



저작자표시-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이 저작물을 영리 목적으로 이용할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

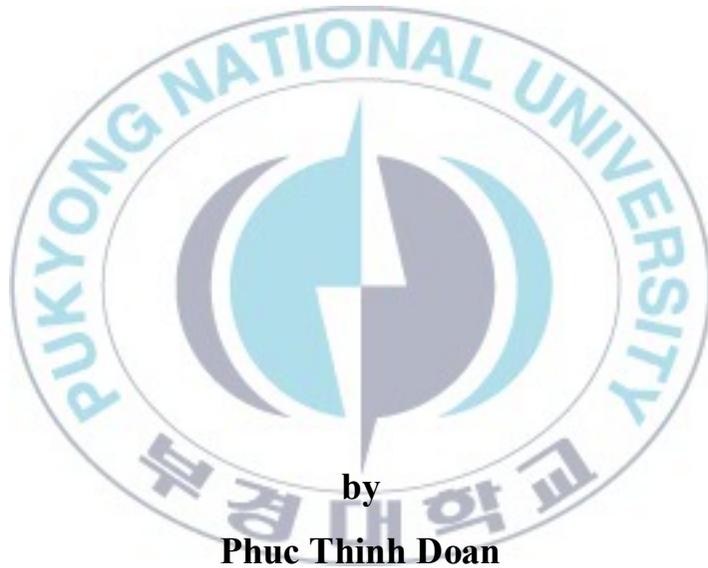
저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Thesis for the Degree of Master of Engineering

**Control of Automated Guided Vehicle for
Path Following Using Camera Sensor**



by
Phuc Thinh Doan

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

February 2011

**Control of Automated Guided Vehicle for
Path Following Using Camera Sensor**

**카메라 센서를 이용한 무인운송차량 (AGV)
의 경로 추종제어**

by

Phuc Thinh Doan

Advisor: Professor Sang Bong Kim

**A thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of**

Master of Engineering

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

February 2011

Control of Automated Guided Vehicle for Path Following Using Camera Sensor

A dissertation

By

Doan Phuc Thinh

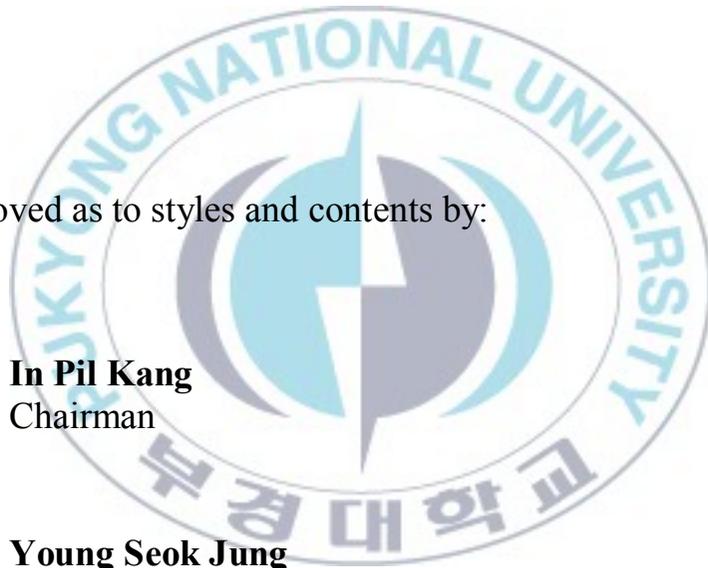
Approved as to styles and contents by:

In Pil Kang
Chairman

Young Seok Jung
Member

Sang Bong Kim
Member

February 2011



Contents

| | |
|---|-------------|
| Acknowledgements | iii |
| Abstract | v |
| List of Figures | viii |
| List of Tables | x |
| Chapter 1: Introduction | 1 |
| 1.1 Background and motivation | 1 |
| 1.2 Researching objective and method | 5 |
| 1.3 Outline of dissertation and summary of contributions | 7 |
| Chapter 2: System modeling | 9 |
| 2.2 Characterization of wheels..... | 10 |
| 2.3 Kinematic modeling..... | 11 |
| 2.4 Dynamic modeling..... | 16 |
| 2.5 Tracking error schematic of AGV | 20 |
| Chapter 3: Hardware and Software Design | 23 |
| 3.1 Hardware description..... | 23 |
| 3.2 Hardware configuration of control system of the AGV | 26 |
| 3.3 Software development | 29 |
| 3.3.1 Image processing and control algorithm calculating program | 30 |
| 3.3.2 Error measuring using camera sensor..... | 31 |

| | | |
|--|---|-----------|
| 3.3.3 | Image processing procedure..... | 32 |
| 3.3.4 | Monitoring and remote control program..... | 34 |
| Chapter 4: | Controller Design..... | 36 |
| 4.1 | Fuzzy controller design..... | 36 |
| 4.2 | Inverse kinematic..... | 39 |
| 4.3 | PID controller design..... | 40 |
| 4.4 | Full structure of the AGV control algorithm for path following using camera sensor..... | 43 |
| Chapter 5: | Simulation and Experimental Results..... | 44 |
| 5.1 | Simulation results..... | 46 |
| 5.2 | Experiment results..... | 51 |
| Chapter 6: | Conclusions and Future Works..... | 55 |
| 6.1 | Conclusions..... | 55 |
| 6.2 | Discussion and future works..... | 57 |
| References..... | | 58 |
| Publications and Conferences..... | | 64 |
| Appendix A..... | | 66 |
| Appendix B..... | | 69 |
| Appendix C..... | | 73 |

Acknowledgements

I would like to take this as an opportunity to express my gratitude to all peoples who help me in numerous ways to complete my thesis.

First, I would like to extend my sincere gratitude with a deep appreciation to my supervisor, Prof. Sang Bong Kim, for kindly offering a valuable opportunity for me to study under his supervisor and every support during the stay in Korea. I would like to thank my dear supervisor for his boundless caring, valuable advice, guidance and great encouragement.

I would like to thank the members of my thesis committee: Prof. In Pil Kang and Prof. Young Seok Jung for their considerable helpful comments and suggestions.

I am very much grateful to Prof. Hak Kyeong Kim for his guidance, enlightening discussions and tremendous encouragement towards me during working time in Korea. I could not finish my thesis soon without his great help during a long time working together.

I would like to give my thankfulness to all member of the Cimec Lab for their cooperation and for all the kindness and friendship. Specially, I would like to thank to Dr. Nguyen Hung, Mr. Nak Soon Choi, Mr. Do Kyoung Lee, Mr. Dae Won Kim, Mr. Cheol Han Park, Mr. Kyeong Mok Lee, Mr. Dae Hwan Kim, Mr. Viet Tuan Dinh, Mr. Giang Hoang, Mr. Seo Kwang Kim, Mr. Woo Young Jeung, Mr. Han Sang Jin, Miss. Bae Min Ji and Miss. Yu Mi Park who always spend

their time helping me with enthusiasm during the time I stayed in the laboratory.

I have a great appreciation and special sincere thanks to Prof. Tan Tien Nguyen from HoChiMinh City University of Technology who taught me the first steps in the field of scientific research and gave me useful advices and opportunities to study abroad.

I have a great pleasure expressing my sincere thanks to all members of Vietnamese Student's Association in Korea, especially Dr. Manh Vu Tran, Dr. Tran Van Tung, Mr. Van Phuoc Bui, Mr. Hong Thom Pham, Mr. Thank Khoa Phung, Mr. Phuoc Hoang Ho and Mr. Duc Anh Hoang for their vigorous support.

My deep appreciation and love is due to my parents, my older sisters and my younger sister for their loving support. The happy memories with my family will always be a persistence inspiration for my journey in this life.

Doan Phuc Think

27th December, Busan, Korea.

**Control of Automated Guided Vehicle for Path Following
Using Camera Sensor**

Phuc Think Doan

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

Abstract

This dissertation is about control of Automated Guided Vehicle (AGV) for path following using camera sensor. The AGV is a tricycle wheeled mobile robot. In this dissertation, the following three problems are considered. The first one is AGV's modeling. The second one is hardware and software design. The last one is control algorithm for path-following of AGV.

The AGV's modeling is reduced from a generic modeling for wheel mobile robot introduced by Guy Campion et al. . According to the Campion's method, first, characterization of wheels is presented. Based on structure and geometric characteristics of each wheel, constraints under pure rolling and no slipping condition are reduced. From these constraints, kinematic modeling is obtained. Dynamic modeling is also reduced by applying the well known Lagrange equation under the above nonholonomic constraints in the motion of the AGV.

After modeling for the proposed AGV, hardware and software design are presented. The AGV has two fixed wheels and one steering wheel. The steering wheel is on the symmetric axis of the AGV and it is directly driven by two DC motors, the first one for its orientation and the second one for its rotation. The AGV has a hoisting compartment for load transporting. Many sensors such as rotary encoder, photoelectric sensor, proximity sensor, switching sensor are used. A controller system based on the integration of a notebook and PIC-based microprocessors is developed. AGV uses a camera sensor for its navigation. An image processing procedure based on AForge.Net framework in C# programming language of Visual Studio 2008 is proposed. Two software programs, the first one used on notebook for image processing to calculate the position error and angle error between AGV's position and the reference line and the second one used on desktop PC for monitoring and remote control of AGV, are developed.

For path-following of the AGV, a controller that integrates two control loops, kinematic control loop and dynamic control loop, is designed for AGV to follow an unknown path. The kinematic control loop based on fuzzy logic framework and the dynamic control loop based on two PID controllers are designed. First of all, tracking errors are defined. The fuzzy controller uses the tracking errors as its inputs such as position error, angle error and derivative of angle error. The fuzzy outputs are derivative of linear velocity, and angular velocity at tracking point on the AGV. After that, an inverse kinematic is used to convert the fuzzy outputs into the steering angle and the rotational angular velocity of the steering wheel. Finally, in dynamic control loop, two PID controllers are used to control two DC motors for

tracking the desired values obtained from the fuzzy controller outputs and the inverse kinematic.

Simulation and experimental results are done to demonstrate the effectiveness of the proposed controller and applicability to the industrial fields of the proposed system.

Keywords: Automated Guided Vehicle, Path Following, Wheeled Mobile Robot, Steering Wheel Mobile Robot, Fuzzy.



List of Figures

| | |
|--|----|
| Fig. 1.1 Vehicle painting robot | 1 |
| Fig. 1.2 Set of six-axis robots used for welding | 1 |
| Fig. 1.3 Examples of industrial AGV | 2 |
| Fig. 2.1 Coordinate for AGV's modeling | 9 |
| Fig. 2.2 Characterization of a wheel..... | 10 |
| Fig. 2.3 Wheel schematic of AGV | 12 |
| Fig. 2.4 Tracking errors of AGV | 21 |
| Fig. 3.1 Structure of the AGV system..... | 23 |
| Fig. 3.2 DC motor driver, F250B | 24 |
| Fig. 3.3 Sensors used in the AGV system..... | 25 |
| Fig. 3.4 Camera sensor, Logitech webcam C600 | 25 |
| Fig. 3.5 Camera sensor set up on the AGV system | 26 |
| Fig. 3.6 Schematic of the control system for the AGV | 27 |
| Fig. 3.7 Display & Control Panel | 28 |
| Fig. 3.8 Manual control bar | 29 |
| Fig. 3.9 Control box in the AGV system..... | 29 |
| Fig. 3.10 Interface of the program on notebook | 30 |
| Fig. 3.11 Measuring the tracking errors using camera sensor | 32 |
| Fig. 3.12 Image processing procedure | 33 |
| Fig. 3.13 Measuring the tracking errors using camera sensor | 35 |
| Fig. 4.1 Structure of a fuzzy controller | 36 |
| Fig. 4.2 Membership functions of the proposed fuzzy controller | 37 |
| Fig. 4.3 Nine situation when the AGV follows the reference path ... | 38 |
| Fig. 4.4 3D rule viewers of Fuzzy controller | 39 |

| | |
|---|----|
| Fig. 4.5 First PID controller for tracking steering angle of the steering wheel..... | 41 |
| Fig. 4.6 Second PID controller for tracking rotational angular velocity of the steering wheel. | 42 |
| Fig. 4.7 Dynamic control loop..... | 42 |
| Fig. 4.8 Full structure of the proposed control algorithm for AGV .. | 43 |
| Fig. 5.1 Reference path | 46 |
| Fig. 5.2 Path following of AGV in simulation | 47 |
| Fig. 5.3 Position error of AGV in simulation | 48 |
| Fig. 5.4 Angle error of AGV in simulation..... | 48 |
| Fig. 5.5 Heading angle of AGV in simulation..... | 49 |
| Fig. 5.6 Steering angle of steering wheel in simulation..... | 49 |
| Fig. 5.7 Linear velocity of tracking point on AGV in simulation | 50 |
| Fig. 5.8 Torque applied for the orientation of steering wheel in simulation..... | 50 |
| Fig. 5.9 Torque applied for the rotation of steering wheel in simulation..... | 51 |
| Fig. 5.10 Position error of AGV in experiment..... | 52 |
| Fig. 5.11 Angle error of AGV in experiment. | 52 |
| Fig. 5.12 Steering angle of steering wheel in experiment..... | 53 |
| Fig. 5.13 Linear velocity of tracking point on AGV in experiment.. | 53 |

List of Tables

| | |
|--|----|
| Table 3.1 Functions displayed on the display & control panel. | 28 |
| Table 4.1 Fuzzy rules | 38 |
| Table 5.1 Numerical parameter values of AGV | 44 |
| Table 5.2 Initial values for simulation | 44 |
| Table 5.3 Saturation values in simulation..... | 45 |
| Table 5.4 Parameters in two PID controllers..... | 45 |



Chapter 1: Introduction

1.1 Background and motivation

Robotics has achieved its greatest success to date in the world of industrial manufacturing. When a robot arm is bolted at its shoulder to a specific position in the assembly line, it can move with great speed and accuracy to perform repetitive tasks such as spot painting and welding as shown in Figs. 1 and 2. In the electronic industry, manipulators place surface-mounted components at a desired position with high precision and can produce portable telephone, laptop computer, etc.



Fig. 1.1 Vehicle painting robot



Fig. 1.2 Set of six-axis robots used for welding

However, these commercial robots suffer from a fundamental disadvantage: lack of mobility. In contrast, an Automated Guided Vehicle (AGV) can move throughout the manufacturing plant. It is a transportation vehicle automatically traveling on a predefined route. AGV is often used to deliver materials around 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.



(a) Inertial-guided automatic trailer loading vehicle



(b) Tow Type AGV



(c) Laser Guided Unitload AGV

Fig. 1.3 Examples of industrial AGV

Designing an AGV involves handling many issues such as acquisition and processing of sensory data, decision making, trajectory planning and motion control.

In motion control, the main problem is to develop a good robust path following control algorithm. In fact, there are many different types of wheeled mobile robot (WMR) and their mobility configurations depend on the number and type of wheels, actuators, single or multibody structure, etc. In 1996, G. Campion et al. classified WMR into 5 types based on characteristics of wheels [2]. Among 5 types of WMR, tricycle WMR is most widely used for industrial AGV.

Several papers on the path following for tricycle AGV in presence of kinematic and dynamic constraints have been proposed. L. Gracia and J. Tornero applied a kinematic control to an industrial forklift through the position control and the inverse kinematics of wheels [6]. A. Kamga et al. proposed a trajectory tracking for a tricycle robot using linearization method [7]. Y. Bestaoui suggested an optimal velocity generation of a rear wheel drive tricycle along a specified path under assumption that the curvature of the path is known [8].

This dissertation focuses to control the path following of the AGV using fuzzy logic control. Fuzzy is a nonlinear intelligent control method imitating the logical thinking of human and independent on accurate mathematical model of a system. For these reasons, over the latest decade, fuzzy logic has been widely used for mobile robots control [12-35]. In 2007, G. Antonelli et al. proposes a path following approach for unicycle-like mobile robot using fuzzy-

logic set of rules which imitates the human driving behavior [33]. J. Baltes et al. introduces a fuzzy logic controller for car-like mobile robots [36]. This fuzzy logic controller reduced errors and also reduced control work 75% less than a traditional sliding mode controller.

Furthermore, to realize the AGV, a navigation sensor is required. There are many types of navigation sensors such as magnetic sensor, inductive sensor, laser sensor, optical sensor, camera sensor and GPS. GPS is only used in outdoor environment. Meanwhile, magnetic sensor, inductive sensor, optical sensor, laser sensor and camera sensor can be used in indoor environment. Magnetic sensor can detect the slight magnetic flux from a magnetic guide tape under the floor and inductive sensor is used to detect the electromagnetic field which is generated by the current in the loops embedded in the floor. Both of magnetic sensor and inductive sensor are durable against dust, water and light. Optic sensor is made of one infrared LED and one photo detector in pair. AGV needs at least two pairs of optic sensor to follow a given line. In comparison with other sensors, laser sensor that includes a scanner and reflectors has advantage in system flexibility. However, in the working environment without dust and water and in the clean floor, camera sensor is a good choice.

Although there are many researches on WMR, to realize an AGV system is a complex work. For the above reasons, a new research for controlling path following of AGV using camera sensor must be needed. And this is also the motivation of this dissertation.

1.2 Researching objective and method

From the above discussions, the objective of this dissertation is to develop an AGV system for path following using camera sensor. The AGV is a tricycle type of WMR. The following three problems are considered. The first one is AGV's modeling. The second one is hardware and software design and the last one is control algorithm for path-following of AGV.

The AGV's modeling is reduced from a generic modeling for wheel mobile robot introduced by Guy Campion et al.. According to the Campion's method, first, characterization of wheels is presented. Based on structure and geometric characteristics of each wheel, constraints by pure rolling and no slipping condition are reduced. From the constraints, kinematic modeling is obtained. Dynamic modeling is also reduced by applying the well known Lagrange equation under the above nonholonomic constraints in the motion of the AGV.

