Thesis for the Degree of Doctor of Philosophy

Path Planning and Fault Detection for Automatic Guided Vehicle Based on Multiple Positioning Modules

by

Pandu Sandi Pratama

Department of Mechanical Design Engineering

The Graduate School

Pukyong National University

August 2015

Path Planning and Fault Detection for Automatic Guided Vehicle Based on Multiple Positioning Modules

다중 위치결정모듈에 기반한 무인운반차용 경로계획과 고장검출

by

Pandu Sandi Pratama

Advisor: Professor Sang Bong Kim

A thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

In the Department of Mechanical Design Engineering,
The Graduate School,
Pukyong National University

August 2015

Path Planning and Fault Detection for Automatic Guided Vehicle Based on Multiple Positioning Modules

A dissertation

by

Pandu Sandi Pratama

Approved as to styles and contents by:

(Chairman) **Yeon Wook Choe**

(Member) **Gi Sik Byun**

Y.S Jung
(Member) **Young Seok Jung**

S.oh
(Member) **Sea June Oh**

김상봉
(Member) **Sang Bong Kim**

May 2015

Acknowledgments

First of all, I would like to humble myself toward Allah SWT, the Loving and Merciful Lord. Because of His grace and help, I can carry on my study in Korea properly.

I would like to express the most gratitude to my supervisor, Professor Sang Bong Kim, for his valuable advice, guidance and support during the process of my studying in Korea. He has been providing a large amount of knowledge, the direction and skill of research, and strong motivation for my researching. His kindness, insight and continuous supports encouraged and helped me to accomplish my research and finish this dissertation scientifically.

I would like to thank the members of my thesis committee: Prof. Yeon Wook Choe, Prof. Gi Sik Byun, Prof. Sea Jun Oh, and Prof. Young Soek Jung who have provided wonderful feedback on my work and great suggestions for better contribution of my dissertation.

I would like to thank Prof. Hak Kyeong Kim for his great helps and advice to research and complete this dissertation. I could not finish my dissertation on time without his great help and advice.

I would like to thank all members of CIMEC Lab. for giving me a comfortable and active environment to achieve my work: Dr. Phuc Thinh Doan, Dr. Hoang Giang, Dr. Than Luan Bui, Van Tu Duong, Nguyen Trong Hai, Chetanraj D. Patil, Woo Yeong Jeong, Yu Mi Park, Min Ji Bae, Jeong Geun Kim, Jung Hu Min, Quang Tuyen Tran, Yuhanes Dedy Setiawan, Amruta Vinod Gulalkari, Jae Hun Jeong, Jin Wook Kim, Seong Gon Jeong, Husam Hasan Aldulaimi, Yun Seok Lee, and all other foreign friends.

Finally, I would like to thank to my father, Sutiyoso, S.T., my mother, Ngesti Handayani Lestari, my brother, Satria Yudha Pratama, and all my close relatives for their love, endless pray, encouragement and support for me not only in the dissertation time but also in the whole of my life. Thank you to all Indonesian students who lived in Busan. And lastly, thank you to all the people who cannot be mention here for the help and support.

Pukyong National University, Busan,
Korea

May 6, 2015

Pandu

Pandu Sandi Pratama

Contents

Path Planning and Fault Detection for Automatic Guided Vehicle Based on Multiple Positioning Modules

Pandu Sandi Pratama

Department of Mechanical Design Engineering,
The Graduate School, Pukyong National University

Abstract

Automatic guided vehicles (AGV) increase the material handling efficiency and reduce the costs by automating manufacturing facilities or warehouse. To make the AGV move automatically, trajectory tracking control for AGV is needed. Furthermore, safety and reliability of AGV is very important factor in AGV operation. To guarantee the safety and reliability of AGV, a fault detection algorithm is required.

To solve this problem, this dissertation presents an implementation and its experimental validation of path planning, trajectory tracking, positioning and fault detection algorithm for sensors and actuators of AGV system based on multiple positioning modules. To do this task, the followings are done. To design the trajectory tracking controller, mathematic modeling of the system is needed. Firstly, in this dissertation, system description and mathematical modelings of a differential drive AGV system are presented. The system description consists of mechanical design, electrical design and software design. Kinematic model of AGV is derived based on the wheel configuration and the nonholonomic constrain. Dynamic model of AGV is obtained from the Lagrangian formula. Secondly, to make the AGV move automatically, a desired path is needed. This dissertation focuses only in coverage path planning problem. This path planning algorithm can be applied to a cleaning robot, a mining robot, etc. Coverage path planning is defined as the task of finding a path that passes all points on a given area. Moreover, energy consumption also has to be considered to guarantee the optimal performance. To solve this problem, a new coverage path planning algorithm based on multi-spanning tree algorithm is proposed. In the coverage navigation of AGV, energy consumption is depends on the number of acceleration and deceleration and the total path length. The main idea to solve this problem is by reducing the turning number and the overlapped path. To do this, a minimal sum altitude algorithm (MSA) is applied to the Morse cell decomposition to minimize the turning number. Furthermore, a modification of spanning tree algorithm is introduced to minimize the overlapped path. The simulations are done to verify the superiority of the proposed algorithm compared to the existing algorithms such as vertical and horizontal cell decomposition, and conventional spanning tree algorithms. Thirdly, two trajectory tracking controllers are proposed. A first trajectory tracking controller is an adaptive backtepping controller. By choosing appropriate Lyapunov function based on its kinematic modeling of AGV with unknown slip parameter, system stability is guaranteed and a control law can be obtained. To solve this problem, an update law is proposed to estimate the unknown slip parameter. A second trajectory tracking controller is a robust servo trajectory tracking controller designed using polynomial differential operator based on internal model principle. The basic idea of the internal model principle is described. The extended system including reference and disturbance in polynomial differential equation form is introduced. The controllability checking of the extended system is done. The state feedback law is obtained by a well known regulator design method. Simulation and experimental are done to verify the effectiveness of the proposed trajectory tracking controller. Simulation and experimental results shows that the proposed controllers successfully make the AGV tracks the generated path. Fourthly, to understand the characteristics of the sensors, mathematic models of the positioning sensors are proposed. The AGV uses positioning modules such as encoder, laser scanner, and laser navigation system to obtain its position information. The basic principle and error condition of each sensor is introduced. Fifthly, a fault detection algorithm based on multiple positioning modules is proposed. The proposed fault detection method uses two or more positioning modules. In this algorithm, Extended Kalman Filter (EKF) is used to detect unexpected deviation of measurement results of two modules are affected by fault. The pairwise differences between the estimated positions obtained from sensors are called as residue. When faults occur, the residue value is greater than the threshold value. Fault identification is obtained by examining the biggest residue. Finally, to demonstrate the capability of the proposed algorithm, it is applied to a differential drive AGV system. The simulation and experimental results show that the proposed algorithm successfully detects faults when the faults occur. Finally, conclusions are presented, and the future works of the proposed algorithms are described.

Keywords: Automatic Guided Vehicles, Differential drive, Extended Kalman filter, Fault detection, Path planning, Tracking control.

List of Figures

List of Tables

Nomenclatures

Variables	**Descriptions**	**Units**
X_A, Y_A	position of AGV in the global reference frame is specified by coordinates	m
θ_A	the angular difference between the global and local reference frames	rad
ξ_1	position vector with position and orientation of the AGV in global coordinate	
ξ_A	position vector with position and orientation of the AGV in local coordinate	
$\mathbf{R}(\theta_A)$	orthogonal rotational matrix	
l	distance from center of AGV to the wheel	m
ϕ	rotational angle around AGV horizontal axle	rad
β	the angle of the wheel plane relative to the chassis	rad
V_A	linear velocity of AGV	m/s
ω_A	angular velocity of AGV	rad/s
r	wheel radius	m
v_L, v_R	linear velocities of the left wheel and the right wheel	m/s
$\dot{\phi}_L, \dot{\phi}_R$	angular velocities of the left wheel and the right wheel	rad/s
$\boldsymbol{\eta}_A$	AGV velocity vector in global coordinate	
τ_R, τ_L	torques exerted on right and left wheels	N/m
m_c	mass of the body without driving wheels	kg

m_w	mass of each wheel	kg
I_m	moment of inertia of each motor	$\mathrm{kg \cdot m^2}$
I_w	moment of inertia of each wheel	
I_c	moment of inertia of the body	
d	distance between geometric center and mass center	m
$\mathbf{D}$	occupancy grid map	
$p(O_i)$	probability that cell i in the occupancy grid map	
$A(i)$	altitude of the i^{th} cell	m
h_i	length of the i^{th} cell	m
a_i	width of the i^{th} cell	m
$Dir(i)$	sweep direction of the i^{th} cell	
$S(j)$	sum of altitude of the j^{th} way to split the graph	m
MSA	minimum value of sum of altitude	m
N	total turning number to cover all area	
L	AGV width	m
x_r, y_r	reference position of AGV in global coordinate	m
θ_r	is AGV reference orientation	rad
V_r	reference linear velocity	m/s
ω_r	reference angular velocity	rad/s
i	slip ratio of two wheels	
c	slip parameter	

$\mathbf{e}_c$	kinematic tracking error vector	
$\boldsymbol{\eta}_\mathbf{d}$	desired kinematic control law vector	
$\mathbf{e}_d$	dynamic tracking error vector	
$\Delta\phi_l$, $\Delta\phi_r$	incremental angles for the left and right wheel	rad
C_m	conversion factor that translates encoder pulses into wheel rotation angle	
g	gear ratio of the reduction gear between the motor where the encoder is attached and the drive wheel	
Δx , Δy	linear displacements of AGV	m
$\Delta\theta$	the change of rotational angle of AGV	rad
k_r , k_l	error constants related to incremental encoder obtained from experiment	
k_x , k_y , k_θ	error constants related with the AGV motion obtained from experiment	

Chapter 1: Introduction

1.1 Background and motivation

Automation is a key aspect for increasing efficiency, improving service quality and saving from both time and manpower. The basic aim of automation technologies is to replace human with machines to perform dirty, difficult or dangerous task. Transportation of goods in indoor environments is one of the most essential examples for these tasks. To do these tasks, an Automatic guided vehicles (AGVs) are needed. An automatic guided vehicle system (AGV) is defined as a materials handling system that uses independently operated, self-propelled vehicles that are guided along defined pathways in the floor [1]. The applications of AGV are various from the factory applications [2–4], hospital applications [5–7], agriculture applications [8–10], and transportation applications [11]. To make the AGV move automatically, a path planning algorithm, a trajectory tracking control and a positioning method are needed. Furthermore, to guarantee the safety and reliability, a fault detection algorithm is required.

Path planning and navigation are a common problem in an Automatic Guided Vehicle (AGV) operation. One of the path planning problems is a coverage path planning. The coverage path planning is defined as the task of finding a path that passes all points on a given area. There are many commercial applications using the coverage path planning algorithm such as aerial survey for agriculture [12], aerial mapping [13], arable farming [14], agriculture

[15], automatic harvesting [16], lawn mowers [17], vacuum cleaner [18], demining robot [19-20], painting [21], window cleaning [22], underwater inspection of complex structure [23], underwater mine countermeasure mission [24], underwater mapping [25] and mining robot [26–29]. There are considerable methods has been proposed to solve the coverage path planning problem [30]. The first method is cell decomposition-based methods such as Boustrophedon [31], Morse cell decomposition [32] and critical point detection [33]. In cell decomposition-based method, it is assumed that the environment has a simple structure and can be defined as non-polygonal space. The cell decomposition can be obtained by sweeping the line from left to right to find the critical points of the polygon obstacles and then drawing lines at these critical points which can be extended vertically in both upward and downward directions. After the given area is decomposed into cells, the cell covering sequence is calculated using optimal method such as deep first search (DFS) algorithm [34], genetic algorithm [35] and ant colony optimization algorithm [36-37] to minimize the overlap. The second method is grid-based methods such as wave front [38] and spanning tree [39–41] algorithms. In wave front algorithm, a distance transform that propagates a wave front from goal to start is used to assign a specific number to each grid element. In spanning tree algorithm, it divided the free area into small squares and computed the spanning tree of the resulting connectivity graph using Prim algorithm [42]. Once the spanning tree is made, the path is obtained by starting at a point near the spanning tree and drawing the lines along its perimeter.

To track the generated trajectory, a trajectory tracking control is needed. In conventional AGV system, there were several kinds of navigation methods to track the generated trajectory such as a vision based line tracking algorithm [43-44], inductive guidance using electrical wire buried under the floor [45], semi-guided navigation methodology by using magnetic tapes [46], navigation method based on wall following algorithm [47] and laser navigation system [48–50]. To track the given trajectory, several control algorithms has been proposed. A PID controller for trajectory tracking was proposed by [51]. The navigation of AGV using fuzzy control was proposed in [47, 52]. Those controllers are easy to be applied to AGV system but not robust. A parameter-based controller design method has been proposed using sliding mode control theory in [53] and using Lyapunov stability in [54-55]. In those controllers, the stability of the system was guaranteed, but to find an appropriate controller law was not an easy task. Therefore, a backstepping control method was proposed in [56]. In the backstepping control method, by choosing appropriate Lyapunov equation, the system stability is guaranteed and the control law can be obtained. In parameter-based controller design, all parameters of the system should be known. However, in real AGV application, some parameters such as slip, wheel diameter, wheel distance and mass are changed during the operation caused by uneven load distribution or manufacturing imperfection, and loading the goods. Therefore, the controller should be adaptive to the changing of parameters. An Adaptive tracking controllers for the dynamic model of a nonholonomic mobile robot were proposed in [57-58] and an adaptive controller for a partially known system for applying it to a two-wheeled welding mobile robot was proposed in [59].

The conventional navigation systems using vision-based line tracking, inductive guidance using electrical wire buried under the floor, and semi-guided navigation methodology by using magnetic tapes have serious limitation. Those navigation systems are not flexible since their path can't be changed easily. Therefore, there are many researches focused on flexible navigation based on the localization sensor. For a wheeled robot, odometry (also known as dead-reckoning) is one of the most popular approaches to achieve this task [60-61]. In practice, optical encoders that are mounted on both drive wheels send discretized wheel increment information to a processor, which continually updates the robot's state using geometric equations. However, with time, odometric localization accumulates errors unbounded due to wheel slippage, floor roughness and discretized sampling of wheel increments. There are many different kinds of positioning technologies using wireless networks such as Global Positioning System (GPS), cellular phone tracking system, WiFi positioning system and RFID Positioning System. All these technologies have different coverage, applications, accessories and limitations. The nature of indoor positioning systems is somewhat different from that of outdoor ones. Typically, the coverage of a positioning system is inversely proportional to its detection accuracy.

Among these technologies, the most popular positioning system is GPS [62]. It is a satellite-based positioning system which is designed for outdoor environment. However, it does not perform well indoors. The GPS signal is easily blocked by most construction materials and hence useless for indoor positioning. There is a problem of accuracy and absolute positioning in a specific indoor area. For indoor environments, positioning systems that rely on the existing network infrastructures such as WiFi positioning system and Bluetooth are able to provide location accuracies ranging from 1 to 10 m, depending heavily on the usage environment [63]. The WiFi network must exist as a part of the communication infrastructure. Otherwise, it will require expensive and time consuming infrastructure deployment. Deriving an accurate propagation model for each WiFi access point in a real world indoor environment is extremely complex, and therefore usually results in a relatively poor positioning accuracy.

Other than the issue of inaccuracy problem in indoor location detection, cost is another challenge in designing indoor positioning systems. Given the fact that most positioning systems are designed for tracking few robots, there is no cost effective way to track a lot of robots. For example, even though WiFi positioning system can be utilized indoors, its development and setup costs are very high when it covers a large area because it requires deployment of expensive WiFi tags for tracking the robot. If the objects to be tracked change frequently, the operation cost of transferring tags for WiFi positioning applications becomes very high.

The indoor RFID positioning issue is a significant research topic [64-65]. An RFID system contains RFID readers and tags and a communication media between them. Each reader has a predetermined power level which corresponds to a certain range where it can detect RFID tags. The readers are placed in known positions dividing the region to subregions. The accuracy of the system depends on the number of these readers and their placements.

Since wireless network-based technologies need an active transmitter and a receiver, the development and the setup cost of a wireless network positioning system are too high to be economically viable. Thus, designing a cost effective indoor positioning system has remained an open challenge so far. Therefore, Laser navigation system as a laser based positioning system was proposed [66–69]. It returns an absolute position of the scanner with respect to a user-defined local coordinate frame. Nav200 system as a laser based positioning system comprises of two main units, a laser scanner unit and a Transputer Position Unit (TPU). This laser scanner continuously rotates 360 degrees to scan passive reflectors and returns its position information to the TPU to compute the scanner's position. On average, this system can provide up to centimeter accuracy with an update rate up to 8Hz.

An interesting approach to solve flexible navigation is a simultaneous localization and mapping (SLAM) method [70–72]. The simultaneous localization and map building (SLAM) problem is concerned with localizing a mobile robot in an unknown workplace and the robot is required to estimate its location as well as the feature locations within its operating environment. In this approach, the reflectors are replaced by distinctive features in the environment such as pillar, column, wall, etc. To detect those features, camera [73], sonar [74], and laser measurement system [71] are used.

The AGV which are working autonomously have to work nearby human and interacts with human. In addition, the AGV can often deal with dangerous task such as transferring volatile materials. Therefore, reliability and safety are the most important parameters in

AGV operation. In industrial applications, a fault detection algorithm reduces productivity loss, helps operator to detect faults and prevent serious damage on the system. The objective of the fault detection algorithm is to develop a system which detects faults as quick as possible with minimum false alarms [75].

Various approaches of fault detection algorithms in different applications have been extensively reported in several literatures. In [76], a fault detection algorithm based on local model neural networks was proposed. This algorithm used local model neural networks (LMNs) to deal with the absence of a mathematical model. However, its computation burden was high, and the model that has been produced by this method depended on the training. For the known mathematical model, a fault detection algorithm based on an unknown input PI observer was proposed in [77-78]. In this algorithm, states of system were estimated with the unknown input PI observer, and faults of system were detected by the fault detection algorithm. This algorithm is easy to apply but only works for linear descriptor system. A real-time model based sensor fault detection algorithm for unmanned ground vehicles was proposed in [79] which used multiple sensors such as GPS, IMU and encoder. However, this method only operated on a sensor level. Another approach applied temporal causal graphs (TCGs) proposed in [80]. The TCG constructed from a bond-graph model of physical process provided a basis for isolation methodology. In [81-82], a particle filter was used for fault detection and isolation. Several discrete states were used for each fault mode. Particle filters were very powerful for tracking systems, and produced a probability distribution over the states, given as samples (particles). It can be used for nonlinear, non-Gaussian and multi modal distributions. However, the computational demands for this algorithm were increased according to the state dimension.

Finally, from the above discussions, to make the AGV move automatically, firstly, a desired path is needed. The path planning algorithm has to find a path that passes all points on a given area and consider the energy consumption. Secondly, a trajectory tracking controller that adaptive to the changing of parameters is needed. Finally, to guarantee the safety and reliability of AGV, a fault detection algorithm is required.

1.2 Objective and researching method

From the above discussions, this dissertation is devoted to present an implementation and its experimental validation of path planning, trajectory tracking, positioning and fault detection algorithm for sensors and actuators of an Automatic Guided Vehicle (AGV) system based on multiple positioning modules. To do these tasks, system modeling, path planning, control algorithm design, fault detection algorithm design and simulation and experiment results to evaluate the proposed algorithm are presented.

A two-wheel differential drive system is one of the simplest and most used structures in mobile robotics applications. It consists of a chassis with two fixed wheel in-line with each other and propelled by electric motors. The differential drive AGV kinematic and dynamic models are derived based on the wheel configuration and the AGV constraints.

Path planning researches focus in finding a path to visit all the points in given area, minimizing overlapped paths, journey time and turning number. To solve this problem, this dissertation proposed a new algorithm using a multi-spanning tree method. To do this, an occupancy grid map is used to represent an obstacle position in the environment. Secondly, Morse cell decomposition method is applied to divide the whole area of work space into cells in the vertical and horizontal directions. To minimize the turning number, a minimal sum of altitude method is applied to the combination of vertical and horizontal cell decompositions. Finally, a spanning tree algorithm is applied to smaller work area and to each cell.

To solve trajectory control problem, this dissertation proposed two trajectory tracking controllers. In this dissertation, a first trajectory tracking controller is an adaptive backtepping controller. By choosing appropriate Lyapunov function based on its kinematic modeling of AGV with unknown slip parameter, system stability is guaranteed and a control law can be obtained. To solve this problem, an update law is proposed to estimate the slip in case of the unknown slip parameter. A second trajectory tracking controller is a robust trajectory servo controller obtained by modifying the concept of the internal model principle. By operating a polynomial differential operator for the state space model and error signal, its extended system is obtained. The servo controller for a given system can be easily designed by a regulator design method using pole assignment.

Usually, a fault detection algorithm only works in sensor level. Therefore, there are different algorithms to detect each sensor fault. In this dissertation, a fault detection algorithm to detect all sensor fault conditions is proposed. The proposed fault detection method uses two or more positioning systems and compares them using Extended Kalman Filter (EKF) to detect unexpected deviation effected by fault. Residue is obtained from the pairwise differences between the estimated positions obtained from sensors. When faults occur, the residue value is greater than the threshold value. Fault isolation is obtained by examining the biggest residue. Simulation and experimental results of several fault conditions are presented. The experiments of fault condition in motor, laser scanner, laser positioning and encoder are presented to verify the effectiveness of the proposed fault detection algorithm. The experimental result shows that the proposed algorithm successfully detects faults when the faults occur.

1.3 Outline of dissertation and summary of contributions

Chapter 1: Introduction

In this chapter, background and motivation of the Automatic Guided Vehicle system is described. Objective and researching method are presented. And the outline of content and summary of contribution of this dissertation is given.

Chapter 2: System Description and Modeling

This chapter provides the description of a differential drive AGV and its hardware configuration. Kinematics of differential drive AGV is presented. The kinematics model presented is derived based on the wheel configuration and nonholonomic constraint. The dynamic model of the differential drive AGV is also provided using Lagrange formula.

Chapter 3: Coverage Path Planning

This chapter proposes a coverage path planning algorithm to minimize the energy consumption. The main idea to solve this problem is to reduce the turning number and the overlapped path. To solve this problem, this dissertation proposed a new algorithm using a multi-spanning tree method. To do this, a minimal sum of altitude algorithm is applied to the combination of vertical and horizontal Morse cell decompositions to minimize the turning number. Furthermore, modification of a conventional spanning tree algorithm is introduced to minimize the overlapped path. The simulation comparisons among the proposed algorithm, the vertical cell decomposition, horizontal cell decomposition, and a conventional spanning tree algorithm are done to verify the superiority of the proposed algorithm.

Chapter 4: Tracking Control Design

This chapter proposes two trajectory tracking controller design methods needed to control the AGV. In the first section, an adaptive backstepping control method for AGV with unknown slip parameter is proposed. A kinematic control law based on kinematic modeling, a dynamic control law based on dynamic modeling using Lagrange formula and update law are designed based on backstepping method using Lyapunov function. In the second section, a robust servo trajectory tracking control design method using polynomial differential operator based on internal model principle is proposed based on a system modeling combining the kinematic modeling and the dynamic modeling. By operating the polynomial differential operator to a given system with disturbance, an extended system is obtained and a feedback control law of the obtained system is design by a well known regulator design method. Finally, the simulation and experiment results of the proposed controllers are shown to track a given trajectory generated by the coverage path algorithm in chapter 3. Simulation and experimental results show that the proposed controllers successfully track the generated trajectory.

Chapter 5: Sensor Model

This chapter provides the characteristics of positioning sensors used in this dissertation. Working principles and mathematical models of each sensor are proposed. Firstly, the basic principle and the mathematic model of an encoder positioning method is introduced. The AGV position is obtained by counting the encoder pulse from the left and the right wheels. Secondly, the basic principle and the mathematic model of the LMS positioning are introduced. The AGV position can be obtained from the landmark position obtained from the environment. Finally, a positioning method and mathematic model of the NAV is described. The AGV position can be obtained from predefined reflectors placed on the environment.

Chapter 6: Fault Detection Algorithm

In this chapter, a fault detection algorithm to detect the fault conditions in AGV sensors and actuator is proposed. The proposed fault detection method uses two or more positioning modules. In the first section, Extended Kalman Filter (EKF) used to detect unexpected deviation of measurement results of the positioning modules are described. Secondly, the residue value is calculated, and the threshold value is introduced. Thirdly, fault isolation is introduced based on the residue value. Fault isolation is obtained by examining the biggest residue. Finally, to demonstrate the capability of the proposed algorithm, the simulation and experiment are done. The simulation and experimental results show that the proposed algorithm successfully detects faults when faults occur.

Chapter 7: Conclusions and Future Works

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

Chapter 2: System Description and Modeling

This chapter describes the design prototype of the experimental differential drive automatic guided vehicle (AGV) system. This AGV prototype design consists of mechanical design and electrical design. The electrical design consists of sensors, controller, actuator and power supply. This chapter also presents the mathematical modeling of the proposed AGV system. The mathematic modeling consists of kinematic modeling and dynamic modeling.

2.1 System description

2.1.1 Mechanical design

The AGV system used in this dissertation is shown in Fig. 2.1. The AGV used in this dissertation has two driving wheels on the left and right sides of AGV are driven by BLDC motor.

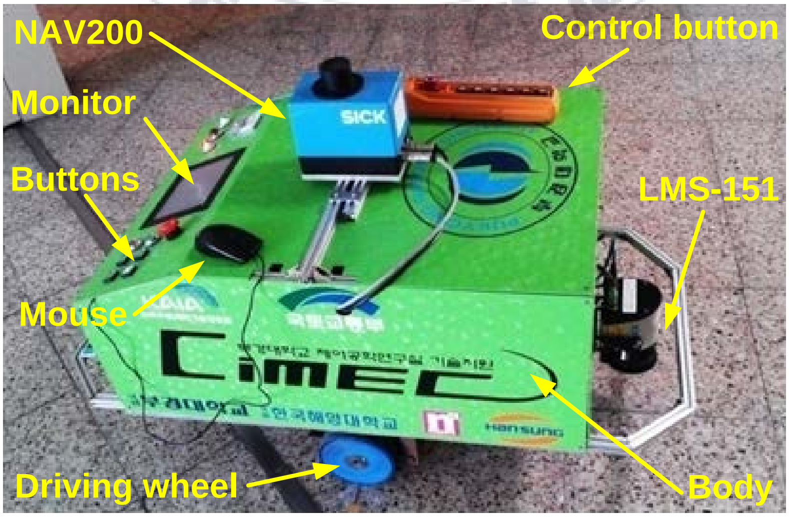


Fig. 2.1 Mechanical design of differential drive AGV

This AGV has dimension 60 cm x 100 cm x 190 cm. This system uses differential drive system. The laser navigation system NAV-200 is mounted on the top of AGV. The laser measurement system LMS-151 is mounted in front sides of AGV. Industrial PC is placed inside the AGV platform. The batteries are placed on the middle of AGV. Touch screen monitor is placed on back side of AGV. Push buttons are provided to connect and disconnect the system electricity. Moreover, the emergency button is installed for safety purposes. In case that the vehicle has to be turned off, this button works as the emergency button which disconnects all electricity in the system if it is pushed.

2.1.1.1 Body frame

The body configuration of the differential drive AGV is shown in Fig. 2.2. The body frame is made of 20 mm x 20 mm aluminum profiles. Aluminum profile is chosen because it is lightweight, strong, non-corrosive, and it can be realizes complex shapes only with dozens of bolt.

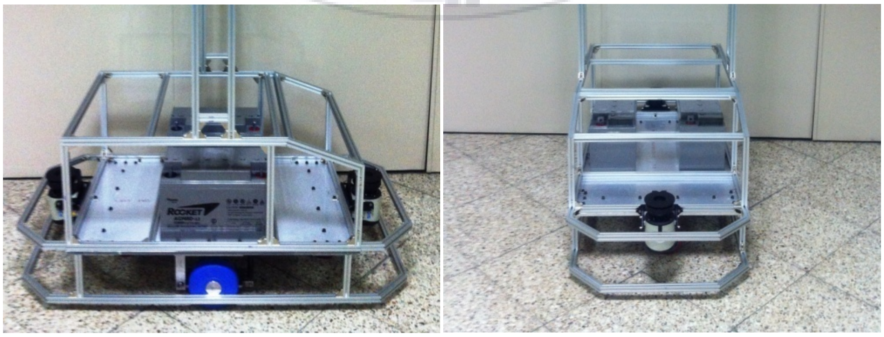


Fig. 2.2 Body frame

2.1.1.2 Wheel configuration

The wheel configuration of this system is shown in Fig. 2.3. This system consist of two driving wheels are mounted on the left and right sides of AGV, and are driven by two BLDC motors. Two passive castor wheels are installed in front and back sides of AGV to support the AGV. The bottom plate is made from 20mm aluminum plate. Suspension is designed to reduce the shock when the AGV move.

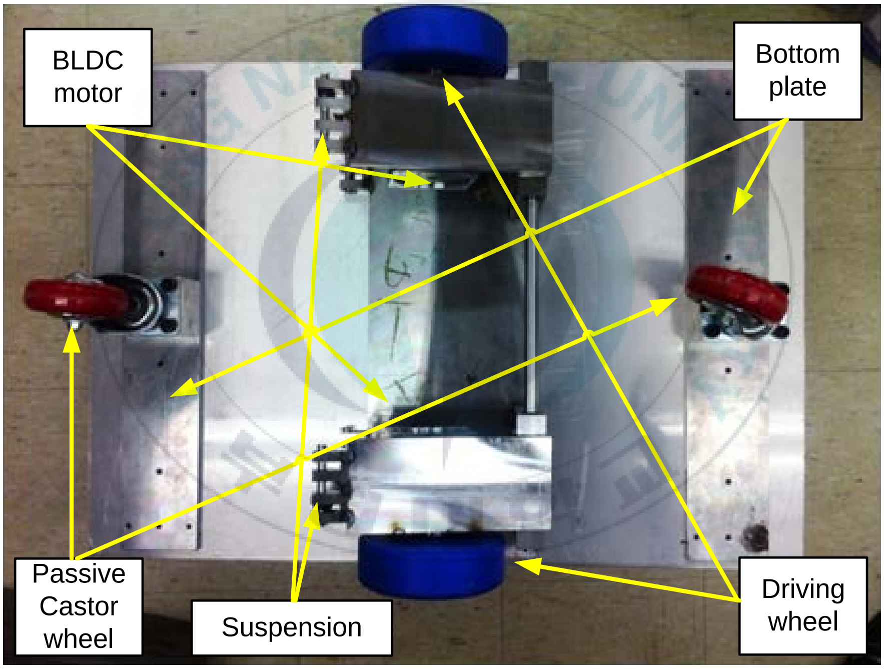


Fig. 2.3 Wheel configuration

2.1.2 Electrical design

Electrical configuration of the proposed control system used for this dissertation is shown in Fig. 2.4.

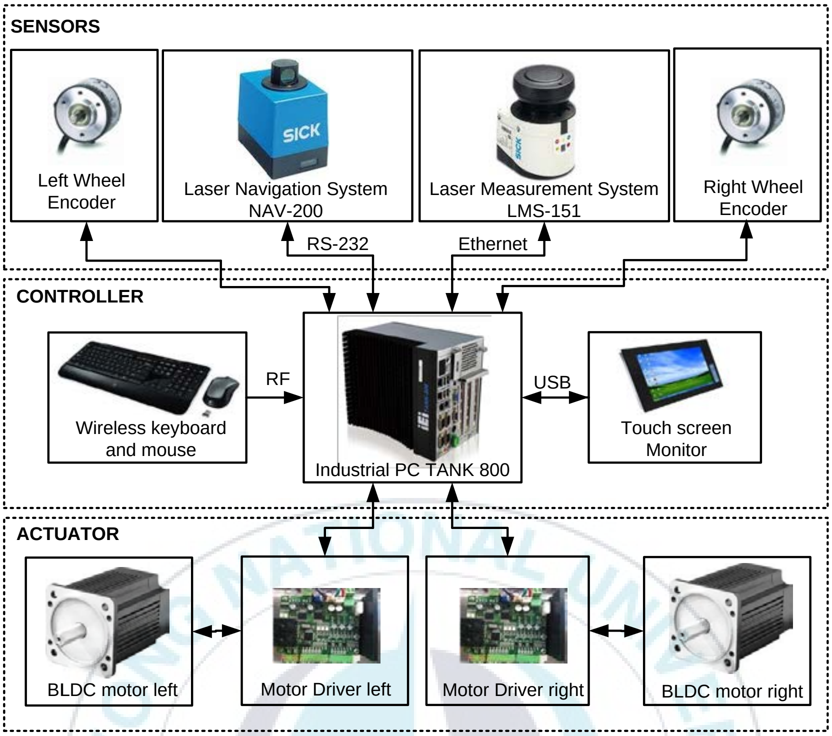


Fig. 2.4 Electrical configuration of AGV

Fig. 2.4 shows the electrical configuration of the AGV. This configuration consists of sensor part, controller part, and actuator part. The sensors part consists of laser navigation system NAV-200 and laser measurement system LMS-151. Laser navigation system NAV-200 is connected with industrial PC via RS-232 serial data communication connections. The transmission rate is 19,200 Hz. Laser measurement systems LMS-151 is connected with industrial PC via TCP/IP interface. This Ethernet interface has a transmission rate 100 Mbit. The controller part consists of industrial PC Tank 800 that is used as main controller and the wireless keyboard and mouse that are used as the input. To show the calculation and monitoring process, the touch screen monitor is used as display. The industrial PC sends analog signal and logic signals to actuator. The actuator part consists of motor driver and BLDC motor. The motor driver receives reference analog signal from controller, and then it controls the speed of BLDC motor based on reference value.

