



HOKKAIDO UNIVERSITY

Title	Development of Robot Vehicles Adaptable to Changing Ground Conditions and Their Work Management System
Author(s)	王, 昊
Degree Grantor	北海道大学
Degree Name	博士(農学)
Dissertation Number	甲第13765号
Issue Date	2019-09-25
DOI	https://doi.org/10.14943/doctoral.k13765
Doc URL	https://hdl.handle.net/2115/75874
Type	doctoral thesis
File Information	Hao_Wang.pdf



**Development of robot vehicles adaptable to
changing ground conditions and their work
management system**

(路面環境に適応可能なロボット車両群とその
作業管理システムの開発)

2019

北海道大学 大学院農学院

環境資源学専攻 博士後期課程

王 昊

Hao Wang

Development of robot vehicles adaptable to changing ground conditions and their work management system



By

Hao Wang

Dissertation

*Submitted to Department of Environmental Resources in
the Graduate School of Agriculture*

*Hokkaido University, Sapporo, Japan, 060-8589 in
partial fulfillment of the requirements for the degree of
Doctor of Philosophy*

2019

*Dedicated to my parents, who with their love, patience, and encouragement
during my studies, made this aim possible.*

TABLE OF CONTENT

Table of Contents

TABLE OF CONTENT	I
ABSTRACT	IV
ACKNOWLEDGEMENT	VI
LIST OF FIGURES	VII
LIST OF TABLES	XI
NOTATIONS	XII
ACRONYMS AND ABBREVIATIONS	XIV
CHAPTER 1 INTRODUCTION	1
1.1 The situation and challenges of agriculture	1
1.2 Solutions in policies and technology	2
1.3 Objectives	4
CHAPTER 2 NAVIGATION OF A ROBOT TRACTOR USING THE JAPANESE QUASI-ZENITH SATELLITE SYSTEM	6
2.1 Research platforms	6
2.1.1 Robot tractors.....	6
2.1.2 The attitude sensor	7
2.2 Satellite-based positioning system	8
2.2.1 Overview of QZSS and augmentation services	10
2.2.2 Complementary signals and augmentation signals	11
2.3 Application of QZSS to the robot tractor	15
2.3.1 Application of PPP-RTK to a Robot Tractor	15
2.3.2 Application of PPP method to a robot tractor	20
2.3.3 Application of PPP-AR method to a robot tractor	27
CHAPTER 3 FIELD REPRESENTATION AND PATH PLANNING	31

3.1	Field representation	32
3.1.1	Boundaries	33
3.1.2	Main working direction.....	34
3.1.3	Field splitting	35
3.1.4	Path planning for a convex field	36
3.2	Headland turning algorithms.....	38
3.2.1	Forward turning	39
3.2.2	Reverse turning	41
CHAPTER 4 VEHICLE MODEL AND STATE ESTIMATION.....		48
4.1	Vehicle modeling	49
4.1.1	The unicycle model of vehicle kinematics.....	49
4.1.2	The bicycle model of lateral vehicle kinematics.....	52
4.1.3	The bicycle model of lateral vehicle dynamics.....	54
4.1.4	System identification	56
4.2	Vehicle state estimation	72
4.2.1	Estimation of a vehicle heading by the fusion of an RTK-GPS and IMU 72	
4.2.2	Improvement of the vehicle state estimation with sensor fusion methods 74	
CHAPTER 5 CONTROL ALGORITHMS FOR AGRICULTURAL ROBOTS		94
5.1	The path tracking algorithm.....	94
5.2	Dynamic circle-back turning the algorithm	96
5.2.1	Specification of the adaptive turning control algorithm	96
5.2.2	Objective function based on weights	97
5.3	Experiments and evaluation.....	102
5.3.1	Evaluation of navigation accuracy.....	103
5.3.2	Comparison to the switch-back turning algorithm.....	112
5.3.3	Turning at the irregular headland.....	113
CHAPTER 6 REMOTE MONITOR AND CONTROL SYSTEM.....		117
6.1	System construction	117
6.2	Software development for remote control	119

CHAPTER 7 RESEARCH SUMMARY	126
REFERENCES.....	128

Development of robot vehicles adaptable to changing ground conditions and their work management system

路面環境に適応可能なロボット車両群とその作業管理システムの開発

Hao Wang

ABSTRACT

Keywords: smart agriculture, farm management system, QZSS, robot tractor, navigation, machine learning

Increased and sustained agricultural productivity is needed to meet the globally increasing demands for food and energy. The aging problem and the labor shortage will cause a series of matters and will affect the development of agriculture seriously. Smart farming is a promising management style to help farmers efficiently manage their farms. Sophisticated farm management systems (FMSs) can plan, monitor, and control agricultural processes. As a central part of the smart agriculture system, autonomous agricultural vehicles adaptable to changing ground conditions and a remote-control system of robot vehicles are developed in this research.

The robot tractor calculates the real-time position in centimeter-level accuracy using GPS and Japanese navigation satellites; i.e., the Quasi-Zenith Satellite System (QZSS). The QZSS provides centimeter-level augmentation service (CLAS) through the L6 signal to enhance the global navigation satellite system (GNSS) positioning accuracy. In addition, the control parameters of the navigation algorithm are tuned automatically for adapting to the change of soil conditions. Finally, a remote monitoring system of an autonomous tractor is developed. The satellite image of the farm and the working conditions of the robot tractor can be monitored in real-time. Also, the manager can control the robot tractor remotely through the system.

1. Navigation of a robot tractor using the Quasi-Zenith Satellite System

This study evaluates the CLAS of the QZSS for controlling a robot tractor. Besides accessing the augmentation information through the L6 signal using a commercial QZSS receiver, this research also introduces a method for using CLAS with a dual-

frequency receiver that cannot receive the L6 signal. Stationary and dynamic positioning experiments prove that the QZSS can improve the accuracy and availability of the current GNSS. The usage of Precise Point Positioning (PPP) in autonomous navigation is restricted in ambiguity resolution. Experiments using a commercial QZSS receiver reveal that the PPP converges to different results at the same position, and the results drift after convergence. A bias identification method based on landmarks was proposed to overcome these limitations. By compensating the biases of the CLAS positioning results relative to the current GNSS, a robot tractor works along with GNSS-based navigation within 5 cm error at a speed of 3.6 km/h.

2. Mapping and path planning

An optimal coverage path planning method is presented to improve field efficiency and to utilize the advantages of autonomous vehicles fully. In addition, several transfer paths are created to optimize the non-working distance and time consumption. Refiling or emptying the machine is not considered in this autonomous path planning. The sequence of the working order is also designated. When the field is divided into several sub-fields, each sub-area has to be visited once without discard. To use the merit of agricultural robots, the backward movement along the navigation path is proposed in this research.

3. Vehicle system identification and state estimation

Vehicles are mostly assumed to comply with specific motion models which describe their dynamic behaviors. This research introduces three vehicle models that are widely applied for control. Besides, data-driven parameter identification and state estimation methods are presented to increase the accuracy of control algorithms. Finally, an adaptive turning algorithm for a four-wheel robot tractor in the headland is presented in this research. The asymmetric steering mechanism is taken into consideration with a vehicle model. An objective function based on weights is used to create the navigation path, connecting by continuous primitives. Field experiments show that the robot tractor can approach the next path with an average lateral deviation of 3.9 cm at a speed of 1.2 m/s during a turn.

4. Farm management system

A field management system is developed for the farm manager to control in remote and to monitor the robot tractor working at the field. Farmers can also access the server to check the work record of the tractor, as well as editing the working plan through a smartphone or a PC. This research presents a part of functional architecture and provides an operational example of the management system.

ACKNOWLEDGEMENT

Above all, I would like to express my gratitude to my advisor, Prof. Dr. Noboru Noguchi, a respectable, responsible, and resourceful scholar. Prof. Noguchi provided me with valuable guidance in every stage of my research.

I also would like to express my thanks to Dr. Kazunobu Ishii and Dr. Hiroshi Okamoto for sharing their knowledge and ideas during my research.

I would like to thank the support obtained through the MEXT scholarship and the R&D program of SIP "Technologies for Creating Next-generation Agriculture, Forestry and Fisheries."

I shall thank my colleagues in the Laboratory of Vehicle Robotics. Thank you for cooperating and helping in many experiments. Especially, I want to thank Mr. Hiroyuki Sato and Mr. Tomonori Wada. Without their help all through these five years, I cannot experiment successfully.

I would also like to thank the committees of CIGR (International Commission of Agricultural and Biosystems Engineering) who inspired my motivation of research by honoring me the Armand Blanc Prize in 2018.

I need to thank my supervisor at Beijing Institute of Technology, Dr. Zhifeng Ma, and my friends, like Mr. Qiang Yao, Ms. Yingyao Bi, and so on, who advise me and support me.

Last but not least, I would like to express my thanks to my family members, including Yaxin and Junbo, for their valuable encouragement and accompany all the time.

LIST OF FIGURES

Fig. 2. 1. Experimental tractors.....	7
Fig. 2. 2. The control unit of the tractor.....	7
Fig. 2. 3. Coverage of QZSS and its trajectory	10
Fig. 2. 4. The overall architecture of the QZSS with augmentation service.....	11
Fig. 2. 5. Area assignment of CLAS augmentation information	14
Fig. 2. 6. The service area of CLAS-E.....	15
Fig. 2. 7. Scheme of the positioning system.	16
Fig. 2. 8. Receiving the augmentation information with the RTK receiver (T05 in Fig. 2. 7).....	17
Fig. 2. 9. Navigation map for dynamic tests.	19
Fig. 2. 10. Satellite number and GNSS solution conditions during the transfer path (from the barn to the field).	19
Fig. 2. 11. Scheme of the positioning system using the PPP method.....	21
Fig. 2. 12. The horizontal error of PPP.	22
Fig. 2. 13. Time-series positioning errors of MADOCA PPP in a static condition (540 min).....	23
Fig. 2. 14. The trajectory of the robot tractor during the dynamic test.....	25
Fig. 2. 15. Time-series lateral deviation and heading error of the robot tractor during Test #3.	26
Fig. 2. 16. Time-series lateral deviation and heading error of the robot tractor during Test #4.	26
Fig. 2. 17. Scheme of the positioning system.	28
Fig. 2. 18. The trajectory of the robot tractor after one trail.....	28
Fig. 2. 19. Offsets between QZSS and RTK-GPS coordinates.....	29
Fig. 2. 20. Estimated visibility of QZS1 from the view of Mackay, Queensland, Australia.....	30
Fig. 2. 21. The lateral error of fertilizer spreading test.....	30
Fig. 3. 1. An example of generated navigation paths using the MMB method.	35
Fig. 3. 2. Sub-fields generation in a field with obstacles.	35
Fig. 3. 3. The effect of the main working direction on each sub-field.....	36
Fig. 3. 4. Scheme of navigation paths.....	36
Fig. 3. 5. A code for the tillage operation.	38
Fig. 3. 7. U-turn at the headland.	40
Fig. 3. 8. Working sequence of six paths.....	40
Fig. 3. 9. Keyhole turning at the headland.....	41

Fig. 3. 10. Slope-turn.	41
Fig. 3. 11. Switch-back turning method.....	42
Fig. 3. 12. Fishtail-turn with minimal headland width.	42
Fig. 3. 13. H-turning modified from U-turn.....	43
Fig. 3. 14. H-turning modified from switch-back turning.	43
Fig. 3. 15. H-turning at the irregular shape of the headland.	44
Fig. 3. 16. Scheme of the CB turning algorithm.	45
Fig. 3. 17. Template of side shift turning.....	47
Fig. 3. 18. Template of backward tracker.	47
Fig. 4. 1. Block diagram of navigation procedure for path following control.....	48
Fig. 4. 2. The unicycle motion model (Lavalle, 2006).	50
Fig. 4. 3. Lateral vehicle motion.	53
Fig. 4. 4. Lateral vehicle dynamics.	54
Fig. 4. 5. Time series of slip angles.	58
Fig. 4. 6. Time series of steering angle.	58
Fig. 4. 7. A linear regression model for system identification.....	59
Fig. 4. 8. Time series of vehicle speed.....	59
Fig. 4. 9. Comparison of measured slip angle and the estimated by the kinematic model.	60
Fig. 4. 10. Comparison of measured yaw rate and the estimated by the kinematic model.	60
Fig. 4. 11. Comparison of IMU measured vehicle heading and the one deducted by the kinematic model.	61
Fig. 4. 12. Comparison of ground truth and estimated vehicle trajectories from Eq. (16).....	61
Fig. 4. 13. The illustration of system identification with data collections in time.....	63
Fig. 4. 14. Time-series slip angles.	64
Fig. 4. 15. Time-series yaw rates.	64
Fig. 4. 16. Comparison of ground truth and estimated vehicle trajectories from Eq. (16).....	65
Fig. 4. 17. Comparisons of vehicle heading.....	65
Fig. 4. 18. Time series of slip angles.	66
Fig. 4. 19. Time series of yaw rates.	67
Fig. 4. 20. Comparison of ground truth and estimated vehicle trajectories from Eq. (16).....	67
Fig. 4. 21. Comparisons of vehicle heading.....	67
Fig. 4. 22. Error analysis of estimating slip angles with an updated dynamic model..	68

Fig. 4. 23. Error analysis of estimating yaw rate with an updated dynamic model.	68
Fig. 4. 24. The variance of turning radii of different steering angles.	70
Fig. 4. 25. Comparison of vehicle heading measured by the IMU, LSM, and GPS.	74
Fig. 4. 26. Time series of gyro bias identified by LSM.	74
Fig. 4. 27. Time series of steering angle as input.	74
Fig. 4. 28. Time series of VTG heading.	75
Fig. 4. 29. Time series of calculated GPS heading.	76
Fig. 4. 30. GPS heading after fusion.	76
Fig. 4. 31. Comparison of the GPS heading and the IMU heading.	76
Fig. 4. 32. Time series of calculated slip angles (beta).	77
Fig. 4. 33. Time series of yaw rate.	77
Fig. 4. 34. Comparison of estimated vehicle trajectories according to Eq. (16) and Eq. (40).	79
Fig. 4. 35. Time series of vehicle heading angles.	81
Fig. 4. 36. Comparison of estimated vehicle trajectories according to Eq. (39).	82
Fig. 4. 37. The comparison of vehicle heading.	83
Fig. 4. 38. Comparison of estimated vehicle trajectories according to Eq. (39).	84
Fig. 4. 39. Comparison of estimated vehicle trajectories.	85
Fig. 4. 40. Four experiments of algorithm evaluation using a four-wheel-type robot tractor.	86
Fig. 4. 41. Comparison of estimated vehicle trajectories according to Eq. (39).	87
Fig. 4. 42. Comparison of estimated vehicle trajectories according to Eq. (39).	88
Fig. 4. 43. Time series of steering angles.	89
Fig. 4. 44. Estimated vehicle positions based on EKF methods.	89
Fig. 4. 45. Comparison of estimated vehicle trajectories according to Eq. (39).	90
Fig. 4. 46. Time series of steering angles.	91
Fig. 4. 47. Estimated vehicle positions based on EKF methods.	91
Fig. 4. 48. Comparison of estimated vehicle trajectories according to Eq. (39).	92
Fig. 5. 1. Lateral controller for a robot tractor.	95
Fig. 5. 2. Flowchart of the DCB turning algorithm.	97
Fig. 5. 3. Working flow of the DCB turning.	99
Fig. 5. 4. Planned paths for turning.	100
Fig. 5. 5. Steering angle and speed during turning.	101
Fig. 5. 6. The trajectory of robot tractor EG105.	103
Fig. 5. 7. Lane change order 2.	104
Fig. 5. 8. Lateral errors in each working path.	104
Fig. 5. 9. Heading errors in each working path.	105

Fig. 5. 10. The trajectory of robot tractor EG83.	106
Fig. 5. 11. Lateral errors in each working path.	107
Fig. 5. 12. Heading errors in each working path.	107
Fig. 5. 13. Trajectories of the robot tractor during the third turn.	110
Fig. 5. 14. The update of vehicle model parameters.	111
Fig. 5. 15. Trajectories of the robot tractor when turning.	111
Fig. 5. 16. Turning trajectories of the robot tractor.	112
Fig. 5. 17. The trajectory in an irregular shape of the field.	114
Fig. 5. 18. Turning trajectory from Path 2 to Path 3.	115
Fig. 5. 19. Lateral error in the straight path of turning.	115
Fig. 5. 20. Speed and Steering angle during turning.	116
Fig. 6. 1. The block diagram of the navigation system.	118
Fig. 6. 2. Working flow of tractor control and monitor.	119
Fig. 6. 3. The user interface of the farm management system.	120
Fig. 6. 4. The map showing on different scales.	121
Fig. 6. 5. Farmland areas showing in the satellite image.	122
Fig. 6. 6. The procedure for creating a new working area.	123
Fig. 6. 7. Parameters for remote monitor and control.	124
Fig. 6. 8. The software for controlling a robot tractor.	124

LIST OF TABLES

Table 1. 1. Number of agricultural management units and the average age of farmers in Japan.....	1
Table 2. 1. Transmission signals of QZSS.....	12
Table 2. 2. Correction information on positioning errors.	13
Table 2. 3. Accuracy analysis of CLAS in static conditions (Test #1).....	18
Table 2. 4. Accuracy analysis of CLAS in static conditions (Test #2).....	18
Table 2. 5. The accuracy of the robot tractor.	20
Table 2. 6. 3D positioning errors in the static test using MADOCA PPP.	23
Table 2. 7. Positioning errors and offset of PPP solution in Test #2.	24
Table 2. 8. Description of dynamic tests.....	25
Table 2. 9. Accuracy of the robot tractor using MADOCA PPP as positioning signal	26
Table 2. 10. The lateral error (RMS) in the fertilizer spreading test.	30
Table 4. 1. Parameters for identification.....	57
Table 4. 2. A sample of training data.....	69
Table 4. 3 Vehicle state variables	72
Table 4. 4 Error analysis of EKF based positioning methods.....	85
Table 4. 5 Positioning accuracy using Eq. (35).	92
Table 4. 6 Error analysis of EKF based positioning methods.....	93
Table 5. 1. Lateral deviation and heading error in each path.....	105
Table 5. 2. Efficiency and accuracy of each lane change	106
Table 5. 3. Lateral deviation and heading error in each path.....	107
Table 5. 4. Efficiency and accuracy of each lane change	108
Table 5. 5. Field tests in different weights combinations	109
Table 5. 6. Comparison of dynamic circle-back turning and switch-back turning....	113
Table 5. 7. Results of 5 lane change maneuvers in Fig. 5. 17.....	114
Table 6. 1. Variables for remote monitor and control and their properties.	125

NOTATIONS

a (in Ch2)	Offset of GPS antenna from the center of gravity along front-rear	[m]
b (in Ch2)	Offset of GPS antenna from the center of gravity along left-right	[m]
a, b (in Ch3)	Center position of turning radius	[m]
d (in Ch2)	Lateral deviation	[m]
d (in Ch3)	Distance between Current path and Next path	[m]
$d_{t'-t_0}$	Euclidean distance between the position in t' and the position in t_0	[m]
Δd	Lateral deviation	[m]
E_i	An edge of a field	
h (in Ch2)	Offset of GPS antenna in the vertical direction	[m]
H_0	Maximum headland distance	[m]
j	index of dataset	
J	objective function	
k_d	Control Gain	
k_φ	Control Gain	
l_f	The distance of front wheel from the COG of the vehicle	[m]
l_r	The distance of rear wheel from the COG of the vehicle	[m]
l	The wheelbase of the vehicle	[m]
m	Vehicle mass	[kg]
I	Yaw moment of inertia	[kg*m ²]
k_f	Frontal tire cornering stiffness	[N/deg]
k_r	Rear tire cornering stiffness	[N/deg]
δ	Front-wheel steering angle	[deg.]
N_0	Total number of the original dataset	
N	Total number of the current dataset	
P' (in Ch2)	Position of the center of gravity of the tractor	
P (in Ch2)	Raw position of GPS antenna	
$P_i^j(x, y, s)$	One point of the edge	
r	Coefficients of the vehicle model	
R	Predicted turning radius of vehicle models	[m]
R'	Measured turning radius	[m]
R_f	Turning radius of forward movement	[m]
R_b	Turning radius of backward movement	[m]
s	Coefficients of the vehicle model	

$\{t, \varepsilon, h\}$	Weights of efficiency (t), accuracy (ε) and headland occupancy (h)	
V_{max}	Maximum speed	[m/s]
V_{min}	Minimum speed	[m/s]
v	Turning speed	[m/s]
w_j	Weights of data in the dataset	
x_i, y_i	Corrected positions of robots	[m]
θ	Forward turning angle	[deg.]
θ_{bias}	Bias of forward turning angle	[deg.]
θ_p	Pitch angle	[deg.]
θ_r	Roll angle	[deg.]
Φ	Yaw angle	[deg.]
Φ_d	Desired heading angle	[deg.]
$\Delta\Phi$	Deviation from Φ_d	[deg.]
$\Delta\Psi$	Steering angle	[deg.]
α, β	Coefficients of vehicle model indicating the relation of turning radii in different directions	
η	Weight of the results in the data set	
μ	Learning rate of the dynamic turning method	
$\varepsilon_{p_i^j p_i^{j+1}} v$	The length of a line segment	[m]
φ	Vehicle orientation	[deg.]
ω	Yaw rate	[deg./s]
v_x	Longitudinal velocity	[m/s]
v_y	Lateral velocity	[m/s]
β	Slip angle	[deg.]

ACRONYMS AND ABBREVIATIONS

AI	Artificial Intelligence
AR	Ambiguity Resolution
AFA	Asian Famers' Association
CAN	Controller Area Network
CANBUS	Controller Area Network Bus
CB	Circle-back Turning
CC	Continuous-curvature
CEP	Circular Error Probable
COG	Center of Gravity
CLAS	Centimeter Level Augmentation Service
CLAS-E	Centimeter Level Augmentation Service for Experiment
CMAS	Centimeter-class Augmentation System
DGPS	Differential Global Positioning System
DCB	Dynamic Circle-back
ECU	Electronic Control Unit
E-W	East-West
EKF	Extend Kalman filter
FAO	Food and Agriculture Organization of the United Nations
FOG	Fiber Optic Gyroscope
FMS	Farm Management System
FMIS	Farm Management Information Systems
GDOP	Geometric Dilution of Precision
GEO	Geostationary Orbit
GEONET	GPS Earth Observation Network
GLONASS	Global Navigation Satellite System
GNSS	Global Navigate Satellite System
GPS	Global Positioning System
HMT	Hydraulic Mechanical Transmission
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technology
IGS	International GNSS Service
IoT	Internet of Things
ISOBUS	International Standard Organization Bus
IMU	Inertial Measurement Unit
INS	Inertial Navigation System

JAXA	Japan Aerospace Exploration Agency
KF	Kalman Filter
LEX	L-band Experiment
LTL	Lock-to-Lock
LQR	Linear Quadratic Regulator
LSM	Least Square Method
LTE	Long Term Evolution
MADOCA	Multi-GNSS Demonstration tool for Orbit and Clock Analysis
MAFF	Ministry of Agriculture, Forestry and Fisheries
MBB	Minimum Bounding Box
MCS	Master Control Station
MEO	Medium Earth Orbit
MGM-Net	Multi-GNSS Monitoring Network
MPC	Model Predictive Controller
MRS	Multiple Reference Station
NRTK	Network RTK
N-S	North-South Direction
PA	Precision Agriculture
PPP	Precise Point Positioning
PTO	Power-Take-Off
QZO	Quasi-Zenith Orbit
QZSS	Quasi-Zenith Satellite System
RTK-GPS	Real Time Kinematic Global Positioning System
RMS	Root Mean Square
SIP	Cross-ministerial Strategic Innovation Promotion Program
SOFA	State of Food and Agriculture
SSR	State Space Representation
TECU	Tractor's Electronic Control Unit
TSP	Traveling Salesman Problem
U-D	Up-Down
UI	User Interface
UTM	Universal Transverse Mercator
UKF	Unscented Kalman Filter
VRS	Virtual Reference Station
XML	Extensible Markup Language

Chapter 1 Introduction

1.1 The situation and challenges of agriculture

The most pressing challenge facing the world today is feeding a growing global population projected to increase from over 7.2 billion currently to over 9.1 billion by 2050 – one third more mouths to feed than there are today (FAO, 2015). Increased and sustained agricultural productivity is needed to meet the globally increasing demands for food and energy. The agricultural sector is going to produce 60% more food for the additional people that the Food and Agriculture Organization of the United Nations (FAO) predicts, and this objection has to be achieved despite the additional threats of arable lands scarcity and the increasing need of freshwater. Besides, the challenge is also compounded by other less predictable factors, such as the impact of climate change and a deteriorating natural resource base, and so on (FAO, 2015).

Limited arable land (only 11% of land suitable for agriculture) is a severe disadvantage of Japan's agriculture. Compared to Japan, the U.S. has 82 times as much arable land but needs to feed twice the population of Japan. The Ministry of Agriculture, Forestry and Fisheries (MAFF) predicts that Japan needs 4.6 million hectares of cultivated land to fulfill the 50% food self-sufficiency rate, which is 39% currently. Similar to other countries, the labor force in agriculture in Japan is dwindling as existing farmers get older and older, and few youths take up farming and their places. According to statistics of the Japanese Ministry of Agriculture, Forestry and Fisheries in Table 1. 1 (Ministry of Agriculture Forestry and Fisheries, 2015), it is evident that Japan is suffering the decreasing and aging of agricultural labor for at least a decade. More seriously, 63.5% of farmers are over 65 years old. The labor force will not only decrease because of the shrinking population but also because of the aging problem.

Table 1. 1. Number of agricultural management units and the average age of farmers in Japan

unit: thousand of management units/people

Year	2005	2010	2015
Number of agricultural management units	2,009	1,679	1,377

Population of farmers	5,556	4,536	3,399
Average age of farmers	63.2	65.8	66.4

With the modernization of agriculture, agricultural acreage per household in Japan increased from 1.9 ha (in 2005) to 2.5 ha (in 2015). In Hokkaido, agricultural acreage per household has increased by 12.9 percent in the past five years (from 2010 to 2015). In 2015, the average agricultural acreage per household in Hokkaido reached to 26.5 ha, and average agricultural acreage per household in other areas of Japan was 1.8 ha, with the increased rate of 14.8% from 2010. Until 2015, 57.8 percent of farming land was larger than 5 ha, and this value was 43.3 percent ten years ago (in 2005). Compared to the data in 2010, the rate of larger than 100 ha farmland increased all over Japan, which respectively increased 27.1 percent in Hokkaido, and 39.6 percent in other areas (MAFF, 2015). Therefore, we can partially conclude that the shortage of labor force in farming work seems more severe with the expansion of scale as well as the aging problems of farmers.

1.2 Solutions in policies and technology

Facing the problems of decreasing labor force and increasing demands, the agricultural industry has to be less depending on manual labor and rely on new technologies to boost labor efficiency. Facing these challenges head-on, the Japanese government has realized that new developments would occur through the interaction of traditionally distinct disciplines. The Cabinet Office has taken initiatives to resolve these problems. For example, eleven Cross-ministerial Strategic Innovation Promotion Programs (SIPs) are under development to promote research and to foster innovation with applied, cross-disciplinary technologies (Council for Science and Technology Policy, 2017). The Technologies for Creating Next-Generation Agriculture, Forestry and Fisheries SIP has invested in the development of new technologies and products for specific applications.

State-of-the-art technologies like Artificial Intelligence (AI), internet of things (IoT), Information and Communication Technology (ICT) and robotics can be used to solve practical problems in agriculture by improving productivity and quality of farming. Many academics and industrials have invested in it. Different from industrial applications dealing with simple, repetitive, well-defined, and a priori known tasks, automation in agriculture requires advanced technologies to deal with the complex and highly variable environment and produce (Nof, 2009). The autonomous agricultural vehicle/robot is mainly used for an agriculture operation, such as tillage,

seeding, spraying, harvesting, transporting, etc., without a human driver. For the autonomous vehicle to execute farming work, it needs to require the knowledge of the localization, kinematic, and dynamic states of the vehicle, and to control the speed and the steering of the vehicle. Generally, an autonomous agriculture robot is outfitted with multiple sensors for sensing the surroundings and working along predetermined paths. In more advanced systems, the robot generates reference trajectories in a map stored in its memory and localizes itself to the map. It is also possible for the robots to map the environment surrounding the vehicle and then generate the trajectories using techniques such as simultaneous localization and mapping (SLAM) (Siciliano and Khatib, 2008).

The position localization and sensing can be done using the absolute positioning system or using local makers such as land features. Different sensors are selected in the automation system to meet the accuracy requirements for different work. A typical sensor package includes one or several sensors such as machine vision, infrared, ultrasound, odometer, inertial measurement unit (IMU), laser scanner, infrared, Global Positioning System (GPS), and so forth (Billingsley and Schoenfish, 1997; Kaizu and Imou, 2008). The GPS operated by the USA is the most common system of absolute positioning. The receiver calculates its position by measuring the distance between itself and there or more GPS satellites (Nof, 2009). Infrared and ultrasound are generally used for obstacle detection, such as scenarios like inner-row navigation and the automation of orchard robots. Machine vision, a single camera or a multi-camera system, is widely used to recognize the surrounding scene under various weather conditions. Similar to a laser, radar, infrared and ultrasound sensor, the machine vision system can be used for relative positioning in row crops, the position and orientation of the vehicle relative to crops. Relative positioning sensors also include odometer and IMU. No single sensor will work for vehicle navigation under all conditions. Combining data from multi-sensor provides better results than the data from the individual sensors. Sensor fusion methods are widely used in robot vehicles, particularly between absolute positioning (e.g., GPS) and relative positioning sensors (e.g., IMU). In the case of agricultural vehicles, different sensors combined with automatic decision provide a robot with ever-increasing sensing capabilities and mighty onboard processing power. Another approach is to use road/farm infrastructures as active facilities that can communicate with all robots and share information in real-time (Siciliano and Khatib, 2008).

Several primary manufacturers have been actively seeking to research on an autonomous tractor. Kinze Manufacturing Inc. (USA) developed an Autonomous

Harvest System that the tractor or the grain cart can run between the combine and unloading site without an operator. The tractor is equipped with cameras, radar, GPS, and guidance equipment to navigate through the field safely. The combine operator oversees it, and issues command to the tractor. The Autonomous Tractor Corp. unveiled its concept vehicle, Spirit tractor, with tracks and no cab. The Spirit operates with a laser and radio navigation system and is designed low-cost, efficient, and reliable. John Deere launched its iTEC Pro system - a global positioning technology based guidance product. Based on satellite signals, iTEC Pro automatically controls the tractor and implement functions with ease and precision. It includes forward speed changes, power take-off (PTO) switching and the raising or lowering of the hitch at precisely the right time and position in the field, and steering the wheel at the headland area. Usually, such a robot could perform farm work more precisely than an experienced human under straight-line conditions. These self-driving machinery cost about 2 to 3 million yen more than conventional vehicles. Given the usually small size of fields in Japan, this additional cost may prevent most farmers from acquiring these new technologies. New technologies and platforms should be developed for cheaper systems. The autonomous system has opened a new promising market. Japan's smart agriculture market in FY2016 has grown up to 10,420 million yen, 107.2% of the size of the previous fiscal year. It is expected to achieve 33,339 million yen by FY2023 (Yano Research Institute Ltd., 2017).

1.3 Objectives

Society 5.0 is the concept of future social development put in place by the Japanese government. Consistent with this concept, this study is to develop efficient robot tractors that can work on a farm automatically to relieve human from tedious and harmful work. Fundamental technologies for intelligent agricultural vehicles should be developed to deal with the complex and loosely structured farming environment (field shape, size, soil properties, and weather conditions). The cost of the automated vehicle should be reduced in order to fit the relatively low profits of farming work. To increase the productivity of the field and to control the robot accurately, the research is conducted as follows:

- 1) The feasibility of using the Japanese Quasi-Zenith Satellite System (QZSS) in precise agriculture situations is established. Instead of a real-time kinematic global positioning system (RTK-GPS), the Centimeter Level Augmentation Service (CLAS) of the QZSS is used to provide positioning information. In

this way, it offers the potential to eliminate the reliance on dense reference stations and the mobile phone network, which is not practical in some rural areas and not economical for most farmers.

- 2) A field is represented in an efficient and informative way. The navigation map covering both the main working area and the headland area consists of parallel navigation paths and transfer paths. Headland turning algorithms are well analyzed and improved to increase the efficiency of a robot tractor.
- 3) In order to increase the accuracy of estimation and the stability of motion control, machine learning control methods are developed in this research. Robot tractors can follow the predesigned navigation straight and curve paths within 5 cm accuracy.
- 4) A field management system is developed for the administrator to control in remote and to monitor the robot tractor working at the field. Farmers can also access the server to check work records of tractors, as well as editing the working plan through a smartphone or a PC. This research presents a part of functional architecture and provides an operational example of the management system.

Chapter 2 Navigation of a robot tractor using the Japanese Quasi-Zenith Satellite System

This chapter introduces the information about the tractor platforms and navigation sensors used in this study. Ground-based sensing system and global navigation satellite system (GNSS) are two widely used methods of developing autonomous vehicles. In this study, GNSS and IMU are used to calculate the position and attitude of the tractor in real time. Instead of an RTK-GPS, the CLAS of the QZSS is used to provide centimeter-level positioning information. The QZSS transmits augmentation information through an L6 signal to enhance the positioning accuracy of the GNSS. Besides accessing the augmentation information through the L6 signal using a commercial QZSS receiver, this paper also introduces a method for using CLAS with a dual-frequency receiver that cannot receive the L6 signal. Stationary and dynamic positioning experiments prove that the QZSS can improve the accuracy and availability of the current GNSS. Compensating for the biases of the CLAS positioning results relative to the current GNSS, a robot tractor works along with GNSS-based navigation within 5 cm accuracy.

2.1 Research platforms

2.1.1 Robot tractors

Tractors used in this study include two four-wheel-type tractors, EG83 and EG105, and two half-crawler-type tractors, EG453 and EG105-C (YANMAR Co., Ltd., Japan) as shown in Fig. 2. 1. Rotaries are attached to the EG83 and EG453. As shown in Fig. 2. 2, the control unit of the tractor system includes an engine control unit for controlling the engine speed. Hydraulic Mechanical Transmission (HMT) is the transmission system of the tractor, which controls the operation of the PTO. Meters are used to measure electricity, the amount of fuel, the velocity, and so on. The Tractor's Electronic Control Unit (TECU) is used for transforming the data between two Controller Area Network (CAN) buses and controlling the hydraulic power auto-steering unit and the implement related control units. Besides, the TECU controls the engine speed, the velocity, PTO, and the implement.



Fig. 2. 1. Experimental tractors.

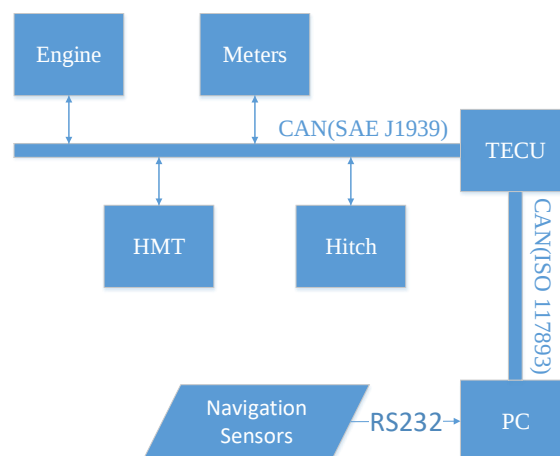


Fig. 2. 2. The control unit of the tractor.

2.1.2 The attitude sensor

The attitude sensor refers to the IMU, which is always used in the Inertial Navigation System (INS). IMU is an electronic device that can measure orientation, and gravitational forces, by a combination of accelerometers, gyroscopes, and magnetometers. Mostly, an IMU contains 3-axis (pitch, roll, and yaw) gyros and 3-axis accelerometers. In addition, the yaw angle (also called a heading angle or the

orientation) is an essential parameter for autonomous navigation. A fiber-optic gyroscope (FOG) is more accurate on the measurement of posture angle, but the price is much higher than a common IMU. In this study, the VN100 (VectorNav Technologies, USA) is used to measure the attitude of the vehicle.

Data collected from the IMU's sensors allows a computer to track a vehicle's position, using the dead reckoning method. Also, IMU is commonly used to correct the robot inclination caused by the uneven ground. Since the position sensor (such as GPS in this study) is usually mounted on the top of the vehicle, which is 2 or 3 meters above of the vehicle's center of gravity (COG). Therefore, the inclination of the vehicle on the uneven ground should be taken into consideration for accurate navigation. In this study, Euler's angles (roll, pitch, and yaw) are introduced to describe such orientation of the robot in 3-dimensional space. The following equations can correct the position data error of an RTK-GPS caused by the inclination of the vehicle:

$$P' = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = P - E \times A$$

$$A = [a \quad b \quad h]^T$$

$$E = \begin{bmatrix} \cos \theta_r \cos \phi + \sin \theta_r \sin \theta_p \sin \phi & \cos \theta_p \sin \phi & -\cos \theta_r \sin \theta_p \sin \phi + \sin \theta_r \cos \phi \\ -\cos \theta_r \sin \phi + \sin \theta_r \sin \theta_p \cos \phi & \cos \theta_p \cos \phi & -\cos \theta_r \sin \theta_p \cos \phi - \sin \theta_r \sin \phi \\ -\sin \theta_r \cos \theta_p & \sin \theta_p & \cos \theta_r \cos \theta_p \end{bmatrix}$$
(1)

where, P' is the position of a tractor's COG. P is the position of GPS antenna on the top of the tractor. a, b , and h are the offset of GPS antenna from the COG along the front-rear, left-right and height direction of the vehicle respectively. θ_r , θ_p , and ϕ are the roll, pitch, and yaw angle of the vehicle measured by the IMU.

2.2 Satellite-based positioning system

Global Navigation Satellite Systems (GNSSs) are the main positioning method in a wide range of applications, such as mobile phones, geodetic surveys, automatic vehicle navigations, and precision agriculture. GNSSs are constellations of satellites, such as the U.S. GPS, the Russian Global Navigation Satellite System (GLONASS) and other newly launched positioning systems, broadcasting satellite positions and navigation messages to the users for positioning. Generally, the positioning accuracy and robustness of multi-GNSSs are related to the number of visible satellites, the oscillation

of satellite orbits, the satellite clock accuracy, and ionospheric activities. If these factors are improved, the satellite-based positioning errors will be much smaller.

The RTK GNSS technique is widely used in applications that require centimeter-level accuracy of the positioning. It calculates the user's position in real time by processing the carrier-phase measurements of GNSS signals. However, it requires a relatively short inter-receiver distance to resolve the ambiguities (Lachapelle et al., 2000). For the single-reference station RTK-GPS method, the positioning accuracy decreases as the baseline increases. The network RTK (NRTK) approach or multiple reference stations (MRS) approach ideally provides positioning with errors independent of the receiver position within the network (Fotopoulos and Cannon, 2001; Lachapelle et al., 2000). As a derivative of the MRS approach, the virtual reference station (VRS) technique allows the rover to receive signals from a nearby virtual reference station within the coverage of the network. Ideally, the VRS-RTK approach can deliver centimeter-level positioning accuracy for an MRS network of reference stations separated by 50-70 km, which is greater than the conventional RTK method (Castleden et al., 2004). However, it does need an MRS network covering the area of interest. Besides, NRTK/MRS also requires a communication link, and costs are involved in receiving the correction data in many cases. For example, stable telecommunications are necessary to locate the rover in near real-time mode, including 3G or 4G mobile telephony that may not be reliable in remote rural areas.

The Precise Point Positioning (PPP) technique, which calculates accurate user positions using the precise orbit and clock of the GNSS, can be used instead of using ground stations as reference points (Castleden et al., 2004; Choy et al., 2015; Zumberge et al., 1997). Beginning in 2011, the Japan Aerospace Exploration Agency (JAXA) developed precise orbit and clock determination software for GPS, GLONASS, QZSS, and Galileo, which is called the "MADOCA (Multi-GNSS Demonstration tool for Orbit and Clock Analysis)". Takasu(2013) and Suzuki(2014) has confirmed the usefulness of MADOCA-PPP by conducting tests in an actual outdoor environment (Suzuki et al., 2014; Takasu, 2013). The surroundings of the travel route consisted of roadside trees in leafy areas and some buildings. From previous research (Choy et al., 2017; Ohshima et al., 2015; Suzuki et al., 2014), it can be seen that a long convergence time and float solutions are severe drawbacks for the application of PPP in autonomous vehicles. Synthesizing the positive characteristics of the PPP and RTK-GPS technique, PPP-RTK further improves the performance of PPP on a regional scale through the provision of a dense regional GNSS infrastructure (Choy et al., 2017).

The integration of these two techniques will lead to improving the position accuracy and convergence time.

The objective of QZSS is to enhance the current availability and performance of GNSS over its coverage. Besides transmitting positioning signals, QZSS also transmits precise correction signals to enhance the performance of GNSS. With the PPP and PPP-RTK methods, the QZSS broadcasts corrections realizing centimeter-class quality for its coverage through two channels of the L6 signal (Miya et al., 2014; Saito et al., 2011). In this way, it offers the potential to eliminate the reliance on dense reference stations and the mobile phone network, which is not practical in some rural areas.

2.2.1 Overview of QZSS and augmentation services

The QZSS, named “Michibiki,” developed by the JAXA, is composed of satellites in Quasi-Zenith orbits (QZO) and geostationary orbits (GEO) which draws an ‘8’-figure ground track in Fig. 2. 3. The QZO satellites come back to the same position in about one day and always maintain a high elevation angle. This 4-satellite constellation has been established, and the QZSS has continuously broadcast navigation and augmentation signals over the coverage area since November 2018 (Seigo et al., 2015; Suzuki et al., 2014).



Fig. 2. 3. Coverage of QZSS and its trajectory (Cabinet Office, 2017a).

Fig. 2. 4 shows the configuration of the QZSS as the regional augmentation system for the GNSS. The QZSS consists of a satellite system corresponding to the space segment, a ground system corresponding to the control segment, and a rover corresponding to the user segment. The control segment consists of the GNSS Earth

Observation Network System (GEONET), including 1,300 digital monitoring stations all over Japan, and the Multi-GNSS Monitoring Network (MGM-Net), including sixty monitoring stations located around the world. The raw data for the positioning signals from the GPS and QZSS are monitored at all times. Augmentation information including correction data, integrity data, and other information is uplinked from the Master Control Station to Quasi-Zenith Satellites (QZSSs) through the tracking station prepared in Okinawa to operate the QZS for 24 h a day. The correction service can be broadcast from the QZS directly to the receiver as well as through the 3G/4G mobile network. With the integrity data in augmentation information, the user can confirm the reliability of the positioning information in real time.

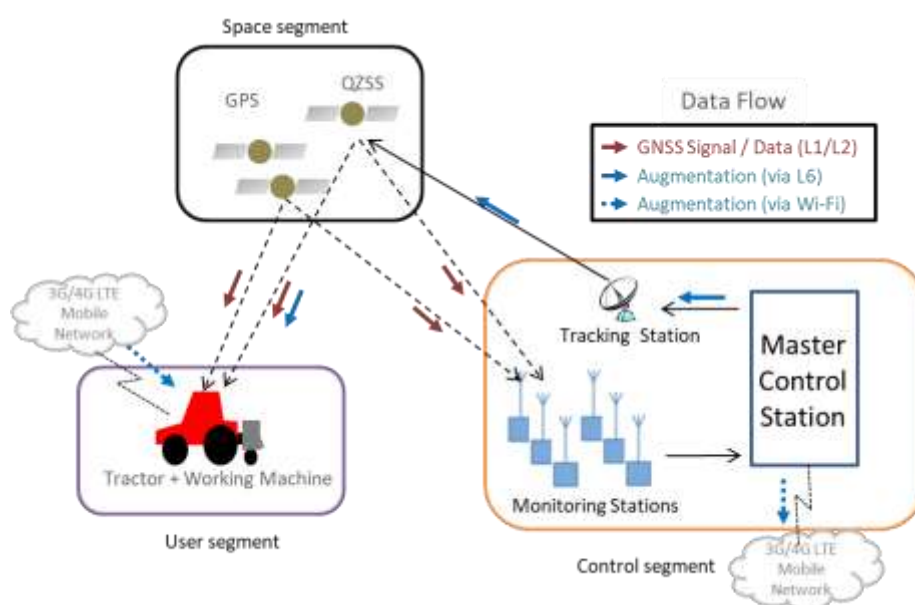


Fig. 2. 4. The overall architecture of the QZSS with augmentation service.