After modeling for the proposed AGV, hardware and software design are presented. The AGV has two fixed wheels and one steering wheel. The steering wheel is on the symmetric axis of the AGV and it is directly driven by two DC motors, the first one for its orientation and the second one for its rotation. The AGV has a hoisting compartment for load transporting. Many sensors such as rotary encoder, photoelectric sensor, proximity sensor, switching sensor are used. A controller system based on the integration of a notebook and PIC-based microcontrollers is developed. The Notebook and microcontroller communicate each other using RS232

protocol. AGV uses a camera sensor for its navigation. An image processing procedure based on AForge.Net framework in C# programming language of Visual Studio 2008 is proposed. Two software programs, the first one used on notebook for image processing to calculate the position error and angle error between AGV's position and the reference line and the second one for monitoring and remote control of AGV, are developed.

For path-following of the AGV, a controller that integrates two control loops, kinematic control loop and dynamic control loop, is designed for AGV to follow an unknown path. The kinematic control loop based on fuzzy logic framework and the dynamic control loop based on two PID controllers are designed. First of all, tracking errors are defined. The fuzzy controller uses the tracking errors as its inputs such as position error, angle error and derivative of angle error. The fuzzy outputs are derivative of linear velocity and angular velocity at a tracking point on the AGV. After that, an inverse kinematic is used to convert the fuzzy outputs into the steering angle and the rotational angular velocity of steering wheel. Finally, in dynamic control loop, two PID controllers are used to control two DC motors for tracking the desired values obtained from the fuzzy controller outputs and the inverse kinematic.

Finally, simulation and experimental results are done to demonstrate the effectiveness of the proposed controller and the applicability to industrial fields of the proposed system.

1.3 Outline of dissertation and summary of contributions

In this section, contents of the dissertation and their contributions are summarized as follows:

Chapter 1: Introduction

In this chapter, background and motivation about AGV is presented. Objective and researching method is described. Outline and summary of contribution of this dissertation are given.

Chapter 2: System Modeling

This chapter presents the modeling of kinematic and dynamic models of the AGV system based on the generic method about wheeled mobile robot introduced by Guy Campion. This chapter is divided into three sections. The first section is about characterization of wheels. The second section presents the kinematic models for AGV. Finally, the dynamic model is given in the last section.

Chapter 3: Hardware and Software Design

This chapter describes the hardware structure of the AGV system including actuators, sensors and hardware configuration of the control system. After that, two software programs are developed in C# language. This chapter is divided into three sections. The first one is hardware description. The second one is hardware configuration of control system of the AGV. The last one is software developing.

Chapter 4: Controller Design

In this chapter, a controller that integrates two control loops, kinematic control loop and dynamic control loop, is designed for

AGV to follow an unknown path. The kinematic control loop based on Fuzzy logic framework and the dynamic control loop based on two PID controllers are designed.

Chapter 5: Simulation and Experimental Results

Simulation and experimental results for AGV are given to show the effectiveness of the proposed controller.

Chapter 6: Conclusions and Future Work

Conclusions for this dissertation and some ideas for future work are presented.



Chapter 2: System modeling

This chapter presents the modeling of kinematic and dynamic models of an AGV system based on the generic method about wheeled mobile robot introduced by Guy Campion et al. [1,2]. The AGV used in this dissertation has two fixed passive wheels and one steering driving wheel. Two fixed passive wheels are installed in front of AGV and driven passively by the steering driving wheel. The steering wheel is on the symmetric axis of the AGV and it is directly driven by two DC motors, the first one for its orientation and the second one for its rotation.

Fig. 2.1 shows the coordinate for AGV's modeling where XOY is the global coordinate frame and X_QY_Q is the moving coordinate frame. A tracking point Q is an intersection point between the common axis of two fixed wheels and the symmetric axis through one steering driving wheel of the AGV.

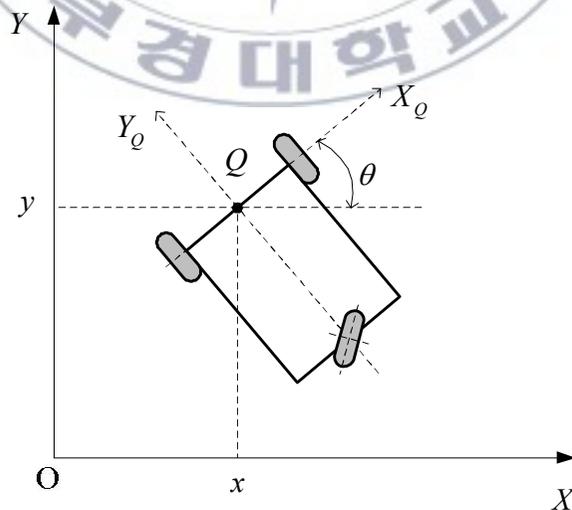


Fig. 2.1 Coordinate for AGV's modeling

The posture vector of the tracking point Q on the AGV is completely specified as follows:

$$\xi = [x \quad y \quad \theta]^T \quad (2.1)$$

where (x, y) are global coordinate of tracking point and θ is the heading angle of the AGV.

2.2 Characterization of wheels

Fig. 2.2 shows the posture of the i^{th} wheel with respect to moving coordinate frame $X_Q Y_Q$. The center of the i^{th} wheel is placed at B_i . The wheel can rotate around a vertical axis at B_i . The position of this point B_i with respect to the AGV is specified by two constants: the length l_i and the angle α_i . The radius of each wheel is R_i . The posture of the i^{th} wheel is characterized by a set of three constant, $\{R_i, l_i, \alpha_i\}$, and two varying angles for its motion, orientation of the wheel, β_i , and the rotational angle, $\phi_i(t)$. Obviously if the i^{th} wheel is fixed, the angle $\beta_i(t)$ becomes a constant.

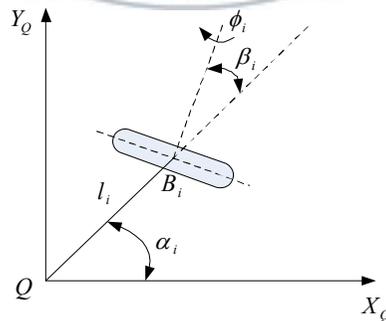


Fig. 2.2 Characterization of a wheel

It is assumed that the wheel satisfies pure rolling and no slipping conditions. In other word, the component in the plane of the wheel and the component orthogonal to the wheel about velocity at the contact point of the wheel with the ground are zero.

Pure rolling condition [1]:

$$\begin{bmatrix} -\sin(\alpha_i + \beta_i) & \cos(\alpha_i + \beta_i) & l_i \cos \beta_i \end{bmatrix} \mathbf{R}(\theta) \dot{\xi} + R_i \dot{\phi}_i = 0 \quad (2.2)$$

No slipping condition [1]:

$$\begin{bmatrix} \cos(\alpha_i + \beta_i) & \sin(\alpha_i + \beta_i) & l_i \sin \beta_i \end{bmatrix} \mathbf{R}(\theta) \dot{\xi} = 0 \quad (2.3)$$

where $R(\theta)$ is a (3×3) orthogonal rotational matrix:

$$\mathbf{R}(\theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

2.3 Kinematic modeling

According to previous description, the AGV with 3-wheels is shown in Fig. 2.3.

The tracking point Q is the middle point between the two wheels B_2 and B_3 , and X_Q is an axis through the centers of two wheels B_2 and B_3 as shown in Fig. 2.3. Point C is the center of mass of the AGV.

The geometric characteristics of three wheels are as follows:

- Wheel 1: $R_1 = r_s$, $l_1 = l$, $\alpha_1 = \frac{3\pi}{2}$, $\beta_1 = \beta$

- Wheel 2: $R_2 = r_f, l_2 = b, \alpha_2 = 0, \beta_2 = 0$
- Wheel 3: $R_3 = r_f, l_3 = b, \alpha_3 = \pi, \beta_3 = 0$

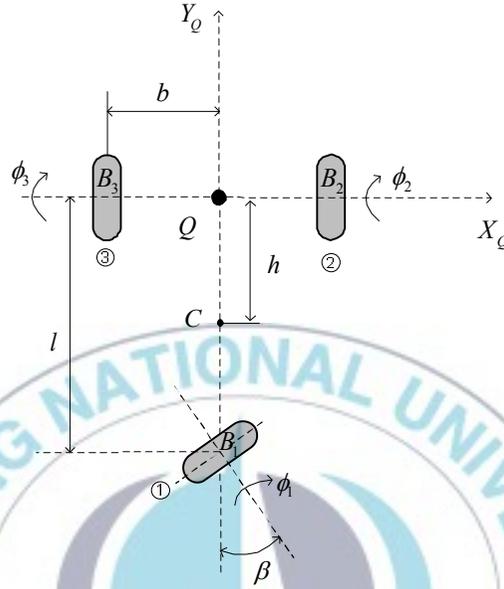


Fig. 2.3 Wheel schematic of AGV

The constraints are reduced from Eq. (2.2) under the above geometric characteristics of AGV's wheels for pure rolling condition:

$$\mathbf{J}_1(\beta)\mathbf{R}(\theta)\dot{\xi} + \mathbf{J}_2\dot{\Phi} = \mathbf{0} \quad (2.5)$$

where

$$\mathbf{J}_1(\beta) = \begin{bmatrix} \cos \beta & \sin \beta & l \cos \beta \\ 0 & 1 & b \\ 0 & -1 & b \end{bmatrix}; \quad \mathbf{J}_2 = \begin{bmatrix} r_s & 0 & 0 \\ 0 & r_f & 0 \\ 0 & 0 & r_f \end{bmatrix}; \quad (2.6)$$

$$\Phi = [\phi_1 \quad \phi_2 \quad \phi_3]^T$$

The constraints are reduced from Eq. (2.3) under the above geometric characteristics of AGV's wheels for no slipping condition:

$$\mathbf{C}_1(\beta)\mathbf{R}(\theta)\dot{\boldsymbol{\xi}} = \mathbf{0} \quad (2.7)$$

where

$$\mathbf{C}_1(\beta) = \begin{bmatrix} \sin \beta & -\cos \beta & l \sin \beta \\ 1 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \quad (2.8)$$

The configuration of the AGV can be described by seven generalized coordinates.

$$\mathbf{q} = [x \ y \ \theta \ \beta \ \phi_1 \ \phi_2 \ \phi_3]^T = [\boldsymbol{\xi}^T \ \beta \ \boldsymbol{\Phi}^T]^T \quad (2.9)$$

where the generalized coordinate \mathbf{q} is partitioned in three blocks:

$$\mathbf{q} = \begin{bmatrix} \boldsymbol{\xi} \\ \beta \\ \boldsymbol{\Phi} \end{bmatrix} \quad \text{with} \quad \begin{cases} \boldsymbol{\xi} = [x \ y \ \theta]^T \\ \boldsymbol{\Phi} = [\phi_1 \ \phi_2 \ \phi_3]^T \end{cases} \quad (2.10)$$

From Eqs. (2.5), (2.7) and (2.9), the following constrain equation is obtained.

$$\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \mathbf{0} \quad (2.11)$$

where $\mathbf{A}(\mathbf{q})$ is the (6×7) constraint matrix.

$$\mathbf{A}(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_1(\beta)\mathbf{R}(\theta) & \mathbf{0} & \mathbf{J}_2 \\ \mathbf{C}_1(\beta)\mathbf{R}(\theta) & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (2.12)$$

It is easy to check that $\text{rank}(\mathbf{A}(\mathbf{q})) = 5$. Consequently, the AGV has two degrees of freedom. Moreover, $\text{rank}(\mathbf{C}_1(\beta)) = 2$. Therefore, the two last components of Eq. (2.8) are equivalent. Without loss of

generality, by selecting the constraint corresponding to the second wheel of Eq. (2.8), the constraint matrix $\mathbf{C}_1(\beta)$ becomes

$$\mathbf{C}_1^*(\beta) = \begin{bmatrix} \sin \beta & -\cos \beta & l \sin \beta \\ 1 & 0 & 0 \end{bmatrix} \quad (2.13)$$

The corresponding constraint for no slipping condition using Eq. (2.13) instead of Eq. (2.7) is rewritten into:

$$\mathbf{C}_1^*(\beta) \mathbf{R}(\theta) \dot{\xi} = \mathbf{0} \quad (2.14)$$

and the constraint matrix $\mathbf{A}(\mathbf{q})$ of Eq. (2.11) becomes

$$\mathbf{A}(\mathbf{q}) = \begin{bmatrix} \mathbf{J}_1(\beta) \mathbf{R}(\theta) & \mathbf{0} & \mathbf{J}_2 \\ \mathbf{C}_1^*(\beta) \mathbf{R}(\theta) & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (2.15)$$

The constraint (2.14) implies that the vector $\mathbf{R}(\theta) \dot{\xi}$ belongs to the null space of the matrix $\mathbf{C}_1^*(\beta)$ as follows:

$$\mathbf{R}(\theta) \dot{\xi} \in N[\mathbf{C}_1^*(\beta)] \quad (2.16)$$

This is equivalent to the following statement. For all time, there exists a time varying scalar $\eta(t)$ such that the following equation is satisfied [Appendix C]:

$$\dot{\xi} = \mathbf{R}^T(\theta) \Sigma(\beta) \eta \quad (2.17)$$

$\Sigma(\beta)$ and η are chosen to satisfy Eq. (2.17) as follows:

$$\Sigma(\beta) = \begin{bmatrix} 0 & -r_s \sin \beta & -\frac{r_s}{l} \cos \beta \end{bmatrix}^T \quad (2.18)$$

$$\eta = \dot{\phi}_1 \quad (2.19)$$

Obviously, the matrix $\Sigma(\beta)$ depends on the steering angle β . Therefore, the posture kinematic model can be expressed as follows:

$$\dot{\xi} = \mathbf{R}^T(\theta)\Sigma(\beta)\eta \quad (2.20)$$

$$\zeta = \dot{\beta} \quad (2.21)$$

By defining $\mathbf{z} = [\xi^T \ \beta]^T$, the following is obtained.

$$\dot{\mathbf{z}} = \mathbf{B}(\mathbf{z})\mathbf{u} \quad (2.22)$$

where

$$\mathbf{u} = \begin{bmatrix} \eta \\ \zeta \end{bmatrix} \quad (2.23)$$

$$\mathbf{B}(\mathbf{z}) = \begin{bmatrix} \mathbf{R}^T(\theta)\Sigma(\beta) & \mathbf{0} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} r_s \sin \beta \sin \theta & 0 \\ -r_s \sin \beta \sin \theta & 0 \\ -\frac{r_s}{l} \cos \beta & 0 \\ 0 & 1 \end{bmatrix} \quad (2.24)$$

The remaining constraint, Eq. (2.5), is now used to derive the equations of the rotational velocities $\dot{\Phi}$ as follows:

$$\dot{\Phi} = -\mathbf{J}_2^{-1}\mathbf{J}_1(\beta)\mathbf{R}(\theta)\dot{\xi} = \mathbf{E}(\beta)\mathbf{R}(\theta)\dot{\xi} \quad (2.25)$$

where

$$\mathbf{E}(\beta) = -\mathbf{J}_2^{-1}\mathbf{J}_1(\beta) \quad (2.26)$$

By combining this equation with the posture kinematic model (2.17) and $\mathbf{R}(\theta)\mathbf{R}^T(\theta) = \mathbf{I}_{3 \times 3}$, the equation for $\dot{\Phi}$ is rewritten as

$$\dot{\Phi} = \mathbf{E}(\beta)\Sigma(\beta)\eta \quad (2.27)$$

where $\mathbf{I}_{3 \times 3}$ is the 3×3 identity matrix

From Eq. (2.26), the following equation is obtained.

$$\mathbf{J}_1(\beta) + \mathbf{J}_2\mathbf{E}(\beta) = \mathbf{0} \quad (2.28)$$

The following compact equation results from Eqs. (2.22) ~ (2.24) and (2.27) and it is called the configuration kinematic model.

$$\dot{\mathbf{q}} = \mathbf{S}(\mathbf{q})\mathbf{u} \quad (2.29)$$

where

$$\mathbf{S}(\mathbf{q}) = \begin{bmatrix} \mathbf{R}^T(\theta)\boldsymbol{\Sigma}(\beta) & \mathbf{0} \\ 0 & 1 \\ \mathbf{E}(\beta)\boldsymbol{\Sigma}(\beta) & \mathbf{0} \end{bmatrix} \text{ and } \mathbf{u} = \begin{bmatrix} \eta \\ \zeta \end{bmatrix} \quad (2.30)$$

2.4 Dynamic modeling

The potential energy is zero since it is assumed that the AGV is moving on a horizontal plane. The friction energy is ignored. Thus, the total kinetic energy of the AGV is given by [1].

$$\mathbf{T} = \frac{1}{2} \dot{\mathbf{q}}^T \begin{bmatrix} \mathbf{R}^T(\theta)\mathbf{M}\mathbf{R}(\theta) - \mathbf{R}^T(\theta)\mathbf{V} & \mathbf{0} \\ \mathbf{V}^T\mathbf{R}(\theta) & I_\beta \\ \mathbf{0} & \mathbf{0} & I_\phi \end{bmatrix} \dot{\mathbf{q}} \quad (2.31)$$

where \mathbf{M} is the 3×3 symmetric matrix defined by:

$$\begin{cases} M_{11} = M_{22} = M^* + \sum_{i=1}^3 m_i \\ M_{12} = M_{21} = 0 \\ M_{13} = M_{31} = M^* h + m_1 l_1 \\ M_{23} = M_{32} = -m_2 l_2 + m_3 l_3 \\ M_{33} = M^* h^2 + I_0 + \sum_{i=1}^3 m_i l_i^2 \end{cases} \quad (2.32)$$

$$\mathbf{V} = \begin{bmatrix} 0 & 0 & I_{p1} \end{bmatrix}^T \quad (2.33)$$

I_β and \mathbf{I}_ϕ have the following form:

$$I_\beta = I_{p1} ; \mathbf{I}_\phi = \begin{bmatrix} I_{r1} & 0 & 0 \\ 0 & I_{r2} & 0 \\ 0 & 0 & I_{r3} \end{bmatrix} \quad (2.34)$$

In Eqs. (2.32) ~ (2.34), various notations depending on the mass distribution of the AGV are used as follows:

- M^* : mass of the AGV without mass of wheels,
- m_i : mass of wheel i ,
- h : is the distance between the tracking point Q and the center of mass C . It is assumed that the center of mass is on the symmetric axis of the AGV,
- I_0 : inertial moment of the AGV without wheels around the vertical axis passing through the center of mass of the AGV,
- I_{pi} : inertial moment of wheel i around the vertical axis passing through point B_i ,
- I_{ri} : inertia moment of wheel i around its axis of rotation.
- l_i : distance between the tracking point Q and each wheel.
- M_{ij} : the element that lies in the i^{th} row and the j^{th} column of matrix M

- $i, j = 1, 2, 3$

The dynamic equation for the AGV can be obtained by applying the well known Lagrange equation for nonholonomic constraints to the motion of the AGV as follows:

$$\frac{d}{dt} \left(\frac{\partial \mathbf{T}}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial \mathbf{T}}{\partial \mathbf{q}} = \boldsymbol{\tau} + \mathbf{A}^T(\mathbf{q})\boldsymbol{\lambda} \quad (2.35)$$

where $\boldsymbol{\tau}$ is the generalized force vector applied to the AGV, $\boldsymbol{\lambda}$ is the Lagrange multiplier vector.

