2013

Robust navigation control and headland turning optimization of agricultural vehicles

Xuyong Tu
Iowa State University

Recommended Citation

Tu, Xuyong, "Robust navigation control and headland turning optimization of agricultural vehicles" (2013). *Graduate Theses and Dissertations*. 13188.
http://lib.dr.iastate.edu/etd/13188

Robust navigation control and headland turning optimization of agricultural vehicles

by

Xuyong Tu

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Major: Agricultural Engineering

Program of Study Committee:
Lie Tang, Major Professor
Stuart Birrell
Shufeng Han
Brian Steward
Qingze Zou

Iowa State University

Ames, Iowa

2013

DEDICATION

To my parents, Chengan Tu and Yingjiao Zheng

for always being supportive and a source of encouragement to me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEGEMENTS

I would like to express my deep and sincere gratitude to many people who made this thesis possible.

Thanks to my advisor, Dr. Lie Tang. I am so grateful I had the opportunity to meet and start working with Dr. Tang in the summer seven years ago. He always gave me innumerous support and inspirations when I needed help. I owe him lots of gratitude for teaching me to solve problems patiently, for helping me out from frustration and giving me confidence, for providing adequate time and space for self-development, and for supporting decisions I made in my life and career.

Thank you to all of the Agricultural Automation and Robotics Lab mates who gave me support and encouragement throughout my education. Especially I want to say thanks to Mr. Kasunobu Hayashi who trained me in electronic and mechanic skills at the beginning of my graduate study. Thanks to Jian Jin, Akash Narkami, Mohd Taufik Bin Ahmad, Ji Li, who gave me help and suggestion for this thesis study. They may not realize the scientific interactions inspired me a lot and my graduate study experience would be different without them.

I am lucky to be advised by numerous people during my graduate study. I would like to acknowledge and thank Professor Brian Steward and Professor Stuart Birrell who provided me valuable suggestions and critical comments on my graduate research work since the first day I came to Power and Machinery group. Thanks to Professor Shufeng Han who loaned me three GPS receivers and taught me how to install and integrate them with the robot system. I owe Professor Qingze Zou gratitude for supporting my thesis

study plan and providing helpful suggestions. Thanks to Professor Shuxin Du for training me during his short visit and helping me accomplish an important project that influenced my career life. I also want to show my hearty gratitude to Professor James Bernard who impressed me by his patience and kindness when I firstly arrived in this continent and gave me advices even after his retirement.

I wish to extend thanks to my mentors and colleagues at Research and Development Department of Cobb-Vantress Inc. My sincere gratitude should be given to my previous and current mentors, Dr. John Hardiman, Dr. Manouchehr Katanbaf, Dr. Mitchell Abrahamsen, and Dr. Mark Cooper, who have always been supportive and encouraging on my Ph.D. study continuity. I need to say thank you to those helpful at Cobb R&D and Production group, Dr. Hilary Pavlidis, Dr. Robyn Sapp, Scott Noble, Cody Polley, and Adam Eastgate, for backing me up when I had difficulty on balancing work and life. I am also grateful to the coworkers at Tyson Foods IS and Engineering Department, Darryl Young, Doug Foreman, and James Ruff, for showing me the impressive dedication of profession and the spirit of team work.

Most importantly, I would like to thank my family for being supportive throughout my education. My parents have always provided me with everything I needed to further my education. Their unconditional love is the best asset I have ever had in my life. Special thanks to my wife Yixiao Sheng. I could not get through all the difficult times without her companion and support. Thanks to my son William Tu whose cuteness is always there to distract me from working and graduate pressures. Thanks to my parents-in-law who gave me support, help and encouragement when I needed them. Also thanks to my upcoming baby for giving me motivation to get a Ph.D. degree before his birth.

Finally, I wish to thank my friends Jie Wang, Hongbin Lu, Chengzong Han, Anan Fang, Yang Li, Zhengqing Gan, Weitao Dai, Qian Wang, Yongcheng Liu, Xiaobo Zhou, Zhiming Qi, Lide Chen, Gang Sun, Qi Wang, Nan Xiao, Xiuyan Yang, Lingyuan Yang, for all the emotional support and those sunny happy days in my life.

ABSTRACT

Autonomous agricultural robots have experienced rapid development during the last decade. They are capable of automating numerous field operations such as data collection, spraying, weeding, and harvesting. Because of the increasing demand of field work load and the diminishing labor force on the contrary, it is expected that more and more autonomous agricultural robots will be utilized in future farming systems.

The development of a four-wheel-steering (4WS) and four-wheel-driving (4WD) robotic vehicle, AgRover, was carried out at Agricultural Automation and Robotics Lab at Iowa State University. As a 4WS/4WD robotic vehicle, AgRover was able to work under four steering modes, including crabbing, front steering, rear steering, and coordinated steering. These steering modes provided extraordinary flexibilities to cope with off-road path tracking and turning situations. AgRover could be manually controlled by a remote joystick to perform activities under individual PID controller of each motor. Socket based software, written in Visual C#, was developed at both AgRover side and remote PC side to manage bi-directional data communication. Safety redundancy was also considered and implemented during the software development.

One of the prominent challenges in automated navigation control for off-road vehicles is to overcome the inaccuracy of vehicle modeling and the complexity of soil-tire interactions. Further, the robotic vehicle is a multiple-input and multiple-output (MIMO) high-dimensional nonlinear system, which is hard to be controlled or incorporated by conventional linearization methods. To this end, a robust nonlinear navigation controller was developed based on the Sliding Mode Control (SMC) theory

and AgRover was used as the test platform to validate the controller performance. Based on the theoretical framework of such robust controller development, a series of field experiments on robust trajectory tracking control were carried out and promising results were achieved.

Another vitally important component in automated agricultural field equipment navigation is automatic headland turning. Until now automated headland turning still remains as a challenging task for most auto-steer agricultural vehicles. This is particularly true after planting where precise alignment between crop row and tractor or tractor-implement is critical when equipment entering the next path. Given the motion constraints originated from nonholonomic agricultural vehicles and allowable headland turning space, to realize automated headland turning, an optimized headland turning trajectory planner is highly desirable. In this dissertation research, an optimization scheme was developed to incorporate vehicle system models, a minimum turning-time objective, and a set of associated motion constraints through a direct collocation nonlinear programming (DCNLP) optimization approach. The optimization algorithms were implemented using Matlab scripts and TOMLAB®/SNOPT tool boxes. Various case studies including tractor and tractor-trailer combinations under different headland constraints were conducted. To validate the soundness of the developed optimization algorithm, the planner generated turning trajectory was compared with the hand-calculated trajectory when analytical approach was possible. The overall trajectory planning results clearly demonstrated the great potential of utilizing DCNLP methods for headland turning trajectory optimization for a tractor with or without towed implements.

CHAPTER 1: GENERAL INTRODUCTION

1.1 Introduction

With the development of robotic technologies, autonomous agricultural robots are expected to relieve farm labour from having to do monotonous, unproductive and uncreative work, such as weeding (Watanabe, 2004), data sampling (Nejati, et al, 2008) and harvesting (Katupitiya, 2005). Autonomous agricultural robots are capable of collecting crop and soil samples (Wang, 2007) because they can be small in size, which allows them to be able to acquire data close to the crops. They are also capable of mowing (Jarvis, 2001), spraying pesticides (Chen, et al, 2002; Zhang, et al, 2005), finding diseases or parasites, and performing precision mechanical weeding. Agricultural robots that are equipped with cameras and sensors were used to harvest fruit and vegetables such as apples (Kataoka, et al, 1999), melons (Edan and Miles, 1993), radicchio (Foglia, et al, 2006) and asparagus (Dong, et al, 2011).

As the basis of autonomous field robot technology, motion control for agricultural robotic vehicle has been an active research area in recent years (Hargas, et al 2002; Kondo, 2005). However, like other MIMO nonholonomic nonlinear systems, agricultural robotic vehicles have system uncertainties and time-varying parameters, especially when working in off-road environment. In addition, external factors such as soil and wind conditions also affect vehicle dynamic characteristics. Both unpredictable internal perturbations and external disturbances create a great challenge of controlling such a nonholonomic nonlinear system. Imprecision in modelling a nonlinear system may come from the actual uncertainty of the plant or from the purposeful choice of a simplified representation of the system dynamics. It is desirable to develop a capable and robust navigation controller that is able to navigate the vehicle accurately and effectively, overcome system uncertainties, resist external disturbances, and mostly important, maintain system stability.

The method of backstepping decomposes complex nonlinear system to lower dimension subsystems that are designed to make the whole system perform as desired by using Lyapunov function and pseudo-control variables. Compared with the feedback linearization method (Kanayama, et al, 1990; Kim and Oh, 1999; Wang, 2005), backstepping does not require an accurate model while still maintains system properties and nonlinearity. Sliding mode control (SMC) of the variable structure control (VSC) has strong robustness to parameter perturbation and external disturbances (Cheng, et al, 2007). In contrast to fuzzy control (Hagras, et al 1999) and adaptive control (Dixon, et al, 2004), SMC is more robust and responsive, which is suitable for off-road conditions. The approach of combining backstepping and sliding mode control has demonstrated good performance for controlling both linear and nonlinear systems (Lin, Shen, 2002; Shen, Lin, 2005). Thus it is worthwhile to investigate how backstepping SMC will perform on an agricultural robotic vehicle.

As an essential part of agricultural vehicle navigation control and field coverage path planning process, headland turning is a process that has to meet the given initial and final conditions when leaving and reentering the crop field. The headland turning process should also be done in a minimum time and travel within a restricted headland space (Jin, 2009). The optimization of headland turning control can reduce time and fuel consumption during headland turning and in turn improve the overall field efficiency of agricultural field equipment. Therefore optimization of the headland turning trajectory is of great interest for agricultural equipment manufacturers. However, this optimization process has until now remained largely unresolved, primarily due to the difficulties of finding such an optimized solution in an enormously large solution space. Off-road vehicle models are complex and non-linear, making linear optimization algorithms unsuitable for solving the problem (Oksanen, 2004). Furthermore, the tractor towed implements further increase the complexity of the model and make this dynamic nonlinear problem even more challenging to solve.

Numerical solutions for the trajectory optimization problem can be categorized into two approaches, indirect methods and direct methods (Conway, 2012). Indirect solutions use analytical necessary conditions from the calculus of variations, which requires the addition of co-state variables of the problem. However, when using indirect methods, the derivation of the necessary conditions, including differential equations, boundary conditions and path constraints, is usually a intractable mathematical procedure. A minor change of any of these constraints may cause considerable amount of computation, even with the help of software such as POST® (Program to Optimize Simulated Trajectories, Sierra Engineering Inc., Carson City, Neveda) or GTS (Geostatistical Temporal-Spatial algorithm, The Aerospace Corporation, El Segundo, CA). In contrary, direct solutions convert the optimization problem into a Nonlinear Programming (NLP) problem which transcribes a continuous optimal control problem into a parameter optimization problem (Conway and Paris, 2011; Ross and Fahroo, 2003; Fahroo and Ross, 2002). A NLP method integrates the system equations stepwise using either implicit or explicit rules. Compared to indirect methods, considerable advancements have been made in direct methods over the last decade (Conway, 2012).