2.1.2.1 Laser measurement system

The structure of LMS-151 is shown in Fig. 2.5 and the specification of LMS-151 is shown in Table 2.1.

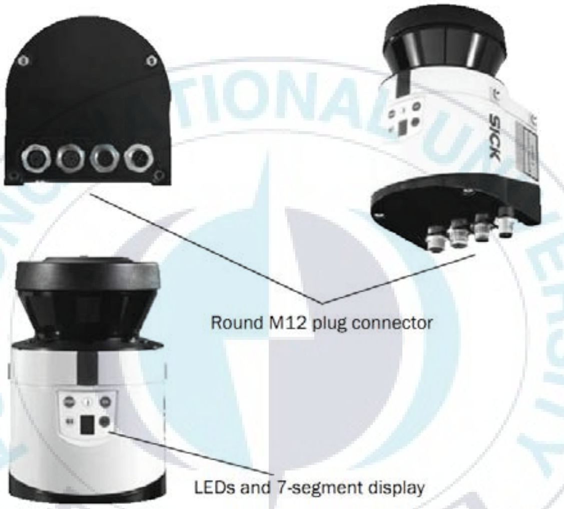


Fig. 2.5 Laser measurement system LMS-151

Table 2.1 Specification of laser measurement system LMS-151

No	Parameter	Value
1	Light source	Infrared (905 nm)
2	Laser class	1 (IEC 60825-1 (2007-3))
3	Field of view	270 °
4	Scanning frequency	25 Hz / 50 Hz
5	Angular resolution	0.25°/0.5°
6	Operating range	0.5 m - 50 m
7	Data communication	Ethernet
8	Data transmission rate	100 MBit
9	Operating voltage:	10.8 V DC - 30 V DC (24 V)
10	Power consumption:	60 W
11	Weight	1.1 kg
12	Dimensions	105 mm x 102 mm x 162 mm

2.1.2.2 Laser navigation system

The second sensor used in this dissertation is laser navigation system NAV-200. The structure of laser navigation system NAV-200 is shown in Fig. 2.6.

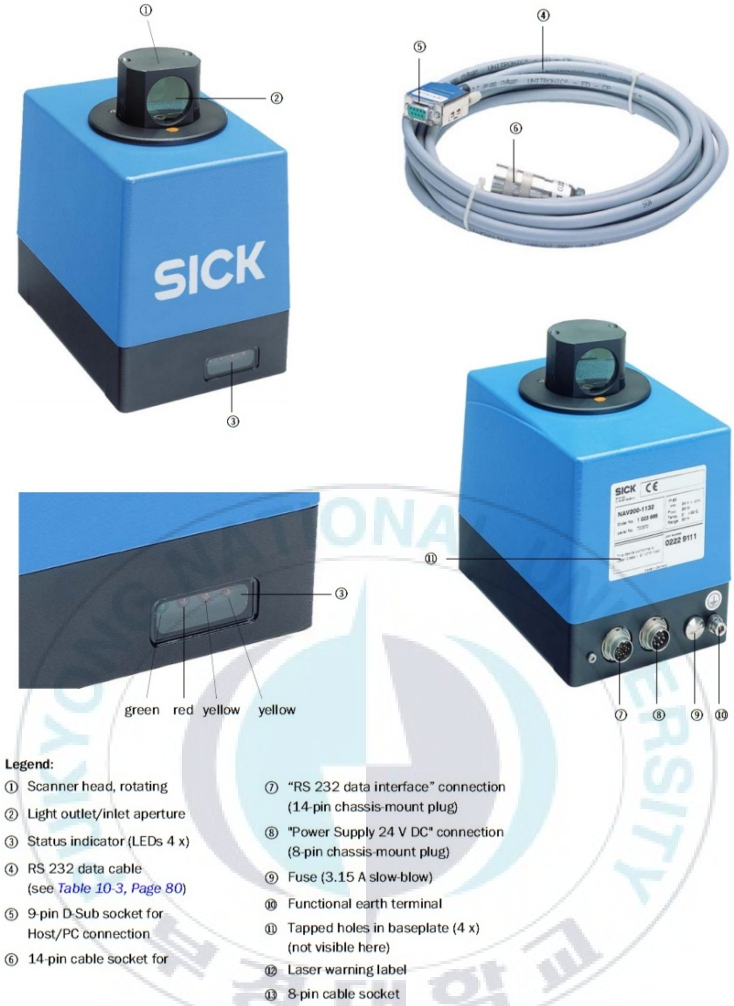

①
②
④
⑤
⑥
SICK
③
③
⑪
green
red
yellow
yellow
⑦
⑧
⑨
⑩
Legend:
① Scanner head, rotating
② Light outlet/inlet aperture
③ Status indicator (LEDs 4 x)
④ RS 232 data cable (see Table 10-3, Page 80)
⑤ 9-pin D-Sub socket for Host/PC connection
⑥ 14-pin cable socket for
⑦ "RS 232 data interface" connection (14-pin chassis-mount plug)
⑧ "Power Supply 24 V DC" connection (8-pin chassis-mount plug)
⑨ Fuse (3.15 A slow-blow)
⑩ Functional earth terminal
⑪ Tapped holes in baseplate (4 x) (not visible here)
⑫ Laser warning label
⑬ 8-pin cable socket

Fig. 2.6 Laser navigation system NAV-200

The specification of laser navigation system NAV-200 is shown Table 2.2.

Table 2.2 Specification of laser navigation system NAV-200

No	Parameter	Value
1	Light source	Infrared (855 nm)
2	Laser class	1
3	Field of view	360 °
4	Scanning frequency	8 Hz
5	Angular resolution	0.1°
6	Operating range	1.2 m – 28.5 m
7	Max. range 10 % reflectivity	28.5 m
8	Data communication	Serial (RS-232)
9	Data transmission rate	19200 Hz
10	Operating voltage:	≥ 24 V DC ± 25%
11	Power consumption:	24 W
12	Weight	3.3 kg
13	Dimensions	176 mm x 178 mm x 115 mm

2.1.2.3 Controller

In this dissertation, industrial PC TANK-800 of the controller part is used as main controller. The structure of industrial PC used as a controller for this dissertation is shown in Fig. 2.7.

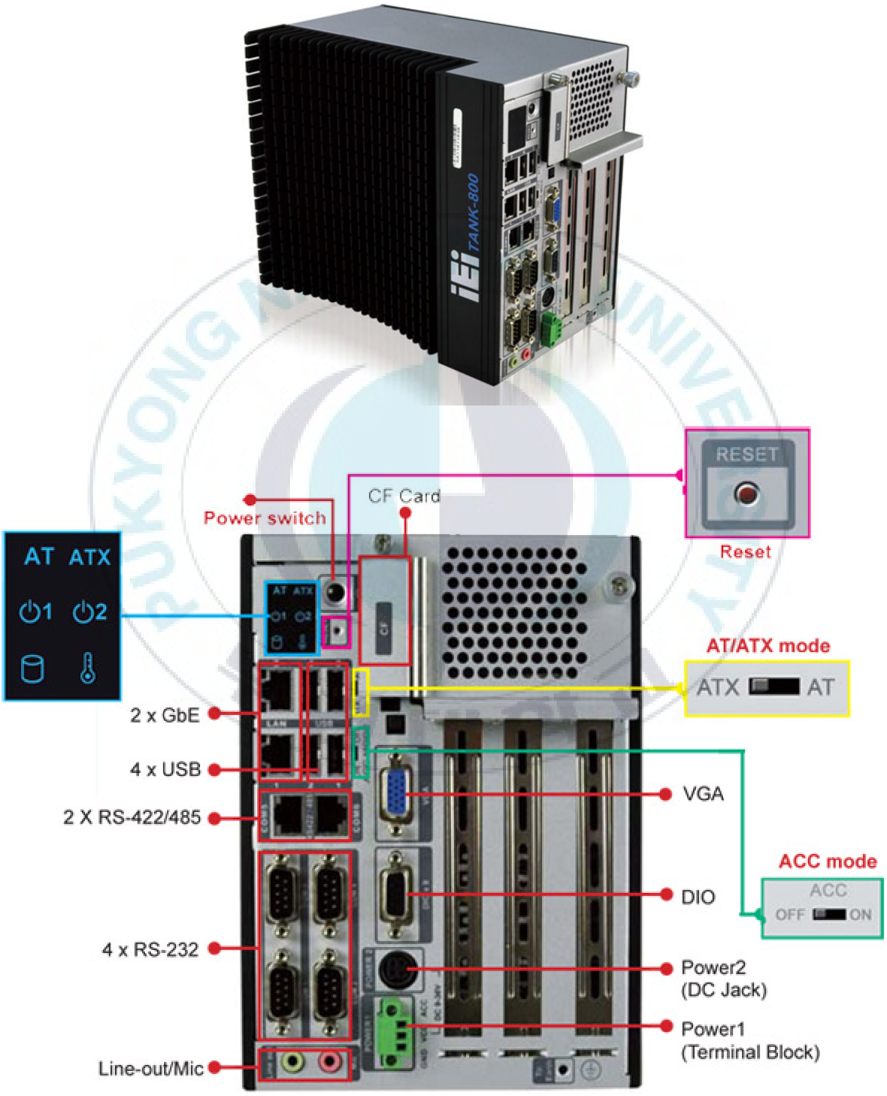


Fig. 2.7 Industrial PC TANK-800

The specification of Industrial PC TANK-800 is shown in Table 2.3.

Table 2.3 Specification of PC TANK-800

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

2.1.2.4 Actuator

The actuator consists of motor driver and BLDC motor. Each motor uses 24VDC power supply. The maximum current allowed by motor driver is 20A. The motor driver used for this dissertation is shown in Fig. 2.8.

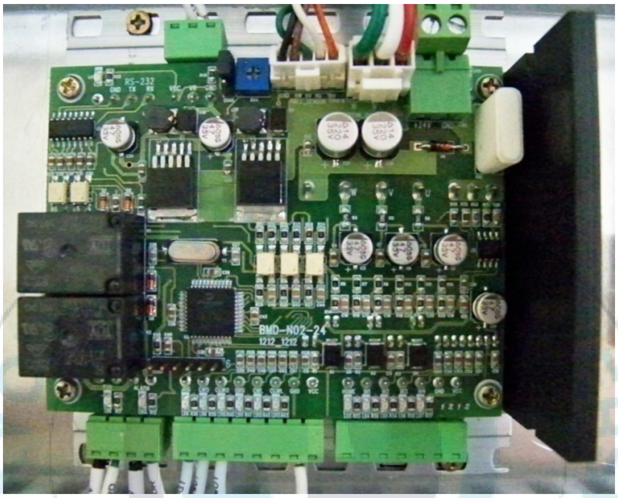


Fig. 2.8 Motor driver

The structure of BLDC motor used for this dissertation is shown in Fig. 2.9.

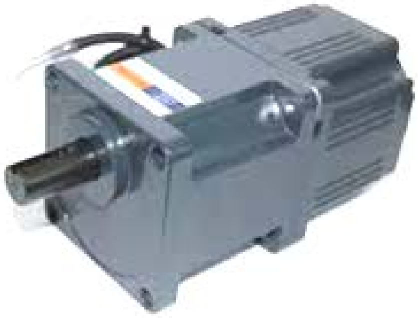


Fig. 2.9 BLDC motor

The specification of BLDC motor SWIS BG90-S024 is shown in Table 2.4.

Table 2.4 Specification of BLDC motor SWIS BG90-S024

No	Parameter	Rating
1	Rated Voltage	24 V
2	Rated Output	200 W
3	Rated Current	11.9 A
4	Rated Speed	3000 RPM
5	Rated Torque	6.3 Kgf-cm
6	Weight	2.7 Kg
7	Voltage constant	14.8 v/Krpm
8	Torque Constant	1.4 kgf.cm
9	Rotor inertia	0.24x10^{-3} kg.m^{2}
9	Armature resistance	0.9 ohm
10	Armature inductance	0.9 mH
11	Mechanical time constant	10 ms
12	Electrical time constant	1.1ms

2.1.2.5 Power supply

In AGV system, all devices need 24V power supply. The battery structure is shown in Fig. 2.10. The specification of battery Rocket AGM 80-12 is shown in Table 2.5.

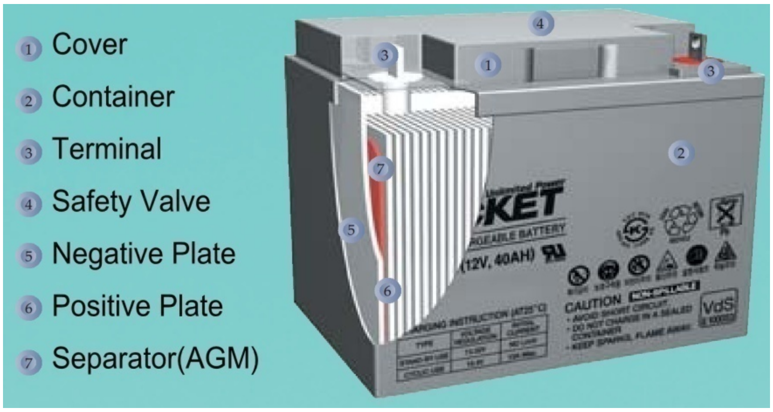


Fig. 2.10 Structure of Battery

Table 2.5 Battery Rocket AGM 80-12 parameters

No	Parameter		Value
1	Nominal Voltage (V)		12 V
2	Capacity (AH)	10HR 1.80(V/Cell)	80.0 AH
		5 HR 1.70 (V/Cell)	68.0 AH
		3 HR 1.67 (V/Cell)	61.5 AH
		1 HR 1.60 (V/Cell)	48.0 AH
		0.5 HR 1.60 (V/Cell)	37.5 AH
3	Dimension (mm)	Length	332 mm
		Width	174 mm
		Height	229 mm
		Total Height	229 mm
4	Weight		24.7 Kg
5	Terminal type		Bolt terminal

2.2 System modeling

The mathematic modeling consists of kinematic modeling and dynamic modeling.

2.2.1 Kinematic modeling

Fig. 2.11 show the coordinate for AGV's modeling where *XOY* is the global coordinate and *xCy* is the moving coordinate of AGV.

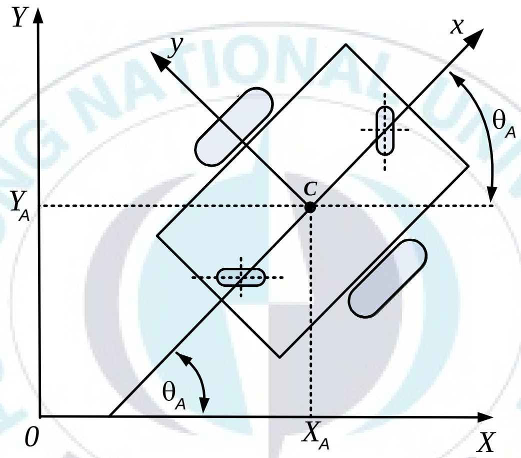


Fig. 2.11 Coordinate of AGV's modeling

To specify the position of the AGV, a point C on the AGV's chassis is chosen as its position reference point. The position of C in the global reference frame is specified by coordinates X_A and Y_A, and the angular difference between the global and local reference frames is given by θ_A.

ξ_1 is described as a position vector with position and orientation of the AGV in global coordinate:

$$\boldsymbol{\xi}_1 = \begin{bmatrix} X_A \\ Y_A \\ \theta_A \end{bmatrix} \tag{2.1}$$

Relation of AGV’s position vector $\boldsymbol{\xi}_1$ in global coordinate *XOY* and its position vector $\boldsymbol{\xi}_A$ in local coordinate *xCy* is described by:

$$\dot{\boldsymbol{\xi}}_A = \mathbf{R}(\theta_A)\dot{\boldsymbol{\xi}}_1 \tag{2.2}$$

where $\mathbf{R}(\theta_A)$ is the orthogonal rotational matrix (3x3)

$$\mathbf{R}(\theta_A) = \begin{bmatrix} \cos\theta_A & \sin\theta_A & 0 \\ -\sin\theta_A & \cos\theta_A & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

In differential drive AGV, wheels are tied together based on AGV chassis geometry. Therefore, their constraints combine to form constraints of overall motion of the AGV chassis. To understand this, firstly, constraint of each wheel is introduced. Since the AGV only use two types of wheel (fixed standard wheel and castor wheel), only these types of wheel are considered in this dissertation.

2.2.1.1 Fixed standard wheel

The fixed standard wheel has no vertical axis of rotation for steering. Its angle to the chassis is fixed, and it is limited to motion back and forth along the wheel plane and rotation around its contact point with the ground plane. Fig. 2.12 depicts a fixed standard wheel and indicates its position relative to the AGV’s local coordinate frame *xCy*. The position of the wheel is expressed in polar coordinates by distance (*l*) and angle (α). The angle of the wheel plane relative to the chassis is denoted by β, which is fixed since the fixed standard wheel is not steerable. The wheel with radius *r* can spin over time, and so its rotational angle around its horizontal axle is a function of time *t: ϕ(t).*

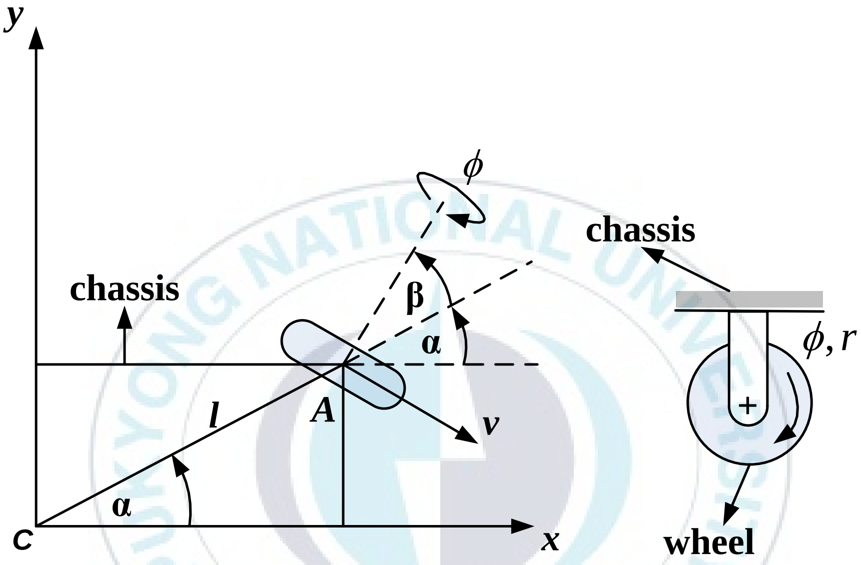


Fig. 2.12 A fixed standard wheel and its parameters

The rolling constraint for this wheel enforces that all motions along the direction of the wheel plane must be accompanied by the appropriate amount of wheel spin so that there is a pure rolling at the contact point:

$$[\sin(\alpha+\beta) \quad \{-\cos(\alpha+\beta)\} \quad (-l)\cos\beta \ ]\,\mathbf{R}(\theta_A)\dot{\boldsymbol{\xi}}_1 - r\dot{\phi} = 0 \quad (2.4)$$

The sliding constraint for this wheel enforces that the component of the wheel’s motion orthogonal to the wheel plane must be zero.

$$[\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad l\sin\beta]\,\mathbf{R}(\theta_A)\dot{\boldsymbol{\xi}}_1 = 0 \qquad (2.5)$$

2.2.1.2 Castor wheel

Castor wheels are able to steer around a vertical axis. However, unlike the standard wheel, the vertical axis of rotation in a castor wheel does not pass through the ground contact point. Fig. 2.13 depicts a castor wheel, demonstrating that formal specification of the castor wheel's position requires an additional parameter. The wheel contact point is now at position *B*, which is connected by a fixed length rigid rod *AB* to the vertical axis point *A* fixed on the chassis. Point *A* has a position specified in the AGV's reference frame as in Fig. 2.13. It is assumed that the plane of the wheel is aligned with *AB* at all times. Similarly to the steered standard wheel, the castor wheel has two parameters that vary as a function of time. ϕ (t) represents the wheel rotation angle over time. β(t) denotes the steering angle and orientation of *AB* over time.

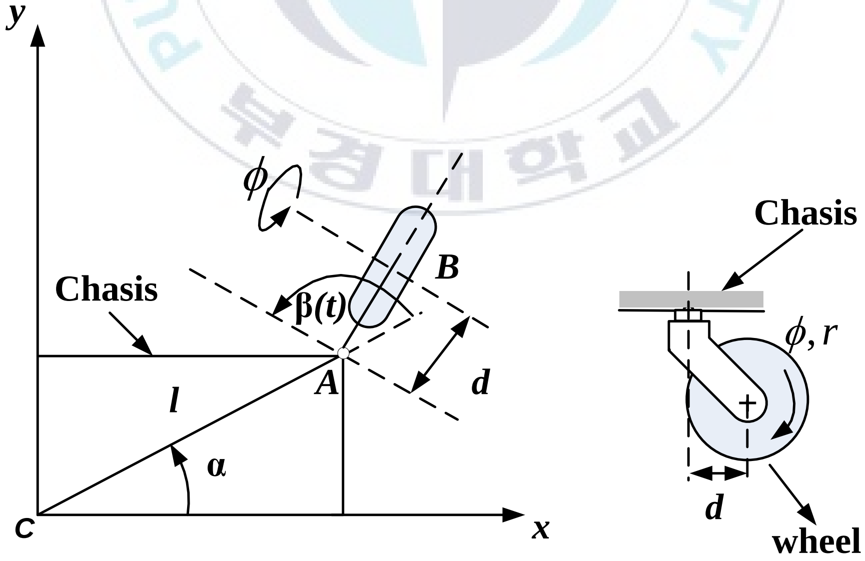


Fig. 2.13 A castor wheel and its parameters

Because the offset axis plays no role during motion that is aligned with the wheel plane for the castor wheel, the rolling constraint is:

$$[\sin(\alpha+\beta) \quad \{-\cos(\alpha+\beta)\} \quad (-l)\cos\beta \;\;]\,\mathbf{R}(\theta_A)\dot{\boldsymbol{\xi}}_1 - r\dot{\phi} = 0 \tag{2.6}$$

The sliding constraint of castor wheel is:

$$[\cos(\alpha+\beta) \quad \sin(\alpha+\beta) \quad l\sin\beta]\,\mathbf{R}(\theta_A)\dot{\boldsymbol{\xi}}_1 = 0 \tag{2.7}$$

2.2.1.3 Total kinematic model

The schematic modeling of differential drive AGV system is shown in Fig. 2.14. As shown in Fig. 2.14, the AGV position in global coordinate XOY is (X_A, Y_A) with orientation angle θ_A measured from X axis of global coordinate. The AGV moves with linear velocity V_A and angular velocity ω_A. This differential drive AGV has two driving standard wheels *W1* and *W2* with radius *r*, located in *y* axis of local coordinate AGV. Linear velocities of the left wheel and the right wheel are denoted by v_L and v_R, respectively. Given a point *C* centered between the two drive wheels, each wheel is a distance *l* from *C*. The castor wheels *W3* and *W4* are located in *x* axis of local coordinate AGV. The castor wheels impose no kinematic constraints on the AGV chassis, since it unpowered and can move freely in all direction. Therefore, only two driving standard wheels *W1* and *W2* have impact on AGV kinematics and are considered in computing the AGV’s kinematic constraints.

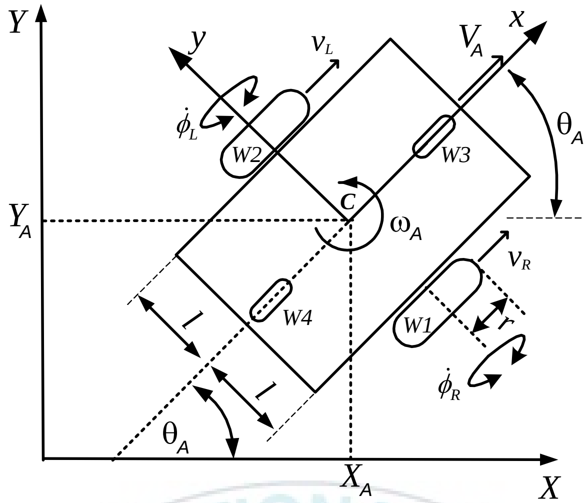


Fig. 2.14 Coordinate of differential drive AGV system

From Fig. 2.14, parameters of the right wheel W1 are $\alpha = -\pi/2$, $\beta = \pi$ and parameters of left wheel W2 are α = π/2, $\beta = 0$. Eq. (2.4) of the rolling constraints with respect to the left and right wheels can be calculated as:

$$\begin{bmatrix} \sin(-\pi/2+\pi) & \{-\cos(-\pi/2+\pi)\} & (-l)\cos\pi \\ \sin(\pi/2+0) & \{-\cos(\pi/2+0)\} & (-l)\cos 0 \end{bmatrix}\mathbf{R}(\theta_A)\dot{\boldsymbol{\xi}}_1 - \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix}\dot{\boldsymbol{\Phi}} = 0 \tag{2.8}$$

where $\dot{\boldsymbol{\Phi}} = [\dot{\phi}_R \quad \dot{\phi}_L]^T$, and *r* is radius of driving wheels.

Eq. (2.8) can be written as:

$$\begin{bmatrix} 1 & 0 & l \\ 1 & 0 & -l \end{bmatrix}\mathbf{R}(\theta_A)\dot{\boldsymbol{\xi}}_{\mathbf{1}} = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix}\dot{\boldsymbol{\Phi}} \tag{2.9}$$

For the sliding constraint, because the two driving wheels are parallel, Eq. (2.5) results in only one independent equation. Using parameters of the right wheel W1, $\alpha = -\pi / 2$, and $\beta = \pi$, the sliding constraints can be calculated as:

$$\begin{bmatrix}\cos(-\pi/2+\pi) & \sin(-\pi/2+\pi) & l\sin\pi\end{bmatrix}\mathbf{R}(\theta_A)\dot{\boldsymbol{\xi}}_1 = 0 \tag{2.10}$$

Eq. (2.10) can be written as:

$$\begin{bmatrix}0 & 1 & 0\end{bmatrix}\mathbf{R}(\theta_A)\dot{\boldsymbol{\xi}}_1 = 0 \tag{2.11}$$

The AGV’s equation related to the rolling and sliding constraints and the wheel angular velocity vector of the AGV’s wheels $\dot{\mathbf{\Phi}}$ is expressed by:

$$\begin{bmatrix}\mathbf{J}_1(\beta) \\ \mathbf{C}_1(\beta)\end{bmatrix}\mathbf{R}(\theta_A)\dot{\boldsymbol{\xi}}_1 = \begin{bmatrix}\mathbf{J}_2\dot{\mathbf{\Phi}} \\ 0\end{bmatrix} \tag{2.12}$$

where $\mathbf{J}_1(\beta) = \begin{bmatrix}1 & 0 & l \\ 1 & 0 & -l\end{bmatrix}$, $\mathbf{C}_1(\beta) = \begin{bmatrix}0 & 1 & 0\end{bmatrix}$ and $\mathbf{J}_2 = \begin{bmatrix}r & 0 \\ 0 & r\end{bmatrix}$

By combining Eqs. (2.9) and (2.12), the rolling constraints of all wheels can now be collected in a single expression:

$$\begin{bmatrix}\begin{bmatrix}1 & 0 & l \\ 1 & 0 & -l\end{bmatrix} \\ \begin{bmatrix}0 & 1 & 0\end{bmatrix}\end{bmatrix}\mathbf{R}(\theta_A)\dot{\boldsymbol{\xi}}_1 = \begin{bmatrix}J_2\dot{\mathbf{\Phi}} \\ 0\end{bmatrix} \tag{2.13}$$

From Eq. (2.13), the kinematic equation of differential drive AGV can be obtained as:

$$\begin{aligned}\dot{\boldsymbol{\xi}}_1 &= \mathbf{R}(\theta_A)^{-1}\begin{bmatrix}1 & 0 & l\\ 1 & 0 & -l\\ 0 & 1 & 0\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{J}_2\dot{\mathbf{\Phi}}\\ 0\end{bmatrix}\\ &= \mathbf{R}(\theta_A)^{-1}\begin{bmatrix}\frac{1}{2} & \frac{1}{2} & 0\\ 0 & 0 & 1\\ \frac{1}{2l} & -\frac{1}{2l} & 0\end{bmatrix}\begin{bmatrix}\mathbf{J}_2\dot{\mathbf{\Phi}}\\ 0\end{bmatrix}\end{aligned} \qquad (2.14)$$

$$\dot{\boldsymbol{\xi}}_1 = \begin{bmatrix}\dot{X}_A\\ \dot{Y}_A\\ \dot{\theta}_A\end{bmatrix} = \mathbf{R}(\theta_A)^{-1}\begin{bmatrix}\dfrac{r\dot{\phi}_R}{2} + \dfrac{r\dot{\phi}_L}{2}\\ 0\\ \dfrac{r\dot{\phi}_R}{2l} + \dfrac{-r\dot{\phi}_L}{2l}\end{bmatrix} \qquad (2.15)$$

where $\mathbf{R}(\theta_A)^{-1} = \begin{bmatrix}\cos\theta_A & -\sin\theta_A & 0\\ \sin\theta_A & \cos\theta_A & 0\\ 0 & 0 & 1\end{bmatrix}$

As shown in Eq. (2.15), velocity of the AGV in Y axis direction is always 0. Therefore, Eq. (2.15) can be expressed into

$$\dot{\boldsymbol{\xi}}_1 = \begin{bmatrix}\dot{X}_A\\ \dot{Y}_A\\ \dot{\theta}_A\end{bmatrix} = \begin{bmatrix}\frac{r}{2}\cos\theta_A & \frac{r}{2}\cos\theta_A\\ \frac{r}{2}\sin\theta_A & \frac{r}{2}\sin\theta_A\\ \dfrac{r}{b} & \dfrac{-r}{b}\end{bmatrix}\begin{bmatrix}\dot{\phi}_R\\ \dot{\phi}_L\end{bmatrix} \Leftrightarrow \dot{\boldsymbol{\xi}}_1 = \mathbf{S}(\boldsymbol{\xi}_1)\dot{\mathbf{\Phi}} \qquad (2.16)$$

The relation between AGV's velocities and left and right wheel angular velocities can be expressed as:

$$\begin{bmatrix} V_A \\ \omega_A \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2l} & \frac{-r}{2l} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \Leftrightarrow \boldsymbol{\eta}_A = \mathbf{T}\dot{\mathbf{\Phi}} \tag{2.17}$$

or vice versa,

$$\begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ \frac{r}{2l} & \frac{-r}{2l} \end{bmatrix}^{-1} \begin{bmatrix} V_A \\ \omega_A \end{bmatrix} \Leftrightarrow \dot{\mathbf{\Phi}} = \mathbf{T}^{-1}\boldsymbol{\eta}_A \tag{2.18}$$

where $\boldsymbol{\eta}_A = \begin{bmatrix} V_A & \omega_A \end{bmatrix}^T$ is a velocity vector, V_A is a linear velocity and ω_A is an angular velocity

Therefore, kinematic equation Eq. (2.16) of the AGV also can be expressed in term of its are shown as follows:

$$\dot{\boldsymbol{\xi}}_1 = \begin{bmatrix} \dot{X}_A \\ \dot{Y}_A \\ \dot{\theta}_A \end{bmatrix} = \begin{bmatrix} \cos\theta_A & 0 \\ \sin\theta_A & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_A \\ \omega_A \end{bmatrix} \Leftrightarrow \begin{aligned} \dot{\boldsymbol{\xi}}_1 &= \mathbf{J}(\boldsymbol{\xi}_1)\boldsymbol{\eta}_A \\ &= \mathbf{S}(\boldsymbol{\xi}_1)\mathbf{T}^{-1}\boldsymbol{\eta}_A \\ &= \mathbf{S}(\boldsymbol{\xi}_1)_A\dot{\mathbf{\Phi}} \end{aligned} \tag{2.19}$$

Nonholonomic constrain makes the AGV restrict moving only in direction normal to the axis of the driving tracks. This nonholonomic constrain can be written as

$$\begin{bmatrix} -\sin\theta_A & \cos\theta_A & 0 \end{bmatrix} \begin{bmatrix} \dot{X}_A \\ \dot{Y}_A \\ \dot{\theta}_A \end{bmatrix} \Leftrightarrow \mathbf{A}(\boldsymbol{\xi}_1)\dot{\boldsymbol{\xi}}_1 = 0 \tag{2.20}$$

2.2.2 Dynamic modeling

The Lagrange formula is used to derive the dynamic equations of the AGV. The system dynamics of nonholonomic AGV is as follows:

$$\mathbf{M}(\xi_1)\ddot{\xi}_1 + \mathbf{V}(\xi_1,\dot{\xi}_1)\dot{\xi}_1 = \mathbf{B}(\xi_1)\boldsymbol{\tau} - \mathbf{A}^T(\xi_1)\boldsymbol{\lambda} \tag{2.21}$$

where $\mathbf{M}(\xi_1) \in \mathbf{R}^{n\times n}$ is a symmetric positive definite inertia matrix; $\mathbf{V}(\xi_1,\dot{\xi}_1) \in \mathbf{R}^{n\times n}$ is a centripetal and Coriolis matrix; $\mathbf{B}(\xi_1) \in \mathbf{R}^{n\times r}$ is an input transformation matrix; $\mathbf{A}\left(\xi_1\right) \in \mathbf{R}^{n\times m}$ is a matrix of nonholonomic constraints; $\boldsymbol{\tau} = \left[\tau_R \quad \tau_L\right]^T \in \mathbf{R}^r$ is an input torque vector consisting of torques exerted on right and left wheels; and $\boldsymbol{\lambda} \in \mathbf{R}^m$ a constraint force vector.