The following compact notation is defined.

$$[\mathbf{T}]_{\mathbf{q}} = \frac{d}{dt} \left(\frac{\partial \mathbf{T}}{\partial \dot{\mathbf{q}}} \right) - \frac{\partial \mathbf{T}}{\partial \mathbf{q}} \quad (2.36)$$

By separating \mathbf{q} into three part, ξ , β and Φ the following equations are obtained.

$$[\mathbf{T}]_{\xi} = \mathbf{R}^T(\theta) \mathbf{J}_1^T(\beta) \boldsymbol{\lambda}_1 + \mathbf{R}^T(\theta) \mathbf{C}_1^{*T}(\beta) \boldsymbol{\lambda}_2 \quad (2.37)$$

$$[\mathbf{T}]_{\beta} = \tau_1 \quad (2.38)$$

$$[\mathbf{T}]_{\Phi} = \mathbf{J}_2^T \boldsymbol{\lambda}_1 + \mathbf{P} \tau_2 \quad (2.39)$$

where $\boldsymbol{\lambda}_1 = [\lambda_{11} \ \lambda_{12} \ \lambda_{13}]^T$ and $\boldsymbol{\lambda}_2 = [\lambda_{21} \ \lambda_{22}]^T$ are the Lagrange multipliers vectors associated with the constrains (2.5) and (2.13), respectively. τ_1 and $\mathbf{P} \tau_2$ are the torques applied to the steering wheel for its orientation and rotation, respectively. \mathbf{P} is given by:

$$\mathbf{P} = [1 \ 0 \ 0]^T \quad (2.40)$$

The Lagrange multipliers are eliminated by adding Eq. (2.37) and Eq. (2.39) premultiplied by the matrices $\boldsymbol{\Sigma}^T(\beta) \mathbf{R}(\theta)$ and

$\Sigma^T(\beta)\mathbf{E}^T(\beta)$, respectively using Eq. (2.28) and $\Sigma^T(\beta)\mathbf{C}_1^{*T}(\beta) = \mathbf{0}$. As the result, the following are obtained.

$$\begin{aligned} & \Sigma^T(\beta)\mathbf{R}(\theta)[\mathbf{T}]_{\xi} + \Sigma^T(\beta)\mathbf{E}^T(\beta)[\mathbf{T}]_{\phi} \\ & = \Sigma^T(\beta)\mathbf{E}^T(\beta)\mathbf{P}\tau_2 \end{aligned} \quad (2.41)$$

$$[\mathbf{T}]_{\beta} = \tau_1 \quad (2.42)$$

Substituting \mathbf{T} from (2.31) to (2.36) can be obtained the following [Appendix A]:

$$\begin{aligned} [\mathbf{T}]_{\xi} &= \mathbf{R}^T(\theta)(\Psi^T\mathbf{M} + \mathbf{M}\Psi)\mathbf{R}(\theta)\dot{\xi}\dot{\theta} + \mathbf{R}^T(\theta)\mathbf{M}\mathbf{R}(\theta)\ddot{\xi} \\ &+ \mathbf{R}^T(\theta)\Psi^T\mathbf{V}\dot{\beta}\dot{\theta} + \mathbf{R}^T(\theta)\mathbf{V}\ddot{\beta} \\ &- \mathbf{D}(\dot{\xi}^T\mathbf{R}^T(\theta)\mathbf{M} + \dot{\beta}\mathbf{V}^T)\Psi\mathbf{R}(\theta)\dot{\xi} \end{aligned} \quad (2.43)$$

$$[\mathbf{T}]_{\beta} = \mathbf{V}^T\mathbf{R}(\theta)\ddot{\xi} + I_{\beta}\ddot{\beta} \quad (2.44)$$

$$[\mathbf{T}]_{\phi} = \mathbf{I}_{\phi}\ddot{\Phi} \quad (2.45)$$

From the kinematic equations (2.20), (2.21), (2.27), the followings are obtained.

$$\dot{\mathbf{q}} = \begin{cases} \dot{\xi} = \mathbf{R}^T(\theta)\Sigma(\beta)\eta \\ \dot{\beta} = \zeta \\ \dot{\Phi} = \mathbf{E}(\beta)\Sigma(\beta)\eta \end{cases} \quad (2.46)$$

From Eq. (2.46), the second order time derivatives of \mathbf{q} is shown as follows[Appendix B]:

$$\ddot{\mathbf{q}} = \begin{cases} \ddot{\xi} = \mathbf{R}^T(\theta)\Sigma(\beta)\dot{\eta} + \mathbf{k}_1(\theta, \beta, \eta, \zeta) \\ \ddot{\beta} = \dot{\zeta} \\ \ddot{\Phi} = \mathbf{E}(\beta)\Sigma(\beta)\dot{\eta} + \mathbf{k}_2(\beta, \eta, \zeta) \end{cases} \quad (2.47)$$

Eliminating the velocities $\dot{\xi}, \dot{\beta}, \dot{\Phi}$ and the acceleration $\ddot{\xi}, \ddot{\beta}, \ddot{\Phi}$ in (2.41) and (2.42) yields [Appendix B]:

$$\mathbf{H}(\beta)\dot{\eta} + \Sigma^T(\beta)\mathbf{V}\dot{\zeta} + f_2(\theta, \beta, \eta, \zeta) = \Sigma^T(\beta)\mathbf{E}^T(\beta)\mathbf{P}\tau_2 \quad (2.48)$$

$$\mathbf{V}^T\Sigma(\beta)\dot{\eta} + I_\beta\dot{\zeta} + f_1(\theta, \beta, \eta, \zeta) = \tau_1 \quad (2.49)$$

where

$$\mathbf{H}(\beta) = \Sigma^T(\beta)(\mathbf{M} + \mathbf{E}^T(\beta)\mathbf{I}_\phi\mathbf{E}(\beta))\Sigma(\beta) \quad (2.50)$$

f_1 and f_2 are scalar functions depending on θ, β, η and ζ .

Finally, the dynamic model and the kinematic model of the AGV given by (2.46), (2.48) and (2.49) are summarized as follows:

$$\begin{cases} \mathbf{V}^T\Sigma(\beta)\dot{\eta} + I_\beta\dot{\zeta} + f_1(\theta, \beta, \eta, \zeta) = \tau_1 \\ \mathbf{H}(\beta)\dot{\eta} + \Sigma^T(\beta)\mathbf{V}\dot{\zeta} + f_2(\theta, \beta, \eta, \zeta) = \Sigma^T(\beta)\mathbf{E}^T(\beta)\mathbf{P}\tau_2 \\ \dot{\xi} = \mathbf{R}^T(\theta)\Sigma(\beta)\eta \\ \dot{\beta} = \zeta \\ \dot{\Phi} = \mathbf{E}(\beta)\Sigma(\beta)\eta \end{cases} \quad (2.51)$$

2.5 Tracking error schematic of AGV

When AGV follows an unknown reference line, a point Q is a tracking point. Schematic of tracking errors of AGV in global coordinate is shown as in Fig. 2.4.

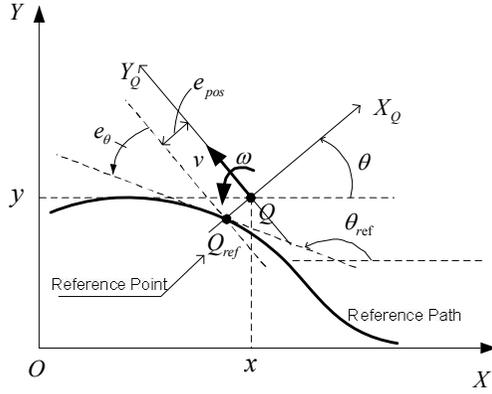


Fig. 2.4 Tracking errors of AGV

In Fig. 2.4, e_{pos} is the position error between the position of the tracking point $Q(x, y)$ on the AGV and the reference point, $Q_{ref}(x_{ref}, y_{ref})$, on the reference path. Q_{ref} is the intersection point of the extension line of X_Q axis and the reference path. e_{θ} is the angle error between the Y_Q axis and the tangent line of the path at the reference point, Q_{ref} .

A posture vector of the tracking point Q on the AGV is denoted as $\xi_Q = [x_Q \quad y_Q \quad \theta_Q]^T$. Therefore, the equation of the extension line of X_Q axis is given as follows.

$$(x - x_Q) \sin \theta_Q - (y - y_Q) \cos \theta_Q = 0 \quad (2.52)$$

The reference path is unknown. The equation of the reference path shown in Eq. 2.53 is denoted as:

$$y = \delta(x) \quad (2.53)$$

Therefore, $x_{Q_{ref}}$ and $y_{Q_{ref}}$ is the solution of the following equations.

$$\begin{cases} y = \delta(x) \\ (x - x_Q) \sin \theta_Q - (y - y_Q) \cos \theta_Q = 0 \end{cases} \quad (2.54)$$

$\theta_{Q_{ref}}$ denotes for the angle of tangent line of $\delta(x)$ at Q_{ref}

$$\theta_{Q_{ref}} = \arctan \left(\left. \frac{d\delta(x)}{dx} \right|_{x=x_{Q_{ref}}} \right) \quad (2.55)$$

The position error and the angle error are obtained by following equations.

$$\begin{cases} e_{pos} = \sqrt{(x_Q - x_{Q_{ref}})^2 + (y_Q - y_{Q_{ref}})^2} \\ e_{\theta} = \left(\frac{\pi}{2} + \theta_Q \right) - \theta_{Q_{ref}} \end{cases} \quad (2.56)$$



Chapter 3: Hardware and Software Design

First, this chapter describes the hardware structure of the AGV system including actuators, sensors and hardware configuration of the control system. Second, two software programs, the first one used on notebook for image processing to calculate the position error and angle error between AGV's position and the reference line and the second one used on desktop PC for monitoring and remote control of AGV, are developed.

3.1 Hardware description

Fig. 3.1 shows the structure of an AGV system used for this dissertation.



Fig. 3.1 Structure of the AGV system.

The AGV has three wheels, two fixed wheels and one steering wheel. The steering wheel is directly driven by two DC motors, the first one for its orientation and the second one for its rotation. The hoisting compartment is driven by a hydraulic actuator.

Each DC motors using 24VDC power supplier is driven by each motor driver, model F250B as shown in Fig. 3.2. Maximum current allowed by the driver is up to 250 A with 24VDC.



Fig. 3.2 DC motor driver, F250B

Fig. 3.3 shows the sensors used in AGV system. For obstacle avoidance, two photoelectric sensors in (a) are installed in front of AGV. Two other photoelectric sensors and two switching sensors in (b) are installed in rear of AGV. Two rotary encoders in (c) are used for measuring AGV's speed and steering angle of the steering wheel. A proximity sensor in (d) is used to detect the original point of the steering angle. Two other switching sensors are used for left-limit and right-limit of the steering wheel, and two more switching sensors are used for up-limit and down-limit of the hoisting component.



Fig. 3.3 Sensors used in the AGV system.

Fig. 3.4 shows Camera sensor, Logitech webcam C600, in this system. Camera sensor is used for measuring position error. The webcam specification is up to 30 frames per second, 2.0-megapixel sensor and Hi-Speed USB 2.0 communication.



Fig. 3.4 Camera sensor, Logitech webcam C600

Fig. 3.5 shows the camera sensor that is set up on the AGV system. The camera is covered by a plastic box. On the box, there are a lot of LEDs installed around the camera that provide the light for the camera.

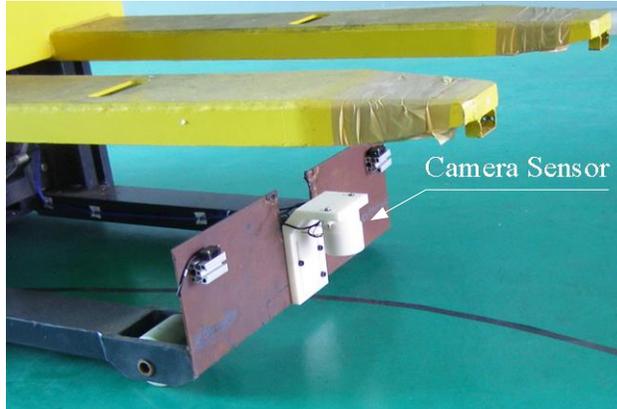


Fig. 3.5 Camera sensor set up on the AGV system

3.2 Hardware configuration of control system of the AGV

Fig. 3.6 shows the schematic of a control system for the AGV. As illustrated in Fig. 3.6, the control system is developed based on the integration of a notebook (Intel Core 2 Dual, 2.8 GHz, 2 GB RAM) and PIC-based microprocessors. The control system is composed of two parts: high level computer control and low level microprocessor control. The former is used for image processing and control algorithm. The latter is used for device control. Notebook and microprocessor communicate each other through RS232 protocol. A desktop personal computer (PC) that communicates with the AGV through wireless communication Bluetooth module, Promi SD202, is used to monitor and control the AGV remotely.

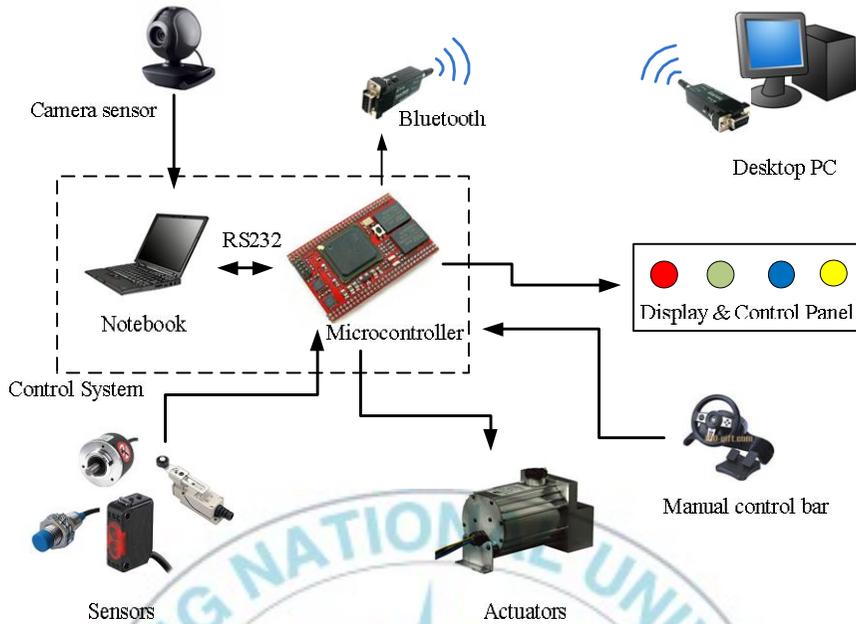


Fig. 3.6 Schematic of the control system for the AGV

The AGV can operate in the following three modes

- Auto mode: the AGV moves from position A to position B by following a predefined route automatically.
- Semi auto mode: the angle of steering wheel is controlled automatically. Meanwhile, other functions such as going forward, going backward and stop of AGV, lifting up, lifting down and stop of the lift part are controlled manually.
- Manual mode: the controller is not used. The AGV is controlled manually.

The Display & Control Panel in Fig. 3.7 is used for displaying AGV's operating status and selecting operating mode. There are 16 lamps. Their functions are shown in Table 3.1.



Fig. 3.7 Display & Control Panel

Table 3.1 Functions displayed on the display & control panel.

| No. | Function | No. | Function |
|-----|------------|-----|---------------------|
| 1 | Power | 9 | Auto |
| 2 | Manual | 10 | Semi Auto |
| 3 | Forward | 11 | Battery empty |
| 4 | Backward | 12 | Controller error |
| 5 | Turn left | 13 | Communication error |
| 6 | Turn right | 14 | Line off |
| 7 | Lift up | 15 | Emergency stop |
| 8 | Lift down | 16 | Spare |

In manual mode, a manual control bar as shown in Fig. 3.8 is used to control the AGV manually when the controller doesn't work.

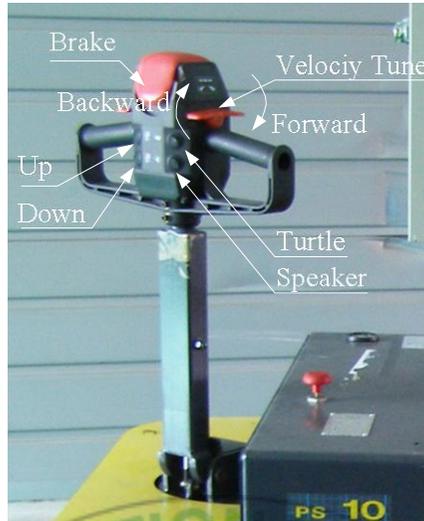


Fig. 3.8 Manual control bar

Fig. 3.9 shows the developed control box in the AGV system. The controller uses three microcontroller PIC18F452s and 5VDC power supplier.