2.2.2 Complementary signals and augmentation signals

In order to complement to the GPS, QZSSs transmit four positioning signals with the same frequency as GPS, namely the L1 C/A, L1C, L2C, and L5 signals in Table 2. 1, and the QZSSs have clocks that are synchronized with GPS. It offers an increased number of available satellites in East Asia and Oceania region. In addition, it is easy to create an idealistic geometry deployment with small Geometric Dilution of Precision (GDOP), because at least one QZO satellite maintains a high elevation angle. Therefore, QZSS can be used in an integrated way with the current GPS for highly precise positioning. Besides, another five augmentation services using the L1-S, L1Sb, L5S, L6D, and L6E signals have been designed to allow for various positioning applications. From November 2018, a combination of QZSS satellites broadcast

navigation and augmentation signals over the coverage area 24 hours a day, seven days a week.

Table 2. 1. Transmission signals of QZSS (Seigo et al., 2015).

Signal	Satellite			Transmission service	Complement	Center frequency
	QZS-1 QZO	QZS-2 QZS-4 QZO	QZS-3 GEO			
L1C/A	○	○	○	Positioning	GPS	
L1C	○	○	○	Positioning	GPS	
L1S	○	○	○	Augmentation (SLAS)		1575.42 MHz
L1Sb	—	—	○	Augmentation (SBAS)		
L2C	○	○	○	Positioning	GPS	1227.60 MHz
L5	○	○	○	Positioning	GPS	
L5S	—	○	○	Augmentation Experimental Use		1176.45 MHz
L6D	○	○	○	Augmentation (CLAS)		1278.75 MHz
L6E		○	○	Augmentation (CLAS-E)		
S-band			○	QZSS Safety Confirmation		2 GHz band

The first Quasi-Zenith Satellite (QZS-1 nicknamed ‘MICHIBIKI’) was launched in September 2010. It is mainly used for confirming the potential of the system for navigation application, as well as identifying possible challenges in its utilization. The real-time CLAS was broadcasted from the QZS-1 using L6 signal, which was also called the L-band Experiment (LEX) signal (Collier et al., 2015). CLAS offers augmentation information for the main islands of Japan and the surrounding ocean areas to achieve several centimeters error positioning. In the current system, the CLAS adopts the State Space Representation (SSR) method to broadcast augmentation data using the L6D data channel of the L6 signal, which is not transmitted by GPS. So dedicated receivers are required to utilize the CLAS (Wübbena et al., 2005). Because of anticipated applications of the CLAS in East Asia and Oceanian region where users can receive QZSS signals, the L6E channel of the L6 signal broadcasts the Centimeter Level Augmentation Service for Experiment (CLAS-E) using QZS-2 to QZS-4.

1. CLAS

From 2010, the Japanese Satellite Positioning Research and Application Center (SPAC) developed the Centimeter-class Augmentation System (CMAS), which is the prototype of the CLAS (Wang and Noguchi, 2019a). However, the CMAS does not have a complementary function for GPS. The augmentation information was broadcast using the LEX signal of the QZS-1 (Saito et al., 2011). The LEX signal was renamed L6 in the latest QZSS. The CMAS uses the monitoring data of GPS and QZS from GEONET to estimate distance-dependent GNSS errors using the wide-area dynamic error model called State Space Modeling (Wübbena et al., 2005). Also, it generates augmentation data in SSR form to users through the LEX signal (Saito et al., 2011; Seigo et al., 2015). The correction information in SSR form can be applied to PPP-RTK positioning. The corrections of positioning errors are divided into two parts, the global items, and local items, in Table 2. 2. The global items indicate the corrections of errors caused by the satellites, including satellite orbit corrections, satellite clock corrections, and satellite phase biases. The local items include slant ionospheric delay parameters and vertical tropospheric delay parameters. All of the users within the coverage of QZSS share the same augmentation information for the global items. However, the augmentation information for the local items varies among different areas. Similar to the VRS-RTK system, the CMAS generates the correction of ionosphere and troposphere delays for a small area of the QZSS's coverage. Therefore, the CMAS could not be used simultaneously all over the country.

Table 2. 2. Correction information on positioning errors (Saito et al., 2011).

Item	Order	Correction information	Update interval [s]
Global item	E1	Clock error	5
	E2	Orbit error	
	E3	Signal bias error	
Local item	E4	Ionospheric delay error	3
		Tropospheric delay error	

As the technology succeeding the CMAS, the CLAS divides the service area of GEONET (covering the territory of Japan and nearby ocean areas) into 12 parts, as shown in Fig. 2. 5. In addition, artificial grid points are defined at 60 km intervals covering each area. The ionospheric delay and tropospheric delay are corrected to each grid point. The user corrects its observations and gives centimeter-accuracy solutions within 1 minute using the PPP-RTK positioning method (Miya et al., 2016). The positioning results obtained by utilizing CLAS neglect the crustal movement that is neglected by the VRS-RTK service in Japan. Therefore, coordinate transformations

are necessary to make a comparison or to use the positioning results of VRS-RTK and PPP-RTK together.

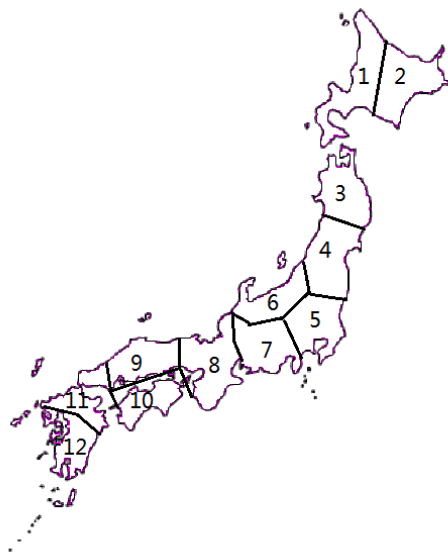


Fig. 2. 5. Area assignment of CLAS augmentation information (Seigo et al., 2015).

2. CLAS-E

CLAS-E is for verifying anticipated applications of the CLAS in the East Asia and Oceanian region. It was broadcasted by using the LEX signal of all the QZSs. Sixty monitoring stations located around the world had been established until the end of 2013, and 20 of them are operated in the Asia Pacific region (Sato et al., 2014). Those monitoring stations make up the Multi-GNSS Monitoring Network, which monitors the location and time information from the multi-GNSS at all times and transmits the data to the Master Control Station (MCS). Radio Technical Commission for Maritime Services (RTCM) message contains compressed observation data or models describing the observations from a network of several permanent reference stations. The classical PPP method, i.e., MADOCA PPP, uses correction messages from International GNSS Service (IGS) and MGM-Net operated by JAXA (Choy et al., 2014). Centimeter level augmentation data, including correction data, integrity data, and other information are uplinked from the MCS to the QZS through the tracking station (Saito et al., 2014). Because the LEX does not deliver the necessary information regarding a position with ambiguity resolution, the MADOCA PPP cannot perform ambiguity resolution. In addition, it usually takes a long time to estimate the effect of the ionospheric delay. The long convergence time is another drawback that limits the use of PPP. On the contrary, the precise point positioning with ambiguity resolution (PPP-AR) uses correction information from regional electronic datum points to achieve centimeter-level accuracy within 30 minutes, which is much faster than the

PPP method (Choy et al., 2017). The latest QZSS provides centimeter-level augmentation information as experimental service is broadcasted via the L6E data channel of the L6 signal. Applying the PPP method and/or PPP-AR method, the globally applicable augmentation information on satellite orbit, clock offset, and code/phase biases are available for QZSS users under its coverage shown in Fig. 2. 6 (Cabinet Office, 2017b).

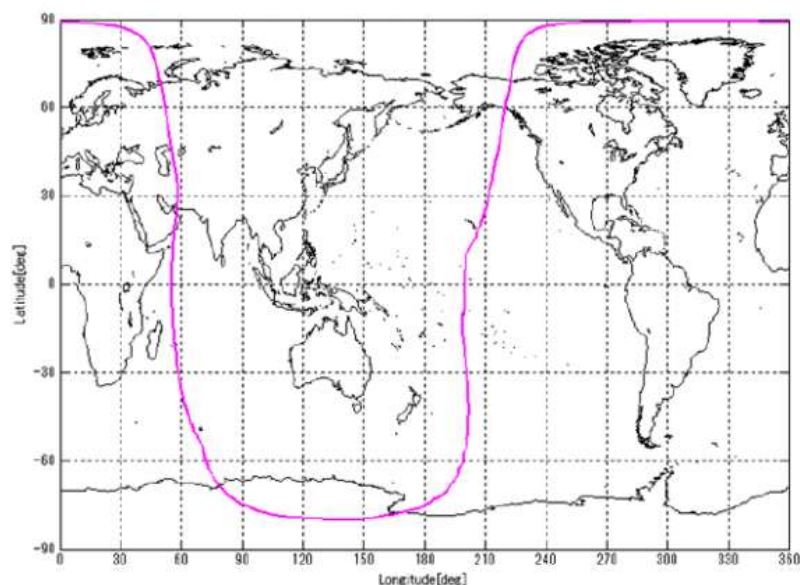


Fig. 2. 6. The service area of CLAS-E.

2.3 Application of QZSS to the robot tractor

Experiments in this paper are conducted under the cooperation of Japan government and related companies such as Mitsubishi Electric Corporation, Magellan Systems Japan, Inc., and Hitachi Zosen Corporation (Wang and Noguchi, 2019a). The LEX signal in this research indicates the L6 signal, including the L6D and L6E channel. The CLAS and CLAS-E are termed CLAS for simplification. The application of PPP-RTK method uses the CLAS, and the applications of PPP/PPP-AR uses the CLAS-E. In addition, the path planning and navigation algorithms used in the experiments will be introduced in the following chapters.

2.3.1 Application of PPP-RTK to a Robot Tractor

To evaluate the accuracy of the CLAS based positioning method, the antenna position is recorded by the VRS-RTK and the QZSS receiver simultaneously. The scheme of the system is illustrated in Fig. 2. 7. The VRS-RTK solution is derived from

the Trimble SPS855 GNSS receiver (T03). To be distinguished from other receivers, it is termed the VRS-RTK receiver. Two ways of receiving augmentation information from QZSS are evaluated in this research. Without additional hardware or software, the PPP-RTK solution is obtained from the GNSS receiver (T04) with centimeter accuracy. It is a multi-frequency receiver that can decode the L6 signal in real time. It is termed the CLAS receiver, the type of which is the AQLOC developed by Mitsubishi Electric Corporation, Japan. Another GNSS receiver (T05), developed by SPAC and Mitsubishi Electric Corporation, is a prototype for CLAS technique evaluation. It is a dual-frequency GPS carrier phase measurement receiver, which cannot decode the L6 signal. Therefore, it represents the RTK receiver that uses the augmentation service of QZSS. Fig. 2. 8 illustrates a method of using the CLAS through the mobile network. Message-decoding software developed by SPAC can be installed on a laptop PC or a smartphone to access the augmentation message from the server. According to the decoded correction messages from the L6 signal and L1/L2 positioning signals from the GPS, the RTK receiver conducts a fixed solution of PPP-RTK with an accuracy of several centimeters. In this way, a dual-frequency GNSS receiver conducts the fixed PPP-RTK solution without the reliance of local reference stations. The QZSS is beneficial to both multi-frequency receivers and dual-frequency receivers in high accuracy positioning applications.

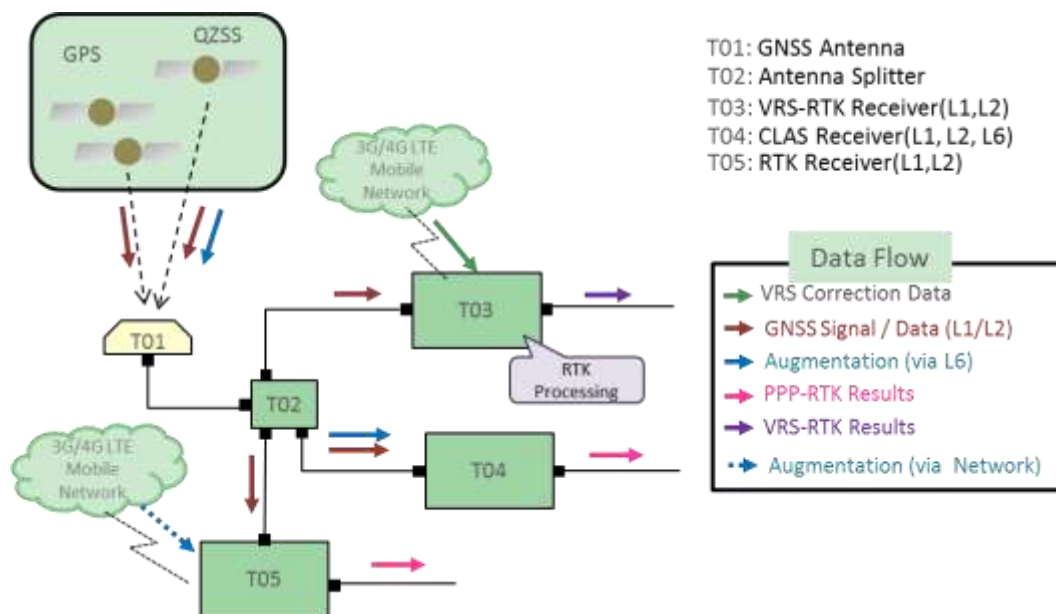


Fig. 2. 7. Scheme of the positioning system.

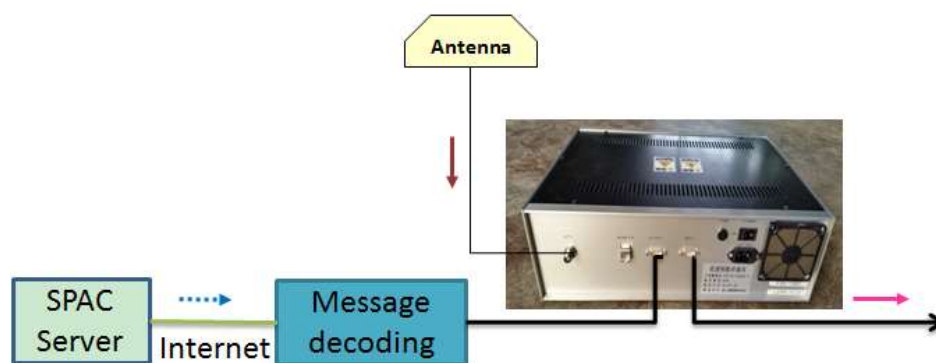


Fig. 2. 8. Receiving the augmentation information with the RTK receiver (T05 in Fig. 2. 7)

1. Real-time positioning test in static and dynamic conditions

Experiments under static conditions have been conducted since November 2015 in the Hokkaido area of Japan (Area 1 in Fig. 2. 5) to evaluate the accuracy and robustness of the QZSS-based positioning system. The outputs of the VRS-RTK receiver and the CLAS receiver are stored in the PC at the rate of 10 Hz, and the output frequency of the RTK receiver is 5 Hz. The difference between the PPP-RTK result and the VRS-RTK result is termed the bias of PPP-RTK. The mean of the bias values within a period is termed the offset of PPP-RTK. Since the positioning results of CLAS-based PPP-RTK neglect the crustal movement, an offset exists between the positioning results of VRS-RTK and those of PPP-RTK at the same position.

1) The accuracy of the QZSS-based positioning system in static conditions

The first series of experiments, including Test #1 and Test #2, was to monitor the availability and accuracy of CLAS information all day long. Test #1 was performed for 20.5 h from 10:55 July 27, 2017 (UTC+9 Time) to 07:31 July 28, 2017 (UTC+9 Time) using the RTK receiver. The correction information was transferred through the mobile network. Table 2. 3 lists the offset, root mean square (RMS) error, and maximum variation of the PPP-RTK relative to the VRS-RTK. E-W stands for East-West direction, N-S for North-South and U-D for Up-Down direction or elevation, respectively. The values of the fixed solution rate in the last column are all over 93%. The RMS errors of PPP-RTK are within 3.9 cm, and the averages are 3.2 cm and 2.4 cm in the N-S and E-W directions. In addition, the maximum horizontal variation was as large as 31.6 cm in the N-S direction. However, the offsets in the horizontal direction fluctuate within ± 1.5 cm. It shows that the PPP-RTK has RTK-like performance (3 cm horizontal RMS error). Filter and sensor fusion methods are necessary for robot control when the maximum variation is taken into consideration.

Table 2. 3. Accuracy analysis of CLAS in static conditions (Test #1).

Time		N-S [cm]			E-W [cm]			U-D [cm]			Fix rate
start	end	offset	RMS	max.	offset	RMS	max.	offset	RMS	max.	
10:55	11:56	-39.7	2.3	10.1	31.9	1.0	5.1	-7.0	7.8	24.7	0.99
11:57	13:43	-39.9	3.5	17.7	33.1	3.2	11.1	-6.1	7.2	25.9	0.97
13:44	16:37	-38.9	3.1	15.5	33.4	2.0	10.2	-7.8	8.2	57.1	0.97
16:37	23:18	-40.3	3.9	31.6	33.1	2.6	22.7	-10.0	8.7	32.1	0.95
23:23	7:31	-40.2	3.4	26.7	33.7	3.1	29.2	-9.5	9.3	56.5	0.93
Average		-39.8	3.2	20.3	33.0	2.4	15.6	-8.1	8.2	39.3	0.96

Test #2 includes three experiments lasting from 18:58 Oct. 24, 2017 (UTC+9 Time) to 10:42 Oct. 25, 2017 (UTC+9 Time). The positioning solutions using the CLAS receiver are listed in Table 2. 4. First of all, the offsets and RMS in the horizontal direction (E-W and N-S directions) fluctuated within ± 1 cm. Besides, the average offsets of the CLAS in the fifth row are -41.4 cm and 33.6 cm in N-S and E-W, respectively. Its positioning results are consistent with Test #1. The horizontal positioning accuracy of PPP-RTK using the CLAS of QZSS is within 3.4 cm (RMS) in the open-sky environment. It can be concluded that the CLAS still provides accurate correction information, even though the ionosphere activity varies from the day time to the night.

Table 2. 4. Accuracy analysis of CLAS in static conditions (Test #2).

Time		N-S [cm]			E-W [cm]			U-D [cm]			Fix rate
start	end	offset	RMS	max.	offset	RMS	max.	offset	RMS	max.	
18:58	21:00	-41.6	3.4	13.7	33.6	2.6	11.5	-8.4	9.3	34.9	1.0
6:40	8:42	-41.3	3.4	23.9	33.7	3.3	27.7	-11.9	10.3	49.4	0.99
8:42	10:42	-40.9	3.4	18.7	34.0	3.4	19.2	-9.4	5.6	26.9	1.0
Average		-41.4	3.4	18.8	33.6	2.9	19.6	-10.1	9.8	42.2	0.99

2) The accuracy of the QZSS based positioning system in dynamic tests

Two kinematic vehicle tests were conducted at an experimental farm at Hokkaido University, Japan, to evaluate the accuracy and robustness of real-time PPP-RTK using QZSS. The robot tractor worked autonomously based on a predesigned navigation map shown in Fig. 2. 9. The robot tractor started from the starting point and went along the transfer path (the red line), performed the tillage while working along four paths (blue lines), and then went back to the barn (endpoint) automatically along the other transfer path. In the transfer path, the velocity of the robot tractor was about 5 km/h. While working at the farm, the velocity of the tractor was about 2.5 km/h to 3.6 km/h.

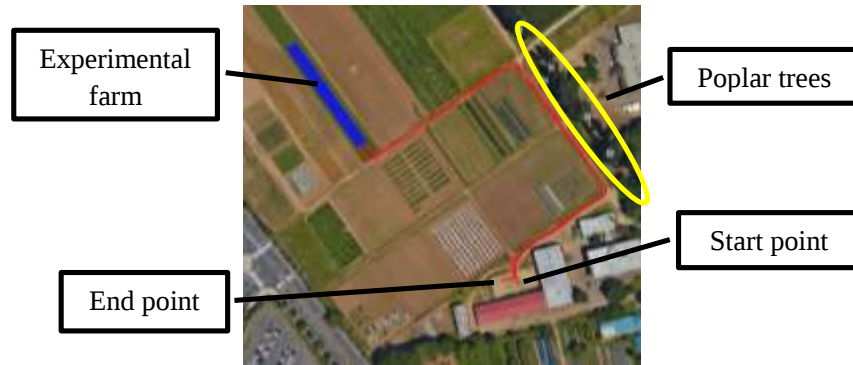


Fig. 2. 9. Navigation map for dynamic tests.

The two graphs in Fig. 2. 10 illustrate the satellite number and GNSS solution in the transfer path (from the starting point to the experimental farm). The number of visible GPS satellites decreased to 3 or 4 (Fig. 2. 10 (a)) when the tractor was near the trees. Therefore, the RTK receiver could not obtain a fixed solution. In contrast, there were at least four visible satellites when the CLAS receiver was used because the positioning signals from both GPS and QZS-1 were used. As a complement to the current GNSS, QZSS increases the number of visible satellites.

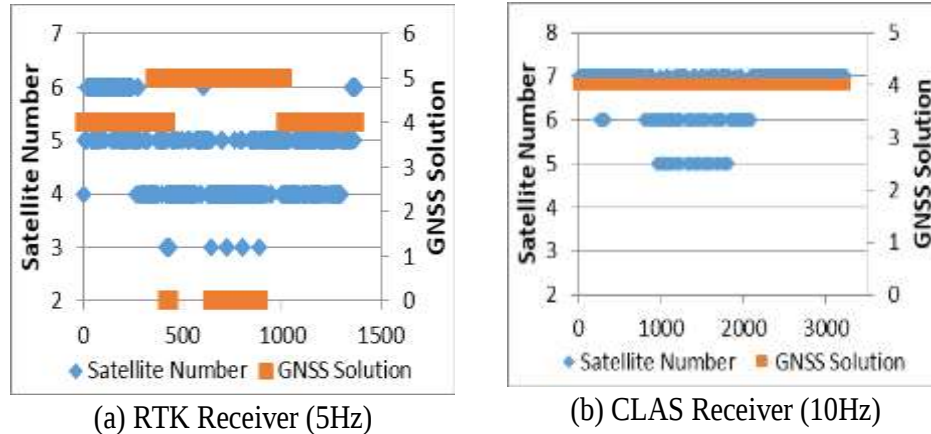


Fig. 2. 10. Satellite number and GNSS solution conditions during the transfer path (from the barn to the field). ‘5’ means the Float solution, ‘4’ means the RTK Fix solution, and ‘1’ indicates GPS Single solution.

The robot tractor was navigated by the RTK receiver and the CLAS receiver separately. The VRS-RTK fixed solutions were also recorded simultaneously to evaluate the navigation accuracy. The accuracy is calculated from the deviations of the robot tractor from the designed path. These deviations are termed “Lateral error_RTK” and “Lateral error_CLAS” in Table 2. 5. By compensating for the bias of the QZSS

relative to VRS-RTK, the robot tractor worked along with the navigation map within a 4.7 cm lateral error. The average navigation accuracy using the PPP-RTK is within 3.5 cm RMS. It means the robot tractor can perform precise farming work with the benefit of QZSS.

Table 2. 5. The accuracy of the robot tractor.

Path order	1	2	3	4	Mean
Lateral error_RTK [cm]	2.9	2.6	4.7	3.7	3.5
Lateral error_CLAS [cm]	2.6	3.4	3.4	3.1	3.1

2. Conclusion

By decoding the augmentation data from the L6 signal in real time, PPP-an RTK solution with centimeter accuracy can be obtained without dependence on local radio or mobile networks. The navigation signals and correction information help perform precise positioning in remote areas, canyons, and rural areas with tall trees.

Consistent with previous studies (Miya et al., 2014; Saito et al., 2011), the horizontal positioning accuracy of the real-time PPP-RTK solution is within 4 cm RMS. However, the maximum variation in horizontal directions is as large as 31.6 cm. Also, this research established the feasibility of using the CLAS-based PPP-RTK in precise agriculture situations. Compensating for the bias of PPP-RTK, the robot tractor worked along the route planned by the VRS-RTK system with an accuracy within 5 cm. When the receiver was near poplar trees, the GPS-based system could not obtain the fixed solution because there were few visible satellites. The results of this research indicate that QZS can be used in an integrated fashion with GPS satellites. Therefore, the QZSS-delivered CLAS is proved to be precise, accurate, and economical for use in precision agriculture.

2.3.2 Application of PPP method to a robot tractor

The antenna position is recorded by the VRS-RTK and the MADOCA PPP simultaneously to evaluate the accuracy of the MADOCA PPP using multi-GNSS (Wang and Noguchi, 2019b). The scheme of the system is illustrated in Fig. 2. 11. We have used a GNSS antenna (Magellan Systems Japan, Inc.), a GNSS receiver (Trimble SPS855, Trimble Inc.), and a QZSS receiver (MJ-3008-GM4-QZS, Magellan Systems Japan Inc.). Without additional hardware or software, the PPP solution is obtained from the QZSS receiver with centimeter accuracy by decoding the LEX signal in real time. The positioning messages from both receivers are in the format of

NMEA-0183. A laptop PC is used to record the positioning information through RS-232C port at the rate of 10 Hz.

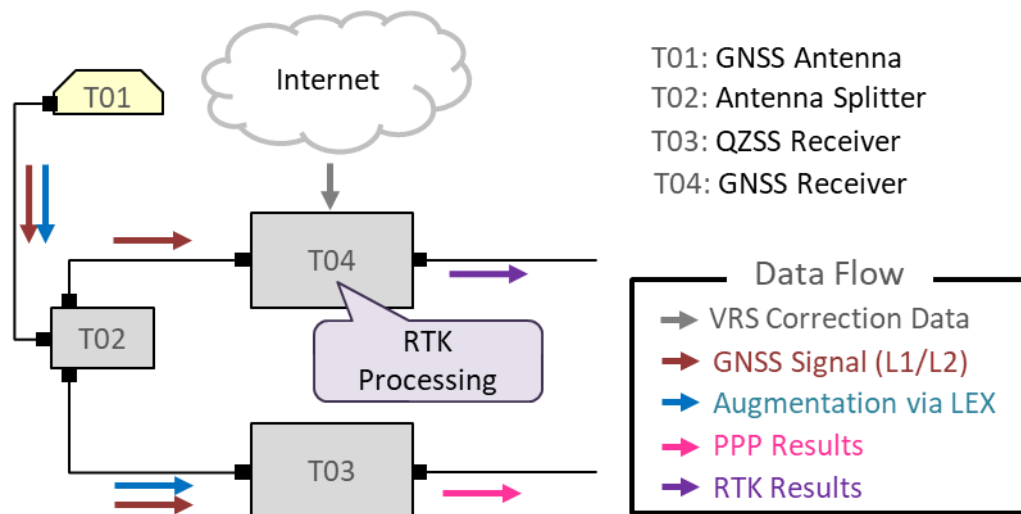


Fig. 2. 11. Scheme of the positioning system using the PPP method.

The experimental tractor used to evaluate the navigation method is a wheel-type tractor, EG105. We have conducted static and dynamic tests to evaluate the accuracy of real-time PPP using a LEX signal from the QZSS. The VRS-RTK fix solutions were recorded simultaneously as the real positions. The difference between the PPP result and the known coordinates of a position is termed the error of PPP. The mean value of the errors within a specific duration is termed the bias of PPP. The MADOCA-PPP result deviates from the VRS-RTK result with biases in the horizontal and vertical directions because crustal movements are not considered in the IGS05 coordinate system, which is used in the augmentation information from the QZSS.

1. The accuracy of the PPP solution in static conditions

The first static test (Test #1) was to evaluate the availability and accuracy of the augmentation service using QZS-1. The horizontal positioning errors of the PPP solution are illustrated in Fig. 2. 12. The positioning system initialized with errors at the green square and converged gradually to the dark area.

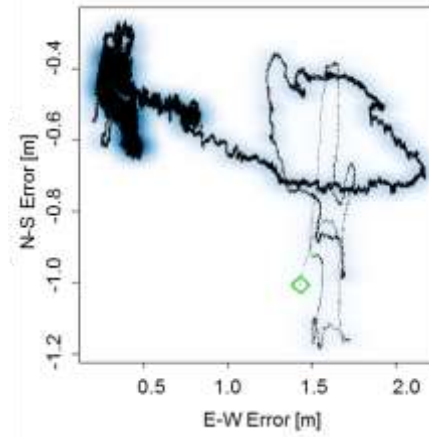


Fig. 2. 12. The horizontal error of PPP. The positioning system initialized with errors at the green square and converged gradually to the dark area. The blue shade illustrates the density of values.

The time-series positioning errors in the E-W, N-S, and U-D directions in Test #1 are illustrated in Fig. 2. 13. It shows clearly that the error of the PPP solution under static conditions fluctuate within 5 cm over time. According to previous studies (Choy et al., 2017), we assume that the horizontal solution from a standard PPP static processing to converge to 5 cm takes 120 min. However, the PPP solution drifts gradually over a long duration. The Root Mean Square (RMS), the bias (mean value of errors) in each direction, the drift of the bias, the maximum errors and the circular error probable (CEP) (Blischke and Halpin, 1966) listed in Table 2. 6 indicate the accuracy of PPP in Test #1. In this research, five centimeter RMS positioning accuracy is required by the robot navigation application. The drift of PPP solutions should be reset once the RMS of the data in the duration is no smaller than 5 cm. Therefore, the time intervals after convergence (from 120 min to 540 min in Test #1) are separated into five terms. Derived from Term 2, the duration of each term is within 90 min. Within 90 min's duration, both of the RMS and CEP at Term 3, 4, 5 are within 5 cm. And the maximum error is within 15 cm. It is as small as 9 cm at Term 6 when the duration is 60 min.

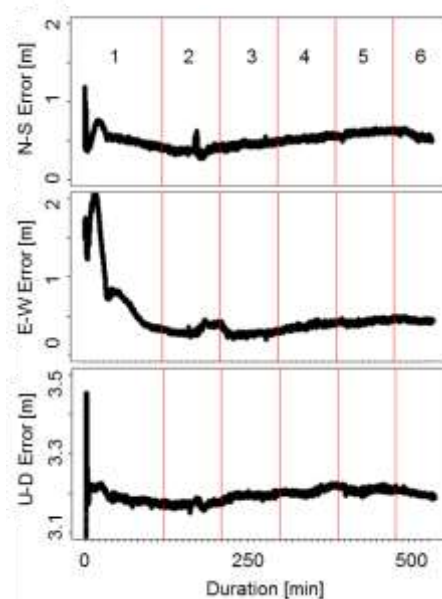


Fig. 2. 13. Time-series positioning errors of MADOCA PPP in a static condition (540 min). The experiment duration is divided into six terms by the vertical lines. The first term (from initialization to 120 min) is the converging period.

Table 2. 6. 3D positioning errors in the static test using MADOCA PPP.

Term	Duration [min]	Direction	RMS [cm]	Max. [cm]	Bias [cm]	Drift [cm]	CEP [cm]
2	90	E-W	5.0	12.3	32.6	0	5.5
		N-S	4.4	24.1	38.2	0	
		U-D	5.3	21.9	3172.4	0	
3	90	E-W	3.5	14.2	28.1	4.5	3.7
		N-S	2.9	10.6	44.7	6.8	
		U-D	5.7	20.3	3192.8	20.4	
4	90	E-W	3.5	9.4	36.9	8.8	3.2
		N-S	2.3	7.4	52.4	7.7	
		U-D	7.8	19.1	3205.4	12.6	
5	90	E-W	2.2	7.1	42.7	5.8	2.7
		N-S	2.5	11.2	59.8	7.4	
		U-D	5.6	18.5	3208.5	3.1	
6	60	E-W	1.6	5.1	44.8	2.1	3.0
		N-S	4.0	9.0	57.7	2.1	
		U-D	5.5	16.5	3200.1	7.6	

The other static test, Test #2, evaluated the repeatability of the PPP solution by initializing the system every 3.5 h (210 min). The test was repeated four times, termed #A, #B, #C, and #D. The RMS, the average bias, maximum errors, and the CEP of

each test are listed in Table 2. 7. Consistent with Test #1, the horizontal errors (RMS in the E-W and N-S directions) and the CEP of each test are within 5 cm 120 min after initialization. Therefore, we confirm that the MADOCA-PPP converges to centimeter-level in 120 min at an open sky environment. The positioning accuracy of MADOCA-PPP is about 5 cm within 90 min reset interval, and the result becomes better with a smaller reset interval. Besides, the change of PPP's biases shows that bias identification and drift reset are necessary after convergence. For example, the PPP receiver measures a landmark (such as a stick near the farm) after convergence in #A. Taking this measurement as the reference, the bias in #B can be identified by measuring the same landmark. The difference between the two measurements is the bias of PPP in #B. With this method, the unwanted drift of PPP after each initialization can be identified without relying on conventional RTK.

Table 2. 7. Positioning errors and offset of PPP solution in Test #2.

No.	Direction	RMS [cm]	Bias [cm]	Max. [cm]	CEP [cm]
#A	E-W	1.6	36.3	4.3	1.8
	N-S	1.5	48.2	5.5	
	U-D	2.8	3199.8	17.9	
#B	E-W	3.9	45.0	15.4	3.4
	N-S	2.1	40.3	11.4	
	U-D	5.6	3166.3	24.3	
#C	E-W	1.4	45.3	5.7	2.0
	N-S	2.0	48.9	7.3	
	U-D	3.4	3191.5	11.5	
#D	E-W	3.2	53.3	9.2	3.5
	N-S	2.9	60.0	8.8	
	U-D	9.3	3205.6	24.1	

2. The accuracy of the PPP solution in dynamic tests

In the dynamic tests, the robot tractor was navigated using the MADOCA PPP method. We proposed practical methods to overcome the limits of the PPP method in terms of its convergence time and long-term accuracy for robot tractor navigation. The robot tractor performed the tillage twice in one predesigned map with four working paths at the farm of Hokkaido University, Japan. Table 2. 8 summarizes the description of the two dynamic tests, marked as #3 and #4. It takes 145 min to obtain a convergence solution and to identify the bias in Test #3. The second dynamic test, Test #4, is conducted three hours after Test #3. During the intermission, the PPP solution is in the convergence condition. Therefore, we can take the immediate output or the mean value

of output during a short period to reset the drift. In this research, we record the output for 2 min at the static condition and calculate the mean value as the positioning result instead of the immediate output. The duration is the trade-off between efficiency and accuracy.

Table 2. 8. Description of dynamic tests.

Test	Bias identification time [min]	Start (UTC Time)	Stop (UTC Time)	Speed [km/h]
#3	145	9/17/2017,04:21	9/17/2017,04:35	2.0
#4	2	9/17/2017,07:54	9/17/2017,08:07	3.6

The trajectory of the robot tractor is shown using the Universal Transverse Mercator (UTM) coordinate system in Fig. 2. 14. After the identification of the bias in the PPP solution at the landmark (point A), the robot tractor worked automatically along the predesigned paths from the beginning of the map (point B) to the end of the map (point C).

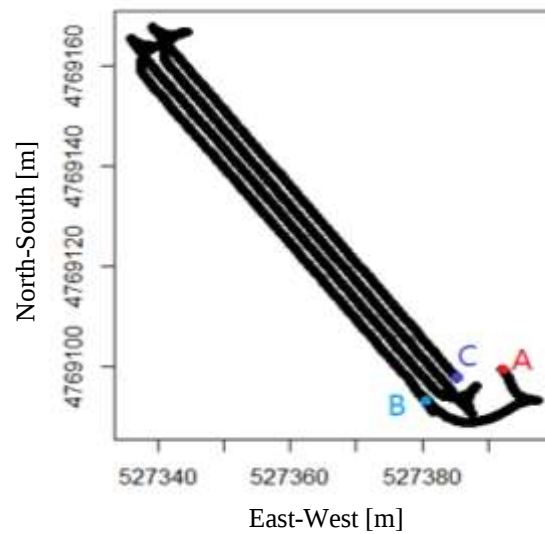


Fig. 2. 14. The trajectory of the robot tractor during the dynamic test. The receiver initialized at Point A. The farming work started at Point B and finished at Point C.

The lateral deviation is the variable that describes the deviation from the antenna to the desired map. The PPP provided positioning information for the robot tractor. The lateral deviation measured by VRS-RTK (Lateral deviation_RTK) illustrated the actual deviation of the robot tractor from the navigation map. Time-series lateral deviation and heading error of the robot tractor during Test #3 and Test #4 are shown in Fig. 2. 15 and Fig. 2. 16, respectively. The lateral deviation is within 10 cm, and the heading error is within 2 degrees during two tests. In addition, the accuracy is stable with the

increase of speed from 2.0 km/h to 3.6 km/h. Three intervals in each figure indicate the headland turning period. At the headland, the accuracy is not evaluated without a navigation map. The accuracy of Test #4 is summarized in Table 2. 9. The RMS lateral deviation in each path is within 4 cm at the speed of 3.6 km/h.

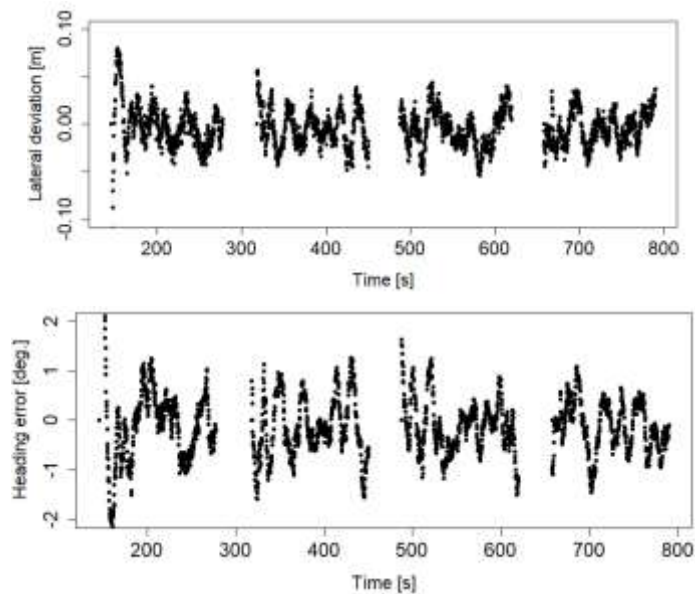


Fig. 2. 15. Time-series lateral deviation and heading error of the robot tractor during Test #3.

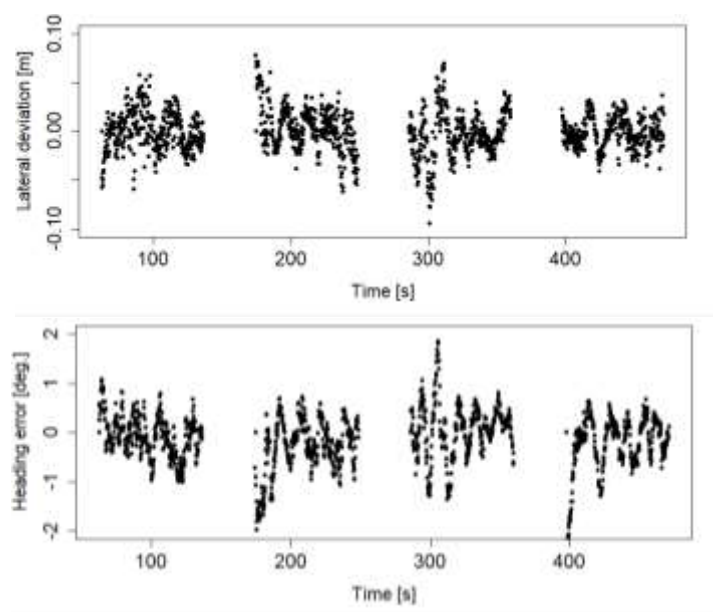


Fig. 2. 16. Time-series lateral deviation and heading error of the robot tractor during Test #4.

Table 2. 9. Accuracy of the robot tractor using MADOCA PPP as positioning signal

Path order	1	2	3	4	Mean
Lateral deviation_RTK [cm]	3.3	4.0	3.3	3.4	3.5

3. Conclusion

We establish the feasibility of using the MADOCA-PPP in precise agriculture situations. The long convergence time and float solutions are severe drawbacks for the application of PPP in autonomous vehicles. Conducting static experiments, we verify that the horizontal solution from a standard PPP static processing to converge to 5 cm takes 120 min. In addition, the bias of the PPP should be identified by measuring the landmark and comparing the current output to the reference value. For further research, the reference position can be the initial value of the PPP solution to decrease the convergence time. Because of the interference of tropospheric delay, the PPP result drifts gradually. If the PPP solution is in convergence, it takes a short period (such as 2 min in this chapter) to reset the drift. The drift reset interval should be within 90 min for the robot navigation application. In this way, the robot tractor can continuously perform farming work at a speed of 3.6 km/h with 5 cm RMS lateral deviation.

2.3.3 Application of PPP-AR method to a robot tractor

The augmentation data providing the corrections of GNSS satellites clocks and ephemeris errors can be broadcasted via the LEX signal of QZS1. In this study, the function of Hitachi Zosen's multi-GNSS analytic software, RTNet, is used to obtain the PPP-AR (Hitachi Zosen Corporation, 2015). Classical PPP methods use correction messages from the International GNSS Service (IGS) and Multi GNSS Monitoring Network operated by JAXA (Choy et al., 2014). The augmentation information does not include the corrections of ionospheric delay parameters and tropospheric delay parameters. Therefore, it takes a long convergence time to estimate the interference of ionosphere and troposphere. On the contrary, RTNet uses correction information from regional electronic datum points to achieve centimeter-level accuracy within 30 minutes, which is much faster than the PPP method (Choy et al., 2017). Since current commercial GNSS receivers can not process the LEX messages, a specific QZSS receiver (T03 in Fig. 2. 17) is used to decode the signals and get the PPP-AR solutions.

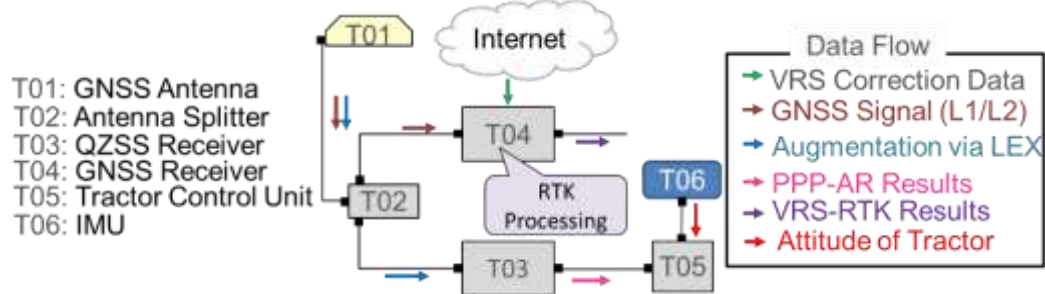


Fig. 2. 17. Scheme of the positioning system.

1. Accuracy evaluation in static conditions

The robot tractor performed farming work at a Homebush cane farm at Mackay in north Queensland where is a typical example of tropical areas with an increased level of ionospheric activity. Fluctuations of the radio wave in the ionosphere and troposphere in this area affect the connection between rover and satellites and reduce the positioning accuracy in the calculated distance between the user and the positioning satellites. Therefore, Mackay, as a typical example of tropical areas, is suitable to evaluate the robustness and the accuracy of the navigation system.

The trajectory of the robot tractor in one test is shown in the blue line in Google Earth (Fig. 2. 18 (a)) and the UTM coordinate system (Fig. 2. 18 (b)), separately. There are five paths from the right to the left of the figure, and the path space is set to 20 m. The working order of the paths is sequent, from Path 1 (near the arrow in Fig. 2. 18) to Path 5 with four times turns. The turning method is the U-turning with 2 m go forward straightly before the start of turning procedure (Torisu et al., 1996). The speed of the robot tractor during work is 6.6 km/h.