Differentiating Eq. (2.16), substituting this result in Eq. (2.21) and multiplying by $\mathbf{S}^T$, the constraint term $\mathbf{A}^T(\xi_1)\boldsymbol{\lambda}$ is eliminated. Dynamics in platform system of the nonholonomic AGV with the constraint in Eq. (2.20) is as follows:

$$\mathbf{S^T M S \ddot{\Phi} + S^T (M\dot{S} + VS)\dot{\Phi} = S^T B}\boldsymbol{\tau} \tag{2.22}$$

Multiplying by $(\mathbf{S}^T\mathbf{B})^{-1}$, Eq. (2.22) can be rewritten as

$$\mathbf{\bar{M}}(\xi_1)\ddot{\mathbf{\Phi}} + \mathbf{\bar{V}}(\xi_1,\dot{\xi}_1)\dot{\mathbf{\Phi}} = \boldsymbol{\tau} \tag{2.23}$$

$$\mathbf{\bar{M}}(\xi_1) = (\mathbf{S}^T\mathbf{B})^{-1}\mathbf{S}^T\mathbf{MS} \in \mathbf{R}^{r\times(n-m)}$$

$$\mathbf{\bar{V}}(\xi_1) = (\mathbf{S}^T\mathbf{B})^{-1}\mathbf{S}^T(\mathbf{M\dot{S}} + \mathbf{VS}) \in \mathbf{R}^{r\times(n-m)}$$

where $\dot{\mathbf{\Phi}}, \ddot{\mathbf{\Phi}} \in \mathbf{R}^{(n-m)}$

$$\bar{\mathbf{M}} = \begin{bmatrix} \frac{r^2}{4b^2}(mb^2+I)+I_w & \frac{r^2}{4b^2}(mb^2-I) \\ \frac{r^2}{4b^2}(mb^2-I) & \frac{r^2}{4b^2}(mb^2+I)+I_w \end{bmatrix}$$

$$I = m_c d^2 + 2m_w b^2 + I_c + 2I_m$$

$$\bar{\mathbf{V}} = \begin{bmatrix} 0 & \frac{r^2}{2b} m_c d\dot{\theta}_A \\ -\frac{r^2}{2b} m_c d\dot{\theta}_A & 0 \end{bmatrix}$$

where $m = m_c + 2m_w$ is total mass of the AGV, m_c is mass of the body without driving wheels, m_w is mass of each wheel, I_m is moment of inertia of each motor, I_w is moment of inertia of each wheel, I_c is moment of inertia of the body, and d is distance between geometric center and mass center.

Chapter 3: Path Planning

This chapter proposes a new coverage path planning algorithm for AGV system. The purpose of the coverage path planning is to make the AGV visit all reachable area in a given environment efficiently. The input of the proposed algorithm is an occupancy grid map.

3.1 Problem statements and assumptions

The problem statements considered for operating an AGV in this dissertation are as follows:

1. The AGV must move through all the points in the target area to be covered completely and fill the region without overlapping paths.
2. The coverage time should be minimized to increase the efficiency and reduce operational cost.
3. To prevent the inertial moment when the AGV turns, the speed of AGV has to be reduced in turning motion. Therefore, to reduce the coverage time, the number of turning should be minimized. Simple motion trajectories such as straight lines or circles are preferred.
4. The AGV should come back to the start position after covering all area.

There are some assumptions used for a proposed path planning algorithm used in this dissertation as follows:

1. The input of the proposed algorithm is an occupancy grid map obtained from the previous observation. The information about a target area is enough. Therefore, the environment is known.
2. The coverable area is assumed as a free space without unknown obstacle. In this area, the coverage path can be planned in any direction without restriction.
3. Exact positioning and navigation is available. Therefore, the AGV can track the desired path easily.

3.2 Path planning algorithm

A proposed coverage path planning method for this dissertation consists of three steps. The first step is to apply Morse cell decomposition method for dividing the whole area of the work space into smaller areas based on the obstacle position. The second step is to find the minimum turning number based on the minimal sum of altitude algorithm. The final step is to plan a shortest coverage path using a multi-spanning tree method.

3.2.1 Occupancy grid map

In order to perform a motion planning, it is usually necessary to define some representation of the environment. One common technique for map representation is an occupancy grid map. The occupancy grid map represents the environment as a block of cells. The area is occupied so that the AGV cannot pass through it, or it is unoccupied so that the AGV can traverse it.

Occupancy grid algorithms represent the map $\mathbf{D}$ as a fine-grained grid over the continuous space of locations in the environment. The most common type of occupancy grid maps are 2d maps that describe a slice of the 3d world. If $\mathbf{D}_i$ denote the grid cell with index i, then the notation $p(\mathbf{D}_i)$ represents the probability that cell i is occupied. In this dissertation, the probability of each grid is chosen as follows:

$$\begin{cases} \text{Cell is occupied} & p(\mathbf{D}_i) = 1 \\ \text{Cell is not occupied} & p(\mathbf{D}_i) = 0 \end{cases} \tag{3.1}$$

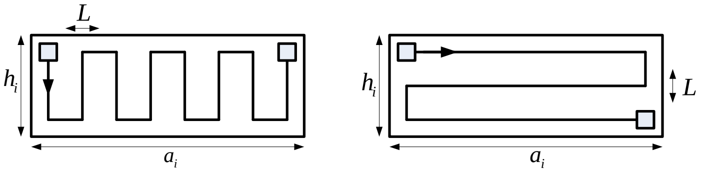
In this dissertation, the environment is assumed to be static. Therefore, the map is always constant. The example of occupancy grid map is shown in Fig. 3.1.

0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	0	0	0	0	1	0
0	0	0	0	0	0	0	0

Fig. 3.1 Occupancy grid map

3.2.2 Minimal sum of altitude cell decomposition

The number of turns required to cover the region is a main factor for saving time cost. In the template-based path planner that the AGV moved with the same sweep direction for every cell, it would generate too many unnecessary turns leading to low efficiency as in Fig. 3.2a [83]. However, if the sweep direction is similar with the shortest side of cell as shown in Fig. 3.2b, the minimum turning number can be obtained. Therefore, the turning number is related with the sweeping direction.



a. Inefficient turns b. Efficient turns

Fig. 3.2 Relation between turning number and sweep direction

To minimize the turning number, a minimal sum of altitude (MSA) algorithm as in [84-85] was used. In MSA, the optimal decomposition was determined by the sweeping direction that minimized the sum of subregion altitude consisting of the diameter functions of hole and perimeter depending on the rotating angle θ of the polygonal as an obstacle. Altitude is defined as the minimum height of cell when the cell is rotated by θ degree. As shown in Fig. 3.2b, the turning number is proportional to the altitude of the cell. Therefore, to minimize the turning number, the sweep direction should be synchronized with the altitude of the cell.

For rectangular shape cell, the altitude $A(i)$ of the i^{th} cell can be calculated as:

$$A(i) = \min(h_i, a_i) \tag{3.2}$$

where h_i and a_i is the length and the width of the i^{th} cell and $i = (1, 2, 3, \cdots)$ is the cell number.

The sweep direction $Dir(i)$ of the i^{th} cell can be obtained as follows:

$$Dir(i) = \begin{cases} \text{Vertical} & \text{if} \quad h_i > a_i \\ \text{Horizontal} & \text{if} \quad h_i < a_i \end{cases} \tag{3.3}$$

The sum of altitude $S(j)$ of the j^{th} way to split the graph is defined as:

$$S(j)=\sum_{1}^{n}A(i) \quad \text{for } i=1,2,3,\cdots \tag{3.4}$$

where n is the total number of cell and $j=(1,2,3,\cdots)$ is the number of way to split the graph $T(x,y)$ into subgraphs.

The main idea of the MSA is to find the minimum value of sum of altitude $S(j)$. The minimal sum of altitude MSA is defined as

$$MSA=\min(S(1),S(2),S(3),\cdots,S(k)) \tag{3.5}$$

where k is the total number of the possible way to split the graph $\mathbf{D}(x,y)$ into subgraphs.

Finally, the total turning number to cover all area N is directly related to the MSA. Therefore, the MSA may reduce N. The value of N is depicted as follows:

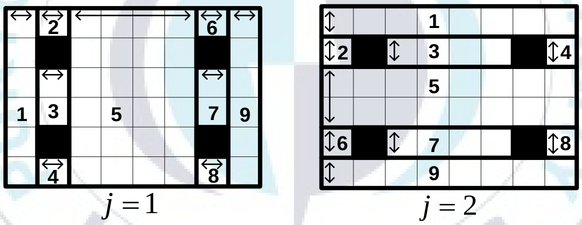
$$N=\frac{1}{L}MSA, \text{ for } i=1,2,3,\cdots \tag{3.6}$$

where L denotes the AGV width.

In [84], the cell decomposition could be obtained by dynamic programming formulation. This approach is difficult to apply in complex environment. To overcome this problem, a new approach is introduced in this dissertation. The optimal cell decomposition is obtained by combining the result of vertical cell decomposition and horizontal decomposition.

To do this, firstly, Morse cell decomposition for discrete map representation as in [32] is applied to the occupancy grid map in Fig. 3.1 to divide the whole area of the work space into smaller areas called cells.

In Morse cell decomposition, it is assumed that the work environment has simple structure and is in non-polygonal space. The cell decomposition can be obtained by sweeping the line horizontally from the left to the right to find the critical points of the polygon obstacles and then drawing lines at only those vertices which can be extended vertically in both upward and downward directions. The result is shown in Fig. 3.3a. If the sweeping line is in vertical direction, the cell decomposition is shown in Fig. 3.3b.



a. Horizontal b. Vertical

Fig. 3.3 Horizontal and vertical cell decomposition

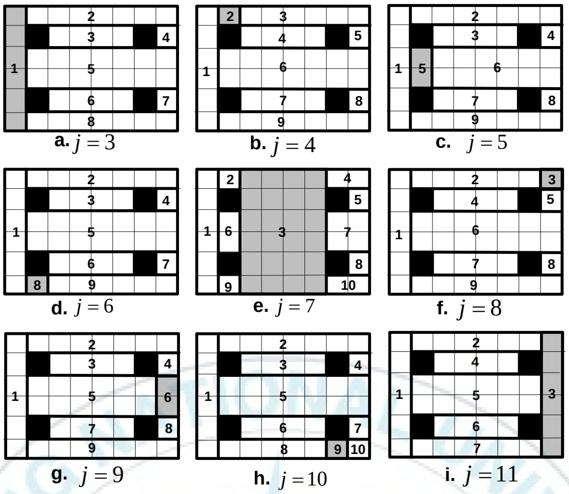
The next step is to calculate the $S(j)$ of cell decomposition using Eq. (3.4). It can be seen that the sum of altitude for horizontal cell decomposition $S(1)$ is 12 seen from Fig. 3.3a and the sum of altitude for vertical cell decomposition $S(2)$ is 10 from Fig. 3.3b. Using Eq. (3.5), it can be calculated that the value of MSA is 10 obtained from vertical cell decomposition method.

To obtain the better MSA value, a proposed cell decomposition in this dissertation is to combine both vertical cell decomposition and horizontal cell decomposition. The idea is by adding cells from cell decomposition with higher sum of altitude to cell decomposition with lower sum of altitude. In the case of Fig. 3.3, cells from 1 to 9 obtained from Fig. 3.3a are exchanged into Fig. 3.3b in turn as shown in Fig. 3.4. There are 9 ways to split the graph $T(x, y)$ into subgraphs as shown in Fig. 3.4.

Firstly, add one cell from cell decomposition with higher sum of altitude to cell decomposition with lower sum of altitude. Secondly, evaluate the current sum of altitude $S(j)$ using Eq. (3.4). If the sum of altitude of the current combination $S(j)$ is smaller or equal to that of the previous combination $S(j-1)$, the current combination is saved and will be used for next calculation. Otherwise, the current combination is discarded. Thirdly, repeat this process until all cells are added.

As shown in Fig. 3.4a, the sum of altitude $S(3)$ after adding the cell number 1 from Fig. 3.3a to the Fig. 3.3b is 9. Because $S(3)$ is less than the $S(2)$ that is 10, the current cell decomposition is saved. Next step is adding cell number 2 from Fig. 3.3a to Fig. 3.3b as shown in Fig. 3.4b. The current sum of altitude $S(4)$ becomes 10. Since the $S(4)$ is higher than $S(3)$, the current combination is discarded. This process is repeated until all cells from Fig. 3.3a are added to Fig. 3.3b. The results of this process are shown in Fig. 3.4. The number of cell and the sum of altitude of each step are shown in

Table 3.1. Table 3.1 shows that the *MSA* is 8 obtained at $j = 11$ as shown in Fig. 3.4i.



a. $j = 3$ b. $j = 4$ c. $j = 5$

d. $j = 6$ e. $j = 7$ f. $j = 8$

g. $j = 9$ h. $j = 10$ i. $j = 11$

Fig. 3.4 Proposed method combining vertical and horizontal cell decompositions

Table 3.1 Number of cells and the sum of altitude of each step obtained from proposed method

Way to split the graph j	Total number of cell n	Sum of altitude $S(j)$
$j = 1$	9	12
$j = 2$	8	10
$j = 3$	8	9
$j = 4$	9	10
$j = 5$	9	10
$j = 6$	9	10
$j = 7$	10	14
$j = 8$	9	10
$j = 9$	9	10
$j = 10$	10	11
$j = 11$	7	8

3.2.3 Multi-spanning tree

After cell decomposition, the next step is to generate the total trajectory based on the spanning tree algorithm. In this dissertation, instead of using the template-based path planner in [32], modification of the conventional spanning tree algorithm in [40-41] is used. In the conventional spanning tree algorithm, the conventional Prim algorithm is applied one time to build the minimum spanning tree of a given area. In the proposed multi-spanning tree algorithm, the conventional Prim algorithm is applied twice, firstly to each cell and secondly to all area.

In [40-41], the first step is to cover the interior of a square shape free area as shown in Fig. 3.5a with a square shape of tiles to construct subgraph as shown in Fig. 3.5b. The width of the square shape of tile is $2L$. The next step is to construct a roadmap G by placing a vertex in the center of each square and by defining an edge that connects the vertices as shown in Fig. 3.5c.

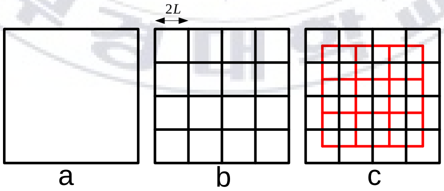


Fig. 3.5 Construction of a roadmap G

The next step is to compute a spanning tree of G. This is a connected subgraph that has no cycles and touches every vertex of G. In this method, to calculate the minimum spanning-tree, Prim’s algorithm [42] is employed.

Prim's algorithm is a greedy algorithm that finds a minimum spanning tree for a connected weighted undirected graph. This means that it finds a subset of the edges that form a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. The input of this algorithm is a weighted graph with vertices V and edges E obtained from the roadmap G.

The calculation of minimum spanning tree based on Prim algorithm can be explained as follows:

1. Initialize a tree with a single vertex chosen arbitrarily from the graph.
2. Grow the tree by one edge of the edges that connect the tree to vertices not yet in the tree, find the minimum-weight edge, and transfer it to the tree.
3. Repeat step 2 (until all vertices are in the tree).

The example of Prim algorithm is shown in Fig. 3.6. Prim's algorithm starts at vertex A. Firstly, AD is chosen since the weight AD is smaller than weight AB as shown in Fig. 3.6a. In the second step at Fig. 3.6b, BD is chosen to add to the tree instead of AB arbitrarily because both have weight 2. Afterwards, AB is excluded because it is between two nodes that are already in the tree as shown in Fig. 3.6c. Finally, CD is connected since there is no other connection as shown in Fig. 3.6d.

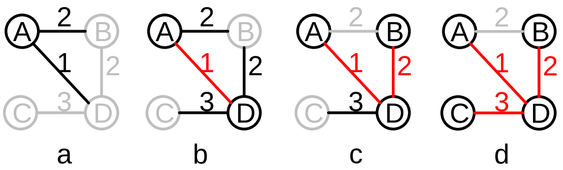


Fig. 3.6 Example of Prim algorithm

Once the spanning tree is made as shown in Fig. 3.7a, the AGV path is obtained by starting at a point near the spanning tree and following along its perimeter. Next step is dividing a single vertex into four smaller vertices and then forms the corresponding roadmap as shown in Fig. 3.7b. Part of the roadmap corresponds to the spanning tree including a loop path that surrounds the spanning tree as shown in Fig. 3.7c. This path visits the centers of the new squares. The resulting path is shown in Fig. 3.7d. The method yields an optimal route.

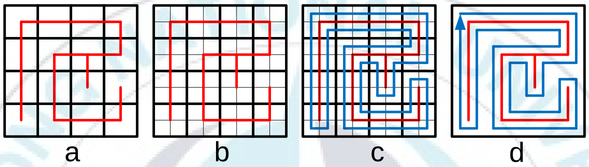


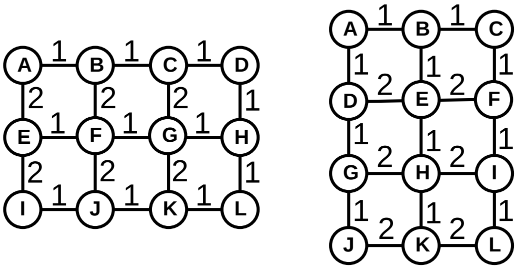
Fig. 3.7 Spanning three based path planning

The above algorithm guarantees that the AGV can move through all the points in the target area and cover it completely, fill the region without overlapping paths and come back to the start position after covering all area. However, it does not guarantee that the number of turning is minimized and the trajectories are simple (e.g., straight lines or circles).

To overcome these problems, this dissertation proposes following modifications. The spanning tree algorithm is applied twice to the result of minimal sum of altitude cell decomposition to guarantee that the number of turning is minimized. Firstly, a spanning tree algorithm is applied to connect all vertices inside each cell. These connections called are as leaves. Secondly, the spanning tree is applied to connect all cells obtained from MSA cell decomposition. These connections are called as branches. Finally, these leaves and branches are combined to construct a spanning tree. Therefore, this proposed algorithm is called as a multi-spanning tree.

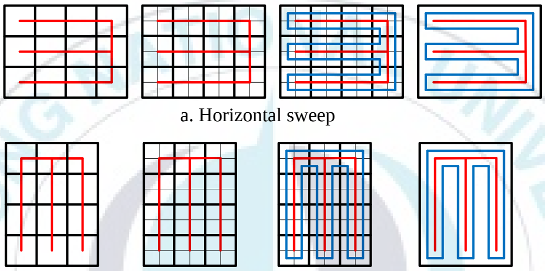
3.2.3.1 Spanning tree for constructing leaves

The first spanning tree is applied to each cell. To guarantee that the turning number in trajectories generated by Prim algorithm are minimized, the modified weight are used to affect the shape of the spanning tree and hence the covering-path geometry. The edge weighting can generate covering paths that tend to scan the work-area along a particular coordinate direction. In this dissertation, the edge weights are modified to follow the sweep direction obtained from Eq. (3.3) as shown in Fig. 3.8. The edges parallel with the sweep direction have lower value than the edges perpendicular with the sweep direction. Furthermore, the edge of the end of sweep direction such as DH and HL in Fig. 3.8a can be modified to make sure the AGV turn in a desired position. In this dissertation, if the sweep line is horizontal as shown in Fig. 3.8a, the edge value on the right side of cell is small. On the other hand, if the sweep line is vertical as shown in Fig. 3.8b, the edge value on the right top of cell is small. The results of spanning tree based on Prim algorithm can be seen in Fig. 3.9.



a . Horizontal sweep b. Horizontal sweep

Fig. 3.8 Modified weight



a. Horizontal sweep

b. Vertical sweep

Fig. 3.9 Proposed spanning three based path planning using Prim algorithm

Applying the above spanning tree algorithm for each cell to minimal sum of altitude cell decomposition in Fig. 3.4i results in the following leaves result.

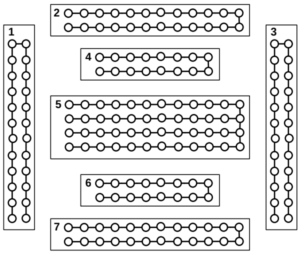
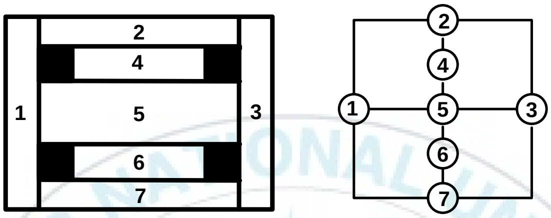


Fig. 3.10 Leaves

3.2.3.2 Spanning tree for constructing branches

The input for the second spanning tree is an adjacency graph to represent a connection relationship and distance relationship among cells. The edges value is obtained from the Euclidean distance between cells. The example of adjacency graph obtained from Fig. 3.11a is shown in Fig. 3.11b.



a. Cell decomposition b. Adjacency graph

Fig. 3.11 Cell decomposition and adjacency graph

Using Prim algorithm, a minimum spanning tree as shown in Fig. 3.12 can be obtained. The weight between vertices is chosen based on the Euclidean distance between centers of the vertices as shown in Fig. 3.12a. Prim's algorithm starts at vertex 1 as shown in Fig. 3.12b. Firstly, edge 1-5 is chosen since the weight of edge 1-5 is smaller than the weights of edge 1-2 and edge 1-7 as shown in Fig. 3.12c. In the second step of Fig. 3.12d, edge 5-4 is chosen to add to the tree instead of edge 5-6 arbitrarily because both edges have weight 1.5. Afterwards, edge 4-2 is chosen because edge 4-2 is edge with the smallest weight among all edges attached to the tree as shown in Fig. 3.12e. Afterwards, edge 5-6 is connected since it is edge with smallest weight among all edges attached to the tree as shown in Fig. 3.12f. Next step, edge 6-7 is chosen because it is edge with smallest weight among all edges attached to the tree as shown in Fig. 3.12g.

Finally, edge 5-3 is connected since it is the smallest weight among edge 2-3, edge 5-3 and edge 7-3 as shown in Fig. 3.12h. Since all vertices are connected to the tree, this algorithm is done. The final minimum spanning tree called branches is obtained as in Fig. 3.12i.

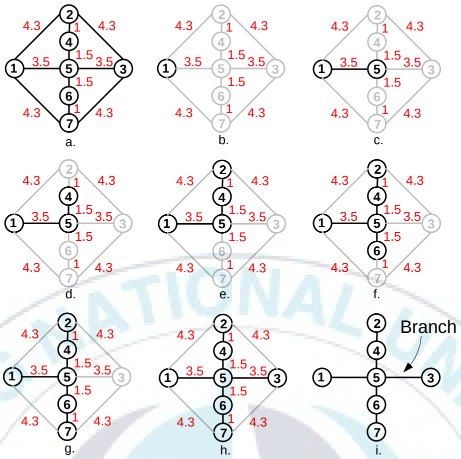


Fig. 3.12 Branches obtained from the proposed Prim algorithm

3.2.3.3 Combining the leaves and the branches

A final result of the proposed multi-spanning tree algorithm can be obtained by combining the leaves in Fig. 3.10 to the branches in Fig. 3.12i. The result is combination of the trees as shown in Fig. 3.13.

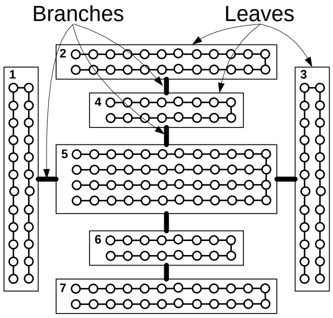


Fig. 3.13 Combination of the leaves and the branches

In the graph, the combinations of the leaves and branches are shown in Fig. 3.14. The positions of the branches connecting between cells are chosen to minimize the turning number. In this dissertation, the branches are positioned at the top right of the cells.

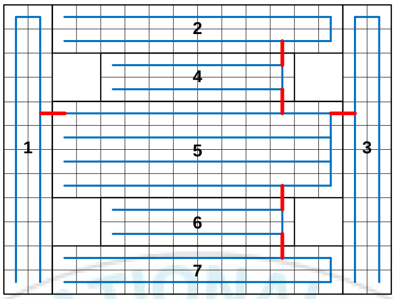
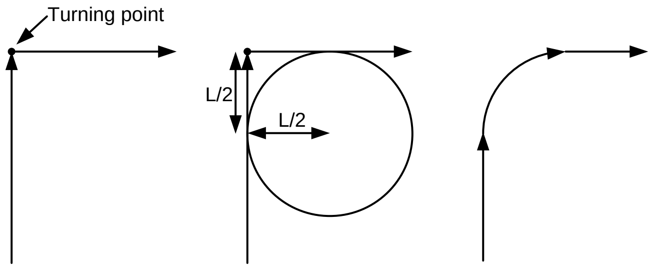


Fig. 3.14 Final result of the proposed multi-spanning tree algorithm

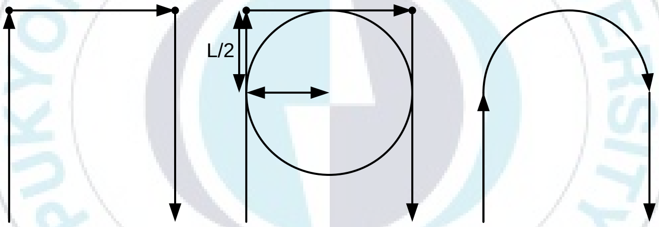
3.2.3.4 Smoothing the turning point

Since a generated path obtained from the conventional spanning tree is sharp, when the AGV reaches the turning point, the AGV has to completely stop, doing turning maneuver, and start moving again to track the generated path well. This condition is wasting the time and energy. To solve this problem, the turning path has to be smoothed to reduce the turning time. To do this, the following are done. Fig. 3.15 shows the smoothing method for 90 degree turning. By introducing a circle with diameter $L/2$ as shown in Fig. 3.15b, the smoothing algorithm must be done as follows. At distance $L/2$ from the turning point, the AGV reduces its linear velocity to half of the nominal linear velocity. Then the AGV starts following the circle until the AGV turn 90 degree. The result of smoothing 90 degree is shown in Fig. 3.15c. Fig. 3.16 shows the smoothing result for 180 degree turning. Instead of turning until 90 degree, the AGV continues following the circle until the AGV turns completely 180 degree. The velocity at turning condition is half of the nominal linear velocity in straight condition.



a. Initial b. Smoothing c. Final

Fig. 3.15 Smoothing 90 degree turning



a. Initial b. Smoothing c. Final

Fig. 3.16 Smoothing 180 degree turning

3.3 Simulation results

To verify the effectiveness of our proposed method, several simulations are done. The proposed method is compared with to the vertical and horizontal cell decompositions, and conventional spanning tree algorithm.

3.3.1 Occupancy grid map

Fig. 3.17 shows the occupancy grid map obtained from observation. The occupancy grid map is an array of probabilities that an obstacle is located in a certain location. That is, it represents the environment as a block of cells. Each cell holds a probability value that the cell is occupied. The black solid grid represents the obstacle in the environment. The white grid represents the free space. The resolution of the occupancy grid map is 0.4 m. The total size of the occupancy grid map for this dissertation is 4.8m $\times$4m.

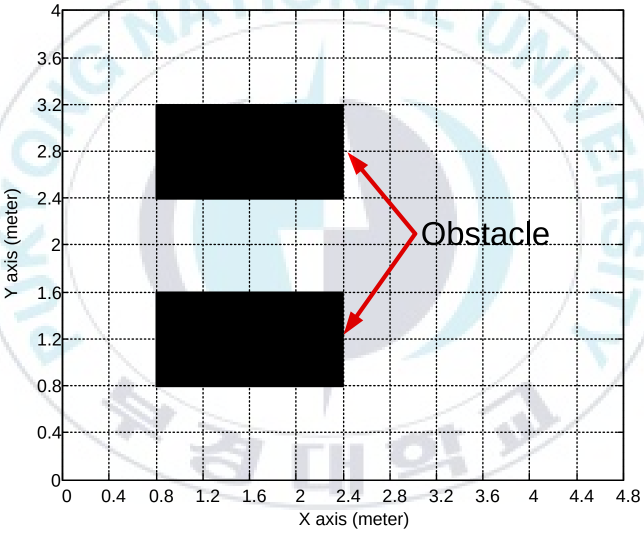


Fig. 3.17 Occupancy grid map

3.3.2 Minimal sum of altitude cell decomposition

Based on the occupancy grid map as in Fig. 3.17, the work environment is divided into several cells. Figs. 3.18– 3.20 show the comparisons among vertical cell decomposition, horizontal cell decomposition, and minimal sum of altitude cell decomposition. Fig. 3.18 shows the vertical cell decomposition obtained using Morse cell decomposition in vertical sweeping direction. It shows that the environment divided into 7 cells. The cell numbering starts from the bottom to the top representing the sweeping direction.

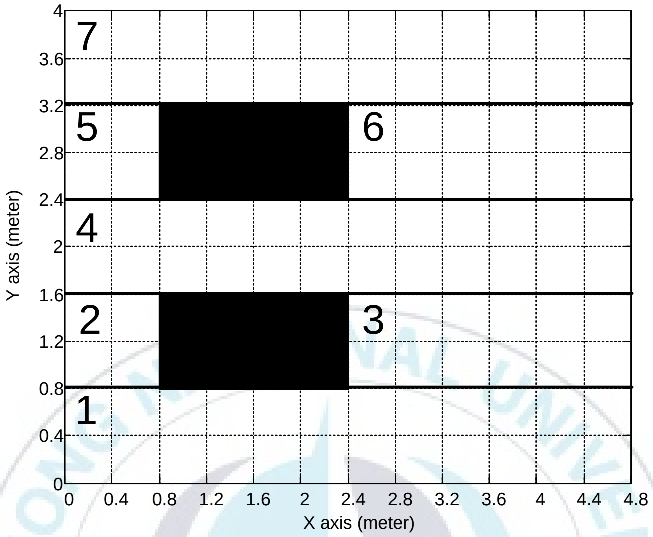


Fig. 3.18 Vertical cell decomposition

Fig. 3.19 shows the horizontal cell decomposition obtained using Morse cell decomposition in horizontal sweeping direction. It shows that the environment divided into 5 cells. The cell numbering starts from the left to the right representing the sweeping direction.

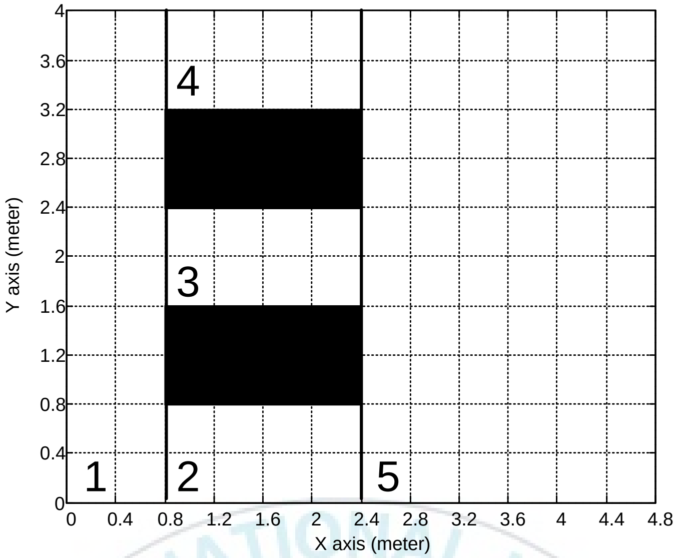


Fig. 3.19 Horizontal cell decomposition

Fig. 3.20 shows the cell decomposition obtained from the proposed algorithm such in section 3.2.2. It shows that the environment divided into 6 cells.

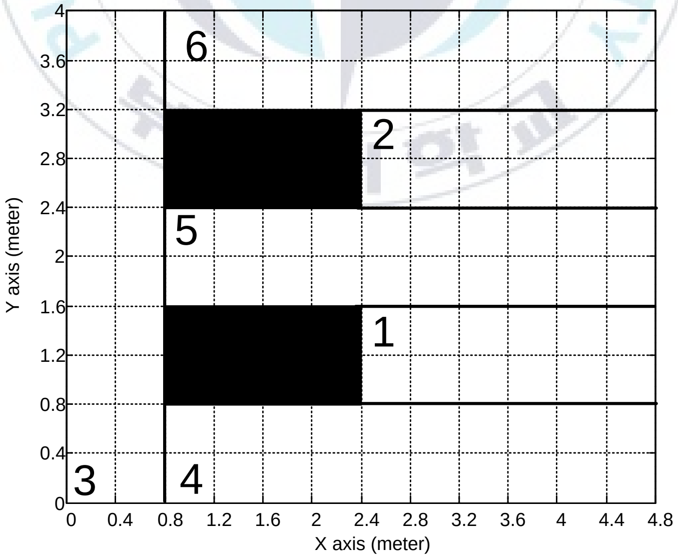


Fig. 3.20 Minimal sum of altitude cell decomposition

From the simulation result, the sum of altitude for the vertical cell decomposition is 9m. The sum of altitude of the horizontal cell decomposition is 7m. The sum of altitude of the proposed algorithm is 6m. Compared with vertical and horizontal cell decompositions, the proposed algorithm has less sum of altitude.

3.3.3 Multi-spanning tree

Figs. 3.21 - 3.22 show comparison between the conventional spanning tree and the proposed spanning tree. Fig. 3.21 shows the minimum spanning tree obtained from the conventional spanning tree algorithm. Fig. 3.22 shows the minimum spanning tree obtained from the proposed algorithm. It shoes that the branches and leaves are following the minimal sum of altitude cell decomposition.