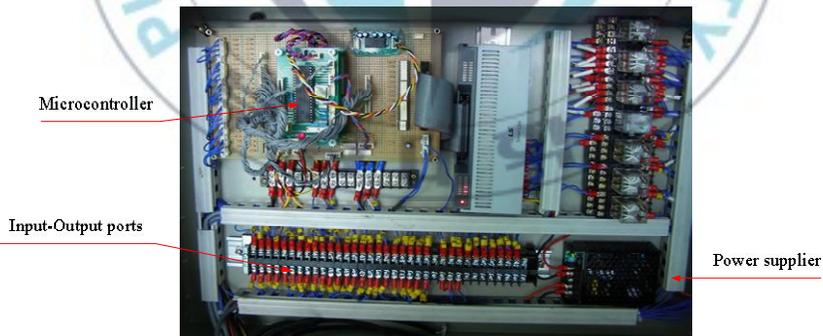


Fig. 3.9 Control box in the AGV system.

3.3 Software development

In this dissertation, two programs are developed using C# language. The first one is used on notebook (Intel core 2 Dual, 2.8

GHZ, 2GB RAM) for image processing and calculating of control algorithm. First, image processing is used to calculate the position error between AGV position and the reference line and angle error. After that, based on the errors, control algorithm is applied. The second one is used on desktop PC for monitoring and remote control of the AGV.

3.3.1 Image processing and control algorithm calculating program

Fig. 3.10 shows the interface of the program on notebook. On the interface, there are 4 partitions.

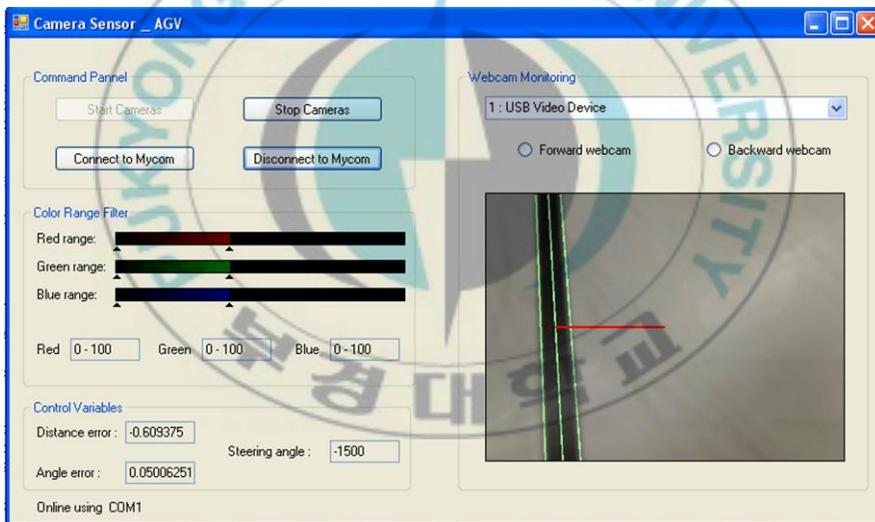


Fig. 3.10 Interface of the program on notebook

The first partition is Command Panel. It includes functions as follows:

- Start Cameras: the camera begins to capture the pictures.
- Stop Cameras: the camera stops capturing the pictures.

- Connect to Mycom: open COM port and begin communicating with Microprocessor.
- Disconnect to Mycom: close COM port and stop communicating with Microprocessor.

The second partition is Color Range Filter. In this partition, the ranges of RGB parameters are setup depending on the color of the reference line.

The third one is Control Variables. In this partition, control variables include two control inputs such as distance error, angle error and a control output such as steering angle are displayed.

The last partition is Webcam Monitoring. In this section, the pictures from camera are displayed. From the images, the left edge, the right edge, the center line of the reference line and distance error are shown.

3.3.2 Error measuring using camera sensor

To achieve the tracking errors, a camera sensor is used. The tracking point Q on the AGV is coincided with the center of the camera frame. Fig. 3.11 shows the schematic for measuring the tracking errors using camera sensor.

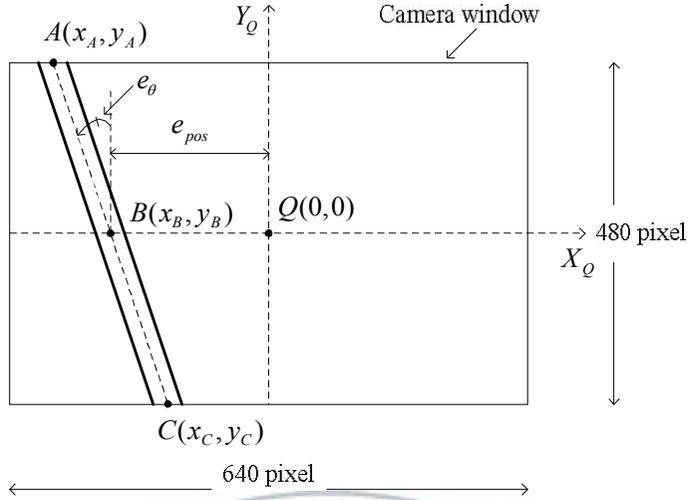


Fig. 3.11 Measuring the tracking errors using camera sensor

Form Fig. 3.11, the tracking errors can be obtained as follows:

$$\begin{cases} e_{pos} = x_Q - x_B \\ e_{\theta} = \arctan\left(\frac{(x_C - x_A)}{(y_C - y_A)}\right) \end{cases} \quad (3.1)$$

3.3.3 Image processing procedure

Images captured from webcam are processed by using AForge.NET Framework, a C# framework designed for developers and researchers in the fields of Computer Vision and Artificial Intelligence - image processing, computer vision, neural networks, genetic algorithms, machine learning, etc. [37].

To calculate position error and angle error form the images captured by camera, image processing procedure includes steps as shown in Fig. 3.12.

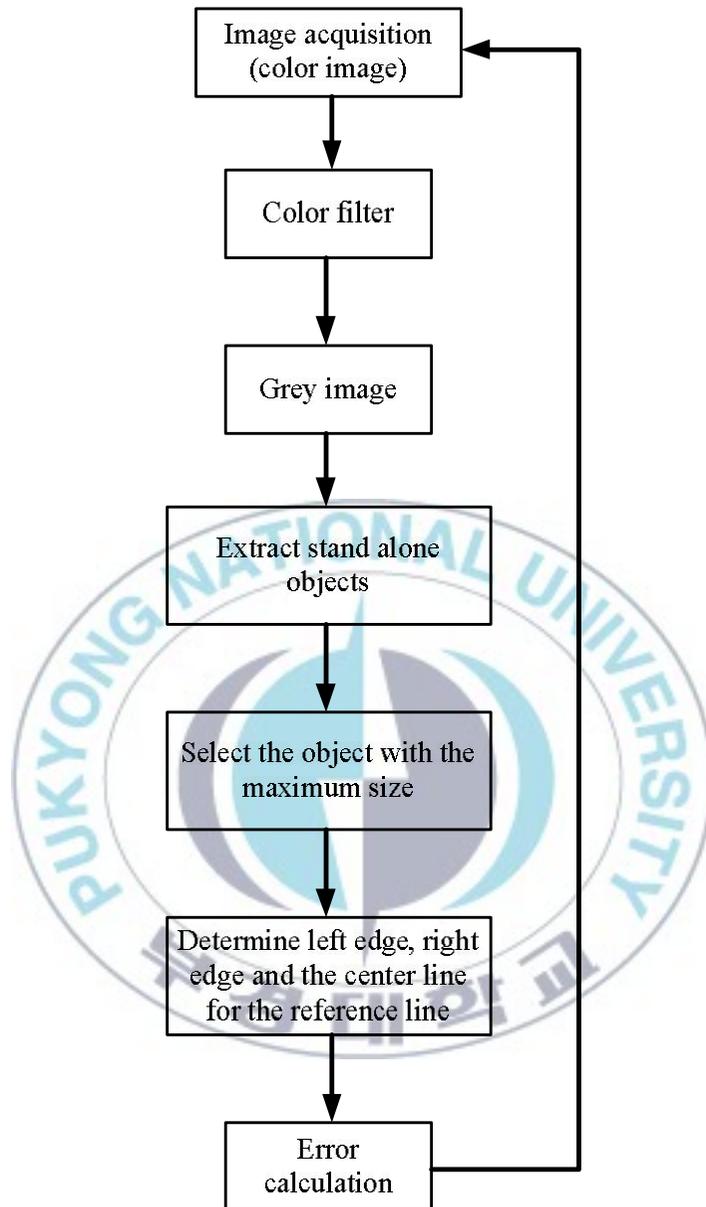


Fig. 3.12 Image processing procedure

The procedure in Fig. 3.12 can be explained as follows:

- First, the color filter filters pixels inside/outside of specified RGB color range. It keeps pixels with colors inside of the

specified range and fills pixels with colors outside of the specified range with specified color (e.g. black color).

- Second, the result image of the first step is changed to grey image.
- Third, extracting objects in grey image are separated by black background. If the size of an object is smaller than the limit minimum size, the object is ignored.
- Fourth, only the object with the maximum size is selected. The others are ignored. The object with the maximum size is the reference line.
- Fifth, left edge and right edge of the selected object are determined. After that, the center line of the reference line is determined.
- Finally, from the position of the center line on the image frame and the center of reference line, the position error and the angle error are calculated.

3.3.4 Monitoring and remote control program

Fig. 3.13 shows the interface of the program used on desktop PC for AGV's remote control and monitoring. On the interface, there are 4 partitions.

The first partition is Control Mode. It includes three radio buttons and one slide bar. Three radio buttons is used for selecting operating mode of the AGV consisting of Auto mode, Semi-Auto mode and Manual mode, respectively. The sliding bar is used to set the velocity of the AGV.

The second partition is Running Mode. This section is used to send commands to the AGV such as going forward, going backward, turning left, turning right and stop moving.

The next partition is Lift Operating. This partition is used to send commands to the hoisting compartment such as lift up, lift down and stop lifting.

The last partition in the right side of the interface is used to monitoring AGV operating status. It is synchronous with the corresponding lamp on the display panel of the AGV.

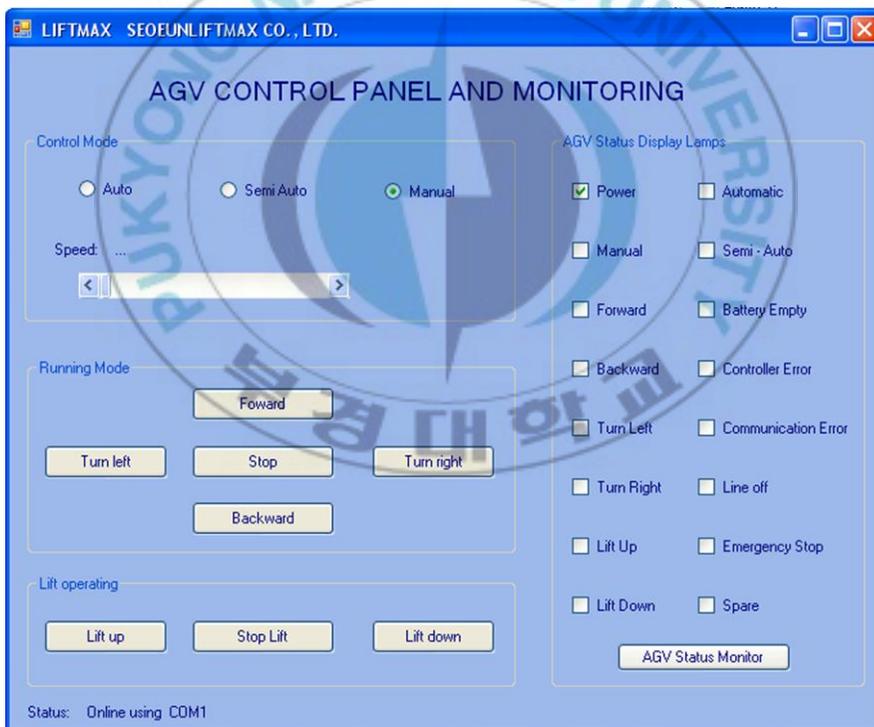


Fig. 3.13 Measuring the tracking errors using camera sensor

Chapter 4: Controller Design

In this chapter, a controller that integrates two control loops, kinematic control loop and dynamic control loop, is designed for AGV to follow an unknown path. The kinematic control loop based on fuzzy logic framework and the dynamic control loop based on two PID controllers are considered.

4.1 Fuzzy controller design

The fuzzy controller uses the tracking errors defined in section 2.5 as its inputs. There are three inputs including position error, e_{pos} , angle error, e_{θ} , and derivative of angle error, \dot{e}_{θ} . The fuzzy outputs are derivative of linear velocity, dv_{ref} , and angular velocity, ω_{ref} , of the AGV at the tracking point Q . Fig. 4.1 shows structure of a fuzzy controller.

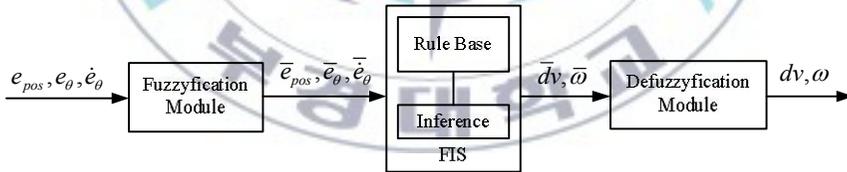


Fig. 4.1 Structure of a fuzzy controller

A fuzzy controller is composed of a Fuzzyfication Module, a Fuzzy Inference System (FIS) and a Defuzzyfication Module.

The Defuzzyfication Module uses centroid method and the FIS uses max-min method.

In addition to the position error and the angle error, the derivative of angle error is also very important, especially when the AGV passes

through sharp and narrow turns. The outputs of the fuzzy controller are derivative of linear velocity, dv , and angular velocity, ω , of the AGV at the tracking point Q . The membership functions are given as follows:

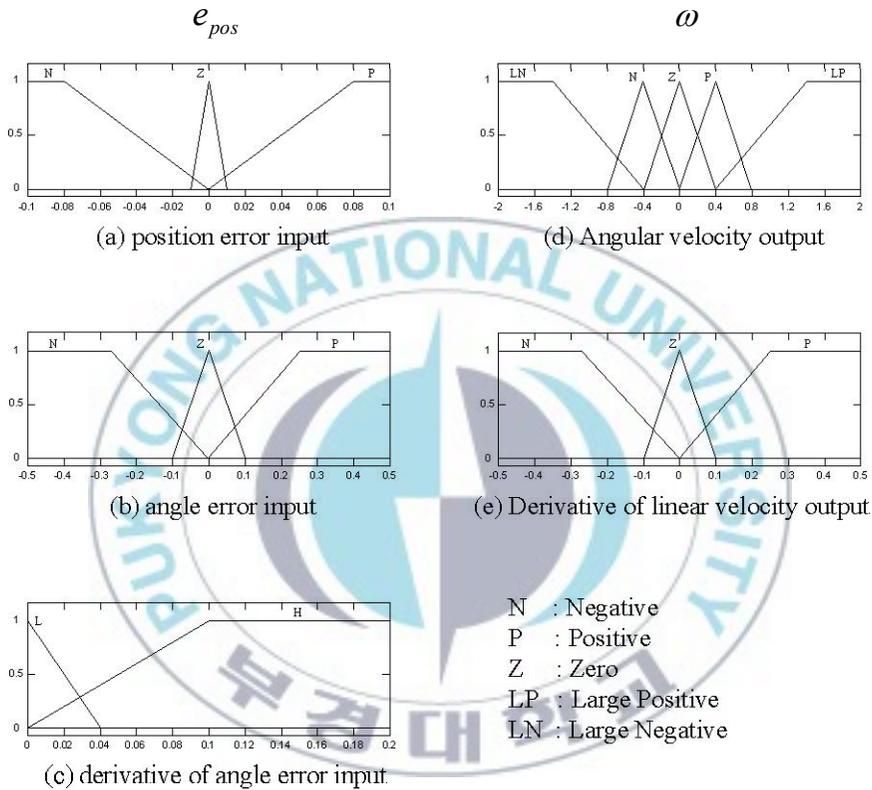


Fig. 4.2 Membership functions of the proposed fuzzy controller

Fuzzy rules are normally created based on human experience and logic. Therefore, it is necessary to analyze the desired outputs based on available inputs. From two inputs, the position error and the angle error, AGV posture is defined in 9 different situations as shown in Fig. 4.3. From 9 situations, 9 rules are generated, respectively. In addition to the nine rules, two other rules are proposed when

derivative of angle error are large, especially while the robot moves away the path with high velocity.

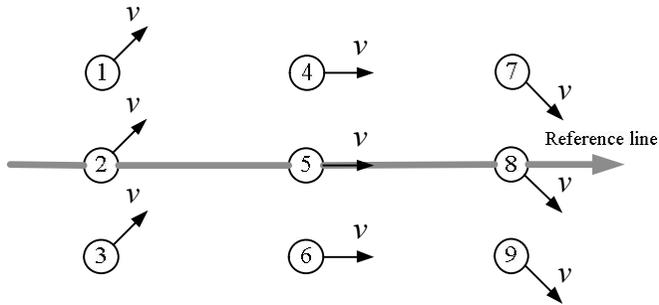


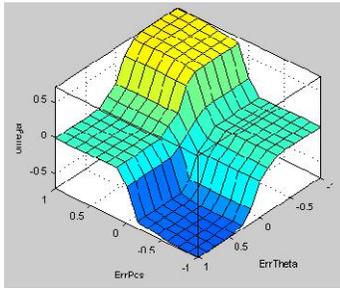
Fig. 4.3 Nine situation when the AGV follows the reference path

Table 4.1 shows the fuzzy rules designed for AGV for path following.