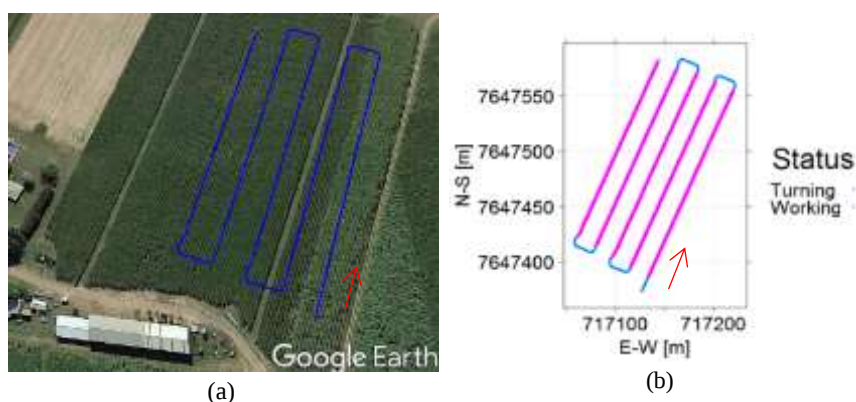


Fig. 2. 18. The trajectory of the robot tractor after one trail. (a) shows the trajectory in Google Earth. (b) shows the trajectory in the UTM coordinate system.

To evaluate the accuracy of the LEX signal based PPP-AR, the VRS-RTK fix solutions are recorded simultaneously during the movement of the tractor. The difference between the two results is termed as the error/offset of PPP-AR. During the test, offsets of the QZSS are calculated in the UTM coordinate shown in Fig. 2. 19. The shade represents the density of points in the heat map. The offset in East-West direction ranges from -3.2 cm to 2.6 cm. The offset in North-South direction ranges from -4.3 cm to 3.1 cm. The horizontal deviation is within 4.3 cm. The CEP is 1.1 cm, 50% rounds land within 1.1 cm of the average impact (Blischke and Halpin, 1966). The center of the estimated circle is (-0.6, -0.9). Therefore, it can be concluded that the PPP-AR results correspond to the VRS-RTK, because the CEP error is about 1.1 cm, and the maximum error is less than 4.3 cm. It means that the LEX signal from QZSS provides robust and accurate augment information for navigation.

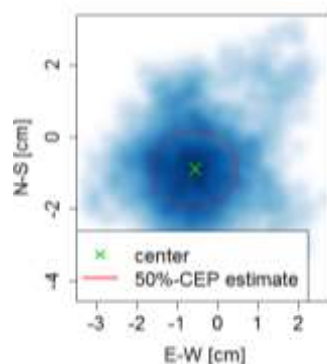


Fig. 2. 19. Offsets between QZSS and RTK-GPS coordinates.

2. Accuracy evaluation in the dynamic condition

To thoroughly verify that augmentation signals transmitted from the QZSS can be used in precision farming in Australia, experimental farming activities, including leveling, tillage, boom spraying, and fertilizer spreading were conducted all day long. In each experiment, the robot tractor should follow the same navigation map. During the experiment, QZS-1 was the only satellite broadcasting the augmentation signal. The elevation of QZS-1 viewed from Mackay is shown in Fig. 2. 20. It illustrates that the elevation is about 20 deg. around 14:00 in Australian Eastern Standard Time (UTC+10) when the signal of the LEX signal is the weakest period during a day. The spreading fertilizer experiment was conducted around 14:56 on December 9, 2016. The lateral error of each path during the experiment is listed in Table 2. 10. It shows that the maximum RMS of the lateral deviation value is 5.4 cm, and the minimum value is 3.2 cm. The average of the deviation in all paths is about 4.6 cm.

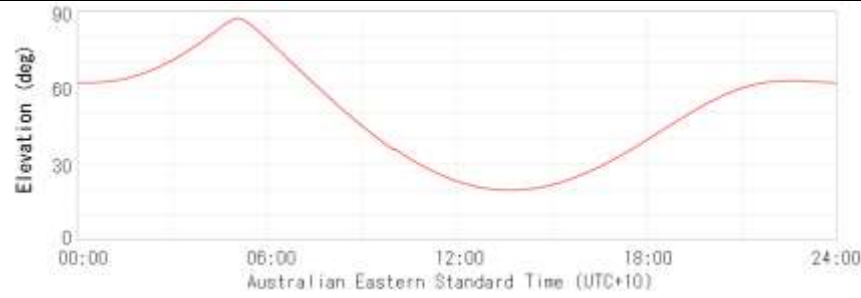


Fig. 2. 20. Estimated visibility of QZS1 from the view of Mackay, Queensland, Australia. Date: December 9, 2016

Table 2. 10. The lateral error (RMS) in the fertilizer spreading test.

Path order	1	2	3	4	5	Average
Lateral error [cm]	3.2	5.3	4.7	4.3	5.4	4.6

The lateral error of all the 5 paths of fertilizer spreading experiment is shown in Fig. 2. 21. The blank areas indicate the turning period from one working path to the next. The maximum lateral error was about 0.3 m after the first turning. The significant deviation and posture at the start of one path are mainly caused by the turning algorithm in limited headland space (Wang and Noguchi, 2016). From the figure, it can be seen that lateral errors are varying around zero. The lateral errors are significant after turning, but they can converge soon within 5 seconds. From these results, it can be concluded that the robot tractor can follow the working map within 4.6 cm lateral deviation at the speed of 6.6 km/h.

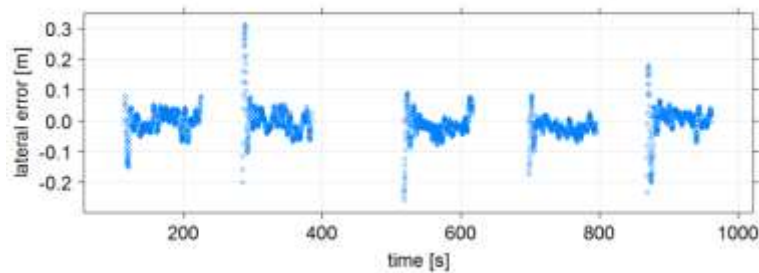


Fig. 2. 21. The lateral error of fertilizer spreading test.

Chapter 3 Field representation and path planning

Research on complete coverage path planning has been carried out in several application areas, such as floor cleaning, lawn mowing, painting, underwater searching, and so on (Hert et al., 1996; Hofner and Schmidt, 1995; Yang and Luo, 2004). The goal is to generate an efficient path (such as, a path of minimum length) for a robot that allows it to cover a given workspace in its entirety. In addition, the operations should be done in a manner that is coordinated with machine characteristics, size, the shape of the workspace, and specific patterns of field operation.

Planning an optimized route is an essential requirement as it is in other scientific disciplines. First, the operations time and distance affect the fuel consumption, with longer and more complex route requiring more fuel. The second, the weight of vehicle and farming implement and the steering maneuverers disturb the soil in vertical and horizontal directions. Therefore, controlled traffic farming can reduce the impact of the machine. In particular, the time lost and soil compaction due to non-working travel affect the field efficiency very much.

In terms of total non-working distance traveled, a large portion of time occurs during transferring from the initial position of the vehicle to the first track, from one track to the next, and from the last track to its desired final position (Bochtis and Sørensen, 2009). If the field is convex without any obstacles, the whole field is usually traversed by a series of parallel straight or contour paths that provide a complete field coverage. Once an optimal coverage direction is determined, the whole field is driven in that direction except headlands if needed (Hameed et al., 2010; Jin, 2009; Mi. Kise et al., 2001). In general, the best direction of paths is defined by the user experience or following the longest edge of the field. However, the shape and size of fields vary along with the rounded forests, road, mountain, lakes, and so on. In addition, shapes are far from convex (Jin, 2009). Searching an optimal direction driven by a cost function is more practical for the complex shape of the field (Oksanen and Visala, 2007).

Regarding the route planning for agricultural field operations, the research on path planning has been directed towards decomposing a complex-geometry field into sub-fields (convex blocks) and the optimum working sequence. Additional research also includes multi-vehicle routing methods (Burger et al., 2013). A prototype geo-web service named GAOS (Geo Arable land Optimisation Service) implements the element approach for spatial optimization of straight cropped swaths and field margins (De

Bruin et al., 2010). It can be operated by non-GIS experts from an ordinary web browser. Also, the optimization of agricultural operations in combination with field boundaries has been taken into account (Jin, 2009).

Identifying an optimal path that visits all sub-fields is an NP-hard problem, one way of solving the coverage path planning problem is to formulate it as a “traveling salesman problem” (TSP) (Oksanen and Visala, 2009; Sørensen et al., 2004). However, the start and the end positions are not always the same during the farming work, which is different from the traditional TSP with the same starting and ending node. In addition, the cost of each edge should not be constant according to the change of sub-fields order.

To agricultural robots, autonomous navigation does not only mean unmanned driving or auto-steering control, but it should also conduct the robot finish the desired work at the field. That is to say, an agricultural robot should be autonomous behave under constraints of working plans. For example, the robot should change the PTO and hitch at headland or in the field. The speed of the vehicle should also be adjusted in a different part of the field, as well as doing different work. Therefore, the navigation includes two meanings; the first one is to locate the position of the vehicle and to control it moving along desired paths in a navigation map; the second is to indicate the work to the robot while moving.

In this research, an optimal coverage path planning method is presented to improve field efficiency and, in particular, to utilize the advantages provided by automatically guided farming equipment fully. Also, several transfer paths are created to optimize the non-working distance and time consumption. Refiling or emptying the machine is not considered in this autonomous path planning. The sequence of the working order is also designated. When the field is divided into several sub-fields, each sub-area has to be visited once without discard. To use the merit of agricultural robotics, the backward movement along the navigation path is proposed in this research. According to the requirements, the graphs of working scenarios will be generated, and the optimized solution can be calculated to minimize the cost.

3.1 Field representation

The farming field is usually composed of the entrance, exit, working areas, and non-working areas. Non-working areas may contain several obstacles, such as huge rocks, roots, trees, and so on. Since traditional agriculture vehicles cannot move in the

vertical direction, the field in question is considered as a 2D closed area that may contain holes. A set of discrete geometric primitives (such as points, line segments, and polygons) are used for the representation of the field. Generally, the field and obstacle boundaries are obtained by driving a vehicle around the field/obstacles edge while logging the trajectories using an RTK-GPS (Hameed et al., 2010). The planned navigation map should cover the whole working area and avoid crossing boundaries. The navigation map, as well as the boundaries, are wholly defined by an ordered set of points with position and working information. The details of the field representation are given in the following section.

3.1.1 Boundaries

The boundaries of the field consist of two parts, the external boundary of the whole field and the internal boundaries that surround the non-working area. In addition, the boundary can be distinguished by “wall-like” boundary and “pass-through” boundary. For example, if a river or trees surround the field, the vehicle cannot get inside or outside of the field through this boundary. Otherwise, the vehicle can pass through the boundary without additional turn or lane change operations. The identification of the boundary condition is essential for designing the entrance/exit of the field and optimizing transfer paths, including headland turns.

Let B denotes the family sets of E_i , representing the boundary of an object, $B = \{E_1, E_2, \dots\}$. For an obstacle, its boundary is represented by one edge. Sequential pair-wise connected boundary points describe an edge, i.e., a point of the edge E_i can be referred to $P_i^j(x, y, s)$, $j \in |E_i|$, and (x, y) denotes the coordinates of the point P_i^j in a 2D coordinate system, which can be either local or global. Moreover, s (with a value of 0 or 1) indicates the vehicle can pass through the boundary or not. For the boundary of the working area, each edge satisfies the criterion that the curvature is no larger than a threshold ϑ . This requirement can be expressed by:

$$\left| \Theta \left(\varepsilon_{P_i^j P_i^{j+1}}, \varepsilon_{P_i^{j+1} P_i^{j+2}} \right) \right| \leq \vartheta, \quad (2)$$

$$\forall i \in \{E_1, E_2, \dots\}, \forall j \in |E_i|,$$

where $\Theta: R^3 \times R^3 \rightarrow [0, 360)$ is an operator that returns the angle between two line segments (Hameed et al., 2010). The length of a line segment, $\varepsilon_{P_i^j P_i^{j+1}}$, is about half of the width of the vehicle in this research. Following a curved path causes damage

to the farming implements such as a seed planter and a rotary. At the headland, the vehicle should follow the edges of the field. Therefore, the curvature of the edge, as well as the curvature of the path, should be limited. The length of the edge is given by:

$$l_{E_i} = \sum_{j=1}^{N-1} \varepsilon_{p_i^j p_i^{j+1}} = \sum_{j=1}^{N-1} \|p_i^j - p_i^{j+1}\|, \quad (3)$$

where $\|p_i^j - p_i^{j+1}\|$ denotes the Euclidean distance between any two points in an edge E_i . N is the total number of points in the edge E_i .

3.1.2 Main working direction

The field is represented as a polygon that consists of straight-line segments or boundaries. The main working direction is usually determined by the farmer's experience and the shape of the field. If the field is convex and it does not contain any obstacles, all pairs of points inside or at the boundary of the field can be connected with a straight line without crossing boundaries of the polygon. The selection of the main driving direction based on the longest boundary of the field has been a rule of thumb for farmers. In this research, the main working direction is determined by two steps. First of all, the main working direction is set as the same as the length of the minimum bounding box (MBB) of the field boundary. Then, the field is filled with navigation paths parallel to the main driving direction, as shown in Fig. 3. 1. The intersection between each navigation path and the field boundary are then obtained. The parts of the navigation path outside the boundary are eliminated (Hameed et al., 2013). The second step is to compare the length of the field longest boundary with all the generated paths. If the edge is more prolonged than paths, the main driving direction is along the edge and navigation paths should be regenerated to parallel to the edge. As an example, the bold black line shown in Fig. 3. 1, represents the field outer boundary and boundaries of obstacles area. The bold dashed line represents the boundary of the obstacle area where the robot can pass through without working. The dashed square represents the MBB of the boundary. Generated paths are shown in black straight lines. If a path intersects with a “wall-like” obstacle area, then it can be divided into two independent paths, each path having two endpoints with one endpoint located on the field boundary and the other endpoint located on the boundary of the obstacle. The longest path (shown in the red line) is longer than all the boundaries of the field, so the MMB determines the main direction in this case.

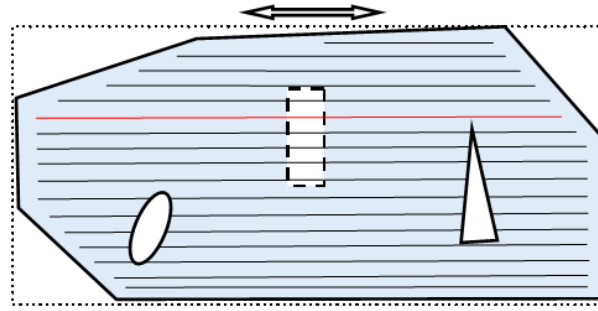


Fig. 3. 1. An example of generated navigation paths using the MMB method.

If the polygon is non-convex, the MBB method can be used to initialize a main working direction as reference for a complex field. For each convex sub-field, the working direction can be determined using the MMB method.

3.1.3 Field splitting

As shown in Fig. 3. 1, some of the navigation paths generated in the first stage intersect with in-field obstacles, and they are divided into independent paths. Given a non-convex field, navigation paths will also intersect with the field boundary with more than two points. Therefore, the field should be split into sub-fields by clustering the navigation paths. Navigation paths in one cluster construct a new sub-field, which should be convex. As shown in Fig. 3. 2, the field is split into seven sub-fields. For each sub-field, the main working direction should be rearranged, and navigation paths should be created in the updated direction. For example, in sub-field No. 3, the navigation paths should be along with the left edge, because it is longer than the length of MBB of the boundary. The optimization of the main working direction in sub-field 3 is to diminish the path number, as well as the headland area. The final results of field splitting after working direction optimization are shown in Fig. 3. 3. The non-working distance decreases with less the headland turning times and distance after optimization.

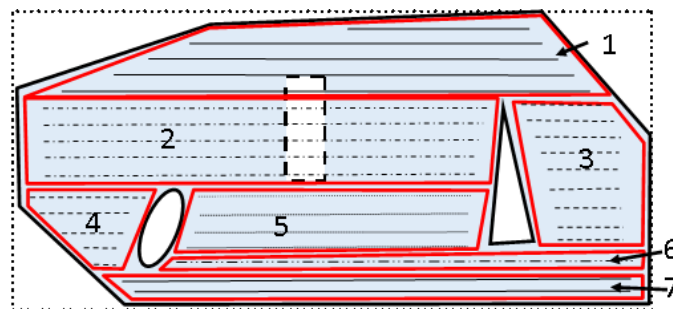


Fig. 3. 2. Sub-fields generation in a field with obstacles.

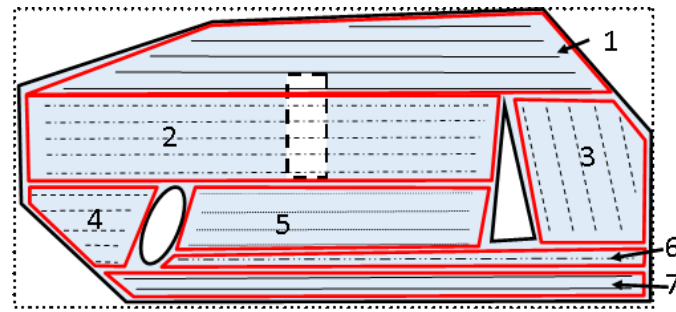


Fig. 3. 3. The effect of the main working direction on each sub-field.

3.1.4 Path planning for a convex field

The field is divided into two parts: the main working area and headland areas, which include upper headland area and lower headland area. Headland areas at the beginning/ending of the navigation path are necessary for the field robot to turn to the next path. The headland area near the start position is called a lower headland, and the opposite area is called an upper headland. The complete coverage path planning problem for a convex field refers to the process of generating parallel tracks to cover the main field area and the headland areas and the process of assigning path order. Fig. 3. 4 shows six paths (black blocks) at the main working area and two paths (blue dots) at each headland. The moving direction from path 1 to path 6 is along with the arrow. The whole field is driven in that direction except for headlands. The navigation paths at headland are parallel to the edge of the field, which may be curved. The number of paths is determined by the width of farming equipment and the width of the area.

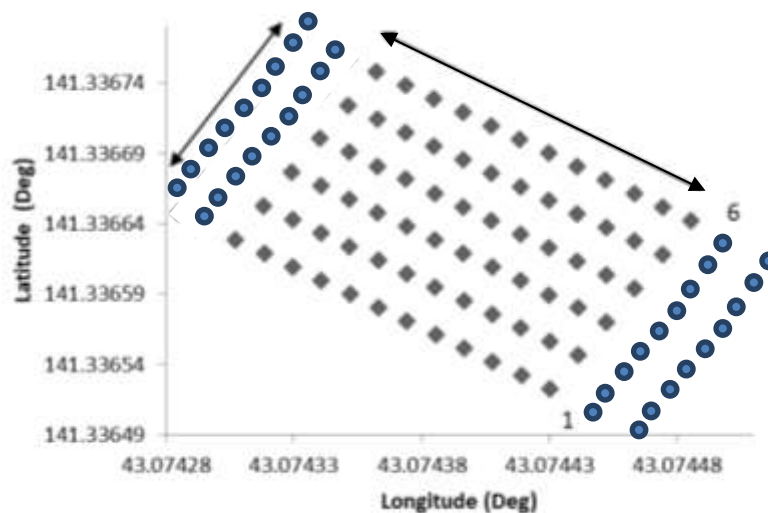


Fig. 3. 4. Scheme of navigation paths.

For common agricultural operations, it is required to perform two essential functions simultaneously: steering the vehicle and operating the equipment (Nof, 2009). Therefore, a navigation path in this study consists of two parts; one is navigation points, which distribute in the area of the farm as an array. The position of each navigation point is expressed in latitude and longitude format. The other part is the code indicating the settings of the tractor and the operations at that position. Traditionally, the operations of planting, weeding, spraying, and fertilizing the field are elementary. That is, the robot tractor does or does not execute the function during a working path. The operations and settings can be designated by the working content and farmer's experience. However, it will be not informative enough for more complex scenarios, such as variable-rate fertilization. In addition, it is impossible to designate the operations for the whole field manually. The robot should behave more intelligently for economic and environmental considerations.

Take the tillage as an example. The navigation points share the same code in the same path, except for the obstacle area. The hitch should be raised to the upward position, and the engine speed should be decreased when the tractor passes by non-working areas. In this research, the specific settings of the tractor and operations, including the path order, engine speed, vehicle speed, PTO (ON/OFF), and the position of the hitch, are designated by the code. Fig. 3. 5 shows the way of decoding in this study. Firstly, the code is transformed from decimal to binary. Secondly, the binary code is decoded from right to left according to the format, and the lowest bit indicates the working condition. The code of working condition is 0 if the navigation point is at the outside of the boundary or if the point has been visited. That is to say, the robot tractor performs tillage work near a point, and then the code of working condition changes to 0 after tillage. The robot can resume work according to this flag in case of any discontinuity. The next seven bits indicate that the path number ranges from 1 to 127. The ninth bit indicates the regulation of engine speed: 1 means maximum engine speed, while 0 means the middle engine speed. Following two bits indicate the control of shuttle (2 bits): 01, 10 and 11 mean forward, backward and neutral, respectively. Other bits indicate the control of vehicle speed (4 bits), PTO (1 bit), and hitch (1 bit).

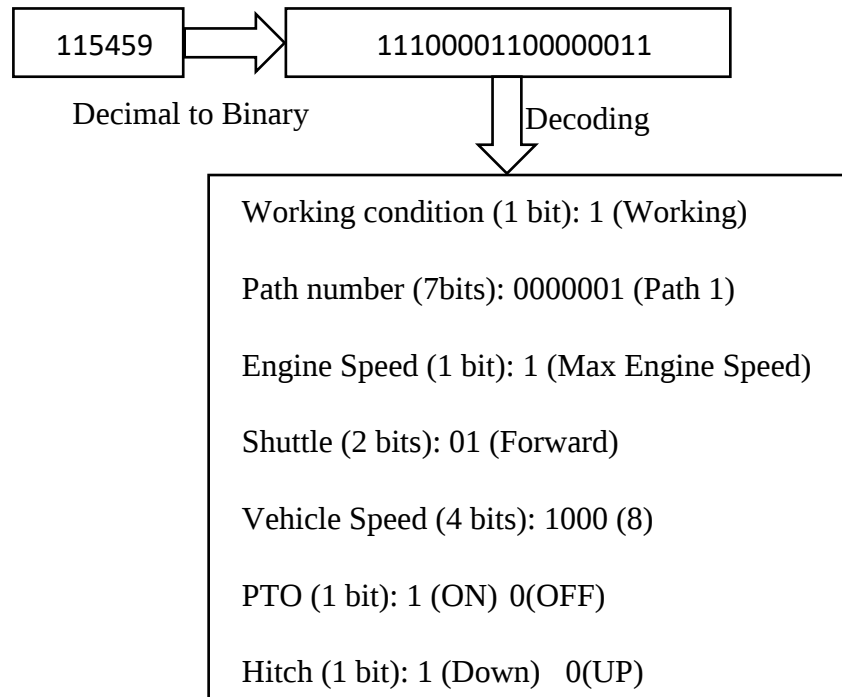


Fig. 3. 5. A code for the tillage operation.

3.2 Headland turning algorithms

Typical operations in the farming field consist of working and turning (transition) phases. Generally, the working operations are done on the mainland, which is typically the central area of the field, and turning operations are done in headland areas that surround the mainland. Each path/lane in the working area is usually called a swath or a track (Backman et al., 2015). A series of parallel swaths traversing the field is established by the experience of the operator and the shape of the field. The operations at the headland are constrained by the flexibility of vehicles and the headland distance for turning. Wheel-type tractors used in arable farming are seldom omnidirectional; for example, the heading of the robot cannot be changed directly along the way of working. It would be desirable to steer all-wheel in 180 deg. independently, but adding such features for current agricultural vehicles is not viable because of the complexity in manufacturing. As we can see, the turning process in the headland characterizes the most significant part of the non-working time that depends on the distance traveled while turning (i.e., the length of trajectories) and the average velocity.

Headland turning solutions should meet with constraints of swaths and the dynamic and kinematic properties of tractors. Constraints of swaths are defined as the swath-to-swath distance, which varies from the size of implement for farming, and the headland distance for turning. Dynamic properties of tractors include the maximum

acceleration/deceleration of the vehicle, maximum steering rate, which describes the time of the actuator system changes of the maximum right to maximum left steering lock. The steering rate is usually reflected through the steering lock-to-lock time (LTL, the time taken to steer from one side to the other side entirely), dependent on the inertia of the steering column, the servo system, and communication delays (Hoffmann et al., 2007). Kinematic properties of vehicles include the minimum turning radius, which is set as an empirical value or calculated from known steering angle and wheelbase, and the maximum lateral acceleration. Many researchers have presented methods to generate turning paths and control algorithms to follow the path accurately.

The objective of this chapter is to introduce various lane change methods that are commonly used nowadays and to analyze their advantage and disadvantage. Besides, a circle-back turning locus, based on continuous primitives, is presented. With regard to the path planning problems for a lane change, generic methods use the combination of primitive graphs, i.e., Dubins Curves (Dubins, 1957) and Reeds-Shepp Curves (Sabelhaus et al., 2013), and some continuous-curvature graphs, like clothoid, polynomial splines, cubic spirals or elastics to generate smooth optimal point-to-point trajectories. According to the moving direction during the lane change, turning maneuvers can be classified as forward turning and reverse turning. Forward turning means during the turning only forward motion is allowed; while the reverse turning means the vehicle can go backward while changing the working path.

3.2.1 Forward turning

1. U-turn

Fig. 3. 6 shows the construction of a typical U-turn, which is one of the most widely used lane-change algorithms. This algorithm is suitable for the wide lane-to-lane distance, which is two times larger than the minimum turning radius (Torisu et al., 1996). However, the distance between working paths is usually 2.64 m, which is much smaller than the turning radius, especially to the large size tractors in Hokkaido, Japan. Therefore, sequences of paths are reordered; that is, one or several paths are skipped from the current path to the next path. For example, as shown in Fig. 3. 7, there are six paths in the field, and the red numbers are the working sequence of the robot tractor, and we can see the order was 1→4→2→5→3→6, which the robot tractor skips two paths for each U-turn.

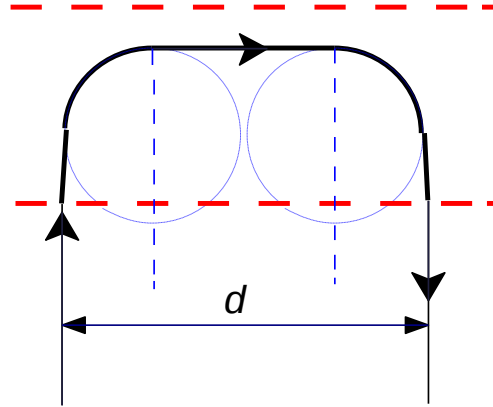


Fig. 3. 6. U-turn at the headland.

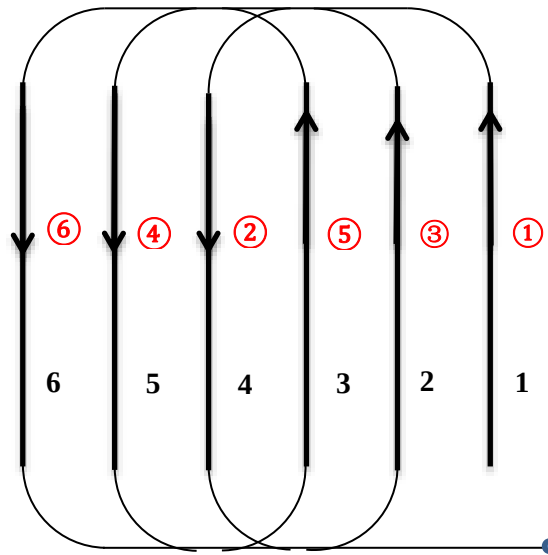


Fig. 3. 7. Working sequence of six paths.

With the change of the distance between lanes, the forward movement distance (d) should also be adjusted, like Gap-turn (Sabelhaus et al., 2013). Furthermore, before turning, the robot should go forward for a certain distance at the end of the path, so that there will be enough space for adjusting the position and heading before entering into the next path.

2. Keyhole turning/Omega-turn

Fig. 3. 8 shows two schemes of Keyhole turning (Takai et al., 2011). Since the robot still keeps moving forward during the turning, in some case, this is also named as forward turning (Kise et al., 2002a). This algorithm is also termed omega-turn or bulb-turn (Sabelhaus et al., 2013). It is suitable for the narrow path for working in each lane without skipping. In practice, the right/left steering angle can be different,

so that the shape of the keyhole is not symmetrical (Kise et al., 2002a). Slop-turn (Sabelhaus et al., 2013) is the extension of Keyhole turning; the trajectory self-intersects in Fig. 3. 9. This type of maneuver is not commonly used in practice, because of large headland occupation and long drivable trajectory. There are other methods, like minimal longitudinal width turning and pinhole turning, similar to keyhole turning in Dennis Sabelhaus (2013). The aim of changing the pattern of maneuvers is to simplify the computation, to diminish the headland occupation or both.

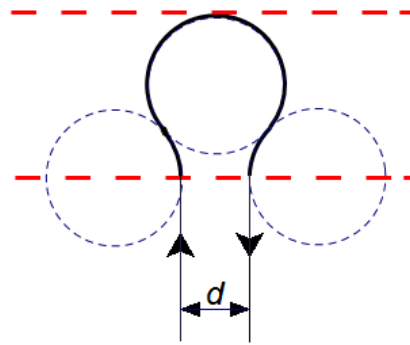


Fig. 3. 8. Keyhole turning at the headland.

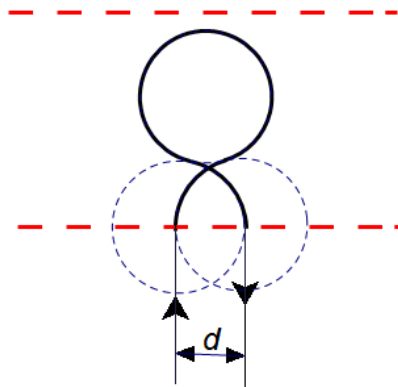


Fig. 3. 9. Slope-turn.

3.2.2 Reverse turning

1. Switch-back turning

Fig. 3. 10 shows the pattern of switch-back turning method, and because of the shape, it is also called as fishtail-turn. The turning starts at the end of the path with steering to the maximum in the right (in the case of Fig. 3. 10).

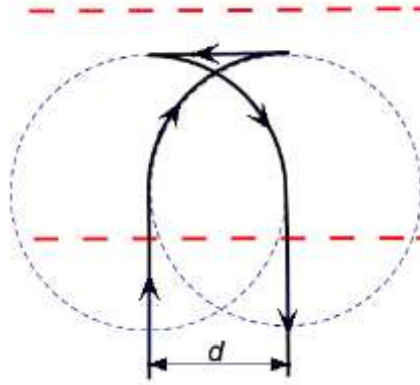


Fig. 3. 10. Switch-back turning method.

The robot stops straightly after turning 90 deg. and goes backward in a certain distance, which can be calculated by the width between paths (d) and the turning radius (Mi. Kise et al., 2001). The last, the robot moves towards the nearest point of the next path. Considering the slip according to different terrain conditions and the uncertain trajectory caused by steering rate, short straight movements are added before and after the switch-back turning method. That is to say, the robot will go forward for a certain distance and then steer to the maximum. In this way, the robot has more space in headland to adjust the heading or lateral error before approaching into the next path.

A modification of the original fishtail turning (Torisu et al., 1998) is shown in Fig. 3. 11. To approach into the next lane on the right side, the robot steers to the opposite direction (left side) at first and goes forward in the minimum turning radius. Then, it steers to the other direction and goes backward along an arch other than a straight path in Fig. 3. 10. Theoretically, this lane change strategy occupies minimal headland width compared with others. Also, before the start of turning, a short distance of the forward movement is necessary to decrease the lateral error after turning. However, because the forward movements of this pattern are relatively short, it is not easy to estimate the turning radius and to control the robot accurately.

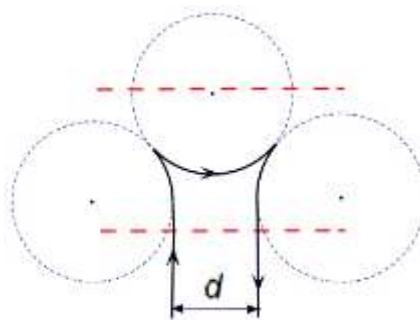


Fig. 3. 11. Fishtail-turn with minimal headland width.

2. H-turning method

The h-turning method in Fig. 3. 12 is widely in practice because of less headland occupation and high accuracy after the lane change. Similar to the U-turn in Fig. 3. 6, the robot tractor steers to the maximum at the end of the path without forwarding movement. What the difference is at the beginning of the next lane, the robot goes backward and then step into the next path to adjust the lateral error and heading error to a tolerable level.

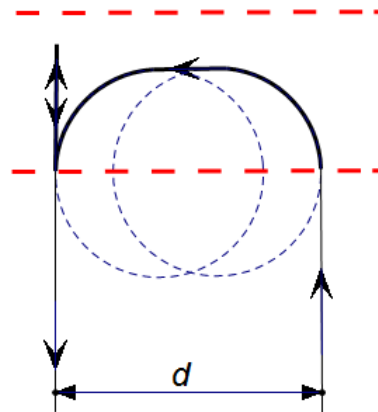


Fig. 3. 12. H-turning modified from U-turn

In addition, the switch-back turning can also be modified to h-turning, as shown in Fig. 3. 13. At the end of the current path, the tractor steers to the full left and starts the maneuver of switch-back turning, and goes backward at the start of next lane to adjust the position of the vehicle. Moreover, this method is also widely used when the headland is irregular in Fig. 3. 14. In this condition, it is less likely for a tractor to turn in a symmetric trajectory, and on the contrary, h-turning can fully make use of the space of irregular headland which is much useful in practice.

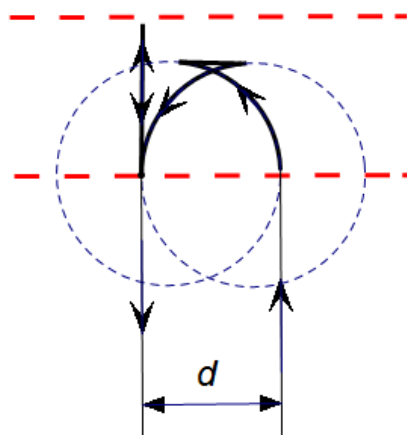


Fig. 3. 13. H-turning modified from switch-back turning.

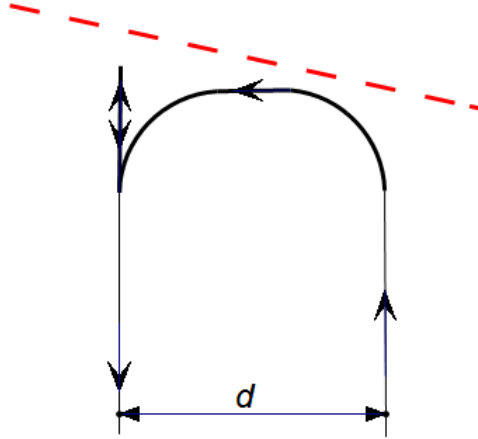


Fig. 3. 14. H-turning at the irregular shape of the headland.

3. Circle-back turning

The turning methods mentioned above can be easily embedded in autonomous navigation systems because maneuvers are all fully specialized. However, those methods are not adaptive for changes in various conditions such as the vehicle speed and the soil property. In practice, additional paths are added at the start or the end of turning procedures in order to smooth the planned path and to decrease the lateral deviation after turning. With these modifications, a robot tractor turns to the next path with a longer trajectory and occupies more headland space. These methods also require the control unit of the robot to be equipped with sufficient memory and computing capacity for designing and optimizing the path when in operation.

The circle-back (CB) turning plans the turning path by imitating the farmer's maneuvers (Wang and Noguchi, 2016). Fig. 3. 15 shows a typical trajectory created by the CB turning. In this example, the path space, the distance between the current path (Path_1) and the next path (Path_2), is d . This pattern of the trajectory is especially suitable for a narrow path space (less than two times the minimum turning radius) condition that needs complex lane-change maneuvers. At the end of the working path, the tractor turns left and moves along arch A-B. Then the tractor steers to the right at point B and goes backward until point C. Finally, the tractor goes forward and approaches the next path from point D. Unlike the algorithms mentioned before, the turning radii of forward movement (R_{left}) and backward movement (R_{right}) are of different values. Furthermore, all of the turning radii are measured during turning, rather than adopting empirical values. According to the measured turning radius, the backward navigation path is optimized in real time.

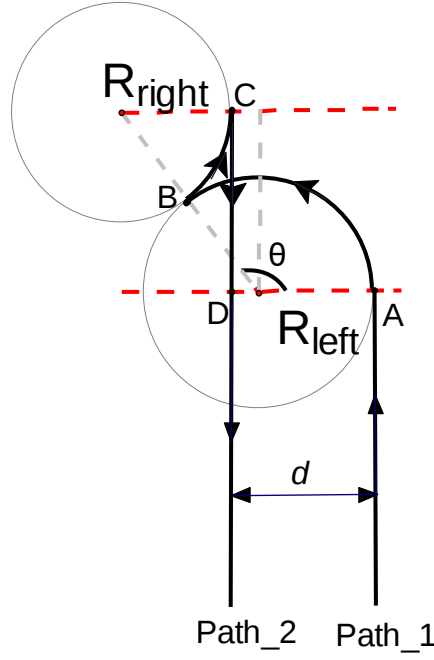


Fig. 3. 15. Scheme of the CB turning algorithm.

Considering vertical and horizontal disturbances applied to the soil at the headland, the robot tractor starts to turn at the end of the straight path without pause. The tractor starts to steer at the end of the current path and to go forward until the yaw angle has changed θ degree:

$$\theta = 90 + \sin^{-1}((R_b + d - R_f)/(R_b + R_f)) * 180 * \pi \quad (4)$$

where R_f and R_b are the turning radii of forward movement and backward movement, respectively. d is the path space. Considering the move during LTL time and the wheel slip, the control algorithm records the position of the vehicle during turning and calculates the turning radius by the least square method (LSM) in:

$$\arg \min(a, b, R) = \sum_{i=1}^N (x_i - a)^2 + (y_i - b)^2 - R^2 \quad (5)$$

where $(x_i, y_i), i \in [1, N]$ are the 2D position of the robot tractor recorded by the RTK-GPS, and a, b , and R are the center position of the fitted circle and the estimated turning radius, respectively.

The lane change maneuver with extensive driving operations plays a crucial role during the automation of agricultural robots. The turning trajectory is usually constrained by vehicle size, headland distance, and working content. Besides the basic methods mentioned before, other modifications use longer trajectory and

headland distance to create smooth paths for turning. The additional paths cause losses in efficiency and operator effort. Because of the accurate and specialized form, turning methods mentioned above are widely used, but they are not necessarily adaptive for dynamic environments, especially when it comes to the off-road vehicle robots. The intention of path planning is to generate a feasible trajectory for an agricultural robot to follow. Because of the steering rate and the delay in response to the command, robots cannot follow trajectories constructed only by Dubins Curves. Therefore, it would be helpful if the control algorithm can estimate the trajectories of the robot during turning and generate the paths online. In this way, we can know how much the steering rate and time delay affect the system and make compensation in response. Furthermore, essential constraints, including the minimum turning radius, are not constant values that need to be estimated or be calculated dynamically.

4. Other methods for side shift

The side shift maneuver (Fig. 3. 16) provides a method for transfer between adjacent paths when the robot locates at a narrow place that prevents the use of U-turning and CB turning (De Carvalho et al., 1997). The robot tractor goes forward at the end of the Path 1 (Step 1) and then go backward into the next path until the start of Path 2 (Step 2 and Step 3). The robot starts working from the beginning of Path 2 in Step 4. This turning method can be used in narrow headland space (Fig. 3. 4) and irregular shape of the field with short straight paths at the main field area. The drawback of this method is that the robot passes the Path 2 two times. It will increase soil compaction and waste fuel. After the work at the main body of the field, the robot should transfer to the navigation path at headland following the maneuvers in Fig. 3. 17. The vehicle turns until the heading of the vehicle parallels to the next path (Path 2 in this case), goes backward to track the Path 2 and starts the work from the beginning of the Path 2. The turning methods mentioned above can be easily embedded in autonomous navigation systems because the turning operations are simplified to follow several Dubins paths. The cost function of each turning maneuver including T_{turn} time spent in one turn [s] and D_{turn} non-working distance [m].

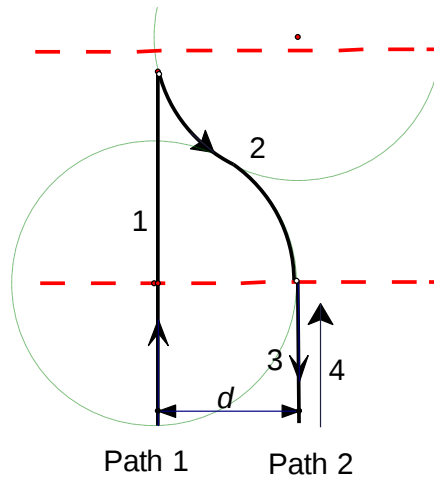


Fig. 3. 16. Template of side shift turning.

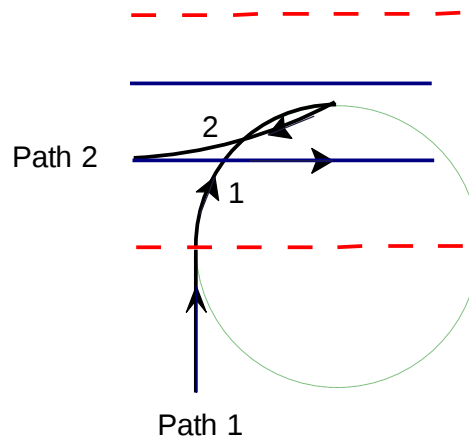


Fig. 3. 17. Template of backward tracker.

Chapter 4 Vehicle model and state estimation

In order to increase the accuracy of the vehicle state estimation and the stability of movement control, we need to understand what determines the position where the robot moves to and how to plan the vehicle to a reference position. To control an autonomous vehicle to perform applications such as path following or obstacle avoidance requires standard robotics knowledge of position, kinematics, and dynamics. The scheme of the path following control is shown in Fig. 4. 1. For a robot tractor, the control input consists of the front-wheel steer, which controls the orientation of the vehicle, and the accelerator/brake for speed control. The output of the system is the response of the vehicle, which includes the position, orientation, acceleration, speed, and so on. The navigator is to calculate the deviation of vehicle's state from the reference path (also termed error of the state) and to control the tractor to track the path. The controller calculates an optimal input to minimize errors of the vehicle state in a certain predictive period. The simulator in this algorithm has two functions: one is to improve the accuracy of the measurement. The other is to predict the vehicle state in the next step. A simulator estimates the predicted vehicle states according to the time-series control inputs and measured states. For a precise estimation, vehicles are mostly assumed to comply with specific motion models which describe their kinematic/dynamic behaviors (Schubert et al., 2008). Combining the estimation of vehicle states by motion models and the measurements by sensors is meaningful for motion prediction and risk assessment for robot tractors.

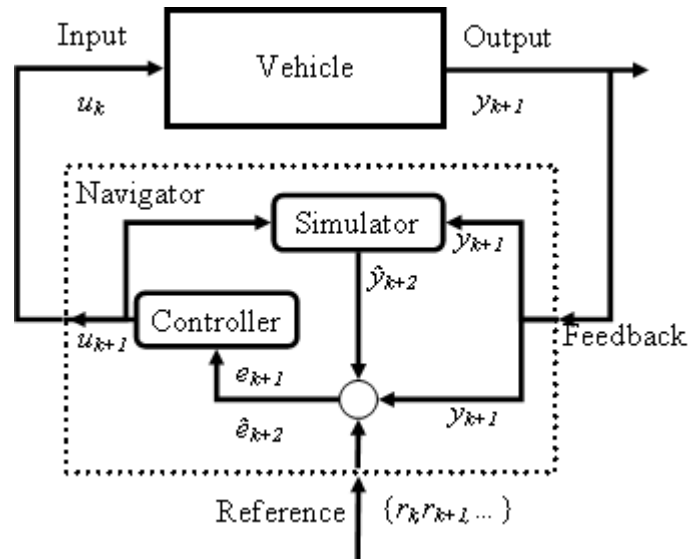


Fig. 4. 1. Block diagram of navigation procedure for path following control. The subscript k indicates the timestamp of the parameter.