Commercial software packages such as TOMLAB® (Tomlab, Stockholm, Sweden), DIDO® (Elissar Global, Carmel, CA), and SOCS® (Boeing, Seattle, WA) are available to facilitate the implementation of direct methods for solving optimal control problems. Therefore, exploration of applying direct methods for agricultural field equipment headland turning trajectory optimization represents a potential significant contribution toward this long-standing problem, which is important to both our research community and the equipment industry.

1.2 Research Objectives

The overall goal of the proposed research is to investigate navigation control of autonomous agricultural vehicles under off-road conditions. To better understand and achieve this goal, the research work was divided into two main objectives.

1) To design and develop a robust navigation controller for the off-road vehicle to follow a given reference trajectory. The specific objectives for this phase were:
    a) To develop a prototype of autonomous agricultural robotic vehicle;
    b) To develop the vehicle model and the theoretical derivation of a robust controller that has ensured system stability and performance;
    c) To validate the controller design by using an autonomous agricultural robotic vehicle under field conditions.
2) To optimize the navigation control trajectory in the field headland turning operation with given boundary conditions and constraints. The specific objectives for this second phase were:
    a) To set up the research framework of assumed headland turning scenario;

b) To build various system models of agricultural vehicles and vehicle-implement combinations in the established framework;

c) To investigate the feasibility of using direct collocation nonlinear programming methods to solve the targeted optimization problem.

1.3 Dissertation Overview

This dissertation consists of four parts which are detailed in Chapter 2 to Chapter 5. In Chapter 2, the development of AgRover robotic vehicle platform is described. Chapter 3 is a research paper about backstepping sliding mode control designed for AgRover. Vehicle dynamic model was established and a sliding mode control based navigation controller was simulated in MATLAB. Chapter 4 introduces the implementation of a SMC controller on AgRover. Field experiment results are presented and discussed. Chapter 5 covers the development of a headland turning optimization procedure, including problem framework setup, vehicle modeling, and the formulation of the optimization problem. Simulation results are presented and discussed in this chapter.

References

Bchtis, D., Vougioukas, S., Tsatsarelis, C., and Ampatzidis, Y., (2007) Optimal Dynamic Motion Sequence Generation for Multiple Harvesters. Agricultural Engineering International: the CIGR Ejournal. Manuscript ATOE 07001. Vol. IX.

Betts, J. T..(1998) Survey of Numerical Methods for Trajectory Optimization, Journal of Guidance, Control, and Dynamics, Vol. 21, No. 2, pp. 193–207. doi: 10.2514 / 2.4231

Chen, C., Li, T., Yeh, Y., Chang, C. (2009) Design and implementation of an adaptive sliding-mode dynamic controller for wheeled mobile robots. Mechatronics; 19(2): 156-166.

Cheng, J., Yi, J., Zhao, D., (2007) Design of a sliding mode controller for a trajectory tracking problem of marine vessels. Control Theory & Applications; 1(1): 233-237.

Conway, B.A. (2012). A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems. J. Optim. Theory Appl. 152:271-306

Conway, B.A., Paris, S.W.(2011). Spacecraft trajectory optimization using direct transcription and nonlin-ear programming. In: Conway, B.A. (ed.) Spacecraft Trajectory Optimization. Cambridge University Press, Cambridge.

Dixon, W., De Queiroz, M., Dawson, D., Flynn, T., (2004) Adaptive tracking and regulation of a wheeled mobile robot with controller/update law modularity. IEEE Transactions on Control Systems Technology; 12(1): 138-147.

Dong, F., Heinemann, W., Kasper, R., (2011) Development of a row guidance system for an autonomous robot for white asparagus harvesting, Computers and Electronics in Agriculture, Vol.79(2), pp.216-225

Edan, Y., Miles, G., (1994) Systems engineering of agricultural robot design, IEEE Transactions of Systems, Man and Cybemetics, Vol. 24(8), pp. 1259

Enright, P.J., Conway, B.A.(1992) Discrete approximations to optimal trajectories using direct transcription and nonlinear programming. J. Guid. Control Dyn. 15, 994–1002.

Fahroo, F., Ross, I.M.(2002) Direct trajectory optimization by a Chebyshev pseudospectral method. J. Guid. Control Dyn. 25, 160–166.

Foglia, M., Reina, G., Corke, P., Sukkarieh, S., (2006) Agricultural robot for radicchio harvesting, Journal of Field Robotics, Vol. 23, pp. 363-377

Gillespie, T. D., (1992) Fundamentals of Vehicle Dynamics. Society of Automotive Engineers, Inc. Warrendale.

Hagras, H., Callaghan, V., Colley, M., Carr-West, M., (1999) A fuzzy-genetic based embedded-agent approach to leaning and control in agricultural autonomous vehicles. Proceeding of the IEEE International Conference Robotics and Automation. p. 1005-1010.

Hagras, H., Colley, M., Callaghan, V., Carr-West, M.,(2002) Online Learning and adaptation of autonomous mobile robots for sustainable agriculture. Autonomous Robots; 13(1): 37-52.

Hargraves, C.R., Paris, S.W.(1987). Direct trajectory optimization using nonlinear programming and collo-cation. J. Guid. Control Dyn. 10, 338–342.

Jin, J. (2009). Optimal field coverage path planning on 2D and 3D surfaces. PhD dissertation, Iowa State University.

Kanayama, Y., Kimura, Y., Miyazaki, F., Noguchi, T., (1990) A stable tracking control method for an autonomous mobile robot. Proceedings of the IEEE conference on Robotics and Automation. p. 384-389.

Kataoka, T., Hiroma, T., Ota, Y., (1999) Development of a harvesting hand for apples, Advanced Robotics, Vol. 13(3), pp. 293-294

Katupitiya, J., Eaton, R., Cole, A., Meyer, C., Rodnay, G., (2005) Automation of an Agricultural Tractor for Fruit Picking. Robotics and Automation: Proceedings of the IEEE International Conference. p. 3201-3206.

Kim, D., Oh, T., (1999) Tracking control of a two-wheeled mobile robot using input-output linearization. Control Engineer Prac; 7: 369-73.

Kondo, N., (2005) Latest agricultural robots and traceability information based on robotic agriculture. Resource: Engineering and Technology for Sustainable World; 12(7): 3-4.

Lin, F., Shen, P., (2002) Adaptive backstepping sliding mode control for linear induction motor drive. IEE Proceedings- Electric Power Application; 149(3): 184-194.

Liu, J., (2005) MATLAB Simulation for Sliding Mode Control: Tsinghua University Press.

Nejati, H., Azimifar, Z., Zamani, M., (2008) Using fast fourier transform for weed detection in corn fields. Proceeding of the IEEE International Conference Systems, Man and Cybernetics. p. 1215-1219.

Oksanen, T., Visala, A. (2004). Optimal Control of Tractor-Trailer System in Headlands. Proceedings of the 7-8 October 2004 Conference. ASAE Publication Number 701P1004.

Oksanen, T., Visala, A. (2009) Coverage Path Planning Algorithms for Agricultural Field Machines. Journal of Field Robotics. 26(8), 651-668.

Ross, I.M., Fahroo, F.(2003) Legendre pseudospectral approximations of optimal control problems. In: Lec-ture Notes in Control and Information Sciences, vol. 295, pp. 327–342. Springer, New York

Shen, P., Lin, F., (2005) Intelligent backstepping sliding-mode control using RBFN for two-axis motion control system. IEE Proceeding electrical Power Application; 152(5): 1321-1342.

Snider, J.M., (2009). Automatic Steering Methods for Autonomous Automobile Path Tracking. Dissertation. Robotics Institute, Carnegie Mellon University.

Wang, S., Wang, P., (2005) Nonlinear Modeling and Analysis of Vehicle Planar Motion Dynamics. IEEE International Conference on Mechatronics; p. 90-95.

Wang, Y., (2007) Robot-assisted sensor network deployment and data collection, Computational Intelligence in Robotics and Automation, pp. 467-472 Jun. 20-23

Watanabe, T., (2004) A duck robot for weeding work on the paddy field. Proceeding of the First IEEE Technical Exhibition Based Conference on Robotics and Automation. p. 81-82

Zhang, S., Liu, W., Wang, Z., (2005) Research on spray-paint robot position control system based on BP neural network, robotics and Biomimetics, IEEE International Conference on, pp. 742-746

Zhou, J., Zhang, M., Liu, G., Li, S., (2008) Fuzzy Control for Automatic Steering and Line Tracking of Agricultural Robot. Proceeding of the International Conference on Computer Science and Software Engineering,. p. 1094-1097.

CHAPTER 2: AGROVER DESIGN AND DEVELOPMENT

2.1 Background Introduction

Agricultural robotics experienced rapid development during the last two decades. Robotics has been extended to numerous bio-systems such as agriculture, forestry (Liu, 2008), and fisheries (Tan, et al, 2006). Applying automation to agriculture has helped create several advancements to the industry while helping farmers save money and time.

Autonomous agricultural robots are capable of collecting crop and soil samples (Wang, 2007) because they are small in size, which allows them to be able to accumulate data close to the crops. They are also capable of mowing (Jarvis, 2001), spraying pesticides (Chen, et al, 2002; Zhang, et al, 2005), finding diseases or parasites, and performing mechanical weeding. Agricultural robots come equipped with cameras and sensors that are used to harvest fruit and vegetables such as apples (Kataoka, et al, 1999), melons (Edan and Miles, 1993), radicchio (Foglia, et al, 2006) and asparagus (Dong, et al, 2011).. Their sensors are used to detect weeds (Have, et al, 2005; Bin Ahmad, 2012) and spray only the area affected by the parasite instead of the entire crop. This has helped to protect our environment by reducing the amount of harmful chemicals released in the air. Advanced computer and electronic technology plays an important role in this tremendous development, and it is also the basis of the research in future.

Because of the stated advantages and increasing need in autonomous in field mobile robots, a four-wheel-steering (4WS) and four-wheel-drive (4WD) robotic vehicle, AgRover, was developed at the Agricultural Automation and Robotics Lab of Iowa State University. The evolution of AgRover consists of two generations, Generation I and Generation II. Gen. I had a pneumatic self-leveling system, which offered AgRover capability that keeping platform flat while working on uneven field. However, because of the limited responsiveness and maneuverability, it was removed during the development of Gen. II. Whereas Gen. II does not have the self-leveling system as Gen. I had, Gen. II has a three-point chassis design and is still able to handle uneven terrain. Moreover, it is more agile and flexible to be able work under off-road conditions because of the lighter platform and weight decrease.

