}\omega \\ u_{wy,i} = v_y + p_{x,i}\omega \end{cases} \quad \cdots (2.25)$$

where ($p_{x,i}$, $p_{y,i}$) is the position of the i^{th} wheel in the local coordinate of 4WIS-AGV as shown in Fig. 2.7.

The i^{th} wheel steering angle δ_i and the i^{th} wheel angular velocity ω_i are given by Eq. (2.26) and Eq. (2.27).

$$\delta_i = \tan^{-1}\left(\frac{u_{wy,i}}{u_{wx,i}}\right) \quad \cdots (2.26)$$

$$\omega_i = \frac{u_{wx,i}\cos\delta_i + u_{wy,i}\sin\delta_i}{R_w} \quad \cdots (2.27)$$

where R_w is the radius of a wheel.

Chapter 3: Hardware Structure of 4WIS-AGV

This chapter describes the experimental 4WIS-AGV that consists of mechanical part design, electrical part design and software development.

3.1 AGV system

Fig. 3.1 shows a Kinect camera sensor-based object following 4WIS-AGV system used in this paper. This system mainly consists of 4WIS-AGV, a Kinect camera sensor, an laser navigation system NAV-200 and a blue colored rectangular board as a candidate object.

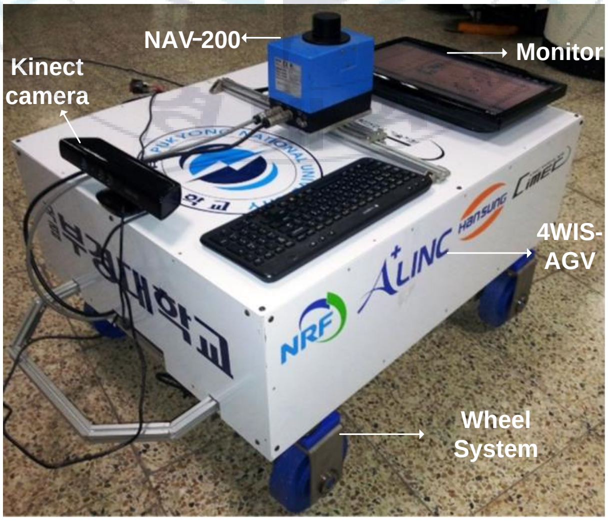


Fig. 3.1 Kinect camera sensor-based object tracking and following 4WIS-AGV system

The 4WIS-AGV has dimension of 1.00 m (l) x 0.70 m (w) x 0.50 m (h). This system consists of body configuration and wheel driving configuration. The wheel driving configuration has four wheel systems [58] where each wheel uses two high power DC motors: one for driving purpose and the other for steering purpose.

The 4WIS-AGV is controlled by Industrial PC (Tank 800) with RS232 communication and eight ATmega128 microcontrollers (one for each motor). Two 12V batteries are used for the power supply to the system. A monitor and a wireless keyboard- mouse are used as an input device for the 4WIS-AGVsystem.

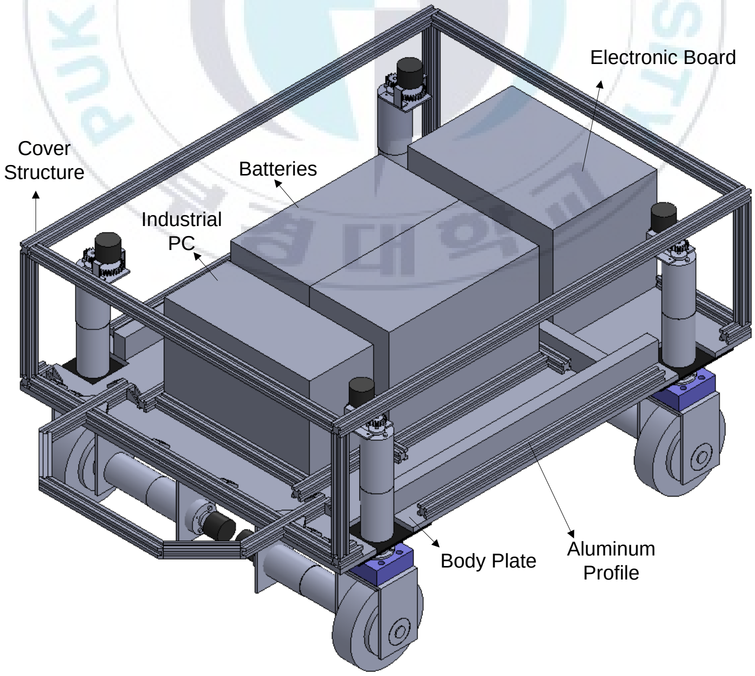
The sensors used for the proposed system are Kinect camera sensor, laser navigation sensor NAV-200 and eight encoders, one for each motor. A Kinect camera sensor is mounted on the top-front side of the 4WIS-AGV whereas NAV-200 is mounted on the top-center of the 4WIS-AGV. The Kinect camera sensor used in this system is basically used for Xbox gaming console which can capture both color and depth images at the speed of 30 frames per second and uses default mode of operation (sensing range for default mode is 0.8m to 4m). NAV-200 with an accuracy ±25 mm is used to obtain the position for both object and 4WIS-AGV in the global coordinate system. Out of eight encoders, four encoders are used to measure wheel steering angles of the steering motors and other four are used to measure angular velocities of the driving motors.

3.1.1 Mechanical part design

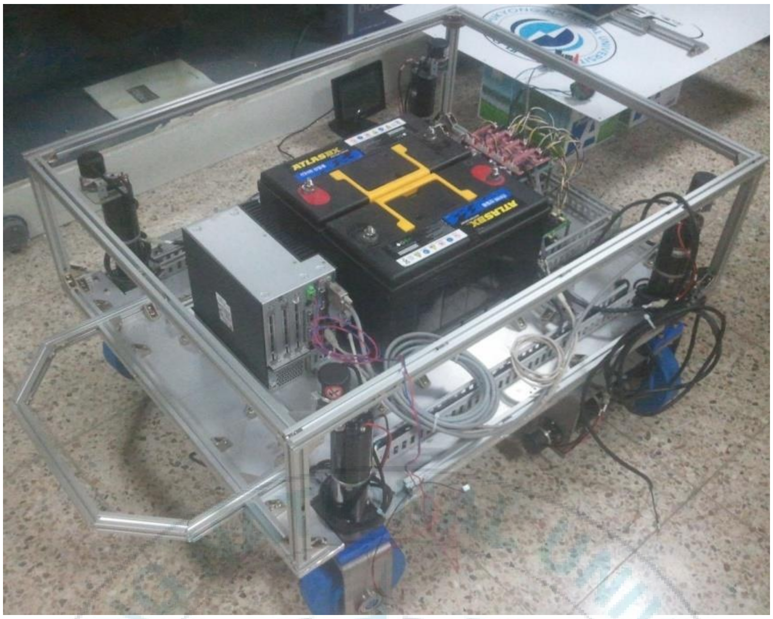
Mechanical parts of the proposed 4WIS-AGV are designed as body configuration and wheel driving configuration.

3.1.1.1 Body configuration

The body configuration of the 4WIS-AGV is shown in Fig. 3.2.



(a) Designed body configuration



(b) Real body configuration

Fig. 3.2 Body configuration

The specification of the vehicle body is listed in Table 3.1. The body configuration consists of body plate, electronic board including microcontrollers and motor drivers, batteries, and industrial PC. The body plate used in this system is shown in Fig. 3.3.

Table 3.1 Specification of the vehicle body

Descriptions	**Symbols**	**Values**
Distance from CG to front wheels	l_{fr}	0.34 m
Distance from CG to rear wheels	l_{re}	0.34 m
Distance between left and right wheels	b	0.625 m

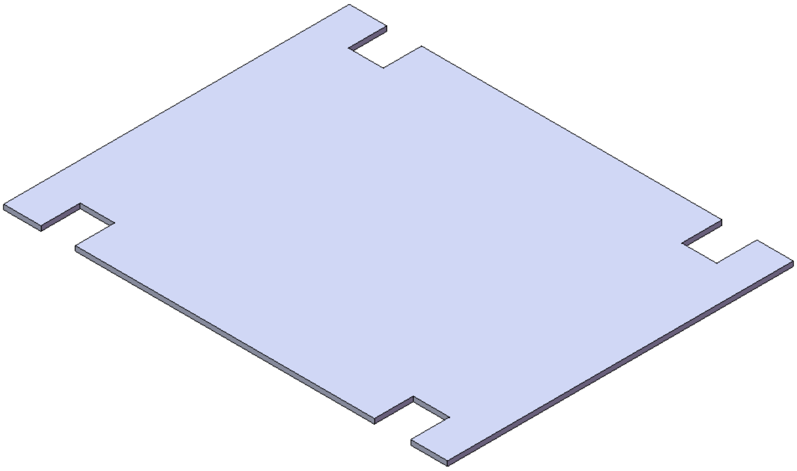


Fig. 3.3 Body plate

On top of this body plate, all equipments are placed. Since the AGV is designed for baggage carrier purpose, the body uses 100mm (T) x 705mm (W) x 905mm (L) aluminum plate such that it can support the system well. Industrial PC and batteries used in this system are explained in more detail in electrical design section. The body cover plate of this vehicle is shown in Fig. 3.4. The body cover plate consists of top cover plate, front cover plate, two side cover plate, and rear cover plate. The cover plate uses 2 mm-thickness aluminum plate.

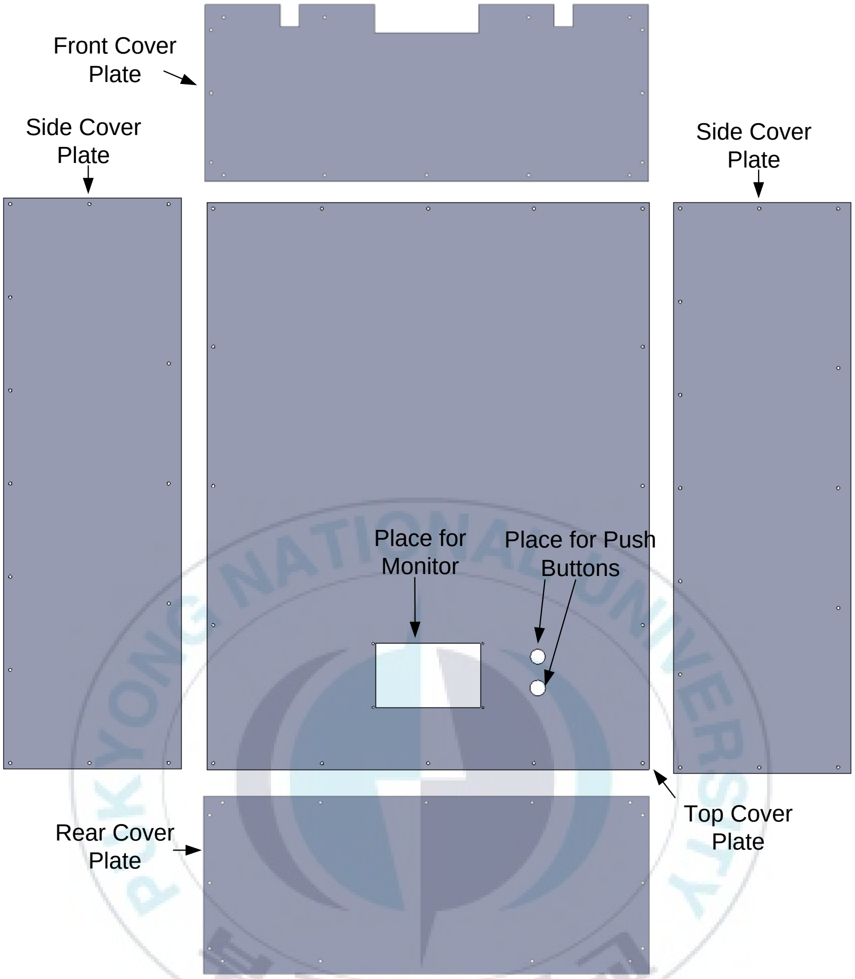
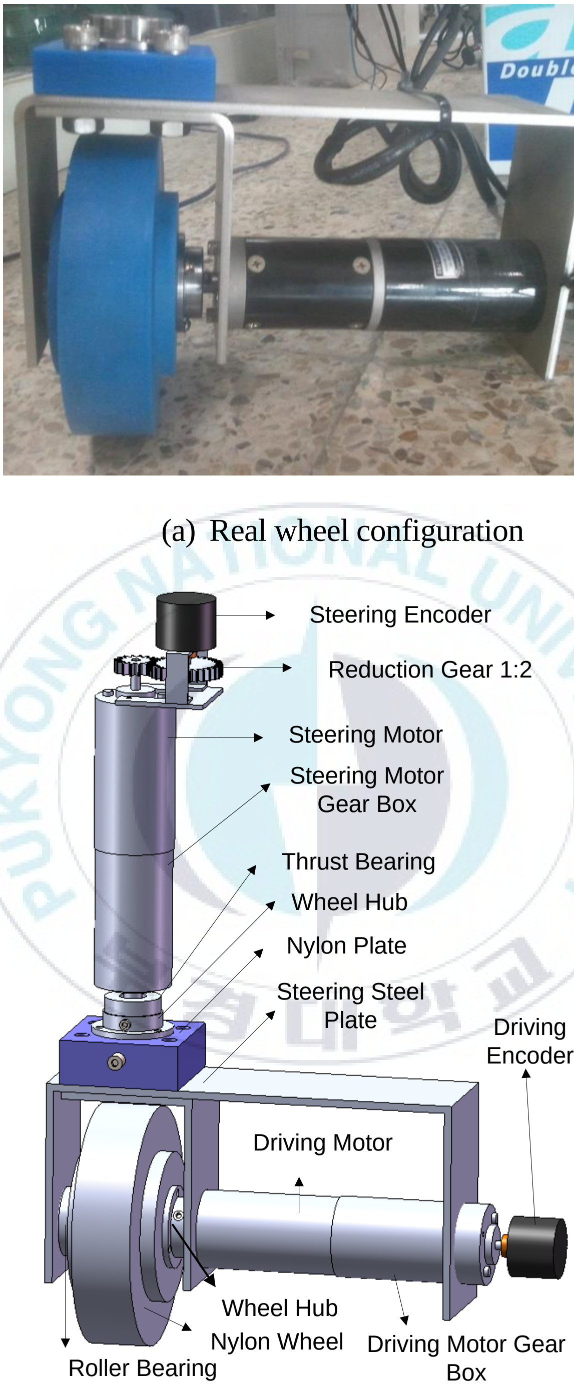


Fig. 3.4 Body cover plate

3.1.1.2 Wheel configuration

The wheel configuration in this system is shown in Fig. 3.5.