4.1 Vehicle modeling

Numerous motion models have been proposed for certain applications. A simple systematization can be achieved by defining a vehicle's motion using kinematics or dynamics. Kinematic models are widely used to develop a motion control strategy for vehicles because they are much simpler than dynamic models. These models are valid if the robot has a constant speed, mechanical structure, and mass. Dynamic models are useful for dynamic variations. These variations include changes in the inertia and center of gravity which caused by the variable carrying load or the replacement of attached equipment. The changes in the terrain topography, texture, or in wheel properties due to wear, contamination, or deformation play a significant role in the vehicle motion (Albagul and Wahyudi, 2004). These variations can easily affect the traction properties and hence, the robot movement and may cause slippage to occur. Therefore, the robot must be able to adapt to the control strategy to cope with variable conditions. In this research, the vehicle's planar motion is modeled online using the previous measurements. More vehicle dynamic models in three-dimensional movements not mentioned here (Hassaan and Mohammed, 2015; Lapapong et al., 2011; Richier et al., 2013). Not only because these models are complex for calculation, but also the variations of internal parameters needed by these dynamic models are not natural to be observed by exteroceptive sensors.

4.1.1 The unicycle model of vehicle kinematics

If the RTK-GPS is available, simple models can be used for positioning and vehicle control. A preferred model is that of a unicycle model, which can give a global description of the vehicle. A unicycle robot has two rear wheels and one orientable front wheel. The posture definition of a unicycle robot is shown in Fig. 4. 2. The robot with one front wheel can move with the desired velocity (v) at a specified heading (φ). The position of the robot is described by the state vector $\xi = (x \ y \ \varphi)$ with respect to an arbitrary inertial frame. In this research, x and y are the coordinates of the COG of the vehicle, while φ describes the orientation of the vehicle to the inertial frame. Assuming the angular acceleration is 0, let ω be the angular velocity of the unicycle orientation in the x - y plane. In the global coordinate frame, the vehicle is represented as a point of mass:

$$\dot{x} = \begin{cases} v/\omega[-\cos(\varphi + \omega\Delta t) + \cos \varphi], & \omega \neq 0 \\ v \sin \varphi, & \omega = 0 \end{cases} \quad (6)$$

$$\dot{y} = \begin{cases} v/\omega [\sin(\varphi + \omega \Delta t) - \sin \varphi], & \omega \neq 0 \\ v \cos \varphi, & \omega = 0 \end{cases}$$

$$\dot{\varphi} = \omega \Delta t$$

At the condition of steering angle (δ) and the wheelbase of the vehicle (l), the angular velocity of the unicycle model can be calculated by:

$$\omega = \frac{v}{l} \tan \delta \quad (7)$$

The state transformation from time k to the next time $k + 1$ can be described by a non-linear function as follows:

$$X_{k+1} = \begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \varphi_{k+1} \end{bmatrix} = f(X_k, \hat{\varepsilon}_k) = \hat{A} \begin{bmatrix} x_k \\ y_k \\ \varphi_k \end{bmatrix} + \hat{\varepsilon}_k \quad (8)$$

$$\hat{\varepsilon}_k \sim N(0, \sigma_a^2)$$

$\hat{\varepsilon}_k$ is the system noise modeled by a zero-mean Gaussian distribution.

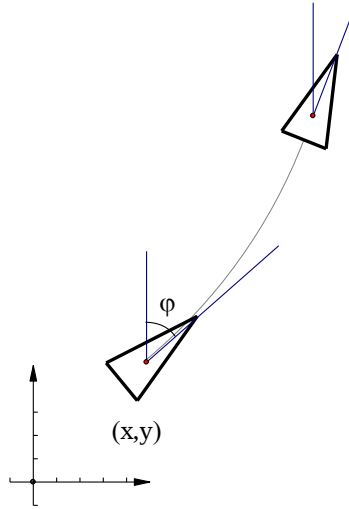


Fig. 4. 2. The unicycle motion model (Lavalle, 2006).

In the condition of a small steering angle and high frequency of measurement (relative to the low velocity), the vehicle's orientation is assumed to remain constant without (yaw) rotations for the small time interval. The instant movement from k to $k+1$ can be assumed to be linear (M. Kise et al., 2001). Under these assumptions, the following equations can calculate the vehicle's state:

$$\begin{aligned}
 x_{k+1} &= x_k + \int_k^{k+1} v \sin(\varphi) dt \\
 &\approx x_k + \frac{v}{2} (\sin \varphi_k + \sin \varphi_{k+1}) \Delta t \\
 y_{k+1} &= y_k + \int_k^{k+1} v \cos(\varphi) dt \\
 &\approx y_k + \frac{v}{2} (\cos \varphi_k + \cos \varphi_{k+1}) \Delta t \\
 \varphi &= \varphi_k + \int_k^{k+1} \omega dt \approx \varphi_k + \Delta t \omega
 \end{aligned} \tag{9}$$

or even simplified as (Nascimento et al., 2018; van Essen and Nijmeijer, 2001):

$$\begin{aligned}
 x_{k+1} &= x_k + v \Delta t \sin(\varphi_k) \\
 y_{k+1} &= y_k + v \Delta t \cos(\varphi_k) \\
 \varphi_{k+1} &= \varphi_k + \Delta t \omega
 \end{aligned} \tag{10}$$

The linearity of the state transition equation allows propagation of the state probability distribution.

$$\begin{aligned}
 x &= x_0 + v \int \sin(\varphi) dt \\
 y &= y_0 + v \int \cos(\varphi) dt \\
 \varphi &= \varphi_0 + \int \omega dt
 \end{aligned} \tag{11}$$

The above equation is a simple dead reckoning process. Besides, the velocity can be assumed to change linearly in the longitudinal control of a multi-robot system. It leads to a constant acceleration (a) vehicle model. The state of the vehicle can be calculated from:

$$\begin{aligned}
 x &= x_0 + \int v \sin(\varphi) dt \\
 y &= y_0 + \int v \cos(\varphi) dt \\
 v &= v_0 + \int a dt
 \end{aligned} \tag{12}$$

$$\varphi = \varphi_0 + \int \omega dt$$

The unicycle model is widely used in the navigation of indoor robots and automobiles. It is also used in the navigation of low-speed agricultural vehicles. However, it does not allow for any slipping or skidding that may occur during off-road driving. The unicycle model is inaccurate, but the tire-ground interactions in slippery farmland are challenging to model. In addition, it performs poorly during GPS outages, as the position of the vehicle relies strongly on the accuracy of the odometer and IMU's gyro and accelerometers. Since the model assumes that there is no correlation between the velocity and the yaw rate, the vehicle heading measurement is sensitive to the yaw rate measurement even if the vehicle keeps instill (Schubert et al., 2008). Dead reckoning is also subject to cumulative errors. Consequently, a more restrictive motion model and state estimation method are necessary for accuracy control.

4.1.2 The bicycle model of lateral vehicle kinematics

A kinematic model based on the Ackerman steering geometry provides a mathematical description of the planar motion without considering the forces that affect the motion (Rajesh Rajamani, 2006). In the bicycle model of the vehicle, as shown in Fig. 4. 3, two front wheels and rear wheels are represented by one single wheel at the front and rear center, respectively.

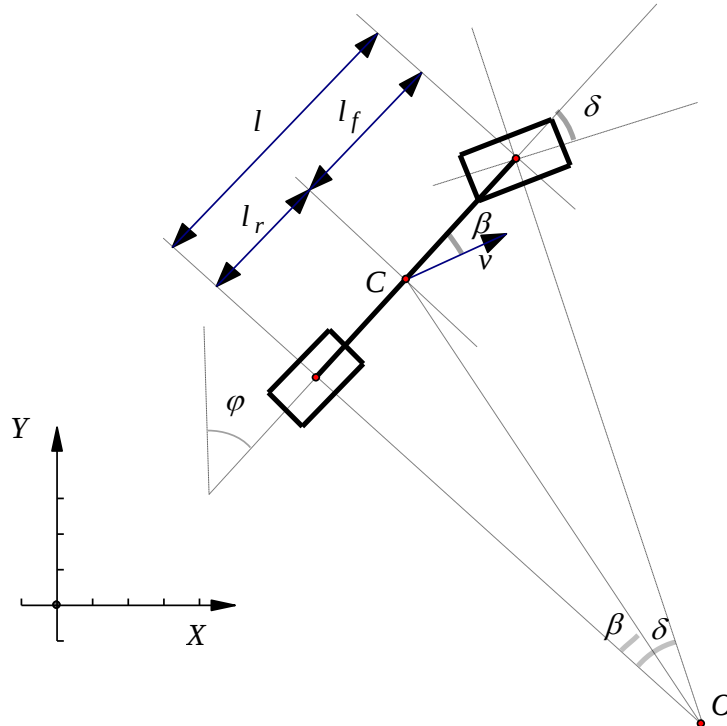


Fig. 4. 3. Lateral vehicle motion.

The model is derived assuming the rear wheel cannot be steered, and the steering angle for the front wheel is represented by δ . In addition, it also assumes that the vehicle has negligible inertia at low-speed motion (less than 5 m/s) of the vehicle. At low speeds, it is reasonable to assume that the velocity at each wheel is in the direction of the wheel when the lateral force generated by the tires is small (Rajesh Rajamani, 2006). Therefore, the vehicle is assumed to move along a circular trajectory with a yaw rate ω around the COG at point C. And the total lateral force generated by tires is:

$$mv^2/R \quad (13)$$

R is the radius of the vehicle's circular path, which is defined by the length of the line OC . The velocity at the COG of the vehicle is denoted by v . The direction of the velocity with respect to the longitudinal axis of the vehicle is called the slip angle of the vehicle β . The distance of front and rear wheels from the COG of the vehicle are l_f and l_r , respectively. The wheelbase of the vehicle is

$$l = l_f + l_r \quad (14)$$

The state of the vehicle includes the position at COG of the vehicle (x, y) and the orientation/heading of the vehicle φ . The motion is given by

$$\begin{aligned} \dot{x} &= v \sin(\varphi + \beta) \\ \dot{y} &= v \cos(\varphi + \beta) \end{aligned} \quad (15)$$

$$\dot{\varphi} = \omega = \frac{v \cos \beta}{l} \tan \delta$$

$$\begin{aligned} x_{k+1} &= x_k + v \Delta t \sin(\varphi_k + \beta) \\ y_{k+1} &= y_k + v \Delta t \cos(\varphi_k + \beta) \end{aligned} \quad (16)$$

$$\varphi_{k+1} = \varphi_k + \Delta t \omega$$

The slip angle can be obtained by

$$\beta = \tan^{-1} \frac{l_r \tan \delta}{l} \quad (17)$$

The vehicle sideslip β can be calculated as the difference of vehicle direction given by the GPS (φ_{GPS}) and the direction given by the IMU (φ_{IMU}), as shown in Eq. (18) (Bevly et al., 2002).

$$\beta = \varphi_{GPS} - \varphi_{IMU} \quad (18)$$

4.1.3 The bicycle model of lateral vehicle dynamics

At higher vehicle speeds and higher vehicle weight, the assumption of the kinematic model can no longer be made. Instead, the dynamic model involves the relation between forces and torques and inertia effects of the vehicle: tire slip and steering servo actuation. The bicycle model of the vehicle, as shown in Fig. 4. 4 is considered. The velocity at the longitudinal axis direction is called the longitudinal velocity v_x . The velocity perpendicular to the longitudinal axis is called the lateral velocity v_y . Therefore, the slip angle of the vehicle can be derived as

$$\tan \beta = v_y / v_x \approx \beta, \text{ for small slip angle.} \quad (19)$$

F_{yf} and F_{yr} are the lateral tire forces of the front and rear wheels respectively.

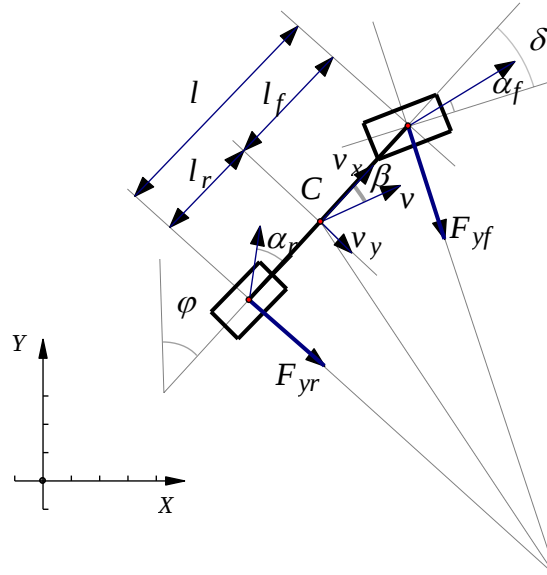


Fig. 4. 4. Lateral vehicle dynamics.

Experimental results show that for small slip angle, lateral tire forces follow these equations:

$$\begin{aligned} F_{yf} &= 2k_f \alpha_f \\ F_{yr} &= 2k_r \alpha_r \end{aligned} \quad (20)$$

where k_f and k_r are called the cornering stiffness of each tire. α_f and α_r are slip angles of each front and rear tires respectively. The slip angle of a tire is defined as the difference between a wheel's orientation and its orientation of the velocity (Rajesh Rajamani, 2006). Regardless of the road bank angle and applying Newton's second law for motion of the vehicle is obtained as

$$\begin{aligned} \sum F_y &= F_{yf} \cos \delta + F_{yr} \\ &= m(v_x \omega + \dot{v}_y) \end{aligned} \quad (21)$$

$$\sum M_Z = I\dot{\omega} = l_f F_{yf} \cos \delta - l_r F_{yr}$$

where m is the mass of the vehicle body, $\dot{v}_y$ is the acceleration of the vehicle in the lateral direction. I is the vehicle moment of the inertia around the z-axis.

$$\begin{aligned} \tan(\delta - \alpha_f) &= \frac{v_y + l_f \omega}{v_x} \\ \tan(\alpha_r) &= \frac{-v_y + l_r \omega}{v_x} \end{aligned} \quad (22)$$

The bicycle is two degrees of freedoms, which are represented by the vehicle position and the yaw rate. Substituting from Eqs. (20), and (22) into Eq. (21), the state-space model can be written as (Rajesh Rajamani, 2006)

$$\begin{aligned} \dot{X} &= \begin{bmatrix} \dot{v}_y \\ \dot{\omega} \end{bmatrix} = A \begin{bmatrix} v_y \\ \omega \end{bmatrix} + B\delta \\ A &= \begin{bmatrix} -2(k_f + k_r)/mv_x & -2(k_f l_f - k_r l_r) - mv_x^2/mv_x^2 \\ -2(k_f l_f - k_r l_r)/I & -2(k_f l_f^2 + k_r l_r^2)/Iv_x \end{bmatrix} \\ B &= \begin{bmatrix} 2k_f/mv_x \\ 2k_f l_f/I \end{bmatrix} \end{aligned} \quad (23)$$

The slip angle is used as one of the state parameters instead of the lateral velocity. Therefore, substituting from Eq. (19) into Eq. (23), the state-space model can also be written as:

$$\dot{X} = \begin{bmatrix} \dot{\beta} \\ \dot{\omega} \end{bmatrix} = A \begin{bmatrix} \beta \\ \omega \end{bmatrix} + B\delta$$

$$A = \begin{bmatrix} -2(k_f + k_r)/mv_x & -2(k_f l_f - k_r l_r) - mv_x^2/mv_x^2 \\ -2(k_f l_f - k_r l_r)/I & -2(k_f l_f^2 + k_r l_r^2)/Iv_x \end{bmatrix} \quad (24)$$

$$B = \begin{bmatrix} 2k_f/mv_x \\ 2k_f l_f/I \end{bmatrix}$$

Both of the Eqs. (23) and (24) are vehicle state equations of dynamic models. They are more complicated than the kinematic model but more accurate for lateral control.

Herein, it is necessary to remark on the bicycle model assumptions. First of all, the left and right steering angles are assumed to be equal in the bicycle model. Because the radius of each trajectory of these wheels travels is different, the steering angle in left and right might be different. Secondly, the lateral tire force is assumed to be proportional to the slip angle. However, the tires-terrain interactions in the ground vehicles are involved. The assumption will not be valid at large slip angles. In the case of large slip angles, the lateral tire force depends on the tire load, slip angle, tire-road friction coefficient, and so on (Rajesh Rajamani, 2006).

4.1.4 System identification

Generally, system identification means to identify the vehicle model parameters listed in Table 4. 1. Parameters like vehicle velocity, orientation, yaw rate, and steering angle can be measured using GPS, IMU, or both. Some parameters related to the position of COG are challenging to measure. The one reason is that the vehicle is too big and too heavy to be measured for a farmer. The other reason is that the vehicle mass and COG will change while loading or unloading. Therefore, the online system identification method will be helpful if it can calculate the parameters based on the observation of the system. The other way of system identification is to describe the vehicle motion using the combination of linear or nonlinear equations. For example, the vehicle model can be identified using neural networks, that is, a combination of nonlinear equations. The neural network simulates vehicle's states based on inputs and continuous states frames. It was proved that the well-trained neuro-controller indicated higher accuracy than the PID controller (Ishii et al., 1994). Many leading methods in machine learning are limited their usage in the online identification of a model. For example, the neural networks do not easily include known constraints and

may be difficult to interpret. Besides, online measurement data are limited and noisy. On the contrary, many techniques, such as sparse identification of nonlinear dynamics modeling procedure and dynamic mode decomposition obtain rapid learning ability based on limited data (Brunton and Kutz, 2019; Kaiser et al., n.d.; Proctor et al., 2016). The vehicle state can be predicted based on the regression models of input and vehicle states.

Table 4. 1. Parameters for identification.

Variable	Description	Units
l_f	the distance of front wheel from the COG of the vehicle	m
l_r	the distance of rear wheel from the COG of the vehicle	m
l	the wheelbase of the vehicle	m
m	vehicle mass	kg
I	yaw moment of inertia	$kg*m^2$
k_f	frontal tire cornering stiffness	$N/deg.$
k_r	rear tire cornering stiffness	$N/deg.$
δ	front wheel steering angle	$deg.$
φ	vehicle orientation	$deg.$
v	vehicle velocity	m/s
ω	yaw rate	$deg./s$
v_x	longitudinal velocity	m/s
β	slip angle	$deg.$

A. Kinematic model identification

One basic idea is to assume that the vehicle state (X) is determined by the input of control parameters (U) following a nonlinear transform like $X = f(U)$. This transform is similar to the vehicle kinematics model. For the vehicle kinematics model in Eqs. (15) and (17), the wheelbase of the vehicle l can be measured manually or be taken from the vehicle's technical specifications. The slip angle can be measured by the Eq. (18). Applying the median filter to the calculated slip angle, the raw data and the filtered slip angle are shown in Fig. 4. 5. The vehicle is a four-wheel-type robot tractor, EG83. The wheelbase is 2.2 m. The steering angle is shown in Fig. 4. 6. According to the Eq. (17), a linear regression model can be calculated based on the steering angle and the slip angle. The slope of the regression line in Fig. 4. 7 is l_r/l . Then the value of l_r can be estimated based on this experiment. Based on the estimated l_r , measured l , and the vehicle speed (in Fig. 4. 8), the vehicle state can be deducted by Eqs. (15) and (17).

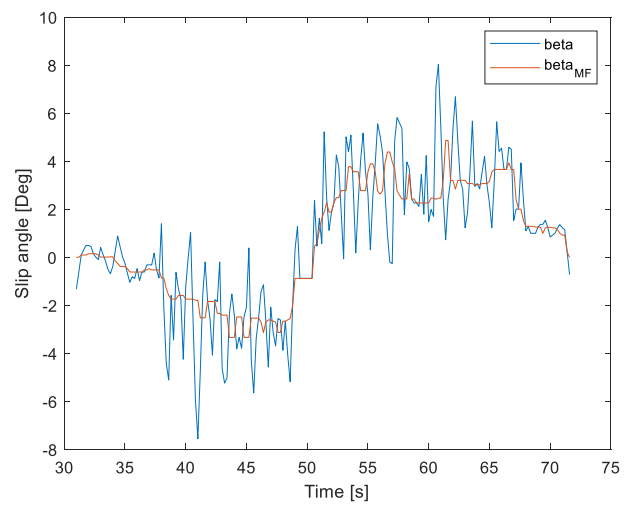


Fig. 4. 5. Time series of slip angles.

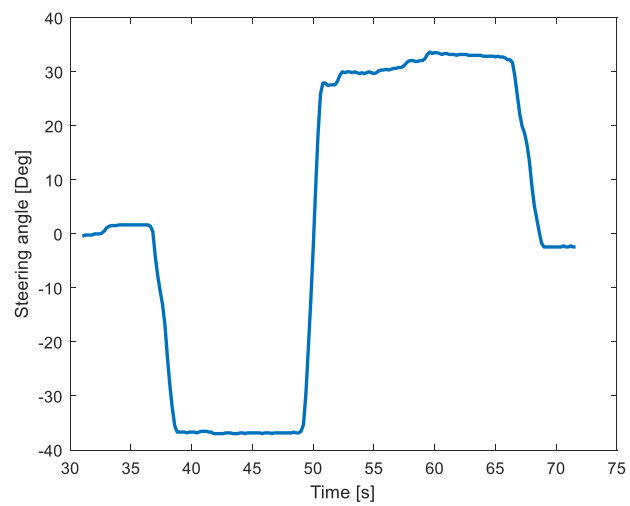


Fig. 4. 6. Time series of steering angle.

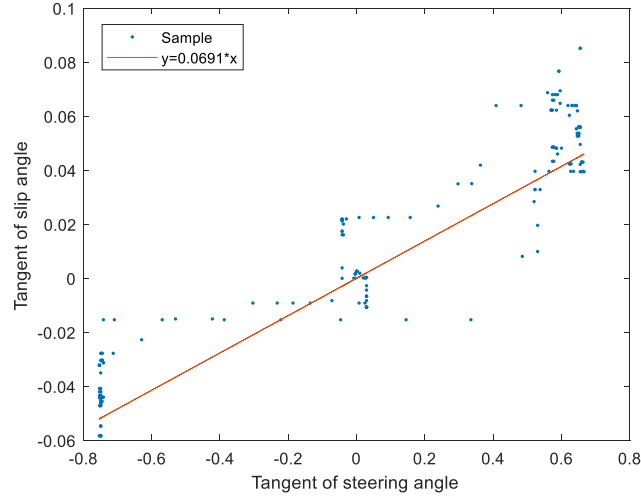


Fig. 4. 7. A linear regression model for system identification. The slope of the regression line equals to l_r/l .

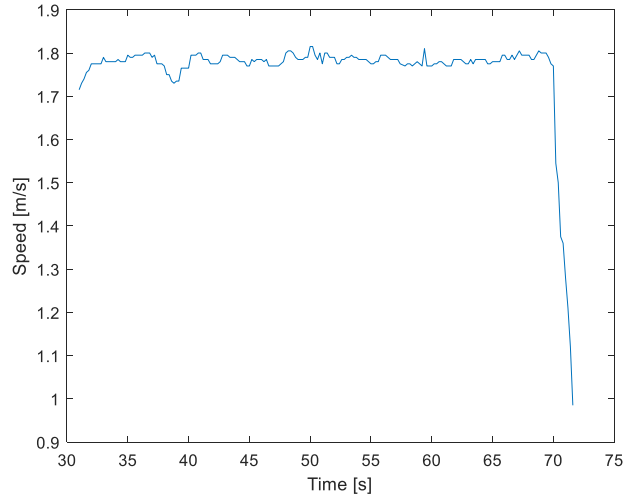


Fig. 4. 8. Time series of vehicle speed.

The comparisons of measured slip angle and yaw rate and the estimated by the kinematic model are shown in Fig. 4. 9 and Fig. 4. 10, respectively. The vehicle states of motion like the vehicle heading and vehicle position derived from the Eq. (16) are shown in Fig. 4. 11 and Fig. 4. 12, respectively. From the change of vehicle heading, it can be seen that the kinematic model overreacts to the change of steering angle. The estimated trajectory can follow the ground truth at the small steering angles, not for the large steering angles. There are several reasons for the uncertainty in estimating vehicle state based on steering angles. The slippage and tire deformation cause under steering or over steering that varies the turning radius. Therefore, the vehicle model

for large steering angle condition is necessary for accurate control. In addition, sensor fusion methods that can combine the information of vehicle model and measurement by IMU and GPS are also helpful to increase the accuracy. The state estimation and sensor fusion methods will be discussed in the next section.

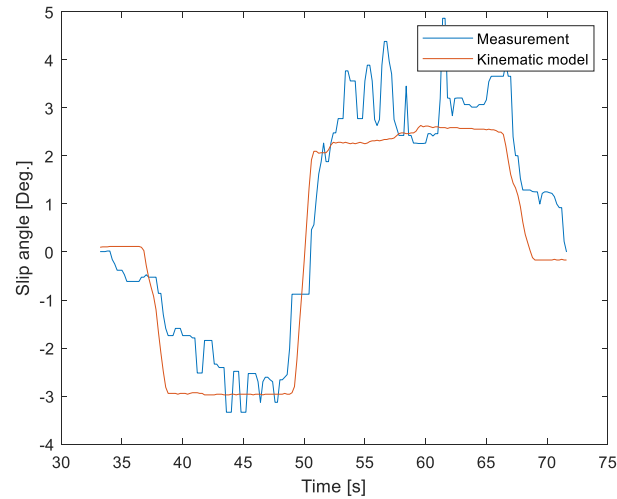


Fig. 4. 9. Comparison of measured slip angle and the estimated by the kinematic model.

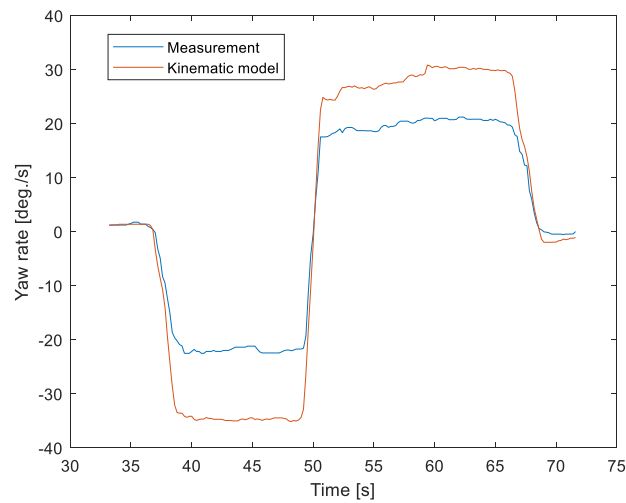


Fig. 4. 10. Comparison of measured yaw rate and the estimated by the kinematic model. Estimated parameters are not suitable for large steering angle conditions where the yaw rate is significant.

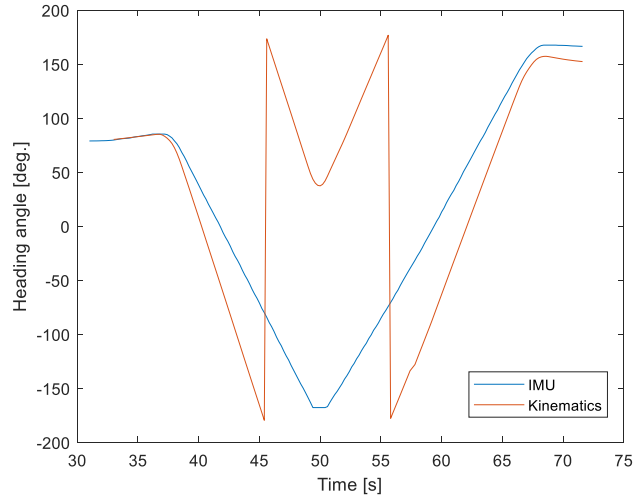


Fig. 4. 11. Comparison of IMU measured vehicle heading and the one deduced by the kinematic model.

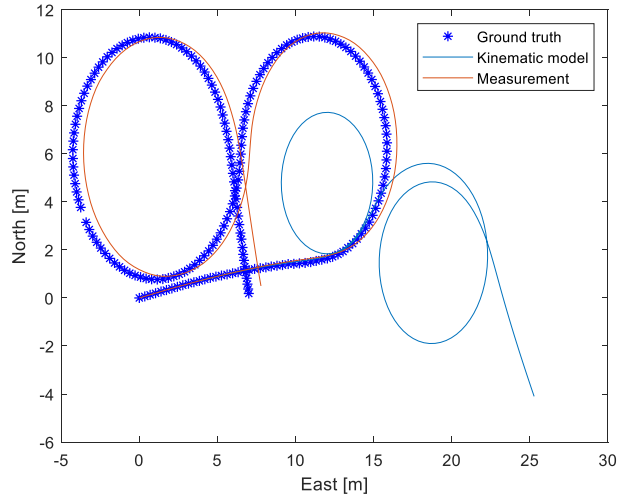


Fig. 4. 12. Comparison of ground truth and estimated vehicle trajectories from Eq. (16). Ground truth is the vehicle position measured by RTK-GPS.

B. System identification using dynamic mode decomposition with control

For a linear dynamic system, the state at time $k+1$ depends only on the state and actions from the previous state at time k given by Eq. (25).

$$X_{k+1} = \begin{bmatrix} \beta_{k+1} \\ \omega_{k+1} \end{bmatrix} = A \begin{bmatrix} \beta_k \\ \omega_k \end{bmatrix} + B u_k = [A \ B] \begin{bmatrix} X_k \\ u_k \end{bmatrix} \quad (25)$$

Fig. 4. 13 illustrates the outline of a system identification mechanism. The data matrixes include the snapshots of data in time domain include the control sequence U , and the vehicle state measurements X and X' in the following descriptions:

$$\begin{aligned}
 X' &= [A \ B] \begin{bmatrix} X \\ U \end{bmatrix} = [A \ B] \Omega \\
 X &= \begin{bmatrix} \vdots & \vdots & \vdots \\ x_1 & \cdots & x_{k-1} \\ \vdots & \vdots & \vdots \end{bmatrix} \\
 X' &= \begin{bmatrix} \vdots & \vdots & \vdots \\ x_2 & \cdots & x_k \\ \vdots & \vdots & \vdots \end{bmatrix} \\
 U &= \begin{bmatrix} \vdots & \vdots & \vdots \\ u_1 & \cdots & u_{k-1} \\ \vdots & \vdots & \vdots \end{bmatrix}
 \end{aligned} \tag{26}$$

where Ω contains both the state and control snapshots. The singular value decomposition (SVD) is a computationally efficient and accurate method for finding the pseudo inverse and eigenvalues of a matrix (Proctor et al., 2016). An SVD is performed on the matrixes Ω , and we can find approximations of the dynamic matrixes A and B using the following computations:

$$\begin{aligned}
 [A \ B] &= X' \Omega^\dagger \\
 &\approx X' V \Sigma^{-1} U \\
 &= [X' V \Sigma^{-1} U_1 \quad X' V \Sigma^{-1} U_2]
 \end{aligned} \tag{27}$$

where $\Omega^\dagger$ is the pseudoinverse of matrix Ω . U_1 is a 2×2 matrix from the beginning two rows of matrix U , and U_2 is the third row of matrix U . By identifying the matrix A and B , a regression model is used to estimate the vehicle state instead of the dynamic model.

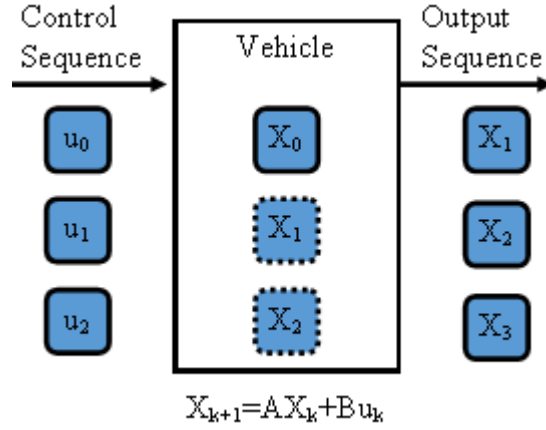


Fig. 4. 13. The illustration of system identification with data collections in time. The aim is to find the dynamic properties of matrixes A and B.

The followings in this section describe a series of experiments for identifying the vehicle dynamics of a four-wheel-type robot tractor (EG83) with a rotary. The experiment was conducted at a concrete road at Hokkaido University. The time interval of each data snapshot was 0.2 s.

A. Experiment 1 – Snapshot matrix size is 80 (16 s)

The data within 16 s are used for identifying a regression vehicle model in the form of a dynamic model:

$$A = \begin{bmatrix} 0.8441 & -0.0322 \\ -0.4370 & 0.2240 \end{bmatrix} \quad (28)$$

$$B = \begin{bmatrix} 0.0299 \\ 0.4942 \end{bmatrix}$$

The slip angles and yaw rates during the experiment are shown in Fig. 4. 14 and Fig. 4. 15, respectively. We can see that the regression model can estimate the yaw rate accurately, but the error of the slip angle becomes significant at the prediction period. Compared with the measurement, the calculated vehicle heading angle and the position are shown in Fig. 4. 16 and Fig. 4. 17, respectively.

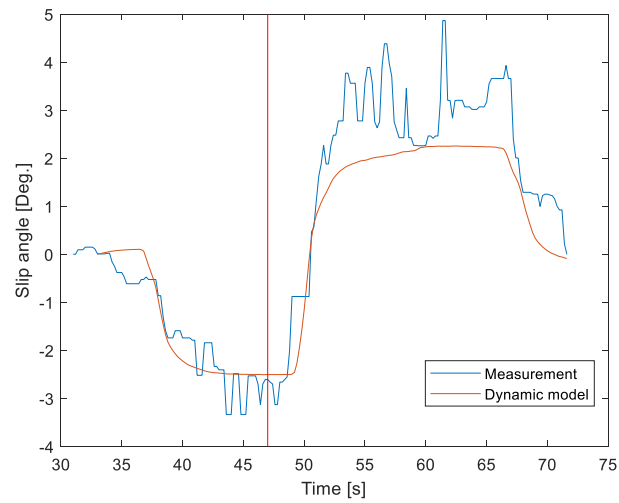


Fig. 4. 14. Time-series slip angles. The data before the red vertical line are for training (system identification).

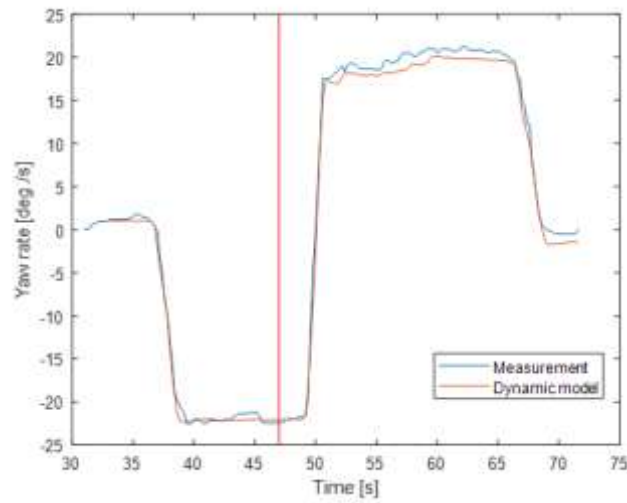


Fig. 4. 15. Time-series yaw rates. The data before the red vertical line are for training (system identification).

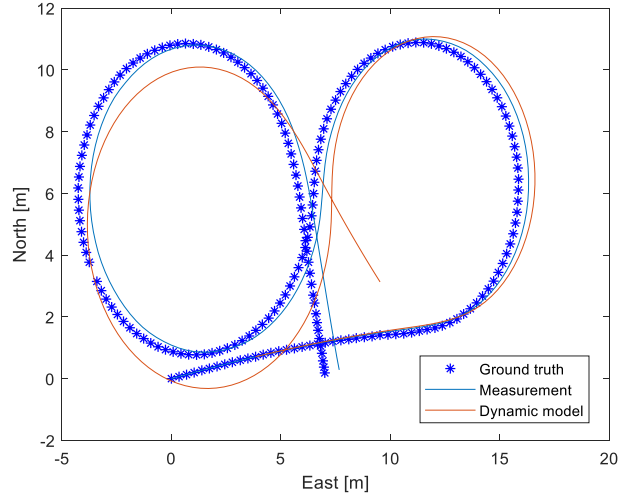


Fig. 4. 16. Comparison of ground truth and estimated vehicle trajectories from Eq. (16). Ground truth is the vehicle position measured by RTK-GPS. The vehicle positions are calculated by the measured yaw angle, slip angle and vehicle speed for the measurement.

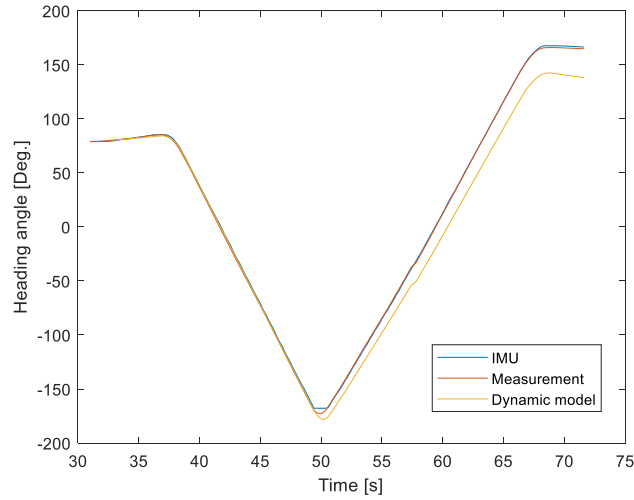


Fig. 4. 17. Comparisons of vehicle heading. The “measurement” vehicle heading uses the measured yaw angle and the slip angle to deduct the heading angle.

B. Experiment 2 – Snapshot matrix size is 180 (16 s)

In the second experiment, the data at the previous 36 s are used for identifying the vehicle dynamics in the following results:

$$A = \begin{bmatrix} 0.8542 & 0.0250 \\ -0.1701 & 0.9010 \end{bmatrix} \quad (29)$$

$$B = \begin{bmatrix} -0.0033 \\ 0.0735 \end{bmatrix}$$

The slip angles and yaw rates during the experiment are shown in Fig. 4. 18 and Fig. 4. 19, respectively. We can see that the regression model can estimate the yaw rate accurately, but the error of the slip angle becomes significant at the prediction period. Compared with the measurement, the calculated vehicle heading angle and the position are shown in Fig. 4. 20 and Fig. 4. 21, respectively. The estimation accuracy is worse than Experiment 1. There are several possible reasons. For example, the long term data is noisy; the vehicle dynamics changes along with tire-road interact; steering to the right side and steering to the left side are different dynamics because of unsymmetric hydraulic power.

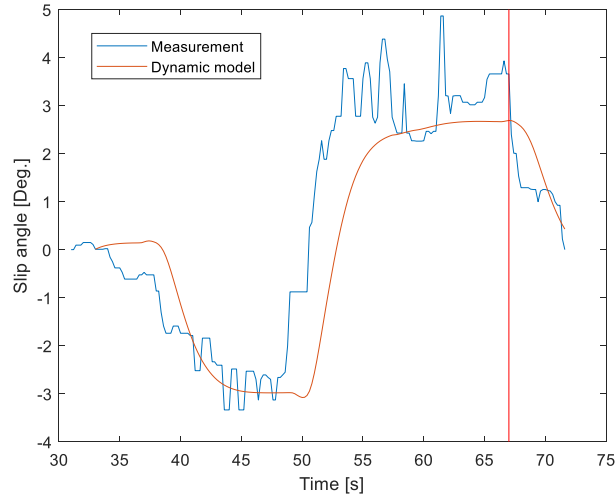


Fig. 4. 18. Time series of slip angles. The data before the red vertical line are for training (system identification).

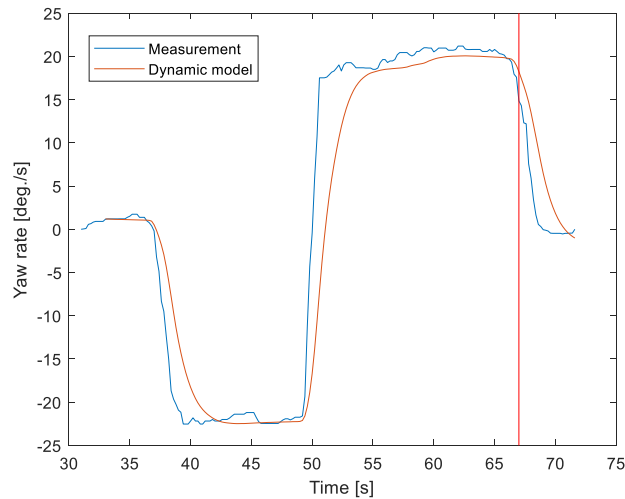


Fig. 4. 19. Time series of yaw rates. The data before the red vertical line are for training (system identification).

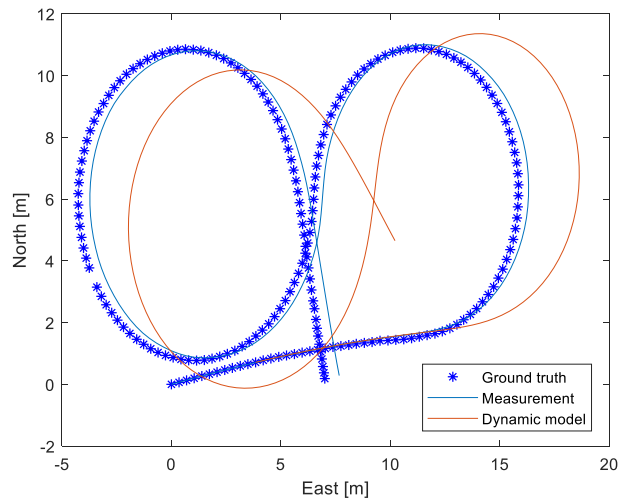


Fig. 4. 20. Comparison of ground truth and estimated vehicle trajectories from Eq. (16). Ground truth is the vehicle position measured by RTK-GPS. The vehicle positions are calculated by the measured yaw angle, slip angle, and vehicle speed for the measurement.

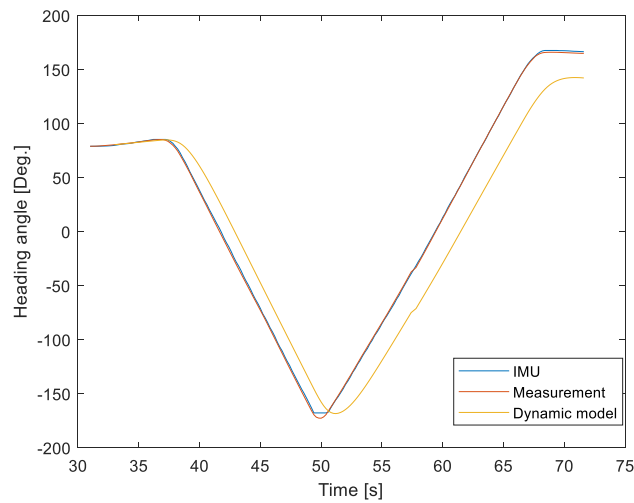


Fig. 4. 21. Comparisons of vehicle heading. The “measurement” vehicle heading uses the measured yaw angle and the slip angle to deduct the heading angle.

C. Experiment 3 – Update the model every 2 s (with data size equals to 10)

To estimate the vehicle dynamics precisely, the online learning mechanism is used to update the vehicle model every 2 s. The data size for system identification is 10.

The measurements and error of estimation are shown in Fig. 4. 22 and Fig. 4. 23. We can see that the updated vehicle model can estimate the results precisely, but the algorithm is sensitive to noise. Additional methods need to be adopted with the updated vehicle model to eliminate the effects of noise.

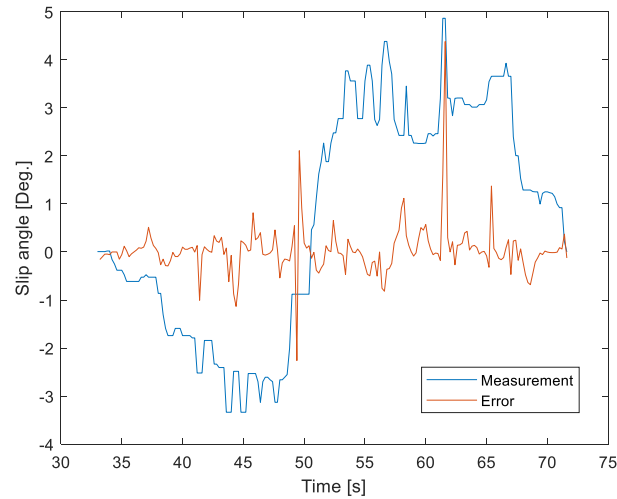


Fig. 4. 22. Error analysis of estimating slip angles with an updated dynamic model.