As a 4WS/4WD robotic vehicle, AgRover can work under four steering modes, crab steer, front wheel steer, rear wheel steer, and coordinated steer. They provide enormous flexibility to AgRover to be able to work out some difficult turning situations. And each of the modes has unique advantages under certain scenarios, which will be discussed in the paper. AgRover Gen. II could be manually controlled by a remote joystick to perform movements and steering mode shift under PID control. Socket based software, written in Visual C#, was developed at both AgRover side and remote PC side to handle the mutual communication. Safety redundancy was considered in the software development and operation procedure design.

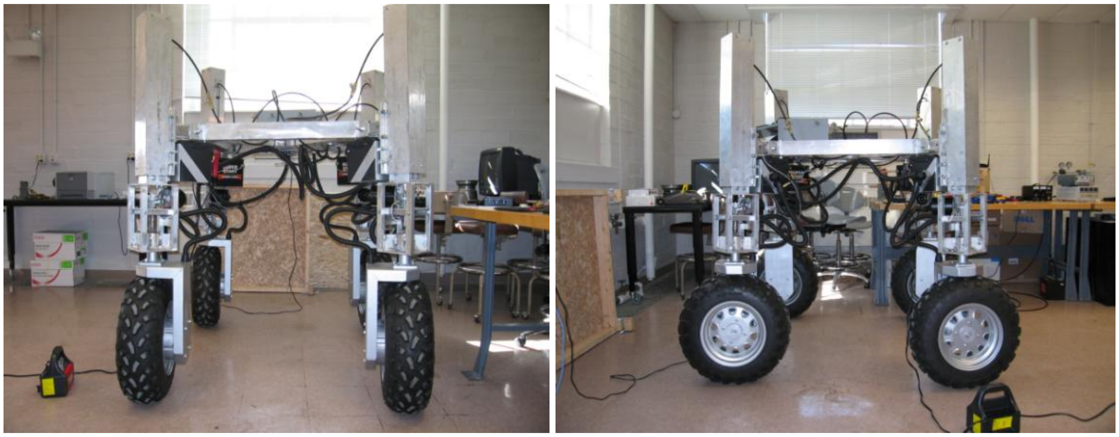
Based on the well developed manual control and precise RTK-GPS technology, an automatic navigation controller was developed and implemented. Related simulation was done by MATLAB (Chapter 3). The shift between manual mode and automatic mode was secure and safe. A series of field experiments of trajectory tracking navigation control were carried out and the results were presented (Chapter 4).

2.2 AgRover Evolution

2.2.1 AgRover Gen. I

The major purpose to develop AgRover was to make a flexible and robust experimental platform that is able to work under off-road conditions and handle possible data sampling and collection job for the potential field research projects. It had unique features compared with other field robots which will be presented and discussed in this chapter.

2.2.1.1 High ground clearance



(2.1a) *(2.1b)*

Figure 2.1: *Front (2.1a) and side (2.1b) view of AgRover Gen. I*

AgRover Gen. I had high ground clearance (Figure 2.1), which was a great advantage for carrying out field experiments. With the help of pneumatic system, the ground clearance could be adjusted from a minimum of 1.3 m to a maximum of 1.8 m.

2.2.1.2 Expandable chassis

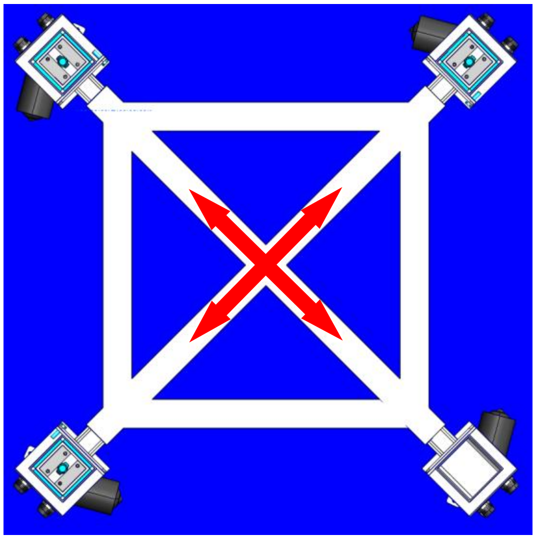
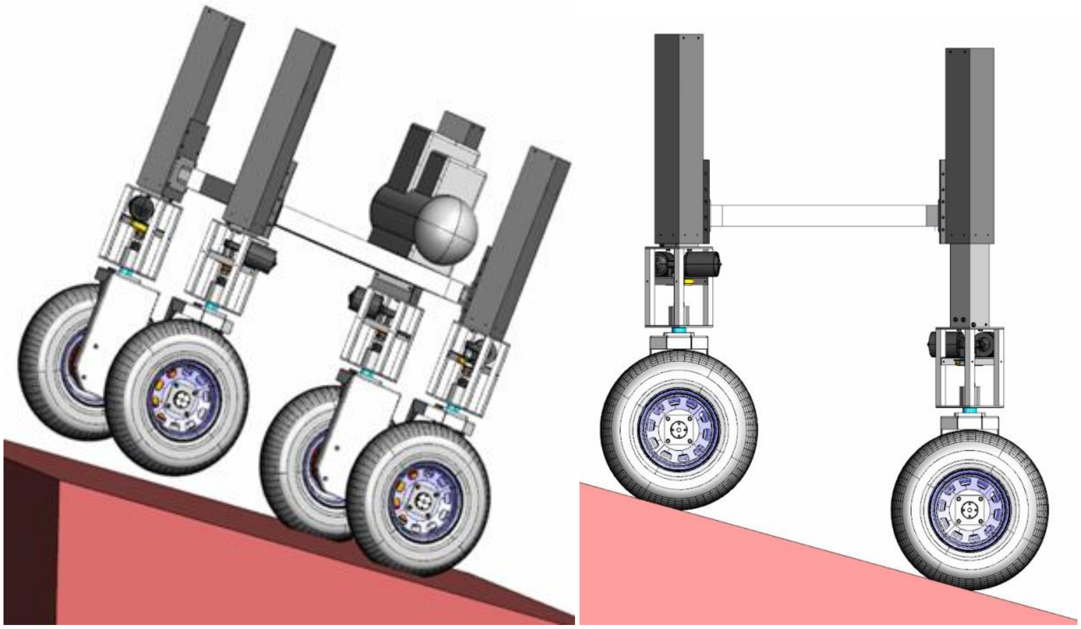


Figure 2.2: *Top view of expandable AgRover chassis*

The retractable leg design and flexible “X” shaped frame in the square shaped chassis of AgRover Gen. I is enabled track width and wheel-base adjustability. The AgRover could then be flexibly adjusted for different row widths in the field.

2.2.1.3 Self-leveling system

As mentioned, AgRover was designed for dealing with off-road conditions. It is normal for the robot to work on hills or uneven field. By carrying a computer, sensors, a generator and an air compressor, AgRover is fairly heavy (270 kg). With the high ground clearance, the platform could also be vibrating or shaking hard in the field. Under motion mode, it may increase the instability of the whole vehicle (Figure 2.3a), it was desired to have a leveling controller to keep the platform steady by utilizing the pneumatic system (Figure 2.3b).



(2.3a) *(2.3b)*

Figure 2.3: *Functionality of AgRover pneumatic self-leveling system*

With heavy carried equipments and high ground clearance, driving on uneven off-road fields could cause extremely dangerous consequences such as roll over or pitch over (Figure 2.3a). An intelligent controller was required that could control individual pneumatic cylinders and maintain the platform flat and stable, so that there was no worry about the safety of the vehicle. Therefore with this secured protection the data collection experiments could even be done under this circumstance. AgRover did have such leveling control system (Figure 2.4).

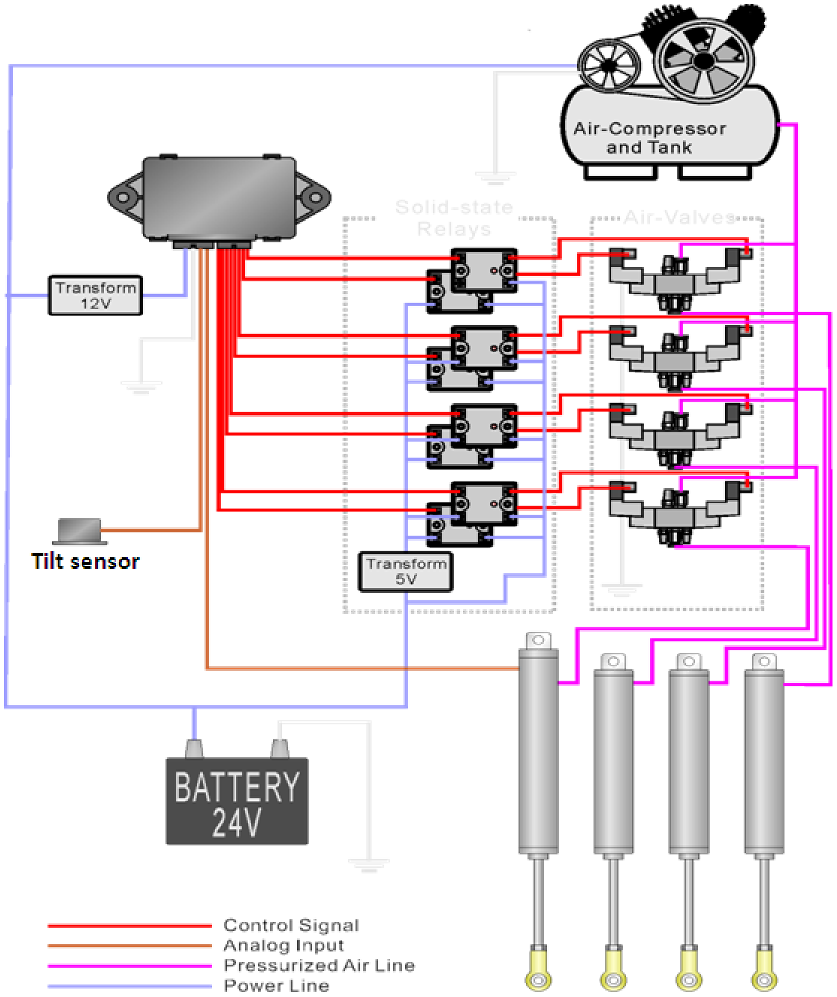


Figure 2.4: *Intelligent pneumatic self-leveling system*

The self-leveling system had its own microcomputer which was independent from but could be communicated with the vehicle central controller. Each leg had a cylinder installed. There was a linear potentiometer on top of the cylinder of the front-left leg, which indicated and acted as the “Control” height compared to the other cylinders. By cooperating with a tilt sensor mounted on the platform, the controller was able adjust the other three cylinders dynamically to balance the platform.

2.2.2 AgRover Gen. II

The pneumatic self-leveling system was an intelligent and valuable design because of its flexibility. However, the bulky pneumatic components and sluggish pneumatic delay did affect the vehicle mobility and agility. After multiple times of field validation and maneuverability tests, it was decided to remove the leveling system from the platform. After redesign and re-integration, AgRover Gen. II came out at Agricultural Automation and Robotics Lab in 2010 (Figure 2.5).