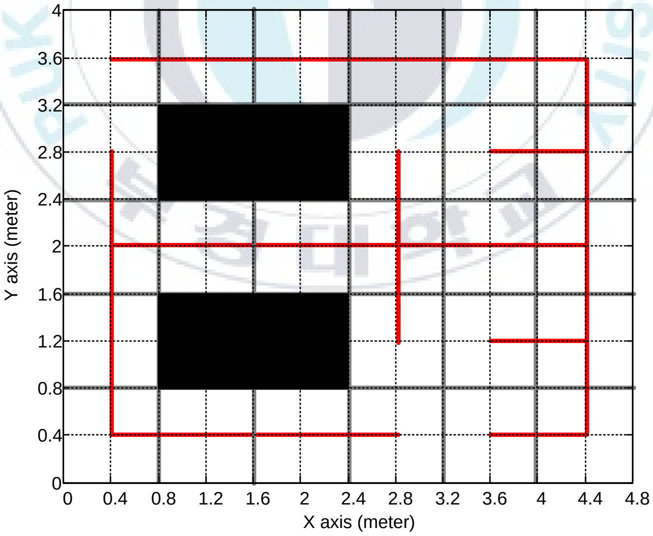


Fig. 3.21 Conventional spanning tree

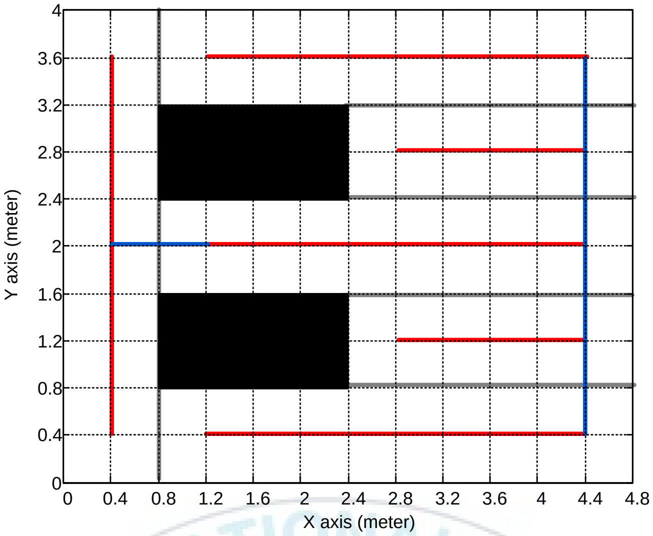


Fig. 3.22 Proposed multi-spanning tree algorithm

3.3.4 Total path generating

Figs. 3.23 – 3.24 show the optimal coverage paths generated by the vertical cell decomposition and horizontal cell decomposition, respectively. Fig. 3.23 shows the coverage path obtained from vertical cell decomposition. The coverage sequence is 7-5-4-2-1-3-6. Fig. 3.23 shows that there exist two overlapped paths in the generated path. Fig. 3.24 shows the optimal coverage path generated by the horizontal cell decomposition. The coverage sequence is 1-2-5-3-4. Fig. 3.24 shows that there exist four overlapped paths in the generated path.

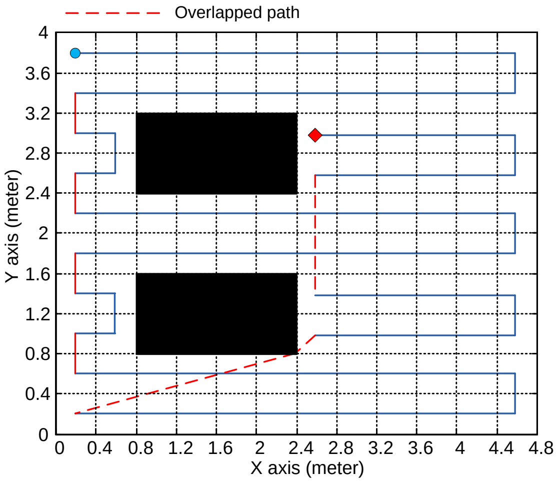


Fig. 3.23 Path generated by vertical cell decomposition

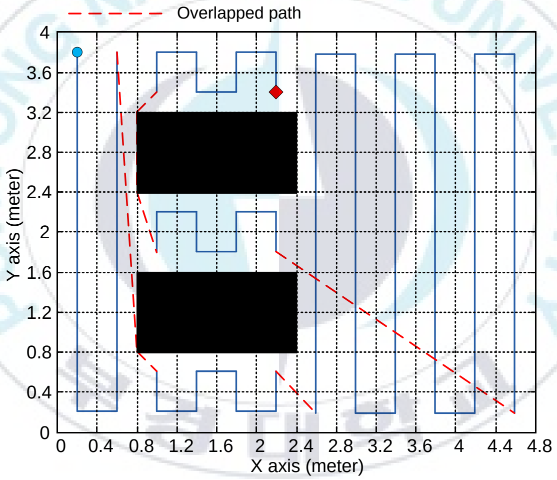


Fig. 3.24 Path generated by horizontal cell decomposition

Fig. 3.25 shows the path generated by conventional spanning tree. Simulation result shows that there is no overlapped path and the path comeback to the start position. Fig. 3.26 shows path generated by the proposed algorithm. Similarly, in this simulation result, there is no overlapped path in the generated path. However, the proposed algorithm has less turning number in the generated path than the conventional spanning tree algorithm.

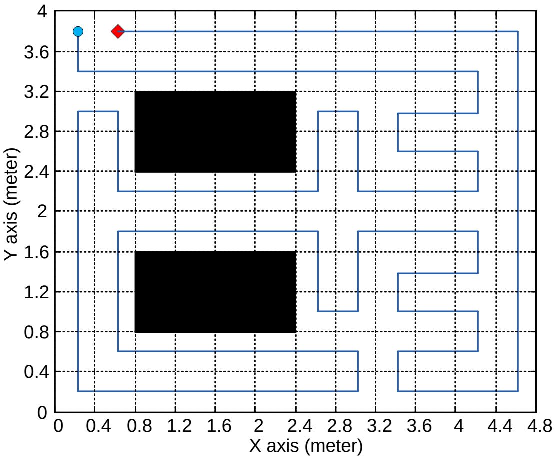


Fig. 3.25 Path generated by the conventional spanning tree

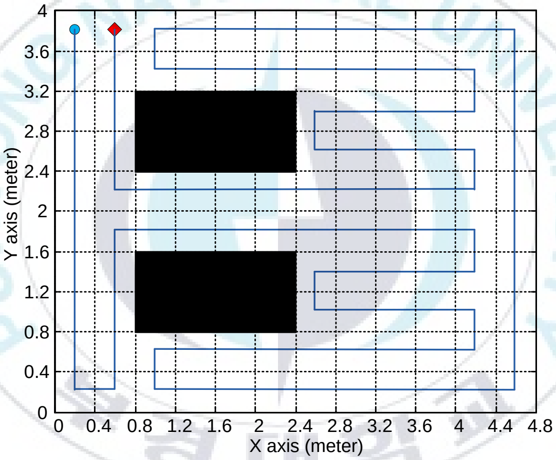


Fig. 3.26 Path generated by the proposed algorithm

3.3.5 Total path after smoothing

After applying the smoothing algorithm to the paths generated in section 3.2.3.4, the smoothed paths of Figs. 3.23-3.26 are shown in Figs. 3.27-3.30.

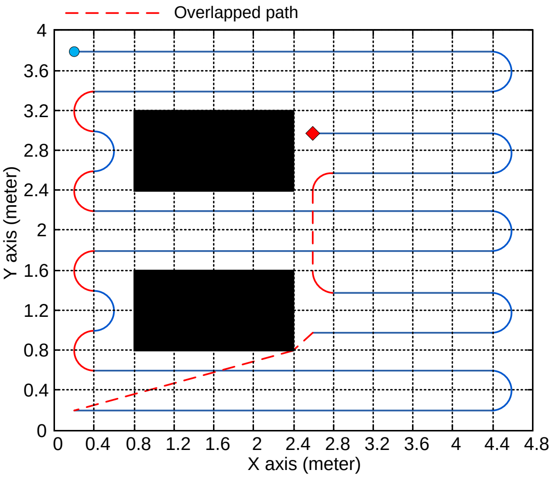


Fig. 3.27 Smooth path generated by vertical cell decomposition

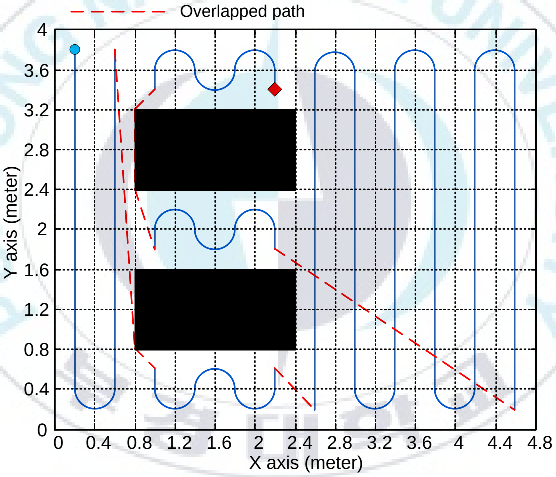


Fig. 3.28 Smooth path generated by horizontal cell decomposition

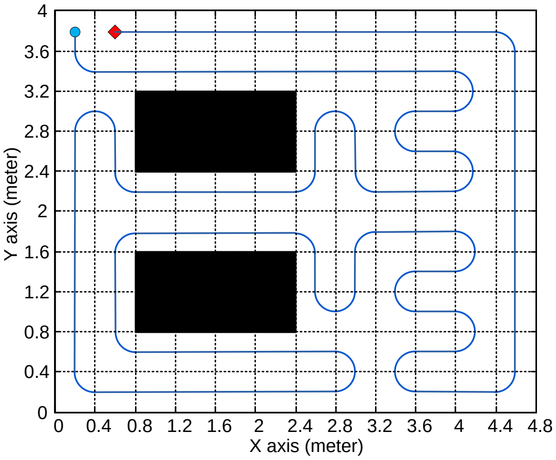


Fig. 3.29 Smooth path generated by conventional spanning tree

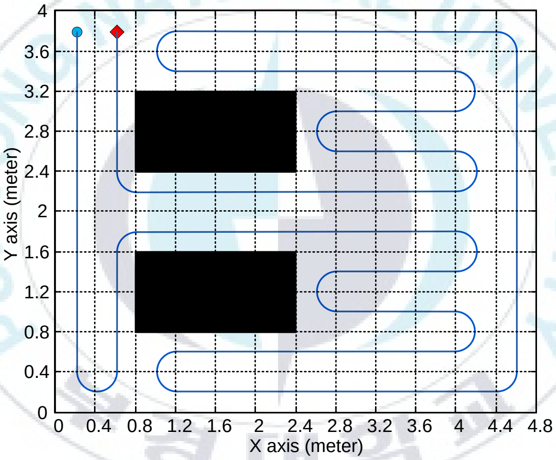


Fig. 3.30 Smooth path generated by the proposed algorithm

Table 3.2 shows the comparisons of the path planning algorithms mention above. Table 3.2 shows that the proposed algorithm successfully generates the shortest path without overlap, has the minimum turning number, make the AGV returns to start position and has fast coverage time compared to the vertical cell decomposition, the horizontal cell decomposition, and the conventional spanning tree algorithm as shown in Figs. 3.27 – 3.30.

Table 3.2 Comparison path planning algorithms after smoothing

Items	Vertical	Horizontal	Spanning tree	Proposed
Overlapping path	7,28 m	8.45m	0 m	0 m
Turning number	11 turn	15 turn	11 turn	9 turn
Comeback to start	No	No	Yes	Yes
Coverage time	280s	302s	225s	206s
Total Length	108.52m	109.59m	103.3m	101.4m

3.4 Summary

This chapter proposed a path planning algorithm for AGV to coverage a defined area based on a multi-spanning tree cell decomposition method. As the input of the proposed algorithm, a given occupancy grid map in the work environment was used. Based on Morse cell decomposition, the occupancy grid map was divided into cells in the vertical and horizontal direction. To minimize the turning number, a minimal sum of altitude method was applied. A spanning tree algorithm for each cell was adapted to generate the trajectory of each cell called leaves. The spanning tree algorithm for connecting cells was applied to connect the trajectory between cells called branches. The generated path was smoothed by the smoothing algorithm. The simulation result showed that the proposed algorithm generated the shorter coverage path without overlapping path, had less turning number and had faster coverage time than the conventional spanning tree, vertical and horizontal cell decomposition methods. For the future plan, since the grid resolution of grid map was similar with the AGV width, the existence of small obstacle could not be confirmed. Therefore, an obstacle avoidance algorithm should be employed in practical application.

Chapter 4: Controller Design

The purpose of this section is to design an adaptive trajectory tracking controller for AGV to track the desired reference trajectory $\boldsymbol{\xi}_{\mathbf{r}} = [X_r(t), Y_r(t), \theta_r(t)]^T$ generated in chapter 3 with reference velocities $\boldsymbol{\eta}_r = \left[V_r(t) \quad \omega_r(t)\right]^T$. $(X_r(t), Y_r(t))$ is reference position of AGV in global coordinate, θ_r is AGV reference orientation, $V_r(t)$ is reference linear velocity and $\omega_r(t)$ is reference angular velocity. If the desired Cartesian trajectory $(X_r(t), Y_r(t))$ is differentiable, the state variable $\theta_r(t)$, $V_r(t)$ and $\omega_r(t)$ can be calculated as:

$$\theta_r(t) = \arctan 2(\dot{Y}_r(t), \dot{X}_r(t)) \tag{4.1}$$

$$V_r(t) = \pm\sqrt{\dot{X}_r^2(t) + \dot{Y}_r^2(t)} \tag{4.2}$$

$$\omega_r(t) = \frac{\dot{X}_r(t)\ddot{Y}_r^2(t) - \dot{Y}_r^2(t)\ddot{X}_r(t)}{\dot{X}_r^2(t) + \dot{Y}_r^2(t)} \tag{4.3}$$

In this chapter, two controllers are proposed: adaptive backstepping controller and internal model principle-based controller.

4.1 Adaptive backstepping control design

A differential drive wheel type is frequently adopted to accomplish different tasks due to their good mobility and simple configuration. However, it is in general difficult to control due to slip phenomena caused by the unknown wheel radii due to uneven load distributions, uneven wear of the wheel or manufacturing imperfection which is an important factor that must be taken into account during the control design. In this dissertation, the slip condition and dynamic of AGV are considered.

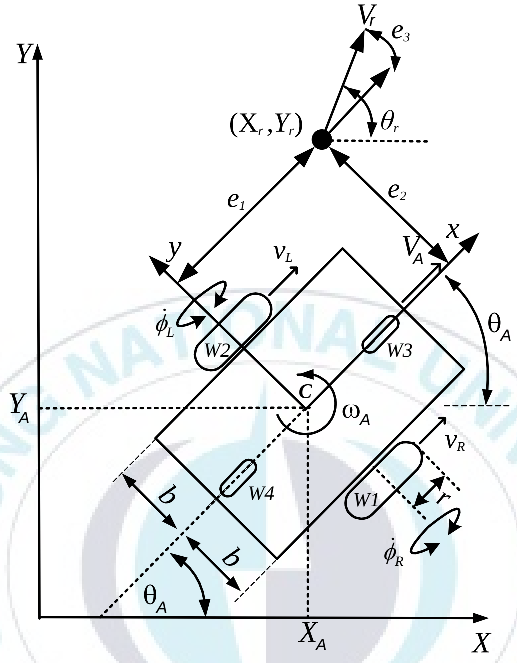


Fig. 4.1 Automatic Guided Vehicles (AGV) system model

The slip ratio of two wheels is defined as follows:

$$i = \frac{(r\dot{\phi}_L - v_L)}{r\dot{\phi}_L} = \frac{(r\dot{\phi}_R - v_R)}{r\dot{\phi}_R} \quad \text{and} \quad 0 \le i < 1 \tag{4.4}$$

where r is the radius of the wheel, $\dot{\phi}_L$ and $\dot{\phi}_R$ are the angular velocities of the left and the right wheels, respectively.

The wheel velocities with the slip ratio of two wheels are obtained

$$v_L = (1-i)r\dot{\phi}_L \quad \text{and} \quad v_R = (1-i)r\dot{\phi}_R \tag{4.5}$$

where v_L and v_R are the linear velocities of the left and right wheels with slip.

A slip parameter is defined as:

$$c = \frac{1}{(1-i)} \quad \text{and} \quad 0 \le i < 1 \tag{4.6}$$

The kinematic modeling of a AGV including the slip parameter can be written as

$$\begin{bmatrix} \dot{X}_A \\ \dot{Y}_A \\ \dot{\theta}_A \end{bmatrix} = \begin{bmatrix} \frac{r}{2c}\cos\theta_A & \frac{r}{2c}\cos\theta_A \\ \frac{r}{2c}\sin\theta_A & \frac{r}{2c}\sin\theta_A \\ \frac{r}{bc} & \frac{-r}{bc} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \Leftrightarrow \dot{\boldsymbol{\xi}}_1 = \mathbf{S}(\boldsymbol{\xi}_1)\dot{\boldsymbol{\Phi}} \tag{4.7}$$

If the velocity vector of the AGV in local coordinate is defined as $\boldsymbol{\eta}_A = \begin{bmatrix} V_A & \omega_A \end{bmatrix}^T$, the relations between $\boldsymbol{\Phi}$ and $\boldsymbol{\eta}_A$ including the slip can be described as follows:

$$\begin{bmatrix} V_A \\ \omega_A \end{bmatrix} = \begin{bmatrix} \frac{r}{2c} & \frac{r}{2c} \\ \frac{r}{bc} & -\frac{r}{bc} \end{bmatrix} \begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} \Leftrightarrow \boldsymbol{\eta}_A = \mathbf{T}\dot{\boldsymbol{\Phi}} \tag{4.8}$$

$$\begin{bmatrix} \dot{\phi}_R \\ \dot{\phi}_L \end{bmatrix} = \frac{1}{r} \begin{bmatrix} c & \frac{bc}{2} \\ c & -\frac{bc}{2} \end{bmatrix} \begin{bmatrix} V_A \\ \omega_A \end{bmatrix} \Leftrightarrow \dot{\boldsymbol{\Phi}} = \mathbf{T}^{-1}\boldsymbol{\eta}_A \tag{4.9}$$

Substituting Eq. (4.9) into Eq. (4.7) yields

$$\dot{\boldsymbol{\xi}}_1 = \begin{bmatrix} \dot{X}_A \\ \dot{Y}_A \\ \dot{\theta}_A \end{bmatrix} = \begin{bmatrix} \cos\theta_A & 0 \\ \sin\theta_A & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_A \\ \omega_A \end{bmatrix} \Leftrightarrow \begin{aligned} \dot{\boldsymbol{\xi}}_1 &= \mathbf{J}(\boldsymbol{\xi}_1)\boldsymbol{\eta}_A \\ &= \mathbf{S}(\boldsymbol{\xi}_1)\mathbf{T}^{-1}\boldsymbol{\eta}_A \end{aligned} \tag{4.10}$$

The following procedure is to design a control law that minimizes the tracking errors.

4.1.1 Non-adaptive kinematic controller design

The first procedure is to determine a desired velocity control law that drives the tracking error between the current posture vector $\boldsymbol{\xi}_1$ and the reference posture vector $\boldsymbol{\xi}_{\mathbf{r}} = \begin{bmatrix} X_r & Y_r & \theta_r \end{bmatrix}^T$. The reference posture vector $\boldsymbol{\xi}_r$ is generated by the path planning algorithm in chapter 3.

As shown in Fig. 4.1, a kinematic tracking error vector $\mathbf{e}_c$ and its time derivative $\dot{\mathbf{e}}_c$ are defined as:

$$\mathbf{e}_c = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta_A & \sin\theta_A & 0 \\ -\sin\theta_A & \cos\theta_A & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_r - X_A \\ Y_r - X_A \\ \theta_r - \theta_A \end{bmatrix} \tag{4.11}$$

$$\dot{\mathbf{e}}_c = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \cos e_3 & 0 \\ \sin e_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} V_r \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} V_A \\ \omega_A \end{bmatrix} \tag{4.12}$$

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

$$V_0 = \frac{1}{2}\left(e_1^2 + e_2^2\right) + \frac{\left(1 - \cos e_3\right)}{k_2} > 0 \tag{4.13}$$

and the kinematic derivatives becomes

$$\begin{aligned} \dot{V}_0 &= e_1\dot{e}_1 + e_2\dot{e}_2 + \dot{e}_3\frac{\sin e_3}{k_2} = e_1\left(-V_A + V_r\cos e_3\right) \\ &\quad + \frac{1}{k_2}\left(\sin e_3\right)\left(\omega_r - \omega_A + k_2 e_2 V_r\right) \end{aligned} \tag{4.14}$$

To achieve $\dot{V}_0 \le 0$, a desired kinematic control law vector is chosen as follows:

$$\boldsymbol{\eta}_\mathbf{d} \equiv \boldsymbol{\eta}_A = \begin{bmatrix} V_A \\ \omega_A \end{bmatrix} = \begin{bmatrix} V_r\cos e_3 + k_1 e_1 \\ \omega_r + k_2 V_r e_2 + k_3\sin e_3 \end{bmatrix} \tag{4.15}$$

Substituting Eq. (4.15) into Eq. (4.14) yields

$$\dot{V}_0 = -k_1 e_1^2 - \frac{k_3}{k_2}\sin e_3 \le 0 \tag{4.16}$$

The conditions $V_0 > 0$ and $\dot{V}_0 \le 0$ show the boundness of $\mathbf{e}_c$, and by Barbalat’s Lemma, Eq. (4.12), Eq. (4.15) and Eq. (4.16), the error $\mathbf{e}_c$ converge to zero.

The desired kinematic control law vector $\boldsymbol{\eta}_A = [V_A \quad \omega_A]$ of (4.15) can be used directly as the reference of dynamic controller $\boldsymbol{\eta}_\mathbf{d} = [V_{Ad} \quad \omega_{Ad}]$. Therefore, since ($\boldsymbol{\eta}_\mathbf{d} = \boldsymbol{\eta}_A$), a desired wheel angular velocity vector $\boldsymbol{\Phi}_\mathbf{d}$ is given by

$$\begin{bmatrix} \dot{\phi}_{Rd} \\ \dot{\phi}_{Ld} \end{bmatrix} = \frac{1}{r} \begin{bmatrix} c & \dfrac{bc}{2} \\ c & -\dfrac{cb}{2} \end{bmatrix} \begin{bmatrix} V_A \\ \omega_A \end{bmatrix} \Leftrightarrow \mathbf{\Phi_d} = \mathbf{T}^{-1}\boldsymbol{\eta}_\mathbf{d} \qquad (4.17)$$

4.1.2 Dynamic controller design

By a feedback linearization of the system, an input torque vector τ is defined by computed torque method as follows:

$$\boldsymbol{\tau} = \mathbf{\bar{M}(q)u_D} + \mathbf{\bar{V}}(\xi_1, \dot{\xi}_1)\dot{\mathbf{\Phi}} \qquad (4.18)$$

where $\mathbf{u_D}$ is the dynamic control input.

From Eq. (4.7), Eq. (2.23) and Eq. (4.18), dynamic control problem can be converted into the kinematic control problem as follows:

$$\begin{cases} \dot{\xi}_1 = \mathbf{S}(\xi_1)\dot{\mathbf{\Phi}} \\ \ddot{\mathbf{\Phi}} = \mathbf{u_D} \end{cases} \qquad (4.19)$$

To convert the kinematic desired wheel angular velocity control law $\dot{\mathbf{\Phi}}_\mathbf{d}$ into the input torque vector $\boldsymbol{\tau}$ that will be applied to the system, a dynamic tracking wheel angular velocity error vector is defined by

$$\mathbf{e_d} = \begin{bmatrix} e_4 \\ e_5 \end{bmatrix} = \dot{\mathbf{\Phi}} - \dot{\mathbf{\Phi}}_\mathbf{d} \qquad (4.20)$$

where $\dot{\mathbf{\Phi}} - \dot{\mathbf{\Phi}}_{\mathbf{d}}$ represents the error between the real wheel angular velocity vector $\dot{\mathbf{\Phi}}$ and desired wheel angular velocity vector $\dot{\mathbf{\Phi}}_{\mathbf{d}}$ of the AGV.

The dynamic control input $\mathbf{u_D} = [u_1 \quad u_2]^T$, which assures that $\mathbf{e_d}$ converges to zero, is given by the following expression:

$$\mathbf{u_D} = \ddot{\mathbf{\Phi}}_{\mathbf{d}} - \mathbf{K_4 e_d} \quad \text{and} \quad \mathbf{K_4} = \begin{bmatrix} k_4 & 0 \\ 0 & k_4 \end{bmatrix} \tag{4.21}$$

where k_4 is a positive constant.

From Eqs. (4.20) and (4.21), the derivative of the error $\mathbf{e_d}$ is given by

$$\dot{\mathbf{e}}_{\mathbf{d}} = \begin{bmatrix} \dot{e}_4 \\ \dot{e}_5 \end{bmatrix} = \ddot{\mathbf{\Phi}} - \ddot{\mathbf{\Phi}}_d = \mathbf{u_D} - \ddot{\mathbf{\Phi}}_d = -\mathbf{K_4 e_d} = -\begin{bmatrix} k_4 & 0 \\ 0 & k_4 \end{bmatrix}\begin{bmatrix} e_4 \\ e_5 \end{bmatrix} \tag{4.22}$$

To show that an entire tracking error vector $\mathbf{e} = [\mathbf{e_c} \quad \mathbf{e_d}]^T$ goes to zero at $t \to \infty$, the following Lyapunov function candidate is chosen:

$$\begin{aligned} V_1 &= V_0 + \frac{1}{2k_4}\left(e_4^2 + e_5^2\right) \\ &= \frac{1}{2}\left(e_1^2 + e_2^2\right) + \frac{\left(1 - \cos e_3\right)}{k_2} + \frac{1}{2k_4}\left(e_4^2 + e_5^2\right) > 0 \end{aligned} \tag{4.23}$$

and its derivatives becomes

$$
\begin{aligned}
\dot{V}_1 &= e_1\dot{e}_1 + e_2\dot{e}_2 + \dot{e}_3\frac{\sin e_3}{k_2} + \frac{e_4}{k_4}\dot{e}_4 + \frac{e_5}{k_4}\dot{e}_5 \\
&= -k_1e_1^2 - \frac{k_3}{k_2}\sin^2 e_3 - e_4^2 - e_5^2 \le 0
\end{aligned}
\tag{4.24}
$$

The conditions $V_1 > 0$ and $\dot{V}_1 \le 0$ shows the boundness of $\mathbf{e} = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 & e_5 \end{bmatrix}^T$, and by Barbalat's Lemma, Eq. (4.12), Eq. (4.15), Eq. (4.21), Eq. (4.22) and Eq. (4.24), it can be proved that the entire tracking error vector $\mathbf{e}$ converges to zero.

4.1.3 Adaptive backstepping control design

If the parameter i in Eq. (4.4) is unknown, c is unknown and the desired wheel angular velocity vector of Eq. (4.17) is invalid. Therefore, an update law to estimate the parameter $c = 1/(1-i)$ is needed.

Replacing $\mathbf{\Phi}$ with the desired wheel angular velocity vector $\mathbf{\Phi_d}$, Eq. (4.8) can be written into

$$
\begin{bmatrix} V_A \\ \omega_A \end{bmatrix} = \begin{bmatrix} \dfrac{r}{2c} & \dfrac{r}{2c} \\ \dfrac{r}{bc} & -\dfrac{r}{bc} \end{bmatrix} \begin{bmatrix} \dot{\phi}_{Rd} \\ \dot{\phi}_{Ld} \end{bmatrix}
\tag{4.25}
$$

Substituting Eq. (4.25) into Eq. (4.12), the derivative errors can be written as follows:

$$\begin{bmatrix}\dot{e}_1\\ \dot{e}_2\\ \dot{e}_3\end{bmatrix}=\begin{bmatrix}V_r\cos e_3\\ V_r\sin e_3\\ \omega_r\end{bmatrix}+\begin{bmatrix}-1 & e_2\\ 0 & -e_1\\ 0 & -1\end{bmatrix}\begin{bmatrix}\frac{r}{2c} & \frac{r}{2c}\\ \frac{r}{bc} & -\frac{r}{bc}\end{bmatrix}\begin{bmatrix}\dot{\phi}_{Rd}\\ \dot{\phi}_{Ld}\end{bmatrix} \tag{4.26}$$

For unknown constant slip parameter, Eq. (4.25) can't be applied. To solve this problem, the estimation $\hat{c}$ of c is needed. By introducing the estimation $\hat{c}$ for c, Eq. (4.9) can be written as:

$$\mathbf{\Phi_d}=\begin{bmatrix}\dot{\phi}_{Rd}\\ \dot{\phi}_{Ld}\end{bmatrix}=\frac{1}{r}\begin{bmatrix}\hat{c} & \frac{b}{2}\hat{c}\\ \hat{c} & -\frac{b}{2}\hat{c}\end{bmatrix}\begin{bmatrix}V_{As}\\ \omega_{As}\end{bmatrix}=\frac{1}{r}\hat{\mathbf{T}}^{-1}\boldsymbol{\eta}_{\mathbf{d}s} \tag{4.27}$$

where $\mathbf{\Phi_d}$ is the estimated desired wheel angular velocity vector, and $\boldsymbol{\eta}_{\mathbf{d}s}$ are the estimated desired velocity vector of the AGV, V_{As} and ω_{As} are estimated desired linear velocity and angular velocity of AGV in $\mathbf{\Phi_d}$.

Replacing c with $\hat{c}$ in Eq. (4.26), the kinematic modeling of a AGV can be written as

$$\begin{bmatrix}\dot{e}_1\\ \dot{e}_2\\ \dot{e}_3\end{bmatrix}=\begin{bmatrix}V_r\cos e_3\\ V_r\sin e_3\\ \omega_r\end{bmatrix}+\begin{bmatrix}-1 & e_2\\ 0 & -e_1\\ 0 & -1\end{bmatrix}\begin{bmatrix}\frac{r}{2\hat{c}} & \frac{r}{2\hat{c}}\\ \frac{r}{b\hat{c}} & -\frac{r}{b\hat{c}}\end{bmatrix}\begin{bmatrix}\dot{\phi}_{Rd}\\ \dot{\phi}_{Ld}\end{bmatrix} \tag{4.28}$$

Substituting Eq. (4.8) into Eq. (4.28) becomes:

$$\begin{bmatrix}\dot{e}_1\\ \dot{e}_2\\ \dot{e}_3\end{bmatrix}=\begin{bmatrix}V_r\cos e_3\\ V_r\sin e_3\\ \omega_r\end{bmatrix}+\begin{bmatrix}-1 & e_2\\ 0 & -e_1\\ 0 & -1\end{bmatrix}\begin{bmatrix}\frac{\hat{c}}{c} & 0\\ 0 & \frac{\hat{c}}{c}\end{bmatrix}\begin{bmatrix}V_{As}\\ \omega_{As}\end{bmatrix} \tag{4.29}$$

The estimation error is defined as follows:

$$\tilde{c} = c - \hat{c} \qquad (4.30)$$

where the c is the true value of the slip parameter, and $\tilde{c}$ is the estimation error of slip parameter.

Using Eq. (4.30), Eq. (4.29) can be rewritten as follows:

$$\dot{\mathbf{e}}_c = \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} V_r \cos e_3 \\ V_r \sin e_3 \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} (1-\frac{\tilde{c}}{c})V_{As} \\ (1-\frac{\tilde{c}}{c})\omega_{As} \end{bmatrix} \qquad (4.31)$$

To obtain an update law for obtaining the estimation $\hat{c}$, the following Lyapunov function is considered.

$$V_3 = V_1 + \frac{\tilde{c}^2}{2\gamma c} > 0 \qquad (4.32)$$

with $c \geq 1$ and $\gamma > 0$,

The derivatives of Eq. (4.32) becomes

$$\dot{V}_3 = e_1\dot{e}_1 + e_2\dot{e}_2 + \dot{e}_3\frac{\sin e_3}{k_2} + \frac{e_4}{k_4}\dot{e}_4 + \frac{e_5}{k_4}\dot{e}_5 + \frac{\tilde{c}\dot{\tilde{c}}}{\gamma c} \qquad (4.33)$$

Substituting Eqs. (4.22) and (4.31) into Eq. (4.33) yields:

$$\begin{aligned} \dot{V}_3 = {} & e_1\left(-V_{As} + V_r \cos e_3\right) + \frac{\sin e_3}{k_2}\left(\omega_r - \omega_{As} + k_2 e_2 V_r\right) \\ & - e_4^2 - e_5^2 - \frac{\tilde{c}}{c}\left[\frac{\dot{\hat{c}}}{\gamma} - \left(V_{As}e_1 + \frac{\omega_A \sin e_3}{2k_2}\right)\right] \end{aligned} \qquad (4.34)$$

To achieve $\dot{V}_3 \leq 0$, an estimated desired kinematic control law vector is chosen as follows:

$$\boldsymbol{\eta}_{\mathbf{d}s} = \begin{bmatrix} V_{As} \\ \omega_{As} \end{bmatrix} = \begin{bmatrix} V_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 V_r e_2 + k_3 \sin e_3 \end{bmatrix} \tag{4.35}$$

Therefore, the estimated desired kinematic control law vector for the system with unknown slip parameter in Eq. (4.35) is equal to the kinematic control law vector for the system without unknown slip parameter in Eq. (4.15).

The update law for obtaining the estimation $\hat{c}$ can be chosen as

$$\dot{\hat{c}} = \gamma \left(V_{As} e_1 + \frac{\omega_{As} \sin e_3}{2k_2} \right) \tag{4.36}$$

Substituting Eqs. (4.35) and (4.36) into Eq. (4.34), $\dot{V}_3$ becomes

$$\dot{V}_3 = \dot{V}_1 = -k_1 e_1^2 - \frac{k_3}{k_2} \sin^2 e_3 - e_4^2 - e_5^2 \leq 0 \tag{4.37}$$

To guarantees the closed-loop stability, it must be shown that the equilibrium point of $\mathbf{e} = \begin{bmatrix} e_1 & e_2 & e_3 & e_4 & e_5 \end{bmatrix}^T = \mathbf{0}$ is asymptotically stable.