(a) Real wheel configuration

(b) Designed wheel configuration

Fig. 3.5 Wheel configuration

This configuration is comprised of two DC motors for driving and steering purposes. Both driving and steering motors are 48.6 W DC motors with driving motor gear box 1:100 for driving purpose and steering motor gear box 1:150 for steering purpose. Each motor is coupled with an encoder which is later used for positioning of the vehicle. Two wheel hubs are used: one to connect the steering motor shaft to the nylon plate, the other is to connect the driving motor shaft and nylon wheel as shown in Fig. 3.6. Roller bearing with 20mm (inner diameter) x 46mm (outer diameter) x 13mm (thickness) is used to hold the nylon wheel at the other side. Nylon plate is used to connect the steering motor with the steering steel plate. Because the steering steel plate has to also support vehicle weight, thrust bearing is used. Nylon plate and steering steel plate are shown in Figs. 3.7 and 3.8, respectively. Reduction gear 1:2 is used to couple the steering motor and steering encoder as shown in Fig. 3.9. Nylon wheel is used for the driving wheel as shown in Fig. 3.10.

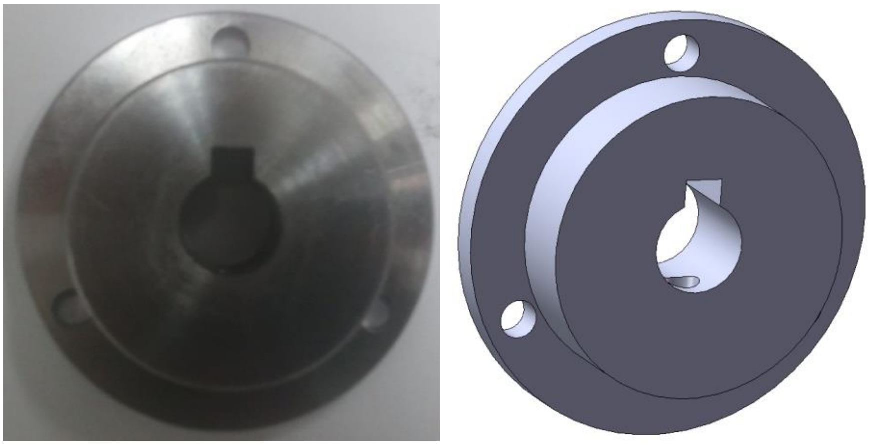


Fig. 3.6 Wheel hub

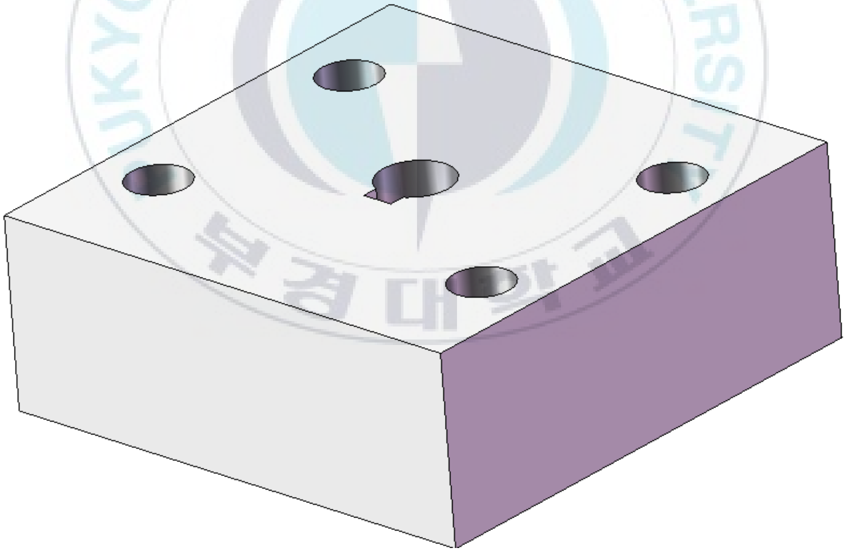


Fig. 3.7 Nylon plate

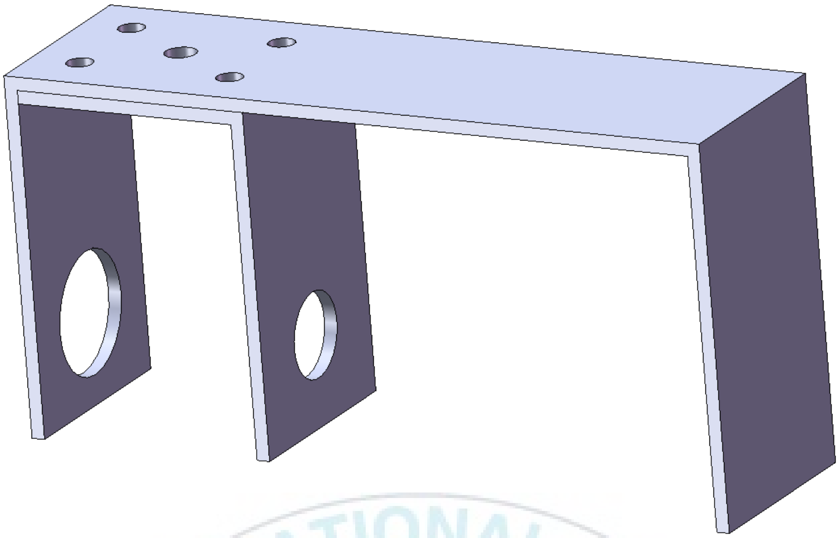


Fig. 3.8 Steering steel plate

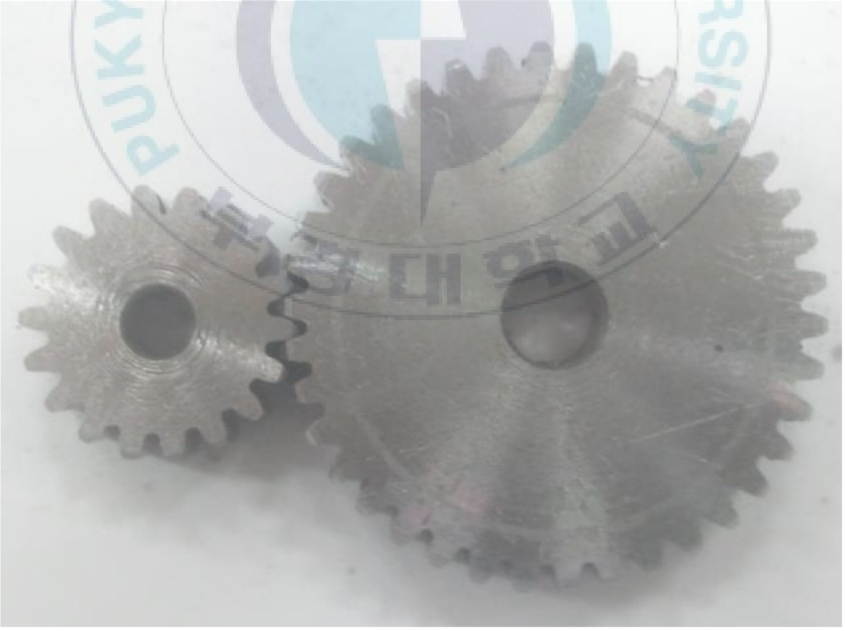


Fig. 3.9 Reduction gear 1:2

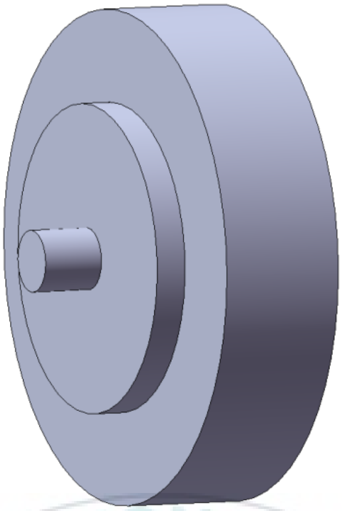


Fig. 3.10 Nylon wheel

3.1.2 Electrical design

Fig. 3.11 shows the control architecture of the proposed system. Kinect camera sensor (both RGB and depth) data and NAV-200 position data are used to obtain position of both object and 4WIS-AGV. Users give command to the 4WIS-AGV through the Graphic User Interface (GUI) on the monitor. Industrial PC receives user’s command, data from Kinect camera sensor and NAV-200, encoder data from Atmega128. The control signals generated from the controller are sent to eight motor drivers. Subsequently, these signals are converted to Pulse Width Modulation (PWM) signals by Atmega128 which are then converted to voltage signals to control four DC motors for steering and four DC motors for driving by motor driver. The software used for programming this system is Microsoft Visual Studio C# Express.

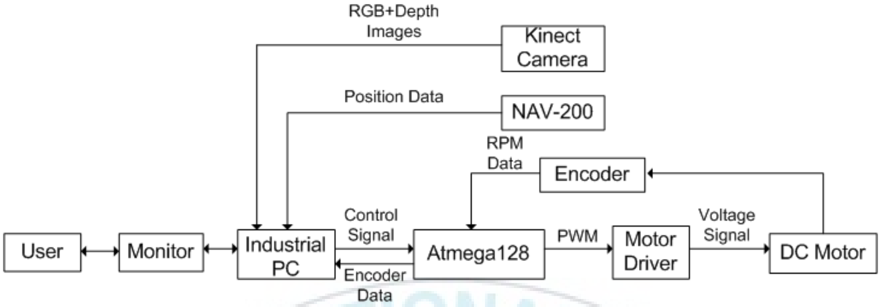


Fig. 3.11 Control architecture of the proposed system

3.1.2.1 DC motors

There are two kinds of DC motors used in the system as shown in Fig. 3.12. One is driving DC motor with 1/100 gear reduction ratio, another is steering DC motor with 1/150 gear reduction ratio. Both the two kinds of DC motors have the same specification, except for the gear box because the higher the gear reduction ratio is, the bigger the gear box length is. Motor performance graph is shown in Fig 3.13 and the detailed specification is listed in Table 3.2.

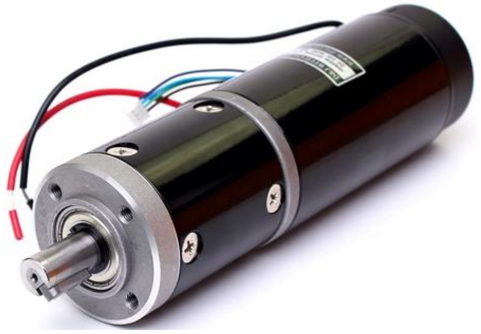


Fig. 3.12 DC motors

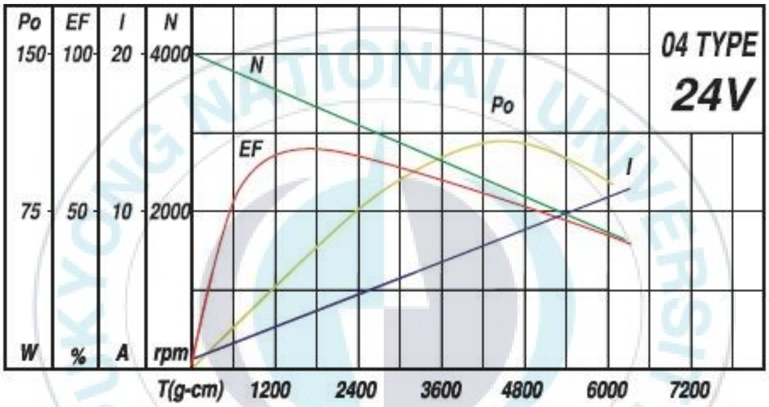


Fig. 3.13 Performance graph of the DC motors

Table 3.2 Specification of DC motor IG-52GM 04TYPE (24V)

No	**Parameters**	**Values**	
		Motor with 1/100 gear reduction ratio	Motor with 1/150 gear reduction ratio
1	Gear head (mm)	84	99.5
2	Weight (g)	1,770	1,940
3	Rated torque (kgf-cm)	78	98
5	Rated speed (RPM)	35	23.5
6	No load speed (RPM)	40	26

3.1.2.2 Encoders

In the 4WIS-AGV system, high-resolution encoders are used instead of built-in encoder from the DC motors. The encoder is rotary encoder E40HB-6-3600-3-V-5 as shown in Fig. 3.14. The specification of encoder is listed in Table 3.3.