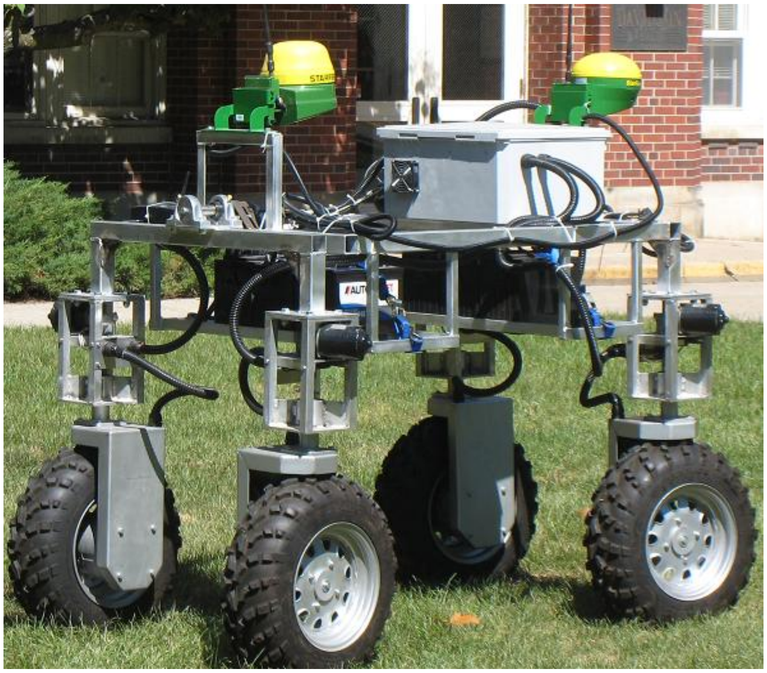


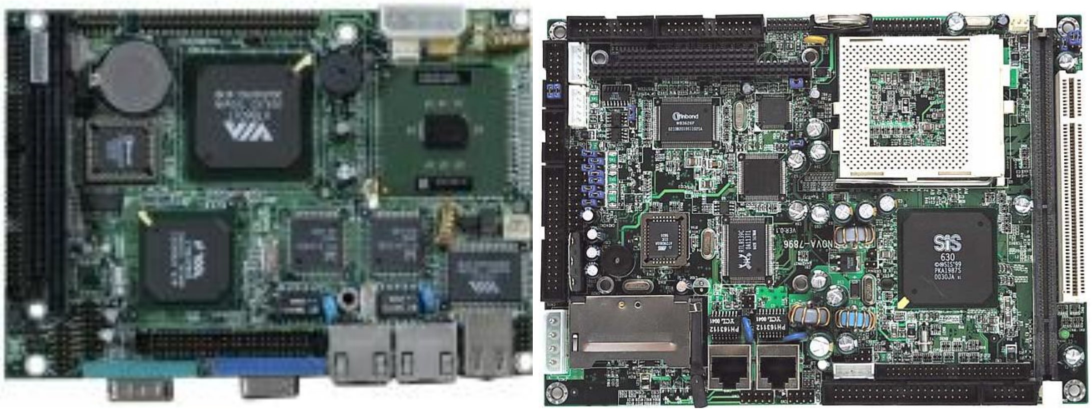
Figure 2.5: *AgRover Gen. II*

2.2.2.1 Lighter chassis and better maneuverability

Though AgRover Gen. II did not have flexibility of leveling or chassis expandability, it maintained remarkable ground clearance to accommodate field tasks. And with lighter curb weight and lower center of gravity, the agility and handling got tremendous improvement.

2.2.2.2 Upgraded central controller

AgRover used embedded industrial computer as its central controller. On Gen. I, the controller was IEI WAFER-C400 with Celeron 300MHz processor. On Gen. II it was upgraded to IEI NOVA 7896 with Pentium III 1.2GHz processor. The system memory got expanded from 256MB to 512MB.

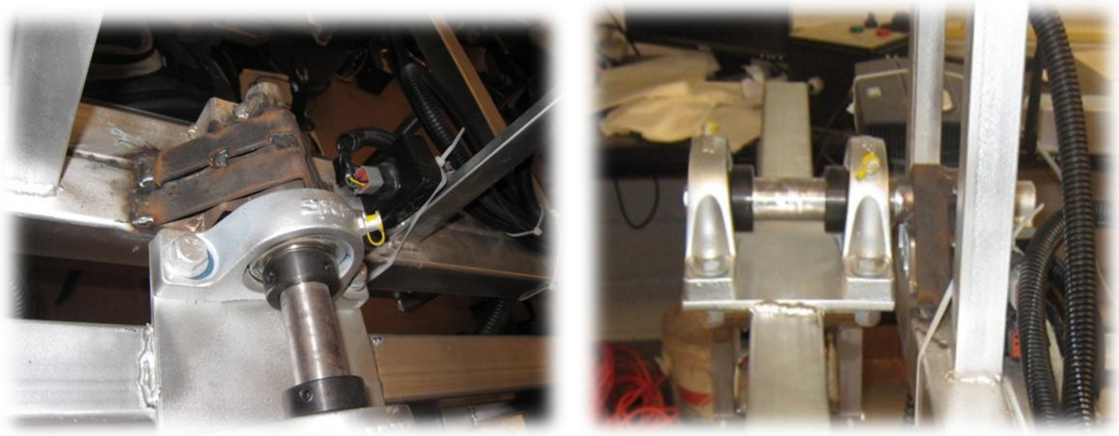


(2.6a) WAFER-C400 Celeron 300MHz *(2.6b) NOVA 7896 Pentium III 1.2GHz*

Figure 2.6: *Embedded industrial computers for AgRover Gen. I (2.6a) and Gen. II (2.6b)*

2.2.2.3 Triangle chassis design

The new chassis was designed according to the “Triangle” (three point) vehicle chassis principle. The platform and the rear axle are connected by a pivot (Figure 2.7).



(2.7a) Top view *(2.7b) Side view*

Figure 2.7: *Pivot of triangle chassis design*

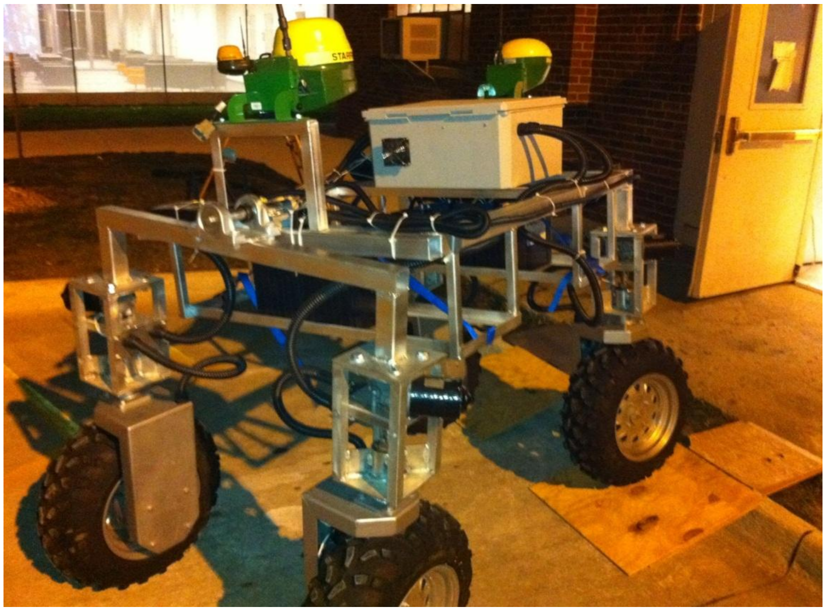


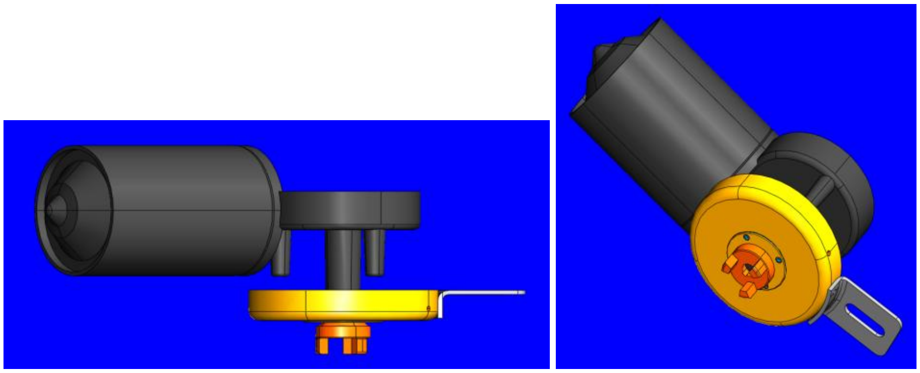
Figure 2.8: *AgRover Gen. II on uneven surface*

Triangle chassis design helped all of the wheels that could provide traction while the vehicle is working on an uneven field situation (Figure 2.8).

2.3 Steering and Driving System

2.3.1 Steering

The steering system for each leg consisted of two components, a steering motor (Figure 2.9a) and an encoder (Figure 2.9b). The steering motor was manufactured by VALEO®, which was driven by a CURTIS® amplifier. The associated encoder's resolution is 2048 pulses/rev, with an index channel available.



(2.9a) Side view *(2.9b) Bottom view*

Figure 2.9: *Steering motor and encoder*

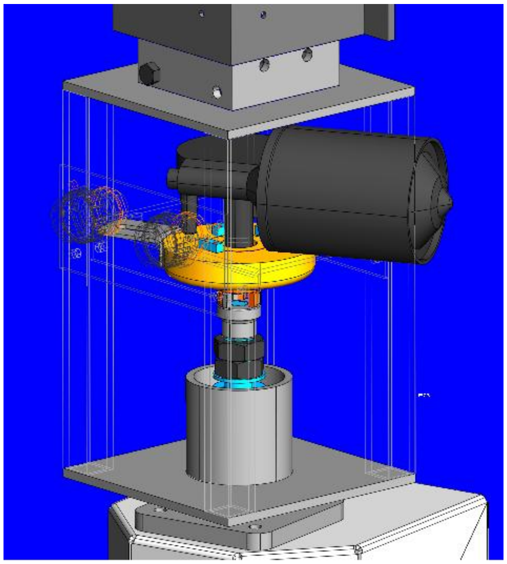
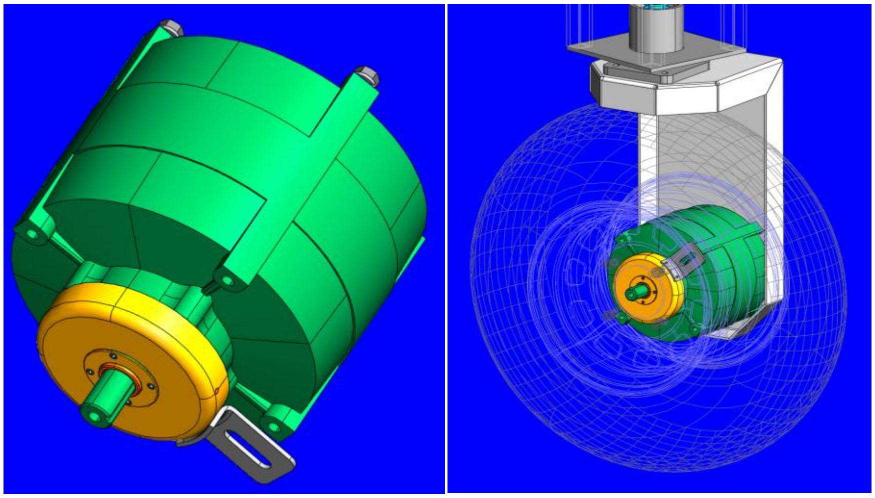


Figure 2.10: *Steering composition*

2.3.2 Driving

Similarly, the driving system for each wheel was a combination of two components as well, a driving motor and an encoder. The HONDA® DC motor worked with 24V and provided 150W output. The encoder's resolution was 1024 pulses/rev.