Proof of asymptotic stability of $\mathbf{e} = \mathbf{0}$

The e_3 is bounded in the domain D as $D = \left\{ e \in R^3 \mid -\pi < e_3 < \pi \right\}$. The Lyapunov function given in Eq. (4.32) is positive definite in D with its derivative vector $\dot{V}_3 \leq 0$ in D. This implies that V_3 is a nonincreasing function that converges to some constant value and the entire tracking error vector $\mathbf{e}$ and the estimation parameter $\tilde{c}$ are bounded. Since the reference velocity vector $\boldsymbol{\eta}_{\mathbf{r}} = \begin{bmatrix} V_r & \omega_r \end{bmatrix}^T$ is assumed to be bounded, the desired kinematic velocity $\boldsymbol{\eta}_{\mathbf{d}s}$ of the AGV is also bounded from Eq. (4.35). Thus, $\dot{\mathbf{e}}$ is bounded by Eqs. (4.22) and (4.31). After all, $\ddot{V}_3(\mathbf{e},\dot{\mathbf{e}})$ given by

$$\ddot{V}_3 = -2k_1 e_1 \dot{e}_1 - \frac{2k_3}{k_2}\sin e_3 \cos e_3 \dot{e}_3 - 2e_4\dot{e}_4 - 2e_5\dot{e}_5 \tag{4.38}$$

is also bounded.

Since V_3 is a nonincreasing function that converges to some constant value, Barbalat's Lemma shows that $\dot{V}_3 \to 0$ as $t \to \infty$ from the boundness of $\ddot{V}_3$. Thus, e_1, e_3, e_4 and e_5 tend to zero as $t \to \infty$. From Eq. (4.36), $\hat{c}$ is constant. e_2 must also converge to zero. From Eqs. (4.31) and (4.35), the followings are obtained.

$$\begin{cases} \dot{e}_1 = V_r + \left[(1-\dfrac{\tilde{c}}{c})(-V_r + e_2(\omega_r + k_2 V_r e_2)) \right] = 0 \\ \dot{e}_2 = 0 \to e_2 = 0,\ \text{constant} \\ \dot{e}_3 = \omega_r - \omega_{As} = -(1-\dfrac{\tilde{c}}{c})V_r k_2 e_2 + \dfrac{\tilde{c}}{c}\omega_r = 0 \end{cases} \tag{4.39}$$

Since $e_3 \to 0$, $\dot{e}_3 = 0$ from Eq. (4.39). If $V_r \neq 0$ and $\tilde{c} = 0$ as $t \to \infty$, then $e_2 \to 0$ as $t \to \infty$. Thus, the equilibrium point $\mathbf{e} = 0$ is asymptotically stable. e_2 is constant for $\tilde{c} \neq 0$.

EOD

Fig. 4.2 shows the block diagram of proposed controller.

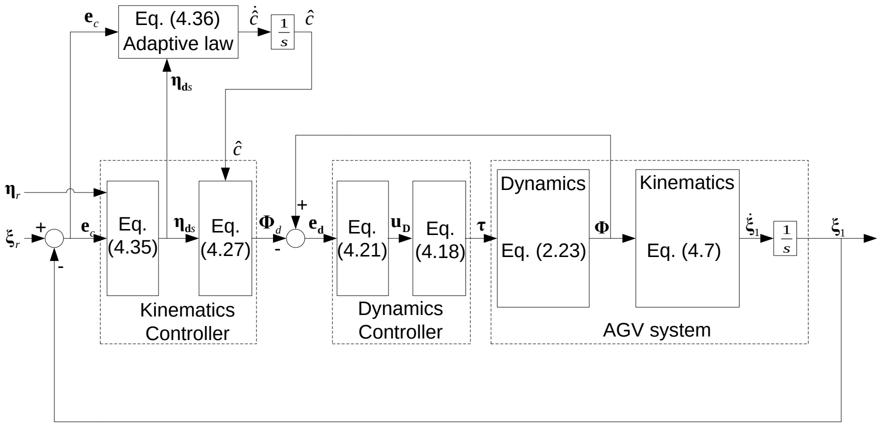


Fig. 4.2 Block diagram of the proposed adaptive backstepping control

4.2 Robust servo controller design

In this dissertation, a different concept based servo controller design method is proposed for AGV trajectory control by adopting polynomial differential operator method based on internal model principle of Kim et al [86-87].

By substituting Eq. (2.18) into dynamic model of AGV in Eq. (2.23), the following can be obtained:

$$\bar{\mathbf{M}}(\xi_1)\mathbf{T}^{-1}\dot{\boldsymbol{\eta}}_A + \bar{\mathbf{V}}(\xi_1,\dot{\xi}_1)\mathbf{T}^{-1}\boldsymbol{\eta}_A = \boldsymbol{\tau} \tag{4.40}$$

Rearrange Eq. (4.40) nonlinear model of dynamic system can be obtained as follows:

$$\dot{\boldsymbol{\eta}}_A = \mathbf{L}\boldsymbol{\eta}_A + \boldsymbol{\Lambda}\boldsymbol{\tau} \tag{4.41}$$

where $\mathbf{L} = (\bar{\mathbf{M}}\mathbf{T}^{-1})^{-1}\bar{\mathbf{V}}(\xi_1,\dot{\xi}_1)\mathbf{T}^{-1}$ and $\boldsymbol{\Lambda} = (\bar{\mathbf{M}}\mathbf{T}^{-1})^{-1}$

To allow the application of linear controller design, the system equations should be rewritten into a linear representation of the AGV dynamic behavior. The set of state variables $\boldsymbol{\zeta}_1 = \begin{bmatrix} X_A & Y_A & \theta_A \end{bmatrix}^T$ will be described in terms of the AGV linear displacement l_A and the AGV orientation θ_A such that $\boldsymbol{\zeta}_2 = \begin{bmatrix} l_A & \theta_A \end{bmatrix}^T$. The complete AGV mathematic model as follows:

$$\begin{bmatrix} \dot{\boldsymbol{\eta}}_{\mathbf{A}} \\ \dot{\boldsymbol{\zeta}}_{\mathbf{2}} \end{bmatrix} = \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{I}_2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_{\mathbf{A}} \\ \boldsymbol{\zeta}_{\mathbf{2}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Lambda} \\ \mathbf{0} \end{bmatrix} \begin{bmatrix} \tau_L \\ \tau_R \end{bmatrix} \tag{4.42}$$

$$\begin{bmatrix} l_A \\ \theta_A \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_{\mathbf{A}} \\ \boldsymbol{\zeta}_{\mathbf{2}} \end{bmatrix} \tag{4.43}$$

where $\mathbf{I}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $\dot{l}_A = V_A$ and $\dot{\theta}_A = \omega_A$

The linear equivalent model was obtained without any simplification assumption. In this way, the linear model is equivalent representation of the original non-linear representation.

The AGV model in Eqs. (42)-(43) can be described by the following MIMO linear time invariant model with a disturbance as

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) + \boldsymbol{\varepsilon} \tag{4.44}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \tag{4.45}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{L} & \mathbf{0} \\ \mathbf{I}_2 & \mathbf{0} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} \boldsymbol{\Lambda} \\ \mathbf{0} \end{bmatrix} \quad \mathbf{C} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\mathbf{x} = \begin{bmatrix} V_A & \omega_A & l_A & \theta_A \end{bmatrix}^T$, $\mathbf{y} = \begin{bmatrix} l_A & \theta_A \end{bmatrix}^T$, and $\mathbf{u} = \begin{bmatrix} \tau_R & \tau_L \end{bmatrix}^T$

where $\mathbf{x} \in \mathbf{R}^n$ is the state vector, $\mathbf{y} \in \mathbf{R}^p$ is the output vector, $\mathbf{u} \in \mathbf{R}^m$ is the dynamic control input torque vector, $\mathbf{A} \in \mathbf{R}^{n \times n}$ is the system matrix, $\mathbf{B} \in \mathbf{R}^{n \times m}$ is the input matrix, $\mathbf{C} \in \mathbf{R}^{p \times n}$ is the output matrix. It is assumed that $(\mathbf{A}, \mathbf{B})$ is controllable, and $(\mathbf{A}, \mathbf{C})$ is observable. $\boldsymbol{\varepsilon} \in \mathbf{R}^n$ is an unmeasurable disturbance vector representing the slip condition.

The output error vector of AGV is defined as

$$\mathbf{e}_{\mathrm{I}} = \begin{bmatrix} e_l \\ e_\theta \end{bmatrix} = \begin{bmatrix} l_r - l_A \\ \theta_r - \theta_A \end{bmatrix} \tag{4.46}$$

where l_r, θ_r are the AGV reference linear displacement and the AGV reference orientation.

The i^{th} output error vector is defined by

$$e_i(t) = y_i(t) - y_{ri}(t) \tag{4.47}$$

where y_{ri} is the i^{th} reference output vector and e_i is the i^{th} output error vector.

It is assumed that the following homogeneous differential equation forms for reference and disturbance using linear polynomial differential operators of $L_r(D)$ and $L_\varepsilon(D)$ are satisfied, respectively:

$$L_r(D) y_{ri}(t) = 0 \tag{4.48}$$

$$L_\varepsilon(D) \varepsilon_i = 0 \tag{4.49}$$

where $L_r(D)$ and $L_\varepsilon(D)$ are assumed as the following polynomial differential operators with constant coefficients.

$$L_r(D) = D^\sigma + \rho_{\sigma-1}D^{\sigma-1} + \cdots + \rho_0 \tag{4.50}$$

$$L_\varepsilon(D) = D^l + \mu_{l-1}D^{l-1} + \cdots + \mu_0 \tag{4.51}$$

where $D = d/dt$ is the differential operator, ρ_i, μ_i are constant coefficients, and σ, l are orders of differential polynomials.

This includes the case of most common type of disturbance and reference signals occurring in practice such as polynomial, sinusoidal type signals, etc. $R(D)$ is the greatest common divisor of $L_r(D)$ and $L_\varepsilon(D)$ as follows:

$$L_r(D) = R(D)U(D) \tag{4.52}$$

$$L_\varepsilon(D) = R(D)V(D) \tag{4.53}$$

where $U(D), V(D)$ are factors of $L_r(D)$ and $L_\varepsilon(D)$, respectively.

If the greatest common divisor $R(D)$ of the differential polynomial operators $L_r(D)$ and $L_\varepsilon(D)$ is equal to a constant, $L_r(D)$ and $L_\varepsilon(D)$ are coprime. Furthermore, finding the least common multiple of the two differential polynomial operators involves finding their common multiple with the smallest order polynomial.

$L(D)$ is defined as the least common multiple of $L_r(D)$ and $L_\varepsilon(D)$ and can be obtained from Eqs. (4.52) - (4.53) using polynomial differential operators with constant coefficients as follows:

$$\begin{aligned} L(D) &= \frac{L_\varepsilon(D)L_r(D)}{R(D)} = U(D)R(D)V(D) \\ &= V(D)L_r(D) \text{ or } U(D)L_\varepsilon(D) \\ &= D^q + \alpha_{q-1}D^{q-1} + \cdots + \alpha_1 D + \alpha_0 \end{aligned} \tag{4.54}$$

where $\dim\{V(D)\} = q - \sigma, \dim\{U(D)\} = q - l, \dim\{R(D)\} = l + \sigma - q,$ and $\dim\{L(D)\} = q \geq \dim\{L_r(D)\}$ or $\dim\{L_\varepsilon(D)\}$.

In order to adopt the internal model principle (IMP) to the tracking control system, we will introduce the polynomial differential operator stated and get an extended system.

Operating $L(D)$ of Eq. (4.54) for ε_i and y_{ri}, the following are obtained

$$\begin{cases} L(D)y_{ri} = U(D)R(D)V(D)y_{ri} = V(D)L_r(D)y_{ri} = 0 \\ L(D)\varepsilon_i = U(D)R(D)V(D)\varepsilon_i = U(D)L_w(D)\varepsilon_i = 0 \end{cases} \tag{4.55}$$

where the dimension of q holds $q \geq l \; or \; q \geq \sigma$.

The adaptation of IMP to the servo control system design is attempted by 3 steps:

[Step 1] by operating the polynomial differential operator to the given system of Eq. (4.44) and also the output error of Eq. (4.47).

[Step 2] an extended system is obtained by using the operated system and output errors obtained through the step 1

[Step 3] for the extended system, a regulator problem is solved based on the well known design method such as pole assignment or optimal control.

4.2.1 Operating the polynomial differential operator

Firstly, to eliminate the effect of disturbance in Eq. (4.44), operating the polynomial differential operator of $L(D)$ to both sides of Eq. (4.44) by using Eq. (4.49) can be written as

$$\frac{d}{dt}\{L(D)\mathbf{x}\} = \mathbf{A}L(D)\mathbf{x} + \mathbf{B}L(D)\mathbf{u} \tag{4.56}$$

The i^{th} output error can be written as

$$e_i(t) = y_i(t) - y_{ri}(t) \text{ for } i = 1 \sim p \tag{4.57}$$

Secondly, operating $L(D)$ to the i^{th} item of Eq. (4.47) and using the property of Eq. (4.55), the followings can be obtained.

$$\begin{aligned} L(D)e_i(t) &= D^q e + \alpha_{q-1}D^{q-1}e + \cdots + \alpha_1 De + \alpha_0 e \\ &= L(D)y_i - L(D)y_{ri} = \mathbf{c}_i^T L(D)\mathbf{x} \end{aligned} \tag{4.58}$$

or

$$D^q e = -\alpha_{q-1}D^{q-1}e - \cdots - \alpha_1 De - \alpha_0 e + \mathbf{c}_i^T L(D)\mathbf{x} \tag{4.59}$$

Eq. (4.59) can be described into the matrix form as follows:

$$\begin{aligned}\dot{\mathbf{z}}_i &= \mathbf{M}_i L(D)\mathbf{x} + \mathbf{N}\mathbf{z}_i \\ &= \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{c}_1^T \end{bmatrix} L(D)x + \mathbf{N}\mathbf{z}_i \end{aligned} \tag{4.60}$$

where

$$\mathbf{N} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & -\alpha_3 & \cdots & -\alpha_{q-1} \end{bmatrix} \in R^{q\times q},$$

$$\mathbf{M}_i = \begin{bmatrix} \mathbf{0} \\ \cdots \\ \mathbf{c}_i^T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \mathbf{c}_i^T \in R^{q\times n}, \quad \mathbf{C}^T = \begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_p \end{bmatrix} \in R^{n\times p},$$

$$\mathbf{c}_i = \begin{bmatrix} c_{i1} & c_{i2} & \cdots & c_{in} \end{bmatrix}^T \in R^n \text{ and } \mathbf{z}_i = \begin{bmatrix} e_i & e_i^{(1)} & \cdots & e_i^{(q-1)} \end{bmatrix}^T \in R^q$$

4.2.2 Extended system

By combining the operated system Eq. (4.56) and Eq. (4.60), an extended system can be obtained as follows:

$$\dot{\mathbf{x}}_e = \mathbf{A}_e\mathbf{x}_e + \mathbf{B}_e\mathbf{v} \tag{4.61}$$

where

$$\mathbf{A_e} = \begin{bmatrix} \mathbf{A} & 0 & \cdots & 0 & 0 \\ \begin{bmatrix} \mathbf{0} \\ \mathbf{c}_1^T \end{bmatrix} & \mathbf{N} & 0 & \vdots & 0 \\ \begin{bmatrix} \mathbf{0} \\ \mathbf{c}_2^T \end{bmatrix} & 0 & \mathbf{N} & 0 & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ \begin{bmatrix} \mathbf{0} \\ \mathbf{c}_p^T \end{bmatrix} & 0 & \cdots & 0 & \mathbf{N} \end{bmatrix} \in R^{(n+pq)\times(n+pq)}, \quad \mathbf{B_e} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \\ \vdots \\ \vdots \\ \mathbf{0} \end{bmatrix} \in R^{(n+pq)\times m},$$

$$\mathbf{x_e} = \begin{bmatrix} L(D)x \\ \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_p \end{bmatrix} \in R^{n+pq}$$

where $\mathbf{x_e}$ is an extended system state variable vector, $\mathbf{v} = L(D)\mathbf{u} \in R^m$ is a new control law for the extended system, and $\mathbf{z} = \begin{bmatrix} \mathbf{z}_1 & \mathbf{z}_2 & \cdots & \mathbf{z}_q \end{bmatrix}^T \in R^{pq}$ is an error variable vector for the extended system.

A new control law for the extended system is defined by the following form:

$$\mathbf{v} = L(D)\mathbf{u} = -\mathbf{F}\mathbf{x}_e \in R^m \tag{4.62}$$

where $\mathbf{F} = \begin{bmatrix} \mathbf{F}_x & \mathbf{F}_z \end{bmatrix} \in R^{m\times(n+q)}$ is a feedback control gain matrix, and $\mathbf{F}_x \in R^{m\times n}$ and $\mathbf{F}_z \in R^{m\times pq}$ are feedback control gain matrices for $L(D)x$ and $\mathbf{z}$, respectively.

A new error variable vector for the extended system can be defined as

$$\boldsymbol{\zeta} = L^{-1}(D)\mathbf{z} \tag{4.63}$$

where $\boldsymbol{\zeta} = \begin{bmatrix} \zeta_1 & \zeta_2 & \cdots & \zeta_q \end{bmatrix}^T \in R^{pq}$, and $\zeta_i \in R^q$ for $i = 1, \cdots, q$.

Using Eq. (4.63), the control law of Eq. (4.44) can be obtained as follows:

$$\mathbf{u} = -\mathbf{F}\mathbf{x}_\zeta = -\begin{bmatrix} \mathbf{F}_x & \mathbf{F}_z \end{bmatrix}\begin{bmatrix} \mathbf{x} \\ \boldsymbol{\zeta} \end{bmatrix} \tag{4.64}$$

where $\mathbf{x}_\zeta \in R^{n+pq}$ is a new extended system variable vector.

Controllability of the extended system can be easily checked. Given the system of Eq. (4.44) with the assumptions of Eq. (4.55) for the disturbance and reference inputs, the extended system Eq. (4.61) obtained by operating $L(D)$ is controllable if the following two conditions are held:

1. The system $(\mathbf{A}, \mathbf{B})$ of Eq. (4.44) is controllable.
2. The following matrix $\bar{\mathbf{V}}_e$ of Eq. (4.65) related with the extended system of Eq. (4.61) has $rank(\bar{\mathbf{V}}_e) = n + pq$.

where

$$\bar{\mathbf{V}}_e = \begin{bmatrix} \mathbf{B} & \boldsymbol{\gamma}_1 & \boldsymbol{\gamma}_2 & \cdots & \boldsymbol{\gamma}_{q-1} & \boldsymbol{\gamma}_q \\ 0 & \mathbf{CB} & \mathbf{CAB} & \cdots & \mathbf{CA}^{q-2}\mathbf{B} & \mathbf{CA}^{q-1} \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{CB} & \mathbf{CA} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{C} \end{bmatrix} \in R^{(n\times pq)\times(n\times mq)} \qquad (4.65)$$

and $\quad m \geq p$

with

$$\boldsymbol{\gamma}_1 = \mathbf{AB} + \alpha_{q-1}\mathbf{B} \in R^{n\times m}$$

$$\boldsymbol{\gamma}_2 = \mathbf{A}^2\mathbf{B} + \alpha_{q-1}\mathbf{AB} + \alpha_{q-2}\mathbf{B} \in R^{n\times m}$$

$$\vdots$$

$$\boldsymbol{\gamma}_{q-1} = \mathbf{A}^{q-1}\mathbf{B} + \alpha_{q-1}\mathbf{A}^{q-2}\mathbf{B} + \cdots + \alpha_1\mathbf{B} \in R^{n\times m}$$

$$\boldsymbol{\gamma}_q = \mathbf{A}^q + \alpha_{q-1}\mathbf{A}^{q-1} + \alpha_{q-2}\mathbf{A}^{q-2} + \cdots + \alpha_1\mathbf{A} + \alpha_0\mathbf{I} \in R^{n\times n}$$

From Eq. (4.61) and Eq. (4.62), the closed loop system of the extended system is obtained as

$$\dot{\mathbf{x}}_e = (\mathbf{A}_e - \mathbf{B}_e\mathbf{F})\mathbf{x}_e \qquad (4.66)$$

which shows that the servo controller problem for Eq. (4.44) with reference and disturbance of Eqs. (4.54)-(4.55) becomes a regulator design problem for the extended system Eq. (4.61). That is, the closed loop system of Eq. (4.66) is asymptotically stabilized by designing the feedback control law of Eq. (4.62) with a feedback control matrix $\mathbf{F}$ so as to be $\mathrm{Re}\{\lambda_i(\mathbf{A}_e - \mathbf{B}_e\mathbf{F}\} < 0$ [87–89].

$$\mathbf{A}_F = \mathbf{A}_e - \mathbf{B}_e\mathbf{F} = \mathbf{A}_e - \mathbf{B}_e\left[\mathbf{F}_x \quad \mathbf{F}_z\right] \qquad (4.67)$$

When the feedback control law of the extended system of Eq. (4.61) is designed based on the above regulator design problem, the output error of Eq. (4.47) becomes $e(t) \to 0$ as $t \to \infty$.

Since $\mathbf{x}_e \to 0$ by regulator design result, that is $\mathbf{x}_e \to 0$ means $\begin{bmatrix} \mathbf{Lx}^T & \mathbf{z}^T \end{bmatrix}^T \to 0$ as follows:

$$\begin{aligned}
\mathbf{x_e} &= \begin{bmatrix} L(D)\mathbf{x}^T & \mathbf{z}_1^T & \mathbf{z}_2^T & \cdots & \mathbf{z}_p^T \end{bmatrix}^T \\
&= \begin{bmatrix} L(D)\mathbf{x}^T & \begin{bmatrix} e_1 & e_2^{(1)} & \cdots & e_1^{(q-1)} \end{bmatrix}^T & \cdots & \begin{bmatrix} e_p & e_p^{(1)} & \cdots & e_p^{(q-1)} \end{bmatrix}^T \end{bmatrix}^T
\end{aligned} \tag{4.68}$$

As the result, the error $e(t) = \begin{bmatrix} e_1(t) & e_2(t) & \cdots & e_p(t) \end{bmatrix}^T \to 0$ as $t \to \infty$.

Using the new input $\mathbf{v}$, the extended system of Eq. (4.61) can be written by

$$\frac{d}{dt}\{L(D)\mathbf{x}\} = [\mathbf{A} - \mathbf{BF}_x] L(D)\mathbf{x} - \mathbf{BF}_z\mathbf{z} + L(D)\boldsymbol{\varepsilon} \tag{4.69}$$

$$\frac{d}{dt}\mathbf{z} = \mathbf{N}_z\mathbf{z} + \mathbf{I}_\zeta L(D)\mathbf{e} \tag{4.70}$$

where $L(D)\boldsymbol{\varepsilon} = 0$ in Eq. (4.55)

$$\mathbf{N}_z = \begin{bmatrix} \mathbf{N} & 0 & \cdots & 0 \\ 0 & \mathbf{N} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{N} \end{bmatrix} \in R^{pq \times pq}, \mathbf{I}_\zeta = \begin{bmatrix} \xi & 0 & \cdots & 0 \\ 0 & \xi & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \xi \end{bmatrix} R^{pq \times p}, \quad \boldsymbol{\xi} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \in R^q$$

The 2nd term for the polynomial differential operator of $L(D)$ in Eq. (4.70) due to Eq. (4.55) has the following relation:

$$L(D)\mathbf{e} = L(D)(\mathbf{y} - \mathbf{y}_r) = \mathbf{C}L(D)\mathbf{x} \tag{4.71}$$

By operating the inverse polynomial differential operator $L^{-1}(D)$ for Eq. (4.69) and Eq. (4.70), the following equations can be obtained due to $L^{-1}(D)\left(L(D)\boldsymbol{\varepsilon}\right) = \boldsymbol{\varepsilon}$ by $L(D)\boldsymbol{\varepsilon} = 0$:

$$\frac{d}{dt}\mathbf{x} = \left[\mathbf{A} - \mathbf{B}\mathbf{F}_x\right]\mathbf{x} - \mathbf{B}\mathbf{F}_z\boldsymbol{\zeta} + \boldsymbol{\varepsilon} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \boldsymbol{\varepsilon} \tag{4.72}$$

$$\frac{d}{dt}\boldsymbol{\zeta} = \mathbf{N}_z\boldsymbol{\zeta} + \mathbf{I}_\zeta\mathbf{e} \tag{4.73}$$

$$\mathbf{u} = -\begin{bmatrix}\mathbf{F}_x & \mathbf{F}_z\end{bmatrix}\begin{bmatrix}\mathbf{x} \\ \boldsymbol{\zeta}\end{bmatrix} \tag{4.74}$$

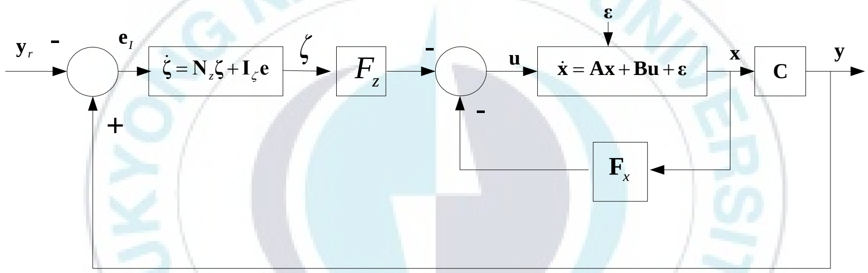
which holds the servo compensator of Eq. (4.70).

4.2.3 Pole assignment

Given the augmented system as in Eq. (4.61) and a pole vector $\mathbf{p}$ of desired self-conjugate closed-loop pole locations, the problem is to computes a gain matrix $\mathbf{F}$ such that the state feedback $\mathbf{u} = -\mathbf{F}\mathbf{x}$ places the closed-loop poles at the locations $\mathbf{p}$. In other words, the eigenvalues of $\mathbf{A}_e - \mathbf{B}_e\mathbf{F}$ match the entries of $\mathbf{p}$.

The optimal algorithm based on [86] is used to places the desired closed-loop poles p by computing a state-feedback gain matrix $\mathbf{F}$. All the inputs of the plant are assumed to be control inputs. The length of $\mathbf{p}$ must match the row size of $\mathbf{A}_e$. This algorithm uses the extra degrees of freedom to find a solution that minimizes the sensitivity of the closed-loop poles to perturbations in $\mathbf{A}_e$ or $\mathbf{B}_e$.

The servo compensator of Eq. (4.73) includes the model of reference and disturbance signals since the matrix $\mathbf{N}_z$ is composed of the least common multiple model of two signals. It proposes the internal model principle based on the polynomial differential operator. The configuration of the proposed servo control system can be described as shown in Fig. 4.3.



$\mathbf{y}_r$
-
$\mathbf{e}_I$
+
$\dot{\boldsymbol{\zeta}} = \mathbf{N}_z\boldsymbol{\zeta} + \mathbf{I}_\zeta\mathbf{e}$
ζ
F_z
-
u
-
$\boldsymbol{\varepsilon}$
$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \boldsymbol{\varepsilon}$
x
C
y
$\mathbf{F}_x$

Fig. 4.3 Configuration of the proposed robust servo control system

4.3 Simulation and experimental results

To clarify the effectiveness of the proposed controllers, simulation and experiment are done. The physical parameter and initial values for the model are given by Table 4.1.

Table 4.1 Parameter and initial values of AGV model

Parameter	Value	Parameter	Value
b	0.3 m	d	0 m
r	0.075 m	$X_A(0)$	1.5m
m_w	1 kg	$Y_A(0)$	2.5m
m_c	80 kg	$\theta_A(0)$	$\pi/2$ rad
I_w	0.187 kg m^2	$V_A(0)$	0m/s
I_m	1.08kg m^2	$\omega_A(0)$	0rad/s
I_c	9.5kg m^2		

The reference trajectory is generated by the proposed path planning algorithm in chapter 3. The reference trajectory is shown as in Fig. 4.4.

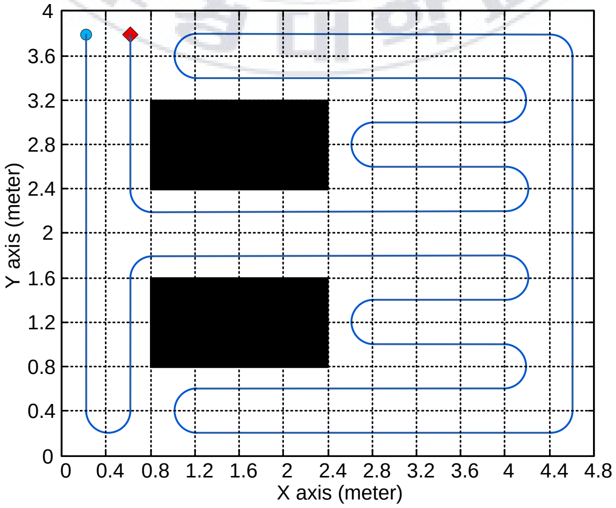


Fig. 4.4 Reference trajectory

4.3.1 Adaptive backstepping controller

The parameters for simulation and experiment of the proposed adaptive backstepping controller are shown in Table 4.3.

Table 4.2 Parameter values for the proposed adaptive backstepping controller

Parameter	Value	Parameter	Value
k_1	3 s^{-1}	k_4	10 s^{-2}
k_2	3 m^{-2}	γ	100 m^{-2}
k_3	2 rad/s	$\hat{c}(0)$	1

The simulation and experiment results are shown in Figs. 4.5 – 4.13. To verify the effectiveness of the proposed adaptive backstepping controller, the slip parameter c = 1.25 is added every time the AGV turning as shown in Fig. 4.5. In straight condition, slip parameter is c = 1. In this simulation and experiment, slip parameters are 1.2 at the time intervals $t = 35s - 49s$, $t = 61s - 68s$, $t = 100s - 114s$, $t = 126s - 140s$, $t = 152s - 166s$, $t = 195s - 209s$, $t = 241s - 248s$, $t = 281s - 288s$, $t = 320s - 334s$, $t = 363s - 377s$, $t = 389s - 403s$, $t = 414s - 428s$, and $t = 461s - 468s$. Otherwise, slip parameters are 1.

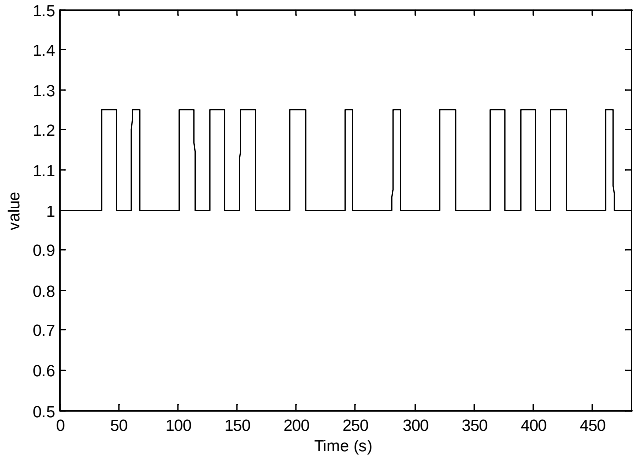


Fig. 4.5 Slip parameter c

Fig. 4.6 shows the desired kinematic control law vector $\boldsymbol{\eta}_{\mathbf{d}}$. The linear velocity V_A is 0.1 m/s in straight path, and V_A is 0.05 m/s in turning path. The angular velocity ω_A is 0 rad/s in straight path and ± 0.25 rad/s in turning path. The kinematic control law makes AGV be adaptive to the changing of slip parameter.

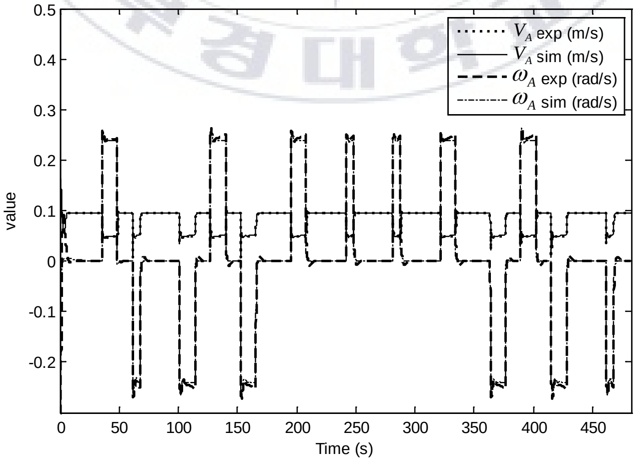


Fig. 4.6 Desired kinematic control law vector $\boldsymbol{\eta}_{\mathbf{d}}$ of the proposed adaptive backstepping controller

Fig. 4.7 shows the dynamic control input torque vector $\boldsymbol{\tau}$. At the starting points and the end points of turning, the torque vector $\boldsymbol{\tau}$ is changed roughly and bounded within ± 6 Nm since the desired wheel angular velocities are changed and slip parameters is changed.

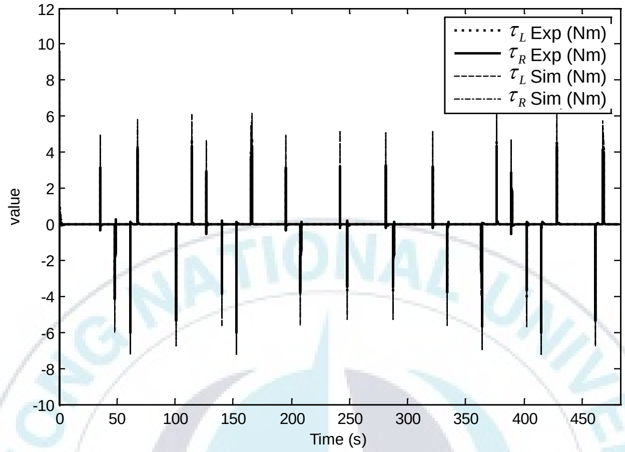


Fig. 4.7 Input torque vector $\boldsymbol{\tau}$ of the proposed adaptive backstepping controller

Fig. 4.8 shows the dynamic control input vector $\mathbf{u_D}$ of the proposed adaptive backstepping controller. The controller values are bounded within $\pm 20\,\text{rad/s}^2$ since the desired wheel angular velocities are changed and slip parameters is changed.