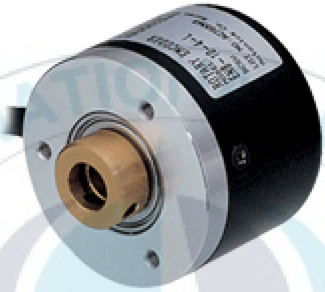


Fig. 3.14 Encoder E40HB-6-3600-3-V-5

Table 3.3 Specification of encoder E40HB-6-3600-3-V-5

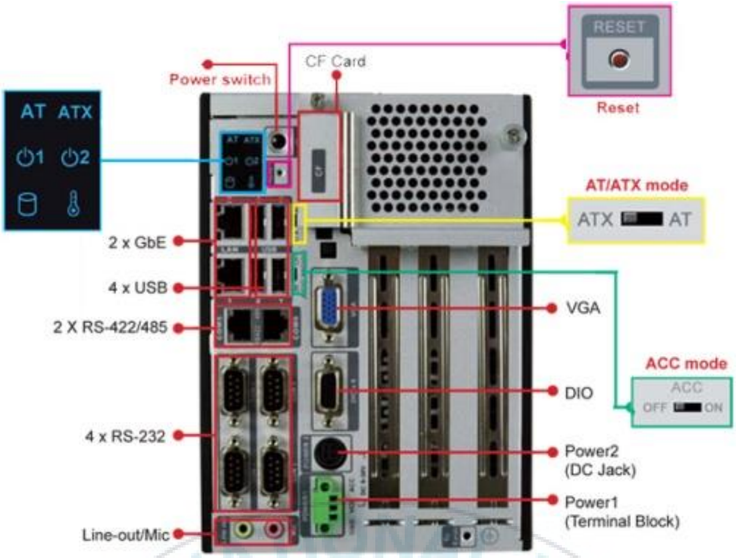
No	Parameters	Values
1	Shaft diameter (mm)	6
2	Resolution (pulse /1 revolution)	3600
3	Output phase	A, B, Z
4	Output	Voltage output
5	Power supply (V)	5
6	Weight (g)	120

These encoders are used to measure the revolution of the driving and steering wheels. The data from encoders are sent to the controller such that the movement of the vehicle can be controlled properly.

3.1.2.3 Industrial PC

Industrial PC TANK-800 is used as the main controller. The structure of industrial PC in this system is shown in Fig. 3.15. Industrial PC is basically a personal PC for industrial application. Thus, Industrial PC is very suitable for the 4WIS-AGVs because these systems need rugged controllers specially designed to operate reliably in harsh usage environments and conditions. Moreover, Industrial PC can perform high complexity calculation and collect high-frequency data from sensors. In this system, all the calculation of the controller and collecting data from positioning sensors are conducted by this PC.

(a) Overall view



(b) Back view

Fig. 3.15 Industrial PC TANK-800

The specification of Industrial PC TANK-800 is shown in Table 3.4.

Table 3.4 Specification of PC TANK-800

No	Parameters		Values
1	Chassis	Dimensions	136 mm x 219 mm x 188 mm
2	Motherboard	CPU	Intel® Atom™ D525 1.8GHz dual-core processor
		Chipset	Intel® ICH8M
		Ethernet	Dual Realtek RTL8111E PCIe GbE controllers support ASF 2.0
3	Storage	SATA	2.5’’ SATA HDD bay
4	System Function	USB	4 x USB 2.0
		Ethernet	2 x RJ-45
		RS-232	4 x DB-9
		RS-422/485	2 x RJ-45
		Display	1 x VGA
		Resolution	Up to 2048x1536
		Audio	1 x Mic-in, 1 x Line-out
		DIO	1 x DB-9
		Interior Expansions	One PCIe x4 (physical one PCIe x16 slot) and two PCI slots
5	Power	Power Supply	10.5V (+/-0.3V) ~ 36V
		Power Consumption	33W (without add-on card)
6	Reliability	Operating Temperature	-20°C ~ 70°C
		Weight	3.0Kg

3.1.2.4 Battery

In the 4WIS-AGV system, all motors need 24V power supply. Therefore, two ATLASBX ITX100 12V batteries are used as shown in Fig. 4.19. The specification of 12V battery ATLASBX ITX100 is listed in Table 3.5.

Fig. 3.16 12V battery ATLASBX ITX100

Table 3.5 Specification of 12V battery ATLASBX ITX100

No	Parameters	Values
1	Nominal Voltage (V)	12 V 100Ah
2	Weight	24.2 kg
3	Terminal type	Bolt terminal
4	Dimension (mm)	330 x 171 x 217

3.1.2.5 Microcontrollers AVR ATmega128

In the 4WIS-AGV system, microcontrollers ATmega128 are used to convert control signals generated by the controller into PWM signals which are then converted into voltage signal by motor driver. The ATmega128 is shown in Fig. 3.17. The specification of ATmega128 is listed in Table 3.6.

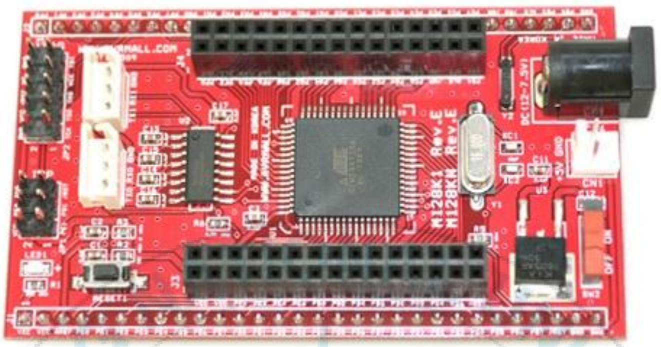


Fig. 3.17 Microcontroller AVR ATmega128

Table 3.6 Specification of microcontroller AVR ATmega128

No	Parameters	Remarks
1	Clock	16MHz
2	RS232 channel interface	Two
3	Operating power	4.5 to operate at 5.5V
4	Board size	4.6 x 7.9 cm
5	RTC-32.768KH	Available
6	ATMEL ISP port, JTAG port	Available
7	All PORT Pin Extension	Available
8	RESET switch	Available
9	Power LED (RED)	Available
10	External input power supply	Available: DC jack (power supply 7.5 ~ 12V Input)

In this system, because the motors are controlled independently, eight ATmega128 microcontrollers are used to control four steering motors and four driving motors.

3.1.2.6 Motor drivers

Each motor has motor driver as shown in Fig. 3.18. Therefore, eight motor drivers are used in the 4WIS-AGV system. This motor driver can work in the range of 12 ~ 24 V DC.

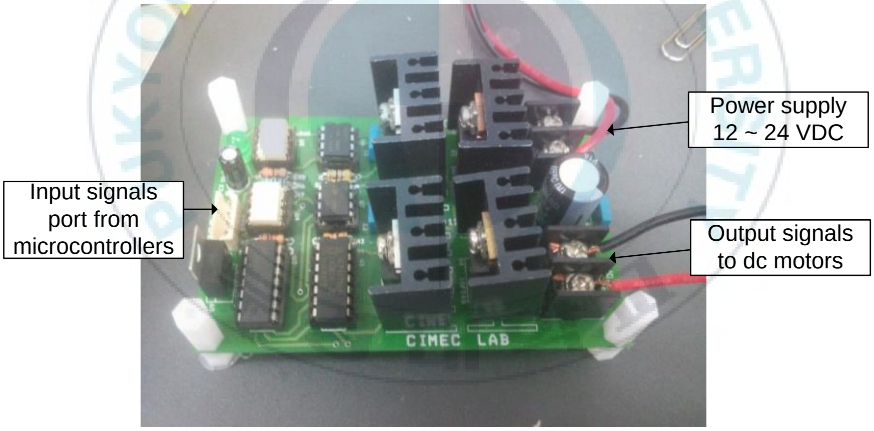

Power supply
12 ~ 24 VDC
Input signals port from microcontrollers
Output signals to dc motors
CIMEC LAB

Fig. 3.18 Motor drivers

3.1.2.7 Monitor

Fig. 3.19 shows the monitor. MYMO touch screen monitor MY-720 is used to display the data of the industrial PC Tank-800 to the users. It has 7 inch screen size. The specification of this monitor is listed in Table 3.7.

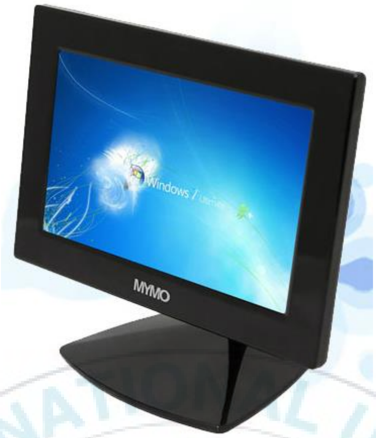


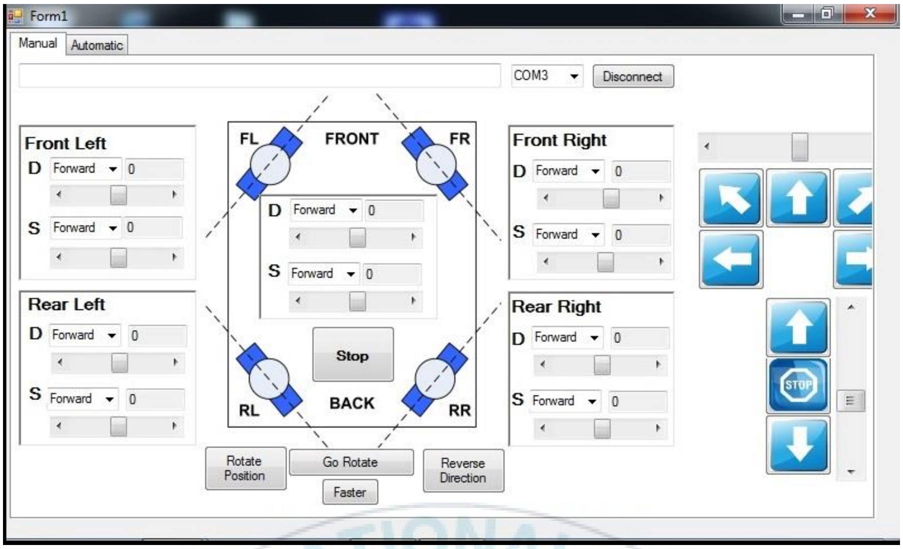
Fig. 3.19 Monitor MYMO MY-720

Table 3.7 Specification of monitor MYMO MY-720

No	Specification	TSP Model
1	Screen size	7.0 inch wide
2	Resolution	WVGA (800*480)
3	Display colors	16.7M
4	Brightness	250 cd/m^2
5	Contrast ratio	350 : 1
6	Response time	30 msec
7	Touch input	Touch screen panel
8	Video input	USB2.0 high speed
9	Connector	USB mini B type
10	Power input	USB power (5V, 500mA)
11	Power consumption	3.3 ~ 4.9W
12	UI, Pivot	Pivot (Landscape, Portrait)
13	Product dimension	194(W) x 133(L) x 180(H)

3.2 Graphical Unit Interface (GUI) software

The GUI software to control the 4WIS-AGV is created using Microsoft Visual Studio 2008 with C# programming language. Manual and automatic menus of the Graphical Unit Interface (GUI) for the 4WIS-AGV are shown in Fig. 3.20 and 3.21, respectively. In the manual menu of the GUI, users can control the 4WIS-AGV manually. The GUI shows that each wheel can be adjusted separately and also adjusted uniformly. Fig. 3.20 shows that there are four boxes for front left (FL), rear left (RL), rear right (RR), and front right (FR) wheels which have driving slider (D) and steering slider (S) in each box. These are used to control the values of each steering and driving motor in each wheel. The box at the middle center is all-control box which controls all the wheels within rotate position so rotate faster reverse direction. The right side box with many arrows is used to control the 4WIS-AGV motion. Five arrows at the top are used to adjust the direction of each wheel. Stop button, up arrow and down arrow under the five arrows are used to control the angular velocity of the driving motor of each wheel.