(2.11a) Driving motor and encoder *(2.11b) Driving composition*

Figure 2.11: *Driving motor, encoder and composition*

2.4 Control System

2.4.1 AgRover control architecture

With the steering and driving hardware, the preparation for the central controller design was done. The AgRover manual control could be decomposed by three levels (Figure 2.12), including remote monitoring and control station, controller on AgRover, and motors with encoders.

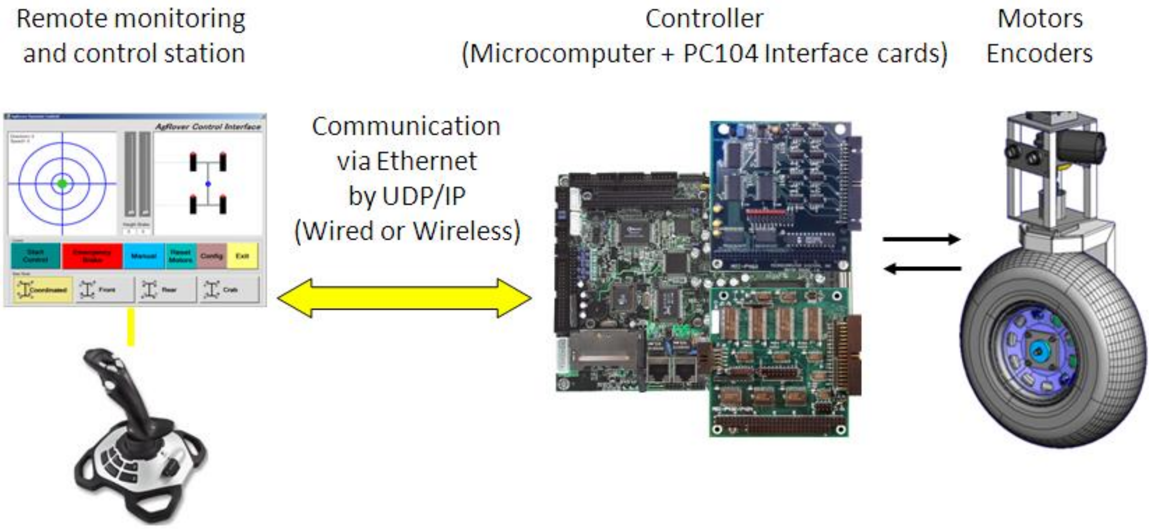


Figure 2.12 *Three levels: AgRover manual control*

The remote monitoring and control station ("remote station" in the text to follow) collected settings and inputs from the operation program or from the joystick, and sent UDP packages to the AgRover controller over Wi-Fi. The controller delivered translated commands to each control port via PC104 cards and read feedback from the encoders, and then transmitted feedback to the remote station. The feedback was instantly visibly displayed on the operation screen, and presented the vehicle working status to the operator.

2.4.2 Remote station

Remote station consisted of a laptop and a joystick (Figure 2.13), where the joystick is a pure input device and the laptop took in charge of the communication and data display.

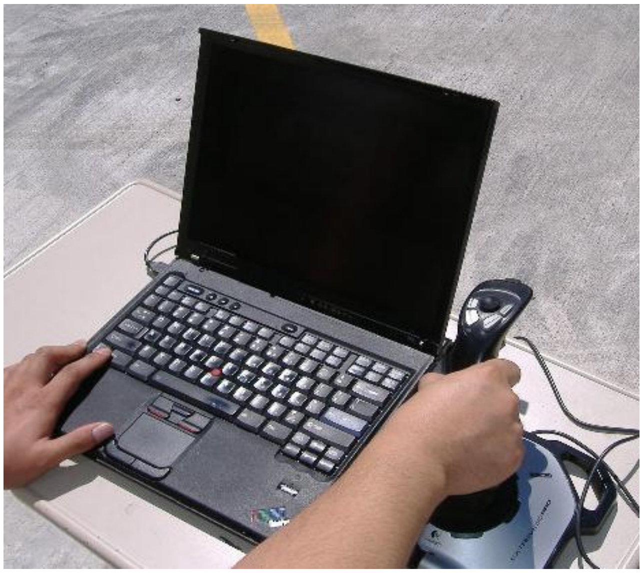
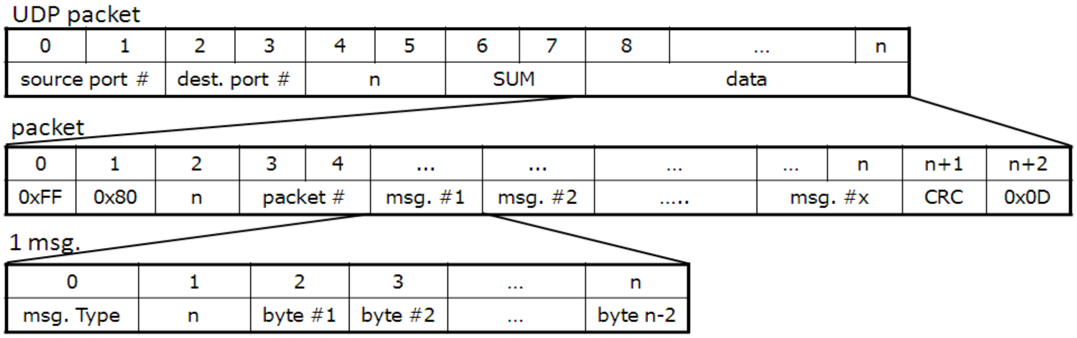


***Figure 2.13:** AgRover remote station: laptop + joystick*

The communication between the remote station and AgRover was via WLAN UDP/IP protocol. And the package composition was as below (Figure 2.14). All data packets, started with OxFF and Ox80 and ended with 0x0D, consisted of 23 different types of messages.



UDP packet

0	1	2	3	4	5	6	7	8	...	n
source port #		dest. port #		n		SUM		data		

packet

0	1	2	3	4	...	...	...	...	n	n+1	n+2
0xFF	0x80	n	packet #		msg. #1	msg. #2	.....	msg. #x		CRC	0x0D

1 msg.

0	1	2	3	...	n
msg. Type	n	byte #1	byte #2	...	byte n-2

***Figure 2.14:** UDP communication package composition*

2.4.2.1 Joystick operation

The setting and operations of AgRover could be handled on the joystick such as steering mode setting (Figure 2.15), driving control (Figure 2.16), and emergency handling (Figure 2.17) etc.