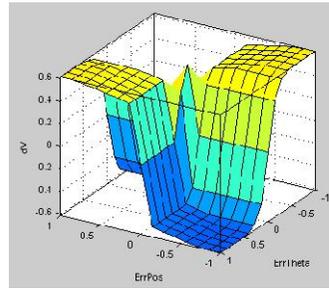
Table 4.1 Fuzzy rules

| Rule | Inputs | | | Outputs | |
|------|-----------|--------------|--------------------|----------|------|
| | e_{pos} | e_{θ} | \dot{e}_{θ} | ω | dv |
| 1 | N | P | L | N | N |
| 2 | Z | P | None | N | Z |
| 3 | P | P | None | Z | P |
| 4 | N | Z | None | N | Z |
| 5 | Z | Z | None | Z | P |
| 6 | P | Z | None | P | Z |
| 7 | N | N | None | Z | P |
| 8 | Z | N | None | P | Z |
| 9 | P | N | L | P | N |
| 10 | N | P | H | LN | N |
| 11 | P | N | H | LP | N |

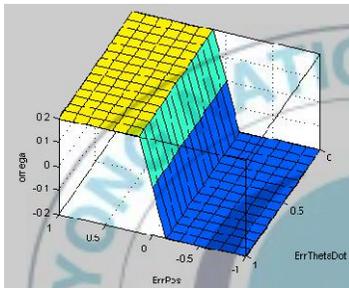
Fig. 4.4 shows 3D rule viewers of the fuzzy controller



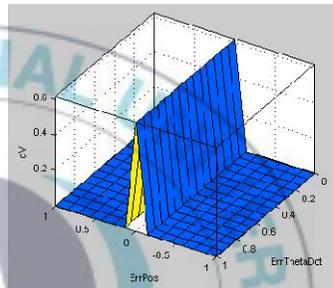
(a) e_{pos} , e_{θ} and ω



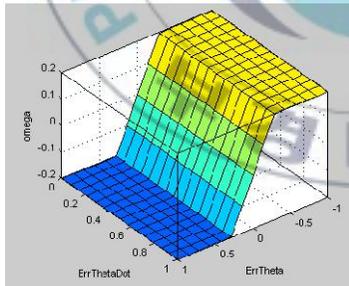
(b) e_{pos} , e_{θ} and dv



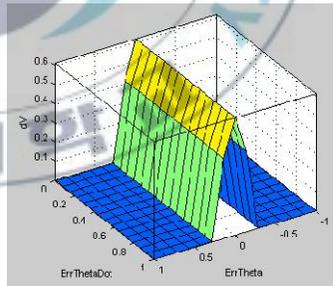
(c) e_{pos} , \dot{e}_{θ} and ω



(d) e_{pos} , \dot{e}_{θ} and dv



(e) \dot{e}_{θ} , e_{θ} and ω



(f) \dot{e}_{θ} , e_{θ} and dv

Fig. 4.4 3D rule viewers of Fuzzy controller

4.2 Inverse kinematic

First of all, an integration is used to obtain v from dv . After that, an inverse kinematic is applied to convert v and ω obtained from the kinematic control loop into η_d and β_d . Where β_d and η_d are denoted as desired steering angle and desired rotation angular velocity of steering wheel that are reference value used in dynamic control loop. The inverse kinematic is given as follows:

Because the linear velocity vector, v , is in the same direction with Y_Q axis of the moving coordinate, the following is obtained.

$$v = -\dot{x} \sin \theta + \dot{y} \cos \theta \quad (4.1)$$

From Eq. (2.17) ~ (2.19), Eq. (4.1) and Eq. (C.2), the followings are obtained.

$$\begin{cases} v = -r_s \eta_d \sin \beta_d \\ \omega = \dot{\theta} = -\frac{r_s}{l} \eta_d \cos \beta_d \end{cases} \quad (4.2)$$

Form Eq. (4.2), the followings are obtained.

$$\begin{cases} \eta_d = -\frac{1}{r_s} (v \sin \beta_d + \omega l \cos \beta_d) \\ \beta_d = \arctan \left(\frac{v}{\omega l} \right) \quad \text{if } \omega \neq 0 \\ \beta_d = \frac{\pi}{2} \quad \text{if } \omega = 0 \end{cases} \quad (4.3)$$

4.3 PID controller design

The dynamic control loop includes two PID controllers. The first one is used for tracking the desired steering angle, β_d . The second

one is used for tracking the desired rotation angular velocity of steering wheel, η_d .

Fig. 4.5 shows the first PID controller for tracking the desired steering angle, β_d . The input of the first PID controller is the deviation between the steering angle of the steering wheel, β , and the desired steering angle, β_d and the output is the torque, τ_1 , generated by DC motor for orientation of the steering wheel.

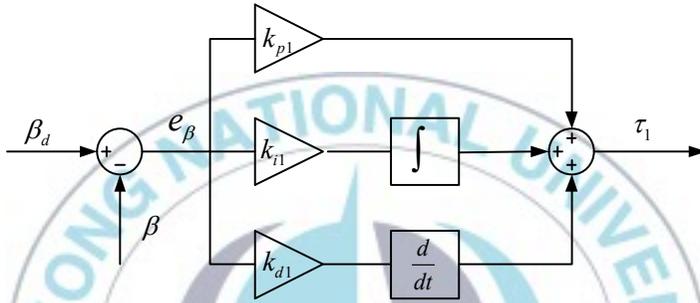


Fig. 4.5 First PID controller for tracking steering angle of the steering wheel

The PID controller 1 is designed as follows:

$$\tau_1 = k_{p1}e_\beta + k_{i1}\int e_\beta + k_{d1}\frac{de_\beta}{dt} \quad (4.4)$$

$$e_\beta = \beta_d - \beta \quad (4.5)$$

where k_{p1} , k_{i1} and k_{d1} are the coefficients of the PID controller 1.

Fig. 4.8 shows the second PID controller for tracking the desired rotation angular velocity of steering wheel, η_d . The input of the second PID controller is the deviation between the rotational angular velocity of the steering wheel, η , and its desired value, η_d . The

output is the torque, τ_2 , generated by DC motor for rotation of the steering wheel.

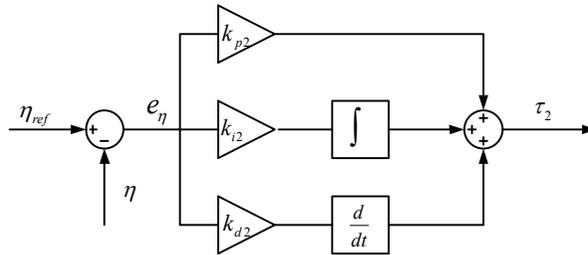


Fig. 4.6 Second PID controller for tracking rotational angular velocity of the steering wheel.

The PID controller 2 is designed as follows:

$$\tau_2 = k_{p2}e_\eta + k_{i2} \int e_\eta + k_{d2} \frac{de_\eta}{dt} \quad (4.6)$$

$$e_\eta = \eta_d - \eta \quad (4.7)$$

where k_{p2} , k_{i2} and k_{d2} are the coefficients of the PID controller 2.

The dynamic control loop is now shown in Fig. 4.7

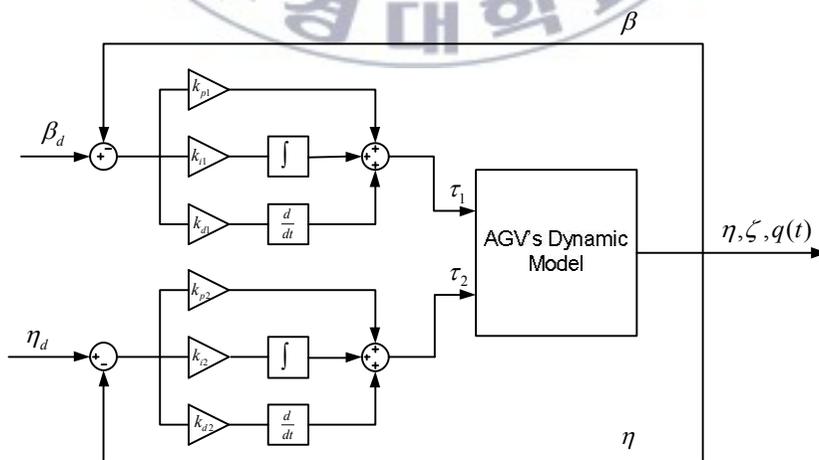


Fig. 4.7 Dynamic control loop

4.4 Full structure of the AGV control algorithm for path following using camera sensor

Fig. 4.8 shows the full structure of the proposed control algorithm for AGV to follow the path using camera sensor. The controller integrates two control loops, kinematic control loop and dynamic control loop.

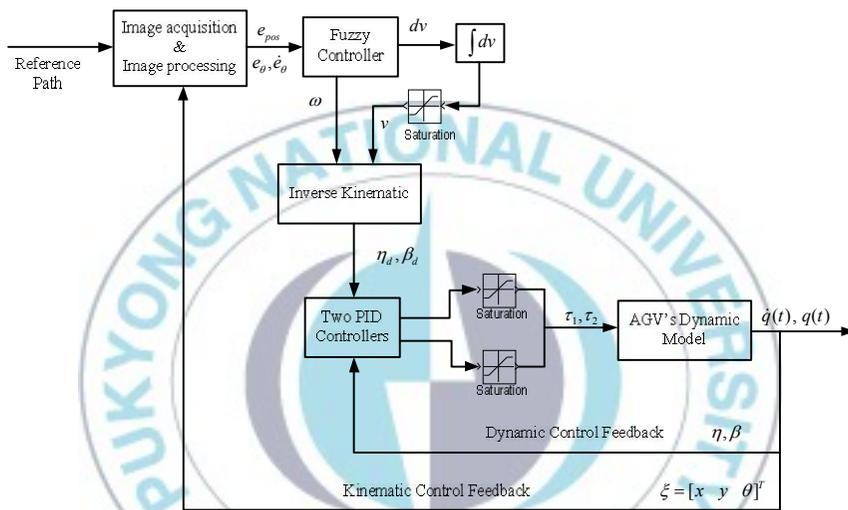


Fig. 4.8 Full structure of the proposed control algorithm for AGV

In practical, the torques produced by two DC motors are limited and the linear velocity of tracking point, Q , on AGV is bounded. Therefore, it is necessary to include saturation functions for linear velocity of tracking point Q and torques produced by two DC motors.

Chapter 5: Simulation and Experimental Results

To verify the effectiveness of the proposed controllers, simulations and experiments have been done for the AGV to follow an unknown path. In the simulation, the numerical parameter values, and the initial values are given in Table 5.1 and Table 5.2.

Table 5.1 Numerical parameter values of AGV

| Parameters | Values | Units | Parameters | Values | Units |
|------------|--------|---------------------|------------|--------|---------------------|
| M^* | 500 | [kg] | I_{p1} | 10 | [kgm ²] |
| m_1 | 10 | [kg] | I_{r1} | 1 | [kgm ²] |
| m_2 | 1 | [kg] | I_{r2} | 0.05 | [kgm ²] |
| m_3 | 1 | [kg] | I_{r3} | 0.05 | [kgm ²] |
| h | 0.6 | [m] | l_1 | 1.22 | [m] |
| r_s | 0.125 | [m] | l_2 | 0.2675 | [m] |
| r_f | 0.05 | [m] | l_3 | 0.2675 | [m] |
| I_0 | 250 | [kgm ²] | | | |

Table 5.2 Initial values for simulation

| Parameters | Values | Units | Parameters | Values | Units |
|------------|--------|-------|------------|------------------|----------|
| x_0 | 4 | [m] | θ_0 | -90 ⁰ | [degree] |
| y_0 | 1,9 | [m] | β_0 | 0 ⁰ | [degree] |

Table 5.3 shows the saturation values used in simulation. Because the powers of two DC motors are limited. The maximum power of the first DC motor is 3 Kw and the maximum power of the second motor is 1 Kw. For this reason, the torques cannot exceed the maximum values are given in Table 5.3.

Table 5.3 Saturation values in simulation

| Parameters | Min | Max | Units |
|------------|-------|------|-------|
| v | 0.5 | 2 | [m/s] |
| τ_1 | -3000 | 3000 | [Nm] |
| τ_2 | -1000 | 1000 | [Nm] |

Table 5.4 shows the parameters in two PID controllers obtained by trial and error.

Table 5.4 Parameters in two PID controllers

| Parameters | Values | Parameters | Values |
|------------|--------|------------|--------|
| k_{p1} | 300 | k_{p2} | 200 |
| k_{i1} | 0.5 | k_{i2} | 5 |
| k_{d1} | 25 | k_{d2} | 5 |

The reference path has five segments with three straight line segments and two curved line segments. The radius of the first curve is 6 m and the radius of the second curve is 4 m. Fig. 5.1 shows the reference path in global coordinate frame.

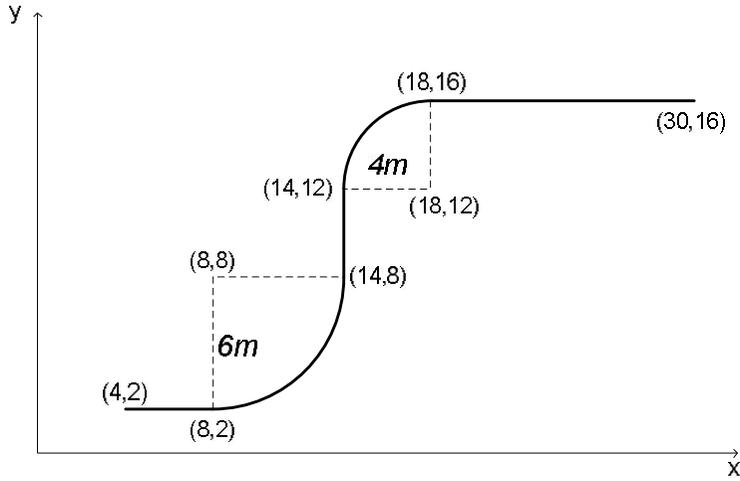


Fig. 5.1 Reference path

5.1 Simulation results

The simulation results of AGV's path following are presented in Figs. 5.2~5.9. Fig. 5.2 shows the path following result in simulation. Fig. 5.3 shows the position error. From initial position error of 0.1 m, the position error reduces to zero after 2.5 seconds. In straight path section, the position error is approximately zero. At the first bending with 6m curvature radius, the position error is around 0.01m. At the second bending with 4 m curvature radius, the position error is around 0.06 m. Fig. 5.4 shows angle error that is bounded by 6° . From Fig. 5.3 and Fig. 5.4, the angle error is big when the position error changes quickly. Fig. 5.5 shows the heading angle of AGV. In Fig. 5.5, the reference value changes in 5 periods corresponding with 5 segment of the reference path. Fig. 5.6 shows the steering angle of the steering wheel. When the AGV moves on straight path, the steering angle is around 90° . At the first bending with big curvature radius, the steering angle is around 80° . It has about 10° difference

from the steering angle when AGV moves on straight path. At the second bending with small curvature radius, the steering angle is around 108° . It has about 18° difference from the steering angle when AGV moves on straight path. Fig. 5.7 shows the linear velocity of AGV at tracking point Q . The linear velocity is bounded from 0.5 m/s to 2 m/s because of saturation function. At the beginning, the linear velocity is 0.5 m/s . After 8s , the linear velocity reaches to 2 m/s and it is decreased a little when AGV moves on the second bending. Fig. 5.8 shows the torque applied for the orientation of steering wheel, and Fig. 5.9 shows the torque applied for the rotation of steering wheel. Because of saturation functions, the first torque is bounded within $\pm 3000 \text{ Nm}$ and the second torque is bounded within $\pm 1000 \text{ Nm}$. These torques are relatively big. However, it is assumed that these torques are accepted with DC motors of 3kW and 1kW , respectively.