COM3

Front Left
D Forward 0
S Forward 0

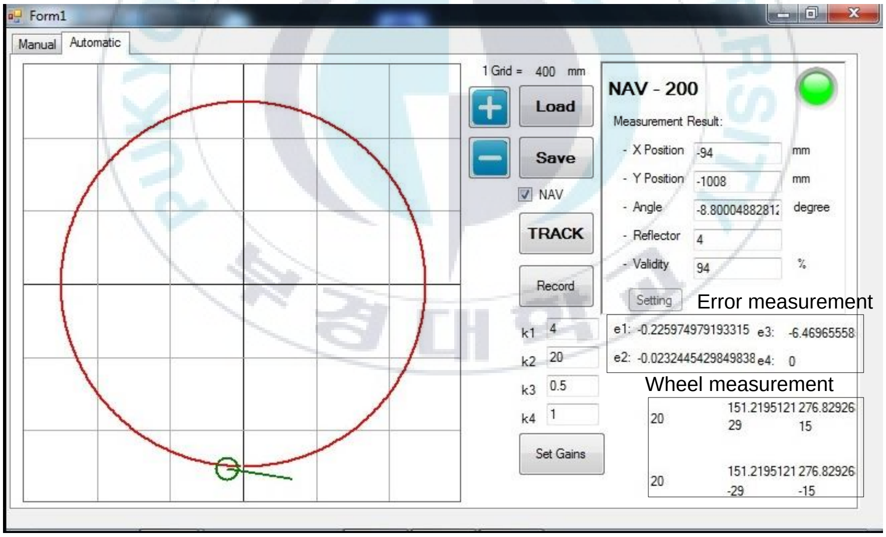
Rear Left
D Forward 0
S Forward 0

D Forward 0
S Forward 0

Front Right
D Forward 0
S Forward 0

Rear Right
D Forward 0
S Forward 0

Fig. 3.20 Manual menu of the 4WIS-AGV GUI



1 Grid = 400 mm

☑ NAV

k1 4
k2 20
k3 0.5
k4 1

Measurement Result:
- X Position -94 mm
- Y Position -1008 mm
- Angle -8.80004882812 degree
- Reflector 4
- Validity 94 %

e1: -0.225974979193315 e3: -6.46965558
e2: -0.0232445429849838 e4: 0

20	151.2195121 29	276.82926 15
20	151.2195121 -29	276.82926 -15

Fig. 3.21 Automatic menu of the 4WIS-AGV GUI

On the other hand, the automatic menu is where users can control the vehicle to track the given trajectories in Fig. 3.21. The left box is the map indicating the position of the 4WIS-AGV. The vehicle is illustrated as a circle with one line to show the heading direction. The circular red continuous line shows the trajectory that the vehicle is expected to track. The view of the map can be zoomed in and out using plus (+) and minus (-) buttons beside the map box. The box in the right side is the NAV-200 current data. NAV-200 shows the x and y positions and angle of the vehicle. The numbers of reflector and the validity are also shown below the position data. As mentioned in NAV-200 section, at least 3 reflectors are required to reflect the signals sent by NAV-200 such that it can generate position data. The validity values are dependent of the amount of the detected reflectors. This menu also shows the error measurement and the wheel measurement. Two numbers at the left side of the wheel measurement are δ_{fr} and δ_{re} in degree at the top and bottom, respectively. The eight numbers at the right side are wheel angular velocities ω_i (i=1,2,3,4) and wheel steering angles δ_i (i=1,2,3,4). Controller gains set for this case are shown at left middle side of the GUI. Gains can be adjusted here. To start the experiment, users should specify the trajectory by using Load button. Subsequently, click the record button which is to save all position data. Finally, press track button to start the trajectory tracking.

3.3 Kinect camera sensor

In this thesis, Kinect camera sensor is used for detecting a moving candidate object, and is installed in the front top of the 4WIS-AGV.

3.3.1 Kinect camera sensor

Kinect camera is firstly introduced as a periphery equipment of XBOX360 game machine by Microsoft in 2010. It’s actually a 3D somatosensory camera. The Kinect SDK for Windows is released in June 2011. It offers API interfaces to help users to develop customized software to couple the Kinect camera with a computer. Users can develop their own software through it.

3.3.2 Kinect Structure

The Microsoft Kinect camera sensor is a revolutionary low-cost RGB-D camera sensor that is primarily built as an input device for Xbox gaming console [59]. Due to its capability of producing better quality image and depth information, this low cost device becomes popular in the field of scientific study especially in the field of computer vision and robotics. A software development kit (SDK) provided by Kinect for Microsoft windows offers easy development of Kinect applications [60]. This SDK allows developers to write Kinect applications in C#, C++ or Visual Basic. NET programming languages. Fig. 3.22 shows Kinect camera sensor Xbox 360, four microphones array, and a tilting system.

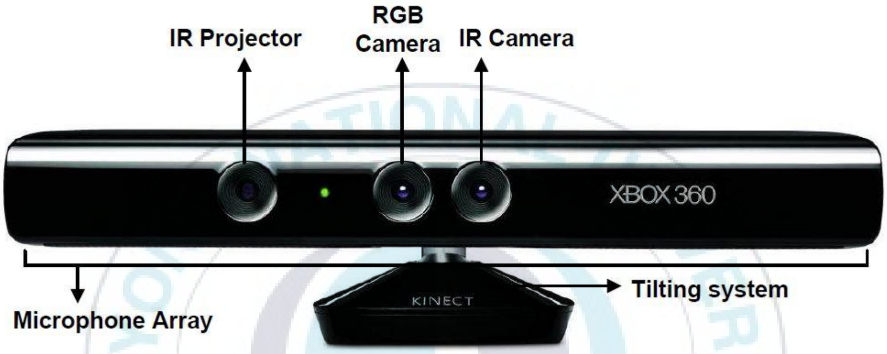


Fig. 3.22 Kinect camera sensor Xbox 360

The specifications of Kinect Xbox camera sensor are shown in Table 3.8.

Table 3.8 Specification of the Kinect Xbox camera sensor

No.	Parameters	Values
1	Depth sensor range (mm)	800 to 4000
2	Horizontal field of view (degree)	57
3	Vertical field of view (degree)	43
4	Physical tilt range (degree)	±27
5	Color data stream	640x480 32-bit color @ 30 FPS
6	Depth data stream	320x240 16-bit depth @ 30 FPS

The Microsoft Kinect also has its own image processor known as the Primesense's PS1080-A2 system on chip (SoC) processor that processes the images captured by the monochrome and infrared camera. Apart from all these components, Kinect camera sensor also has a power adapter for external power supply and a USB adapter to be connected with a computer.

The depth camera is comprised of an IR laser projector and a monochrome CMOS sensor called as IR camera, which captures video data in 3D under any ambient light conditions.

Chapter 4: 4WIS-AGV and Object Position Measurement

4.1 4WIS-AGV position measurement using NAV-200

In this paper, the position of the 4WIS-AGV in global coordinate is obtained by using laser navigation system NAV-200. The basic principle of NAV-200 installed at the top center of AGV is shown in Fig. 4.1.

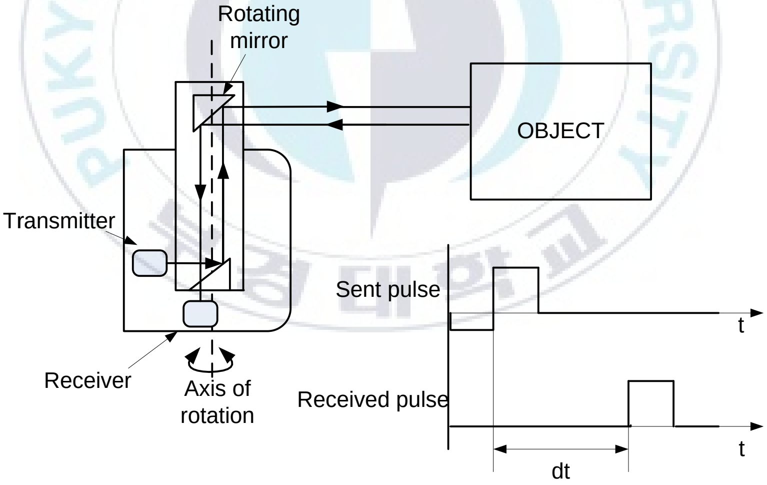


Fig. 4.1 Laser navigation system NAV-200 basic principle

The NAV-200 calculates its own position and orientation based on the fixed reflectors positioned (R1, R2, R3, … , Rn) in the environment, and its coordinates are detected two-dimensionally in a

360° scanning angle using a laser beam invisible to the human eye. In the Fig. 4.2, n is the number of detected reflectors. $\left(X_i, Y_i\right)$ are coordinate of the i^{th} reflector in global coordinate frame. One revolution of the scanner head here is equivalent to a scan and each revolution generates one reading for each detected reflector. The measurement result are distance between sensor and reflector (d_1, d_2, d_3,⋯,d_n) and measurement angle (θ_1, θ_2, θ_3, …, θ_n).

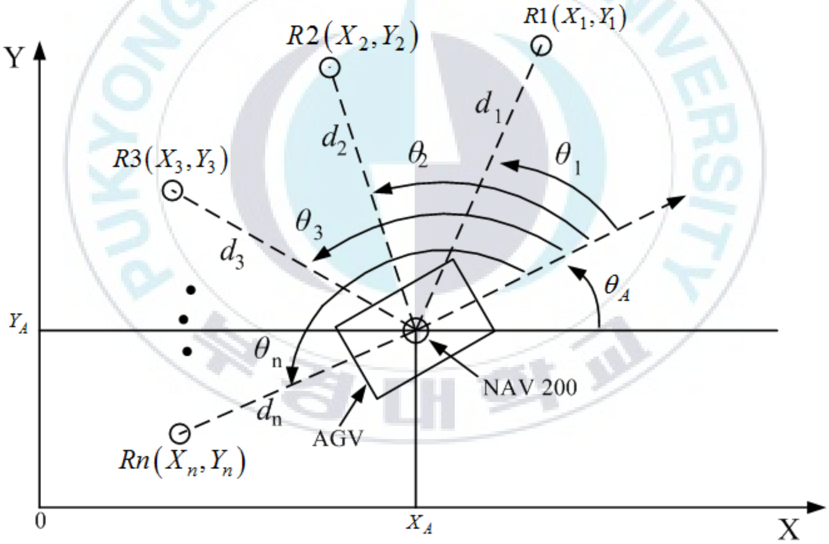


Fig. 4.2 Position measurement of the 4WIS-AGV using NAV-200

The specification of laser navigation system NAV-200 is shown in Table 4.1.

Table 4.1 Specification of laser navigation system NAV-200

No	Parameters	Values
1	Light source	Infrared (855 nm)
2	Laser class	1
3	Field of view	360 °
4	Scanning frequency	8 Hz
5	Angular resolution	0.1°
6	Operating range	1.2 m – 28.5 m
7	Max. range 10 % reflectivity	28.5 m
8	Data communication	Serial (RS-232)
9	Data transmission rate	19,200 Hz
10	Operating voltage:	≥ 24 V DC ± 25%
11	Power consumption:	24 W
12	Weight	3.3 kg
13	Dimensions	176 mm x 178 mm x 115 mm

The coordinates of the reflectors used are stored in the nonvolatile reflector memory of the NAV-200 as a reference. Three detected reflectors in a layer are sufficient for position measurement. For position measurement, the NAV-200 measures the distances and angles of the reflectors and works out its own position from this data. By comparing the reflector data in the NAV-200 memory and the measurement result, the position and orientation of AGV can be determined.

The coordinate of the AGV $\left(X_A, Y_A\right)$ and it's orientation θ_A can be obtained by Eqs. (4.1)~(4.2).

$$\begin{aligned} X_A &= \left[1/(m_2 - m_1)(1+m^2)\right] \\ &\quad \{(m_2 m - 1)\times\left[(m_1 + m)X_1 + (m_1 m - 1)Y_1\right] \\ &\quad -(m_1 m - 1)\times\left[(m_2 + m)X_2 + (m_2 m - 1)Y_2\right]\} \end{aligned} \qquad \dots\ (4.1)$$

$$\begin{aligned} Y_A &= \left[1/(m_2 - m_1)(1+m^2)\right] \\ &\quad \{(m_1 + m)\times\left[(m_2 + m)X_2 + (m_2 m - 1)Y_2\right] \\ &\quad -(m_2 + m)\times\left[(m_1 + m)X_1 + (m_1 m - 1)Y_1\right]\} \end{aligned} \qquad \dots\ (4.2)$$

$$\theta_A = \mathrm{Atan2}\left(Y_1 - Y_A, X_1 - X_A\right) - \theta_1 \ \dots\ (4.3)$$

where $m_1 = \tan\theta_1$, $m_2 = \tan\theta_2$, $m_3 = \tan\theta_3$ and $m = \tan\theta_A$.