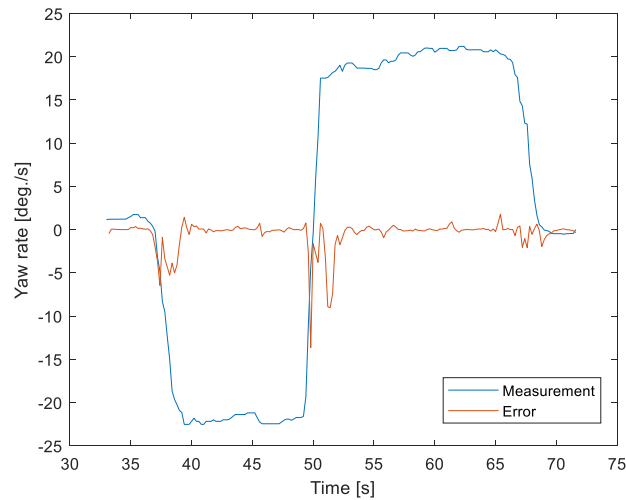


Fig. 4. 23. Error analysis of estimating yaw rate with an updated dynamic model.

C. Identifying the steering model for headland turns

A data-driven steering model derived from the unicycle model is developed in this research for headland turns. It considers the LTL time, delay response caused by the inertia of the vehicle and control units, and the slid during headland turn. Different from the bicycle model, in which the lateral forces in left and right wheels are assumed to be equal in value and direction, the data-driven vehicle model deals with the

movements of the tractor in different directions separately. As the vehicle reverses during turning, the actual turning radius varies according to the lateral tire force of front and rear wheels and the slip angle of the tires. Due to the asymmetric mechanism and the variance of hydraulic steering power, the turning radii of steering to the right and steering to the left are different. Additionally, the soil type or moisture, steering angle, and speed affect the turning radii. Instead of estimating unknown parameters separately, the vehicle (with or without a farming implement) is analyzed as a mass point. Variations of turning radii indicate the change of vehicle-soil interactions.

As the asymmetric mechanism and hydraulic power of the steering part, the radii of turning right and turning left are different in one tractor. In addition, the turning radii in different moving directions are also not the same. To simplify the experiment, the position data of the tractor are recorded at a speed of 1.0 m/s in different soil conditions. In the experiments, the tractor steers at $\sigma = 30, 33, 35, 38$, and 40 deg. to the right and the left. Table 4. 2 shows a sample of the dataset for modeling the vehicle. The data are classified by the steering direction, the steering angle, the direction of the movement, the turning radius of the trajectory, and weights of data in each row. The values of radii in the fifth column are mean values of several experiments at different turning speeds or soil conditions. The weight of each group of data in a row is related to the accuracy of the data. The initial dataset shares the same weight, $1/N_0$ (N_0 being the total number of the initial dataset). The training data plotted in Fig. 4. 24 indicates the relation of turning radii in different directions (forward and backward) at several steering angles (Fig. 4. 24 (a) and Fig. 4. 24 (b)) and the relation of steering angle and turning radius (Fig. 4. 24 (c) and Fig. 4. 24 (d)).

Table 4. 2. A sample of training data.

Weight	Steer	Angle [deg.]	Direction	Radius [m]
$1/N_0$	Left	30	Forward	5.6
$1/N_0$	Right	40	Backward	4.2
...	...	...	...	...

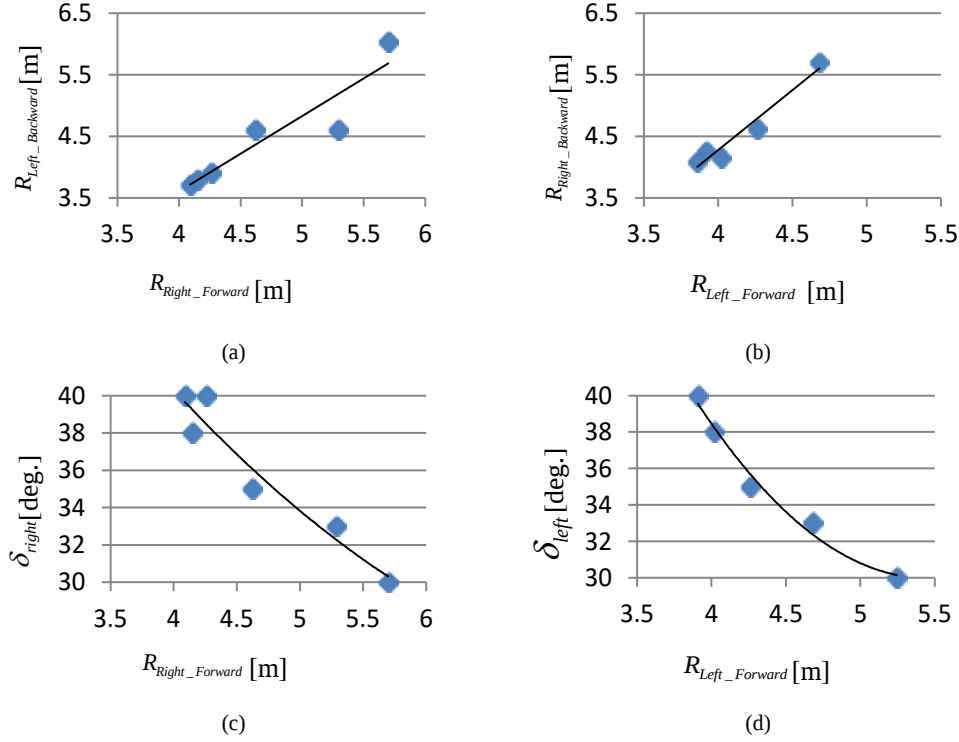


Fig. 4. 24. The variance of turning radii of different steering angles. (a) and (b) show the relations of turn radii in different directions; (c) and (d) show the relations of turning radius and steering angle.

In case of overfitting, a linear function indicates the relation of turning radii in different directions:

$$\begin{aligned} R_{Left_Backward} &= \alpha_0 * R_{Right_Forward} + \beta_0 \\ R_{Right_Backward} &= \alpha_1 * R_{Left_Forward} + \beta_1 \end{aligned} \quad (30)$$

$R_{Left_Backward}$ and $R_{Left_Forward}$ are the turning radii of a turning left maneuver when the tractor moves backward or forward, respectively; $R_{Right_Backward}$ and $R_{Right_Forward}$ represent the turning radii of backward and forward movements with the steering to the right, respectively. Steering angle for headland turning is represented as a quadratic function of turning radius in:

$$\begin{cases} \delta_{right} = s_{00} * R_{Right_Forward}^2 + s_{01} * R_{Right_Forward} + s_{02} \\ \delta_{left} = s_{10} * R_{Left_Forward}^2 + s_{11} * R_{Left_Forward} + s_{12} \end{cases} \quad (31)$$

where δ_{right} and δ_{left} are steering angles on the right and left. Large or maximum steering action is commonly used at headland turns, and experiments in this study reveal that the quadratic function fits the data better than a tangent function in expressing the nonlinear relation of turning radius and steering angle. Eqs. (30) and (31) compose

the vehicle model for designing and optimizing the navigation path. The coefficients of the vehicle model can be rewritten as:

$$r = \begin{bmatrix} \alpha_0 & \beta_0 \\ \alpha_1 & \beta_1 \end{bmatrix} \quad (32)$$

and

$$s = \begin{bmatrix} s_{00} & s_{01} & s_{02} \\ s_{10} & s_{11} & s_{12} \end{bmatrix} \quad (33)$$

During a field test, the vehicle model should be tuned based on the result of each turn. Turning radii of forward and backward movements represent the state of the tractor in the current field. In case raw measured data are not consistent, data from different experiments are processed and combined with the initial dataset. The update of weights in the dataset follows the rules in:

$$\omega_j = \begin{cases} \frac{1}{N_0 + \sum_{i=0}^{T-1} \eta_i}, j \in [0, N_0 - 1] \\ \frac{\eta_i}{N_0 + \sum_{i=0}^{j-N_0} \eta_i}, j \in [N_0, N - 1] \end{cases} \quad (34)$$

$$\eta = \max(0, 1 - \mu * \|R_m - R_m'\|_1)$$

$$s.t. \quad \mu \in [0, 1], N \in [N_0, 2N_0]$$

where μ , set as a constant value, is the learning rate of the dynamic turning method. R_m is the predicted turning radius of the vehicle model, and R_m' is the measured value by Eq. (5). Processed data for calculating the initial vehicle model are stored from number 0 to N_0-1 continuously and share the same value of weight. The real-time measured data indexed from N_0 to $N-1$ are added to the dataset with different weights, which are related to the errors of the model during each field test. In case of overwhelming data, the weights of experiments are limited to a state smaller than the weights of data in the original dataset, and the amount of dataset (N) is defined to be smaller than a threshold (such as two times of N_0 in this study). In this research, the number of the training dataset (N_0) is 11 in this example. The amount of training dataset is defined to be 20, that is to say, the vehicle model would be tuned according to the previous 9 turns. The reinforcement learning mechanism tunes the model and adapts it to different conditions, and also guarantees robustness.

4.2 Vehicle state estimation

Estimating vehicle state is a crucial prerequisite for precision driving. Inaccurate pose estimation leads to poor driving decisions and can cause the vehicle runs out of the route (Thrun et al., 2006). In this research, the vehicle state comprises a total of 11 variables. The design of this parameter space is indicated in Table 4. 3. The RTK-GPS system provides both absolute position, heading, and velocity measurements. The IMU measures the attitude and yaw rate of the vehicle. The error of the yaw angle/heading/orientation measured by the IMU is termed gyro bias.

Table 4. 3 Vehicle state variables

No. of values	State variable
2	Position (longitude, latitude)
2	Velocity (GPS velocity, wheel encoder velocity)
3	Attitude (Euler angles: roll, pitch, and yaw/heading/orientation)
1	Gyro bias
1	Slip angle
1	Yaw angle

The integration of the IMU's gyroscopes for orientation, coupled with wheel velocities and parameters measured by GPS is able to estimate the pose of the vehicle more accurately than by any of the single sensors. The most common approach for this sensor fusion task is the Kalman Filter (KF) or one of its derivatives, i.e., Extend Kalman Filter (EKF) (Schubert et al., 2008). The sensor fusion based vehicle state estimation method can deduct the pose of the vehicle within the order of centimeters accuracy during the GPS outages of up to 2 min long (Thrun et al., 2006).

4.2.1 Estimation of a vehicle heading by the fusion of an RTK-GPS and IMU

The absolute orientation of the vehicle with the GPS coordinate system is also termed vehicle heading. Vehicle heading can be measured according to the accurate position measured by the RTK-GPS or measured by the IMU. However, the IMU heading may not be the same as the vehicle's orientation because of the misalignment of the installment. In addition, the IMU heading drifts along with the time. The misalignment of IMU heading and the drift error are the gyro bias in Table 4. 3. Considering the gyro bias ($\Delta\phi$), vehicle orientation (ϕ) can be calculated by correcting the instant output of IMU (ϕ_{IMU}) as follows.

$$\varphi_k = \varphi_{IMU_k} + \Delta\varphi_k \quad (35)$$

According to Eq. (9), the vehicle's movement from k to $k+1$ can be measured by the difference of GPS position (d_{GPS} in Eq. (36)) or be estimated from the INS (d_{INS} in Eq. (37)).

$$d_{GPS} = \begin{pmatrix} x_{k+1} - x_k \\ y_{k+1} - y_k \end{pmatrix} \quad (36)$$

$$d_{INS} = \begin{pmatrix} \frac{v}{2}(\sin \varphi_k + \sin \varphi_{k+1})\Delta t \\ \frac{v}{2}(\cos \varphi_k + \cos \varphi_{k+1})\Delta t \end{pmatrix} \quad (37)$$

Estimating the gyro bias is to minimize the cost function using the least square method (LSM) as follows (M. Kise et al., 2001; Takai et al., 2011).

$$J_k = \sum_{i=k-N}^k \|d_{GPS} - d_{INS}\|^2 \quad (38)$$

The LSM based vehicle heading identification method is a simple but robust method which uses few sample data. However, the LSM is suitable for linear and weakly nonlinear system with performance loss. An experiment was conducted at Hokkaido University on a concrete road with a half-crawler-type robot tractor. The velocity of the tractor ranges from 1.5 m/s to 3.5 m/s. The steering angle during the zigzag movement is within ten degrees. The heading angles measured by IMU, LSM, and GPS are shown in Fig. 4. 25. The identified bias is shown in Fig. 4. 26. The LSM filters the IMU and GPS data from 16 s after the initialization. It takes ten seconds for the LSM to identify a stable gyro bias.

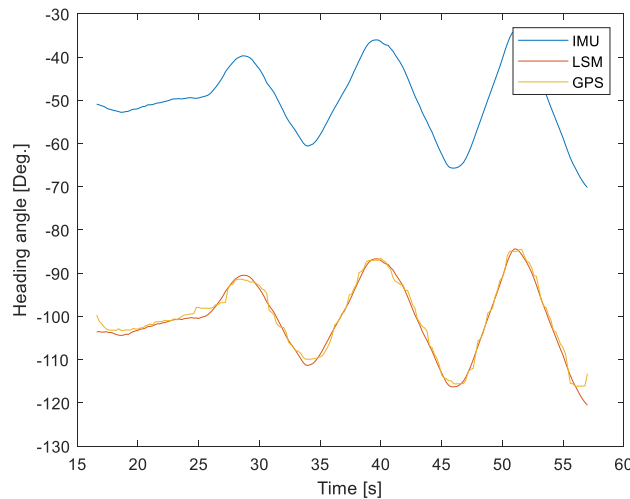


Fig. 4. 25. Comparison of vehicle heading measured by the IMU, LSM, and GPS. LSM combines the raw data of IMU and noisy heading calculated by RTK-GPS.

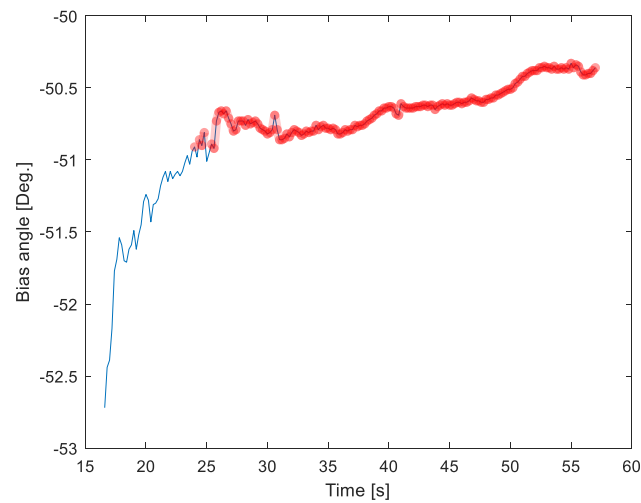


Fig. 4. 26. Time series of gyro bias identified by LSM. The value becomes stable from the red shade area.

4.2.2 Improvement of the vehicle state estimation with sensor fusion methods

The robot tractor movement can be easily affected by the variations of velocity, terrain topography and texture, and so on. Sensor fusion methods, like KF and EKF, are introduced to improve the accuracy of estimating the vehicle's motion.

The steering angle all over the trajectory is shown in Fig. 4. 27.

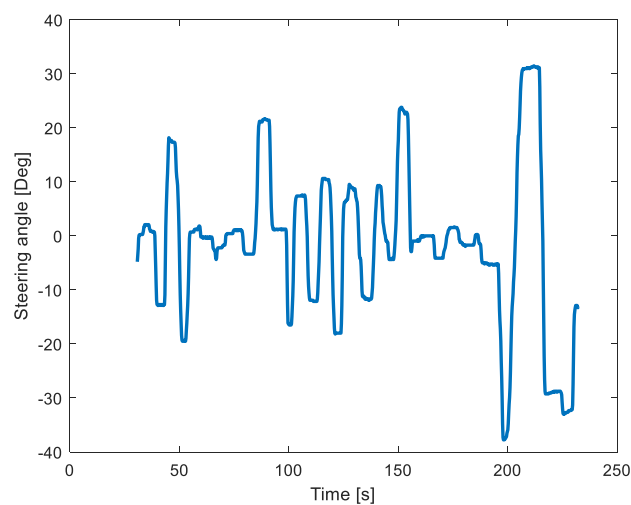


Fig. 4. 27. Time series of steering angle as input.

There are two sources of vehicle heading given by GPS. The one is the records of GPVTG from the NMEA messages. It is termed the VTG heading. The other is calculated from the continuously GPS positions. It is termed the calculated heading. Fig. 4. 28 and Fig. 4. 29 shows the VTG heading angle and the calculated heading angles, separately. A median filter is applied to each of the data series for real-time computing. The two sources are fused by applying a weighted average. The GPS heading after fusion (φ_{GPS} in Eq. (18)) is shown in yellow line in Fig. 4. 30. Comparing with the heading given by IMU, φ_{GPS} is shown in Fig. 4. 31 in blue line. Sideslip angle can be calculated with the filtered GPS heading and the filtered IMU heading. A median filter is also applied to the slip angle calculated from Eq. (18). The raw data of slip angle and the filtered results are shown in Fig. 4. 32.

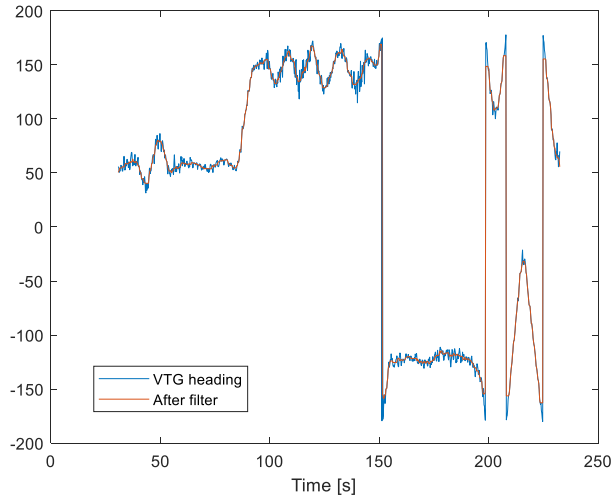


Fig. 4. 28. Time series of VTG heading.

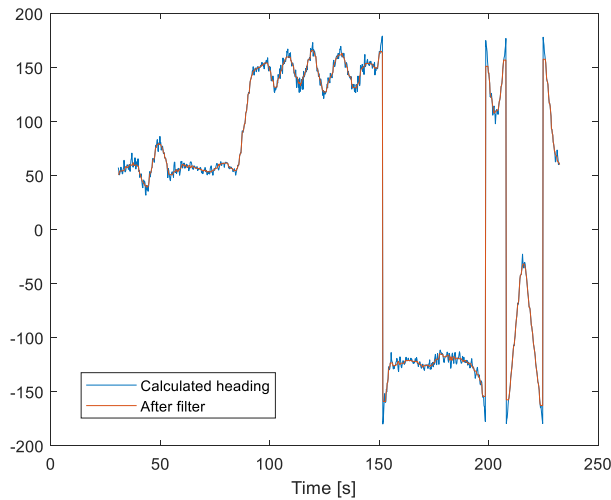


Fig. 4. 29. Time series of calculated GPS heading.

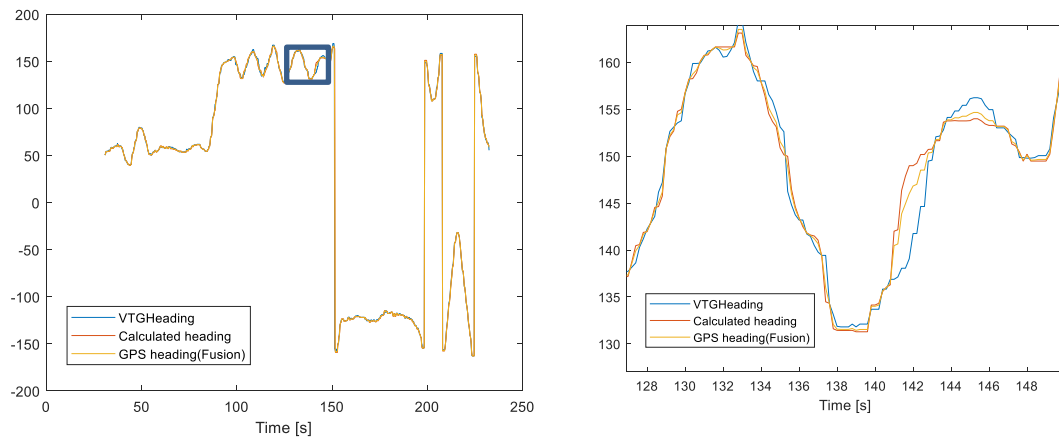


Fig. 4. 30. GPS heading after fusion. The GPS heading is fused from the weighted average of VTG heading and calculated heading. The square area at the left figure is enlarged at the right side.

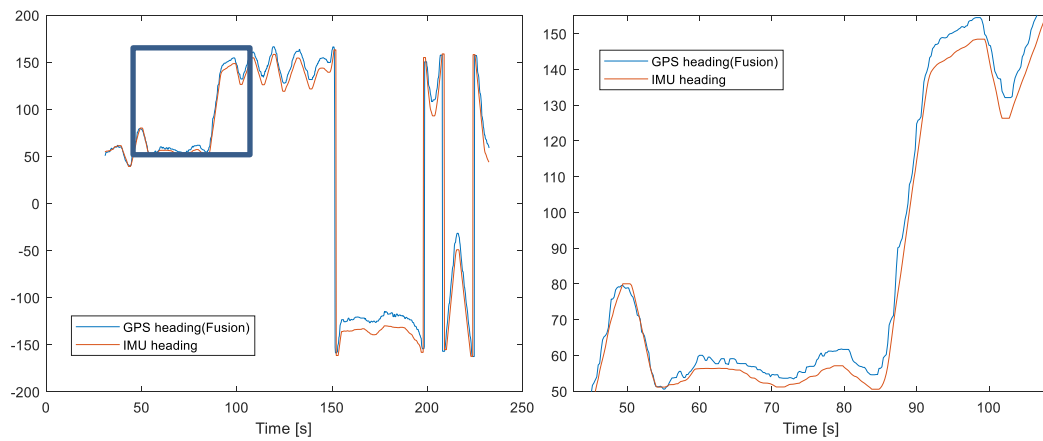


Fig. 4. 31. Comparison of the GPS heading and the IMU heading. The square area at the left figure is enlarged at the right side.

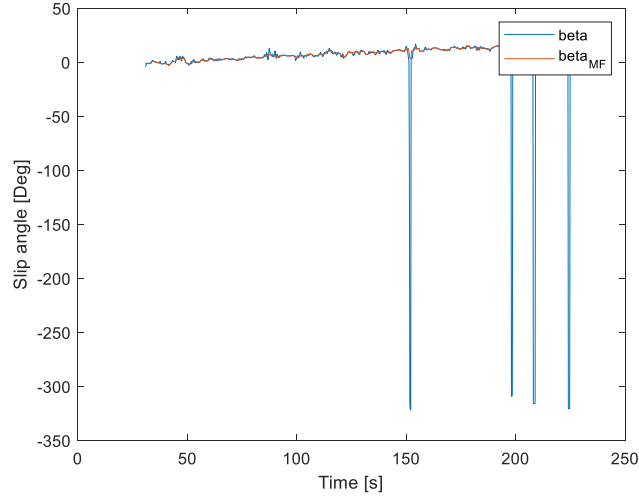


Fig. 4. 32. Time series of calculated slip angles (beta). The red line indicates the slip angle after the median filter.

The yaw rate is the angular rate in z-axis that can be measured by the IMU, shown in Fig. 4. 33. A median filter is applied to clean the data.

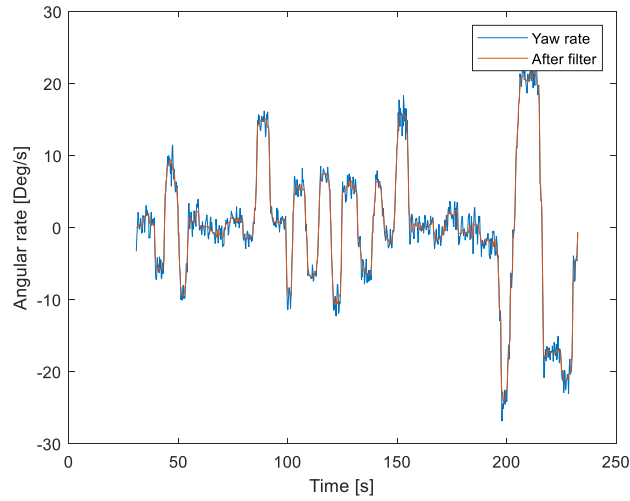


Fig. 4. 33. Time series of yaw rate. The red line indicates the yaw rate after the median filter.

In this experiment, the vehicle position is deducted using the original dead reckoning method in Eq. (39). Vehicle's heading can be from the sensor (GPS or IMU) or calculated by Eq. (40).

$$x = x_0 + \int v \sin(\varphi + \beta) dt \quad (39)$$

$$y = y_0 + \int v \cos(\varphi + \beta) dt$$

$$\varphi = \varphi_0 + \int \omega dt \quad (40)$$

Since the GPS heading are derived from the position measured by RTK-GPS, the accumulated error of GPS heading should be “0” in theory. However, the GPS heading is noisy and do not fit the constraints of a 2D vehicle model. Therefore, GPS heading cannot be used directly for real-time navigation. In addition, the drift of IMU heading becomes serious in long-duration usage. The blue line in Fig. 4. 34 indicates the estimated position by Eq. (39), and the heading angles are the raw data from IMU. The “IMU heading” trajectory deviated from the ground truth because of the cumulative errors. For the same reason, the trajectory calculated from the measured slip angles and yaw rates, the purple line termed “Omega+Beta”, also deviated from the ground truth. Owing to small cumulative errors, the estimated trajectory calculated from the LSM heading performs better than other methods. However, the LSM method assumes the vehicle moves in a linear movement. Therefore, it cannot follows the curve path precisely, as shown in the right figure in Fig. 4. 34.

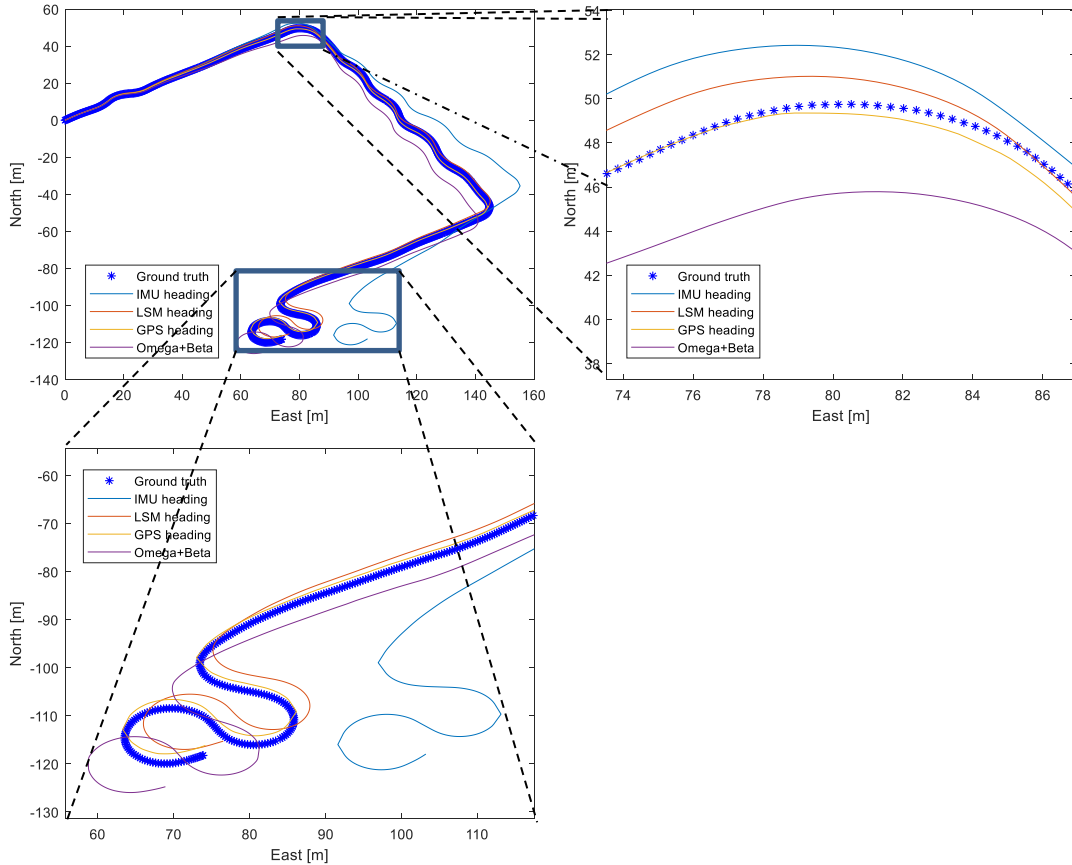


Fig. 4. 34. Comparison of estimated vehicle trajectories according to Eq. (16) and Eq. (40). The sources of vehicle heading include IMU, LSM method, and filtered GPS heading. For the Omega+Beta, the data sources are the yaw rate from filtered IMU angular rate and the measured slip angle according to Eq. (18).

To improve the measurement of vehicle heading and the deduction of vehicle's position, the sensor fusion methods such as KF and its modifications, EKF and Unscented Kalman Filter (UKF) (Noguchi et al., 1998; Thrun et al., 2006). The KF combines measurements and estimations of parameters with Bayesian inference and iterates over the two probabilistic process steps, which are the prior prediction step and the posterior update step. In the first step, the KF predicts the state of vehicle based on the dynamic model (Eq. (24)). It also predicts the uncertainty by estimating its covariance. The prediction step is described as follows:

$$\begin{aligned}\hat{X}_{k|k-1} &= \begin{bmatrix} \beta_{k-1} \\ \omega_{k-1} \end{bmatrix} = A_k \hat{X}_{k-1|k-1} + B_k u_k \\ P_{k|k-1} &= A_k P_{k-1|k-1} A_k^T + Q_k\end{aligned}\tag{41}$$

Q_k is the covariance matrix of the system which accounts for uncertainties in the process. The next step is to compare the prediction with the measurement and to update the estimation of the state. The measurement/observation state function can be described by:

$$\begin{aligned}Z_k &= H X_k \\ H &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\end{aligned}\tag{42}$$

The transformation from the estimation space to the measurement space is linear. The update of vehicle state and covariance follows the equation:

$$\begin{aligned}\hat{X}_{k|k} &= \hat{X}_{k|k-1} + K_k (Z_k - H \hat{X}_{k|k-1}) \\ P_{k|k} &= (I - K_k H) P_{k|k-1} \\ K_k &= P_{k|k-1} H^T (H P_{k|k-1} H^T + R_k)^{-1}\end{aligned}\tag{43}$$

K_k is the Kalman Gain, which is the weight of the prediction to the measurement. R_k is the covariance matrix of observation noise that describes the accuracy of measurements. The yaw rate and slip angle can be estimated based on the vehicle

model in Eq. (24). Based on a default vehicle model, the estimated vehicle heading angle after KF is shown as the orange line in Fig. 4. 35. With the online-learning ability, the vehicle's dynamic model can be updated based on the real-time data other than a constant value model. The heading angle measured by KF with the updated model is termed KF+Updated. The estimated trajectory based on the Eq. (39) is shown in Fig. 4. 36. The KF method can follow the ground truth precisely at the straight path as shown in Fig. 4. 36(b). However, it drifted because of cumulative errors during the continuous curves. By learning the data in real time, the KF+Updated trajectory follows the ground truth from the beginning to the end with small cumulative errors. Therefore, both the KF and KF+Updated can improve the estimation of the vehicle heading. And the estimation is more robust for the non-linear movement with an updated vehicle model. For a further step, the fusion methods (such as the EKF and UKF) can be adopted once again when the accurate position and orientation information are available. The state transform function of Eq. (39) and Eq. (40) can be described by the Eq. (8). In EKF, we approximate the transformation function f by a Jacobian matrix J .

$$J_k = \begin{bmatrix} 1 & 0 & v \cos(\varphi_k + \beta_k) \Delta t \\ 0 & 1 & -\sin(\varphi_k + \beta_k) \Delta t \\ 0 & 0 & 1 \end{bmatrix} \quad (44)$$

The measurement state includes the position from GPS and the heading angle from IMU. The transform function of the observation process is a linear equation as follows:

$$Z_k = H X_k$$

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (45)$$

The prediction of EKF is described as follows:

$$X_{k|k-1} = f(X_{k-1|k-1}, \hat{\epsilon}_k) \quad (46)$$

$$P_{k|k-1} = J_k P_{k-1|k-1} J_k^T + Q_k$$

The update step of EKF is the same as the KF in Eq. (43). For comparison, the dynamic vehicle model with default value as well as the online updated dynamic vehicle model is tested in this research. The estimated vehicle headings of both EKF and EKF with updated vehicle model (EKF+Updated) are shown in time series in Fig. 4. 37. Moreover, the estimated vehicle positions using Eq. (16) are shown in Fig. 4. 38. It

can be seen that both trajectories of EKF and EKF+Updated follow the ground truth well. It means the cumulative heading errors are small for both methods. In addition, the positioning in North-East measured by GPS is adopted in EKF. It can also estimate the position by combining the estimation of the vehicle model and the GPS information. Compared with the ground truth, the estimated position by EKF and EKF with updated vehicle model are shown in Fig. 4. 39. Two methods can estimate the vehicle precisely with the combination of measured position from the GPS and vehicle model. The overall positioning accuracy of EKF and EKF+Updated is listed in Table 4. 4. We can conclude that the EKF with/without an updated vehicle model can estimate the vehicle heading and vehicle position in high accuracy.

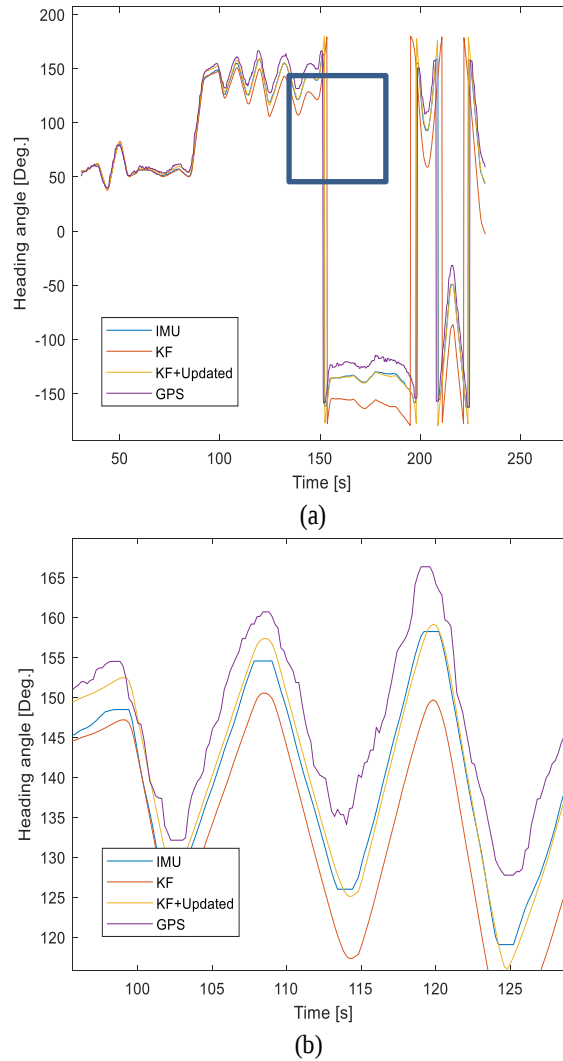


Fig. 4. 35. Time series of vehicle heading angles. The square area in (a) is enlarged in (b). Fusing the measured yaw rate and slip angle and the estimated values from the default vehicle model, the Kalman Filter gets a better estimation of heading,

termed KF. Different from KF, KF+Updated is the estimated heading angles by a real-time updated vehicle model. It is between the GPS and the IMU with a smooth yellow line.

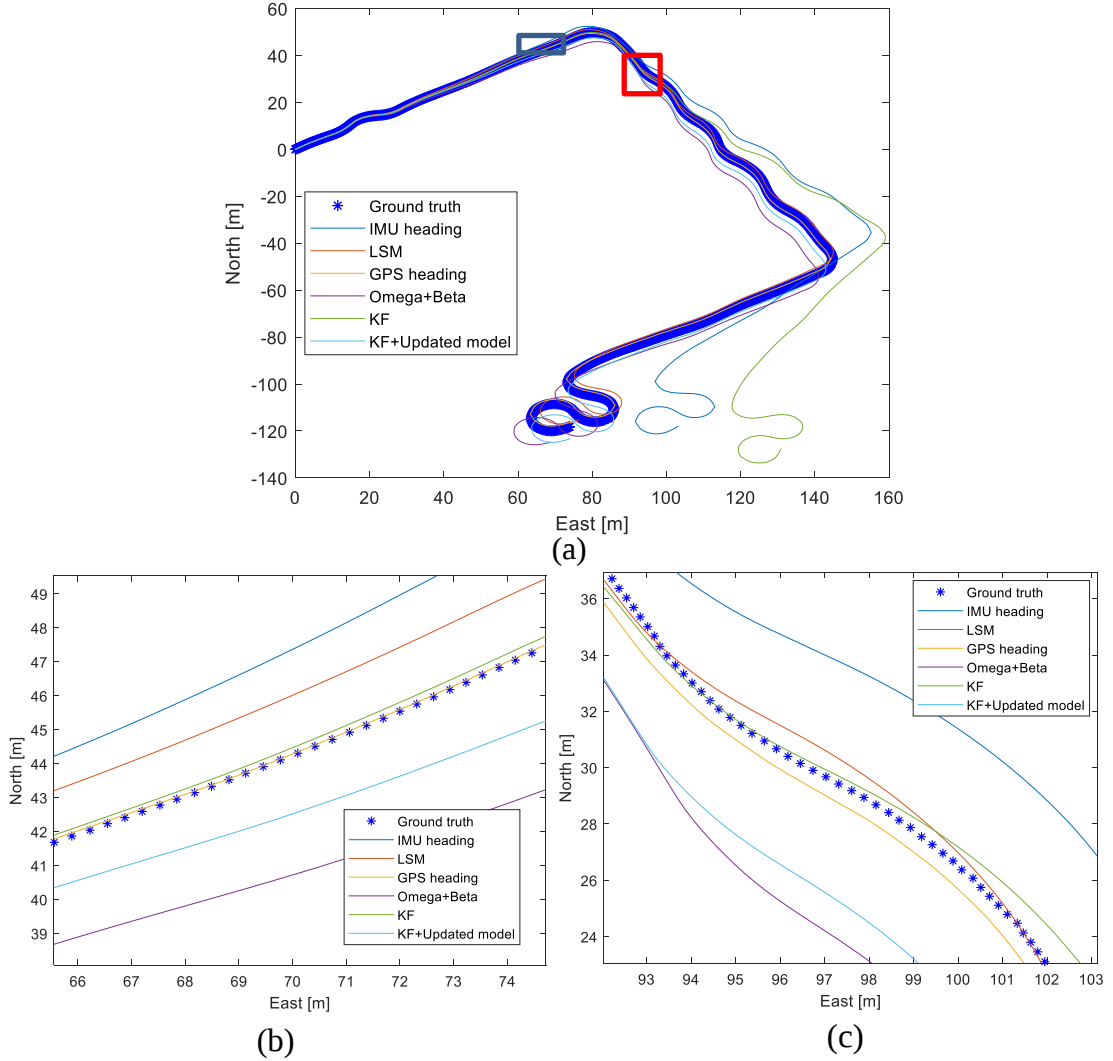


Fig. 4. 36. Comparison of estimated vehicle trajectories according to Eq. (39). The blue square area and the red area in (a) are enlarged in the (b) and (c), respectively.

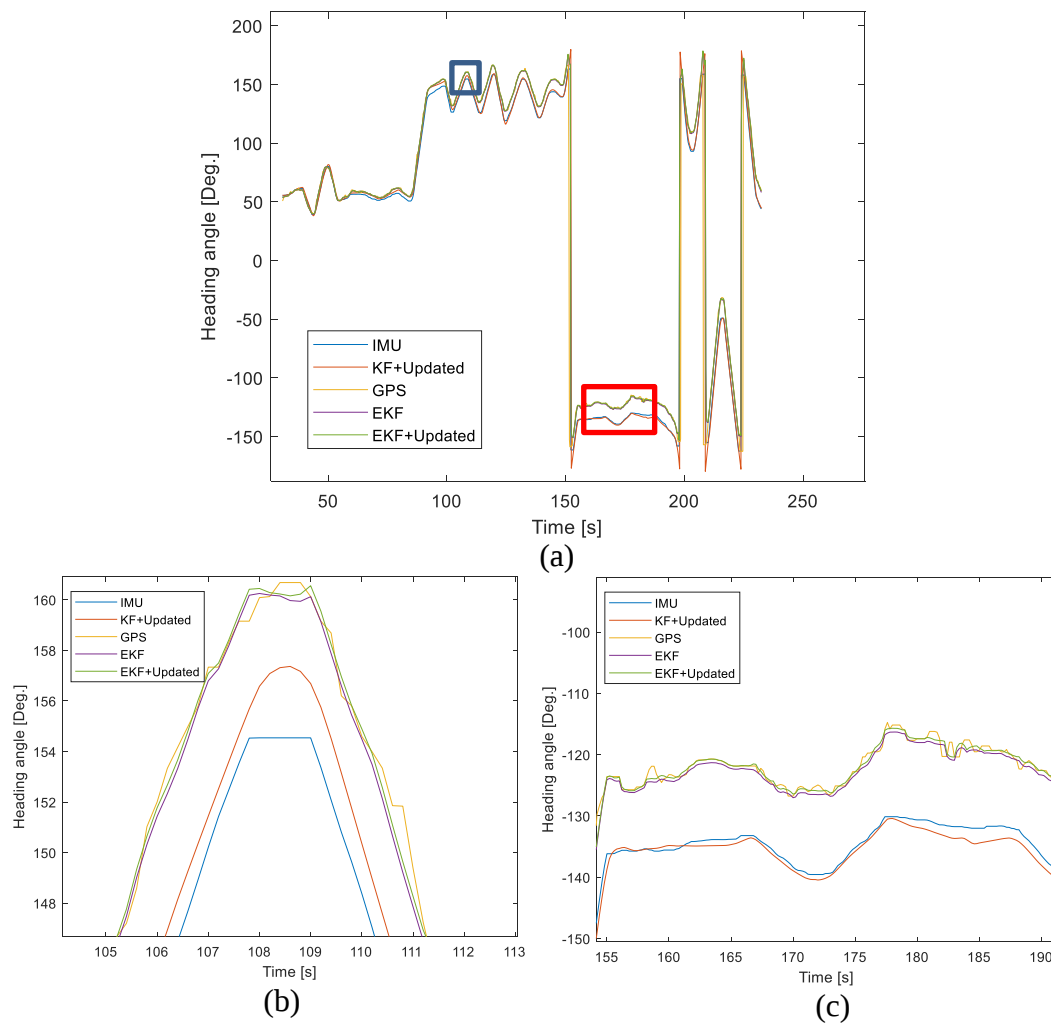


Fig. 4. 37. The comparison of vehicle heading. EKF heading combines the estimation of vehicle heading and the measurement of heading from IMU. The estimation of vehicle heading is from the KF mentioned above. EKF+Updated uses the estimation states by the real-time updated dynamic model. Both the EKF and EKF+Updated can make a fusion of IMU and GPS heading.