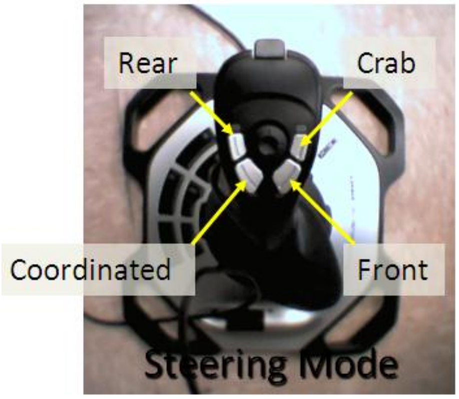


Figure 2.15: *Steering mode setting on the joystick*

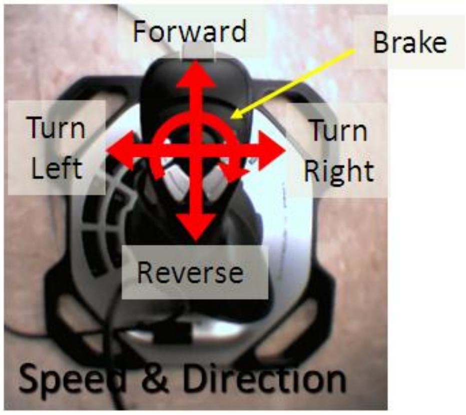


Figure 2.16: *Steering and driving control on the joystick*

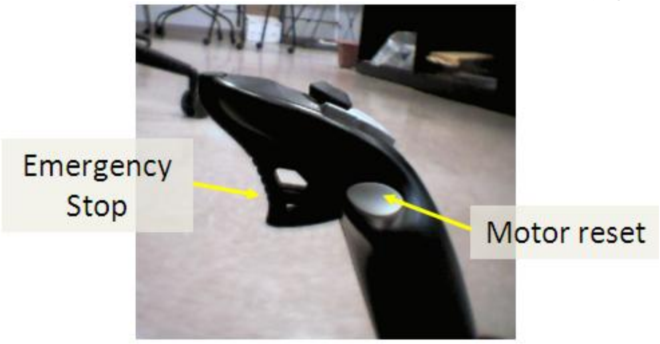
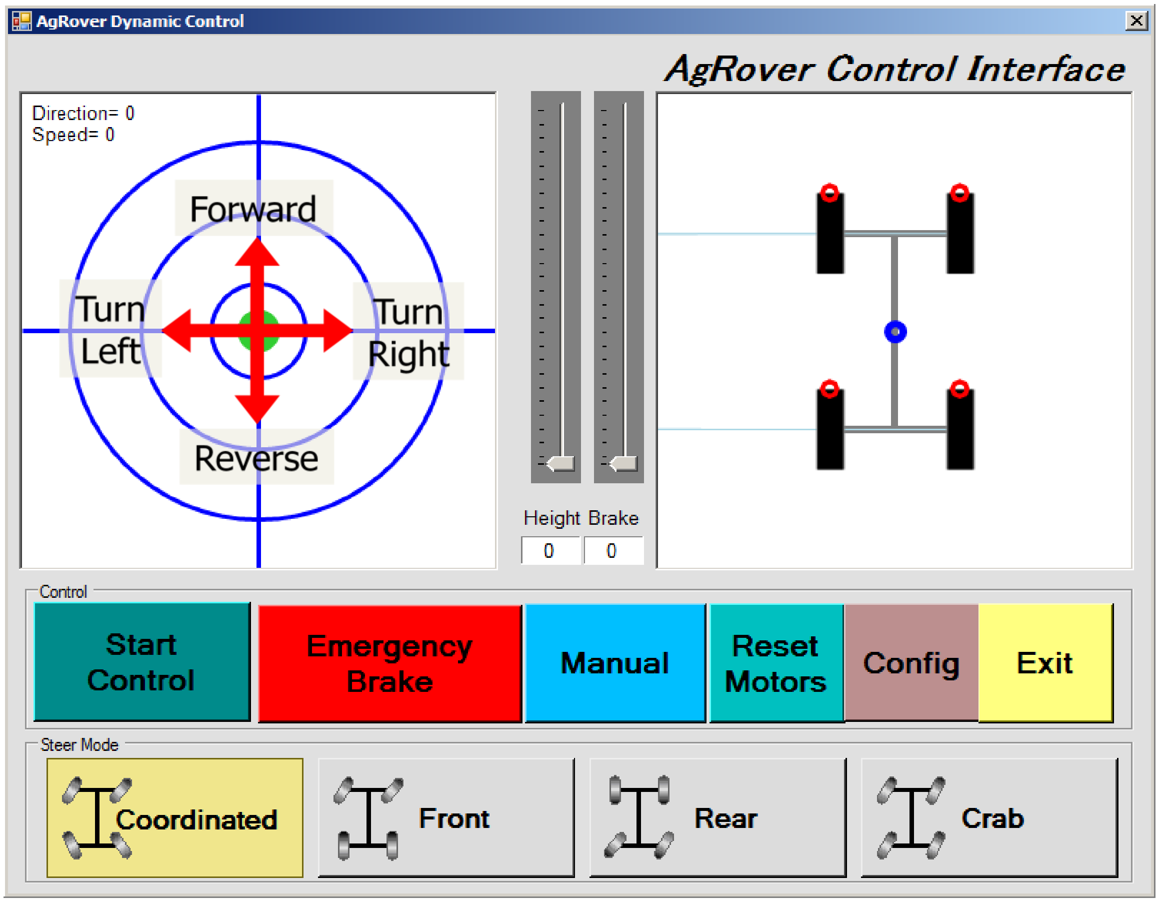


Figure 2.17: *Emergency handling on the joystick*

2.4.2.2 Remote operation program

All functions on the joystick could be realized on the remote operation window (Figure 2.18), in addition to using joystick. For example, driving control could be done by dragging the green bubble on the “Direction and Speed” interface. The steering mode could be selected by clicking the buttons at the bottom row. Moreover, the buttons of “Emergency Brake” and “Reset Motors” did the same job as their counterparts on the joystick.



Direction= 0
Speed= 0

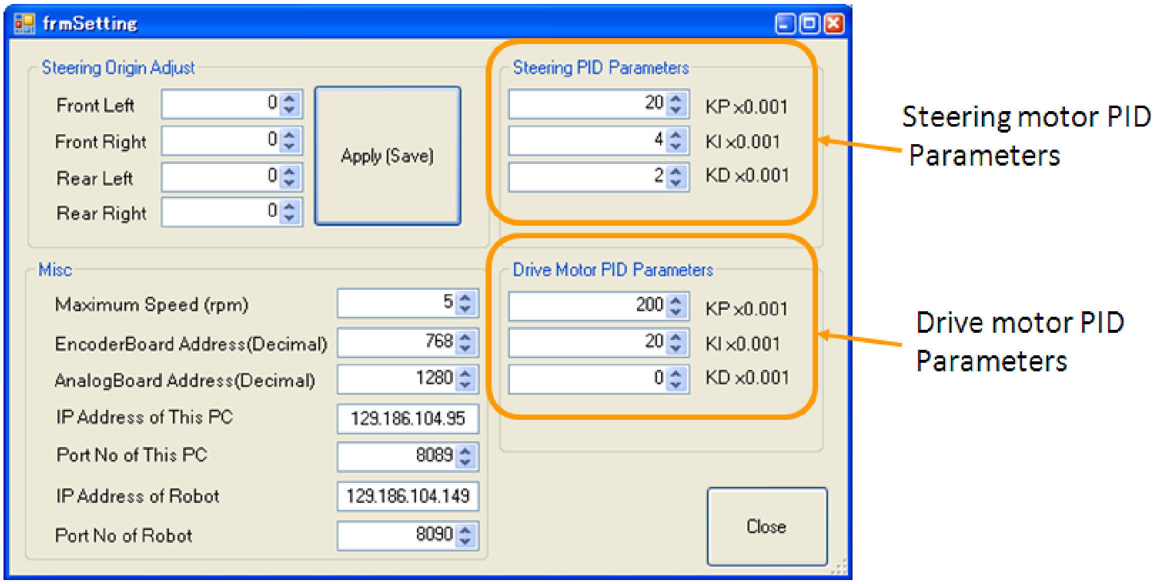
Height: 0
Brake: 0

Figure 2.18: *Remote operation and display window*

2.4.3 AgRover local controller

The hardware control of AgRover was managed on the central controller. The execution and feedback were both communicating through PC104 with the microcomputer. The control parameters were set by the operator on the remote operation program (illustrated on Figure 2.19).

By clicking the “Config” button on Figure 2.18, a setting window popped up (Figure 2.19). The PID parameters for both steering and driving motors could be adjusted respectively, and will be set to all the motors. The initial value of the steering angle for each wheel could be customized as well. It did have the flexibility for the user to choose which IP to go if there were multiple robots available.



frmSetting

Steering Origin Adjust

Front Left 0
Front Right 0
Rear Left 0
Rear Right 0

Misc

Maximum Speed (rpm) 5
EncoderBoard Address(Decimal) 768
AnalogBoard Address(Decimal) 1280
IP Address of This PC 129.186.104.95
Port No of This PC 8089
IP Address of Robot 129.186.104.149
Port No of Robot 8090

Steering PID Parameters

20 KP x0.001
4 KI x0.001
2 KD x0.001

Drive Motor PID Parameters

200 KP x0.001
20 KI x0.001
0 KD x0.001

__Figure 2.19:__ Remote operation and display window

2.4.4 Motors and encoders

After setting the control parameters, according to the orders from the remote station, the controller started manipulating eight motors and monitoring eight encoders.

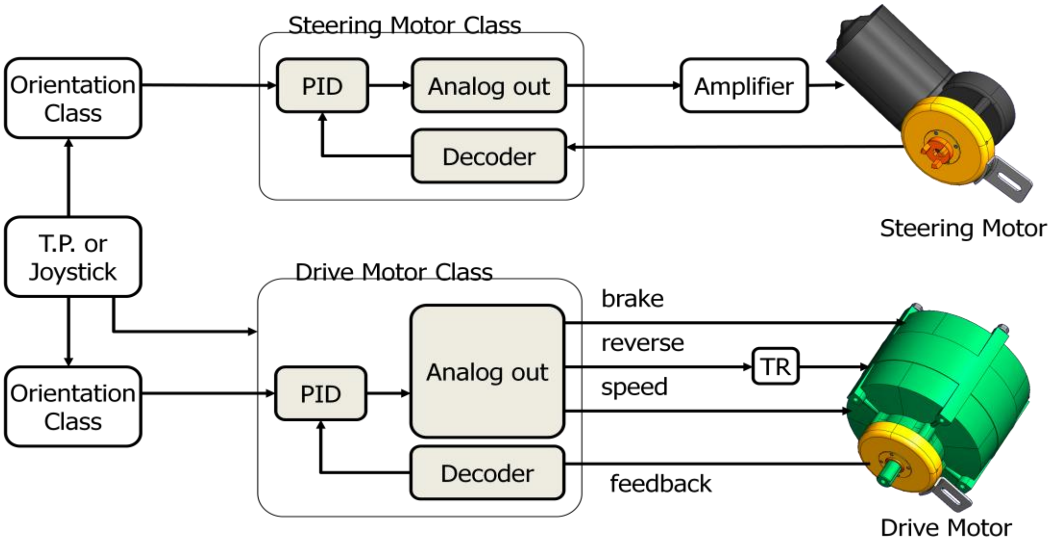


Figure 2.20: *Steering and driving motor control flow chart*

All steering and driving motors were controlled by PID controllers. And the control diagram is as Figure 2.21.

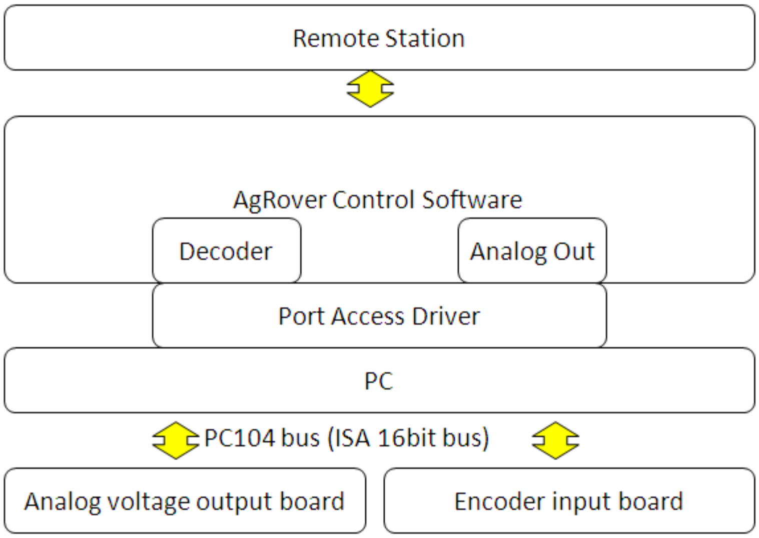


Figure 2.21: *AgRover controller program architecture*

2.4.5 Safety redundancy

For hardware safety concern, there are self-diagnostic rules defined on the AgRover controller that can help protect the robot itself.

Table 2.1: *Self diagnostic situations that may lead to emergency stop*

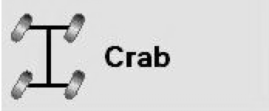
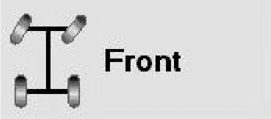
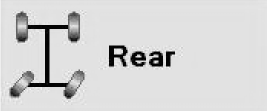
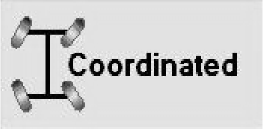
No.	Description	Reaction
1)	Steering motor or encoder failure	AgRover stop and alarm on
2)	Driving motor or encoder failure	AgRover stop and alarm on
3)	One or more wheels get stuck	AgRover stop and alarm on
4)	Battery power is low	AgRover stop and alarm on
5)	Wireless communication with PC is dropped	AgRover stop and alarm on

The main purpose of this self-checklist was to help control the robot properly, protect the hardware from electrical or mechanical damages, and also to remind the operator of manipulation mistakes.