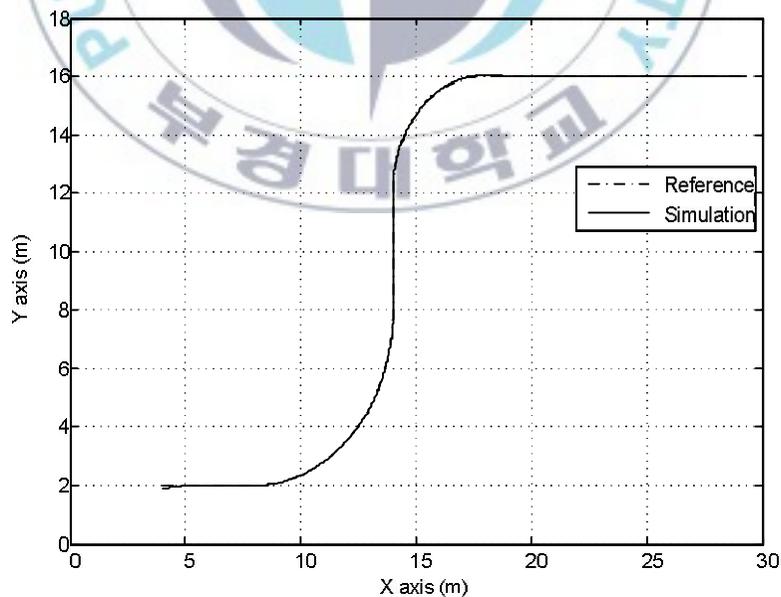


Fig. 5.2 Path following of AGV in simulation

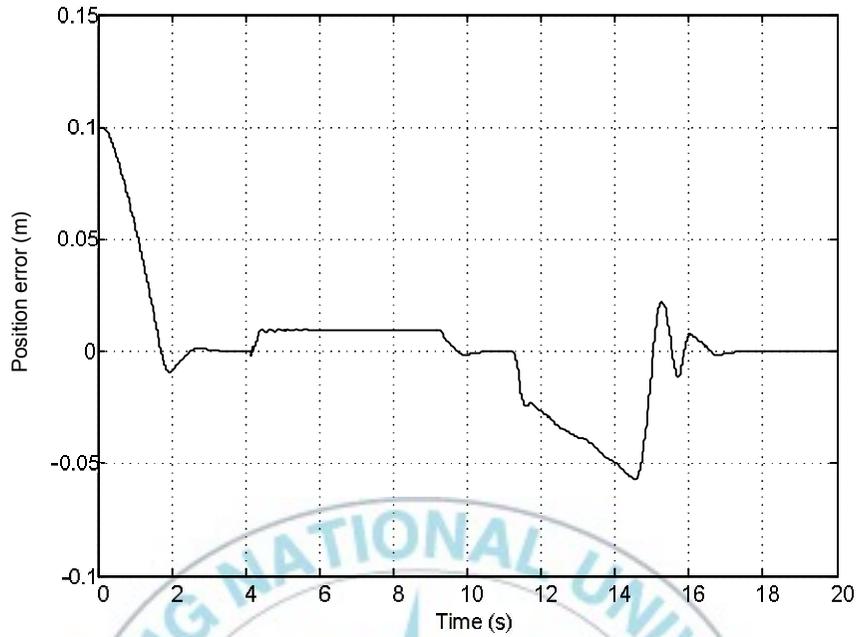


Fig. 5.3 Position error of AGV in simulation

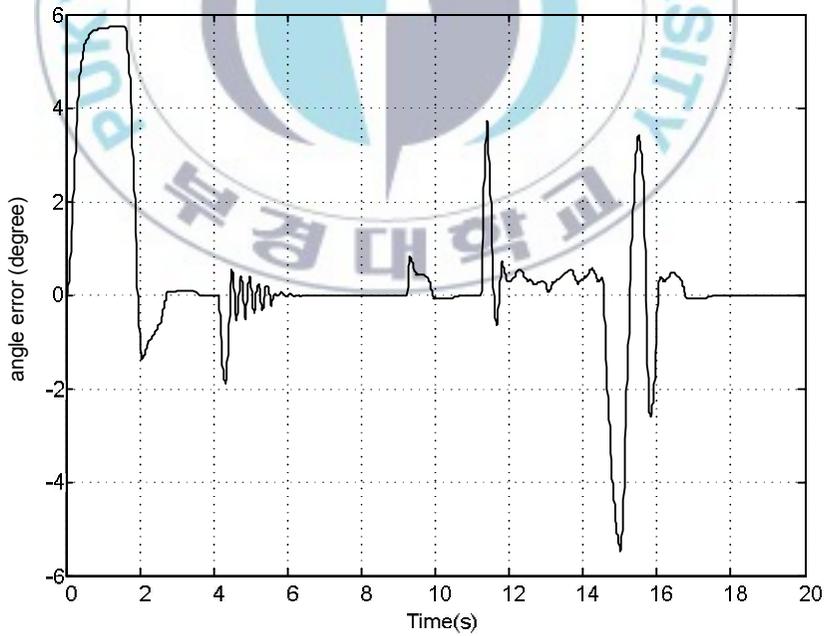


Fig. 5.4 Angle error of AGV in simulation

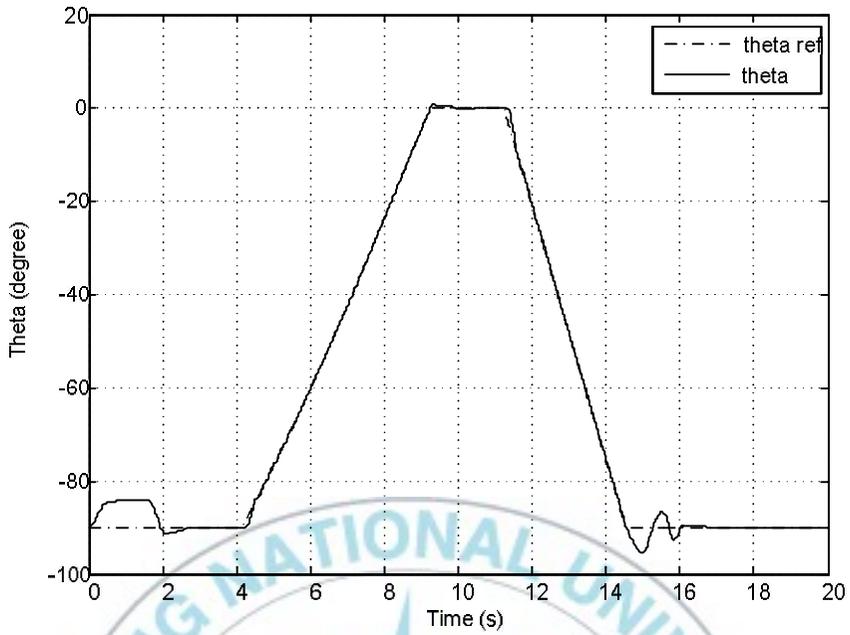


Fig. 5.5 Heading angle of AGV in simulation

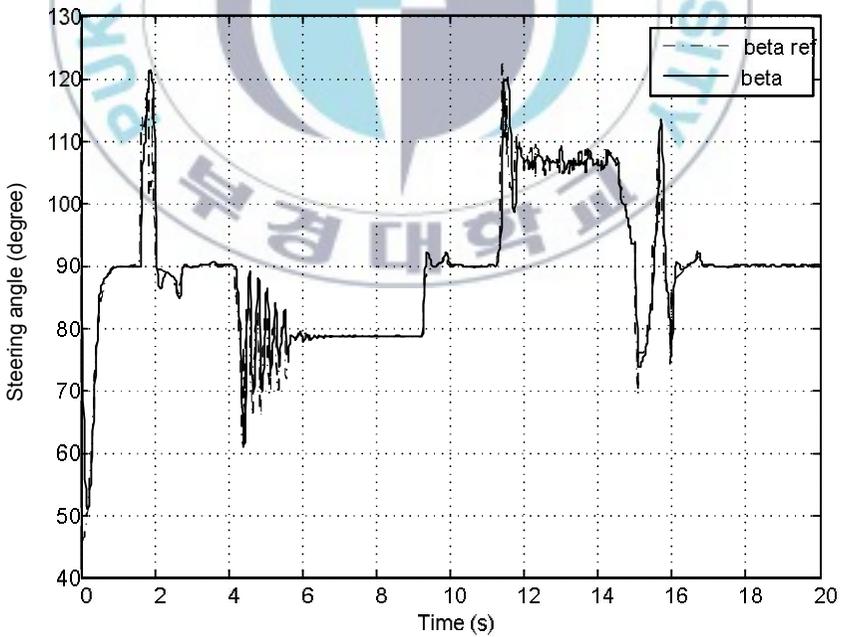


Fig. 5.6 Steering angle of steering wheel in simulation

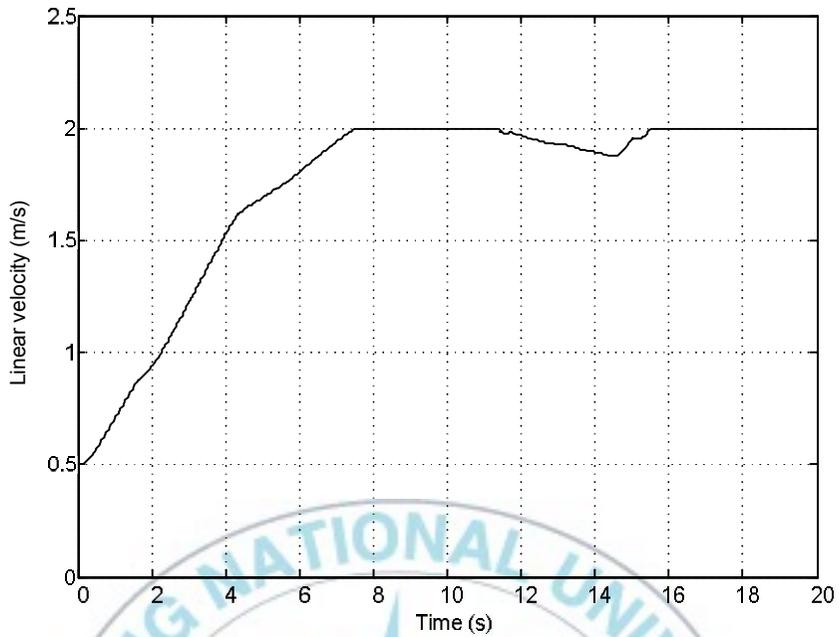


Fig. 5.7 Linear velocity of tracking point on AGV in simulation

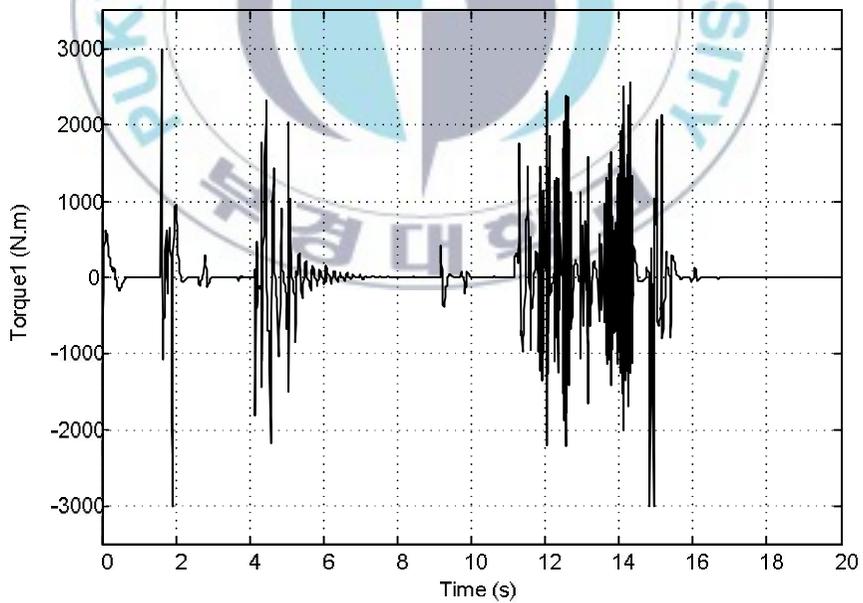


Fig. 5.8 Torque applied for the orientation of steering wheel in simulation

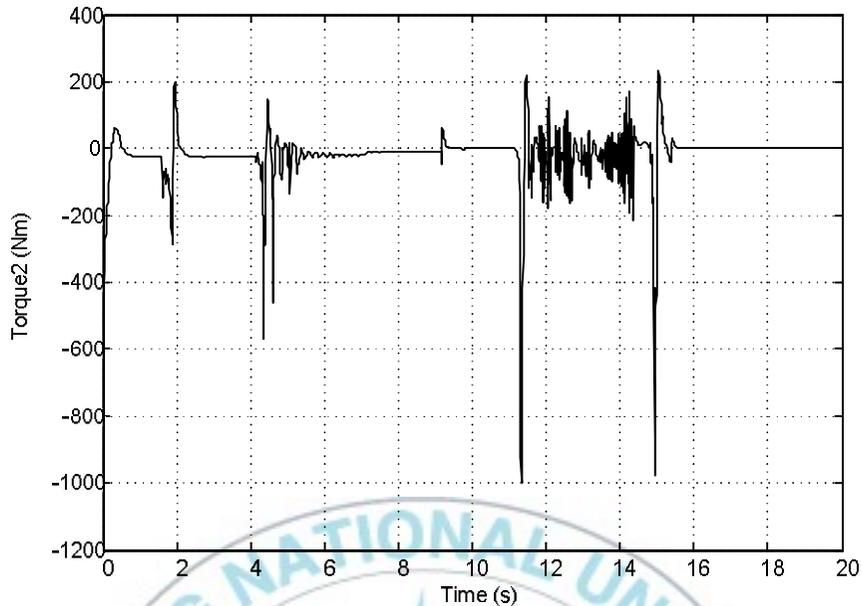


Fig. 5.9 Torque applied for the rotation of steering wheel in simulation

5.2 Experiment results

Experiments are carried out on a real AGV with load capacity of 2000 kg. In experiment, the AGV operates in free load mode. The experimental results for path following of the AGV are shown in Figs. 5.10~5.13. It shows that the experimental results of the tracking errors of the AGV, steering angle of steering wheel and linear velocity at tracking point of the AGV are bounded along the simulation results. Fig. 5.10 shows the position error. The experimental result for position error is bounded within $\pm 0.1m$. Fig. 5.11 shows the angle error. The experimental result for angle error is bounded within $\pm 8^\circ$.

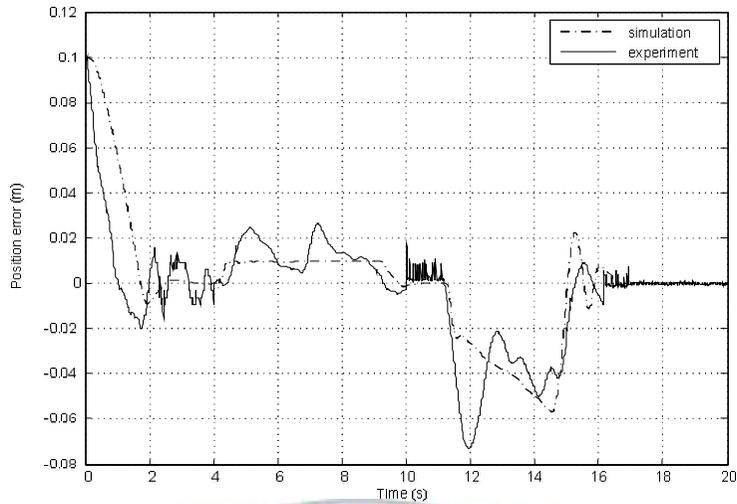


Fig. 5.10 Position error of AGV in experiment

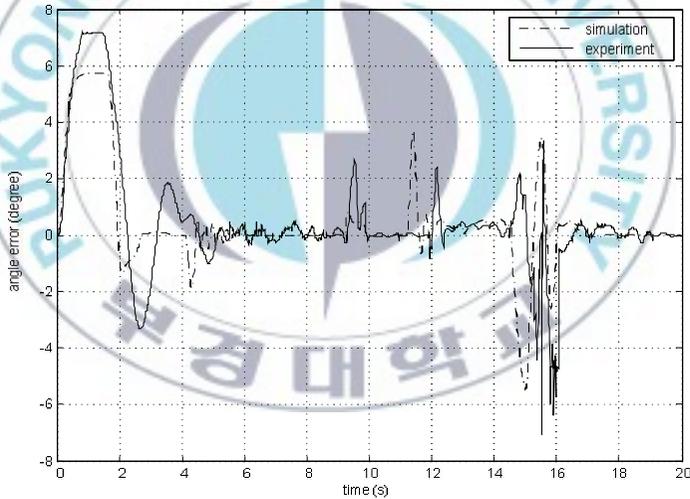


Fig. 5.11 Angle error of AGV in experiment.

Fig. 5.12 shows the steering angle of the steering wheel. The steering angle is bounded within $90^{\circ} \pm 40^{\circ}$ and bounded within $[50^{\circ}, 130^{\circ}]$. Fig. 5.13 shows the linear velocity at the tracking point

of the AGV. The linear velocity is limited by 2 m/s approximately after 6 seconds.

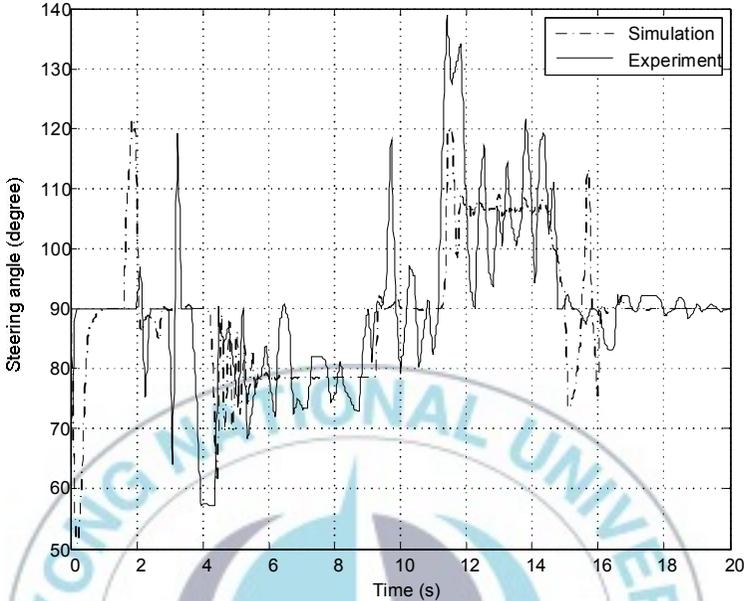


Fig. 5.12 Steering angle of steering wheel in experiment

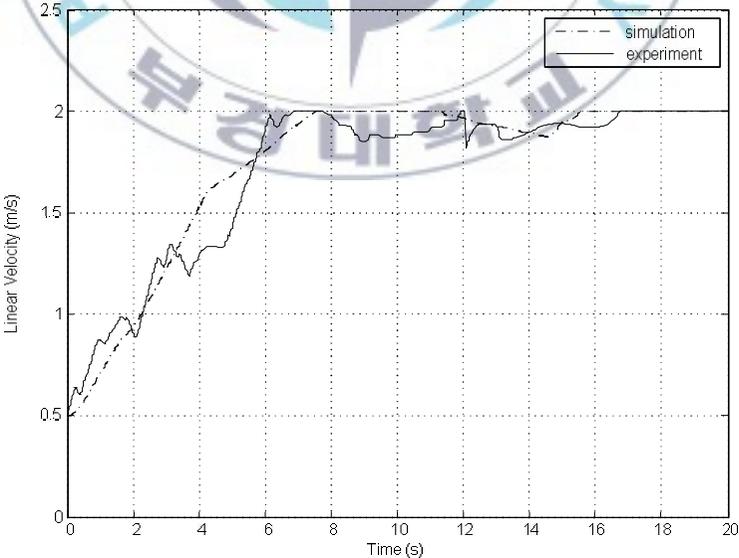


Fig. 5.13 Linear velocity of tracking point on AGV in experiment

The summary obtained from the simulation and experimental results are obtained as follows:

- The smaller the curvature radius is, the bigger the position error of the AGV and the angle error of the AGV and the steering angle of the steering wheel are. Moreover, the smaller the curvature radius is, the more quickly of the torques for the orientation and rotation of the steering wheel change.
- The proposed controller can be applied for AGV to follow any smooth curve path with large enough curvature radius. However, to obtain a good path following performance of AGV, the DC motor for orientation of steering wheel has to provide a big enough torque.
- Experimental results are bounded along the simulation results within some ranges: position error within $\pm 0.1m$, angle error within $\pm 8^\circ$ and steering angle within $90^\circ \pm 40^\circ$

Chapter 6: Conclusions and Future Works

6.1 Conclusions

This dissertation develops an industrial AGV, a tricycle wheeled mobile robot. The conclusions of this dissertation are summarized as follows:

- AGV's modeling including kinematic model and dynamic model is considered. Based on structure and geometric characteristics of each wheel, constraints under pure rolling and no slipping condition are reduced. From these constraints, kinematic modeling is obtained. Dynamic modeling is also reduced by applying the well known Lagrange equation under the above nonholonomic constraints in the motion of the AGV.
- The second one is hardware design. The AGV has three wheels including the steering wheel and two fixed wheels. The steering wheel is driven by two DC motors, the first one for its orientation and the second one for its rotation. AGV has a hoisting component for load transporting. The hoisting component is driven by the hydraulic actuator. Sensor for obstacle avoidance such as photoelectric sensors, switching sensor are used. A proximity sensor is used to detect the original point of steering wheel. Two rotary encoders are used to measure AGV's speed and steering angle of the steering wheel. Switching sensors are used to limit movement of steering wheel and hoisting component. Camera Logitech C600 is used for navigation.