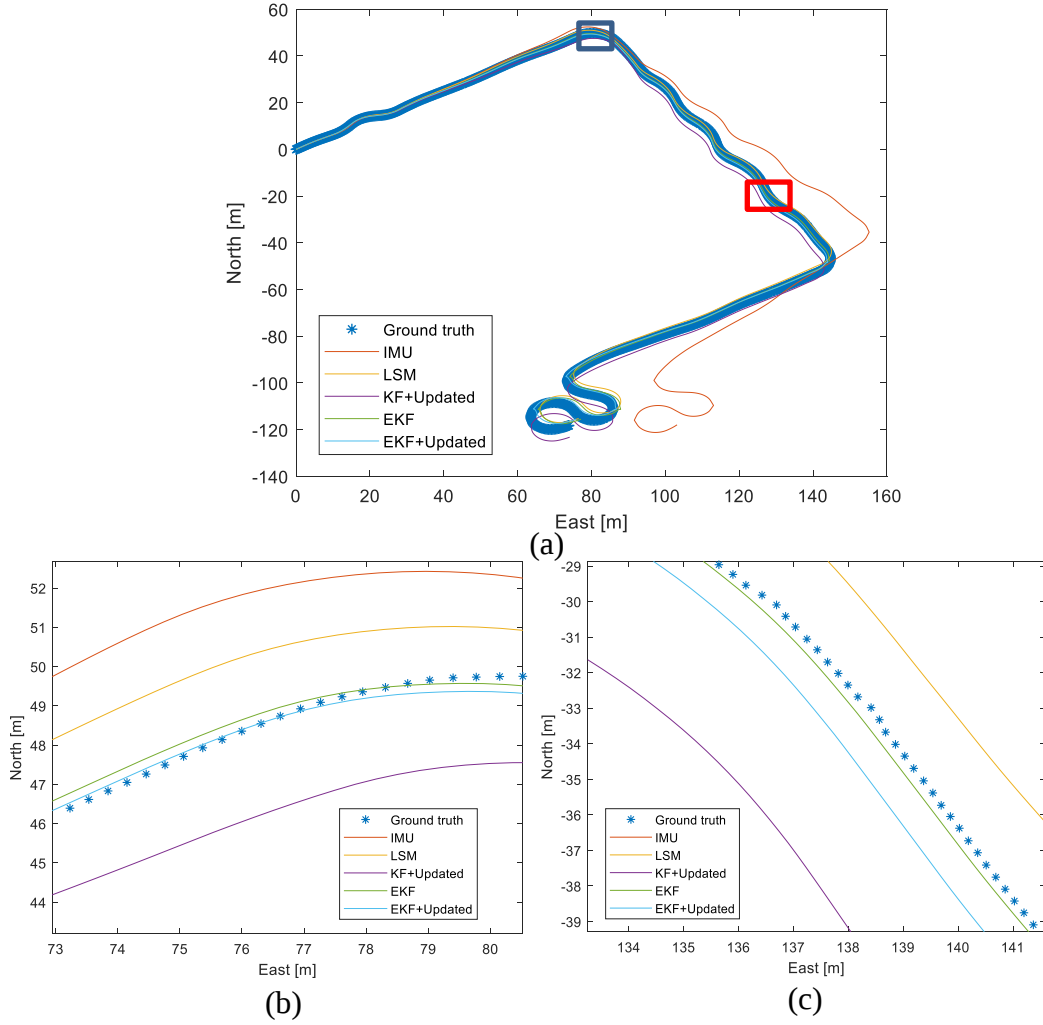


Fig. 4. 38. Comparison of estimated vehicle trajectories according to Eq. (39). The sources of vehicle heading include the raw data of IMU heading, LSM method, KF with updated vehicle model, EKF and EKF with updated vehicle model. The trajectory generated based on EKF heading oscillate between two sides of the ground truth. However, the oscillation is small within centimeter level. The heading calculated by the EKF+Updated is more stable than the EKF heading. The EKF+updated trajectory goes along with the ground truth without oscillation.

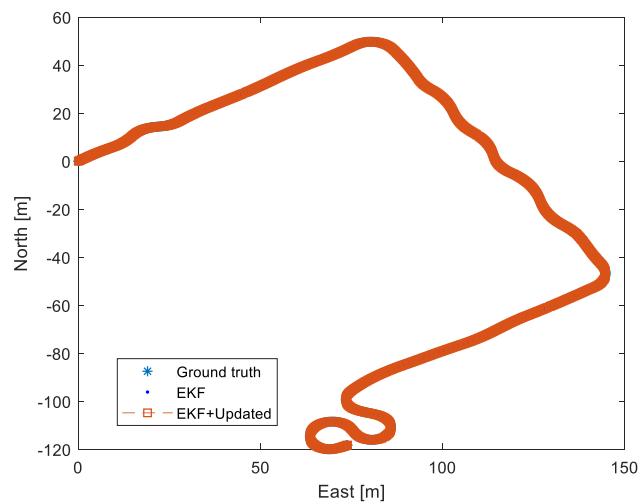


Fig. 4. 39. Comparison of estimated vehicle trajectories.

Table 4. 4 Error analysis of EKF based positioning methods.

	EKF	EKF+Updated
RMS [cm]	0.4	0.4
Max. [cm]	6.6	6.6

To further evaluate the proposed vehicle state estimation methods, a series of experiments were conducted at the experiment farm of Hokkaido University. Fig. 4. 40 shows four scenarios for accuracy comparison. The LSM method works well for the straight path because it simplifies the movement of the tractor during each time interval is linear. Combining the identified vehicle dynamic model, the EKF can estimate the vehicle state precisely even during the GPS is not fix. The positioning results are shown in Fig. 4. 42.

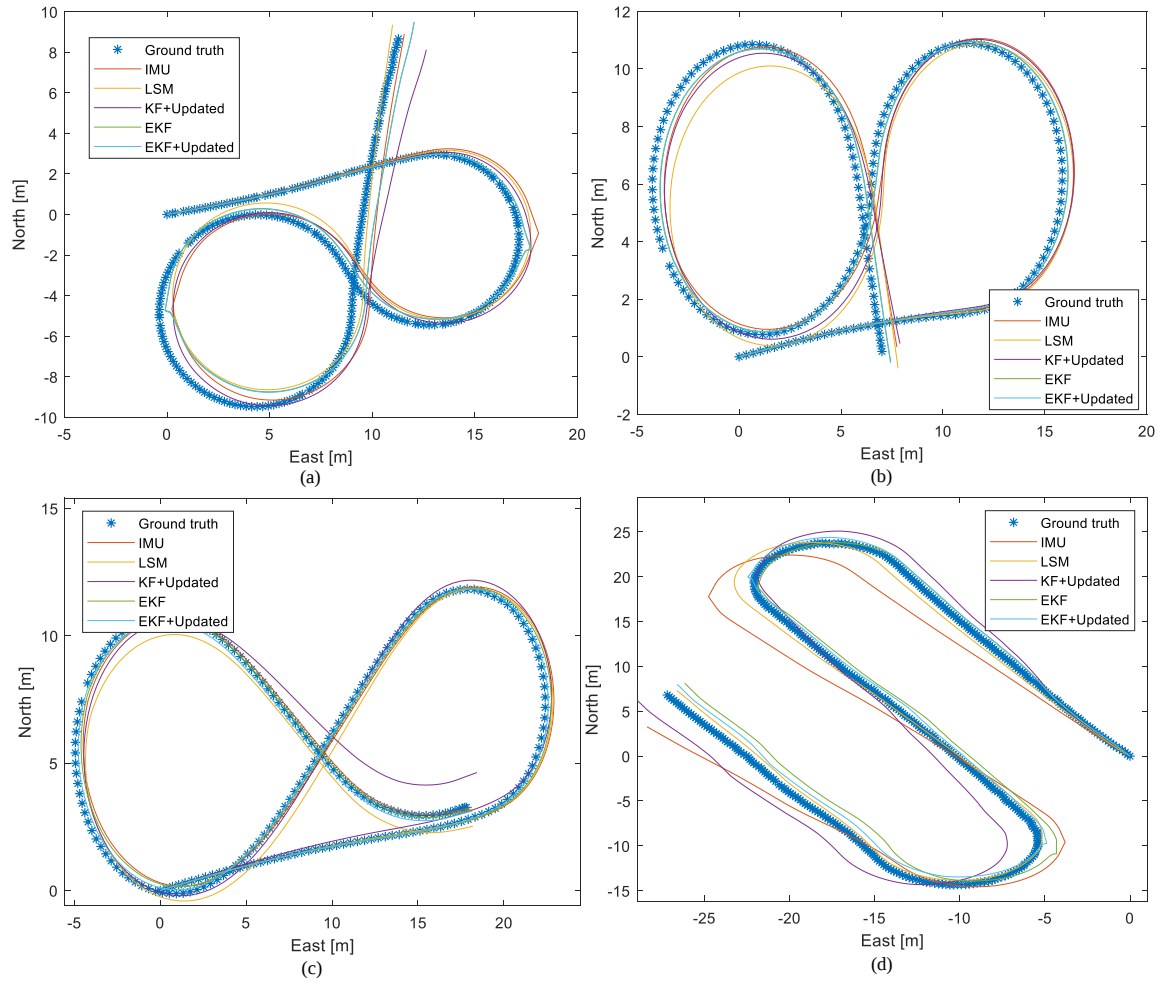


Fig. 4. 40. Four experiments of algorithm evaluation using a four-wheel-type robot tractor. LSM method cannot estimate the vehicle's heading precisely because of the vehicle's non-linear movement.

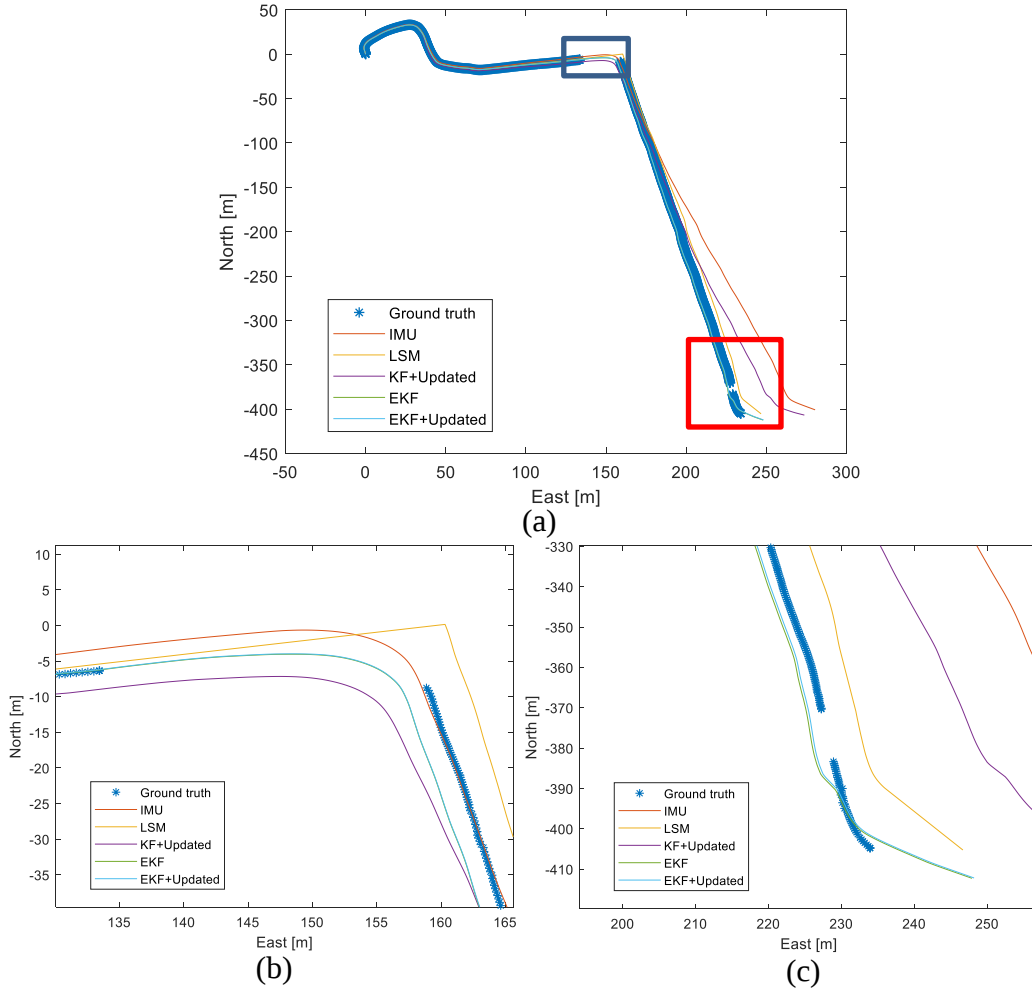


Fig. 4. 41. Comparison of estimated vehicle trajectories according to Eq. (39). Interspaces of the ground truth at the figure (b) and (c) indicate the RTK-GPS is not available. The LSM works when the RTK-GPS is available. During the turning at figure (b), the LSM heading is a constant value, which is the instant heading before the un-fix solution of GPS at (135,-6). When the RTK-GPS is not available, the vehicle model does not update, so the EKF and EKF+Updated are the same during the turning at figure (b).

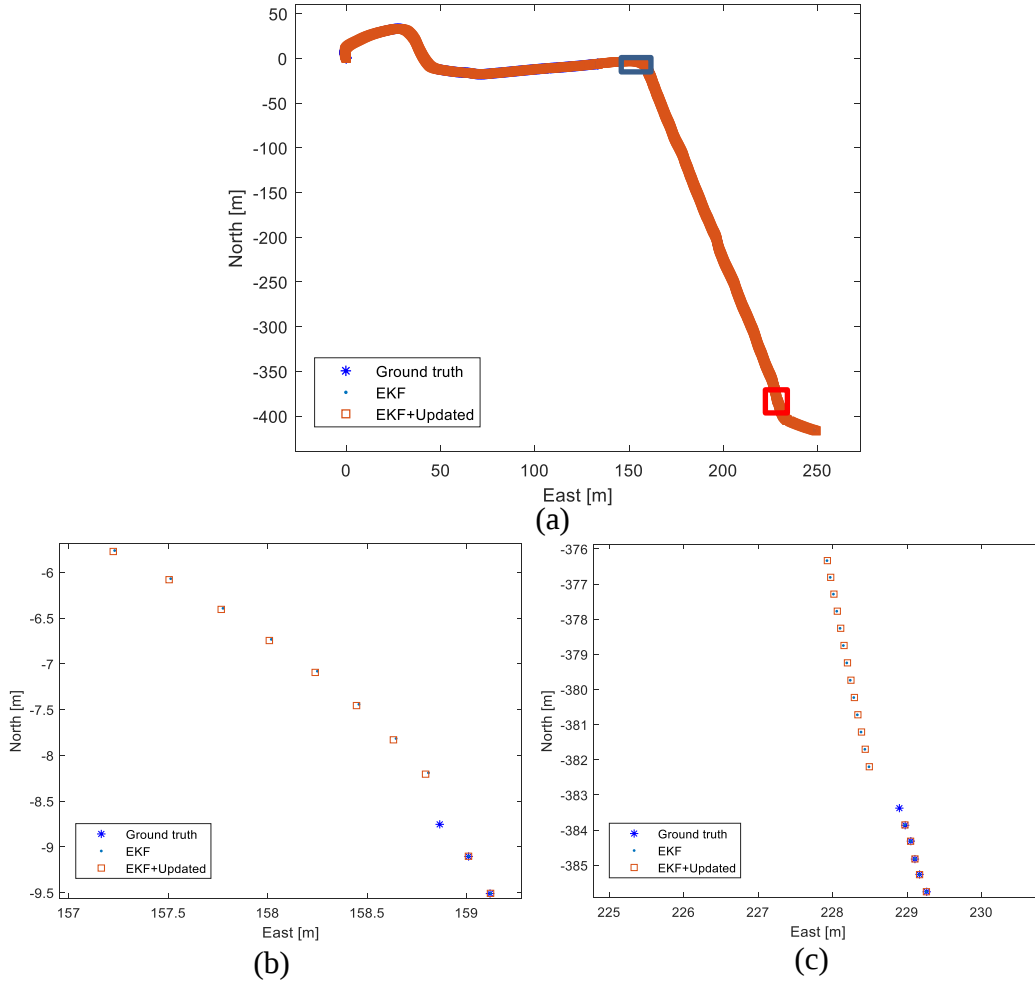


Fig. 4. 42. Comparison of estimated vehicle trajectories according to Eq. (39).

The methods are also tested on a half-crawler-type robot tractor (EG453). The crawler-type vehicle makes use of skid and slips during turning. It is challenging to model vehicle dynamics using a simple vehicle model. The vehicle was tested on a concrete road with speed ranges about 1 m/s to 3 m/s. The vehicle steers within a small degree, as shown in Fig. 4. 43. The snake-like trajectory is shown in Fig. 4. 44. Both the EKF and EKF+Updated can follow the ground truth well with a maximum deviation of 1.7 cm. The estimated vehicle position using different heading sources by Eq. (16) are listed in Fig. 4. 45. EKF and the EKF+Updated can accurately predict the vehicle's state without cumulative positioning error. In addition, the proposed LSM method can also accurately estimate the vehicle heading as well as the position in this condition.

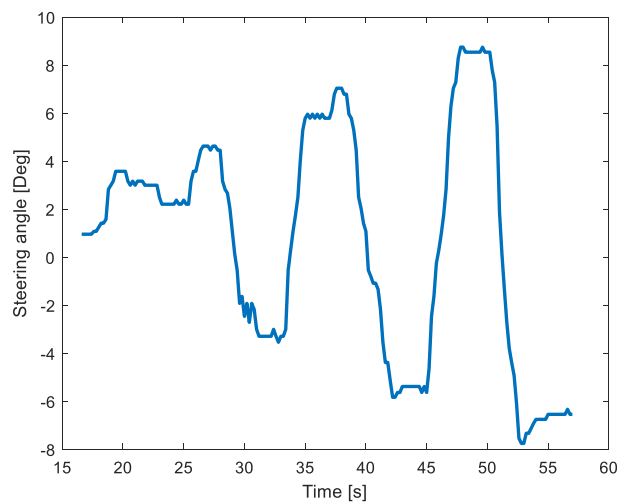


Fig. 4. 43. Time series of steering angles. Steering to the left is defined to be minus.

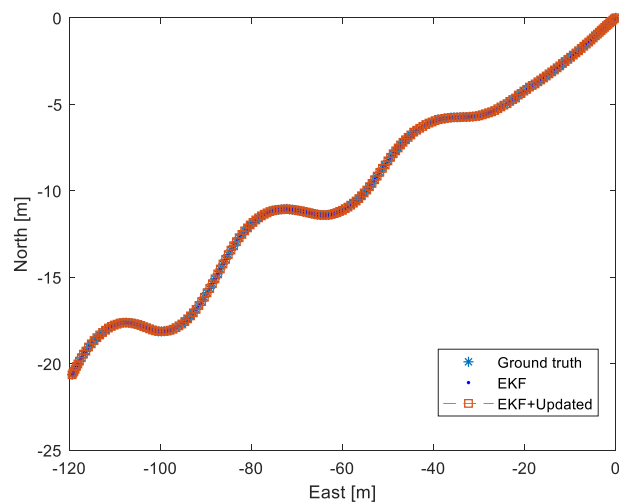


Fig. 4. 44. Estimated vehicle positions based on EKF methods. The ground truth is the vehicle's position measured by an RTK-GPS. The initial position of the vehicle is at (43.0727439, 141.3382012), which is the global position of these figures' origin.

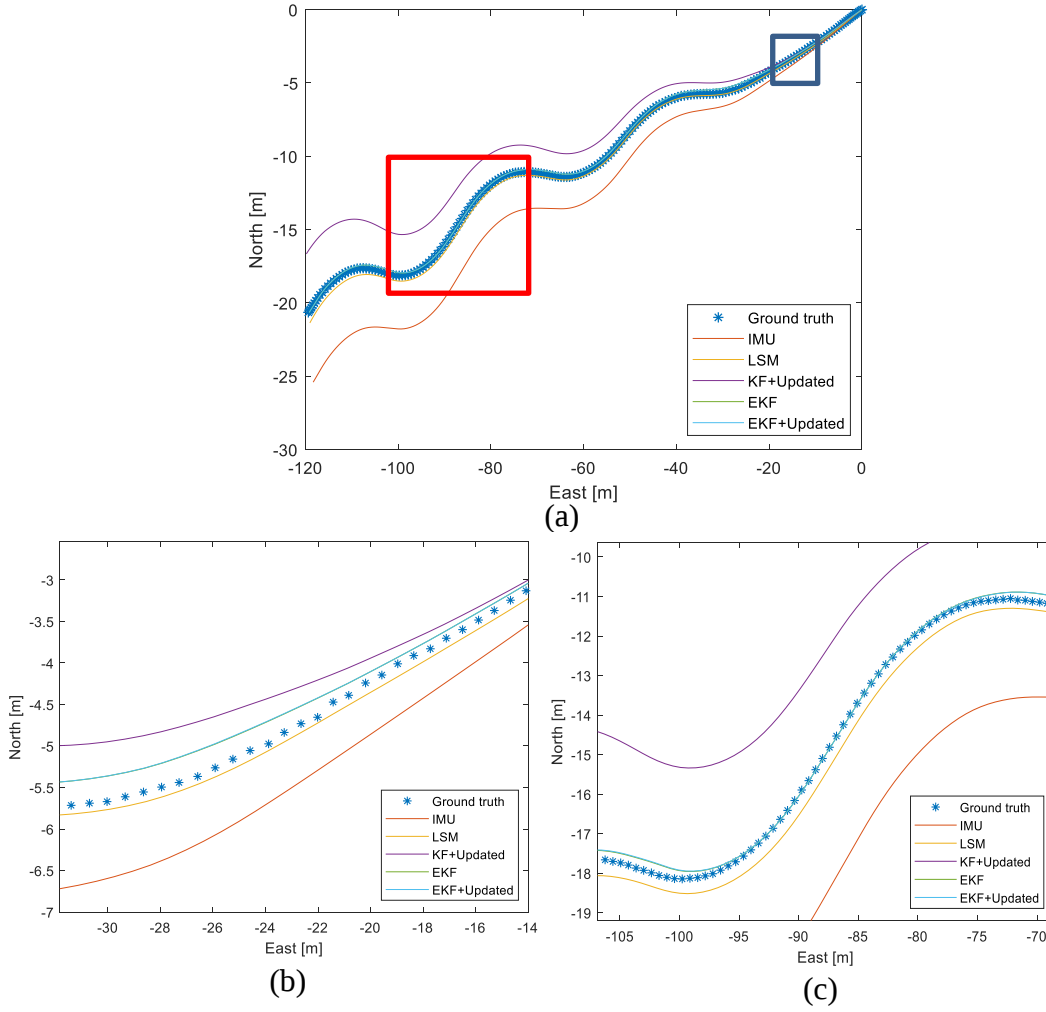


Fig. 4.45. Comparison of estimated vehicle trajectories according to Eq. (39). The raw data of IMU heading drift during the movement. The drift affects the calculation of slip angles and the estimation of the vehicle model. Therefore, the KF+Updated method could not work well in this case.

The proposed state estimation methods are evaluated at the significant steering angle conditions, as shown in Fig. 4.46. The vehicle positions estimated by EKF and EKF+Updated are shown in Fig. 4.47. The deduced vehicle positions using estimated heading information are shown in Fig. 4.48. The KF+Updated estimate vehicle more precisely than other methods, according to Fig. 4.48(b) and Fig. 4.48(c). It owes to the accurate measurement of slip angle and yaw rate in this experiment. However, KF+Updated method is more sensitive to the noise of measurement than the EKF and EKF+Updated. The deduced positioning accuracy in Fig. 4.48 is listed in Table 4.5. The overall performance of KF+Updated is not as good as the EKF and the EKF+Updated. Consistent with the previous experiment, the EKF and the

EKF+Updated are almost the same in positioning the vehicle when the GPS information is available, as shown in Table 4. 6.

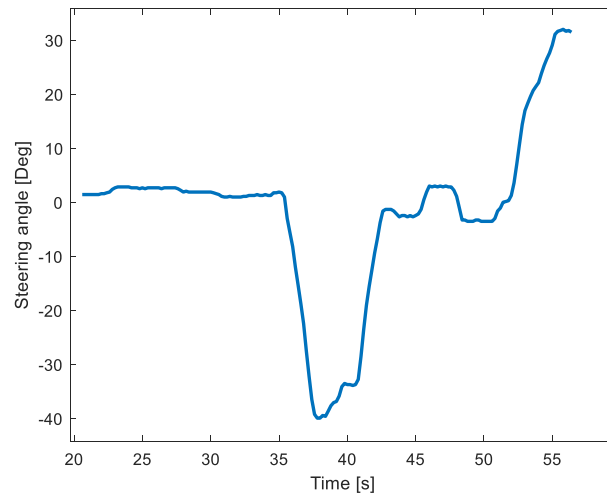


Fig. 4. 46. Time series of steering angles. The half-crawler-type robot tractor runs at the concrete road at the velocity ranging from 1 m/s to 3 m/s. Steering angle ranges from -40 deg. (left side) to 40 deg. (right side).

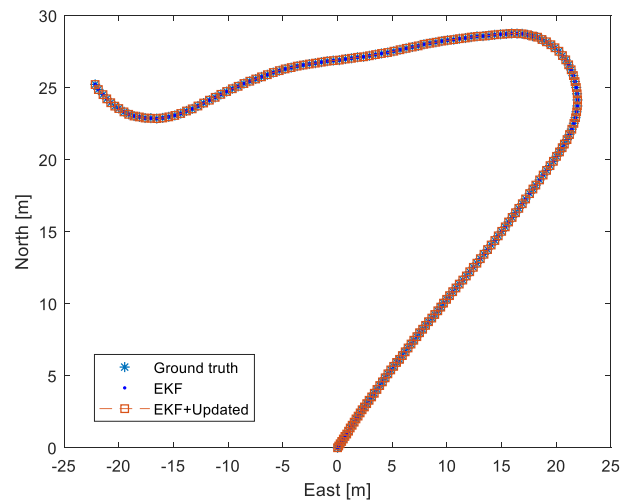


Fig. 4. 47. Estimated vehicle positions based on EKF methods. The origin of this trajectory is from the (43.06928773, 141.3407263) in the latitude and longitude coordinate.

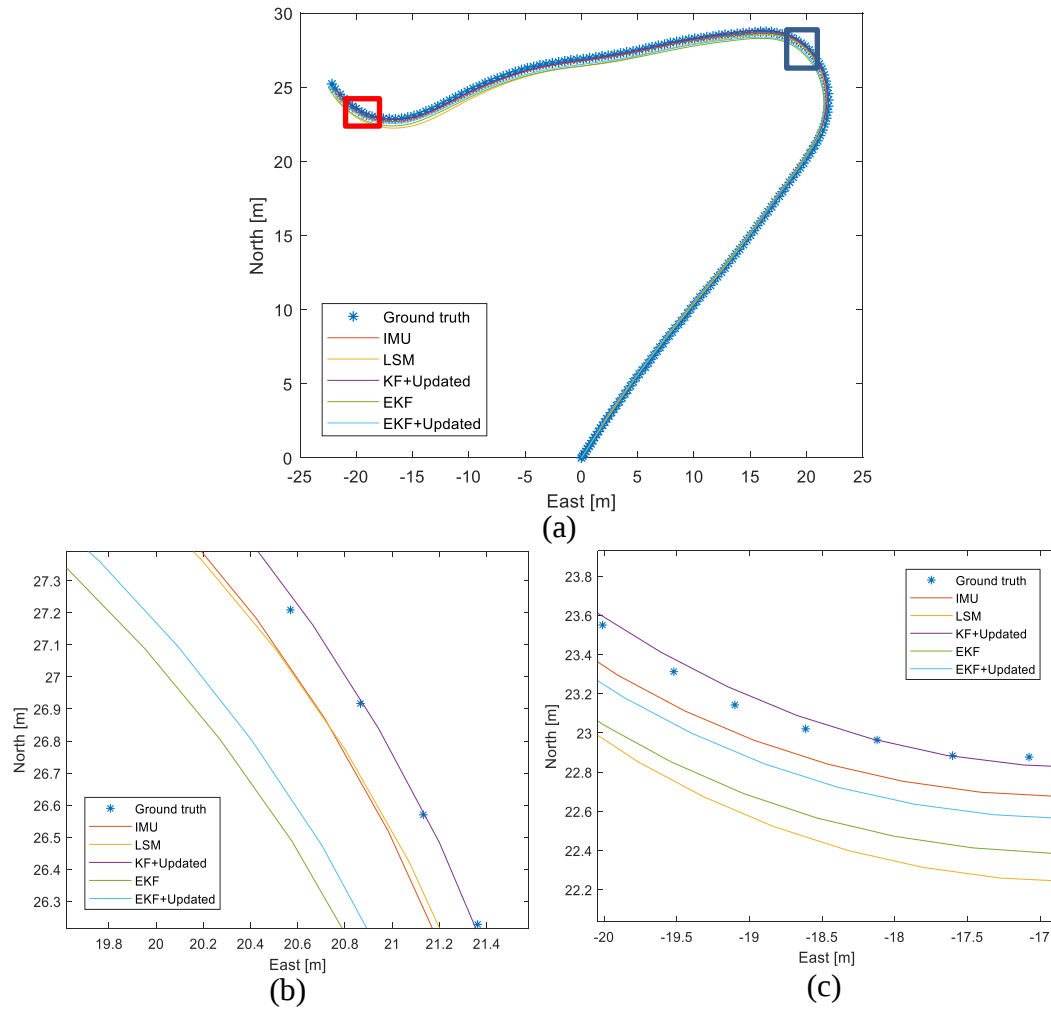


Fig. 4. 48. Comparison of estimated vehicle trajectories according to Eq. (39). IMU does not drift during this short period (about 60 s). The LSM methods precisely estimate the vehicle heading during the straight path and the small steering part. Because of two times small radius turning, the cumulative heading error of LSM is much larger than other methods' error. The KF+Updated based positioning results fluctuate along with the ground truth.

Table 4. 5 Positioning accuracy using Eq. (39).

	KF+Updated	EKF	EKF+Updated
RMS [cm]	70.5	48.7	45.4
Max. [cm]	107.5	77.7	77.1

Table 4. 6 Error analysis of EKF based positioning methods

	EKF	EKF+Updated
RMS [cm]	0.4	0.4
Max. [cm]	1.6	1.6

Chapter 5 Control algorithms for agricultural robots

For an agricultural vehicle, the control problem is usually described as steering control, also termed lateral control. In the case of vehicle fleets or multi-robot cooperation, the velocity control or the longitudinal control will be necessary. The lateral controller in this research computes the steering angle command and adjusts the pose of a vehicle to track a reference path, given the desired vehicle's orientation and position. The intuition of the lateral controller is to minimize the difference between the real vehicle state and the reference. This chapter is not to develop new control algorithms, but to introduce applicable control algorithms and to evaluate the path planning method and state estimation methods through a series of field experiments.

5.1 The path tracking algorithm

Various optimal controllers, such as the linear quadratic regulator (LQR) controller, the linear or nonlinear model predictive controller (MPC) are created for trajectory tracking (ISHII et al., 1998; Karkee and Steward, 2010; Kise et al., 2002b; Lages and Vasconcelos Alves, 2006; Tashiro, 2013). These algorithms use a set of modifications in the robot model, cost function, and optimizer aiming to minimize the steady-state error in a specific predicted interval (Nascimento et al., 2018). Take the MPC as an example (Kaiser et al., 2018; Kühne et al., 2004; Nascimento et al., 2018). The cost function can be defined as a quadratic function of the states, control input, and the input rate:

$$\begin{aligned} \min_{\hat{u}(\cdot|x_j)} J(x_j) = & \min_{\hat{u}(\cdot|x_j)} \left[\sum_{k=0}^{N_p} \|\hat{x}_{j+k} - \bar{x}_k\|^2 Q \right. \\ & \left. + \sum_{k=1}^{M_c-1} \left(\|\hat{u}_{j+k}\|^2 R_u + \|\Delta\hat{u}_{j+k}\|^2 R_{\Delta u} \right) \right] \end{aligned} \quad (47)$$

where N_p is the prediction horizon. M_c is the size of the control sequence during the prediction horizon. The cost function J includes two items, and each is computed as the weighted norm of a vector, i.e., $\|u\|^2 R_u = u^T R_u u$. Q , R_u and $R_{\Delta u}$ are weighting matrices, with $Q \geq 0$, $R_u > 0$ and $R_{\Delta u} > 0$. The cost function penalizes the deviations of the predicted state $\hat{x} = \{\hat{x}_0, \dots, \hat{x}_{N_p}\}$ along with the reference state $\bar{x} = \{\bar{x}_0, \dots, \bar{x}_{N_p}\}$ and also the expenditures of control input $\hat{u}(\cdot|x_j) = \{\hat{u}_{j+1}, \dots, \hat{u}_{j+M_c}\}$ and the input rate $\Delta\hat{u}_k = \hat{u}_k - \hat{u}_{k-1}$. For a whole navigation

procedure shown in Fig. 4. 1, the MPC controller generate the optimal control input based on the current error from reference state and estimated state error based on the simulator's prediction. The MPC is flexible in the formulation of the objective function with user-defined constraints. In addition, it is also suitable for linear and nonlinear systems with an identified model.

Prediction of the vehicle state is not essential for a path tracking controller. For example, in this research, a steering controller is based on a feedback function of the lateral error and heading error, as shown in Fig. 5. 1. The lateral error, d , is the distance from the tractor's COG position to the closest point on a navigation path. It is also termed a cross-track error (Thrun et al., 2006). The heading error, $\Delta\varphi$, is the difference between the reference heading and current heading angle in the ground coordinates. The reference heading follows the curvature of a certain distance of navigation path. The length of the curvature l is termed look ahead distance. That is to say, the reference point is l meters ahead of the orthogonal projection of the current robot position on the map. The steering angle is calculated based on a lateral error and a heading error bearing from the navigation map by

$$\begin{aligned}\Delta\varphi &= \varphi_d - \varphi \\ \Delta\Psi &= k_\varphi\Delta\varphi - k_d d\end{aligned}\tag{48}$$

where, k_φ and k_d are empirical values that are different from each other for different platforms, with $k_\varphi > 0$ and $k_d > 0$.

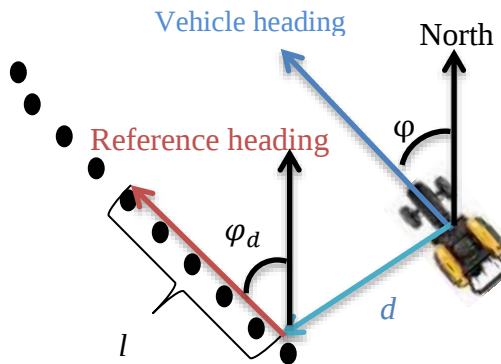


Fig. 5. 1. Lateral controller for a robot tractor

Similarly, a nonlinear feedback function of lateral error can be used in steering control (Hoffmann et al., 2007; Thrun et al., 2006). The control law is given by:

$$\Delta\Psi = \Delta\varphi + \tan^{-1} \frac{k_d d}{v} \quad (49)$$

where v is the current velocity of the vehicle. Different from the Eq. (48), the control law adjusts the steering angle in nonlinear proportion to the lateral error. Moreover, the error converges exponentially to zero. Both of the Eqs. (48) and (49) works well for low speeds.

5.2 Dynamic circle-back turning the algorithm

The turning maneuver with extensive driving operations plays a crucial role in the automation of agricultural robots. Challenges of autonomous turning control for agricultural robots are to design feasible navigation path and to implement adaptive control techniques. In pursuit of stable control performance, the control algorithms should be adaptive based on the change of vehicle speed and soil conditions. To simplify maneuvers at the headland and to increase the accuracy after turning, an adaptive control algorithm called dynamic circle-back (DCB) turning is presented (Wang and Noguchi, 2018). Derived from the CB turning method, the learning function based vehicle model is adopted in the DCB turning. It is especially suitable for the distance of the narrow path (less than two times of minimum turning radius) condition that needs complex turning maneuvers. The turning maneuvers are optimized by compressing the turning trajectory, shortening the time consumption, and diminishing the deviation after turning. The designed operations are accessible for the robot to perform. Also, considering the lock-to-lock time and the slide of the vehicle, the control algorithm records the position of the vehicle and calculates the turning radius in real time. By adjusting the weights of demands (high accuracy after turning, less time consumption, and less headland occupancy), the algorithm creates an optimal path according to the estimation of the forward turning radius. In addition, a reinforcement learning method is used to update the steering model after each turn. The performance of the algorithm and improvements in the algorithm compared with conventional schemes were evaluated in dynamic field tests.

5.2.1 Specification of the adaptive turning control algorithm

A flowchart of turning algorithm is shown in Fig. 5. 2. The input of the turning algorithm includes the position of the vehicle, the posture of the vehicle (yaw, pitch and roll angles), the speed, the steering angle, the lateral and the heading deviations from the navigation map, and limitations of speed, steering angle and headland occupancy.

By balancing the designer's demands (higher accuracy, less time consumption, and smaller headland occupancy), the objective function of the dynamic turning method creates an optimal pathway according to the vehicle model. The vehicle model indicates relations of the steering angle and the turning radius in different sides with forward or backward movement. The regression analysis is used to estimate the turning radius of the forward movement in real time. In addition, considering the movement during LTL time, delay response caused by the inertia of the vehicle, and the sideslip while turning, the objective function calculates the turning radius of forward movement and optimizes the pathway using the vehicle model in real time. The error of the vehicle model is represented as the difference between the estimated turning radius (from the vehicle model) and the calculated turning radius (from the regression analysis). The performance of turning control is evaluated by the error analysis mechanism, which considers the error of the model and the lateral deviation at the end of the backward movement (the Point C in Fig. 3. 15). After each headland turn, a reinforcement learning method is adopted in real time to tune the vehicle model by considering measured data as the training dataset.

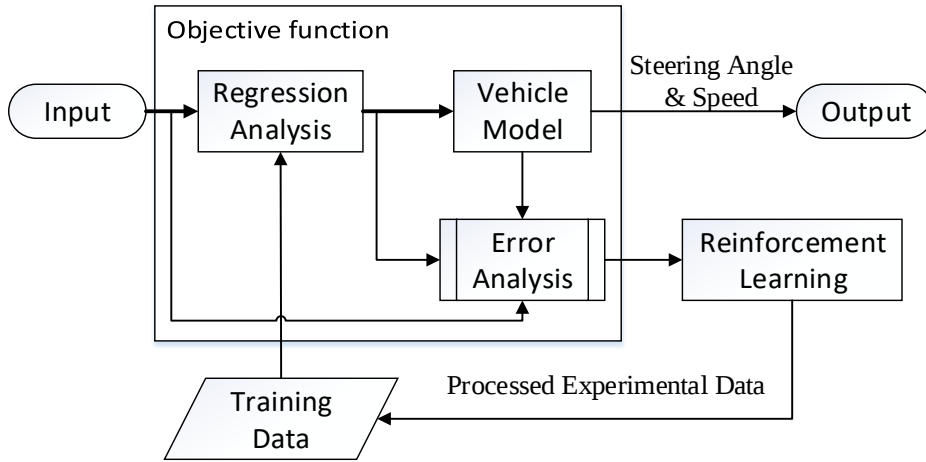


Fig. 5. 2. Flowchart of the DCB turning algorithm.

5.2.2 Objective function based on weights

To adjust the turning route according to different regulations, such as efficiency, speed, headland space, and so on, an objective function (J) is defined to optimize the time consumption (T) and the lateral deviation from navigation path (Δd) in:

$$J(T, \Delta d) = h \cdot H_0 + t \cdot T + \varepsilon \cdot \Delta d \quad (50)$$

$$s.t. \quad h + t + \varepsilon = 1$$

which takes the weights of headland distance (h), time (t), end-point accuracy (ε) requirements, and reference headland distance H_0 as inputs, and then designs the

navigation path and decides the steering angle (δ) and turning speed (v). During the turning (with forward and backward movement), the vehicle keeps a constant speed determined by:

$$v = V_{\min} + t(V_{\max} - V_{\min}) \quad (51)$$

where $V_{\min}$ and $V_{\max}$ are the minimum and maximum speed limitations of turning. In addition, if the maximum headland distance is larger than 8.0 m, the desired headland occupation will be calculated according to

$$H = (H_0 - 8.0) * (1 - h) + 8.0 \quad (52)$$

The values of weights are three hyper-parameters that need the user to adjust according to different requirements.

Considering vertical and horizontal disturbances applied to the soil at the headland, the turning starts at the end of the straight path without pause. The working flow of the DCB turning method used in this study is shown in Fig. 5. 3. First of all, the optimal trajectory angle will be calculated based on the default vehicle model and constraints. Fig. 5. 4 shows three typical examples of paths created by different combinations of weights in the objective function. If the headland distance is short, we have to compromise on end-point accuracy and time consumption. In this case, the trajectory A-B₁-C₁-D might be an optimal choice. If time is limited, the tractor should follow a path such as A-B₃-C₃-D with less trajectory and high turning speed. Under an ideal condition, point C₂ will be on the extension line of the next path, so the lateral deviation of the vehicle would be zero along the path of C₂-D. Forward turning angle is determined by:

$$\begin{aligned} \theta = 90 + \sin^{-1}((R_b + w - R_f)/(R_b + R_f)) * 180 * \pi + 10 \\ * (h - t) * (1 - \varepsilon) \end{aligned} \quad (53)$$

Once the optimal trajectory is determined, the vehicle executes the turning maneuver following the reference path. Since the vehicle turns without stopping before the CB turn, the steering wheel cannot immediately match the commanded angle. The delay of steering is caused by the inertia of the steering column, the servo system, and communication delays.

During the forward movement, the vehicle deviates from the reference path because of the wheel slippage and estimation error of the vehicle model (Eq. (31)). Therefore, the forward turning radius should be updated according to Eq. (5). According to Eq. (30), the backward turning radius should be renewed based on the updated forward turning radius. Moreover, the updated the reference navigation path by recalculating the forward turning angle by Eq. (53). During the backward movement, the backward turning radius will be calculated based on the actual trajectory. Both the calculated

forward turning radius and the backward turning radius during a CB turning will be added to the training data according to Eq.(34).

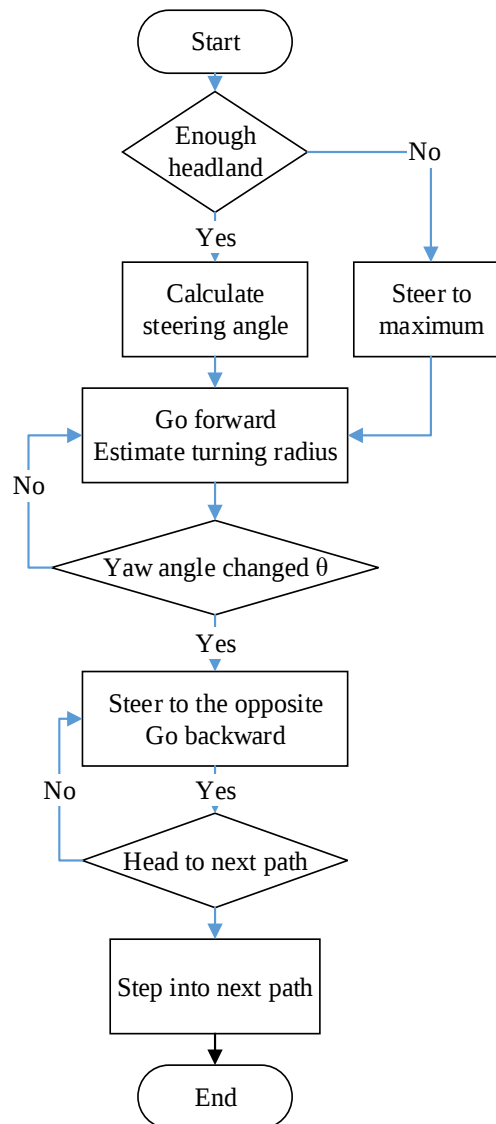


Fig. 5. 3. Working flow of the DCB turning.

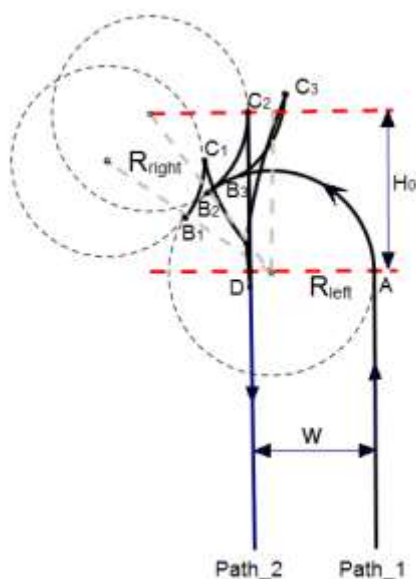


Fig. 5. 4. Planned paths for turning.

The deviation from the reference path caused by the wheel slippery and the low response of the steering column can be compensated by updating the turning radius. However, the vehicle cannot stop immediately at the desired position because of the delay response of the control unit and the inertia of the vehicle.