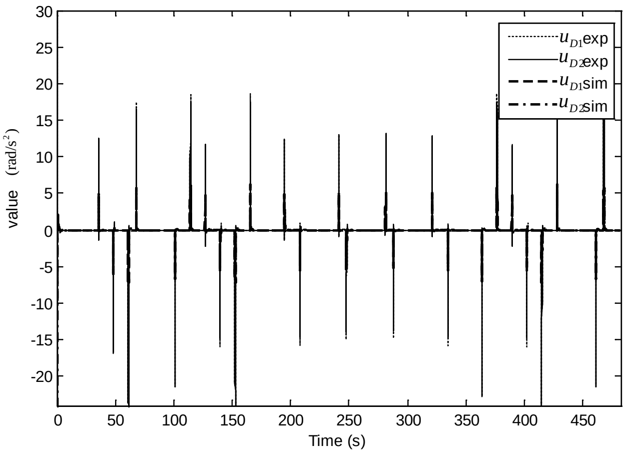


Fig. 4.8 Dynamic control input vector $\mathbf{u_D}$ of the proposed adaptive backstepping controller

Fig. 4.9 shows the trajectory tracking result obtained from simulation and experiment. At the beginning, the AGV adjusts its posture quickly to reduce the initial error and tracks the reference trajectory successfully.

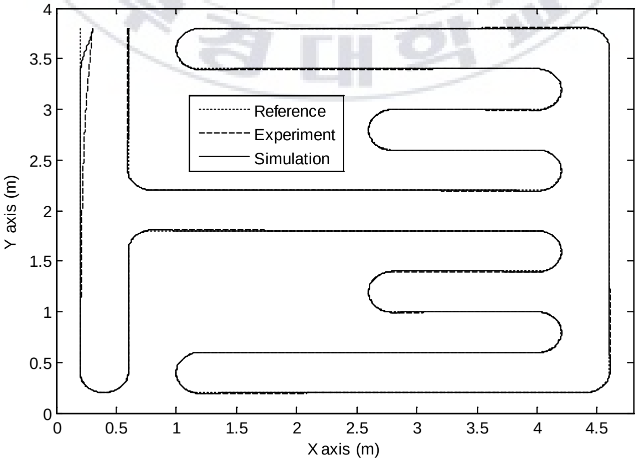


Fig. 4.9 Result for the planned reference trajectory tracking of the proposed adaptive backstepping controller

Fig. 4.10 shows the kinematic tracking error vector $\mathbf{e}_c$. It shows the kinematic tracking errors converge to zero after 5 s and at the times $t = 35s$ and $t = 50s$, the kinematic tracking errors increase due to the step change in the slip parameter. At the starting points and the end points of turning, the kinematic tracking error vector $\mathbf{e}_c$ is changed roughly since the reference velocities are changed. The e_1 is bounded within $\pm 0.01m$, the e_2 is bounded within $\pm 0.015m$, and the e_3 is bounded within $\pm 0.03rad$.

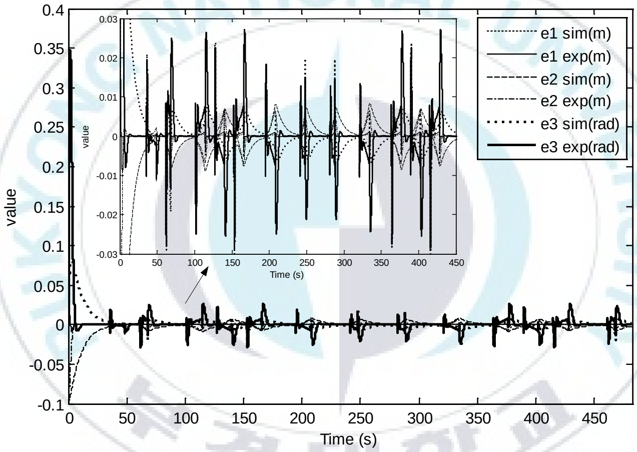


Fig. 4.10 Kinematic tracking error vector $\mathbf{e_c}$ for the proposed adaptive backstepping controller

Fig. 4.11 shows the dynamic tracking wheel angular velocity error vector $\mathbf{e_d}$. It can be seen that the tracking wheel angular velocity error vector also converges to zero. The dynamic tracking wheel angular velocity error vector $\mathbf{e_d}$ converges to zero faster than the kinematic tracking wheel angular error vector $\mathbf{e}_c$. At the starting points and the end points of turning, the dynamic tracking wheel angular velocity error vector $\mathbf{e_d}$ is changed roughly since the reference wheel angular velocities are changed. The error vector $\mathbf{e_d}$ is bounded within $\pm 0.06 rad/s$.

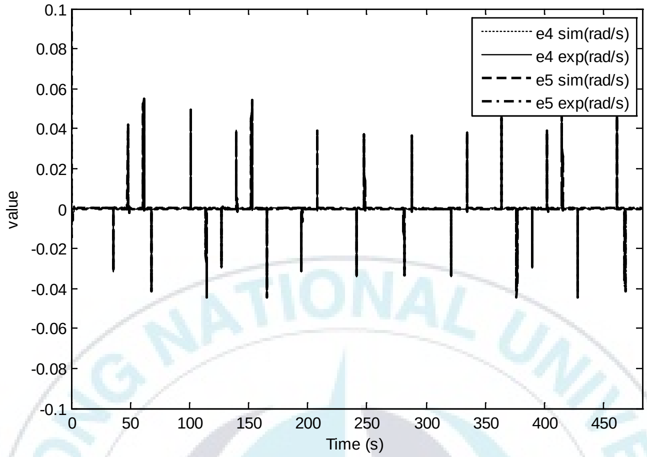


Fig. 4.11 Dynamic tracking velocity error vector $\mathbf{e_d}$ for the proposed adaptive backstepping controller

Fig. 4.12 shows the estimation $\hat{c}$ of the slip parameter. From $t = 0s$ to $t = 35s$, the value of c is 1. This represents the condition without slip $(i = 0)$. From $t = 35s$ to $t = 50s$, the value of c is 1.25. This represents the condition with slip $(i = 0.2)$. The simulation and experiment result shows that estimation $\hat{c}$ follows the real value c. At $t = 0s$, the errors e_c, e_d are not equal to zero. Therefore, the value of estimation $\hat{c}$ is changed. At $t = 5s$, the value of estimation $\hat{c}$ converges to the real value c. At $t = 35s$ and $t = 50s$, the value of estimation $\hat{c}$ successfully converges to the real value. At the starting points and the end points of turning, the estimation $\hat{c}$ of the slip parameter is changed roughly since the reference velocities are changed.

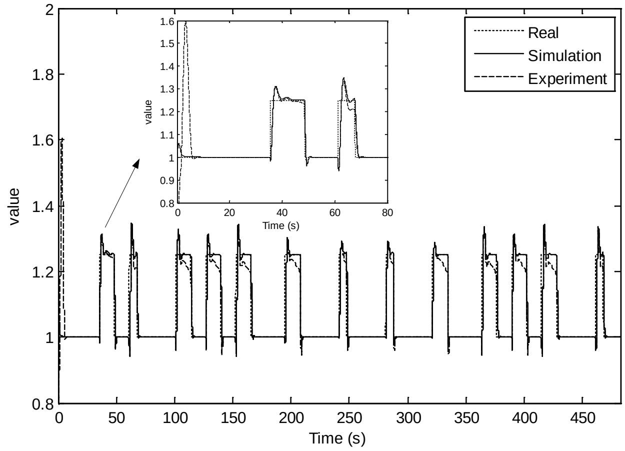


Fig. 4.12 Estimation $\hat{c}$ of the slip parameter of the proposed adaptive backstepping controller

Fig. 4.13 shows the update law $\dot{\hat{c}}$ for estimating of the slip parameter. The value of update law is changed since the value of slip parameter is increased. The update law is bounded within -0.6~0.4 s^{-1}.

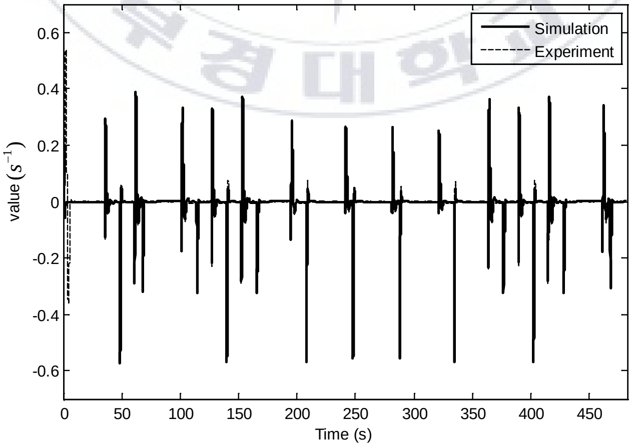


Fig. 4.13 Update law of the proposed adaptive backstepping control

4.3.2 Robust servo controller

The parameter values of the extended system for simulation and experiment in the proposed robust servo controller designed based on internal model principle are shown in Table 4.3.

Table 4.3 Parameter values for the proposed robust servo controller

Parameter	Value
Matrix $\mathbf{A}$	$\mathbf{A}=\begin{bmatrix} -0.0127 & 2.9703\cdot 10^{-4} & 0 & 0 \\ 0.0074 & -5.09999\cdot 10^{-4} & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$
Matrix $\mathbf{B}$	$\mathbf{B}=\begin{bmatrix} 0.0039 & -0.005 \\ -0.0017 & 0.0035 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$
Matrix $\mathbf{C}$	$\mathbf{C}=\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Parameter values of the extended system, control gain, and servo compensator as parameters for step reference input in the robust servo controller are shown in Table 4.4 as follows:

Table 4.4 Parameter values for step reference input in the proposed robust servo controller

Parameter	Value
Matrix $\mathbf{A}_e$	$\mathbf{A}_e = \begin{bmatrix} -0.127 & 0.0003 & 0 & 0 & 0 & 0 \\ 0.0074 & -0.0005 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$
Matrix $\mathbf{B}_e$	$\mathbf{B}_e = \begin{bmatrix} 0.0039 & -0.0050 \\ -0.0017 & 0.0035 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$
Matrix $\mathbf{N}$	$\mathbf{N} = [0]$
Poles	{-11+i,-11-i,-11,-12,-13,-14}
Gain matrix $\mathbf{F}_x$	$\mathbf{F}_x = \begin{bmatrix} 25884 & 36971 & 330528 & 477222 \\ 12689 & 28848 & 161930 & 367067 \end{bmatrix}$
Gain matrix $\mathbf{F}_z$	$\mathbf{F}_z = \begin{bmatrix} 1388635 & 2049383 \\ 677444 & 1553576 \end{bmatrix}$
Matrices of servo compensator	$\mathbf{N}_z = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}, \quad I_\zeta = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

Parameter values of the extended system, control gain, and servo compensator as parameters for ramp reference input in the robust servo controller are shown in Table 4.5 as follows:

Table 4.5 Parameter values for ramp reference input in the proposed robust servo controller

Parameter	Value
Matrix $\mathbf{A}_e$	$\mathbf{A}_e = \begin{bmatrix} -0.127 & 0.0003 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.0074 & -0.0005 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$
Matrix $\mathbf{B}_e$	$\mathbf{B}_e = \begin{bmatrix} 0.0039 & -0.0050 \\ -0.0017 & 0.0035 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$
Matrix $\mathbf{N}$	$\mathbf{N} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}$ for ramp
Poles	{-11+i,-11-i,-11,-12,-13,-14,-15,-16}

Gain matrix $\mathbf{F}_x$	$\mathbf{F}_x = \begin{bmatrix} 32736 & 50704 & 598620 & 100500 \\ 15827 & 40082 & 285930 & 793640 \end{bmatrix}$
Gain matrix $\mathbf{F}_z$	$\mathbf{F}_z = \begin{bmatrix} 14649000 & 4842700 & 28634000 & 8790900 \\ 6820500 & 2284000 & 22570000 & 6935300 \end{bmatrix}$
Matrices of servo compensator	$\mathbf{N}_z = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad I_\zeta = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix}$

The simulation and experiment results are shown in Figs. 4.14 - 4.19. The reference value for the system is shown in Fig. 4.14. The slope of the reference values are related to the linear velocity and angular velocity. The linear velocity of AGV V_r is 0.1 m/s in straight path, and 0.05 m/s in turning path. The angular velocity of AGV ω_r is 0 rad/s in straight line, +0.25 rad/s when the AGV turning to the left and -0.25 rad/s when the AGV turning to the right.

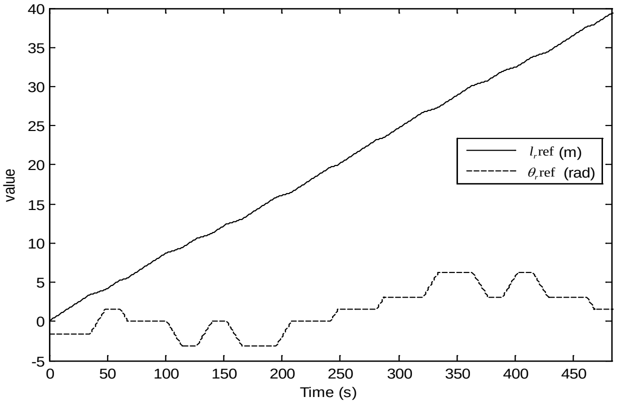


Fig. 4.14 Reference values for the proposed robust servo controller

Fig. 15 shows the disturbance occurred in turning condition. In this dissertation, disturbance related to velocity is assumed as $\varepsilon_1 = 0, \varepsilon_2 = 0$, and disturbance related to linear displacement ε_3 and orientation ε_4 are assumed as step function with amplitude 0.025m and 0.0125 rad, respectively.

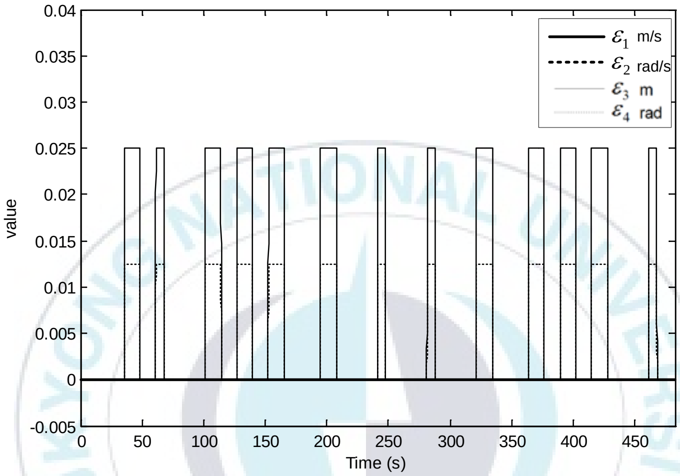


Fig. 4.15 Unmeasureable disturbance occurred in turning condition for the proposed robust servo controller

Fig. 4.16 shows the dynamic control input torque vector $\mathbf{u}$ for the robust servo controller. At the starting points and the end points of turning, the dynamic control input torque vector $\mathbf{u}$ is changed roughly to reduce the error. The dynamic control input torque vector $\mathbf{u}$ are bounded within $\pm 6Nm$.

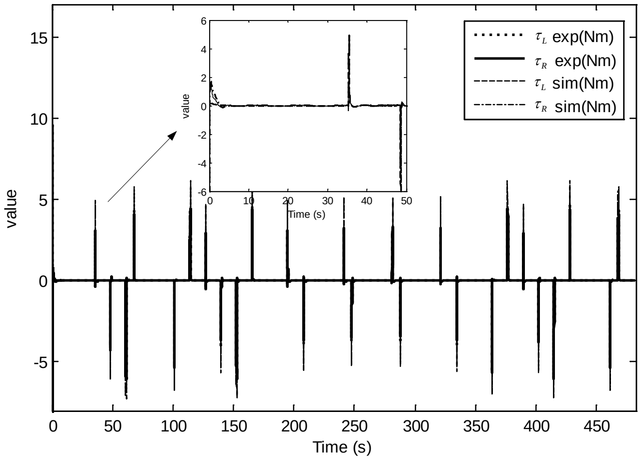


Fig. 4.16 Dynamic control input torque vector **u** for the proposed robust servo controller

Fig. 4.17 shows the planned reference trajectory tracking result obtained from simulation and experiment. At the beginning, the AGV adjusts its posture quickly to reduce the initial error and tracks the reference trajectory successfully.

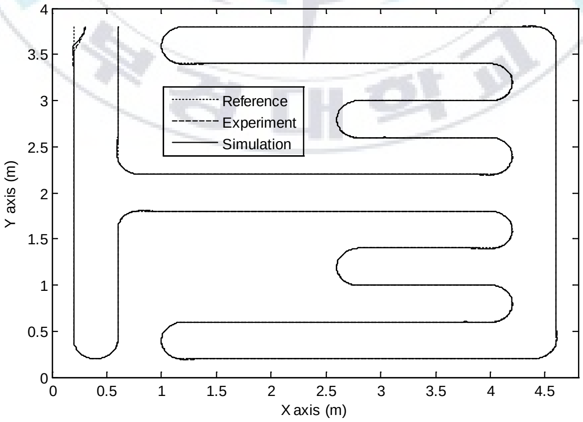


Fig. 4.17 Result for the planned reference trajectory tracking of the proposed robust servo controller

Fig. 4.18 shows the tracking output error vector of AGV $\mathbf{e}_I = \begin{bmatrix} e_l & e_\theta \end{bmatrix}$. It shows the output errors converge to zero after 5 s. At the starting points and the end points of turning, the tracking output error vector $\mathbf{e}_I$ is changed roughly since the reference values are change. The e_l is bounded within $\pm 0.02m$, and e_θ is bounded within $\pm 0.025rad$.

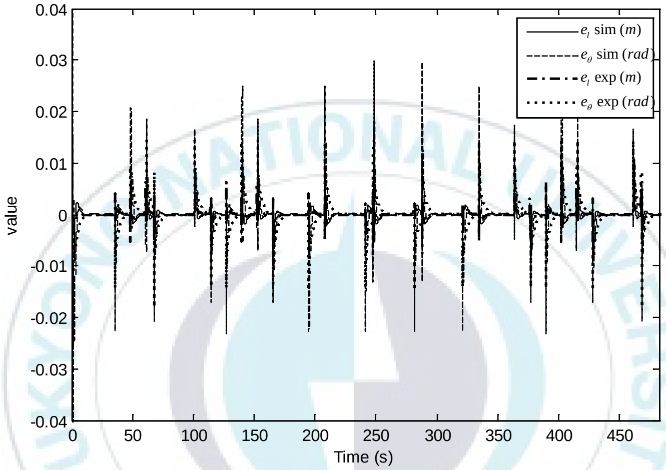


Fig. 4.18 Output error vector $\mathbf{e}_I$ for the proposed robust servo controller

Fig. 4.19 shows the velocity vector of AGV. The linear velocity of AGV V_A is 0.1 m/s in straight path, and V_A is 0.05 m/s in turning path. The angular velocity of AGV ω_A is 0 rad/s in straight line, and ± 0.25 rad/s in turning line.

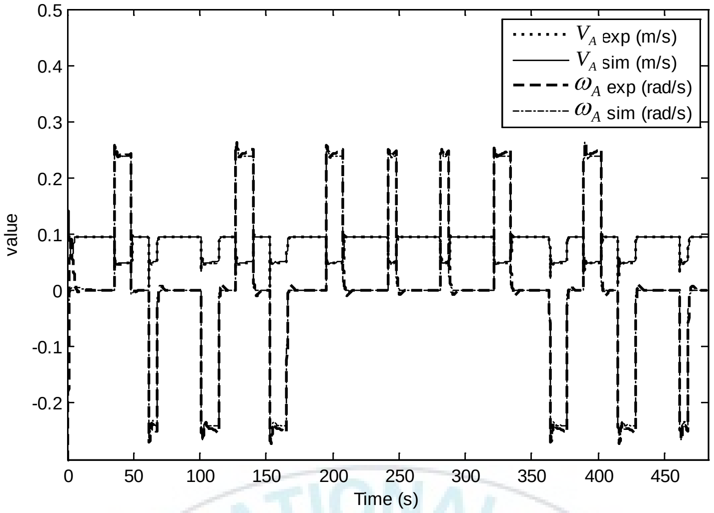


Fig. 4.19 Velocity vector of AGV in the proposed robust servo controller

Fig. 4.20 shows the reference and outputs of AGV for in simulation and experiment in global coordinate for the robust servo controller controller. The AGV linear displacement l_A tracks its reference with ramp type well. The AGV orientation θ_A tracks its reference combined step type and ramp type well.

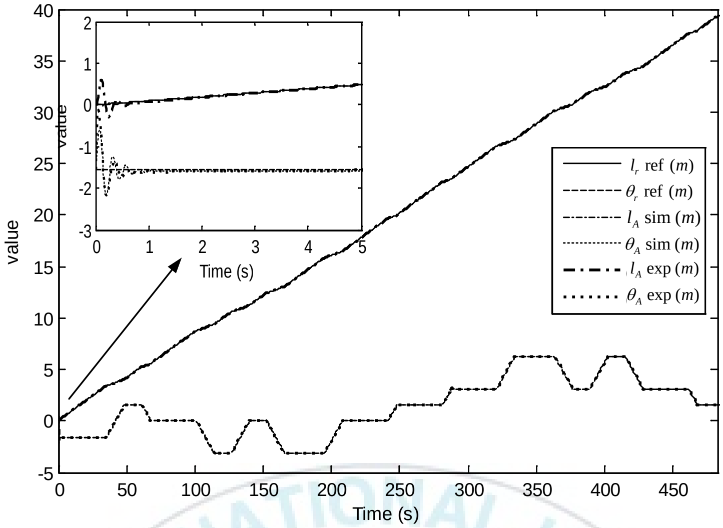


Fig. 4.20 Reference and outputs of AGV for the proposed robust servo controller

4.4 Summary

This chapter proposed two trajectory tracking controllers for an AGV to track the trajectory generated by the proposed path planning algorithm in chapter 3. Firstly, an adaptive kinematic tracking controller based on backstepping method was proposed for AGV with unknown slip to track the planned coverage path. Therefore, an update law to make the AGV adaptive to the unknown slip was proposed. Moreover, a dynamic tracking controller for the AGV with unknown slip to track its reference velocity was proposed. Secondly, a robust servo trajectory controller based on internal model principle was proposed. The extended system applying polynomial differential operator for a given reference and a disturbance to a given system. The controllability checking of the extended system was done. The state feedback law was obtained by a well known regulator design method. Finally, the simulation and experiment results showed that the proposed two tracking controllers made the AGV with unknown slip track the planned path reference well. In the adaptive backstepping controller, the linear velocity of AGV V_A is 0.1 m/s in straight path, and V_A is 0.05 m/s in turning path. The angular velocity of AGV ω_A is 0 rad/s in straight line, and ± 0.25 rad/s in turning line. In the kinematic tracking errors, the e_1 is bounded within $\pm 0.01m$, the e_2 is bounded within $\pm 0.015m$, and the e_3 is bounded within $\pm 0.03rad$. The dynamic tracking wheel angular velocity error vector $\mathbf{e_d}$ is bounded within $\pm 0.06 rad/s$. The update law is bounded within -0.6~0.4 s^{-1}. In the robust servo controller, the control input torque vector $\mathbf{u}$ are bounded within $\pm 6Nm$. The e_l is bounded within $\pm 0.02m$, and e_θ is bounded within $\pm 0.025rad$.

Chapter 5: Sensor Model

In this dissertation, positioning modules consist of encoders, laser scanner LMS-151, and laser navigation system NAV-200. Basically, in common applications, encoder is used to measure the motor velocity, LMS-151 is used for the object detection and avoidance, and only NAV-200 is purely used to measure the AGV position. In previous fault detection algorithms, several fault detection algorithms was designed for each sensor.

In this dissertation, instead of developing several algorithms for each sensor, the information from each sensor are processed to obtain common information such as position, and detect the fault condition by cross checking the result of each sensor. Each sensor has its measurement error. Therefore, to minimize the positioning error, positioning methods are combined using Kalman Filter. However, because the AGV system is nonlinear, Extended Kalman filter is used.

In Extended Kalman filter, process model, process noise covariance and measurement noise covariance are important parameters. Therefore, modeling of each positioning module is presented, and its noise covariance is defined.

In this chapter, the basic principle of each sensor is described. The external and internal conditions that influence in measuring error are explained. The mathematical models of the following three sensors are introduced.

5.1 Encoder

Encoder positioning or well known as dead reckoning (derived from “deduced reckoning” of sailing days) is a simple mathematical procedure for determining the present location of a vehicle by advancing some previous position through known course and velocity information over a given length of time.

The conversion from encoder pulse to AGV position can be obtained using the following procedure. From the left and right wheel encoders pulse increments N_L and N_R are obtained every T sampling interval. The incremental angles for the left wheel $\Delta\phi_l$ and right wheel $\Delta\phi_r$ can be obtained as:

$$\begin{cases} \Delta\phi_l = C_m N_L \\ \Delta\phi_r = C_m N_R \end{cases} \tag{5.1}$$

The conversion factor C_m that translates encoder pulses into wheel rotation angle is defined as:

$$C_m = 1/gC_e \tag{5.2}$$

where C_e is encoder resolution in pulses per revolution, and g is gear ratio of the reduction gear between the motor where the encoder is attached and the drive wheel.

Finally, the mathematical model for encoder positioning is

$$\begin{bmatrix} X_{AE} \\ Y_{AE} \\ \theta_{AE} \end{bmatrix}_{k|k-1} = \begin{bmatrix} X_{AE} \\ Y_{AE} \\ \theta_{AE} \end{bmatrix}_{k-1} + \begin{bmatrix} \cos(\theta_{AE}) & -\sin(\theta_{AE}) & 0 \\ \sin(\theta_{AE}) & \cos(\theta_{AE}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_E \\ 0 \\ \Delta\theta_E \end{bmatrix}_k \tag{5.3}$$

$$\begin{bmatrix} \Delta x_E \\ \Delta \theta_E \end{bmatrix} = \frac{r}{2} \begin{bmatrix} 1 & 1 \\ \frac{1}{b} & -\frac{1}{b} \end{bmatrix} \begin{bmatrix} \Delta \phi_r \\ \Delta \phi_l \end{bmatrix} \tag{5.4}$$

where Δx_E and Δy_E are linear displacements of AGV in local coordinate of AGV, $\Delta \theta_E$ is the change of rotational angle of AGV, subscript k is the current sampling time, subscript $k-1$ is the previous sampling time, and subscript $k|k-1$ is the current time with respect to the previous sampling time. The input for this model is $\mathbf{u}_E = \begin{bmatrix} \Delta \phi_r & \Delta \phi_l \end{bmatrix}^T$ and $\Delta \phi_R$ and $\Delta \phi_L$ are the incremental angle change of right and left wheel.

The encoder positioning for differential drive AGV is based on the accumulation of wheel rotation using simple equations as in Eq. (5.3) ~ (5.4) that are easily implemented. However, the encoder positioning is also based on the assumption that wheel revolutions can be translated into linear displacement relative to the floor. Therefore, small measurement errors will cause drift after passing through the integration.

There are several reasons for inaccuracies in the translation of wheel encoder readings into linear motion. All of these error sources are divided in to two categories: systematic errors and non-systematic errors.

Systematic errors are caused by:

1. Unequal wheel diameters.
2. Average of actual wheel diameters differs from nominal wheel diameter.

3. Actual wheelbase differs from nominal wheelbase.
4. Misalignment of wheels.
5. Finite encoder resolution.
6. Finite encoder sampling rate.

Non-systematic errors are caused by:

1. Travel over uneven floors.
2. Travel over unexpected objects on the floor.
3. Wheel-slippage due to: slippery floors, over acceleration, fast turning (skidding), external forces (interaction with external bodies), internal forces (castor wheels), and nonpoint wheel contact with the floor.

The clear distinction between systematic and non-systematic errors is of great importance for the effective reduction of encoder errors. On most smooth indoor surfaces, systematic errors contribute much more to encoder errors than non-systematic errors. Therefore, only systematic errors are considered in this chapter.

The encoder error of a wheel incurred in one unit of travel is assumed to be randomly distributed with zero mean and to be independent of the encoder error incurred in the next or previous unit of travel. The variance of the cumulative error is then the sum of the variance of each independent segment and is therefore proportional to the distance travelled. The error covariance matrix of encoder $\mathbf{Q}_{\mathbf{E}}$ at each sampling time is expressed as:

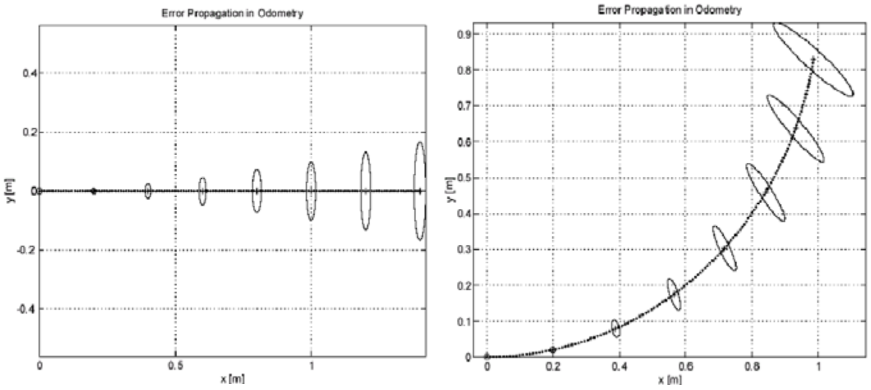
$$\mathbf{Q}_{\mathbf{E}} = \begin{bmatrix} k_r \left|\Delta\phi_r\right| & 0 \\ 0 & k_l \left|\Delta\phi_l\right| \end{bmatrix} \tag{5.5}$$

where $\Delta\phi_R$ and $\Delta\phi_L$ are the change of right and left wheel rotation angle, k_r and k_l is error constants related to incremental encoder obtained from experiment.

In Eq. (5.5) the following assumptions are made:

- The two errors of the individually driven wheels are independent;
- The variance of the errors (left and right wheels) are proportional to the absolute value of the traveled distances

Fig. 5.1 shows typical examples of how the position errors grow with time. The results have been computed using the error model presented above. Fig. 5.1a shows growth of the pose uncertainty for straight-line movement: Note that the uncertainty in y axis grows much faster than in the direction of movement. This results from the integration of the uncertainty about the AGV’s orientation. The ellipses drawn around the AGV positions represent the uncertainties in the x,y direction. Fig. 5.1b shows growth of the pose uncertainty for circular movement with constant turning radius: Again, the uncertainty perpendicular to the movement grows much faster than that in the direction of movement. Note that the main axis of the uncertainty ellipse is not perpendicular to the direction of movement of the AGV.



a. Straigh movement b. Circular movement

Fig. 5.1 Growth of the position uncertainty

5.2 Laser measurement system LMS-151

The basic principle of laser measurement system LMS-151 is shown in Fig. 5.2. The LMS transmits pulsed laser beams using a laser diode. If such a laser pulse is hit at an obstacle, it is reflected at its surface. The reflection is detected by the laser measurement system’s receiver using a photodiode. The distance to the object is calculated from the propagation time (dt) between sending pulses and receiving pulse that the light requires from emission to reception of the sensor.

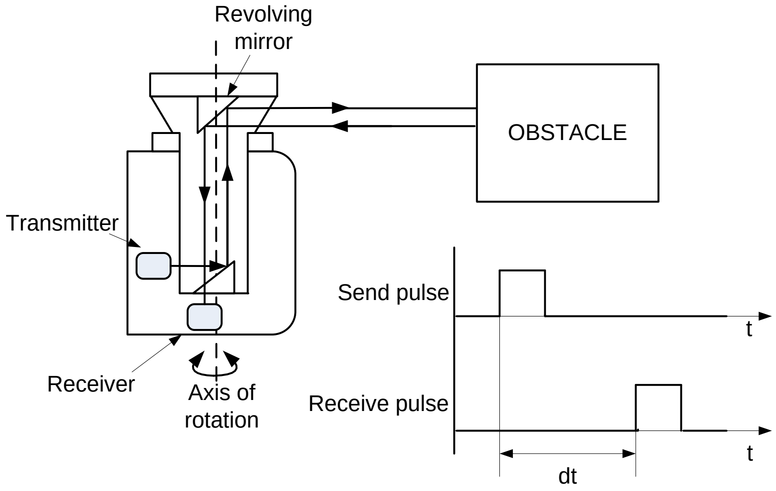


Fig. 5.2 Laser measurement system basic principle

The transmitted laser beam is deflected using a revolving mirror driven by servo motor at scanning frequency of 25 Hz and scans the surroundings in a circular manner as shown in Fig. 5.3. The scanning area defined as W (R_{max}, θ_{start}, θ_{end}) is given by maximum scanning distance R_{max} = 50 m, start scanning angle θ_{start} = -45° and end scanning angle θ_{end} = 225°. The measurements are triggered at regular angular steps using an angular encoder. The LMS scans with angular step ($d\theta$) of 0.5°. The DSP based scanner controller manipulates the operation of device and monitors the status of the scanner. Furthermore, the DSP sends the measurement result using ethernet protocol when the main controller of AGV requests the measurement result from LMS.