- The control system is developed based on the integration of a notebook (Intel Core 2 Dual, 2.8 Ghz, 2 GB RAM) and PIC microprocessors. The notebook communicates with PIC microprocessors by RS232 protocol. The control system includes two parts, high level computer control and low level microprocessor control. The former is used for image processing and control algorithm. The latter is used for device control. Furthermore, a desktop personal computer that communicates with the AGV through wireless communication Bluetooth is used to monitor and control the AGV remotely.
- Two software programs are developed in C# programming language of Visual Studio 2008. The first one is used on the notebook for image processing based on AForge.NET Framework. The second one is used on desktop PC for monitoring and remote control of the AGV.
- Control algorithm for path-following of AGV is proposed based on integration of two control loops, kinematic control loop and dynamic control loop. The kinematic control loop based on fuzzy logic framework and the dynamic control loop based on two PID controllers are designed. The proposed controller shows that the AGV can follow the reference path with small curvature radius of 4m. Simulation and experimental results show that the tracking errors are bounded within $\pm 0.1 m$, $\pm 8^\circ$. To achieve a good performance of AGV for path following, the torques of DC driving motors must be big enough. Simulation in this dissertation requires the torque coming

to 3000 Nm for the first motor and 1000 Nm for the second motor applied to steering wheel.

- Experimental results are bounded along the simulation results within some ranges: position error within $\pm 0.1m$, angle error within $\pm 8^\circ$ and steering angle within $90^\circ \pm 40^\circ$

The effectiveness of the proposed system is shown through simulation and experimental results. The AGV can follow the desired path with a large enough curvature radius smoothly. So the system can be applicable and implemented in practical

6.2 Discussion and future works

In dynamic control loop, there exist a lot of kind of unknown disturbances and noise which decrease the control performances such as the stability, frequency response and loading sensitivity. Therefore, the control method based on conventional PID is limited. To improve the control performance, other control methods should be considered.

To move throughout the manufacturing facility or a warehouse, AGV must be able to move both of directions, forward and backward. However, this dissertation only considers the forward moving of the AGV. In backward moving, when the reference path is sharpened bent or narrow, AGV is impossible to follow its path. To overcome that problem, a solution must be proposed. In this solution, the camera sensor is fixed on steering wheel frame.

References

- [1] B. d' Andréa-Novel, G. Bastin and G. Campion, "*Modeling and Control of Nonholonomic Wheeled Mobile Robots*," Proceeding of the 1991 IEEE International Conference on Robotics and Automation Sacramento, California, Apr. 1991.
- [2] Guy Campion, Georges Bastin and Brigitte D' Andréa -Novel, "*Structural Properties and Classification of Kinematic and Dynamic Models of Wheeled Mobile Robots*," IEEE Transactions on Robotics and Automation, Vol. 12, No. 1, Feb. 1996.
- [3] Yilin Zhao and Spencer L. BeMent, "*Kinematics, Dynamics and Control of Wheeled Mobile Robots*," Proceeding of the 1992 IEEE International Conference on Robotics and Automation Nice, France, May. 1992.
- [4] L. Gracia and J. Tornero, "*Kinematic Control of Wheeled Mobile Robots*," Latin American Applied Research, 2008.
- [5] A. Bloch, M. Reyhanoglu and N. McClamroch, "*Control and Stabilization of Nonholonomic Dynamic Systems*," IEEE Trans. on Automatic Control, Vol. 37, No. 11, pp. 1746-1756, Nov. 1992.
- [6] L. Gracia and J. Tornero, "*Kinematic Control of Wheeled Mobile Robots*," Latin American Applied Research, 2008.

- [7] André Kanga and Ahmed Rachid, “*Speed, Steering Angle and Path Tracking Controls for a Tricycle Robot,*” in Proc. IEEE Int. Sym. Computer-Aided Control System Design, Dearborn, MI, Sept. 1996.
- [8] R. Fierro and F. L. Lewis, “*Control of a Nonholonomic Mobile Robot: Backstepping Kinematics into Dynamics,*” Elsevier Journal of Robotic Systems, Vol. 14, No. 3, pp. 149-163, 1997.
- [9] Mohammad Eghtesad, Dan S. Neculescu, “*Experimental Study of the Dynamic based Feedback Linearization of an Autonomous Wheeled Ground Vehicle,*” Elsevier Journal of Robotics and Autonomous Systems, Vol. 47, pp. 47-63, 2004.
- [10] Nilanjan Chakraborty, Ashitava Ghosal, “*Dynamic Modeling and Simulation of a Wheeled Mobile Robot for Traversing Uneven Terrain Without Slip,*” Journal of Mechanical Design, Vol. 127, No. 5, pp. 901-910, 2004.
- [11] Yang Bae Jeon, Sang Bong Kim, “*Modeling and Motion Control of Mobile Robot for Lattice Type Welding,*” KSME Int. Journal, Vol. 16, No. 1, pp. 83-93, 2002.
- [12] E. Aguirre and A. González, “*Fuzzy Behaviors for Mobile Robot Navigation: Design, Coordination and Fusion,*” Int. J. Reas., Vol. 25, pp. 255-289, 2000.
- [13] D. Driankov and A. Saffiotti, “*Fuzzy Logic Techniques for Autonomous Vehicle Navigation,*” New York: Physica - Verlag, 2001.

- [14] H. Maaref and C. Barret, “*Sensor – Based Fuzzy Navigation of an Autonomous Mobile Robot in an Indoor Environment,*” *Contr. Eng. Practice*, Vol. 8, pp. 757-768, 2000.
- [15] M. Maeda, M. Shimakawa and S. Murakami, “*Predictive Fuzzy Control of an Autonomous Mobile Robot with Forecast Learning Function,*” *Contr. Eng. Practice*, Vol. 8, pp. 757-768, 2000.
- [16] X. Yang, M. Moallen and R. Patel, “*An Improved Fuzzy Logic Based Navigation System for Mobile Robots,*” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, NV, pp. 1709-1714, 2003.
- [17] I. Baturone, F. Moreno- Velo, S. Sánchez-Solano and A. Ollero, “*Automatic Design of Fuzzy Controllers for Car-Like Autonomous Robots,*” *IEEE Trans. Fuzzy Syst.*, Vol. 12, pp. 447-465, 2004.
- [18] T. H. Li, S. J. Chang, “*Autonomous Fuzzy Parking Control of a Car-Like Mobile Robots,*” *IEEE trans. Syst., Man, Cybern. A, Syst., Humans*, Vol. 33, No. 4, pp. 451-465, 2003.
- [19] T. H. Li, S. J. Chang and Y. X. Chen, “*Implementation of Human Like Driving Skills by Autonomous Fuzzy Behavior Control on an FPGA Based Car-Like Mobile Robot,*” *EEE trans. Ind. Electron.*, Vol. 50, pp. 867-880, 2003.
- [20] T. Lee, H. Lam, F. Leung and P. Tam, “*A practical Fuzzy Logic Controller for the Path Tracking of Wheeled Mobile Robots,*”

in Proc. IEEE Ann. Conf. Ind. Electron. Soc., Denver, CO, pp. 574-579, 2001.

- [21] O. Sánchez, A. Ollero and G. Heredia, “*Adaptive Fuzzy Control for Automatic Path Tracking of Outdoor Mobile Robots. Application to Romeo 3r*” in Proc. 6th Int. Conf. Fuzzy Syst., Barcelona, Spain, pp. 593-599, 1997.
- [22] J. Baltes and R. Otte, “*A Fuzzy Logic Controller for Car-Like Mobile Robots.*” In Proc. IEEE Int. Symp. Computat. Intell. Robot. Automat., Monterey, CA, pp. 89-94, 1999.
- [23] G. Antonelli and S. Chiaverini, “*Experiments of Fuzzy Lane Following for Mobile Robots,*” in Proc. 2004 Amer. Contr. Conf., Boston, MA, pp. 1079-1084, Jun. 2004.
- [24] G. Antonelli, S. Chiaverini and G. Fusco, “*Real-time Path Tracking for Unicycle-Like Mobile Robots Under Velocity and Acceleration Constraints,*” in Proc. 2001 Amer. Contr. Conf., Arlington, VA, pp. 119-124, Jun. 2001.
- [25] S. Bentalba, A. E. Hajjaji and A. Rachid, “*Fuzzy Control of a Mobile Robot: A New Approach,*” in Proc. 1997 IEEE Int. Conf. Contr. Applicat., Hartford, CT, pp.69-72, 1997.
- [26] B. Lakehal, Y. Amirat and J. Pontnau, “*Fuzzy Steering Control of a Mobile Robot,*” in Proc. IEEE Int. Conf. Indust. Autom. Contr., Emerging Technol., Taipei, Taiwan, pp. 383-386, 1995.
- [27] G. Schuster, “*Simulation of Fuzzy Motion Controlled Four-Wheel Steered Mobile Robot,*” in Proc. IEEE Int. Conf. Intell. Eng. Syst., Budapest, Hungary, pp. 89-94, 1997.

- [28] A. Das, R. Fiero, V. Kumar, B. Southall and C. Taylor, “*Real-Time Vision-Based Control of a Nonholonomic Mobile Robot*,” in Proc. 2001 IEEE Int. Conf. Robot. Autom., Seoul, Korea, pp. 1714-1719, May 2001.
- [29] B. Southall and C. Taylor, “*Stochastic Road Shape Estimation*,” in Proc. 8th IEEE Int. Conf. Comput. Vision, Vancouver, CA, pp. 205-212, 2001.
- [30] A. De Luca, G. Oriolo and M. Vendittelli, “*Stabilization of The Unicycle Via Dynamic Feedback Linearization*,” in Proc. 6th IFAC Symp. Robot Contr., Wien, pp. 397-402, 2000.
- [31] Z. Zhang, “*Flexible Camera Calibration by Viewing a Plane From Unknown Orientations*,” in Proc. 7th IEEE Int. Conf. Comput. Vision, Kerkyra, GR, pp. 20-27, Sep. 1999.
- [32] R. –E. Precup, S. Preitl and Z. Preitl, “*Fuzzy Control Solution for a Class of Tricycle Mobile Robots*,” in Proc. 2006 IEEE Int. Conf. Mechatronics, Budapest, pp. 203-208, 2006.
- [33] G. Antonelli, S. Chiaverini and G. Fusco, “*A Fuzzy-Logic-Based Approach for Mobile Robot Path Tracking*,” IEEE Trans. Fuzzy Systems, Vol. 15, No. 2, pp. 211-221, April 2007.
- [34] Chia-Feng Juang and Chia-Hung Hsu, “*Reinforcement Ant Optimized Fuzzy Controller for Mobile-Robot Wall-Following Control*,” IEEE Trans. Ind. Electron. Vol. 56, No. 10, pp. 3931-3940, Oct. 2009.

- [35] K. R. S. Kodagoda, W. S. Wijesoma and E. K. Teoh, “*Fuzzy Speed and Steering Control of an AGV*,” IEEE Trans. Contr. Sys. Tech., Vol. 10, No. 1, pp. 112-120, January 2002.
- [36] J. Baltes and R. Otte, “*A Fuzzy Logic Controller for Car-like Mobile Robots*,” in Proc. IEEE Int. Conf. Comp. Intell. Rob. Auto., 1999.
- [37] <http://www.aforgenet.com>



Publications and Conferences

A. Publications

- [1] Phuc Thinh Doan, Hoang Duy Vo, Hak Kyeong Kim and Sang Bong Kim, “*A New Approach for Development of Quadruped Robot Based on Biological Concepts,*” IJPEM International Journal of Precision Engineering and Manufacturing, Vol. 11, No. 4, pp. 559-568, Aug. 2010.
- [2] Doan Phuc Thinh, Nguyen Van Duy Phuc, Nguyen Tan Tien, “*Study on Control of Two-wheeled Welding Mobile Robot,*” Vietnamese Journal of Information Technology, Control and Automation, T.24, S.4, 2008.

B. Conferences

- [1] Phuc Thinh Doan, Tuan Dinh Viet, Hak Kyeong Kim and Sang Bong Kim, “*Control of Automated Guide Vehicle for Path Following Using Camera Sensor,*” Proceeding of The 2010 International Technical Conference on Advanced Engineering, pp. 13-19, PKNU, Busan, Korea, Nov. 2010.
- [2] Phuc Thinh Doan, Sang-Kwun Jeong, Tuan Dinh Viet, Hak Kyeong Kim and Sang Bong Kim, “*A Control Method for Path Following of Automated Guide Vehicle,*” Proceeding of IEEE The 7th International Conference on Ubiquitous Robots and Ambient Intelligence, pp.409-412, Busan, Korea, Nov. 2010.

- [3] Sang Bong Kim, Phuc Thinh Doan and Hak Kyeong Kim, “*Fuzzy Control of Omnidirectional Mobile Platform for Tracking the Curved Path,*” Proceeding of the Conference, The Korean Society of Marine Engineering, Mokpo Marine University, Korea, pp. 281-282, Apr. 2010
- [4] Phuc Thinh Doan, Hak Kyeong Kim and Sang Bong Kim, “*A Study on Control for Rhythm Motion of Quadruped robot’s Operating in Walk Gait and Trot Gait,*” Proceeding of the 11th Conference on Science and Technology International Symposium on Mechanical Engineering, Vietnam, pp. 322-327, Oct. 2009.
- [5] Dang Khoa Pham, Phuc Thinh Doan, Xuan Quoc Vo, Kim Hwan Seong and Tan Tien Nguyen, “*Study on Information Management of Newspaper Vending Machine Network Based on Global System for Mobile Communications Technology (GSM),*” Proceeding of the 11th Conference on Science and Technology International Symposium on Mechanical Engineering, Vietnam, pp. 94-98, Oct. 2009.
- [6] Van Duy Phuc, Phuc Thinh Doan, Hoai Phuong Nguyen and Tan Tien Nguyen, “*Two-Wheeled Mobile Robot Control using Fuzzy and PID Control Simultaneously,*” Proceeding of The Second KMU-HCMUT Joint Workshop, Korea, pp. 156-159, Nov. 2008.
- [7] Doan Phuc Thinh, Nguyen Van Duy Phuc and Nguyen Tan Tien, “*Study on Control of Two-wheeled Welding Mobile Robot,*” Proceeding of the 4th Vietnam Conference on Mechatronics - VCM2008, Da Nang, Vietnam, Oct. 2008.

Appendix A

In this section, $[\mathbf{T}]_{\xi}$, $[\mathbf{T}]_{\beta}$ and $[\mathbf{T}]_{\phi}$ are considered

The kinetic energy in (2.31) can be rewritten as follows:

$$\mathbf{T} = \frac{1}{2}(\dot{\xi}^T \mathbf{R}^T \mathbf{M} \mathbf{R} \dot{\xi} + 2\mathbf{V}^T \mathbf{R} \dot{\xi} \dot{\beta} + I_{\beta} \dot{\beta}^2 + \dot{\Phi}^T \mathbf{I}_{\phi} \dot{\Phi}) \quad (\text{A.1})$$

First, $\frac{\partial \mathbf{T}}{\partial \dot{\xi}}$ and its derivative, $\frac{d}{dt} \left(\frac{\partial \mathbf{T}}{\partial \dot{\xi}} \right)$, are obtained as follows:

$$\frac{\partial \mathbf{T}}{\partial \dot{\xi}} = \mathbf{R}^T(\theta) \mathbf{M} \mathbf{R}(\theta) \dot{\xi} + \mathbf{R}^T(\theta) \mathbf{V} \dot{\beta} \quad (\text{A.2})$$

$$\frac{d}{dt} \left(\frac{\partial \mathbf{T}}{\partial \dot{\xi}} \right) = \frac{d}{dt} (\mathbf{R}^T(\theta) \mathbf{M} \mathbf{R}(\theta) \dot{\xi}) + \frac{d}{dt} (\mathbf{R}^T(\theta) \mathbf{V} \dot{\beta}) \quad (\text{A.3})$$

$$\begin{aligned} \frac{d}{dt} \left(\frac{\partial \mathbf{T}}{\partial \dot{\xi}} \right) &= \frac{d}{dt} (\mathbf{R}^T(\theta) \mathbf{M} \mathbf{R}(\theta)) \dot{\xi} + \mathbf{R}^T(\theta) \mathbf{M} \mathbf{R}(\theta) \ddot{\xi} \\ &\quad + \frac{d}{dt} (\mathbf{R}^T(\theta) \mathbf{V}) \dot{\beta} + \mathbf{R}^T(\theta) \mathbf{V} \ddot{\beta} \end{aligned} \quad (\text{A.4})$$

where

$$\frac{d}{dt} (\mathbf{R}(\theta)) = \frac{\partial \mathbf{R}}{\partial \theta} \dot{\theta} = \begin{bmatrix} -\sin \theta & \cos \theta & 0 \\ -\cos \theta & -\sin \theta & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\theta} = \boldsymbol{\Psi} \mathbf{R}(\theta) \dot{\theta} \quad (\text{A.5})$$

with
$$\boldsymbol{\Psi} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\begin{aligned} \frac{d}{dt} (\mathbf{R}^T(\theta) \mathbf{M} \mathbf{R}(\theta)) &= \frac{d}{dt} (\mathbf{R}^T(\theta)) \mathbf{M} \mathbf{R}(\theta) + \mathbf{R}^T(\theta) \frac{d}{dt} (\mathbf{M} \mathbf{R}(\theta)) \\ &= \frac{d}{dt} (\mathbf{R}^T(\theta)) \mathbf{M} \mathbf{R}(\theta) + \mathbf{R}^T(\theta) \mathbf{M} \frac{d}{dt} (\mathbf{R}(\theta)) \\ &= \mathbf{R}^T(\theta) (\boldsymbol{\Psi}^T \mathbf{M} + \mathbf{M} \boldsymbol{\Psi}) \mathbf{R}(\theta) \dot{\theta} \end{aligned} \quad (\text{A.6})$$

$$\frac{d}{dt}(\mathbf{R}^T(\theta)\mathbf{V}) = \frac{d}{dt}(\mathbf{R}^T(\theta))\mathbf{V} = \mathbf{R}^T(\theta)\boldsymbol{\Psi}^T\mathbf{V}\dot{\theta} \quad (\text{A.7})$$