From Eqs. (4.1)~(4.2), *m* is given by Eq. (4.4).

$$m = \frac{(m_3 - m_1)(Y_1 - Y_2 - m_1 X_1 + m_2 X_2) - (m_2 - m_1)(Y_1 - Y_3 - m_1 X_1 + m_3 X_3)}{(m_3 - m_1)(m_1 Y_1 + X_1 - m_2 Y_2 - X_2) - (m_2 - m_1)(m_1 Y_1 + X_1 - m_3 Y_3 - X_3)} \qquad \dots\ (4.4)$$

4.2 Object position measurement using Kinect camera sensor

This section first explain the procedure for getting the center position of a candidate object using Kinect camera sensor and then estimate the moving candidate object position using Kalman filter algorithm.

In this thesis, a color-based object detection algorithm [61] using Aforge.NET C# framework is used to obtain the center position of the blue colored candidate object in the local coordinate frame (i.e., in Kinect RGB image frame). Fig. 4.3 shows a RGB image frame having detected object with its center coordinates (x_m, y_m) within a desired object area.

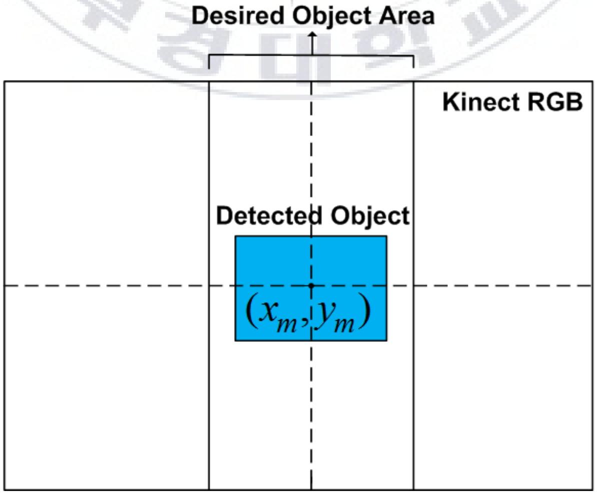


Fig. 4.3 Detected object within desired area inside RGB image frame

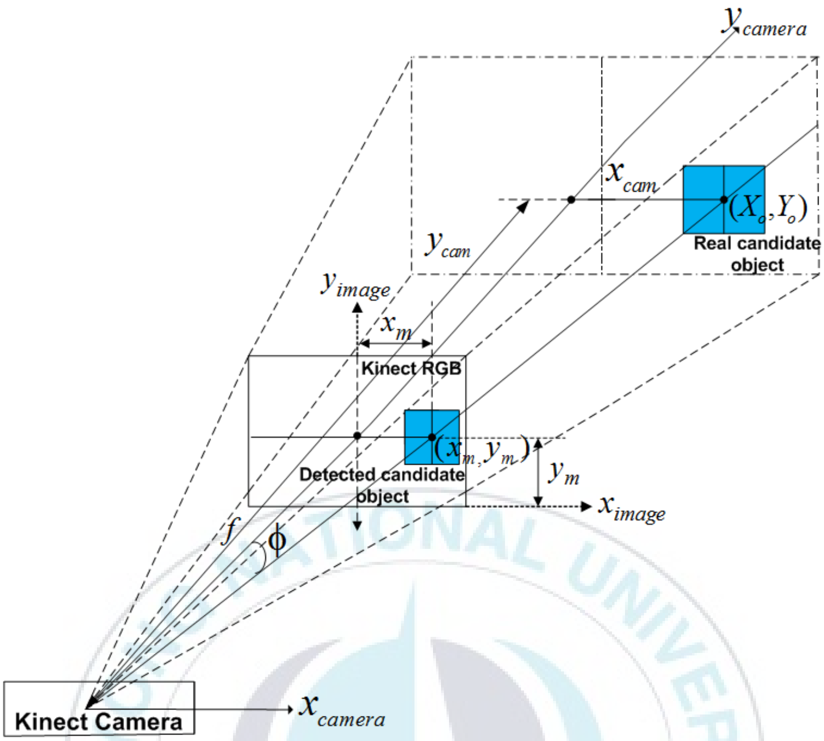


Fig. 4.4 Extended view of the real candidate object and the detected candidate object in a Kinect RGB image frame

Fig. 4.4 shows an extended view of the real candidate object and the detected candidate object in a RGB image frame from the Kinect camera sensor. According to the image principle [51], an angle ϕ between the center of the detected object and the center of the RGB image in Fig. 4.4 is given by Eq. (4.5).

$$\phi = \arctan\left(\frac{x_m \alpha}{f}\right) \qquad \dots (4.5)$$

where x_m is the object position in x direction from the center of the Kinect RGB image, α is the coefficient of transformation from pixel to length and f is the focal length of the Kinect camera. Since the value of x_m is in pixels, it is necessary to convert this value into meter value as x_{cam} given by Eq. (4.6).

$$x_{cam} = y_{cam} \tan\phi \qquad \dots (4.6)$$

where y_{cam} represents the depth of the object from the Kinect camera sensor in meter and is obtained directly from the Kinect depth data.

With enough small focal length, the position (x_{cam}, y_{cam}) of the detected object in local coordinate frame can be transformed into the global coordinate frame by Eq. (4.7) as shown in Fig. 2.7 and Fig. 4.4:

$$\begin{bmatrix} X_o \\ Y_o \end{bmatrix} = \begin{bmatrix} X_A \\ Y_A \end{bmatrix} + \begin{bmatrix} -\sin\theta_A & \cos\theta_A \\ \cos\theta_A & \sin\theta_A \end{bmatrix} \begin{bmatrix} x_{cam} \\ y_{cam} \end{bmatrix} + \begin{bmatrix} \cos\theta_A \\ \sin\theta_A \end{bmatrix} D \qquad \dots (4.7)$$

where (X_o, Y_o) is the current position of the detected object in global coordinate frame and $D = 0.45$ m is the distance between the Kinect camera sensor and the NAV-200 sensor placed on the 4WIS-AGV.

Chapter 5: Object Tracking Using Kalman Filter

To follow the moving object well, it is essential to track its position. However, due to the variations in the ambient light conditions and noise of the Kinect camera sensor, exact tracking becomes difficult. Therefore, to deal with these problems, Kalman filter is adopted in this paper [62]. Kalman filter estimates the current state of the target moving object by using estimated state from the previous time step and the current measurement, recursively.

The positions and velocities of the moving object are described in global coordinate frame OXY by the linear state space as Eq. (5.1).

$$\mathbf{x}_k = \left[X_o \quad Y_o \quad \dot{X}_o \quad \dot{Y}_o \right]^T \qquad \cdots (5.1)$$

where $\dot{X}_o$ is the object velocity in X direction i.e., derivative of position X_o with respect to time and $\dot{Y}_o$ is the object velocity in Y direction i.e., derivative of position Y_o with respect to time.

Between $(k-1)$ and k time steps, object undergoes a constant acceleration of a_k that is normally distributed (Gaussian) with mean 0 and standard deviation σ_a. The dynamic behavior of the moving object is given by the state vector $\mathbf{x}_k$ at k time step as Eq. (5.2).

$$\mathbf{x}_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{G}a_k = \mathbf{F}\mathbf{x}_{k-1} + \mathbf{w}_k \qquad \cdots (5.2)$$

where $\mathbf{F}$ is the state transition matrix and $\mathbf{w}_k$ is the Gaussian process noise at k time step with state covariance matrix $\mathbf{Q}$ such that $\mathbf{w}_k \sim N(0,\mathbf{Q})$ and are obtained as follows:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and}$$

$$\mathbf{Q} = \mathbf{G}\mathbf{G}^T\sigma_a^2 = \begin{bmatrix} T^4/4 & 0 & T^3/2 & 0 \\ 0 & T^4/4 & 0 & T^3/2 \\ T^3/2 & 0 & T^2 & 0 \\ 0 & T^3/2 & 0 & T^2 \end{bmatrix}\sigma_a^2$$

where $\mathbf{G} = \begin{bmatrix} \frac{T^2}{2} & \frac{T^2}{2} & T & T \end{bmatrix}^T$ and T is the sampling time.

The measurement vector $\mathbf{z}_k$ of the true position of an object $\mathbf{x}_k$

obtained from the Kinect camera sensor at *k* time step is expressed by Eq. (5.3).

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{v}_k \qquad \dots (5.3)$$

where $\mathbf{H}$ is the measurement matrix and $\mathbf{v}_k$ is the Gaussian measurement noise vector at *k* time step with covariance matrix $\mathbf{R}$ such that $\mathbf{v}_k \sim N(0,\mathbf{R})$ and are obtained as follows:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ and } \mathbf{R} = \begin{bmatrix} \sigma_p^2 & 0 & 0 & 0 \\ 0 & \sigma_p^2 & 0 & 0 \\ 0 & 0 & \sigma_v^2 & 0 \\ 0 & 0 & 0 & \sigma_v^2 \end{bmatrix}$$

In this paper, the objective of the Kalman filter is to obtain the estimated global position $(\hat{X}_o, \hat{Y}_o)$ of the detected moving object i.e., (X_o, Y_o) using the measurement data from the Kinect camera sensor. The Kalman filter exhibit two steps: prediction and update step as follows:

Prediction step: This step is a combination of predicted stateestimation and predicted estimation covariance given by Eq.

(5.4).

$$\begin{cases} \mathbf{x}_k^- = \mathbf{F}\hat{\mathbf{x}}_{k-1} \\ \mathbf{P}_\mathbf{k}^- = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^\mathbf{T} + \mathbf{Q} \end{cases} \qquad \ldots (5.4)$$

where $\mathbf{x}_k^-$ is the predicted state estimation vector at k time step and $\mathbf{P}_\mathbf{k}^-$ is the predicted estimation covariance matrix at k time step.

Update step: This step is a combination of optimal Kalman gain, updated state estimation and updated estimation covariance given by Eq. (5.5).

$$\begin{cases} \mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}^T(\mathbf{H}\mathbf{P}_k^-\mathbf{H}^T + \mathbf{R})^{-1} \\ \hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_k^-) \\ \mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_k^- \end{cases} \qquad \ldots (5.5)$$

where $\mathbf{K}_k$ is the optimal Kalman gain matrix at k time step, $\hat{\mathbf{x}}_k = [\hat{X}_0, \hat{Y}_0, \dot{\hat{X}}_0, \dot{\hat{Y}}_0]^T$ is the updated state estimation vector at k time step and $\mathbf{P}_k$ is the updated estimation covariance matrix at k time step.

Chapter 6: Controller Design

This section explains the controller design for the 4WIS-AGV that can follow the reference position (X_R, Y_R) and reference orientation θ_R with vehicle reference linear velocities $(v_{xr,} v_{yr})$ and vehicle reference angular velocity ω_R keeping a given safe distance between 4WIS-AGV and a moving object. The distance that has to maintain is chosen as $D_{ref} = 1.25$ m.

The reference position coordinates (X_R, Y_R) in Fig. 2.7 can be obtained by Eq. (6.1).

$$\begin{bmatrix} X_R \\ Y_R \end{bmatrix} = \begin{bmatrix} \hat{X}_o \\ \hat{Y}_o \end{bmatrix} - \begin{bmatrix} \cos\theta_R \\ \sin\theta_R \end{bmatrix} D_{ref} \qquad \cdots (6.1)$$

where $(\hat{X}_o, \hat{Y}_o)$ is the estimated positions of the moving candidate object by Kalman filter.

The reference vehicle linear velocity v_{Ar} in global coordinate can be expressed as Eq. (6.2).

$$v_{Ar} = \dot{X}_R \cos\theta_R + \dot{Y}_R \sin\theta_R = \dot{\hat{X}}_0 \cos\theta_R + \dot{\hat{Y}}_0 \sin\theta_R \qquad \ldots (6.2)$$

The reference vehicle linear velocities in local coordinate in Fig. 4 can be expressed as Eq. (6.3) and Eq. (6.4).

$$v_{xr} = v_{Ar} \cos e_3 \qquad \ldots (6.3)$$

$$v_{yr} = v_{Ar} \sin e_3 \qquad \ldots (6.4)$$

The reference orientation angle θ_R from Eq. (6.2) is calculated by Eq. (6.5).

$$\theta_R = \arctan\left(\frac{\hat{X}_o - X_R}{\hat{Y}_o - Y_R} \right) \qquad \ldots (6.5)$$

where $\dot{\hat{X}}_o, \dot{\hat{Y}}_o$ are the estimated velocities of the object obtained from Kalman filter.

The reference angular velocity ω_R can be obtained as Eq. (6.6).

$$\omega_R = \dot{\theta}_R \qquad \ldots (6.6)$$

Referring to Fig. 2.7, a tracking error vector $\mathbf{e}$ can be defined by Eq. (6.7).

$$\mathbf{e} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta_A & \sin\theta_A & 0 \\ -\sin\theta_A & \cos\theta_A & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_A - X_R \\ Y_A - Y_R \\ \theta_A - \theta_R \end{bmatrix} \qquad \cdots (6.7)$$

where e_1 is the error in local coordinate x-axis, e_2 is the error in local coordinate y-axis and e_3 is the vehicle orientation error.