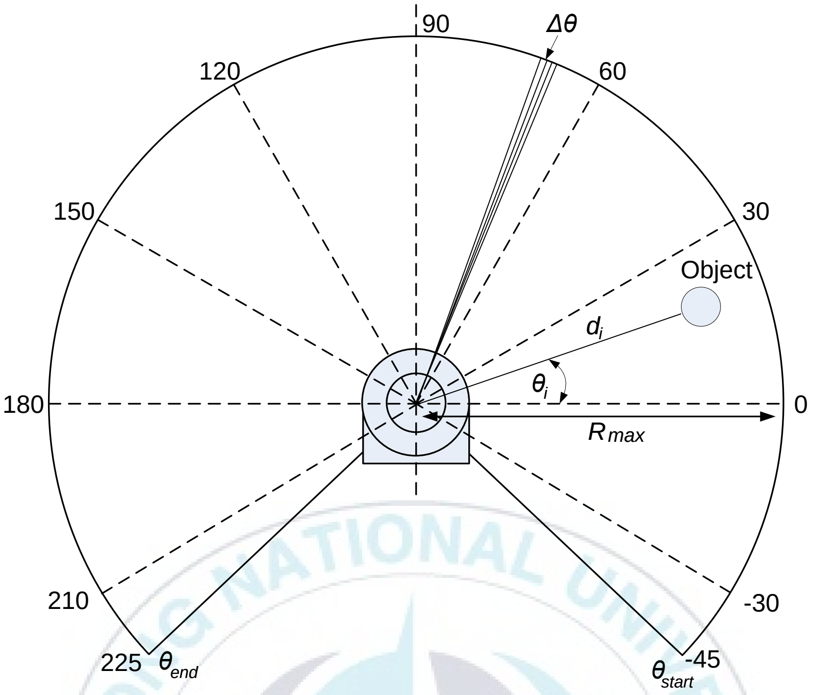
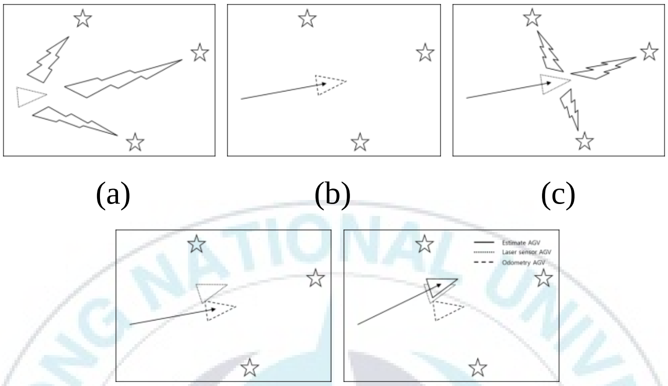


Fig. 5.3 Scanning area of LMS-151

When a single laser beam is reflected by an object as shown in Fig. 5.3, it senses the measurement distance d_i and scanning angle $\theta_{i.}$. The scanning results are expressed in polar coordinate and consist of scanning angle and measurement distance as (d_i , θ_i). The LMS scans the environment with angular step ($\Delta\theta$) of 0.5° and scanning area 270°. Therefore, in single cycle scan, it gets 540 data.

For laser scanner positioning module, Simultaneous Localization and Mapping (SLAM) algorithm is applied based on location of the landmarks as shown in Fig. 5.4. Firstly, landmarks are detected as shown in Fig. 5.4(a). When the encoder data change because the AGV moves, AGV new position is predicted using EKF prediction step based on encoder data as shown in Fig. 5.4(b). Secondly, landmarks are detected from the AGV new position as in Fig. 5.4(c).

The AGV then associates these landmarks to observations of landmarks that are previously observed. Re-observed landmarks are used to update the AGV position using EKF update step as shown in Fig. 5.4(d). The real position, encoder position and estimated position are shown in Fig. 4(e).



(a) (b) (c)

(d) (e)

Fig. 5.4 Laser scanner positioning modeling.

The position estimation will drift because of the integration, matching error, noise and numerical errors being accumulated overtime. The mathematical model of laser scanner positioning module is

$$\begin{bmatrix} X_{AL} \\ Y_{AL} \\ \theta_{AL} \end{bmatrix}_{k|k-1} = \begin{bmatrix} X_{AL} \\ Y_{AL} \\ \theta_{AL} \end{bmatrix}_{k-1} + \begin{bmatrix} \cos(\theta_{AL}) & -\sin(\theta_{AL}) & 0 \\ \sin(\theta_{AL}) & \cos(\theta_{AL}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x_L \\ \Delta y_L \\ \Delta \theta_L \end{bmatrix}_k \qquad (5.6)$$

where the input for this model is $\mathbf{u}_L = \left[\Delta x_L \quad \Delta y_L \quad \Delta\theta_L\right]^T$, Δx_L and Δy_L are linear displacements of AGV in local coordinate of AGV, $\Delta\theta_L$ is the change of rotational angle of AGV.

The process noise covariance noise is

$$\mathbf{Q}_{\mathbf{L}k-1} = \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_\theta \end{bmatrix} \tag{5.7}$$

where k_x, k_y and k_θ are error constants related with the AGV motion obtained from experiment. Eq. (5.6) is a reasonable model because SLAM error is independent from the AGV direction. The error depends on the difference between current scanning result and previous scanning result.

5.3 Laser navigation system NAV-200

The basic principle of NAV-200 installed at the top center of AGV is shown in Fig. 5.5. Basically, the NAV-200 measures the distance in similar way with LMS-151. The NAV200 calculates its own position and orientation based on the fixed reflectors positioned (R1, R2, R3,…, Rn) in the environment, and its coordinates are detected two-dimensionally in a 360° scanning angle using a laser beam invisible to the human eye. *n* is the number of detected reflectors. $\left(X_i, Y_i\right)$ are coordinate of the i^{th} reflector in global coordinate frame. One revolution of the scanner head here is equivalent to a scan and each revolution generates one reading for each detected reflector. The measurement result are distance between sensor and reflector ($d_1, d_2, d_3,\cdots,d_n$) and measurement angle ($\theta_1, \theta_2, \theta_3,\cdots, \theta_n$).

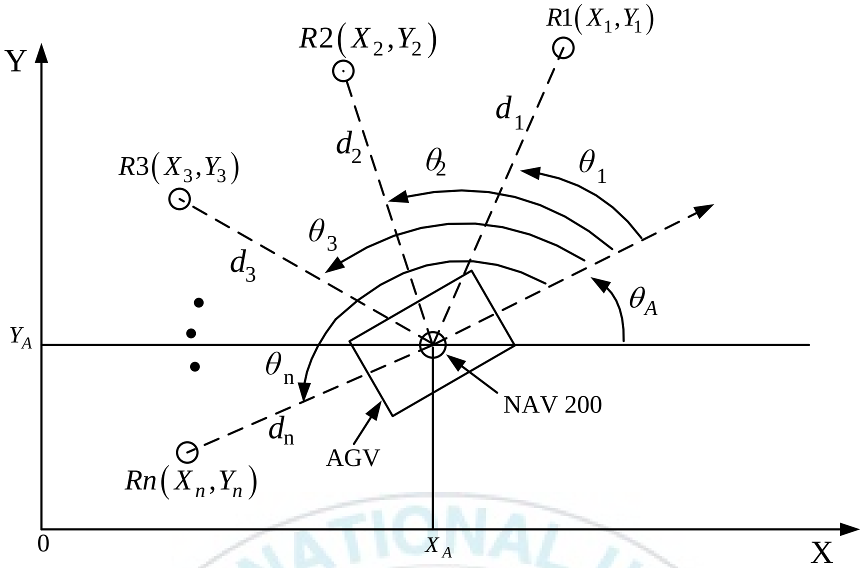


Fig. 5.5 Position measurement of the NAV-200

The coordinates of the reflectors used are stored in the nonvolatile reflector memory of the NAV200 as a reference. Three detected reflectors in a layer are sufficient for position measurement. For position measurement, the NAV200 measures the distances and angles of the reflectors and works out its own position from this data. By comparing the reflector data in the NAV memory and the measurement result, the position and orientation of AGV can be determined.

The coordinate of the AGV (X_A, Y_A) and it's orientation θ_A can be obtained as follows:

$$\begin{aligned} X_A = & \left[1/(m_2 - m_1)(1+m^2)\right] \\ & \{(m_2 m - 1)\times\left[(m_1 + m)X_1 + (m_1 m - 1)Y_1\right] \\ & -(m_1 m - 1)\times\left[(m_2 + m)X_2 + (m_2 m - 1)Y_2\right]\} \end{aligned} \quad (5.8)$$

$$
\begin{aligned}
Y_A = & \left[1/(m_2 - m_1)(1+m^2)\right] \\
& \{(m_1 + m)\times\left[(m_2+m)X_2 + (m_2 m - 1)Y_2\right] \\
& -(m_2+m)\times\left[(m_1+m)X_1 + (m_1 m - 1)Y_1\right]\}
\end{aligned}
\tag{5.9}
$$

$$
\theta_A = \text{Atan2}\left(Y_1 - Y_A, X_1 - X_A\right) - \theta_1 \tag{5.10}
$$

where $m_1 = \tan\theta_1$, $m_2 = \tan\theta_2$, $m_3 = \tan\theta_3$ and $m = \tan\theta_A$. From Eqs. (5.8)~(5.9), m is given by:

$$
m = \frac{(m_3 - m_1)(Y_1 - Y_2 - m_1 X_1 + m_2 X_2) - (m_2 - m_1)(Y_1 - Y_3 - m_1 X_1 + m_3 X_3)}{(m_3 - m_1)(m_1 Y_1 + X_1 - m_2 Y_2 - X_2) - (m_2 - m_1)(m_1 Y_1 + X_1 - m_3 Y_3 - X_3)} \tag{5.11}
$$

Unlike another positioning module, NAV-200 is a position sensor. Therefore, the position data can be obtained directly through serial data communication without any further calculation. However, the position obtained from this sensor also has error caused by measurement noise. Therefore, the mathematical model of NAV-200 positioning model including measurement noise is as follows:

$$
\begin{bmatrix} X_A \\ Y_A \\ \theta_A \end{bmatrix}_{k|k-1} = \begin{bmatrix} X_A \\ Y_A \\ \theta_A \end{bmatrix}_{k} + \begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix}_{k} \tag{5.12}
$$

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

5.4 Summary

In this chapter, the basic principle of each sensor was described. The external and internal conditions that influence in measurement error were explained. The mathematical models of three sensors such as encoder, LMS-151 and NAV-200 were introduced.

Chapter 6: Fault Detection

The schematic diagram of fault detection algorithm is shown in Fig. 6.1. Extended Kalman Filter (EKF) calculates the measurement probability distribution of the AGV position for nonlinear models driven by Gaussian noise. Using the probability distribution of innovation obtained from EKF, it is possible to test if the measured data are fit with the models. When the faults occur, the models will not be valid and the innovation will not be Gaussian and white. The residue calculation estimates the residue from pairwise differences between the estimated positions obtained from positioning modules. Fault isolation is obtained by examining the biggest residue. Finally, from the Table 6.1 as shown in the next section, the fault position is known.

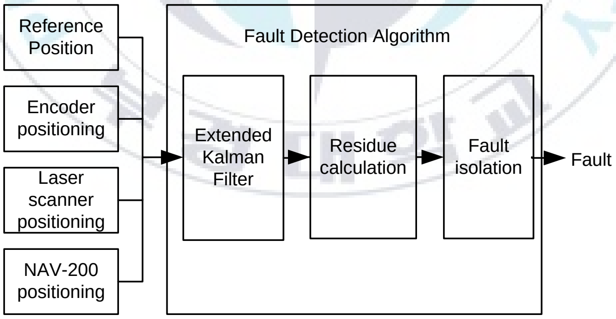


Fig. 6.1 Schematic diagram of fault detection algorithm

The list of fault condition that can be detected by this algorithm is described as:

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

- The fault condition in LMS-151 caused by communication error, power loss and damage on the sensor.
- The fault condition in NAV-200 caused by communication error, power loss, sensor damage and singularity when the AGV moves at the long corridor or open space.
- The fault condition in actuator caused by inverter damage, motor damage, and communication error.

6.1 Extended Kalman filter

Firstly, pairwise of positioning modules, as shown in Table 6.1, are tested using Kalman filter. Kalman filter theory provides a method to calculate measurement probability for a linear model driven by Gaussian noise. Since the proposed system is nonlinear, an Extended Kalman Filter (EKF) is used.

In the EKF, the state transition and observation models is not a linear functions of the state but may be differentiable functions as follows:

$$\begin{aligned} \mathbf{x}_k &= f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \\ \mathbf{z}_k &= h(\mathbf{x}_k) + \mathbf{v}_k \end{aligned} \tag{6.1}$$

where $\mathbf{x}_k$ is a state vector at time k, $\mathbf{w}_{k-1}$ is process noise vector and $\mathbf{v}_k$ is an observation noise vector which are assumed to be zero mean multivariate Gaussian noises with covariance $\mathbf{Q}_k$ and $\mathbf{R}_k$ at time k, respectively, $f(\cdot)$ is a nonlinear process vector differentiable function at time k, $h(\cdot)$ is nonlinear observation vector differentiable function at time k, and $\mathbf{z}_k$ is an output vector at time k.

In this dissertation, $f(\cdot)$ in the state function can be obtained from Eq. (2.19) for reference speed, Eq. (5.3), for encoder positioning module, and Eq. (5.6) for laser scanner positioning module.

For the measurement model, $\mathbf{z}_k$ is described as the following equation:

$$\mathbf{z}_k = \mathbf{x}_k + \mathbf{v}_k \tag{6.2}$$

EKF consists of two steps as follows:

Prediction step:

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= f(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}) \\ \mathbf{P}_{k|k-1} &= \mathbf{F}_{k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k-1}^T + \mathbf{W}_k\mathbf{Q}_{k-1}\mathbf{W}_k^T \end{aligned} \tag{6.3}$$

where $\mathbf{F}_{k-1} = \left.\dfrac{\partial f}{\partial \mathbf{x}}\right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}}$ and $\mathbf{W}_k = \left.\dfrac{\partial f}{\partial \mathbf{u}}\right|_{\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_{k-1}}$

where $\hat{\mathbf{x}}_{k|k-1}$ and $\mathbf{P}_{k|k-1}$ are the predicted state estimation and predicted covariance matrix estimation at time k obtained from the measurement data at time $k-1$, respectively.

Update step:

$$\begin{aligned} \tilde{\mathbf{y}}_k &= \mathbf{z}_k - h(\hat{\mathbf{x}}_{k|k-1}) \\ \mathbf{S}_k &= \mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_k \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k^T\left(\mathbf{S}_k\right)^{-1} \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k\tilde{\mathbf{y}}_k \\ \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1} \end{aligned} \tag{6.4}$$

where $\mathbf{H}_k = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}_{k|k-1}}$

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

6.2 Residue calculation

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

$$s_t = \tilde{y}_k^T S_k^{-1} \tilde{y}_k \tag{6.5}$$

where s_t is the residue at time k.

If the value of s_t is larger than threshold value Th, one of the pairwise sensors is fault as follows:

$$Fault\ condition \begin{cases} normal \quad 0 \quad if \quad s_t < Th \\ fault \qquad X \quad if \quad s_t > Th \end{cases} \tag{6.6}$$

The threshold value is related with the sensitivity of the system to the fault.

6.3 Fault isolation

Finally, to isolate the fault, 4 test conditions has been setup as shown in Table 6.1. The residue that is more than threshold value is examined, which is then matched at the table relation between residues and faults. Several fault conditions are considered as follows:

1. In normal condition, all positioning modules provide a correct position. Therefore, all tests are normal since all residues are less than threshold value.
2. When the encoder is broken, all tests related with encoder such as test T2 and test T3 have the fault results since the positioning from encoder is false and the resulting residue from both tests T2 and T3 are greater than threshold value.
3. When the BLDC motor is broken, all tests related with reference speed such as test T1 and test T3 have the fault results since the positioning from reference speed is false and the resulting residue from both tests T1 and T3 are greater than threshold value.
4. When laser scanner is broken, all tests related with laser scanner such as test T1 and test T2 have the fault result since positioning from laser scanner is false. The resulting residue from both tests T1 and T3 are greater than threshold value.

5. When NAV-200 is broken, the NAV would not provide the correct position. Therefore, the residue from test T4 is greater than the threshold value.

Table 6.1 Relation between residues and faults

Test		Fault				
		N	*F1*	*F2*	*F3*	*F4*
		Normal	Encoder	BLDC motor	Laser scanner	NAV
T1	Reference speed + laser scanner (LMS)	0	0	X	X	0
T2	Encoder + laser scanner (LMS)	0	X	0	X	0
T3	Reference speed + encoder	0	X	X	0	0
T4	Reference speed + NAV	0	0	0	0	X

where 0 means normal, and X means fault.

6.4 Experimental results

To verify the effectiveness of the proposed fault detection algorithm, several experiments has been conducted. During the experiments, the AGV follows the given trajectory as shown at Fig. 6.2. The experimental environment is corridor surrounded by wall and windows. The algorithm was run in realtime on industrial computer inside the AGV. In the experiment, the sensor outputs are observed at 10Hz. The fault data and any other information are saved inside industrial computer during experiment and can be copied after the experiment is finished.

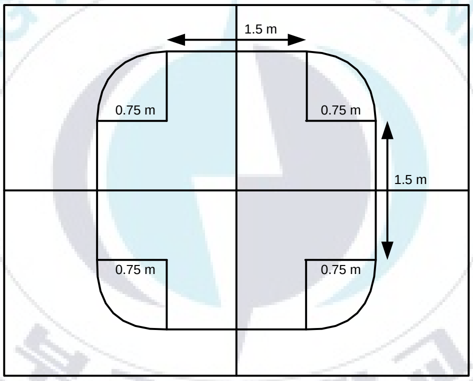


Fig. 6.2 Trajectory for experiment

The proposed algorithm is tested during several conditions T1, T2, T3, and T4 as shown in Table 6.1. During the experiment, the trajectories from test conditions such as reference speed + laser scanner (LMS), Encoder + laser scanner (LMS), Encoder + reference velocity, and reference velocity + NAV 200 can be observed. The residue of each test also can be obtained and it is compared with the threshold value Th. In these experiments, the threshold value Th is chosen as 500.

6.4.1 Normal condition

The experimental results for normal condition *N* in Table 6.1 are shown in Fig. 6.3 and Fig. 6.4. Fig. 6.3 shows that trajectory tracking in several conditions T1, T2, T3, and T4 obtained from each positioning module fused by EKF are almost similar since fault condition doesn't exist. Therefore, the residues values in all test conditions shown in Fig. 6.4 are less than threshold value Th.

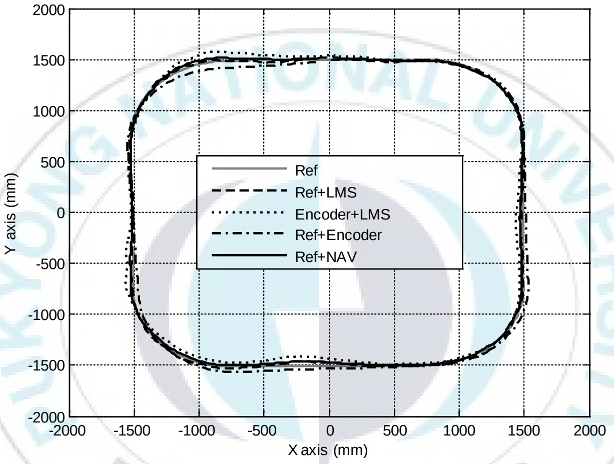


Fig. 6.3 Trajectory tracking in normal condition

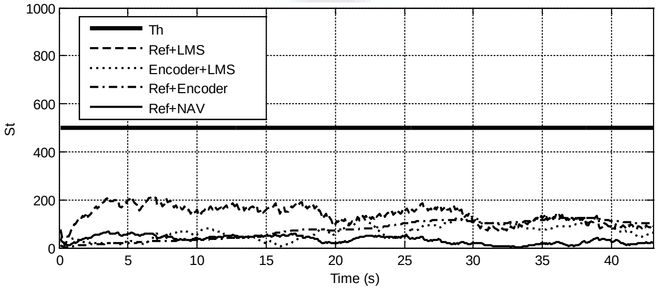


Fig. 6.4 Residue in normal condition

6.4.2 Encoder fault

The experimental results for encoder fault *F1* in Table 6.1 are shown in Fig. 6.5 and Fig. 6.6. To resemble the encoder fault condition, at t = 20 s, the controller stops receiving the data from encoders. Therefore, the encoder positioning modules don't provide the positioning data anymore. Fig. 6.5 shows the trajectory tracking when the encoder fault happens. In Fig. 6.6, the residue values of encoder + laser scanner (LMS) and reference speed + encoder are increased over the threshold value at t = 21.6 s and t = 22.2 s, respectively. From the Table 6.1, it is easily can be observe that the encoder fault happens.

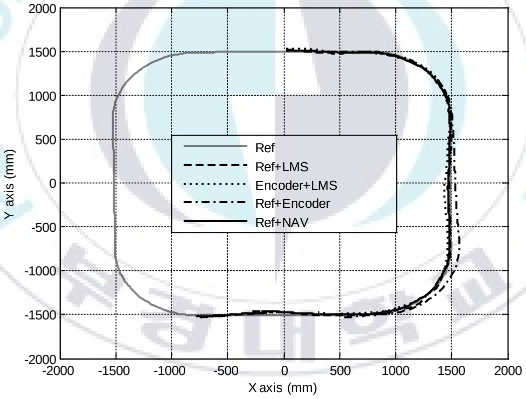


Fig. 6.5 Trajectory tracking when encoder fault occurs

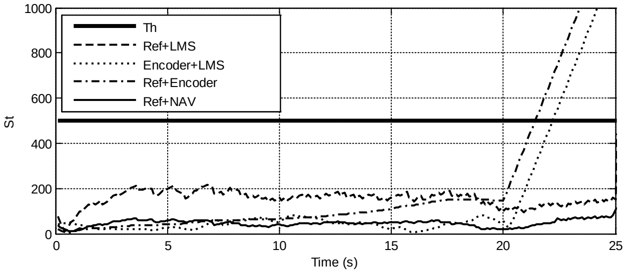


Fig. 6.6 Residue when encoder fault occurs

6.4.3 BLDC motor fault

The experimental results for BLDC motor fault *F2* in Table 6.1 are shown in Fig. 6.7 and Fig. 6.8. The fault condition can be obtained by turning off the motors when the AGV moves.

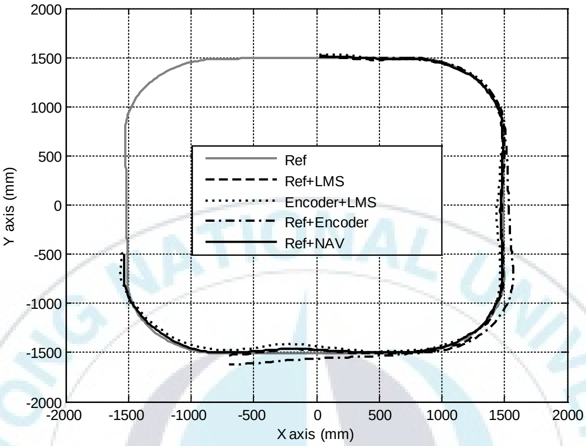


Fig. 6.7 Trajectory tracking when BLDC motor fault occurs

Fig. 6.8 shows that since the motor stops at t = 25 s, the residue values of Reference speed + laser scanner (LMS) and Reference speed + encoder are increased over the threshold value at about t = 26.2 s. From Table 6.1, it is easily can be observed that the BLDC motor fault happens.

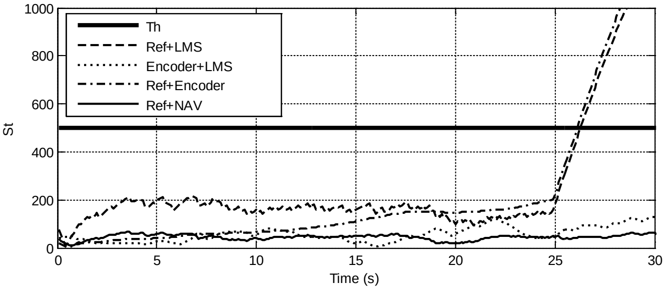


Fig. 6.8 Residue when BLDC motor fault occurs

6.4.4 Laser scanner fault

The experimental results for laser scanner (LMS) fault *F3* in Table 6.1 are shown in Fig. 6.9 and Fig. 6.10. The fault condition can be obtained by pulling of the power connection of the sensor when the AGV moves at t = 30 s.

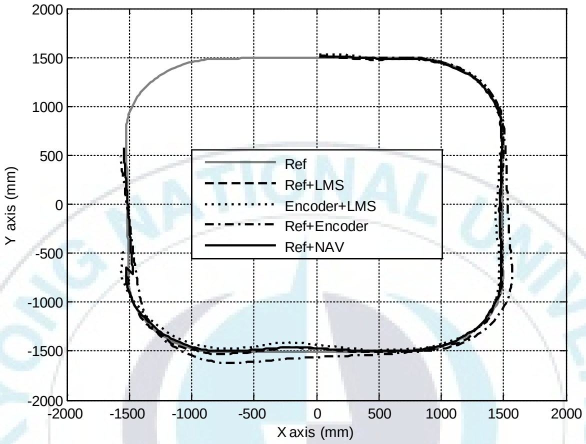


Fig. 6.9 Trajectory tracking when laser scanner fault occurs

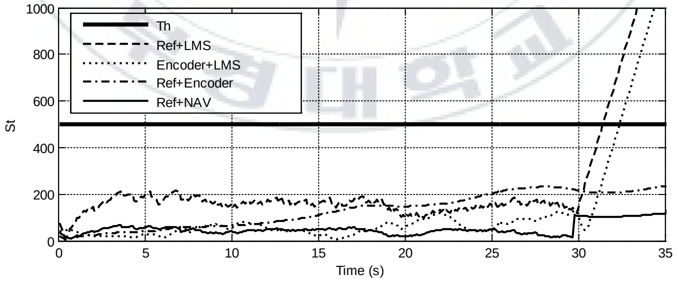


Fig. 6.10 Residue when laser scanner fault occurs

Since the laser scanner doesn’t provide any data, the SLAM positioning cannot provide a new position data. Therefore, as shown in Fig. 6.10, the residue values of Reference speed + laser scanner

(LMS) and Encoder + laser scanner (LMS) are over the threshold value at t = 31.2 s and t = 32.3 s, respectively.

6.4.5 NAV fault

The experimental results for NAV fault *F4* in Table 6.1 are shown in Fig. 6.11 and Fig. 6.12. The fault condition can be obtained by pulling of the power connection of the sensor when the AGV moves at t = 35 s. Since the NAV sensor doesn't provide any data, as shown in Fig. 6.12, the residue value of reference + NAV is increased and over the threshold value at t = 36.1 s.

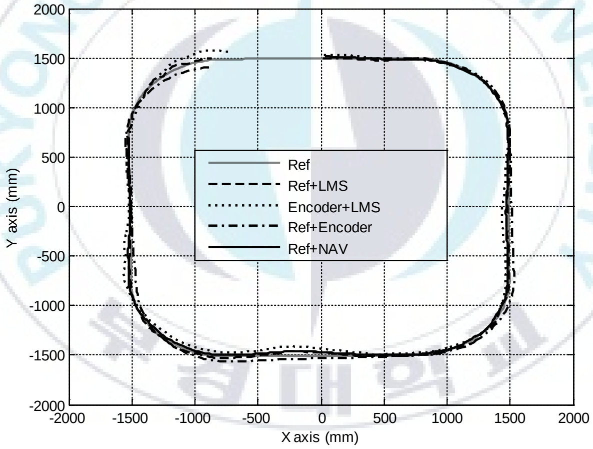


Fig. 6.11 Trajectory tracking when NAV-200 fault occurs

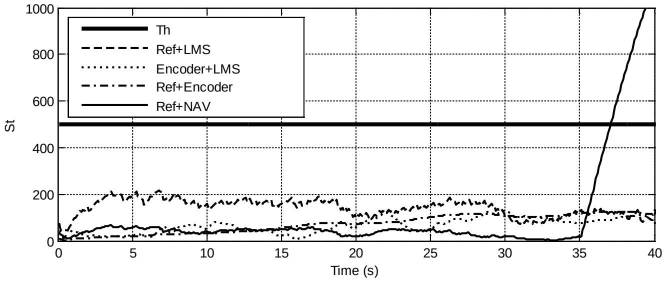


Fig. 6.12 Residue when NAV-200 fault occurs

6.5 Summary

This chapter proposed implementation and experimental validation of fault detection algorithm for Automatic Guided Vehicle (AGV) system based on multiple positioning modules. Experimental results in normal condition and fault conditions showed that the proposed algorithm successfully detected the fault conditions. The algorithm could calculate the residue of each pairs positioning modules based on extended Kalman filter. Based on those residue values, the fault condition was detected. 5 test conditions has been setup to verify the proposed algorithm for normal condition, encoder broken fault, BLDC motor fault, laser scanner fault, and NAV fault. In the simulation and experiments, the threshold value Th is chosen as 500. In normal condition, residues values in all test conditions shown less than threshold value Th. When encoder fault at t=20s, the residue values of encoder + laser scanner (LMS) of test 1 and reference speed + encoder of test 2 are increased over the threshold value at t = 21.6 s and t = 22.2 s, respectively. When the BLDC motor fault at t = 25 s, the residue values of reference speed + laser scanner (LMS) of test 1 and reference speed + encoder of test 3 are increased over the threshold value at about t = 26.2 s. When the laser scanner fault at t=30s, the residue values of test 1 and encoder + laser scanner (LMS) of test 2 are over the threshold value at t = 31.2 s and t = 32.3 s, respectively. When the NAV fault at t= 35s, the residue value of reference + NAV of test 4 is increased and over the threshold value at t = 36.1 s.

Chapter 7: Conclusions and Future Works

7.1 Conclusions

This dissertation proposed a path planning algorithm to cover a given working area, a controller design method to track the path generated by the path planning algorithm and fault detection algorithm based on multi positioning module for differential drive Automatic Guided Vehicle (AGV) system. The conclusions of this dissertation are summarized as follows:

- AGV's mechanical and electrical system design was proposed. The mechanical design consisted of body frame and wheel configuration. Electrical system consisted of sensors, controller, and actuator. Kinematic and dynamic models of a differential drive system were considered. Based on the sliding constraint and rolling constraint of fixed standard driving wheels and castor wheels, the kinematic modeling of a differential drive AGV was obtained. The dynamic equation of the AGV was derived using the Lagrange formula.
- Path planning researches focused in finding a path to visit all the points in a given area, and minimizing overlapped paths, journey time and turning number. To solve this problem, this dissertation proposed a new path planning algorithm using a multi-spanning tree method. To do this, an occupancy grid map was used to represent an obstacle position in the environment. Secondly, Morse cell decomposition method was applied to divide the whole area of work space into cells in the vertical and horizontal

directions. To minimize the turning number, a minimal sum of altitude method was applied to the combination of vertical and horizontal cell decompositions. Thirdly, a spanning tree algorithm was applied to smaller work area and to each cell. Fourthly, the smoothing algorithm was applied to smoothing the turning path. Finally, the simulation results showed that the proposed algorithm generated the shorter coverage path without overlapping path, had less turning number and had faster coverage time than the conventional spanning tree, vertical and horizontal cell decomposition methods.

- Two trajectory tracking controllers for an AGV to track the reference trajectory planned by the proposed coverage path planning algorithm in chapter 3 were proposed. Firstly, an adaptive kinematic tracking controller based on backstepping method was proposed for the AGV with unknown slip parameter to track the planned coverage path. An update law to make the AGV adaptive to the unknown slip was proposed. Moreover, a dynamic tracking controller for the AGV with unknown slip to track its reference velocity given as a kinematic tracking control law was proposed. A second trajectory tracking controller is a robust trajectory servo controller obtained by modifying the concept of the internal model principle. By operating a polynomial differential operator for the state space model with disturbance and error signal, its extended system was obtained. The servo controller for the extended system could be easily designed by a well known regulator design method using pole assignment. Finally, the simulation and experiment results showed that the proposed two tracking controllers made the AGV with unknown or unmeasurable disturbance track the planned reference path. In adaptive backstepping controller, the linear velocity of AGV V_A with unknown slip parameter was 0.1 m/s in straight path, and V_A was 0.05 m/s in turning path. The angular velocity of AGV ω_A was 0 rad/s in straight line, and ± 0.25 rad/s in turning line. In kinematic tracking errors, the e_1 was bounded within $\pm 0.01m$, the e_2 was bounded within $\pm 0.015m$, and the e_3 was bounded within $\pm 0.03rad$. The dynamic tracking wheel angular velocity error vector $\mathbf{e_d}$ was bounded within $\pm 0.06 rad / s$. The update law was bounded within -0.6~$0.4\, s^{-1}$. In the robust servo controller, the dynamic control input torque vector $\mathbf{u}$ were bounded within $\pm 6Nm$. The disturbance related to the velocities were assumed as $\varepsilon_1 = 0, \varepsilon_2 = 0$, and disturbance related to linear displacement ε_3 and disturbance related to orientation ε_4 were assumed as step function with amplitude 0.025m and 0.0125 rad, respectively. The slope of the reference values were related to the linear velocity and angular velocity. The linear velocity of AGV V_r was 0.1 m/s in straight path, and 0.05 m/s in turning path. The angular velocity of AGV ω_r was 0 rad/s in straight line, +0.25 rad/s when the AGV turning to the left and -0.25 rad/s when the AGV turning to the right. The e_l was bounded within $\pm 0.02m$, and e_θ was bounded within $\pm 0.025rad$.

- The basic principle of each sensor was described. The positioning sensors used in this dissertation were encoder, laser scanner, and NAV-200. The external and internal conditions that influence in measurement error were explained. The mathematical models of positioning sensor modules were introduced.