2.5 Steering Modes

Because AgRover is a four-wheel steering robotic vehicle, it has four steering modes to select from (Table 2.2), which are crab steer, front wheel steer, rear wheel steer and coordinated steer.

Table 2.2: *Four steering modes of AgRover*

 Crab	All of four wheels have the same steering angle
 Front	Only two front wheels can be steered.
 Rear	Only two rear wheels can be steered.
 Coordinated	Each wheel has independent steering angle.

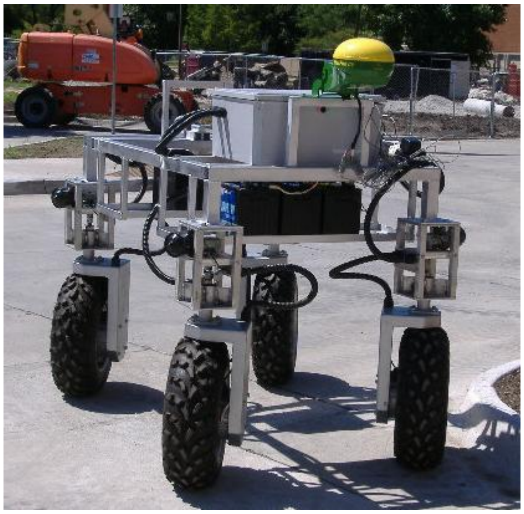


Figure 2.22: *Crabbing steering mode*

Crab steer can have the robot turn fast while the vehicle speed is low and there is no need to adjust the orientation.

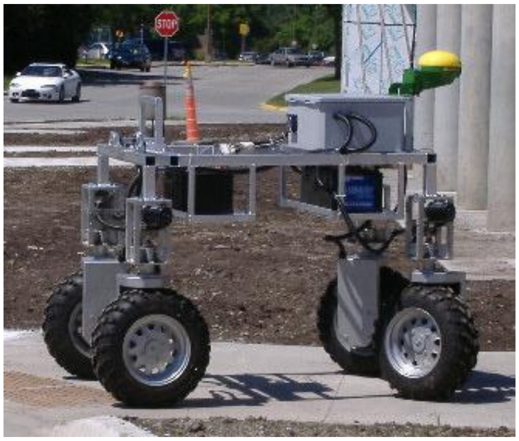


Figure 2.23: *Front steering mode*

Front wheel steer mode is similar like a conventional car. It showed promising performance on following a straight line trajectory, which will be discussed in Chapter 3.

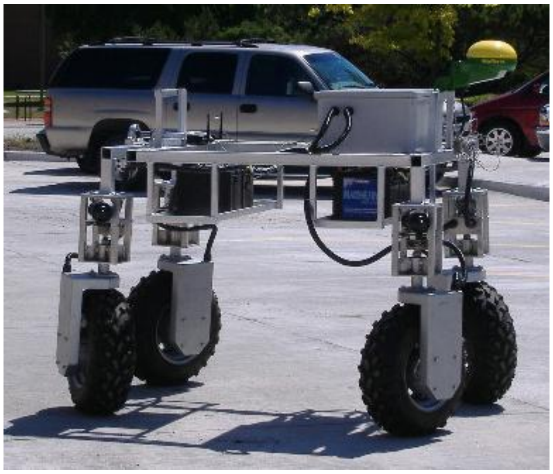


Figure 2.24: *Coordinated steering mode*

Coordinated steer has the most degrees of freedom, and it is the most flexible and agile way to turn the vehicle. Coordinated steering performed well in the curve trajectory tracking field experiments, which will be presented in Chapter 3.

2.6 Automatic Navigation Control

The manual operation is not enough for an autonomous robot, and automatic navigation control is desired. An RTK-GPS based automatic navigation control system was designed and integrated (Figure 2.25).

2.6.1 System integration

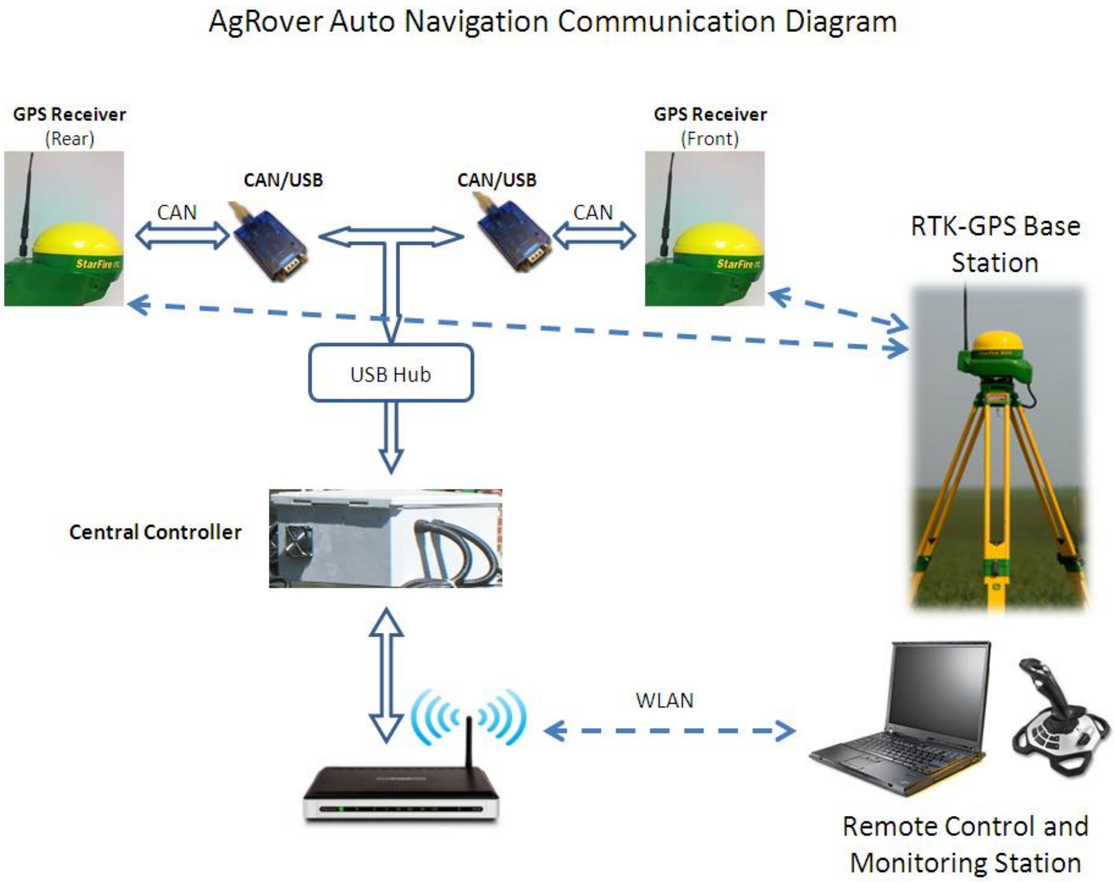


Figure 2.25: *Automatic navigation control system integration*

Two StarFire GPS receivers (JOHN DEERE®) were installed at front and rear end of the chassis respectively (Figure 2.5), and were both set working under real time kinematic (RTK) mode. By processing the GPS data from both of the receivers and cooperating with designated RTK-GPS base station, the location and orientation of AgRover could be determined.

2.6.2 Control structure

The central body controller was the “brain” for the whole navigation control system. It was taking care of all the motors and encoders, and monitoring the feedback from the GPS receivers.

The central controller could be divided into two levels, the upper level controller and the lower level controllers (Figure 2.26). The upper level controller read the given reference path information and translated it into the vehicle local coordinate system. By processing the GPS data from front and rear receivers, the upper level controller was able to tell the system the errors and generate control signals, which were distributed to individual lower level motor controllers. The lower level controllers are in charge of controlling steering and driving motors with PID control. The control loop for lower level controller is fast and time efficient, which guaranteed that the hardware would not fail even if the upper level processing and calculation took longer time interval.

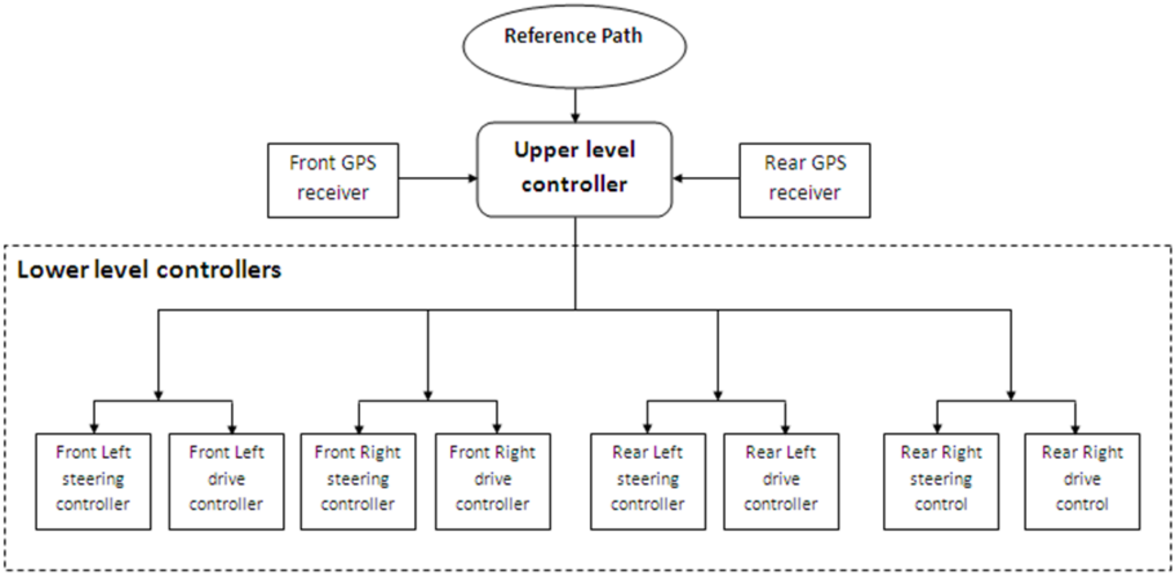

Reference Path
Front GPS receiver
Upper level controller
Rear GPS receiver
Lower level controllers
Front Left steering controller
Front Left drive controller
Front Right steering controller
Front Right drive controller
Rear Left steering controller
Rear Left drive controller
Rear Right steering control
Rear Right drive control

Figure 2.26: *Architecture of central controller*

The control algorithm applied in the upper level controller was sliding mode control (SMC), which was investigated theoretically in Chapter 3, and validated with field experiments in Chapter 4.

2.7 Conclusions

The development of AgRover was productive and inspiring. According to the off-road working capability requirements, AgRover Gen. I and Gen. II were designed and developed. The mechanic and electronic system design were presented and introduced. With this capable field working platform, further research activities of automatic navigation control could be conducted. As an autonomous mobile robot, AgRover is expectable to carry out field experiments, which were further investigated and studied in Chapter 4.