To minimize these errors i.e., to make the 4WIS-AGV follow the moving object, a backstepping controller is designed for the 4WIS-AGV. Using backstepping control method, a candidate Lyapunov function (clf) is chosen as Eq. (6.8).

$$V = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 + \frac{1}{2}e_3^2 \qquad \ldots (6.8)$$

The time derivative of clf $\dot{V}$ is obtained by Eq. (6.9).

$$\dot{V} = e_1\dot{e}_1 + e_2\dot{e}_2 + e_3\dot{e}_3 \qquad \ldots (6.9)$$

Time derivative of the tracking error vector can be obtained by Eq. (6.10).

$$\dot{\mathbf{e}} = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & e_2 \\ 0 & 1 & -e_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} - \begin{bmatrix} \cos\theta_A & \sin\theta_A & 0 \\ -\sin\theta_A & \cos\theta_A & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{xr} \\ v_{yr} \\ \omega_R \end{bmatrix} \quad \ldots (6.10)$$

Substituting Eq. (6.10) in Eq. (6.9) yields Eq. (6.11).

$$\begin{aligned} \dot{V} &= e_1(v_x - v_{xr}\cos e_3 - v_{yr}\sin e_3) + e_2(v_y - v_{xr}\cos e_3 - v_{yr}\sin e_3) \\ &\quad + e_3(\dot{\theta}_A - \dot{\theta}_R) \end{aligned} \qquad (6.11)$$

Lyapunov stability criterion states that system is stable when $\dot{V} \le 0$. Therefore, to meet this condition, a control law vector U can be chosen by Eq. (6.12).

$$U = \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} = \begin{bmatrix} -k_1e_1 + v_{xr}\cos e_3 + v_{yr}\sin e_3 \\ -k_2e_2 - v_{xr}\sin e_3 + v_{yr}\cos e_3 \\ -k_3e_3 + \omega_R \end{bmatrix} \quad \ldots (6.12)$$

where k_1, k_2 and k_3 are positive constants.

Substituting Eq. (6.12) to Eq. (6.11) yields Eq. (6.13).

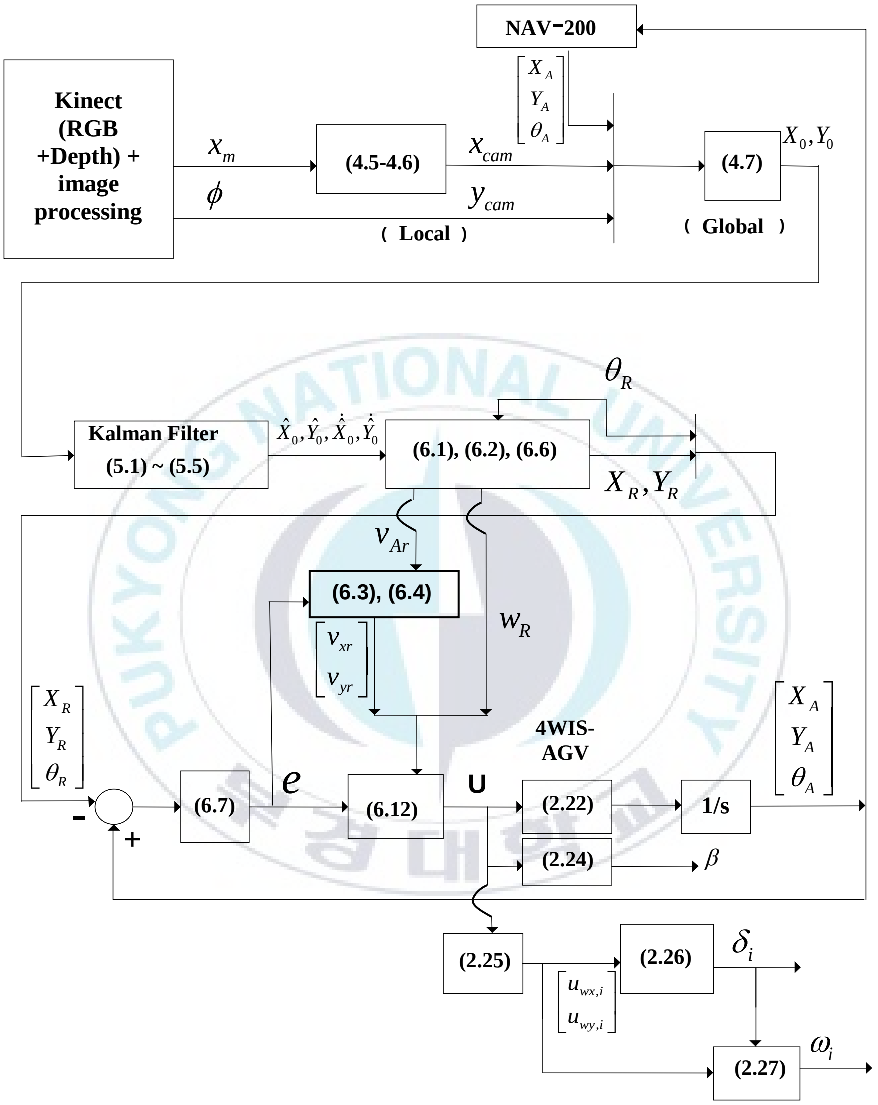
$$\dot{V} = -k_1e_1^2 - k_2e_2^2 - k_3e_3^2 \qquad \dots (6.13)$$

which shows that the value of $\dot{V} \leq 0$ by using the chosen control law vector U. In other words, the system becomes stable when e_1, e_2, e_3 $\rightarrow$ 0 as $t \rightarrow \infty$ using (6.10), (6.12), (6.13) and Barbalet's lemma.

For zero-sideslip maneuver, the vehicle sideslip angle β is considered as zero. Therefore, the vehicle linear velocity v_y in y axis of local coordinate becomes $v_y = v_{yr} = 0$ at steady state ($e_3 = 0$) and $\beta = 0$. The reference vehicle linear velocity v_{xr} in x axis of local coordinate with $\beta = 0$ and $e_3 = 0$ can be expressed as Eq. (6.14).

$$v_{xr} = v_{Ar} = \dot{\hat{X}}_o \cos\theta_R + \dot{\hat{Y}}_o \sin\theta_R \qquad \dots (6.14)$$

The block diagram of the proposed system is shown in Fig. 6.1.



NAV-200
Kinect (RGB +Depth) + image processing
x_m
ϕ
(4.5-4.6)
x_{cam}
y_{cam}
(Local)
$\begin{bmatrix} X_A \\ Y_A \\ \theta_A \end{bmatrix}$
(4.7)
X_0, Y_0
(Global)
θ_R
Kalman Filter (5.1) ~ (5.5)
$\hat{X}_0, \hat{Y}_0, \dot{\hat{X}}_0, \dot{\hat{Y}}_0$
(6.1), (6.2), (6.6)
X_R, Y_R
v_{Ar}
(6.3), (6.4)
w_R
$\begin{bmatrix} v_{xr} \\ v_{yr} \end{bmatrix}$
$\begin{bmatrix} X_R \\ Y_R \\ \theta_R \end{bmatrix}$
-
+
(6.7)
e
(6.12)
U
4WIS-AGV
(2.22)
1/s
$\begin{bmatrix} X_A \\ Y_A \\ \theta_A \end{bmatrix}$
(2.24)
β
(2.25)
(2.26)
δ_i
$\begin{bmatrix} u_{wx,i} \\ u_{wy,i} \end{bmatrix}$
(2.27)
ω_i

Fig. 6.1 Block diagram of the proposed controller

Chapter 7: Simulation and Experimental Results

The performance and the effectiveness of the proposed system are verified by simulation and experiment. The parameter and initial values for the simulation and experiment are shown in Table 7.1.

Table 7.1 Parameter and initial values

Description	**Symbol**	**Value**	**Unit**
Initial position of the 4WIS-AGV	(X_A, Y_A)	(0, 0)	m
Initial values of 4WIS-AGV orientation and sideslip angle	(θ_A, β_A)	(0, 0)	rad
Initial position of the candidate object	(X_o, Y_o)	(1, 2.5)	m
Initial velocity of the candidate object	$(\dot{X}_o, \dot{Y}_o)$	(0.1, 0.1)	m/s
Controller gain 1	k_1	1	-
Controller gain 2	k_2	1	-
Controller gain 3	k_3	1	-

Based on the initial positions of the 4WIS-AGV and the candidate object, the simulation and experimental results of the proposed system are shown in Figs. 7.1-7.3. Fig. 7.1 shows the tracking error vector. The initial values of the tracking errors e_1, e_2, e_3 are $-0.5, -1.2, -1.1,$ respectively. In the simulation case, the errors $e_1, e_2, e_3 \to 0$ at 5 s, 2.5 s, 8 s, respectively. In the experimental case, error e_1 is bounded between ± 0.05 m after 7 s, error e_2 is bounded between ± 0.01 m after 5 s, and error e_3 is bounded between ± 0.1 rad after 12 s. Fig. 7.2 shows the results for the control law vector *U*. Fig. 7.2(a) shows the results for the vehicle linear velocity in x direction. In the simulation, the maximum linear velocity is 0.65 m/s and then it decreased and keeps constant to 0.17 m/s at 5 s. In the experiment, the maximum velocity is 0.48 m/s and then it is bounded between 0.16~0.18 m/s after 7 s. Fig. 7.2(b) shows the results for the vehicle linear velocity in y direction. It can be observed that in the simulation result, the maximum velocity 1.35 m/s and then it decreased and keeps constant to -0.05 m/s at 2.5 s, whereas in the experiment, the maximum velocity is 0.95 m/s and then it is bounded between $-0.05 \sim 0$ m/s after 7 s. The simulation and experimental results for the vehicle angular velocity are shown in Fig. 7.2(c). In the simulation, the maximum angular velocity is 0.46 rad/s and then it converged to 0 at 8 s, whereas in the experimental case, the maximum angular velocity is 0.28 rad/s and then it is decreased and bounded between ±0.02 rad/s after 10 s. Fig. 7.3 shows the simulation and experimental results of the estimated values of the candidate object movement such as $\hat{X}_o, \hat{Y}_o, \dot{\hat{X}}_o$ and $\dot{\hat{Y}}_o$ obtained using Kalman filter. It can be observed that the Kalman filter can track the moving candidate object well with small errors of ± 0.025 m, ± 0.05 m, ± 0.01 m/s and ±0.01 m/s in $\hat{X}_o, \hat{Y}_o, \dot{\hat{X}}_o$ and $\dot{\hat{Y}}_o$, respectively. Figs. 7.4-7.5 show the clear view of the candidate object movement and the 4WIS-AGV movement in the global coordinate frame, respectively. The experiment is performed to verify the performance of the proposed system and its results are shown in Fig. 7.6. In this case, after the successful detection of the blue colored object, the 4WIS-AGV system tracks the position of the detected object and changes its orientation towards it. As the object moves forward, the 4WIS-AGV follows and tracks the moving object making all the errors approximately equal to zero. The 4WIS-AGV continues to follow the object keeping the reference distance (1.25 m) between the object and the 4WIS-AGV.