Fig. 5. 5 shows changes in vehicle speed and steering angle during the turn. The blue area indicates that the tractor goes forward during that period, and the orange area indicates the backward movement period. The distance of two red lines represents the LTL time, which was about 2 s. At 598.0 s the tractor should start to go backward, but because of the inertia of a massive body, the tractor did not stop until 599.4 s, which was 1.4 s later. The backward turning should be stopped at 605.4 s, but the tractor did not stop and go forward until 606.2 s, which was 0.8 s later. It means that the command of going backward or forward should be sent to ECU ahead of the desired vital points. The forward turning angle can be revised as follows:

$$\begin{aligned} \theta &= 90 + \sin^{-1}((R_b + w - R_f)/(R_b + R_f)) * 180 * \pi + 10 \\ &* (h - t) * (1 - \varepsilon) + \theta_{bias} \end{aligned} \quad (54)$$

$$\theta_{bias} = \sin^{-1}[\Delta d / (R_b + R_f)]$$

where Δd is the lateral deviation after backward turning (for example, the lateral deviation at point C₂ in Fig. 5. 4).

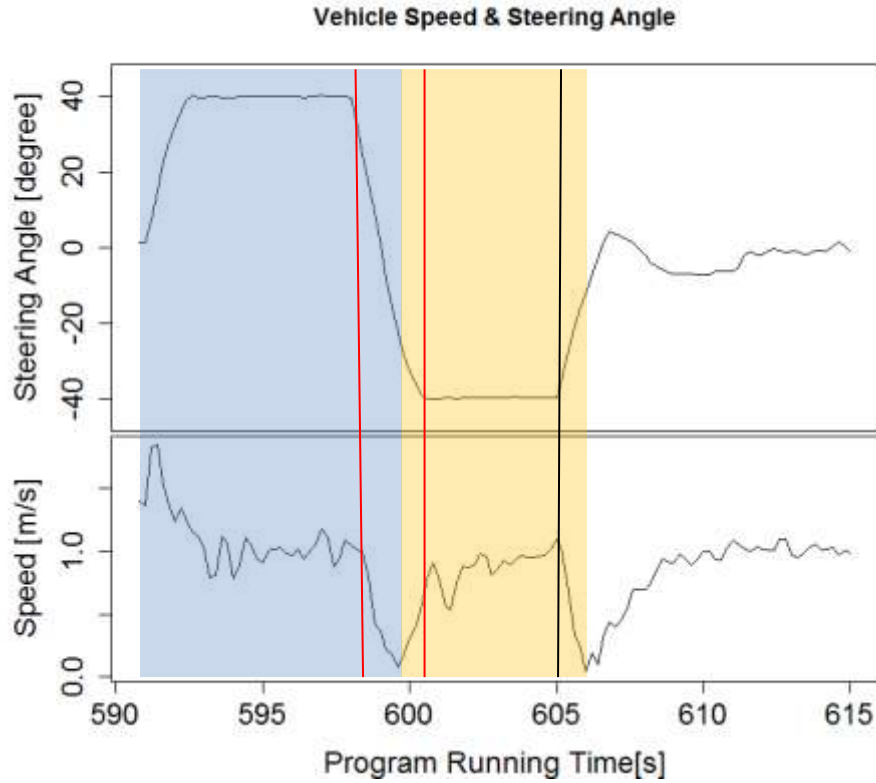


Fig. 5. 5. Steering angle and speed during turning.

In a nutshell, in the dynamic turning method, the tractor can adjust the steering angle autonomously according to the length of the headland and optimize the turning trajectories in real time. To clarify the maneuvers clearly, the pipeline of the DCB maneuvers is listed here:

The pipeline of the DCB maneuvers

Inputs:

- $weights[3] = \{t, \varepsilon, h\}$: weights of efficiency, accuracy and headland occupancy requirements
- r : coefficient matrix of turning radius model: Eq. (30)
- s : coefficient matrix of steering angle model: Eq. (31)
- H_0 : maximum headland distance
- w : distance between paths
- V_{min} : minimum speed for turning
- V_{max} : maximum speed for turning
- d : offset to the nearest working path
- $\Delta\Psi$: heading error to the nearest working path

Outputs:

- θ : forward turning angle
- $Steer$: steering angle

- v : speed for turning

Initialization:

calculate the speed v , steering angle $Steer$
 generate an initial trajectory for guidance according to Eq. (53)

while go forward **do**

calculate the turning radius R_f according to Eq. (5)
 update the reference path according to Eq. (53)

until heading angle changed θ

while going backward

calculate the turning radius R_b

until heading angle changed $(180 - \theta)$ or $\Delta\Psi == 0$ (Eq. (48))

reinforcement learning

Add R_f and R_b into dataset (Eq.(34))
 Update the vehicle model (Eq. (30) and Eq. (31))

while go forward **do**

path tracking following Eq. (47)

until approach into the next working path

estimate the bias of model according to Eq. (54)

5.3 Experiments and evaluation

In order to evaluate the performance of the developed robot tractor system, several tests were conducted at an experiment farm of Hokkaido University, Japan. Two robots were used to evaluate the performance of the navigation algorithm and the dynamic circle-back turning algorithm proposed in this research. Firstly, the EG105 without implement was used to test the accuracy of the system. Moreover, comparison tests were conducted to evaluate the improvement of DCB relative to a traditional turning algorithm, switch-back turning. Thirdly, the other wheel-type robot tractor, EG83, with implement was used to test the performance of the robot in different weights of maneuver function in Eq. (50). By changing the weights, the robot can fulfill different requirements of working. Last but not least, the robot tractor, EG105, performed the DCB turning algorithm in an irregular shape of the field where the distance of headland is limited.

5.3.1 Evaluation of navigation accuracy

In Hokkaido, the distance between two rows is 66 cm in general, and the width between two working paths, swath width, is usually set to be 2.64 m because in one working path the implement can seed or weed four rows simultaneously. In this case, a 10-path map was made for evaluating the accuracy of autonomous navigation algorithm. The trajectory of the robot tractor during the test is shown in the UTM coordinate system in Fig. 5. 6. There are ten paths from left to right of the test map and the distance between each path was set to be 2.64 m. The working order of paths was sequent, from Path 1 (near the arrow in Fig. 5. 6) to Path 10 with 9 times headland turns. The velocity of the robot was set to be 1.0 m/s for turning and 1.4 m/s for working. That is to say, weights of efficiency, accuracy, and headland distance are 0, 1 and 0, respectively. The maximum speed limit and the minimum speed limit were set as the same, 1.0 m/s. Also, the maximum headland distance H_0 was set as 7.0 m so that the tractor would steer to the maximum during the turning.

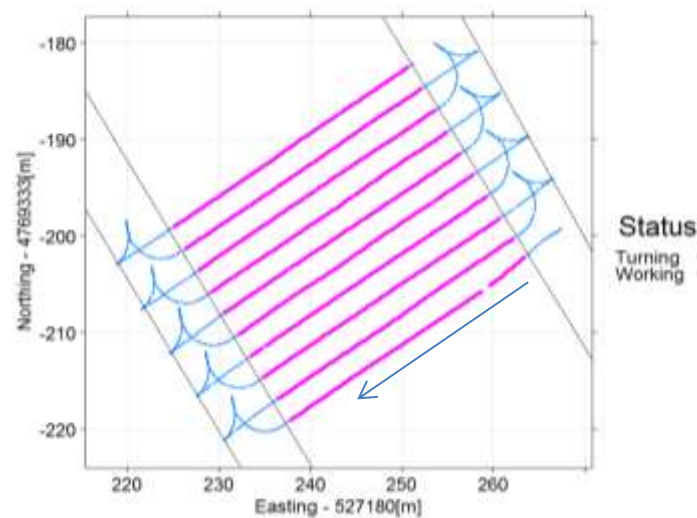


Fig. 5. 6. The trajectory of robot tractor EG105.

Take the second lane change as an example, red circles in Fig. 5. 7 show the forward trajectory, black dots are the backward trajectory, and blue blocks illustrate the straight path before the next path. The blue lines are the regression analysis of the trajectories. To express the length of the trajectory clearly, the origin of the coordinate in Fig. 5. 7 was shifted to (527180, 4769333). We found that two circles fitted the forward and backward turning trajectories well. That is to say, the robot followed the trajectories generated by DCB turning algorithm very well, including continuous paths before steering to the desired angle.

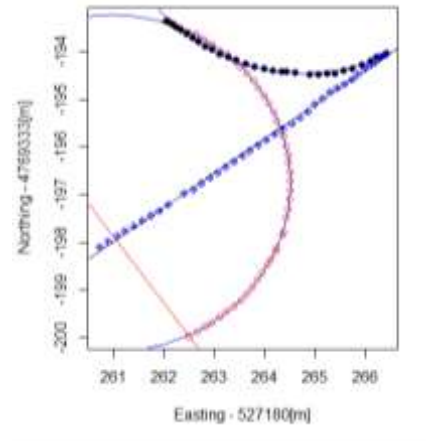


Fig. 5. 7. Lane change order 2.

Fig. 5. 8 shows the lateral error of each path. The minus value of lateral error means the robot on the left side of the reference path, and the positive value means the robot on the right side of the path. We can see that the maximum deviation (absolute value of the lateral error) in the path was less than 10 cm without outliers in the box plot. Mean values of the deviation in all paths were all less than 5 cm, and RMS of lateral deviation were all less than 6 cm, as shown in Table 5. 1. The maximum RMS of lateral deviation value was 5.71 cm, the minimum value was 3.12 cm, and the average of RMS errors was 4.10 cm. From Fig. 5. 9, we could see absolute values of heading errors were all less than three deg., which means the tractor moves smoothly without significant steering adjustment in the working lane. The maximum RMS of heading errors was 1.14 deg., and the average heading error was 0.98 deg. in Table 5. 1. From this result, we can conclude that the robot tractor can automatically run at 1.4 m/s within 6.0 cm and 1.2 deg. on a lateral error and heading error, respectively.

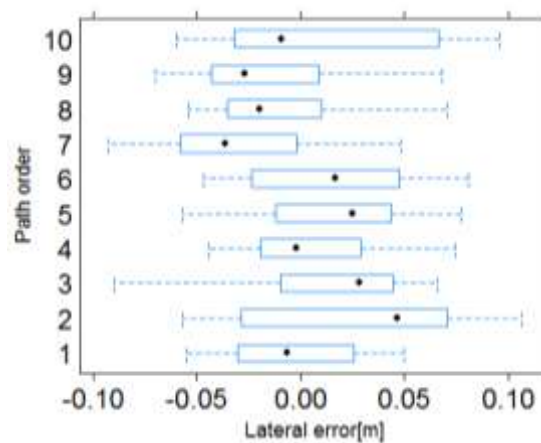


Fig. 5. 8. Lateral errors in each working path.

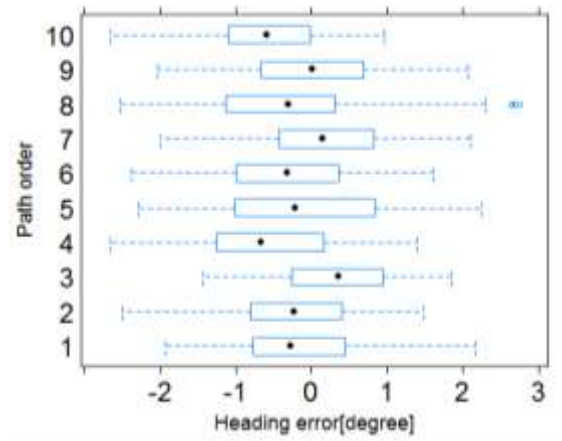


Fig. 5. 9. Heading errors in each working path.

Table 5. 1. Lateral deviation and heading error in each path.

RMS path	Lateral deviation [cm]	Heading error [deg.]
path1	3.12	0.93
path2	5.71	0.87
path3	4.37	0.84
path4	2.89	1.09
path5	3.95	1.10
path6	3.89	1.01
path7	4.61	0.86
path8	3.36	1.14
path9	3.89	0.95
path10	5.24	1.03
average	4.10	0.98

The efficiency and accuracy of each lane-change were analyzed from 4 aspects (in Table 5. 2), including the length of turning trajectory, time consumption, occupied headland distance, and the lateral deviation after turning. In Table 5. 2, the average trajectory distance was about 20.4 m and the time consumption was about 25 s. The average headland distance was about 6.6 m. The maximum later error was 7.1 cm, and the average deviation after turning (lateral deviation at point D in Fig. 3. 15) was about 4.0 cm. Therefore, the DCB turning method was proved to be stable enough in the 1.0 m/s velocity of turning.

Table 5. 2. Efficiency and accuracy of each lane change.

Order	1	2	3	4	5	6	7	8	9
Trajectory [m]	20.16	20.26	20.01	20.63	20.67	20.17	20.70	20.57	20.58
Time [s]	24.8	24.8	24.6	25.0	25.2	24.6	25.4	24.8	25.2
Headland [m]	6.48	6.60	6.42	6.71	6.69	6.59	6.73	6.65	6.62
Deviation [cm]	7.09	-6.59	2.59	2.37	1.02	-3.43	3.62	-2.82	6.44

To fully evaluate the performance of the robot system, the robot EG83 with implement behind the tractor was tested in a 10-path map. The trajectory of the robot tractor during the test is shown in Fig. 5. 10. There were ten paths from left to right of the test map, and the distance between each path was set to 2.5 m. The working order of paths was sequent, from Path 1 to Path 10 with 9 times lane change. The velocity of the robot was set to be 1.0 m/s during turning and 1.4 m/s during working. Moreover, the maximum headland distance H_0 was set as 7.0 m so that the tractor would steer to the maximum during the turning.

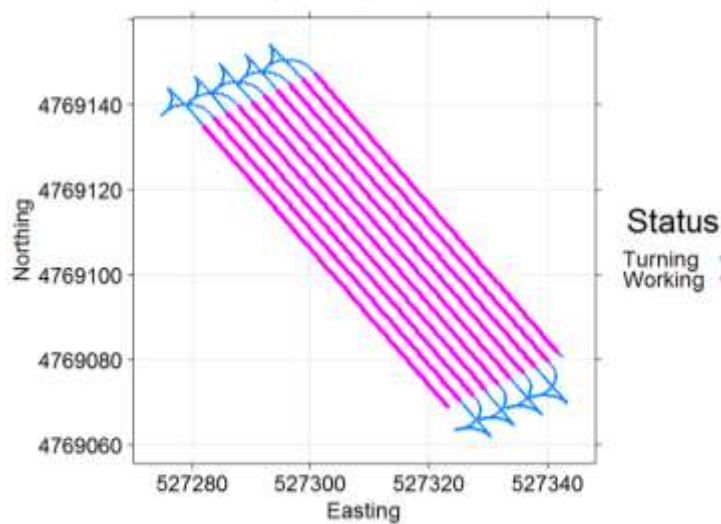


Fig. 5. 10. The trajectory of robot tractor EG83.

Fig. 5. 11 shows the lateral error of each path. We can see that the maximum deviation in the path was less than 7 cm with several outliers in the box plot. Mean values of the deviation in all paths were all less than 3 cm, and the average RMS of lateral errors was 2.5 cm, as shown in Table. 4.3. The maximum RMS of lateral errors was 2.78 cm; the minimum value was 2.16 cm. Absolute values of heading errors in Fig. 5. 12 were less than 3 degrees, which means the tractor moves smoothly without significant steering adjustment in the working lane. The maximum RMS of heading errors was 1.28 deg., and the average heading error was 1.12 deg., as shown in Table 5. 3. From this result, we can conclude that the robot tractor EG83 with implement can

automatically run at 1.4 m/s within 3.0 cm and 1.12 deg. on the lateral error and the heading error, respectively.

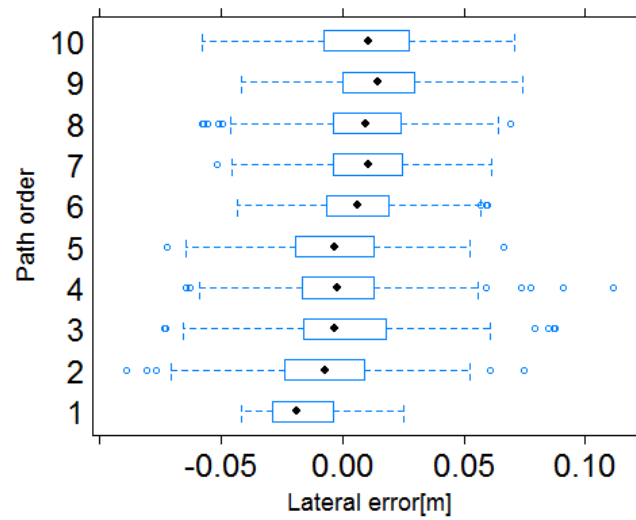


Fig. 5. 11. Lateral errors in each working path.

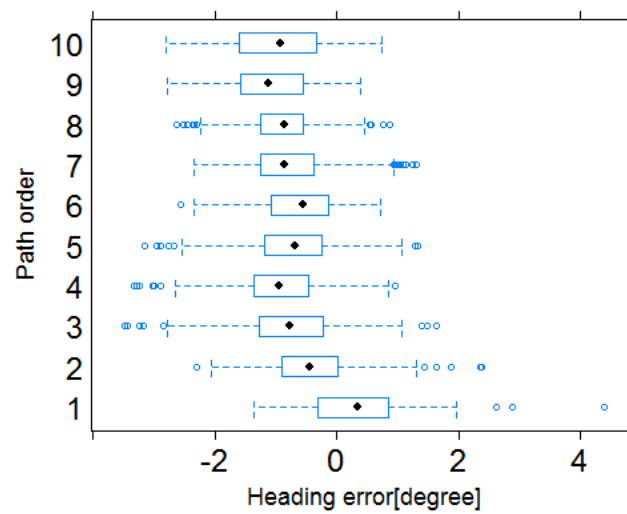


Fig. 5. 12. Heading errors in each working path.

Table 5. 3. Lateral deviation and heading error in each path.

RMS path	Lateral deviation [cm]	Heading error [deg.]
path1	2.26	1.25
path2	2.69	0.87
path3	2.78	1.19

path4	2.66	1.23
path5	2.21	1.08
path6	2.16	0.88
path7	2.42	1.05
path8	2.39	1.01
path9	2.65	1.28
path10	2.63	1.24
average	2.49	1.12

The efficiency and accuracy of each lane-change were analyzed from 4 aspects in Table 5. 4, including the length of turning trajectory, time consumption, occupied headland distance, and the lateral error after turning maneuver. As was shown in Table 5. 4, the average trajectory distance was about 21.4 m, and the time consumption was about 24.2 s. The average headland distance was about 7.0 m. The maximum later deviation was 7.42 cm, and the average deviation after turning was about 4.3 cm. Same as previous experiments based on the robot EG105, the DCB turning method was proved to be stable enough in the 1.0 m/s velocity of turning.

Table 5. 4. Efficiency and accuracy of each lane change.

Order	1	2	3	4	5	6	7	8	9
Trajectory [m]	21.05	21.31	20.81	21.31	21.84	22.06	21.57	21.15	21.58
Time [s]	23.8	24.2	23.4	24.2	24.8	25.0	24.4	24.2	24.4
Headland [m]	6.88	7.15	6.70	7.04	7.10	7.31	7.01	7.05	6.85
Deviation [cm]	3.94	-0.58	10.6	-1.17	6.09	3.35	3.81	-7.42	2.09

To fully evaluate the preformation of the DCB turning algorithm, the robot EG83 with rotary behind the tractor was tested in a 10-path map. The width between two paths was set to 2.5 m. The reference headland distance was set to 9.5 m. The minimum speed for turning was set at 0.7 m/s. Due to the oscillations caused by the uneven ground surface at the headland, the field test was conducted up to 1.2 m/s turning speed. Experiments were conducted six times with different combinations of weights in the maneuver function in Eq. (50). Results of the experiments were listed in Table 5. 5. The efficiency and accuracy of the robot tractor were analyzed from 5 aspects: occupied headland distance, length of the turning trajectory, time consumption, the average speed during turning, and lateral deviation of the end-point. When the ratio of h was set as 1, the robot would fully steer to one side and turn in low speed to diminish the headland occupation as well as the overall trajectory. However, the lateral deviation after turning was 13.79 cm, which was much larger than the others. In addition, the lateral deviation was less than 5 cm when the headland distance was larger than 8.02 m. It was also proved that when the robot could turn into the next path precisely at the speed of 1.2 m/s (when the ratio of t was 1).

Table 5. 5. Field tests in different weights combinations.

[h,t, ϵ]	[0,1,0]	[0,0,1]	[1,0,0]	[0.1,0.6,0.3]	[0.6,0.4,0]	[0.4,0.2,0.4]
Headland [m]	9.75	9.26	6.62	9.31	8.02	8.34
Trajectory [m]	28.7	27.4	20.4	27.4	24.2	25.0
Time [s]	30.7	33.8	25.1	32.4	27.1	30.7
Speed [m/s]	0.93	0.81	0.81	0.85	0.89	0.81
Deviation [cm]	3.9	2.3	13.8	4.1	5.1	2.6

The DCB turning algorithm can regulate the maneuvers of the robot at the headland according to the constraints of the field and manual settings. Taking the third turn of the 10-path map as an example, trajectories of the robot tractor are shown in Fig. 5. 13. To express the length of the trajectory clearly, the origin of the coordinate was shifted to (527180.0, 4769333.0) in UTM coordinate. The red line indicates the edge of the field, and the black line is the extension line of the next working path. First, the robot turned along the red circles and went backward along with the black dots. Then the robot tractor moved into the next path as indicated by blue blocks. The turning paths are different from each other because of the change of weights in the Eq. (50). For example, the robot tractor's trajectory is shown in Fig. 5. 13(a) when the ratio of t is set to 1. The robot steers to a particular steering angle to occupy the whole headland space and turns at the maximum speed. In Fig. 5. 13(c), the robot entirely steers to the left side and turns at a low speed so as to decrease the headland distance because the ratio of h is set to 1. In Table 5. 5, it is observed that the headland distance and the turning trajectory in the third column are smaller than values in other conditions. However, the average lateral deviation after turning is 13.8 cm, which is much larger than deviations in other conditions. Even though the robot tractor might deviate from the next path after the backward movement as shown in Fig. 5. 13(a) and Fig. 5. 13(c), the robot tractor can make use of the headland space and approach the next path smoothly. The weights should be adjusted according to the requirements of headland distance, time, and end-point accuracy. Accordingly, the robot tractor will turn to next working path along different trajectories similar to those in Fig. 5. 13(d)-(f).

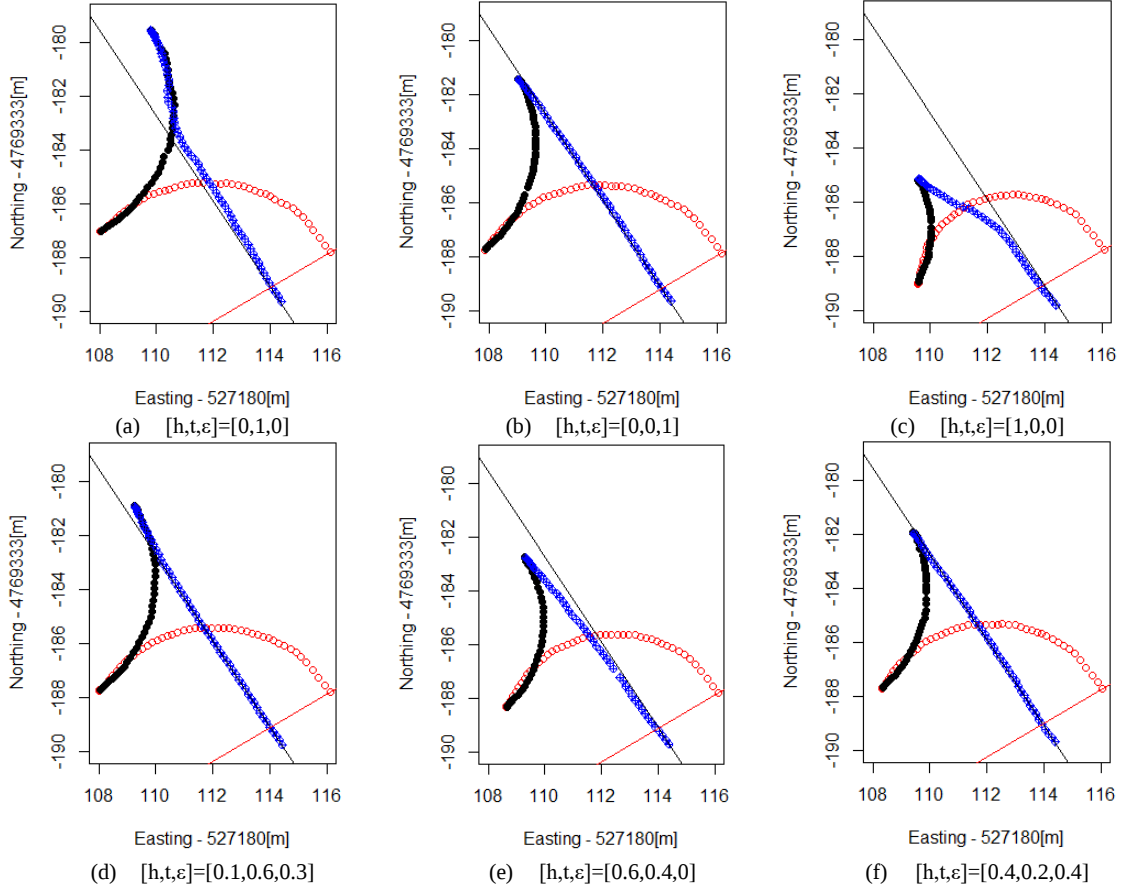


Fig. 5. 13. Trajectories of the robot tractor during the third turn.

Updating the vehicle model during the field test is a key factor for the robot tractor to follow the path accurately. When the weight of accuracy (ϵ) was set to 1, the robot tractor should ideally stop at the extension line of the next path (point C_2 in Fig. 5. 4) after the backward movement as shown in Fig. 5. 13 (b). However, deviation from the planned path is inevitable because of various factors such as the error of the initial vehicle model, sideslip of the robot tractor when turning, the time delay of the control unit. The procedure for updating the vehicle model in the DCB turning algorithm is shown in Fig. 5. 14. For the first turning, the vehicle steers to the right/left with the initial model. It can be observed that the vehicle model became stable after tuning parameters one time. Besides, gaps of parameters after the third update illustrate that the model is adaptive to the variations of external factors and converges again soon. The trajectory of the first turn is shown in Fig. 5. 15(a). For comparison, the trajectory of the third turn in Fig. 5. 13(b) is enlarged and is shown in Fig. 5. 15(b). It is found that the trajectory of the robot tractor coincided with the navigation path after one-time parameter tuning. Therefore, we can make a compromise solution by weighing the headland distance and requirements of efficiency and accuracy. Moreover, the steering angle during turning can be adjusted based on the desired headland distance.

It is meaningful for protecting the soil surface when the headland space is large enough for turning.

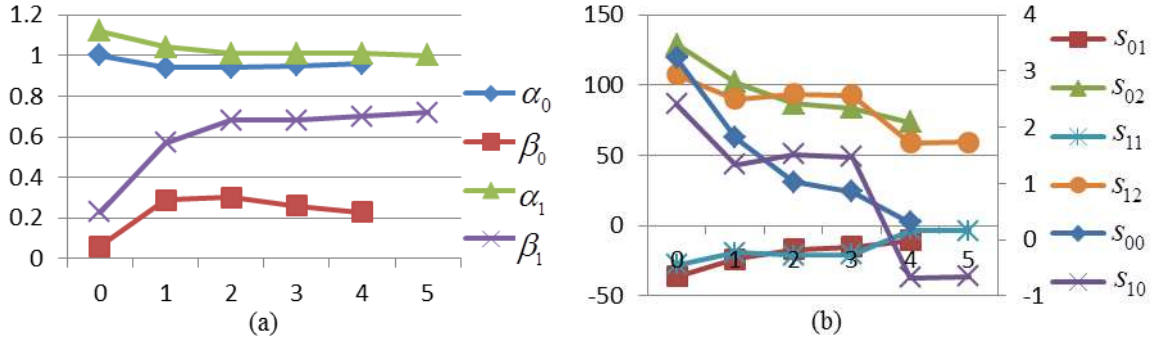


Fig. 5. 14. The update of vehicle model parameters. (a) shows the update of the vehicle model in the Eq. (32). (b) shows the update of the vehicle model in the Eq. (33). S_{00} and S_{10} are plotted on the secondary axis.

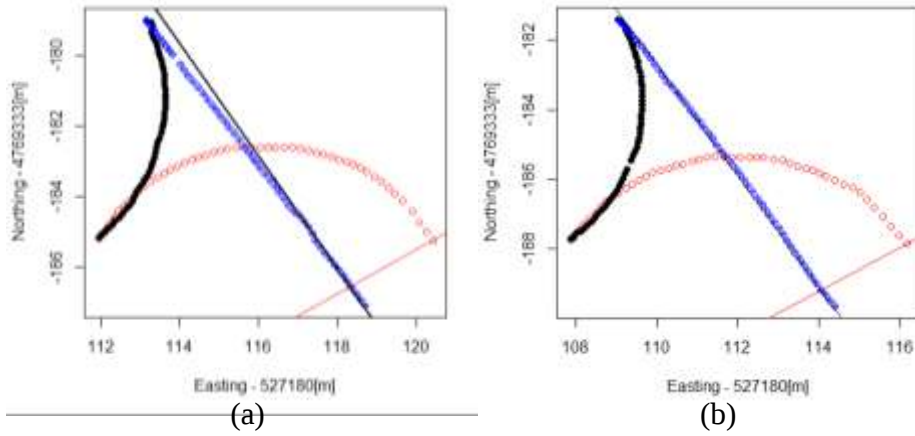


Fig. 5. 15. Trajectories of the robot tractor when turning. Weights of the objective function were set to $[h, t, \epsilon] = [0, 0, 1]$.

In a nutshell, it is proved that the steering angle during a turn could be adjusted on the basis of the desired headland distance. When the headland space is large enough for turning, the robot tractor turns to next path with a small steering angle, which is crucial for not damaging the soil surface. Besides, by calculating the radius of the trajectory during a turn, the algorithm could optimize the planned path in real time. With the learning mechanism, the DCB algorithm could analyze the accuracy of measured data after each turn, tune the vehicle model and make it applicable to different conditions, and guarantee robustness. One limitation of the DCB turning control algorithm is that it is not suitable for the tractor with a towed implement system, because the position of the implement (i.e., a trailer) is controlled by several actuators including the steering of the tractor and the hydraulically controlled joint. The DCB

turning algorithm works well for the tractor with a mounted implement, such as the rotary, the plow and the sprayer, which does not move horizontally relative to the tractor.

5.3.2 Comparison to the switch-back turning algorithm

The switch-back turning as a representation of traditional methods was compared with the DCB turning method in four maps with different path space, which are 2, 2.5, 3.0, and 3.5 m. The robot tractor in this experiment was EG105 without an implement behind the tractor. The speed of turning for the two methods was the same, 1.0 m/s. For DCB turning, the weights were set to $[0.5, 0, 0.5]$ in the tests, which meant h and ε were equally considered. Field experiments were conducted in maps with ten working paths for both the switch-back turning and DCB turning methods. The reference headland distance was set to 7.0 m, and the tractor therefore fully steered to the left/right at the end of the path. For the switch-back turning method, the forward movement before turning was 2.0 m to approach to the next lane with a small deviation. Fig. 5. 16 shows trajectories of the robot tractor with the different turning algorithms. The robot tractor started to turn at the red point in Fig. 5. 16 and steered entirely to the right in this example.

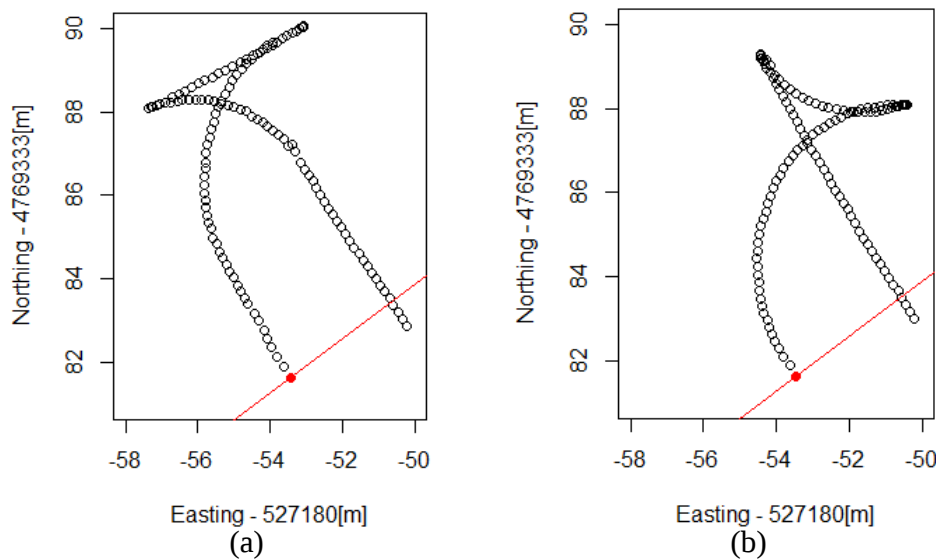


Fig. 5. 16. Turning trajectories of the robot tractor. Path space was 3.5 m.

As shown in Table 5. 6, four aspects were analyzed in the comparison experiments: length of turning trajectory, time consumption, occupied headland distance, and lateral deviation of the end-point. The maximum headland occupancy of DCB turning was no larger than that of switch-back turning. The trajectory of DCB turning was 21% less than that of switch-back turning when the width of two paths was 3.0 m.

Corresponding to the decrease of the trajectory, the time consumption of the DCB turning was 17% less than that of the switch-back turning algorithm. When the DCB turning algorithm was used, the improvement of accuracy after turning was about 79% when the path space was set to 2.0 m.

Table 5. 6. Comparison of dynamic circle-back turning and switch-back turning.

w [m]	2.0		2.5		3.0		3.5	
Method	DC B	Switch- back	DC B	Switch- back	DC B	Switch- back	DC B	Switch- back
Trajectory [m]	21. 7	25.5	20. 9	23.9	20. 0	25.3	19. 8	24.2
Time [s]	26	31	26	30	25	30	25	30
Headland [m]	7.4	7.4	6.9	7.0	6.5	7.3	6.2	7.4
Deviation [cm]	3.6	17.3	5.4	14.2	5.6	16.4	7.7	19.2

5.3.3 Turning at the irregular headland

Fig. 5. 17 illustrates a field with an edge that is not perpendicular to the path. The tractor should take more headland distance if the lane change algorithm is Switch-back turning or Keyhole turning. The h-turning method is modified from U-turning when a field is not a rectangle. Some other revised algorithms (Tu, 2013) were invited for the specific conditions of fields. Generally, solutions for the irregular shape of the field try to extend the trajectory at the beginning of or the end of the turning maneuver to fit the shape of the headland. On the contrary, the DCB turning algorithm makes use of the shape of fields and turns to the next path in a narrow headland distance without any modification to the algorithm. As is shown in Fig. 5. 17, the black quadrilateral is the shape of the field. There are six paths from left to right of the test map, and the distance between two paths is 2.5 m. The pink lines are the trajectories of the robot tractor during working inside the field, and the blue lines are the lane change trajectories.

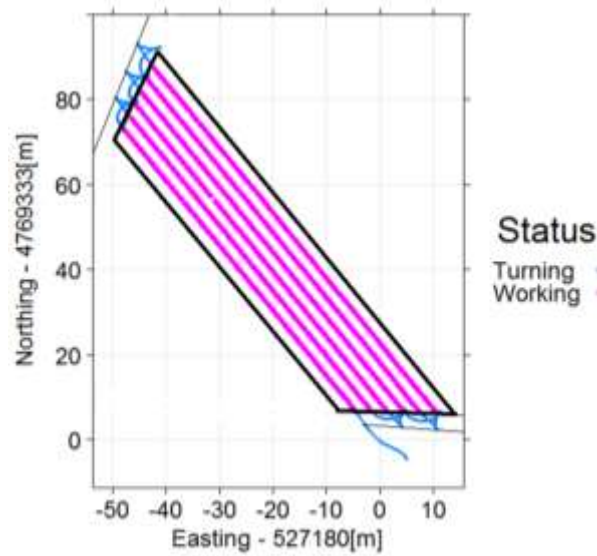


Fig. 5. 17. The trajectory in an irregular shape of the field.

Just for an explanation for the possibility and accuracy of DCB turning, the angle between the headland and a path was set as 51 deg. or so, and the distance between each path was 2.5 m in this experiment. Results about each turning are listed in Table 5. 7, and the turning order is from 1 to 5 in increasing order. It is evident that the trajectories and consumed time are less than those in Table 5. 2 because the distance of moving straight forward is shorter than the previous field caused by the irregular shape of the headland. Moreover, maximum headland distance is about 4.1 m, which is much smaller than 6.9 m (the headland distance in 2.5 m path space condition in Table 5. 6).

Table 5. 7. Results of 5 lane change maneuvers in Fig. 5. 17.

Order	Trajectory [m]	Time [s]	Headland [m]	Deviation [cm]
1	18.94	23.8	3.74	3.3
2	18.21	22.8	3.58	5.3
3	19.01	23.8	3.89	2.6
4	18.92	24.0	3.75	5.6
5	20.73	25.6	4.11	1.4

To illustrate the DCB turning algorithm in detail, lane change 2 was enlarged in Fig. 5. 18. The red dots are the forward trajectory of turning to the right, and the black ones show the backward trajectory of turning left, the blue ones show the straightly forward-moving trajectory. Fig. 5. 19 shows lateral errors while going forward along the straight black line (the extension line of the next path) in Fig. 5. 18. At the end of the backward turning, the lateral deviation at point C (in Fig. 3. 15) is about 20 cm, and the lateral error decreased to 5 cm within 6 s or so.

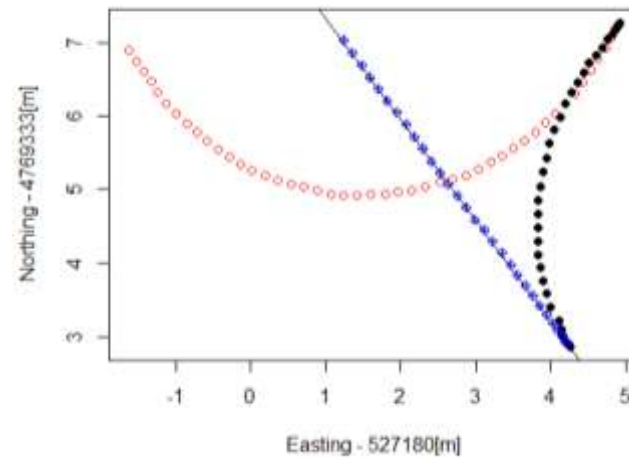


Fig. 5. 18. Turning trajectory from Path 2 to Path 3.

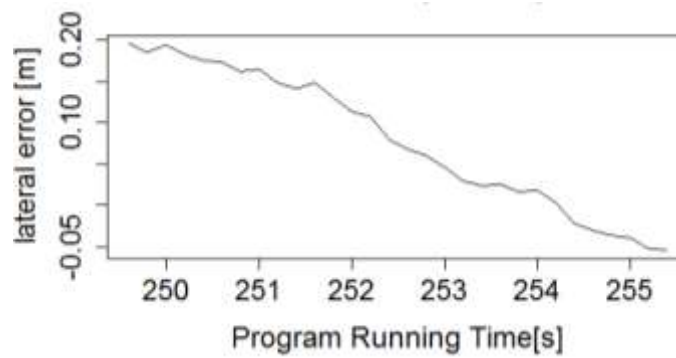


Fig. 5. 19. Lateral error in the straight path of turning.

In addition, Fig. 5. 20 shows changes in vehicle speed and steering angle during a turn. The speed during a turn was 1.0 m/s, and the speed in operation was about 1.4 m/s. The maximum steering angle was ± 40 deg. during the turn. The graphs also show that the robot tractor did not stop when the steering angle was changed. Although the robot tractor deviates from the optimal path when the steering angle is changed, the method for estimating the turning radius in real time can decrease the error caused by deviation. Moreover, the reinforcement learning method is used to update the vehicle model and to regulate the turning angle of the basis of the experimental data. Therefore, the robot tractor could turn into the next path with the precision of 3.9 cm at a speed of 1.2 m/s when the ratio of t was 1. In all, the robot can also approach into next path in a relatively small lateral error (the maximum was 5.6 cm in Table 5. 7) even though the headland distance was limited (less than 5 m in this case). Moreover, the cost of oil and time can be saved because of the decrease in trajectory.

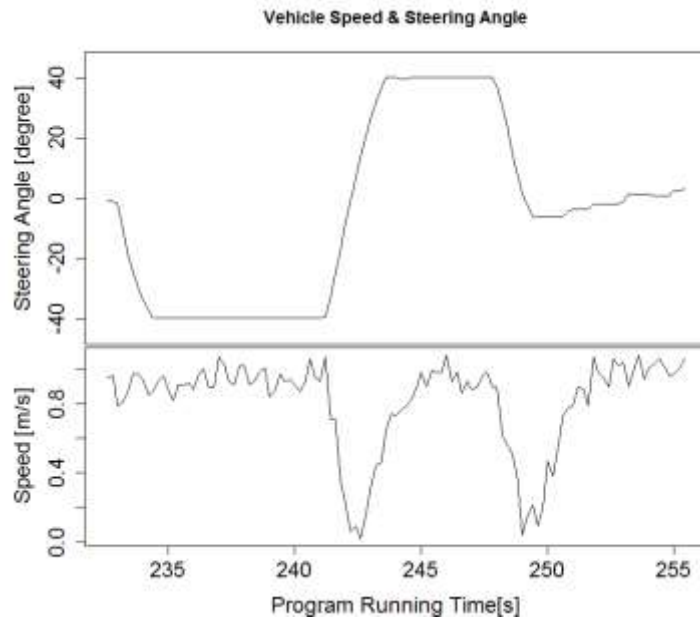


Fig. 5. 20. Speed and Steering angle during turning. Minus value of steering angle means the tractor steers to the left side.

Chapter 6 Remote monitor and control system

Smart farming is a management style to help farmers effectively manage their farms. A farm management system (FMS) can plan, monitor, and control agricultural processes. As a central part of the smart agriculture system, a robot tractor and a remote control system are developed in this research. A field management system is developed for the administrator to control in remote and to monitor the robot tractor working at the field. Farmers can also access the server to check the work record of the tractor, as well as editing the working plan through a smartphone or a PC. This chapter presents a part of functional architecture and provides an operational example of the management system.

Precision farming systems with the automation of agricultural machinery are considered to play an essential role in improving productivity and quality of farming activities. During past years, sophisticated farm management systems have been developed to plan, monitor, and control agricultural processes over the Internet and wireless sensor network (Kaloxylos et al., 2012; Othman and Shazali, 2012). Specifically, systems that can collect and process the information (such as, temperature, humidity, soil moisture, luminosity measurements, etc.) and assist farmers managing the farms are called farm management information systems (FMISs) (Kaloxylos et al., 2014). However, few precise farming systems implement the function of monitoring and controlling an agricultural vehicle remotely. Also, there is still no standardized system to enable cohesive interoperability among functionalities. This research is not to construct a comprehensive FMS/FMIS but rather to focus on functions related to monitoring and controlling autonomous agricultural vehicles.

6.1 System construction

The flow chart of the navigation algorithms in this study is shown in Fig. 6. 1. The administrator makes the working plan and assigns the work to a robot tractor through the management system. The working plan file records the farm ID and the working sequence. The robot tractor is designed to follow the working plan and to work in each field map. The navigation map is consist of point-to-point trajectories, working trajectories inside the field and transfer trajectories connecting one field to another. The control algorithm calculates the deviation of the vehicle from the desired map, which is expressed in lateral error and heading error. Moreover, the controller adjusts

the speed of the tractor and the steering angle. Besides, the administrator can pause the tractor in an emergency through the remote control system, the priority of which is higher than the control algorithm. The highest priority is the obstacle detection mechanism, which can adjust the speed and pause the work when an obstacle is detected. The vehicle will slow down to pass the obstacle at the side of the tractor. It will stop immediately once the obstacle in front of the tractor is detected. In addition, the warning message will be sent to the administrator. If the human or the animal is detected, the horn of the tractor will alarm and warn the obstacle to leave.