- Implementation and experimental validation of fault detection algorithm for Automatic Guided Vehicle (AGV) system based on multiple positioning modules was done. Experimental results in normal condition and fault condition showed that the proposed fault detection algorithm successfully detected the fault condition. The fault detection algorithm could calculate the residue of each pairs positioning modules based on the extended Kalman filter. Based on those residue values, the fault condition was detected. 5 test conditions has been setup to verify the proposed algorithm for normal condition, encoder broken fault, BLDC motor fault, laser scanner fault, and NAV fault. In the simulation and experiments, the threshold value Th was chosen as 500. In normal condition, residues values in all test conditions shown less than threshold value Th. When encoder fault at t=20s, the residue values of encoder + laser scanner (LMS) of test 1 and reference speed + encoder of test 2 are increased over the threshold value at t = 21.6 s and t = 22.2 s, respectively. When the BLDC motor fault at t = 25 s, the residue values of reference speed + laser scanner (LMS) of test 1 and reference speed + encoder of test 3 are increased over the threshold value at about t = 26.2 s. When the laser scanner fault at t=30s, the residue values of test 1 and encoder + laser scanner (LMS) of test 2 are over the threshold value at t = 31.2 s and t = 32.3 s, respectively. When the NAV fault at t= 35s, the residue value of reference + NAV of test 4 was increased and over the threshold value at t = 36.1 s.

7.2 Future works

- For the path planning algorithm future plan, since the grid resolution of grid map was similar with the AGV width, the existence of small obstacle could not be confirmed. Therefore, an obstacle avoidance algorithm should be employed in practical application.
- Further research in nonlinear systems would possibly provide additional control methodologies for the AGV. Further work should be done to evaluate the robustness of the system.
- Hardware integration was a major implementation issue. Evaluating actual hardware and getting to work together would be a major challenge. Also, sampling time issues would have to be considered since data from each sensor would be received at different rates.
- Cost versus performance, as in any engineering design, would have significant impact on an actual commercial implementation of an AGV.

References

[1] S. A. Kumar and N. Suresh, *Production and Operations Management*, 2nd editio. New Delhi: New age international Ltd., 2006.

[2] R. K. Duhan and P. Gahlot, “The Study of Production Output of Flexible Manufacturing Cells with AGV and Simulation with SIMAN Language,” *International Journal of Engineering Sciences*, Vol. 1, No. 2, pp. 44–47, 2012.

[3] F. L. Almeida, B. M. Terra, P. A. Dias, and G. M. Gonçalves, “Transport with Automatic Guided Vehicles in the Factory of the Future,” Proceeding of *The 15th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2010*, 2010.

[4] D. Herrero-Perez and H. Martinez-Barbera, “Modeling Distributed Transportation Systems Composed of Flexible Automated Guided Vehicles in Flexible Manufacturing Systems,” *IEEE Transactions on Industrial Informatics*, Vol. 6, No. 2, pp. 166–180, 2010.

[5] A. G. Özkil, S. Dawids, Z. Fan, and T. Sørensen, “Design of a Robotic Automation System for Transportation of Goods in Hospitals,” Proceeding of *The 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation, CIRA 2007*, pp. 392–397, 2007.

[6] A. G. Ozkil, Z. Fan, S. Dwids, H. Aanaes, J. K. Kristensen, and K. H. Christensen, “Service Robots for Hospitals: A Case Study of Transportation Tasks in a Hospital,” Proceeding of

The 2009 IEEE International Conference on Automation and Logistics, ICAL 2009, pp. 289–294, 2009.

[7] F. Carreira, T. Canas, A. Silva, and C. Cardeira, “I-Merc: A Mobile Robot to Deliver Meals inside Health Services,” Proceeding of *2006 IEEE Conference on Robotics, Automation and Mechatronics*, 2006.

[8] G. Belforte, R. Deboli, P. Gay, P. Piccarolo, and D. Ricauda Aimonino, “Robot Design and Testing for Greenhouse Applications,” *Biosystems Engineering*, Vol. 95, No. 3, pp. 309–321, 2006.

[9] D. L. Peterson, B. S. Bennedsen, W. C. Anger, and S. D. Wolford, “A Systems Approach to Robotic Bulk Harvesting of Apples,” *Transactions of the ASAE*, Vol. 42, No. 4, pp. 871–876, 1999.

[10] T. Chateau, C. Debain, F. Collange, L. Trassoudaine, and J. Alizon, “Automatic Guidance of Agricultural Vehicles Using a Laser Sensor,” *Computers and Electronics in Agriculture*, Vol. 28, No. 3, pp. 243–257, 2000.

[11] H. Rashidi and E. P. K. Tsang, “Novel Constraints Satisfaction Models for Optimization Problems in Container Terminals,” *Applied Mathematical Modelling*, Vol. 37, No. 6, pp. 3601–3634, 2013.

[12] A. Barrientos, J. Colorado, J. Del Cerro, A. Martinez, C. Rossi, D. Sanz, and J. Valente, “Aerial Remote Sensing in Agriculture: A Practical Approach to Area Coverage and Path Planning for Fleets of Mini Aerial Robots,” *Journal of Field Robotics*, Vol. 28, No. 5, pp. 667–689, 2011.

[13] E. Santamaria, F. Segor, I. Tchouchenkov, and R. Schönbein, "Path Planning for Rapid Aerial Mapping with Unmanned Aircraft Systems a Fast Complete Coverage Path Planning Algorithm with No Backtracking," Proceeding of *ICONS 2013 : The Eighth International Conference on Systems*, pp. 82–87, 2013.

[14] J. Jin and L. Tang, "Coverage Path Planning on Three-Dimensional Terrain for Arable Farming," *Journal of Field Robotics*, Vol. 28, No. 3, pp. 424–440, 2011.

[15] T. Oksanen and A. Visala, "Coverage Path Planning Algorithms for Agricultural Field Machines," *Journal of Field Robotics*, Vol. 26, No. 8, pp. 651–668, 2009.

[16] T. Pilarski, M. Happold, H. Pangels, M. Ollis, K. Fitzpatrick, and A. Stentz, "The Demeter System for Automated Harvesting," *Autonomous Robots*, Vol. 13, No. 1, pp. 9–20, 2002.

[17] P. M. Hsu and C. L. Lin, "Optimal Planner for Lawn Mowers," Proceeding of *IEEE 9th International Conference on Cybernetic Intelligent Systems, CIS 2010*, pp. 1–7, 2010.

[18] T. Palleja, M. Tresanchez, M. Teixido, and J. Palacin, "Modeling Floor-Cleaning Coverage Performances of Some Domestic Mobile Robots in a Reduced Scenario," *Robotics and Autonomous Systems*, Vol. 58, No. 1, pp. 37–45, 2010.

[19] G. Cabrita, B. D. Gouveia, D. C. Santos, and J. a Prado, "An Autonomous All Terrain Robotic System for Field Demining Missions," *Robotics and Autonomous Systems*, pp. 1–19, 2015.

[20] E. U. Acar, H. Choset, Y. Zhang, and M. Schervish, "Path Planning for Robotic Demining: Robust Sensor-Based

Coverage of Unstructured Environments and Probabilistic Methods," *The International Journal of Robotics Research*, Vol. 22, No. 7–8, pp. 441–466, 2003.

[21] P. N. Atkar, "Uniform Coverage of Automotive Surface Patches," *The International Journal of Robotics Research*, Vol. 24, No. 11, pp. 883–898, 2005.

[22] J. Zhu, D. Sun, and S. K. Tso, "Application of a Service Climbing Robot with Motion Planning and Visual Sensing," *Journal of Robotic Systems*, Vol. 20, No. 4, pp. 189–199, 2003.

[23] A. Marino and G. Antonelli, "Experiments on Sampling/ Patrolling with Two Autonomous Underwater Vehicles," *Robotics and Autonomous Systems*, Vol. 67, pp. 61–71, 2015.

[24] S. Sariel, T. Balch, and N. Erdogan, "Naval Mine Countermeasure Missions," *IEEE Robotics and Automation Magazine*, Vol. 15, No. 1, pp. 45–52, 2008.

[25] E. Galceran, R. Campos, M. Carreras, and P. Ridao, "3D Coverage Path Planning with Realtime Replanning for Inspection of Underwater Structures," *Journal of Field Robotics*, Vol. 1, pp. 1–32, 2014.

[26] J. Chung, "Deep-Ocean Mining Technology III: Developments," Proceeding of *The Eighth (2009) ISOPE Ocean Mining Symposium*, pp. 1–7, 2009.

[27] Z. Wang, S. Liu, L. Li, B. Yuan, and G. Wang, "Dynamic Simulation of COMRA's Self-Propelled Vehicle for Deep Ocean Mining System," Proceeding of *The Fifth Ocean Mining Symposium*, pp. 112–120, 2003.

[28] K. Herzog, E. Schulte, M. A. Atmanand, and W. Schwarz, "Slip Control System for a Deep-Sea Mining Machine," *IEEE*

Transactions on Automation Science and Engineering, Vol. 4, No. 2, pp. 282–286, 2007.

[29] S. M. Yoon, S. Hong, S. J. Park, J. S. Choi, H. woo Kim, and T. K. Yeu, “Track Velocity Control of Crawler Type Underwater Mining Robot through Shallow-Water Test,” *Journal of Mechanical Science and Technology*, Vol. 26, No. 10, pp. 3291–3298, 2012.

[30] E. Galceran and M. Carreras, “A Survey on Coverage Path Planning for Robotics,” *Robotics and Autonomous Systems*, Vol. 61, No. 12, pp. 1258–1276, 2013.

[31] H. Choset, “Coverage of Known Spaces: The Boustrophedon Cellular Decomposition,” *Autonomous Robots*, Vol. 9, pp. 247–253, 2000.

[32] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, “Morse Decompositions for Coverage Tasks,” *The International Journal of Robotics Research*, Vol. 21, No. 4, pp. 331–344, 2002.

[33] E. Garcia and P. G. De Santos, “Mobile-Robot Navigation with Complete Coverage of Unstructured Environments,” *Robotics and Autonomous Systems*, Vol. 46, No. 4, pp. 195–204, 2004.

[34] R. Tarjan, “Depth-First Search and Linear Graph Algorithms,” Proceeding of *12th Annual Symposium on Switching and Automata Theory*, Vol. 1, No. 2, pp. 146–160, 1971.

[35] P. A. Jimenez, B. Shirinzadeh, A. Nicholson, and G. Alici, “Optimal Area Covering Using Genetic Algorithms,” Proceeding of *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 1–4, 2007.

[36] A. Bakhtiari, H. Navid, J. Mehri, and D. D. Bochtis, “Optimal Route Planning of Agricultural Field Operations Using Ant Colony Optimization,” *Agricultural Engineering International: CIGR Journal*, Vol. 13, No. 4, pp. 1–16, 2011.

[37] P. R. Srivastava, K. Baby, and G. Raghurama, “An Approach of Optimal Path Generation Using Ant Colony Optimization,” Proceeding of *TENCON 2009 - IEEE Region 10 Conference*, pp. 1–6, 2009.

[38] A. Zelinsky, R. A. Jarvis, J. C. Byrne, and S. Yuta, “Planning Paths of Complete Coverage of an Unstructured Environment by a Mobile Robot,” Proceeding of *International Conference on Advanced Robotics*, pp. 533–538, 1993.

[39] N. Hazon and G. A. Kaminka, “On Redundancy, Efficiency, and Robustness in Coverage for Multiple Robots,” *Robotics and Autonomous Systems*, Vol. 56, No. 12, pp. 1102–1114, 2008.

[40] Y. Gabriely and E. Rimon, “Spanning-Tree Based Coverage of Continuous Areas by a Mobile Robot,” *Annals of Mathematics and Artificial Intelligence*, Vol. 31, No. 1–4, pp. 77–98, 2001.

[41] S. M. LaValle, *Planning Algorithms*. 2006.

[42] R. C. Prim, “Shortest Connection Networks and Some Generalizations,” *Bell System Technical Journal*, Vol. 36, No. 6, pp. 1389–1401, 1957.

[43] L. Armesto and J. Tornero, “Automation of Industrial Vehicles: A Vision-Based Line Tracking Application,” Proceeding of *ETFA 2009 - 2009 IEEE Conference on Emerging Technologies and Factory Automation*, 2009.

[44] C. R. Kelber, R. S. Dreger, J. Schirmbeck, and D. A. Borges, “Nonlinear Steering Control Strategy for an Optical Stripe Tracker,” Proceeding of *7th International Workshop on Advanced Motion Control. Proceedings (Cat. No.02TH8623)*, pp. 546–550, 2002.

[45] C. Chen, B. Wang, and Q. T. Ye, “Application of Automated Guided Vehicle (AGV) Based on Inductive Guidance for Newsprint Rolls Transportation System,” *Journal of Donghua University*, Vol. 21, No. 2, pp. 88–92, 2004.

[46] T. Fujimoto, J. Ota, T. Arai, T. Ueyama, and T. Nishiyama, “Semi-Guided Navigation of AGV through Iterative Learning,” Proceeding of *International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium*, pp. 968–973, 2001.

[47] Y. S. Chen and L. Yao, “Robust Type-2 Fuzzy Control of an Automatic Guided Vehicle for Wall-Following,” Proceeding of *2009 International Conference of Soft Computing and Pattern Recognition*, pp. 172–177, 2009.

[48] E. Jung, H. Cho, J. Do, J. Kim, and S. Kim, “Implementation of Laser Navigation System Using Particle Filter,” Proceeding of *11th International Conference on Control, Automation and Systems*, pp. 1636–1638, 2011.

[49] H. Golnabi, “Role of Laser Sensor Systems in Automation and Flexible Manufacturing,” *Robotics and Computer-Integrated Manufacturing*, Vol. 19, No. 1–2, pp. 201–210, 2003.

[50] E. Jung, K. Jung, J. Kim, and S. Kim, “Localization for Fork-Lift AGV Using Extended Kalman Filter,” *Advances in Mechanical Engineering*, Vol. 1, No. 1, pp. 15–20, 2011.

[51] Y. A. Turki and A. A. Abdulkareem, "PSO-Based Optimum Design of PID Controller for Mobile Robot Trajectory Tracking," *International Journal of Computer Applications*, Vol. 47, No. 23, pp. 30–35, 2012.

[52] W. S. Wijesoma, P. P. Khaw, and E. K. Teoh, "Control and Navigation of an Outdoor AGV Using Fuzzy Reasoning," Proceeding of *199 IEEE/IEEJ/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383)*, pp. 544–549, 1999.

[53] R. Solea, A. Filipescu, and U. Nunes, "Sliding-Mode Control for Trajectory-Tracking of a Wheeled Mobile Robot in Presence of Uncertainties," Proceeding of *2009 7th Asian Control Conference*, pp. 1701 – 1706, 2009.

[54] T. L. Bui, P. T. Doan, S. S. Park, H. K. Kim, and S. B. Kim, "AGV Trajectory Control Based on Laser Sensor Navigation," *International Journal of Science and Engineering*, Vol. 4, No. 1, pp. 16–20, 2013.

[55] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A Stable Tracking Control Method for an Autonomous Mobile Robot," Proceeding of *IEEE International Conference on Robotics and Automation*, Vol. 30, No. 5, pp. 523–530, 1990.

[56] Z. P. Jiang and H. Nijmeijer, "Tracking Control of Mobile Robots: A Case Study in Backstepping," *Automatica*, Vol. 33, No. 97, pp. 1393–1399, 1997.

[57] L. Feng, Y. Koren, and J. Borenstein, "A Model-Reference Adaptive Motion Controller for a Differential-Drive Mobile Robot," Proceeding of *The 1994 IEEE International Conference on Robotics and Automation*, pp. 3091–3096, 1994.

[58] T. Fukao, H. Nakagawa, and N. Adachi, “Adaptive Tracking Control of a Nonholonomic Mobile Robot,” *IEEE Transactions on Robotics and Automation*, Vol. 16, No. 5, pp. 609–615, 2000.

[59] T. H. Bui, T. L. Chung, S. B. Kim, and T. T. Nguyen, “Adaptive Tracking Control of Two-Wheeled Welding Mobile Robot with Smooth Curved Welding Path,” *KSME International Journal*, Vol. 17, No. 11, pp. 1682–1692, 2003.

[60] K. S. Chong and L. Kleeman, “Accurate Odometry and Error Modelling for a Mobile Robot,” Proceeding of *International Conference on Robotics and Automation*, Vol. 4, No. April, pp. 2783–2788, 1997.

[61] C. Carlson, J. Gerdes, and J. Powell, “Error Sources When Land Vehicle Dead Reckoning with Differential Wheelspeeds,” *Navigation*, Vol. 51, pp. 13–27, 2004.

[62] Y. Zhang and K. T. Chong, “A GPS / DR Data Fusion Method Based on the GPS Characteristics for Mobile Robot Navigation,” *International Journal of Control and Automation*, Vol. 7, No. 10, pp. 119–132, 2014.

[63] J. Biswas and M. Veloso, “WiFi Localization and Navigation for Autonomous Indoor Mobile Robots,” Proceeding of *IEEE International Conference on Robotics and Automation*, pp. 4379–4384, 2010.

[64] S. Ting, S. Kwok, A. Tsang, and G. Ho, “The Study on Using Passive RFID Tags for Indoor Positioning,” *International Journal of Engineering Business Management*, Vol. 3, No. 1, pp. 9–15, 2011.

[65] H. D. Chon, S. Jun, H. Jung, and W. An, "Using RFID for Accurate Positioning," *Journal of Global Positioning Systems*, Vol. 3, No. 1–2, pp. 32–39, 2004.

[66] D. Wang, C. Low, B. He, and M. Pham, "Accurate Positioning for Real-Time Control Purpose Integration of GPS, NAV200 and Encoder Data," Proceeding of *8th Control, Automation, Robotics and Vision Conference, 2004.*, Vol. 1, pp. 161–166, 2004.

[67] H. Kawata, A. Ohya, and S. Yuta, "Development of Compact and Light-Weight LRF Based Positioning Sensor for Mobile Robot Localization," Proceeding of *International Conference on Advanced Robotics*, pp. 1–6, 2009.

[68] F. Scheer and S. Müller, "Indoor Tracking for Large Area Industrial Mixed Reality," Proceeding of *Joint Virtual Reality Conference of ICAT - EGVE - EuroVR*, pp. 21–28, 2012.

[69] J. G. Kim, D. H. Kim, S. K. Jeong, H. K. Kim, and S. B. Kim, "Development of Navigation Control Algorithm for AGV Using D * Search Algorithm," *International Journal of Science and Engineering*, Vol. 4, No. 2, pp. 34–38, 2013.

[70] N. M. Kwok, Q. P. Ha, and G. Fang, "Data Association in Bearing-Only SLAM Using a Cost Function-Based Approach," Proceeding of *IEEE International Conference on Robotics and Automation*, pp. 4108–4113, 2007.

[71] A. Diosi and L. Kleeman, "Laser Scan Matching in Polar Coordinates with Application to SLAM," Proceeding of *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS*, pp. 1439–1444, 2005.

[72] H. Hu and D. Gu, “Landmark-Based Navigation of Industrial Mobile Robots,” *International Journal of Industry Robot*, Vol. 27, No. 6, pp. 458–467, 2000.

[73] G. Spampinato, J. Lidholm, L. Asplund, and F. Ekstrand, “Stereo Vision Based Navigation for Automated Vehicles in Industry,” Proceeding of *ETFA 2009 - 2009 IEEE Conference on Emerging Technologies and Factory Automation*, 2009.

[74] A. Elfes, “Sonar-Based Real-World Mapping and Navigation,” *IEEE Journal on Robotics and Automation*, Vol. 3, No. 3, 1987.

[75] F. Baghernezhad and K. Khorasani, “A Robust Fault Detection Scheme with an Application to Mobile Robots by Using Adaptive Thresholds Generated with Locally Linear Models,” Proceeding of *2013 IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, pp. 9–16, 2013.

[76] E. N. Skoundrianos and S. G. Tzafestas, “Fault Diagnosis on the Wheels of a Mobile Robot Using Local Model Neural Networks,” *IEEE Robotics & Automation Magazine*, Vol. 11, No. 3, pp. 83–90, 2004.

[77] W. Y. Jeong, P. S. Pratama, B. H. Jun, and S. B. Kim, “Fault Detection for Underwater Hexapod Robot Based on PI Observer,” Proceeding of 한국해양과학기술공동학술대회 논문집, pp. 1772–1777, 2013.

[78] J. Min, W. Jeong, P. S. Pratama, and Y. D. Setiawan, “Fault Diagnosis System of Caterpillar Wheel Type Pipeline

Inspection Robot," Proceeding of *International Symposium on Mechatronics and Robotics*, pp. 1–5, 2013.

[79] A. Monteriu, P. Asthana, K. Valavanis, and S. Longhi, "Experimental Validation of a Real-Time Model-Based Sensor Fault Detection and Isolation System for Unmanned Ground Vehicles," Proceeding of *2006 14th Mediterranean Conference on Control and Automation*, pp. 1–8, 2006.

[80] M. Ji, Z. Zhang, G. Biswas, and N. Sarkar, "Hybrid Fault Adaptive Control of a Wheeled Mobile Robot," *IEEE/ASME Transactions on Mechatronics*, Vol. 8, No. 2, pp. 226–233, 2003.

[81] Z. Cai, Z. Duan, J. Cai, X. Zou, and J. Yu, "A Multiple Particle Filters Method for Fault Diagnosis of Mobile Robot Dead-Reckoning System," Proceeding of *International Conference on Intelligent Robots and Systems, IROS*, pp. 480–485, 2005.

[82] Z. Cai, J. Yu, and Z. Duan, "Adaptive Particle Filter for Unknown Fault Detection of Wheeled Mobile Robots," Proceeding of *The 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1312–1315, 2006.

[83] P. S. Pratama, T. Yeu, S. Hong, H. K. Kim, and S. B. Kim, "Path Planning Method for Coverage of Uneven Terrain Environment Using Morse Cell Decomposition," Proceeding of *Korean Society of Ocean Engineering Conference 2014*, pp. 151–154, 2014.

[84] W. H. Huang, "Optimal Line-Sweep-Based Decompositions for Coverage Algorithms," Proceeding of *IEEE International*

Conference on Robotics and Automation, Vol. 1, pp. 27–32, 2001.

[85] P. M. Hsu, C. L. Lin, and M. Y. Yang, “On the Complete Coverage Path Planning for Mobile Robots,” *Journal of Intelligent & Robotic Systems*, Vol. 74, No. 3–4, pp. 945–963, 2013.

[86] S. B. Kim, P. S. Pratama, D. H. Kim, and Y. S. Jung, “Controller Design for MIMO Servo System Using Polynomial Differential Operator,” *to appear in the International Journal of Control*, 2015.

[87] D. hwan Kim, “Servo Controller Design Using Polynomial Differential Operator Method and Its Application,” Pukyong National University, 2015.

[88] K. Furuta, A. Sano, and D. Atherton, *State Variable Methods in Automatic Control*. Michigan: Wiley, 1988.

[89] K. Furuta and S. B. Kim, “Pole Assignment in a Specified Disk,” *IEEE Transactions on Automatic Control*, Vol. 32, No. 5, pp. 423–427, 1987.

[90] S. B. Kim and K. Furuta, “Regulator Design with Poles in a Specified Region,” *International Journal of Control*, Vol. 47, No. 1, pp. 143–160, 1988.

Publication and Conference

Published Journal

1. Phuc Thinh Doan, **Pandu Sandi Pratama**, Suk Yoel Kim, Hak Kyeong Kim, "*Developmet of Digital Gas Metel Arc Welding System and Welding Current Control Using Self-tuning Fuzzy PID*", Journal of The Korean Society of Ocean Engineering (KSOE), Vol. 6, pp. 1-8, 2012. **KOREA**
2. **Pandu Sandi Pratama**, Sang Kwun Jeong, Soon Sil Park, Sang Bong Kim, "*Moving Object Tracking and Avoidance Algorithm for Differential Driving AGV Based on Laser Measurement Technology*", International Journal of Science and Engineering, Vol. 4, No.1, pp. 11-15, 2013. **SCOPUS**
3. **Pandu Sandi Pratama**, Bui Thanh Luan, Thien Phuc Tran, Hak Kyeong Kim, Sang Bong Kim, "*Trajectory Tracking Algorithm for Automatic Guided Vehicle Based on Adaptive Backstepping Control Method*", Lecture notes in Electrical Engineering, pp. 535-544, 2013. **SCOPUS**
4. Yuhanes Dedy Setiawan, **Pandu Sandi Pratama**, Sang Kwon Jeong, Vo Hoang Duy, and Sang Bong Kim, "*Experimental Comparison of A* and D* Lite Path Planning Algorithms for Differential Drive Automated Guided Vehicle*," Lecture notes in Electrical Engineering, pp. 535-544, 2013. **SCOPUS**

Journal under submission

1. **Pandu Sandi Pratama**, Yuhanes Dedy Setiawan, Dae Hwan Kim and Sang Bong Kim, “*Trajectory Tracking and Fault Detection Algorithm for Automatic Guided Vehicle Based on Multiple Positioning Modules*,” Submission to International Journal of Control, Automation and Systems, 2015. **SCIE**
2. Yuhanes Dedy Setiawan, **Pandu Sandi Pratama**, Jin Wook Kim, Dae Hwan Kim and Sang Bong Kim. “*Path Replanner and Tracking Controller for Differential Drive Automated Guided Vehicles*,” Submission to Journal of Mechanical Science and Technology, 2015. **SCIE**
3. **Pandu Sandi Pratama**, Yuhanes Dedy Setiawan, Dae Hwan Kim, and Sang Bong Kim, “*Positioning and Obstacle Avoidance of Automatic Guided Vehicle in PartiallyKnown Environment*,” Submission to International Journal of Control, Automation and Systems, 2015. **SCIE**
4. Amruta Vinod Gulalkari, Giang Hoang, **Pandu Sandi Pratama** and Sang Bong Kim, “*Object Tracking and Following Six-legged Robot System using Kinect Camera Based on Kalman Filter and Backstepping Controller*,” Submission to Journal of Mechanical Science and Technology, 2015. **SCIE**
5. Yuhanes Dedy Setiawan, **Pandu Sandi Pratama**, Amruta Vinod Gulalkari, Hak Kyeong Kim and Sang Bong Kim, “*Path Tracking Controller Design of Four Wheel Independent Steering Automatic Guided Vehicle*,” Submission to International Journal of Control, Automation and Systems, 2015. **SCIE**
6. Dae Hwan Kim, **Pandu Sandi Pratama**, Phuc Thihn Doan, Hak Kyeong Kim, Young Seok Jung, and Sang Bong Kim, “*Servo*

System Design for Speed Control of AC Induction Motors Using Polynomial Differential Operator," Submission to International Journal of Control, Automation and Systems, 2015. **SCIE**

Conference Paper

1. **Pandu Sandi Pratama**, Ngo Manh Dung, Soon Sil Park, Hak Kyeong Kim, Gi Sig Byun and Sang Bong Kim, "*Voltage Control and Current Control for a Digital Gas Metal Arc Welding System Using P Control Method and Fuzzy-Sliding Mode Method*", Proceeding of The 2011 International Symposium on Mechatronics and Robotics, Ho Chi Minh city, Vietnam, pp. 185-192, 2011.
2. **Pandu Sandi Pratama**, Tan-Tung Phan, Hak Kyeong Kim, and Sang Bong Kim, "*A Hybrid Controller Design for Keeping Constant Voltage and Current of a Gas Metal Arc Welding System*", Proceeding of the 2011 International Symposium on Advance Engineering, Busan, Korea, pp. 121-126, 2011.
3. **Pandu Sandi Pratama**, Sang Kwun Jeong, Soon Sil Park, Sang Bong Kim, "*Moving Object Tracking and Avoidance Algorithm for Differential Driving AGV Based on Laser Measurement Technology*", Proceeding of 1st International Joint Conference on Advanced Engineering, Semarang, Indonesia, pp. 78-83, 2012.
4. Chetanraj Patil, **Pandu Sandi Pratama**, Hak Kyeong Kim, Sang Bong Kim, "*Control of Two Wheeled Inverted Pendulum, using Kalman Filter and Backstepping Controller*", Proceeding of 2012 International Symposium on Advanced Mechanical and Power Engineering, Sanghai, China, pp. 34, 2012.

5. Woo-Young Jeong, **Pandu Sandi Pratama**, Bong-Huan Jun and Sang-Bong Kim, *"Fault Detection for Underwater Hexapod Robot Based on PI Observe,"* Proceeding of conference The Korean Society of Ocean Engineering, Jeju, Korea, 2012.
6. **Pandu Sandi Pratama**, Thanh Luan Bui, Jeong-Geun Kim, Hak-Kyeong Kim, Sang-Bong Kim, "*Trajectory Tracking of Automatic Guided Vehicle Based on Simultaneous Localization and Mapping Method*" Proceeding of The Korean Society of Logistic Science and Technology, Uiwang, Korea, 2013.
7. Yuhanes Dedy Setiawan, **Pandu Sandi Pratama**, Hak-Kyeong Kim, Sang-Bong Kim, *"Modeling and Control System Design of Four Wheel Independent Steering Automatic Guided Vehicle Based on Optimal Control Method,"* Proceeding of The Korean Society of Logistic Science and Technology, Uiwang, Korea, 2013.
8. Yuhanes Dedy Setiawan, **Pandu Sandi Pratama**, Woo Young Jeong, Hak Kyeong Kim, and Sang Bong Kim, "*Control System Design of Four Wheeled Independent Steering Automatic Guided Vehicles Based on Backstepping Control Method,*" Proceedings of The 2013 International Symposium on Advanced Engineering, Busan, Korea, pp. 85-88, 2013.
9. Jung-Hu Min, Woo-Young Jeong, **Pandu Sandi Pratama**, Yuhanes Dedy Setiawan, Hak-Kyeong Kim and Sang-Bong Kim, "*Fault Diagnosis System of Caterpillar Wheel Type Pipeline Inspection Robot,*" Proceeding of International Symposium on Mechatronics and Robotics, Ho Chi Minh city, Vietnam, 2013.
10. Dae-Hwan Kim, **Pandu Sandi Pratama**, Suk-Min Yoon, Tae-Kyeong Yeu, Sup Hong, Hak-Kyeong Kim, and Sang-Bong Kim,

"*Navigation Control for Automated Guided Vehicle Using D* Algorithmand Backstepping Control Method*," Proceeding of International Symposium on Mechatronics and Robotics, Ho Chi Minh city, Vietnam, 2013.

11. Junghu Min, Yuhannes Dedy Setiawan, **Pandu Sandi Pratama**, Sang Bong Kim, Hak Kyeong Kim, "*Development and Controller Design of Wheeled-type Pipe Inspection Robot*," Proceeding of 2014 IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 789-795, 2014.
12. **Pandu Sandi Pratama**, Yuhannes Dedy Setiawan, Dae Hwan Kim, Young Seok Jung, Hak Kyeong Kim, Sang Bong Kim, Sang Kwun Jeong, Jin Il Jeong, "*Fault Detection Algorithm for Automatic Guided Vehicle Based on Multiple Positioning Modules*," Proceeding of 2014 IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI), Greater Noida, India, pp. 751-757, 2014.
13. Yuhannes Dedy Setiawan, **Pandu Sandi Pratama**, Jin Wook Kim, Dae Hwan Kim, Young Seok Jung, Sang Bong Kim, Suk Min Yoon, Tae Kyeong Yeo, Sup Hong, "*Path Replanning and Controller Design for Trajectory Tracking of Automated Guided Vehicles*," Proceeding of 2014 IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI), Greater Noida, India, pp. 771-777, 2014.
14. Amruta Vinod Gulalkari, Giang Hoang, **Pandu Sandi Pratama**, Hak Kyeong Kim, Sang Bong Kim, Bong Huan Jun, "*Object Following Control of Six-legged Robot using Kinect Camera*," Proceeding of 2014 IEEE International Conference on

Advances in Computing, Communications and Informatics (ICACCI), Greater Noida, India, pp. 758-764, 2014.

15. **Pandu Sandi Pratama**, Jin-Wook Kim,Jung-Hu Min, Suk-Min Yoon, Tae-Kyeong Yeu, Sup Hong, Hak Kyeong Kim and Sang Bong Kim, "*Path Planning Method for Coverage of Uneven Terrain Environment Using Morse Cell Decomposition*," Proceeding of conference The Korean Society of Ocean Engineering, Jeju, Korea, pp.151-154, 2014.
16. Yuhanes Dedy Setiawan, **Pandu Sandi Pratama**, Dae Hwan Kim, Suk Min Yoon, Tae Kyoung Yeu, Sub Hong, Hak Kyeong Kim and Sang Bong Kim, "*Path Planning Based on A* Algorithm and Its Tracking Controller for Underwater Mining Robots*," Proceeding of conference The Korean Society of Ocean Engineering, Jeju, Korea, pp.147-150, 2014.
17. Yuhanes Dedy Setiawan, **Pandu Sandi Pratama**, Hak Kyeong Kim and Sang Bong Kim, "*Trajectory Tracking Controller Design for Four Wheel Independent Steering Automatic Guided Vehicles*," Proceeding of 2014 International Symposium on Advanced Mechanical and Power Engineering, Busan, Korea, pp. 109-113, 2014.
18. **Pandu Sandi Pratama**, Jin-Wook Kim, Suk-Min Yoon, Tae-Kyeong Yeu, Sup Hong, Hak Kyeong Kim and Sang Bong Kim, "*Path Planning Method and Trajectory Tracking Controller for Coverage Uneven Terrain Structure of The Ocean Floor*," Proceeding of 2014 International Symposium on Advanced Mechanical and Power Engineering, Busan, Korea, pp. 122-132, 2014.

19. **Pandu Sandi Pratama**, Amruta Vinod Gulalkari, Dae-Hwan Kim, Hak-Kyeong Kim, Young Seok Jung, Jung-Hu Min, and Sang Bong Kim, "*Fault Detection and Identification of Induction Motor for a Crossbelt Sorter Shooting System*," Proceeding of The Korean Society of Logistic Science and Technology, KINTEX, Korea, 2015.