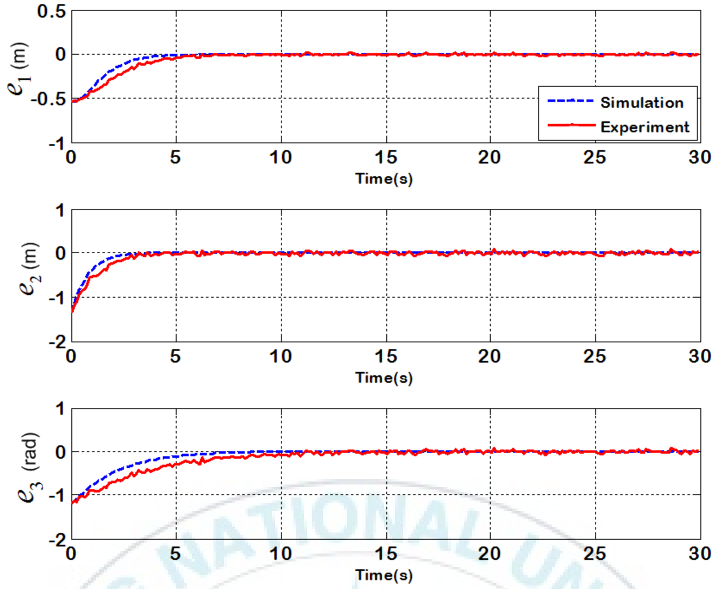
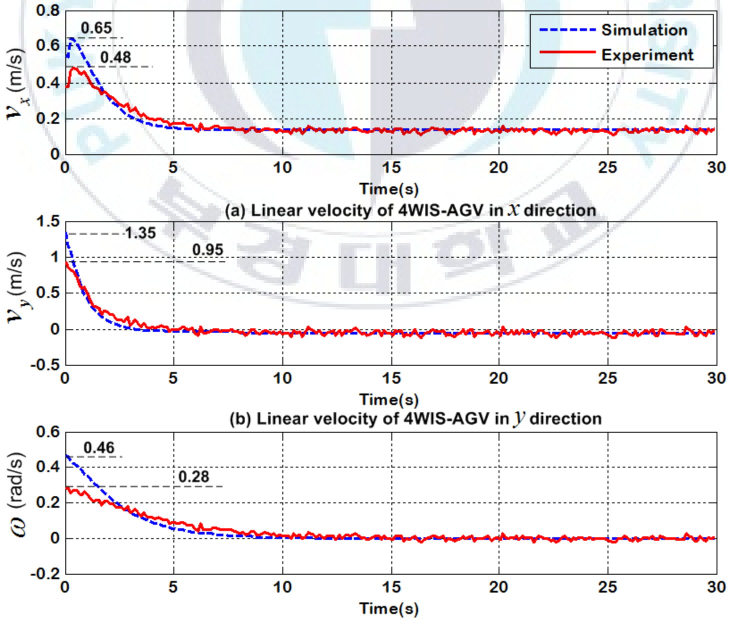


Fig. 7.1 Tracking error vector



(a) Linear velocity of 4WIS-AGV in x direction

(b) Linear velocity of 4WIS-AGV in y direction

(c) Angular velocity of 4WIS-AGV

Fig. 7.2 Control law vector U

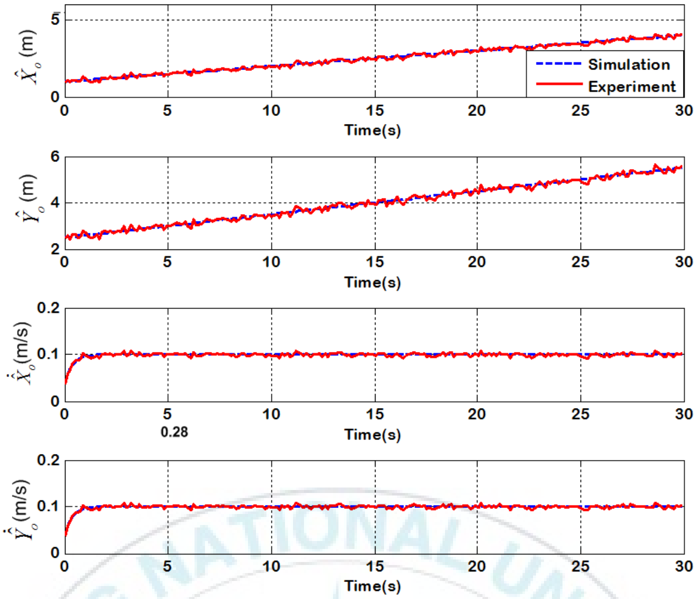


Fig. 7.3 Estimated values of candidate object movement obtained using Kalman filter

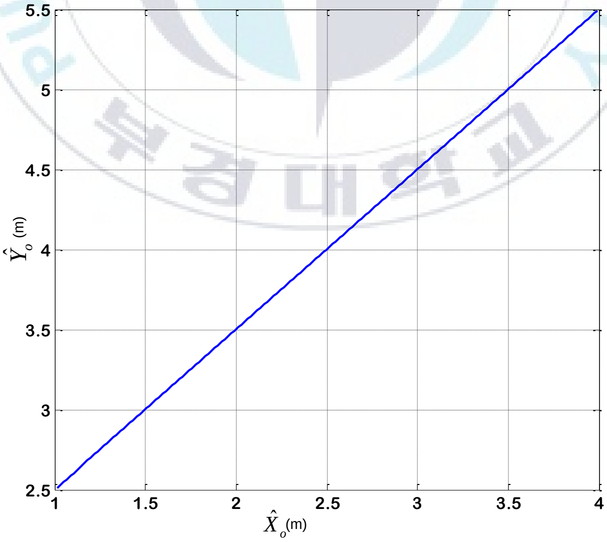


Fig. 7.4 Candidate object movement in global coordinate

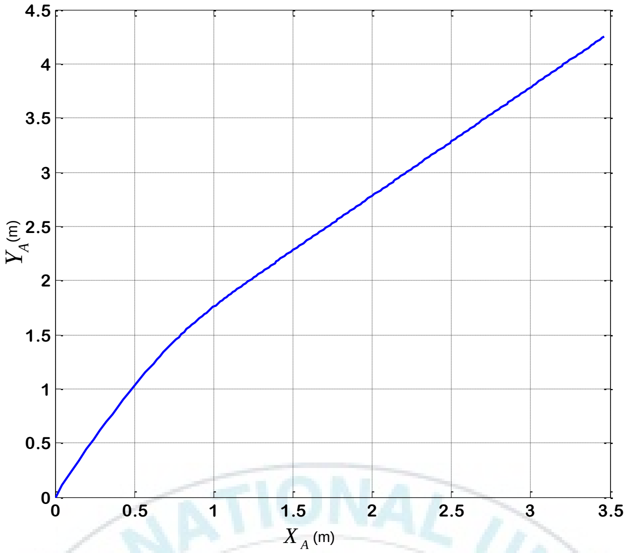
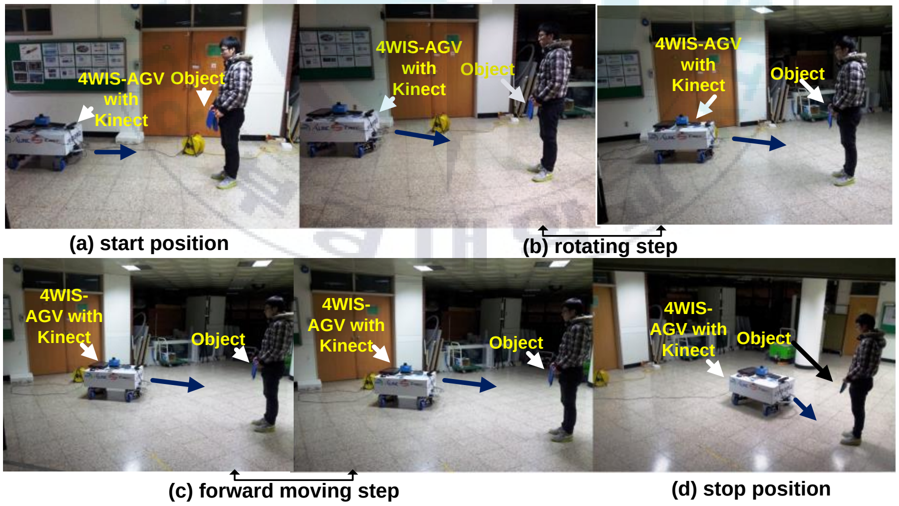


Fig. 7.5 4WIS-AGV movement in global coordinate



(a) start position

(b) rotating step

(c) forward moving step

(d) stop position

Fig. 7.6 Images showing sequence of events occurred during object following of the 4WIS-AGV system.

Chapter 8: Conclusions and Future Works

8.1 Conclusions

This thesis is about the moving object tracking and following control of the 4WIS-AGV system using Kinect camera sensor. The conclusions of this thesis are summarized as follows:

- **In chapter 2,** firstly, the basic terminologies and equations for a 4WIS-AGV system were described. Secondly, two kinds of special maneuver for the 4WIS-AGV system were described, i.e. parallel steering maneuver and zero-sideslip maneuver were introduced.
- **In chapter 3,** the hardware structure and software of the 4WIS-AGV used to implement the designed controller was described. The system was comprised of mechanical part design such as body configuration and 4 wheel configurations and electronic part design such as industrial PC tank 800, DC motors, encoders, batteries, microcontroller AVR Atmega128, motor drivers, and monitor. The developed GUI software program to control the 4WIS-AGV was described. The body configuration had many components such as body frames and body plates.

Each wheel configuration had one driving DC motor and one steering DC motor. As a result, it can move in any direction in its environment. The laser navigation system NAV-200 was used to get vehicle position data. The Kinect camera sensor was used to get the real position of the moving candidate object.

- **In chapter 4,** using laser navigation system NAV 200 sensor data, global position coordinates of the AGV were obtained. To do this task, the laser navigation system NAV 200 basic principle was presented. Using Kinect camera sensor, the real position of the blue colored moving object in local coordinate frame was obtained. To do this task, a color-based detection algorithm using Aforge.NET C# framework is used to obtain its center position in local coordinate frame using image processing method. The position of the detected object in local coordinate frame was transformed into the global coordinate frame.
- **In chapter 5,** to estimate the position and velocity coordinates of the moving candidate object and follow the moving candidate object exactly, the Kalman filter was described.

- **In chapter 6,** a controller design was proposed using backstepping method based on Lyapunov stability theory. The designed controller made the 4WIS-AGV follow the reference position and reference orientation with vehicle linear velocities and vehicle reference angular velocity with keeping a given safe distance between 4WIS-AGV and the moving object.
- **In chapter 7,** The simulation and experiment results was presented to prove the effectiveness and the applicability of the proposed controller. The Kinect-based 4WIS-AGV system could follow the moving object successfully with the proposed control algorithm. The 4WIS-AGV follows and tracks the moving object within the small tracking errors. In the simulation case, the errors $e_1, e_2, e_3 \rightarrow 0$ at 5 s, 2.5 s, 8 s, respectively. In the experimental case, error e_1 was bounded between ±0.05 m after 7 s, error e_2 was bounded between ±0.01 m after 5 s, and error e_3 was bounded between ±0.1 rad after 12 s. The 4WIS-AGV could follow the object keeping the given safe distance (1.25m) between the object and the 4WIS-AGV.

Therefore, the development results of the 4WIS-AGV used in this thesis can be employed in warehouse handing due to its omnidirectional ability and its performance mentioned above especially when material handing at narrow work space is needed.

8.2 Future Works

This thesis presented a development result of an object tracking following 4WIS-AGV system using Kinect camera sensor. However, the modification and development for upgrading and extending the system will be done totally. There are some ideas that will be considered as future works:

- Because the current system uses a color-based object detection algorithm, it has some limitations if similar color objects appear in the field of view. Therefore, to deal with this problem, a shape detection algorithm based on Kinect depth data will be developed to apply the image processing algorithm to similar colored objects.
- The proposed controller considered only kinematic modeling. The dynamic modeling of the 4WIS-AGV should be stated to improve the control performance. With dynamic modeling, the mass of the robot, the reaction force on each wheel and other parameters can be considered to design a better controller for the AGV.

References

[1] T. Tsumura, "*AGV in Japan, Recent Trends of Advanced Research, Development, and Industrial Applications*", Proceedings of the IEEE/RSJ/GI International conference on Intelligent Robots and Systems, pp. 1477 – 1484, 1994.

[2] J. H. Chou, "*Automatic Guided Vehicle*", Proceedings of the International IEEE/IAS Conference on Industrial Automation and Control: Emerging Technologies, pp. 241 – 245, 1995.

[3] T. Tsumura, "*Survey of Automated Guided Vehicle in Japanese Factory*", Proceedings of the IEEE International Conference on Robotics and Automation, pp. 1329 – 1334, 1986.

[4] L. Schulze and A. Wullner, "*The Approach of Automated Guided Vehicle Systems*", Proceedings of the IEEE International Conference on Service Operations and Logistics, and Informatics, pp. 522 – 527, 2006.

[5] D. Lecking, O. Wulf, and B. Wagner, "*Variable Pallet Pick-Up for Automatic Guided Vehicles in Industrial Environments*", Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, pp. 1169 – 1174, 2006.

[6] K. Tanaka, K. Sawada, S. Shin, K. Kumagai, and N. Yoneda, "*Modeling and Calibration of Automatic Guided Vehicle*", Proceedings of the SICE Annual Conference, pp. 2485 – 2490, 2010.

[7] F. L. Almeida, B. M. Terra, P. A. Dias, and M. Goncalves, "*Transport with Automatic Guided Vehicles in the Factory of*

the Future", Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation, pp. 1 – 4, 2010.

[8] X. Cheng and R. Tao, "*Design of Automatic Guided Vehicles and Dunking Robot System*", Proceedings of the Third International Conference on Intelligent Human-Machine Systems and Cybernetics, pp. 3 – 6, 2011.

[9] B. Sert, J. Maddox, and P. Veatch, "*Laser Assisted Intelligent Guidance for Automated Guided Vehicles*", Proceedings of Symposium on Intelligent Vehicles, pp. 201 – 206, 1993.

[10] L. Wang, J. Shu, T. Emura, and M. Kumagai, "*A 3D Scanning Laser Rangefinder and its Application to an Autonomous Guided Vehicle*", Proceedings of the 51st Vehicular Technology Conference, pp. 331 – 335, 2000.

[11] C. C. Tsai, H. H. Lin, and K. H. Wong, "*Laser-based Position Recovery of a Free-Ranging Automatic Guided Vehicle*" Proceedings of the IEEE International Conference on Mechatronics, pp. 1 – 6, 2005.