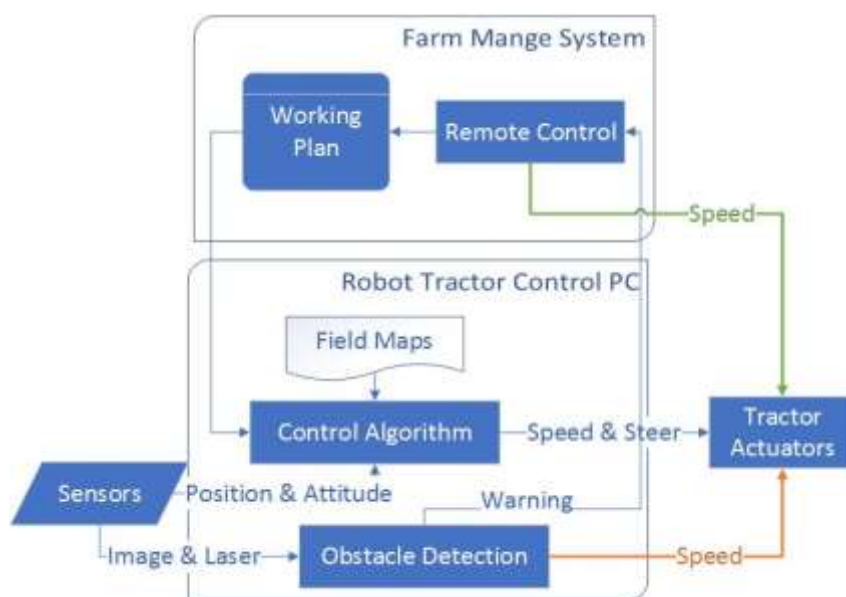


Fig. 6. 1. The block diagram of the navigation system.

The monitor and control of a robot vehicle are essential functionalities of the precise agriculture system. The data flow and communication strategy are illustrated in Fig. 6. 2. The commands from the remote administrator are distributed to the control PC of the robot tractor as well as the monitor near the farm. Commands and information are formatted into Extensible Markup Language (XML) information set and are transmitted relying on Hypertext Transfer Protocol (HTTP). Since HTTP protocols are installed and running on all operating systems, the remote farm management system at the server side allows clients to invoke web services and receive responses independent of programming language and platforms, such as a tablet, smartphone, and so on. Also, the administrator can monitor the surroundings of the robot tractor by four video cameras mounted on the cabin. Users can create the working plan and check the working record of the tractor through the web service. Since the submission needs to be confirmed by the administrator, the standard users cannot control the robot tractor in real time. The monitor beside the farm receives commands from the administrator and control the robot tractor through 2.4 GHz Long Term Evolution (LTE)

mobile router. In addition, the monitor can pause the robot tractor and change the working status of implement through a radio with 150 MHz width waveband. The working condition of the tractor will be uploaded to the administrator and the monitor simultaneously.

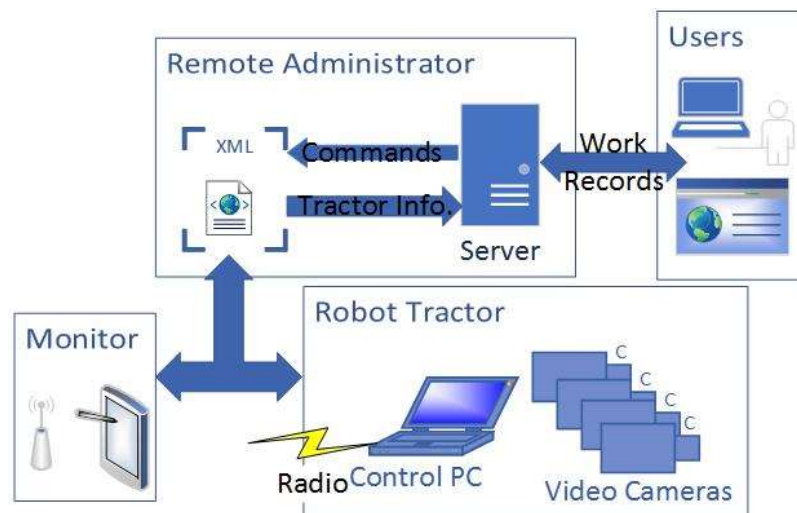


Fig. 6. 2. Working flow of tractor control and monitor.

6.2 Software development for remote control

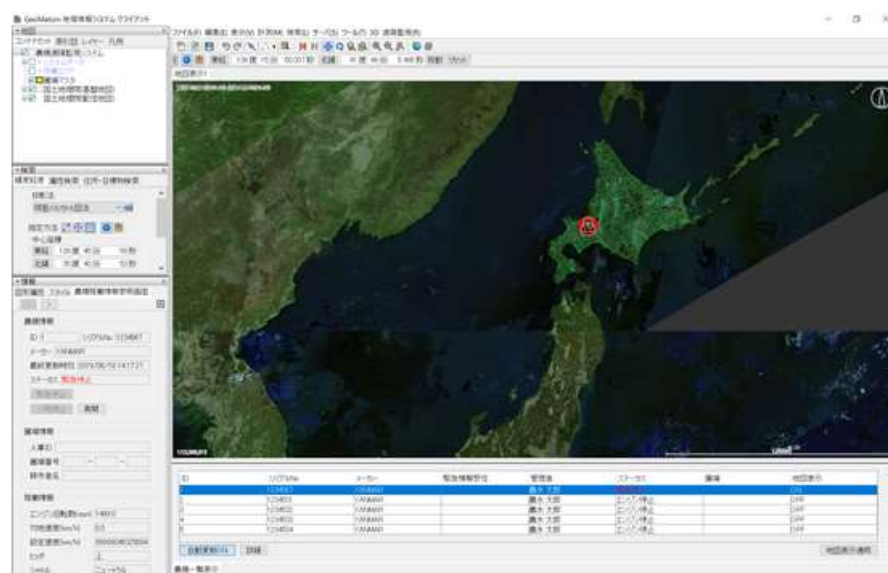
An FMS software is developed for the administrator to create the working plan and to control the robot tractor remotely. Cooperating with a Japanese company, Hitachi Solutions, Ltd., the prototype of FMS software is developed in the Windows7 operation system. It is termed Geomation. The language of the software is in Japanese.

The screenshot of the user interface (UI) is shown in Fig. 6. 3. The panel labeled “1” controls the information showing in the interface in section 4 and section 5. The information includes the working area, area of farmland, source of map, robots ID, and so on. In Fig. 6. 3, five robot tractors are working at the same time in a field. Their ID, serial number, maker, operator ID, working status, and so forth, are shown in section 5. Each robot tractor accesses the system through the preassigned ID and the password. The number on the icon indicates the tractor’s ID. The icon’s color indicates the working status of the tractor. The blue indicates the robot moving in the non-working area (outside the farmland). The green indicates the robot working in the specified farm, and the farm number is shown in section 5 of Fig. 6. 3. The red color indicates that the robot is paused manually or has not started yet. The robot tractor will stop automatically once the safety sensors detect an obstacle. The color of the icon will not change to red in this case.

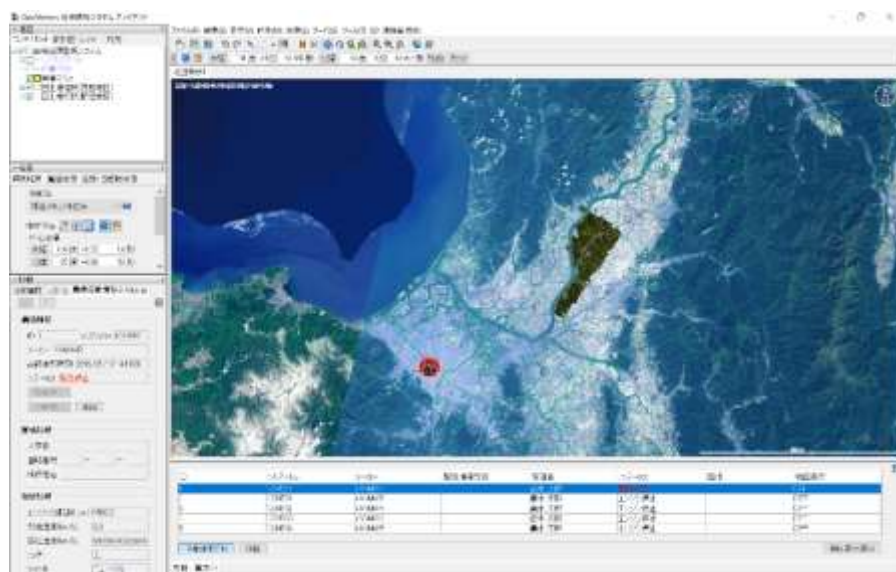


Fig. 6. 3. The user interface of the farm management system.

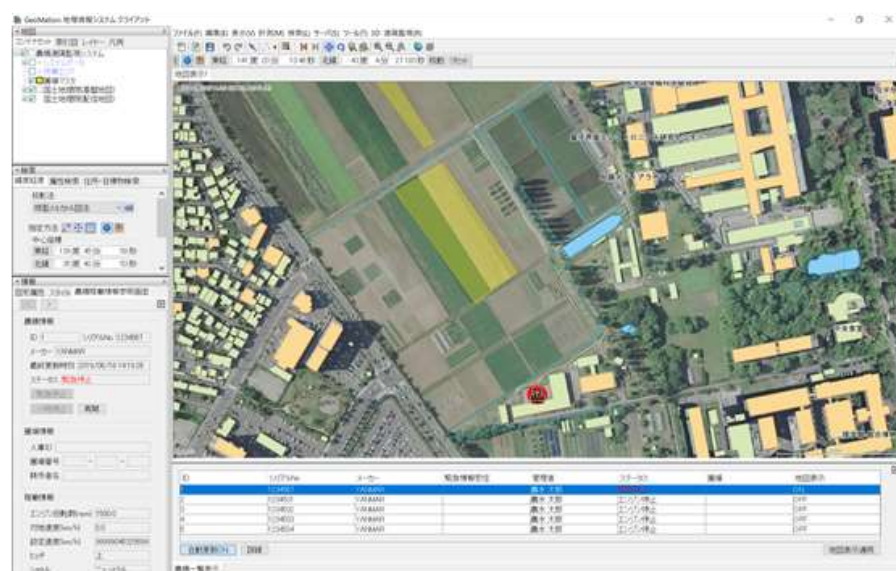
The map is an aerial photograph in section 4 of Fig. 6. 3. The Geospatial Information Authority of Japan provides the source of the map. Besides seamless aerial photographs, satellite images are also available for large scale, as shown in Fig. 6. 4. The size of the tractor's icon does not change while changing the scale of the map.



(a)



(b)



(c)

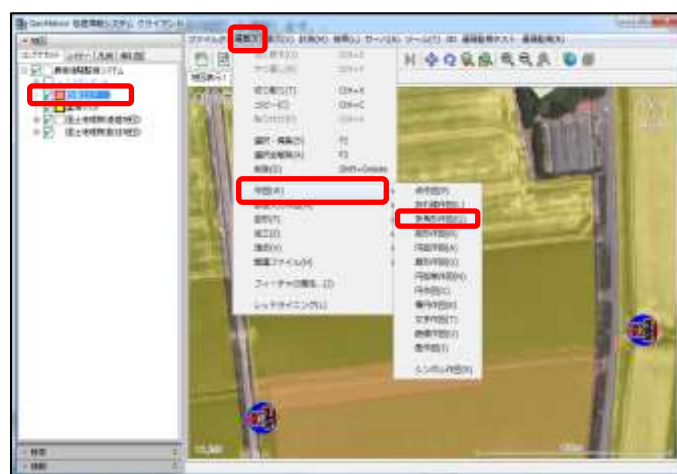
Fig. 6. 4. The map is showing on different scales. (a) and (b) are satellite images with different scales, and (c) is the seamless aerial photograph.

The dark area in the center of Fig. 6. 4(b) locates at the north of Iwamizawa City of Hokkaido, Japan. All the farmland under management is represented as many yellow areas showing in Fig. 6. 5 after zooming in. The shape of the yellow area is restricted to the boundary of each field. The user can zoom into a field by indicating its latitude and longitude position in section 2 of Fig. 6. 3.

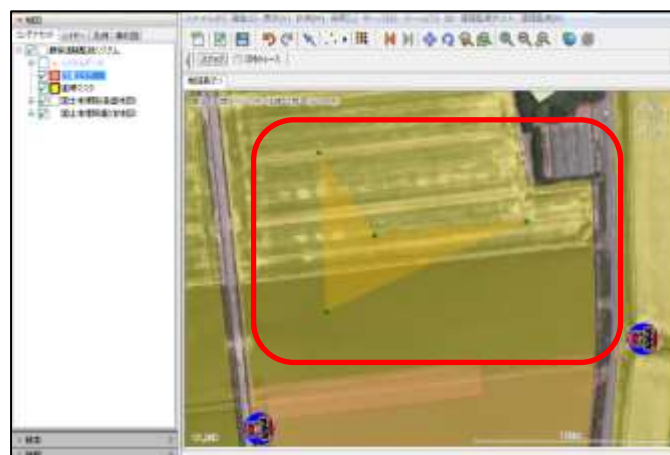


Fig. 6. 5. Farmland areas are showing in the satellite image.

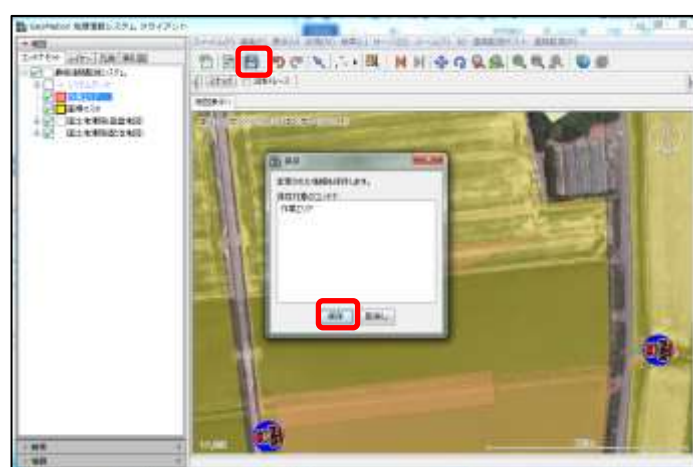
The working area includes the field area and area for transferring. The shapes of the field and working area are arbitrary. They can be imported from a user-defined boundary file. Fig. 6. 6 illustrates the other way of creating a new working area. The shape of the working area is created as a polygon by inserting vertexes.



(a)



(b)



(c)

Fig. 6. 6. The procedure for creating a new working area. The user can define the working area as a rectangular, polygon, circle, ellipse, or any combination of lines and curves. (b) shows a polygon working area (dark yellow area) inside a farm area (light yellow area). The created working area can be saved in the database, as shown in (c).

The communication data between the server and each robot is shown in Fig. 6. 7. The working plan file includes the directory map file, working path number and order, equipment size, and so on. The working status, including the process of farming work and the positioning information, should be uploaded to the server from the tractor. In addition, the status of the tractor includes velocity, engine speed, the status of PTO, and the position of the hitch. The user can check the details of each robot's working status and can also send the command through the buttons in section 3 of Fig. 6. 3.

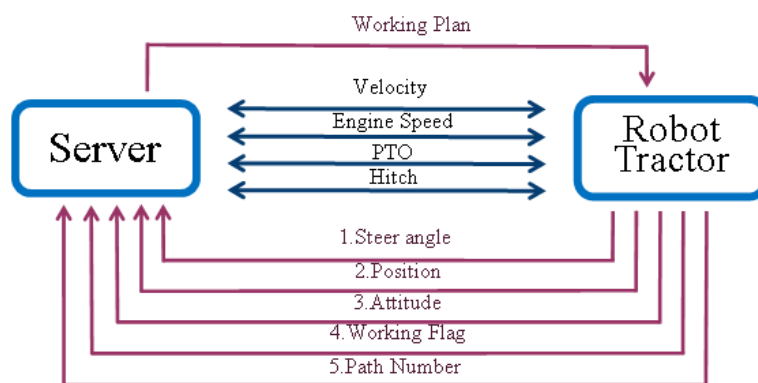


Fig. 6. 7. Parameters for remote monitor and control.

The client software for each robot tractor is shown in Fig. 6. 8. All the parameters for robot monitoring and controller in Fig. 6. 7 are showing in the panel (Fig. 6. 8(a)). The definition and format of these variables are listed in Table 6. 1. As shown in Fig. 6. 2, two ways are available for remote control. The user can control the robot through the radio. The communication protocol is an RS232 serial, as shown in Fig. 6. 8(b). The wireless radio is limited in the data transferring capacity. Therefore, the position and orientation data are only available for the administrator using Geomation. Both the monitor near the farmland and the remote administrator can control the working status of a robot tractor, such as pausing/resuming a robot, adjusting the speed, and so on.

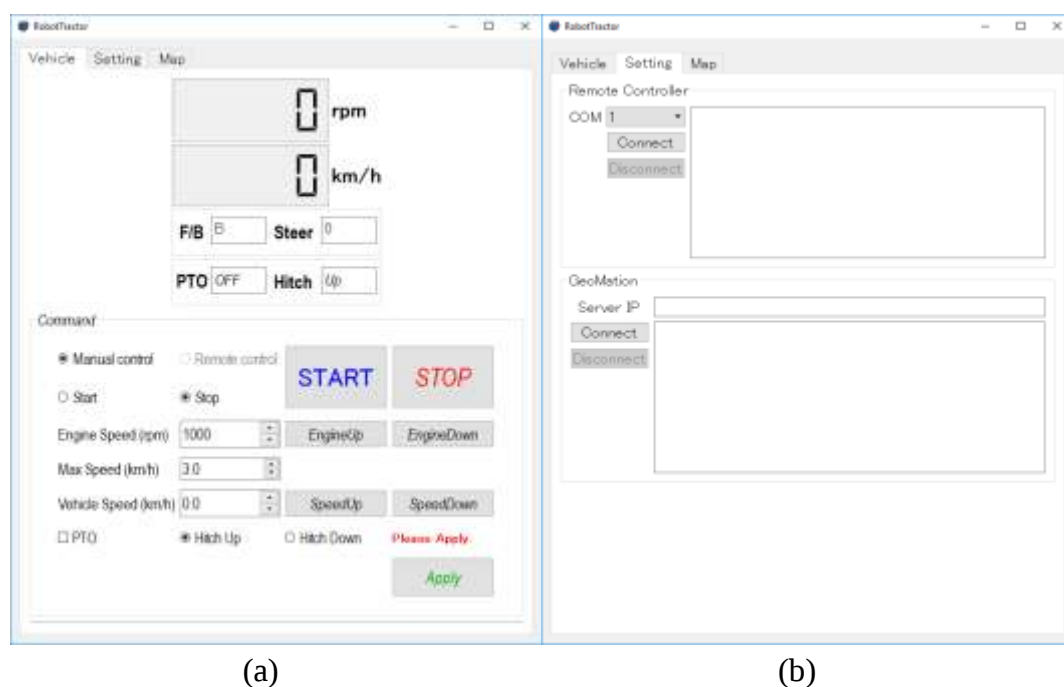


Fig. 6. 8. The software for controlling a robot tractor. The “vehicle” tab shown in (a) includes the vehicle status and the control command. This robot can connect to the

FMS software by entering the server's IP address in (b). The user can change the parameters and control the tractor.

Table 6. 1. Variables for remote monitor and control and their properties.

Variables	No. of values	Type	Unit
Steering angle	1	double	deg.
Position (longitude, latitude, altitude, north, east)	5	double	deg./m
Attitude (roll, pitch, and yaw)	3	double	deg.
Working flag	1	int	/
Path number	1	int	/
Velocity	1	double	m/s
Engine speed	1	int	rev/min
PTO	1	bool	/
Hitch	1	bool	/

Chapter 7 Research summary

In this study, autonomous robot tractors were developed by GPS and IMU as navigation sensors. Based on map navigation method, the robot tractor could conduct fieldwork, such as tillage, fertilizer, spraying, and so on. Research platforms include wheel type tractors (EG83 and EG105) and half-crawler type tractors (EG105-C and EG453), commercial products of YANMAR Company. Each tractor was equipped with a controller PC, the TECU communicated with the controller PC through CAN-BUS. GPS and IMU data were received via RS232 and processed by the controller PC. Fusing the position information and posture information, proposed steering controller calculates a steering angle so that the robot tractor can perform field work along with the reference paths.

1. Navigation of a robot tractor using the Quasi-Zenith Satellite System

This study evaluates the CLAS of the QZSS for controlling a robot tractor. Besides accessing the augmentation information through the L6 signal using a commercial QZSS receiver, this research also introduces a method for using CLAS with a dual-frequency receiver that cannot receive the L6 signal. Stationary and dynamic positioning experiments prove that the QZSS is able to improve the accuracy and availability of the current GNSS. The usage of Precise Point Positioning (PPP) in autonomous navigation is restricted in ambiguity resolution. Experiments using a commercial QZSS receiver reveal that the PPP converges to different results at the same position, and the results drift after convergence. A bias identification method based on landmarks was proposed to overcome these limitations. By compensating for the biases of the CLAS positioning results relative to the current GNSS, a robot tractor works along with GNSS-based navigation within 5 cm accuracy at a speed of 3.6 km/h.

2. Mapping and path planning

An optimal coverage path planning method is presented to improve field efficiency; and in particular, to fully utilize the advantages provided by automatically guided farming equipment. In addition, several transfer paths are created to optimize the non-working distance and time consumption. Refilling and unloading the tank are not considered in this autonomous path planning. The sequence of the working order is also designated. When the field is divided into several sub-fields, each sub-area has to be visited once without discard. To use the merit of agricultural robots, the backward movement along the navigation path is proposed in this research.

3. Vehicle system identification and state estimation

Vehicles are mostly assumed to comply with specific motion models which describe their dynamic behaviors. This research introduces three vehicle models that are widely used for control. Besides, data-driven system identification and state estimation methods are presented to increase the accuracy of control algorithms. Combining the estimation of the vehicle model and sensor measurements, the vehicle state can be estimated accurately. The control algorithm and solutions have been transferred to other industries, such as the automation and driving assist system for high way snow sweeper (G. O. J. Cabinet Office, 2017).

Finally, an adaptive turning algorithm for a four-wheel robot tractor in the headland is presented in this research. The algorithm could create an efficient path, with connected primitives, and optimize maneuvers during a turn. The vehicle model in DCB turning is represented as the relation of turning radii in different directions and the relation of turning radius and steering angle in the forward movement. The trajectory pattern and the dynamic vehicle model in the DCB turning algorithm make it more accurate than the traditional switch-back turning method in the same headland distance. Under the control of the DCB turning algorithm, the experimental robot tractor could turn to the next working path with or without a rotary at a speed of 1.2 m/s. The average lateral deviation after turning was about 3.9 cm, which was a significant improvement compared with the switch-back turning method. Moreover, it brings less headland occupation and time consumption as well. Therefore, it can be concluded that the adaptive turning algorithm is accurate and robust enough for controlling robot tractors.

One limitation of the turning maneuvers presented in this thesis is that they are not suitable for a vehicle with a towed implement system because the position of the implement (i.e., a trailer) is controlled by several actuators including the steering of the tractor and the hydraulically controlled joint. However, this turning algorithm works well for the tractor with a mounted implement, such as the rotary, the plow, and the sprayer, which does not move horizontally relative to the tractor.

4. A Farm Management System

A field management system is developed for the administrator to control in remote and to monitor the robot tractor working at the field. Farmers can also access the server to check the work record of the tractor, as well as editing the working plan through a smartphone or a PC. This research presents a part of functional architecture and provides an operational example of the management system.

REFERENCES

- Albagul, A., Wahyudi, 2004. Dynamic Modelling and Adaptive Traction Control for Mobile Robots. *Int. J. Adv. Robot. Syst.* 1, 149–154. <https://doi.org/10.5772/4638>
- Backman, J., Piirainen, P., Oksanen, T., 2015. Smooth turning path generation for agricultural vehicles in headlands. *Biosyst. Eng.* 139, 76–86. <https://doi.org/10.1016/j.biosystemseng.2015.08.005>
- Bevly, D.M., Gerdes, J.C., Parkinson, B.W., 2002. A New Yaw Dynamic Model for Improved High Speed Control of a Farm Tractor. <https://doi.org/10.1115/1.1515329>
- Billingsley, J., Schoenfisch, M., 1997. The successful development of a vision guidance system for agriculture. *Comput. Electron. Agric.* 16, 147–163. [https://doi.org/10.1016/S0168-1699\(96\)00034-8](https://doi.org/10.1016/S0168-1699(96)00034-8)
- Blischke, W.R., Halpin, A.H., 1966. Asymptotic Properties of Some Estimators of Quantiles of Circular Error. *J. Am. Stat. Assoc.* 61, 618. <https://doi.org/10.2307/2282775>
- Bochtis, D.D., Sørensen, C.G., 2009. The vehicle routing problem in field logistics part I. *Biosyst. Eng.* 104, 447–457. <https://doi.org/10.1016/j.biosystemseng.2009.09.003>
- Brunton, S.L., Kutz, J.N., 2019. Data-driven science and engineering: machine learning, dynamical systems, and control, 1st ed. Cambridge University Press.
- Burger, M., Huiskamp, M., Keviczky, T., 2013. Complete Field Coverage as a Multi-Vehicle Routing Problem. *IFAC Proc. Vol.* 46, 97–102. <https://doi.org/10.3182/20130828-2-SF-3019.00050>
- Cabinet Office, 2017a. What is the Quasi-Zenith Satellite System (QZSS)? [WWW Document]. Quasi-Zenith Satell. Syst. URL http://qzss.go.jp/en/overview/services/sv02_why.html (accessed 11.7.17).
- Cabinet Office, 2017b. Quasi-Zenith Satellite System Service Level Information for Centimeter Level Augmentation Service for Experiment (SLI-CLAS-E-001).
- Cabinet Office, G.O.J., 2017. NEXCO東日本、みちびきを活用した除雪車運転支援システムを試行導入 | ニュース/アーカイブ | みちびき (準天頂衛星システム: QZSS) 公式サイト - 内閣府 [WWW Document]. URL http://qzss.go.jp/news/archive/nexco_171107.html (accessed 6.3.19).
- Castleden, N., Hu, G.R., Featherstone, W.E., Abbey, D. a., Earls, C.J., Weihing, D., Øvstedal, O., 2004. First results from Virtual Reference Station (VRS) and Precise Point Positioning (PPP) GPS research at the Western Australian Centre for

- Geodesy. J. Glob. Position. Syst. 3, 79–84. <https://doi.org/10.5081/jgps.3.1.79>
- Choy, S., Harima, K., Li, Y., Choudhury, M., Rizos, C., Wakabayashi, Y., Kogure, S., 2015. GPS Precise Point Positioning with the Japanese Quasi-Zenith Satellite System LEX Augmentation Corrections. J. Navig. 68, 769–783. <https://doi.org/10.1017/s0373463314000915>
- Choy, S., Harima, K., Li, Y., Choudhury, M., Rizos, C., Wakabayashi, Y., Kogure, S., 2014. High accuracy real-time precise point positioning using the Japanese Quasi-Zenith satellite system LEX signal. CEUR Workshop Proc. 1307.
- Choy, S., Sunil Bisnath, B., Chris Rizos, B., 2017. Uncovering common misconceptions in GNSS Precise Point Positioning and its future prospect. GPS Solut. 21, 13–22. <https://doi.org/10.1007/s10291-016-0545-x>
- Collier, P., Harima, K., Choy, S., Kakimoto, H., Kogure, S., 2015. Utilisation of the Japanese Quasi-Zenith Satellite System (QZSS) Augmentation System for Precision Farming in Australia, IGNSS, Gold Coast, Australia.
- Council for Science and Technology Policy, 2017. Brouchure SIP “Pioneering the Future: Japanese Science, Technology and Innovation 2017” [WWW Document]. Cabinet Off. URL https://www8.cao.go.jp/cstp/panhu/sip_english/sip_en.html (accessed 6.7.19).
- De Bruin, S., Janssen, H., Klompe, A., Klompe, L., Lerink, P., Vanmeulebrouk, B., 2010. GAOS: Spatial optimisation of crop and nature within agricultural fields, in: The International Conference on Agricultural Engineering, Towards Environmental Technologies (AgEng2010). Clermont-Ferrand, France, p. 8.
- De Carvalho, R.N., Vidal, H.A., Vieira, P., Ribeiro, M.I., 1997. Complete coverage path planning and guidance for cleaning robots, in: ISIE '97 Proceeding of the IEEE International Symposium on Industrial Electronics. IEEE, Guimaraes, Portugal, Portugal, pp. 677–682. <https://doi.org/10.1109/ISIE.1997.649051>
- Dubins, L.E., 1957. On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. Am. J. Math. 79, 497–516.
- FAO, 2015. FAO and Post 2015.
- Fotopoulos, G., Cannon, M.E., 2001. An Overview of Multi-Reference Station Methods for Cm-Level Positioning. GPS Solut. 4, 1–10.
- Hameed, I.A., Bochtis, D., Sørensen, C.A., 2013. An optimized field coverage planning approach for navigation of agricultural robots in fields involving obstacle areas. Int. J. Adv. Robot. Syst. 10, 1–9. <https://doi.org/10.5772/56248>
- Hameed, I.A., Bochtis, D.D., Sørensen, C.G., Nøremark, M., 2010. Automated generation of guidance lines for operational field planning. Biosyst. Eng. 107,

- 294–306. <https://doi.org/10.1016/J.BIOSYSTEMSENG.2010.09.001>
- Hassaan, G.A., Mohammed, A.-A., 2015. Frequency Response of 10 Degrees of Freedom Full-Car Model For Ride Comfort. *Int. J. Sci. Res. Eng. Technol.* 4. <https://doi.org/10.1234/IJSRET.2015.120952>
- Hert, S., Tiwari, S., Lumelsky, V., 1996. A terrain-covering algorithm for an AUV. *Auton. Robots* 3, 91–119. <https://doi.org/10.1007/BF00141150>
- Hitachi Zosen Corporation, 2015. Utilizing a Self-steering Robotic Tractor in the Developmental Phases of Rice --Feasibility Study on Using Quasi-Zenith Satellite System for Precision Farming in Australia [WWW Document]. Hitachi Zosen Corp. URL <http://www.hitachizosen.co.jp/english/release/2015/01/001512.html> (accessed 1.5.17).
- Hoffmann, G.M., Tomlin, C.J., Montemerlo, M., Thrun, S., 2007. Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design, Experimental Validation and Racing, in: 2007 American Control Conference. IEEE, pp. 2296–2301. <https://doi.org/10.1109/ACC.2007.4282788>
- Hofner, C., Schmidt, G., 1995. Path planning and guidance techniques for an autonomous mobile cleaning robot. *Rob. Auton. Syst.* 14, 199–212. [https://doi.org/10.1016/0921-8890\(94\)00034-Y](https://doi.org/10.1016/0921-8890(94)00034-Y)
- Ishii, K., Terao, H., Noguchi, N., 1994. Studies on Self-learning Autonomous Vehicles (Part 1). *J. Japanese Soc. Agric. Mach.* 56, 53–60. https://doi.org/10.11357/jsam1937.56.4_53
- ISHII, K., TERAOKA, H., NOGUCHI, N., KISE, M., 1998. Studies on Self-learning Autonomous Vehicles (Part 4). *J. JAPANESE Soc. Agric. Mach.* 60, 53–58. https://doi.org/10.11357/jsam1937.60.2_53
- Jin, J., 2009. Optimal field coverage path planning on 2D and 3D surfaces. Iowa State University.
- Kaiser, E., Kutz, J.N., Brunton, S.L., 2018. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit. *Proc. R. Soc. A Math. Phys. Eng. Sci.* 474, 20180335. <https://doi.org/10.1098/rspa.2018.0335>
- Kaiser, Eurika, Kaiser, E., Kutz, J.N., Brunton, S.L., n.d. Sparse identification of nonlinear dynamics for model predictive control in the low-data limit.
- Kaizu, Y., Imou, K., 2008. A dual-spectral camera system for paddy rice seedling row detection. *Comput. Electron. Agric.* 63, 49–56. <https://doi.org/10.1016/J.COMPAG.2008.01.012>
- Kaloxyllos, A., Eigenmann, R., Teye, F., Politopoulou, Z., Wolfert, S., Shrank, C., Dillinger, M., Lampropoulou, I., Antoniou, E., Pesonen, L., Nicole, H., Thomas, F., Alonistioti, N., Kormentzas, G., 2012. Farm management systems and the

- Future Internet era. Comput. Electron. Agric. 89, 130–144.
<https://doi.org/10.1016/J.COMPAG.2012.09.002>
- Kaloxylos, A., Groumas, A., Sarris, V., Katsikas, L., Magdalinos, P., Antoniou, E., Politopoulou, Z., Wolfert, S., Brewster, C., Eigenmann, R., Maestre Terol, C., 2014. A cloud-based Farm Management System: Architecture and implementation. Comput. Electron. Agric. 100, 168–179.
<https://doi.org/10.1016/J.COMPAG.2013.11.014>
- Karkee, M., Steward, B.L., 2010. Study of the open and closed loop characteristics of a tractor and a single axle towed implement system. J. Terramechanics 47, 379–393. <https://doi.org/10.1016/j.jterra.2010.05.005>
- Kise, M., Noguchi, N., Ishii, K., Terao, H., 2002a. Field Mobile Robot Navigated by RTK-GPS and FOG (Part 3): Enhancement of turning Accuracy by Creating Path applied with Motion Constraints. J. Japanese Soc. Agric. Mach. 64, 102–110.
- Kise, M., Noguchi, N., Ishii, K., Terao, H., 2002b. Field Mobile Robot Navigated by RTK-GPS and FOG (Part 4): The Steering Controller Applied Optimal Controller. J. Japanese Soc. Agric. Mach. 64, 76–84.
- Kise, M., Noguchi, N., Ishii, K., Terao, H., 2001. Field Mobile Robot Navigated by RTK-GPS and FOG (Part 1): Estimation of Absolute Heading Angle by Sensor-fusion with RTK-GPS and FOG. J. Japanese Soc. Agric. Mach. 63, 74–76.
- Kise, Mi., Noguchi, N., Ishii, K., Terao, H., 2001. Field Mobile Robot Navigated by RTK-GPS and FOG (Part 2): Autonomous operation by applying navigation map. J. Japanese Soc. Agric. Mach. 63, 80–85.
- Kühne, F., Lages, W.F., Gomes da Silva, J.M., 2004. Model Predictive Control of a Mobile Robot Using Linearization, in: Proceedings of Mechatronics and Robotics. Aachen, pp. 525–530.
- Lachapelle, G., Alves, P., Fortes, L.P., Cannon, M.E., 2000. DGPS RTK Positioning Using a Reference Network, in: Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000). pp. 1165–1171.
- Lages, W.F., Vasconcelos Alves, J.A., 2006. REAL-TIME CONTROL OF A MOBILE ROBOT USING LINEARIZED MODEL PREDICTIVE CONTROL. IFAC Proc. Vol. 39, 968–973. <https://doi.org/10.3182/20060912-3-DE-2911.00166>
- Lapapong, S., Brown, A.A., Brennan, S.N., 2011. Frequency Characteristics and Explanation of Notches Seen in Frequency Responses of Vehicles, in: The Second TSME International Conference on Mechanical Engineering. Krabi, p. 8.
- Lavalle, S.M., 2006. PLANNING ALGORITHMS. Cambridge University Press.
- Ministry of Agriculture Forestry and Fisheries, 2015. 2015 census of agriculture and

- forestry in Japan [WWW Document]. e-Stat. URL <https://www.e-stat.go.jp/stat-search/files?page=1&layout=datalist&toukei=00500209&tstat=000001032920&cycle=7&year=20150&month=0&tclass1=000001077437&tclass2=000001077396&tclass3=000001085297> (accessed 6.7.19).
- Miya, M., Fujita, S., Sato, Y., Ota, K., Hirokawa, R., Takiguchi, J., 2016. Centimeter Level Augmentation Service (CLAS) in Japanese Quasi-Zenith Satellite System, its User Interface, Detailed Design, and Plan, in: Proceedings of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016). Portland, Oregon, pp. 2864–2869.
- Miya, M., Sato, Y., Fujita, S., Motooka, N., Saito, M., Takiguchi, J., 2014. Augmentation Service (CLAS) in Japanese Quasi-Zenith Satellite System, its Preliminary Design and Plan, in: Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014). Tampa, Florida, pp. 645–652.
- Nascimento, T.P., Dórea, C.E.T., Gonçalves, L.M.G., 2018. Nonlinear model predictive control for trajectory tracking of nonholonomic mobile robots. *Int. J. Adv. Robot. Syst.* 15, 172988141876046. <https://doi.org/10.1177/1729881418760461>
- Nof, S.Y. (Ed.), 2009. Springer Handbook of Automation. Springer Berlin Heidelberg, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-540-78831-7>
- Noguchi, N., Reid, J.F., Will, J., Benson, J.W.E.R., Stombaugh, T.S., 1998. Vehicle Automation System Based On Multi-Sensor Integration, in: 15 AT THE 1998 ANNUAL INTERNATIONAL MEETING, PAPER NO. 983111, ASAE, 2950 NILES RD., ST. St. Joseph, pp. 49085--9659.
- Ohshima, Y., Soga, H., Kawai, S., Kato, I., Taguchi, N., Mutoh, K., Yoshikawa, K., Osawa, T., Taniyama, A., Sakamoto, A., Sawamura, T., Tanaka, Y., 2015. Application of Precise Point Positioning (PPP) and Precise Real-Time Ephemeris to Achieve “where-in-lane” Positioning Accuracy for Vehicles (PDF Download Available), in: Proceedings of the ION 2015 Pacific PNT Meeting. Honolulu, pp. 583–590.
- Oksanen, T., Visala, A., 2009. Coverage path planning algorithms for agricultural field machines. *J. F. Robot.* 26, 651–668. <https://doi.org/10.1002/rob.20300>
- Oksanen, T., Visala, A., 2007. Path Planning Algorithms for Agricultural Machines. *Agric. Eng. Int. CIGR Ejournal IX*.
- Othman, M.F., Shazali, K., 2012. Wireless Sensor Network Applications: A Study in Environment Monitoring System. *Procedia Eng.* 41, 1204–1210. <https://doi.org/10.1016/J.PROENG.2012.07.302>
- Proctor, J.L., Brunton, S.L., Kutz, J.N., 2016. Dynamic mode decomposition with

- control. SIAM J. Appl. Dyn. Syst. 15, 142–161.
- Rajesh Rajamani, 2006. Vehicle Dynamics and Control. Springer.
- Richier, M., Lenain, R., Thuilot, B., Debain, C., 2013. Rollover prevention of All-Terrain Vehicle during aggressive driving using multi-model observer. Application to ATVS in off-road context, 2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies, AEECT 2013. <https://doi.org/10.1109/AEECT.2013.6716435>
- Sabelhaus, D., Röben, F., Meyer zu Helligen, L.P., Schulze Lammers, P., 2013. Using continuous-curvature paths to generate feasible headland turn manoeuvres. Biosyst. Eng. 116, 399–409. <https://doi.org/10.1016/j.biosystemseng.2013.08.012>
- Saito, M., Sato, Y., Miya, M., Shima, M., Omura, Y., Takiguchi, J., Asari, K., 2011. Centimeter-class Augmentation System Utilizing Quasi-Zenith Satellite, in: Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011). Portland, Oregon, pp. 1243–1253.
- Saito, M., Takiguchi, J., Okamoto, T., 2014. Establishment of Regional Navigation Satellite System Utilizing Quasi-Zenith Satellite System.
- Sato, K., Tateshita, H., Wakabayashi, Y., Kakimoto, H., Kogure, S., 2014. Asia Oceania Multi-GNSS Demonstration Campaign, in: XXV FIG Congress. Kuala Lumpur, Malaysia.
- Schubert, R., Richter, E., Wanielik, G., 2008. Comparison and evaluation of advanced motion models for vehicle tracking, in: 2008 11th International Conference on Information Fusion. IEEE, Cologne, pp. 1–6.
- Seigo, F., Masakazu, M., Junichi, T., 2015. Establishment of Centimeter-class Augmentation System utilizing Quasi-Zenith Satellite System [in Japanese]. Syst. Control Inf. 59, 126–131.
- Siciliano, B., Khatib, O. (Eds.), 2008. Springer Handbook of Robotics. Springer Berlin Heidelberg, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-540-30301-5>
- Sørensen, C.G., Bak, T., Jørgensen, R.N., 2004. Mission planner for agricultural robotics, in: Proceedings of AgEng. Leuven, Belgium, pp. 894–895.
- Suzuki, T., Kubo, N., Takasu, T., 2014. Evaluation of Precise Point Positioning Using MADOCA-LEX via Quasi-Zenith Satellite System, in: ION International Technical Meeting 2014. Institute of Navigation, San Diego, CA, pp. 460–470.
- Takai, R., Barawid, O., Noguchi, N., 2011. Autonomous navigation system of crawler-type robot tractor. IFAC Proc. Vol. 18, 14165–14169. <https://doi.org/10.3182/20110828-6-IT-1002.03355>
- Takasu, T., 2013. Development of Multi-GNSS Orbit and Clock Determination

- Software “MADOCA,” The 5th Asia Oceania Regional Workshop on GNSS.
- Tashiro, T., 2013. Vehicle steering control with MPC for target trajectory tracking of autonomous reverse parking, in: 2013 IEEE International Conference on Control Applications (CCA). IEEE, pp. 247–251. <https://doi.org/10.1109/CCA.2013.6662766>
- Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., Van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., Mahoney, P., 2006. Stanley: The Robot that Won the DARPA Grand Challenge. *J. F. Robot.* 23, 661–692. <https://doi.org/10.1002/rob.20147>
- Torisu, R., Nakatsubo, T., Imae, J., 1996. Minimum time control problem of tractor lateral motion, 1: Formulation and field experiment for minimum time control of tractor lane change maneuver. *J. Japanese Soc. Agric. Mach.* 58, 5–12.
- Torisu, R., Tanaka, K., Imae, J., 1998. Optimal Path of Headland for Tractors by Optimal Control Theory (Part 2): Combination of Forward and Backward Maneuvers. *J. Japanese Soc. Agric. Mach.* 60, 45–53.
- Tu, X., 2013. Robust navigation control and headland turning optimization of agricultural vehicles.
- van Essen, H.A., Nijmeijer, H., 2001. Non-linear model predictive control for constrained mobile robots, in: 2001 European Control Conference (ECC). IEEE, pp. 1157–1162. <https://doi.org/10.23919/ECC.2001.7076072>
- Wang, H., Noguchi, N., 2019a. Navigation of a robot tractor using the centimeter level augmentation information via Quasi-Zenith Satellite System. *Eng. Agric. Environ. Food.* <https://doi.org/10.1016/J.EAEF.2019.06.003>
- Wang, H., Noguchi, N., 2019b. Navigation of a Robot Tractor Using the Japanese Quasi-Zenith Satellite System (Part 1) - Application of Precise Point Positioning Method to a Robot Tractor -. *J. Japanese Soc. Agric. Mach. Food Eng.* 84, 250–255.
- Wang, H., Noguchi, N., 2018. Adaptive turning control for an agricultural robot tractor. *Int. J. Agric. Biol. Eng.* 11, 113–119. <https://doi.org/10.25165/j.ijabe.20181106.3605>
- Wang, H., Noguchi, N., 2016. Autonomous Maneuvers of a Robotic Tractor for Farming, in: 2016 IEEE/SICE International Symposium on System Integration (SII). IEEE, Sapporo, pp. 592–597.
- Wübbena, G., Schmitz, M., Bagge, A., 2005. PPP-RTK: Precise Point Positioning

- Using State-Space Representation in RTK Networks, in: Proceedings of ION GNSS. Long Beach, pp. 13–16.
- Yang, S.X., Luo, C., 2004. A Neural Network Approach to Complete Coverage Path Planning. *IEEE Trans. Syst. Man Cybern. Part B* 34, 718–724. <https://doi.org/10.1109/TSMCB.2003.811769>
- Yano Research Institute Ltd., 2017. Smart Agriculture Market in Japan: Key Research Findings 2017. Tokyo.
- Zumberge, J.F., Heflin, M.B., Jefferson, D.C., Watkins, M.M., Webb, F.H., 1997. Precise point positioning for the efficient and robust analysis of GPS data from large networks. *J. Geophys. Res. Solid Earth* 102, 5005–5017. <https://doi.org/10.1029/96JB03860>