Therefore, from Eqs. (A.4), (A.6) and (A.7), the following is obtained.

$$\begin{aligned} \frac{d}{dt}\left(\frac{\partial\mathbf{T}}{\partial\dot{\xi}}\right) &= \mathbf{R}^T(\theta)(\boldsymbol{\Psi}^T\mathbf{M} + \mathbf{M}\boldsymbol{\Psi})\mathbf{R}(\theta)\dot{\xi}\dot{\theta} + \mathbf{R}^T(\theta)\mathbf{M}\mathbf{R}(\theta)\ddot{\xi} \\ &\quad + \mathbf{R}^T(\theta)\boldsymbol{\Psi}^T\mathbf{V}\dot{\theta}\dot{\beta} + \mathbf{R}^T(\theta)\mathbf{V}\ddot{\beta} \end{aligned} \quad (\text{A.8})$$

Second, $\frac{\partial\mathbf{T}}{\partial\xi}$ is calculated as follows:

$$\begin{cases} \frac{\partial\mathbf{T}}{\partial x} = 0 \\ \frac{\partial\mathbf{T}}{\partial y} = 0 \\ \frac{\partial\mathbf{T}}{\partial\theta} = (\dot{\xi}^T\mathbf{R}^T(\theta)\mathbf{M} + \dot{\beta}\mathbf{V}^T)\frac{\partial}{\partial\theta}(\mathbf{R}(\theta))\dot{\xi} \end{cases} \quad (\text{A.9})$$

where

$$\begin{aligned} \frac{\partial\mathbf{T}}{\partial\theta} &= (\dot{\xi}^T\mathbf{R}^T(\theta)\mathbf{M} + \dot{\beta}\mathbf{V}^T)\frac{\partial}{\partial\theta}(\mathbf{R}(\theta))\dot{\xi} \\ &= (\dot{\xi}^T\mathbf{R}^T(\theta)\mathbf{M} + \dot{\beta}\mathbf{V}^T)\boldsymbol{\Psi}\mathbf{R}(\theta)\dot{\xi} \end{aligned} \quad (\text{A.10})$$

Therefore, from Eqs. (A.9) and (A.10), the following is obtained.

$$\frac{\partial\mathbf{T}}{\partial\xi} = \begin{bmatrix} \frac{\partial\mathbf{T}}{\partial x} \\ \frac{\partial\mathbf{T}}{\partial y} \\ \frac{\partial\mathbf{T}}{\partial\theta} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{\partial\mathbf{T}}{\partial\theta} \end{bmatrix} = \mathbf{D}\frac{\partial\mathbf{T}}{\partial\theta} = \mathbf{D}(\dot{\xi}^T\mathbf{R}^T(\theta)\mathbf{M} + \dot{\beta}\mathbf{V}^T)\boldsymbol{\Psi}\mathbf{R}(\theta)\dot{\xi} \quad (\text{A.11})$$

with $\mathbf{D} = [0 \ 0 \ 1]^T$

From (A.8), (A.10) and (A.11) the following is obtained.

$$\begin{aligned}
[\mathbf{T}]_{\xi} &= \mathbf{R}^T(\theta)(\boldsymbol{\Psi}^T \mathbf{M} + \mathbf{M} \boldsymbol{\Psi}) \mathbf{R}(\theta) \dot{\xi} \dot{\theta} + \mathbf{R}^T(\theta) \mathbf{M} \mathbf{R}(\theta) \ddot{\xi} \\
&\quad + \mathbf{R}^T(\theta) \boldsymbol{\Psi}^T \mathbf{V} \dot{\beta} \dot{\theta} + \mathbf{R}^T(\theta) \mathbf{V} \ddot{\beta} \\
&\quad - \mathbf{D}(\dot{\xi}^T \mathbf{R}^T(\theta) \mathbf{M} + \dot{\beta} \mathbf{V}^T) \boldsymbol{\Psi} \mathbf{R}(\theta) \dot{\xi}
\end{aligned} \tag{A.12}$$

$[\mathbf{T}]_{\beta}$ and $[\mathbf{T}]_{\Phi}$ can be obtained easily from Eq. (A.1) as follows:

$$\begin{cases} \frac{\partial \mathbf{T}}{\partial \dot{\beta}} = \mathbf{V}^T \mathbf{R}(\theta) \dot{\xi} + I_{\beta} \dot{\beta} \\ \frac{\partial \mathbf{T}}{\partial \beta} = 0 \end{cases} \quad \text{then } [\mathbf{T}]_{\beta} = \mathbf{V}^T \mathbf{R}(\theta) \ddot{\xi} + I_{\beta} \ddot{\beta} \tag{A.13}$$

$$[\mathbf{T}]_{\Phi} = \mathbf{I}_{\Phi} \ddot{\Phi} \tag{A.14}$$



Appendix B

In this section, the velocities $\dot{\xi}$, $\dot{\beta}$, $\dot{\Phi}$ and the accelerations $\ddot{\xi}$, $\ddot{\beta}$, $\ddot{\Phi}$ in equations (2.41) and (2.42) must be eliminated.

Eq. (2.46) is written as follows:

$$\begin{cases} \dot{\xi} = \mathbf{R}^T(\theta) \boldsymbol{\Sigma}(\beta) \eta \\ \dot{\beta} = \zeta \\ \dot{\Phi} = \mathbf{E}(\beta) \boldsymbol{\Sigma}(\beta) \eta \end{cases} \quad (\text{B.1})$$

Furthermore, from Eqs. (2.1) and (2.20), the followings are obtained.

$$\theta = \mathbf{D}^T \xi \quad \text{with} \quad \mathbf{D} = [0 \ 0 \ 1]^T \quad (\text{B.2})$$

$$\dot{\theta} = \mathbf{D}^T \dot{\xi} = \mathbf{D}^T \mathbf{R}^T(\theta) \boldsymbol{\Sigma}(\beta) \eta \quad (\text{B.3})$$

The second order time derivative of ξ in Eq. (B.1) is given as follows:

$$\ddot{\xi} = \mathbf{R}^T(\theta) \boldsymbol{\Sigma}(\beta) \dot{\eta} + \frac{d}{dt}(\mathbf{R}^T(\theta)) \boldsymbol{\Sigma}(\beta) \eta + \mathbf{R}^T(\theta) \frac{d}{dt}(\boldsymbol{\Sigma}(\beta)) \eta \quad (\text{B.4})$$

Using Eq. (A.5), the following is reduced as:

$$\frac{d}{dt}(\mathbf{R}^T(\theta)) = \mathbf{R}^T(\theta) \boldsymbol{\Psi}^T \dot{\theta} \quad \text{with} \quad \boldsymbol{\Psi} = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{B.5})$$

From Eq. (2.18), the following is reduced as:

$$\frac{d}{dt}(\boldsymbol{\Sigma}(\beta)) = \frac{\partial}{\partial \beta}(\boldsymbol{\Sigma}(\beta)) \dot{\beta} = \begin{bmatrix} 0 \\ -\mathbf{R} \cos \beta \\ \frac{\mathbf{R}}{l} \sin \beta \end{bmatrix} \dot{\beta} = \mathbf{L} \boldsymbol{\Sigma}(\beta) \dot{\beta} \quad (\text{B.6})$$

$$\text{with } \mathbf{L} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & l \\ 0 & -\frac{1}{l} & 0 \end{bmatrix}$$

Using Eqs. (B.4), (B.5) and (B.6), Eq. (B.7) is obtained.

$$\begin{aligned} \ddot{\xi} &= \mathbf{R}^T(\theta)\boldsymbol{\Sigma}(\beta)\dot{\eta} + \mathbf{R}^T(\theta)\boldsymbol{\Psi}^T\boldsymbol{\Sigma}(\beta)\dot{\theta}\eta + \mathbf{R}^T(\theta)\mathbf{L}\boldsymbol{\Sigma}(\beta)\dot{\beta}\eta \\ &= \mathbf{R}^T(\theta)\boldsymbol{\Sigma}(\beta)\dot{\eta} + \mathbf{k}_1(\theta, \beta, \eta, \zeta) \end{aligned} \quad (\text{B.7})$$

where

$$\mathbf{k}_1(\theta, \beta, \eta, \zeta) = \mathbf{R}^T(\theta)\boldsymbol{\Psi}^T\boldsymbol{\Sigma}(\beta)\mathbf{D}^T\mathbf{R}^T(\theta)\boldsymbol{\Sigma}(\beta)\eta^2 + \mathbf{R}^T(\theta)\mathbf{L}\boldsymbol{\Sigma}(\beta)\zeta\eta \quad (\text{B.8})$$

The second order time derivative of β of Eq. (B.1) is given as follows:

$$\ddot{\beta} = \dot{\zeta} \quad (\text{B.9})$$

From Eq. (B.1), the second order time derivative of Φ is given as follows:

$$\ddot{\Phi} = \mathbf{E}(\beta)\boldsymbol{\Sigma}(\beta)\dot{\eta} + \frac{d}{dt}(\mathbf{E}(\beta))\boldsymbol{\Sigma}(\beta)\eta + \mathbf{E}(\beta)\frac{d}{dt}(\boldsymbol{\Sigma}(\beta))\eta \quad (\text{B.10})$$

where

$$\mathbf{E}(\beta) = -\mathbf{J}_2^{-1}\mathbf{J}_1(\beta) \quad (\text{B.11})$$

$$\frac{d}{dt}(\mathbf{E}(\beta)) = -\mathbf{J}_2^{-1}\frac{d}{dt}(\mathbf{J}_1(\beta)) = -\mathbf{J}_2^{-1}\mathbf{N}(\beta)\dot{\beta} \quad (\text{B.12})$$

$$\text{with } \mathbf{N}(\beta) = \frac{\partial \mathbf{J}_1(\beta)}{\partial \beta} = \begin{bmatrix} -\sin \beta & \cos \beta & -l \sin \beta \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (\text{B.13})$$

Using Eqs. (B.10), (B.11) and (B.12), Eq. (B.14) is obtained.

$$\begin{aligned} \ddot{\Phi} &= \mathbf{E}(\beta)\boldsymbol{\Sigma}(\beta)\dot{\eta} - (\mathbf{J}_2^{-1}\mathbf{N}(\beta) - \mathbf{E}(\beta)\mathbf{L})\boldsymbol{\Sigma}(\beta)\zeta\eta \\ &= \mathbf{E}(\beta)\boldsymbol{\Sigma}(\beta)\dot{\eta} + \mathbf{k}_2(\beta, \eta, \zeta) \end{aligned} \quad (\text{B.14})$$

where

$$\mathbf{k}_2(\beta, \eta, \zeta) = (\mathbf{J}_2^{-1} \mathbf{N}(\beta) - \mathbf{E}(\beta) \mathbf{L}) \boldsymbol{\Sigma}(\beta) \zeta \eta \quad (\text{B.15})$$

From Eqs. (B.7), (B.9) and (B.14), the followings are obtained.

$$\begin{cases} \ddot{\boldsymbol{\xi}} = \mathbf{R}^T(\theta) \boldsymbol{\Sigma}(\beta) \dot{\eta} + \mathbf{k}_1(\theta, \beta, \eta, \zeta) \\ \ddot{\boldsymbol{\beta}} = \dot{\zeta} \\ \ddot{\boldsymbol{\Phi}} = \mathbf{E}(\beta) \boldsymbol{\Sigma}(\beta) \dot{\eta} + \mathbf{k}_2(\beta, \eta, \zeta) \end{cases} \quad (\text{B.16})$$

Using Eqs. (B.1), (B.16), (A.12), (A.14) and $\mathbf{R}(\theta) \mathbf{R}^T(\theta) = \mathbf{I}_{3 \times 3}$, Eq. (2.41) can be reduced as follows:

$$\mathbf{H}(\beta) \dot{\eta} + \boldsymbol{\Sigma}^T(\beta) \mathbf{V} \dot{\zeta} + f_2(\theta, \beta, \eta, \zeta) = \boldsymbol{\Sigma}^T(\beta) \mathbf{E}^T(\beta) \mathbf{P} \tau_2 \quad (\text{B.17})$$

where

$$\mathbf{H}(\beta) = \boldsymbol{\Sigma}^T(\beta) (\mathbf{M} + \mathbf{E}^T(\beta) \mathbf{I}_\phi \mathbf{E}(\beta)) \boldsymbol{\Sigma}(\beta) \quad (\text{B.18})$$

$$\begin{aligned} f_2(\theta, \beta, \eta, \zeta) = & \boldsymbol{\Sigma}^T(\beta) (\boldsymbol{\Psi}^T \mathbf{M} + \mathbf{M} \boldsymbol{\Psi}) \boldsymbol{\Sigma}(\beta) \mathbf{D}^T \mathbf{R}^T(\theta) \boldsymbol{\Sigma}(\beta) \eta^2 \\ & + \boldsymbol{\Sigma}^T(\beta) \mathbf{M} \mathbf{R}(\theta) \mathbf{k}_1(\theta, \beta, \eta, \zeta) \\ & + \boldsymbol{\Sigma}^T(\beta) \boldsymbol{\Psi}^T \mathbf{V} \mathbf{D}^T \mathbf{R}^T(\theta) \boldsymbol{\Sigma}(\beta) \eta \zeta \\ & + \boldsymbol{\Sigma}^T(\beta) \mathbf{R}(\theta) \mathbf{D} (\boldsymbol{\Sigma}^T(\beta) \mathbf{M} \eta + \mathbf{V}^T \zeta) \boldsymbol{\Psi} \boldsymbol{\Sigma}(\beta) \eta \\ & + \boldsymbol{\Sigma}^T(\beta) \mathbf{E}^T(\beta) \mathbf{k}_2(\beta, \eta, \zeta) \end{aligned} \quad (\text{B.19})$$

f_2 is a function only depending on $\theta, \beta, \eta, \zeta$

From Eqs. (B.1), (B.16) and (A.13), Eq. (2.42) can be reduced as follows:

$$\mathbf{V}^T \boldsymbol{\Sigma}(\beta) \dot{\eta} + \mathbf{V}^T \mathbf{R}(\theta) \mathbf{k}_1(\theta, \beta, \eta, \zeta) + \mathbf{I}_\beta \dot{\zeta} = \tau_1 \quad (\text{B.20})$$

From Eqs. (B.17) and (B.20), Eqs. (2.42) and (2.41) can be rewritten as follows:

$$\begin{cases} \mathbf{V}^T \boldsymbol{\Sigma}(\beta) \dot{\eta} + \mathbf{I}_\beta \dot{\zeta} + f_1(\theta, \beta, \eta, \zeta) = \tau_1 \\ \mathbf{H}(\beta) \dot{\eta} + \boldsymbol{\Sigma}^T(\beta) \mathbf{V} \dot{\zeta} + f_2(\theta, \beta, \eta, \zeta) = \boldsymbol{\Sigma}^T(\beta) \mathbf{E}^T(\beta) \mathbf{P} \tau_2 \end{cases} \quad (\text{B.21})$$

where

$$f_1(\theta, \beta, \eta, \zeta) = \mathbf{V}^T R(\theta) k_1(\theta, \beta, \eta, \zeta) \quad (\text{B.22})$$

From Eq. (B.21), the following is obtained.

$$\begin{bmatrix} \mathbf{V}^T \boldsymbol{\Sigma}(\beta) & \mathbf{I}_\beta \\ \mathbf{H}(\beta) & \boldsymbol{\Sigma}^T(\beta) \mathbf{V} \end{bmatrix} \begin{bmatrix} \dot{\eta} \\ \dot{\zeta} \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \boldsymbol{\Sigma}^T(\beta) \mathbf{E}^T(\beta) \mathbf{P} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (\text{B.23})$$



Appendix C

From Eq. (2.4), Eq. (2.6), Eq. (2.8), Eq. (2.9), Eq. (2.11) and Eq. (2.12), the following is obtained.

$$\begin{cases} \dot{x} \sin(\beta + \theta) - \dot{y} \cos(\beta + \theta) + l\dot{\theta} \sin \beta = 0 \\ \dot{x} \cos \theta + \dot{y} \sin \theta = 0 \\ \dot{x} \cos(\beta + \theta) + \dot{y} \sin(\beta + \theta) + l\dot{\theta} \cos \beta + r_s \dot{\phi}_1 = 0 \\ -\dot{x} \sin \theta + \dot{y} \cos \theta + b\dot{\theta} + r_f \dot{\phi}_2 = 0 \\ \dot{x} \sin \theta - \dot{y} \cos \theta + b\dot{\theta} + r_f \dot{\phi}_3 = 0 \end{cases} \quad (C.1)$$

From Eq. (C.1), the followings are obtained.

$$\begin{cases} \dot{x} = r_s \dot{\phi}_1 \sin \beta \sin \theta \\ \dot{y} = -r_s \dot{\phi}_1 \sin \beta \cos \theta \\ \dot{\theta} = -\frac{r_s}{l} \dot{\phi}_1 \cos \beta \end{cases} \quad (C.2)$$

From Eq. (C.1), the followings are obtained.

$$\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} r_s \sin \beta \sin \theta \\ -r_s \sin \beta \cos \theta \\ -\frac{r_s}{l} \cos \beta \end{bmatrix} \dot{\phi}_1 = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -r_s \sin \beta \\ -\frac{r_s}{l} \cos \beta \end{bmatrix} \dot{\phi}_1 \quad (C.3)$$

Choosing $\eta = \dot{\phi}_1$, $\Sigma(\beta) = \begin{bmatrix} 0 & -r_s \sin \beta & -\frac{r_s}{l} \cos \beta \end{bmatrix}^T$ and Eq. (2.4), the following can be obtained:

$$\dot{\xi} = \mathbf{R}^T(\theta) \Sigma(\beta) \eta \quad (C.4)$$