[12] G. Zhang and Z. Pan, "*The Application Research of Mobile Robots and Wireless Sensor Network in Laser Automatic Guided Vehicles*", Proceedings of the Third International Conference on Measuring Technology and Mechatronics Automation, pp. 708 – 711, 2011.

[13] K. Fan, Q. Yang, W. Yan, and B. Lei, " *Fuzzy-PID based Deviation-correcting Control System for Laser Guided AGV*", Proceedings of the International Conference on Modeling, Identification and Control, pp. 472 – 477, 2012.

[14] H. Song, J. Kim, E. Jung, J. Lee, and S. Kim, "*Path-Tracking Control of a Laser Guided Vehicle Using Fuzzy Inference*

System", Proceedings of the International Conference on Control, Automation and Systems, pp. 1666 – 1668, 2012.

[15] T. L. Bui, P. T. Doan, H. K. Kim, and S. B. Kim, "*Trajectory Tracking Controller Design for AGV Using Laser Sensor Based Positioning System*", Proceedings on 9th Asian Control Conference, pp. 1 – 5, 2013.

[16] B. F. Buxton, D. A. Castelow, M. Rygol, P. F. McLauchlan, and S. B. Pollard, "*Developing a Stereo Vision System for Control of an AGV*", Proceedings of the Second International Specialist Seminar on the Design and Application of Parallel Digital Processors, pp. 79 – 83, 1991.

[17] E. M. Petriu, N. Trif, W. S. McMath, and S. K. Yeung, "*Environment Encoding for AGV Navigation Using Vision*", Proceedings of the IEEE – IEE Vehicle Navigation and Information System Conference, pp. 525 – 528, 1993.

[18] T. Y. Sun and S. J. Tsai, "*Fuzzy Adaptive Mechanism for Improving the Efficiency and Precision of Vision-based Automatic Guided Vehicle Control*", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, pp. 263 – 269, 2005.

[19] A. Ye, H. Zhu, Z. Xu, C. Sun, and K. Yuan, "*A Vision-Based Guidance Method for Autonomous Guided Vehicles*", Proceedings of IEEE International Conference on Mechatronics and Automation, pp. 2025 – 2030, 2012.

[20] J. Kang, J. Lee, H. Eum, C. H. Hyun, and M. Park, "*An Application of Parameter Extraction for AGV Navigation Based on Computer Vision*", Proceedings of the 10th

International Conference on Ubiquitous Robots and Ambient Intelligence, pp. 622 – 626, 2013.

[21] B. Kotze, G. Jordaan, and H. Vermaak, "*Reconfigurable navigation of an Automatic Guided Vehicle 87 tilizing omnivision*", Proceedings of the 6th Robotics and Mechatronics Conference, pp. 80 – 86, 2013.

[22] X. Liang, T. Tomizawa, H. M. Do, Y. S. Kim, and K. Ohara, "*Multiple Robots Localization Using Large Planar Camera Array For Automated Guided Vehicle System*", Proceedings of the IEEE International Conference on Information and Automation, pp. 984 – 990, 2008.

[23] S. Borthwick and H. Durrant-Whyte, "*Dynamic Localisation of Autonomous Guided Vehicles*", Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 92 – 97, 1994.

[24] H. Wang and N. Liu, "*Design and Recognition of a Ring Code for AGV Localization*", Proceedings of the Fourth International Conference on Information, Communications and Signal Processing, pp. 532 – 536, 2003.

[25] X. Liang, Y. Sumi, B. K. Kim, H. M. Do, and Y. S. Kim, "*A Large Planar Camera Array for Multiple Automated Guided Vehicles Localization*", Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, pp. 608 – 613, 2008.

[26] A. Azenha and A. Carvalho, "*A Neural Network Approach for AGV Localization Using Trilateration*", Proceedings of the 35th Annual Conference of IEEE Industrial Electronics, pp. 2699 – 2702, 2009.

[27] D. Ronzoni, R. Olmi, C. Secchi, and C. Fantuzzi, “*AGV Global Localization Using Indistinguishable Artificial Landmarks*”, IEEE International Conference on Robotics and Automation, pp. 287 – 292, 2011.

[28] Q. Sun, H. Liu, Q. Yang, and W. Yan, “*On the Design for AGVs: Modeling, Path Planning and Localization*”, Proceedings of the 2011 IEEE International Conference on Mechatronics and Automation, pp. 1515 – 1520, 2011.

[29] C. Kirsch, F. Kunemund, D. Heb, and C. Rohrig, “*Comparison of Localization Algorithms for AGVs in Industrial Environments*”, Proceedings of the 7th German Conference on Robotics, pp. 1 – 6, 2012.

[30] J. Lategahn, M. Muller, and C. Rohrig, “*Global Localization of Automated Guided Vehicles in Wireless Networks*”, Proceedings of the 1st IEEE International Symposium on Wireless Systems within the Conferences on Intelligent Data Acquisition and Advanced Computing Systems, pp. 7 -12, 2012.

[31] H. H. Cho, H. J. Song, M. H. Park, J. Y. Kim, S. B. Woo and S. S. Kim, “*Independence Localization System for Mecanum Wheel AGV*”, Proceedings of the 22nd IEEE International Symposium on Robot and Human Interactive Communication, pp. 192 – 197, 2013.

[32] M. D. Cecco, “*Self-Calibration of AGV Inertial-Odometric Navigation Using Absolute-Reference Measurements*”, Proceedings of the IEEE Instrumentation and Measurement Technology Conference, pp. 1513 – 1518, 2002.

[33] R. Carelli, C. Soria, O. Nasisi, and E. Freire, “*Stable AGV Corridor Navigation with Fused Vision-Based Control Signals*”,

Proceedings of the 28th Annual Conference of the Industrial Electronics Society, pp. 2433 – 2438, 2002.

[34] Z. Li, X. Weng, and Z. Su, "*A Navigation Method of Information Fusion and Mutual Aid Based on Map for Logistic AGV*", Proceedings of the Sixth International Conference on Measuring Technology and Mechatronics Automation, pp. 21 – 24, 2014.

[35] R. Rajagopalan, R. M. H. Cheng, and S. Lequoc, "*Guidance Control Scheme Employing Knowledge-base for AGV Navigation*", Proceedings of the American Control Conference, pp. 942 – 947, 1992.

[36] R. Cucchiara, E. Perini, and G. Pistoni, "*Efficient Stereo Vision for Obstacle Detection and AGV Navigation*", Proceedings of the 14th International Conference on Image Analysis and Processing, pp. 291 – 296, 2007.

[37] R. V. Bostelman, T. H. Hong, and R. Madhavan, "*Towards AGV Safety and Navigation Advancement – Obstacle Detection Using a TOF Range Camera*", Proceedings of the 12th International Conference on Advanced Robotics, pp. 460 – 467, 2005.

[38] A. K. Ray, M. Gupta, L. Behera, and M. Jamshidi, "*Sonar Based Autonomous Automatic Guided Vehicle (AGV) Navigation*", Proceedings of the IEEE International Conference on System of Systems Engineering, pp. 1 – 6, 2008.

[39] G. G. Rigatos, "*Derivative-free Distributed Filtering for Integrity Monitoring of AGV Navigation Sensors*", Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, pp. 299 – 304, 2012.

[40] H. Wang, X. Zhang, and J. Zhang, "*GPS/DR Information Fusion for AGV Navigation*", Proceedings of the World Automation Congress (WAC), pp. 1 – 4, 2012.

[41] N. Isozaki, D. Chugo, S. Yokota, and K. Takase, "*Camera-based AGV Navigation System for Indoor Environment with Occlusion Condition*", Proceedings of the IEEE International Conference on Mechatronics and Automation, pp. 778 – 783, 2011.

[42] A. Lucas, C. Christo, M. P. Silva, and C. Cardeira, "*Mosaic Based Flexible Navigation for AGVs*", Proceedings of the IEEE International Symposium on Industrial Electronics (ISIE), pp. 3545 – 3550, 2010.

[43] W. S. Wijesoma, P. P. Khaw, and E. K. Teoh, "*Control and Navigation of an Outdoor AGV Using Fuzzy Reasoning*", Proceedings of the IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems, pp. 544 – 549, 1999.

[44] Y. Du, X. Gao, Z. Liu, and M. Sun, "*The new Navigation System for Automatic Guided Vehicle*", Proceedings of the Control and Decision Conference, pp. 4653 – 4658, 2008.

[45] P. S. Pratama, S. K. Jeong, S. S. Park, and S. B. Kim, "*Moving Object Tracking and Avoidance Algorithm for Differential Driving AGV Based on Laser Measurement Technology*", International Journal of Science and Engineering, Vol. 4, No. 1, pp. 11-15, 2013.

[46] P. T. Doan, T. T. Nguyen, V. T. Dinh, H. K. Kim, and S. B. Kim, "*Path Tracking Control of Automatic Guided Vehicle Using Camera Sensor*", Proceedings of the 1st International

Symposium on Automotive and Convergence Engineering, 2011.

[47] B. Li and F. Yu, "*Optimal Model Following Control of Four-Wheel Active Steering Vehicle*", Proceedings of the 2009 IEEE International Conference on Information and Automation, June 22 – 25, 2009, Macau, China, pp. 881-886, 2009.

[48] L. Dong and X. Lin, "*Monocular-vision-based Study on Moving Object Detection and Tracking*", 4th International Conference on New Trends in Information Science and Service Science, Gyeongju, Republic of Korea, 2010.

[49] J. Stuckler and S. Behnke, "*Combining Depth and Color Cues for Scale and Viewpoint-invariant Object Segmentation and Recognition Using Random Rorests*", International Conference on Robotics and Intelligent Systems, Taipei, Taiwan, 2010.

[50] I. H. Kim, D. E. Kim, Y. S. Cha, K. H. Lee, and T. Y. Kuc, "*An Embodiment of Stereo Vision System for Mobile Robot for Real-Time Measuring Distance and Object Tracking*", International Conference on Control, Automation and Systems, Seoul, Republic of Korea, 2007.

[51] G. Xing, S. Tian, H. Sun, W. Liu, and H. Liu, "*People-following System Design for Mobile Robots Using Kinect Sensor*", 25th Chinese Control and Decision Conference, Guiyang, China, 2013.

[52] N. Mir-Nasiri, "*Camera-based 3D Object Tracking and Following Mobile Robot*", 2006 IEEE Conference on Robotics, Automation and Mechatronics, Bangkok, Thailand, 2006.

[53] I. Amdouni, N. Jeddi, and L. E. Amraoui, "*Optimal Control Approach Developed to Four-wheel Active Steering Vehicles*",

5th International Conference on Modeling, Simulation and Applied Optimization, Hammamet, Tunisia, 2013.

[54] Z. Yang, Z. Wang, W. Su, and J. Zhang, "*Multi-mode Control Method Based on Fuzzy Selector in the Four Wheel Steering Control System*", 8th International Conference on Control and Automation, Xiamen, China, 2010.

[55] C. Chen and Y. Jia, "*Nonlinear Decoupling Control of Four-wheel-steering Vehicles with an Observer*", Int J, of Control, Auto & Sys, Vol. 10, pp. 697-702, 2012.

[56] M. F. Selekwa and J. R. Nistler, "*Path Tracking Control of Four Wheel Independently Steered Ground Robotic Vehicles*", Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), pp. 6355 – 6360, 2011.

[57] D. Wang and F. Qi, "*Trajectory Planning for a Four-Wheel-Steering Vehicle*", Proceedings of the IEEE International Conference on Robotics & Automation, pp. 3320 – 3325, 2001.

[58] Y. D. Setiawan, "*Development and Controller Design of Four Wheel Independent Steering Automatic Guided Vehicles*", Master Degree Thesis, Pukyong National University, Busan, Republic of Korea, 2015.

[59] R. A. El-laithy, J. Huang, and M. Yeh, "*Study on the Use of Microsoft Kinect for Robotics Applications*", Proceedings of the Position Location and Navigation Symposium, pp. 1280-1288, 2012.

[60] J. Han, L. Shao, D. Xu, and J. Shatton, "*Enhanced Computer Vision with Microsoft Kinect Sensor: A Review*", IEEE

Transactions on Cybernetics, Vol. 43, No. 5, pp. 1318-1334, 2013.

[61] A. V. Gulalkari, G Hoang, P. S. Pratama, H. K. Kim, S. B. Kim, and B. H. Jun, "*Object Following Control of Six-legged Robot Using Kinect Camera*", 3rd International Conference on Advances in Computing, Communications and Informatics, Delhi, India, 2014.

[62] X. Li, K. Wang, W. Wang, and Y. Li, "*A Multiple Object Tracking Method Using Kalman Filter*", 2010 IEEE International Conference on Information and Automation, Harbin, China, 2010.

Publications and Conferences

A. Conferences

[1] X. K. Ding, P. S. Pratama, H. K. Kim, and S. B. Kim, "*Moving Object Following Controller Design for Four Wheel Independent Steering Automatic Guided Vechicle Using Kinect Camera*", 6th International Symposium on Advanced Engineering(ISAE 2015) 2015. 10.22-24, Pukyong National University, Busan, South Korea, pp. 81-85, 2015.