References

Bin Ahmad, M.T., (2012) Development of an automated mechanical intra-row weeder for vegetable crops, Master thesis, Iowa State Unversity.

Chen, H., Xi, N., Sheng, W., Song, M., Chen, Y., (2002) CAD-based automated robot trajectory planning for spray painting of free-form surfaces, Industrial Robot: An International Journal, Vol. 29, No. 5, pp. 426-433

Dong, F., Heinemann, W., Kasper, R., (2011) Development of a row guidance system for an autonomous robot for white asparagus harvesting, Computers and Electronics in Agriculture, Vol.79(2), pp.216-225

Edan, Y. ; Miles, G.E., (1993) Design of an agricultural robot for harvesting melons, Transactions Of The ASAE, Vol.36(2), pp.593-603

Edan, Y., Miles, G., (1994) Systems engineering of agricultural robot design, IEEE Transactions of Systems, Man and Cybemetics, Vol. 24(8), pp. 1259

Foglia, M., Reina, G., Corke, P., Sukkarieh, S., (2006) Agricultural robot for radicchio harvesting, Journal of Field Robotics, Vol. 23, pp. 363-377

Have, H., Nielsen, J., Blackmore, S., Theilby, F., (2005) Autonomous weeder for Christmas trees-basic development and tests, Danish Environmental Protection Agency, Pesticides Research No. 97

Jarvis, R., (2001) A tele-autonomous heavy duty robotic lawn mower, Proceeding Australian Conference on Robotics and Automation, Sydney, Nov. 14-15

Kataoka, T., Hiroma, T., Ota, Y., (1999) Development of a harvesting hand for apples, Advanced Robotics, Vol. 13(3), pp. 293-294

Kim, S.D., (2009) Design and Control of Autonomous Crop Tracking Robotic Weeder: Green Weeder, Master thesis, The University of New South Wales.

Liu, H., Zhang, L., Guo, X., Du, D., (2008) Research and application of robot technique in forestry, Automation and Logistics. ICAL. International conference on, pp. 2501-2505

Tan, X., Kim, D., Usher, N., Laboy, D., Jackson, J., Kapetanovic, a., Rapai, J., Sabadus, B., Zhou, X., (2006) An autonomous robotic fish for mobile sensing, Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct. 9-15, Beijing, China

Wang, Y., (2007) Robot-assisted sensor network deployment and data collection, Computational Intelligence in Robotics and Automation, pp. 467-472 Jun. 20-23

Zhang, S., Liu, W., Wang, Z., (2005) Research on spray-paint robot position control system based on BP neural network, robotics and Biomimetics, IEEE International Conference on, pp. 742-746

CHAPTER 3: BACKSTEPPING-BASED SLIDING MODE CONTROL FOR AN AGRICULTURAL ROBOTIC VEHICLE

A paper to be submitted to *Journal of Field Robotics*

Xuyong Tu, Lie Tang

3.1 Abstract

This paper introduces a new controller design based on backstepping and sliding mode control (SMC) techniques for an agricultural robotic vehicle. This agricultural robotic vehicle is four-wheel-steering (4WS) and four-wheel-drive (4WD), which has multiple degrees of freedom and nonlinearity. The navigation control of this vehicle needs to cope with multiple inputs (front and rear steering angles, acceleration) and multiple outputs (heading, velocity and location). To deal with this multiple-inputs and multiple-outputs (MIMO) nonlinear system, backstepping-based SMC approach was chosen in the controller design, which also met the Lyapunov stability requirement. The performance of the developed navigational controller was simulated in Matlab Simulink. Simulation results are presented; and the dynamic performance with parameter uncertainty and external disturbances is analyzed.

Keywords: backstepping, sliding mode control, MIMO, path following, agricultural robotic vehicle

3.2 Introduction

Motion control for agricultural robotic vehicle has been an active research area in recent years (Hargas, et al 2002; Kondo, 2005). With the development of robotic technologies, autonomous agricultural robots are expected to relieve farm labour from having to do monotonous, unproductive and uncreative work, such as weeding (Watanabe, 2004), sampling (Nejati, et al, 2008) and harvesting (Katupitiya, 2005). Like other nonholonomic nonlinear systems, agricultural robotic vehicles have system uncertainties and time-varying parameters, especially when working in off-road environment. In addition, external factors such as soil and wind conditions also affect vehicle dynamic characteristics. Both unpredictable internal perturbations and external disturbances create a great challenge of controlling such nonholonomic nonlinear system.

Four-wheel-steering (4WS) provides improved manoeuvrability at low speed when compared with conventional vehicles with front wheel steering only. Four-wheel-drive (4WD) improves the traction control capability of the vehicle when coping with variable soil surfaces. The prototype 4WD/4WS vehicle, AgRover, is constructed with a rigid frame with four identical wheels. The platform has adjustable high clearance to accommodate field conditions where crops are presented with variable heights. The robotic vehicle has multiple inputs (acceleration, front steering angle, rear steering angle) and multiple outputs (heading, velocity and location), which constitute a typical nonholonomic multiple-input and multiple-output (MIMO) high-order nonlinear system.

Imprecision in modelling a nonlinear system may come from the actual uncertainty of the plant or from the purposeful choice of a simplified representation of the system dynamics. The method of backstepping decomposes complex nonlinear systems to lower dimension subsystems that are designed to make the whole system perform as desired by using Lyapunov function and pseudo-control variables. Compared with feedback linearization method (Kanayama, et al, 1990; Kim and Oh, 1999; Wang, 2005), backstepping does not require accurate models while still maintains system properties and nonlinearity. Sliding mode control (SMC) of variable structure control (VSC) has strong robustness to parameter perturbation and external disturbances (Cheng, et al, 2007). In contrast to fuzzy control (Hagras, et al 1999) and adaptive control (Dixon, et al, 2004), SMC is more robust and fast, which is suitable and responsive under off-road environment. Combining backstepping and sliding mode control has demonstrated good performance for both linear and nonlinear systems (Lin, Shen, 2002; Shen, Lin, 2005).

In this paper, a backstepping-based SMC controller for path-following control for AgRover, a small prototype 4WS/4WD agricultural robotic vehicle, is introduced. Vehicle dynamic modelling, procedure of backstepping-based SMC are presented. To verify the robustness of the developed navigational controller, paths of a step perturbation and high-frequency parameter and external disturbances are adopted during the simulation process.

3.3 4WS System Introduction

3.3.1 Introduction to AgRover Platform

Dynamic modeling and simulation are useful in evaluating the performance of controllers for robotic vehicles. This section presents the modeling of AgRover, which is a 4WD/4WS mobile research platform, with the integrated pneumatic suspension and self-leveling system (Figure 3.1).

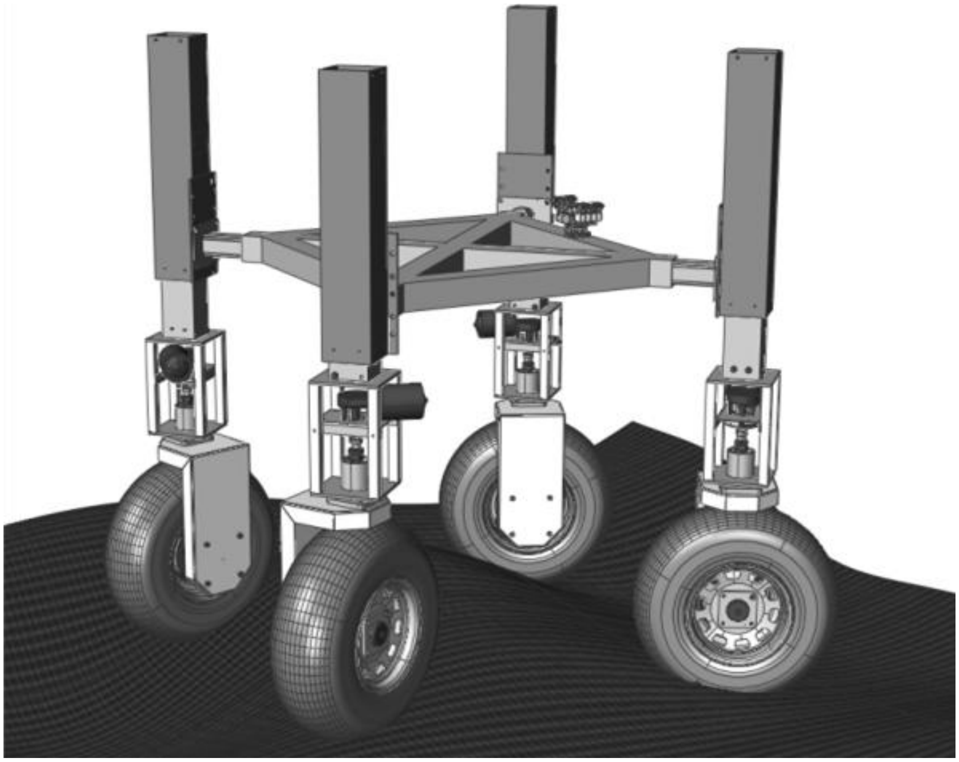


Figure 3.1: *The overall structure of AgRover*

Each wheel leg has one pneumatic cylinder and one square shape sliding tube. One of the four cylinders has position feedback for platform leveling and clearance control. Some other features of the vehicle are: (1) adjustable vehicle height for providing needed ground clearance at different crop growth stages; (2) adjustable width to variable row width; (3) capability to rough terrain; and (4) high degree of manoeuvrability.

The AgRover consists of five subsystems, including four independent wheel assemblies and a central body. The body subsystem has all basic vehicle parameters (e.g. corner stiffness) which define the characteristics of AgRover. The central controller gathers the location update of central gravity (CG) from GPS receiver and heading information from gyroscope. The model was setup in Matlab Simulink.

The following table (Table 3.1) shows all the system parameters affecting the performance of AgRover.

Table 3.1: *Nomenclature*

Symbol	Description	Value and Unit
M	Mass of AgRover	*270 kg*
a	Distance from CG to front axle	*0.8 m*
b	Distance from CG to rear axle	*0.8 m*
I_{zz}	Turning moment of inertia	*157.27* Nms^2/rad
$C_{\alpha f}$	Front tire corner stiffness	*765.79 N/rad*
$C_{\alpha r}$	Rear tire corner stiffness	765.79 N/rad
μ	Friction coefficient between tires and ground	0.45
u	Longitudinal velocity	*m/s*
v	Lateral velocity	*m/s*
r	Yaw rate	*rad/s*
Ψ	Heading angle	*rad*
δ_f	Front wheel steering angle	*rad*
δ_r	Rear wheel steering angle	*rad*
α	Side slip angle	*rad*
X, Y	Global Coordinates	*m*
F_{yf}	Front lateral force	*N*
F_{yr}	Rear lateral force	*N*
N_f	Front load force	*N*
N_r	Rear load force	*N*
Acc	Longitudinal acceleration	m/s^2

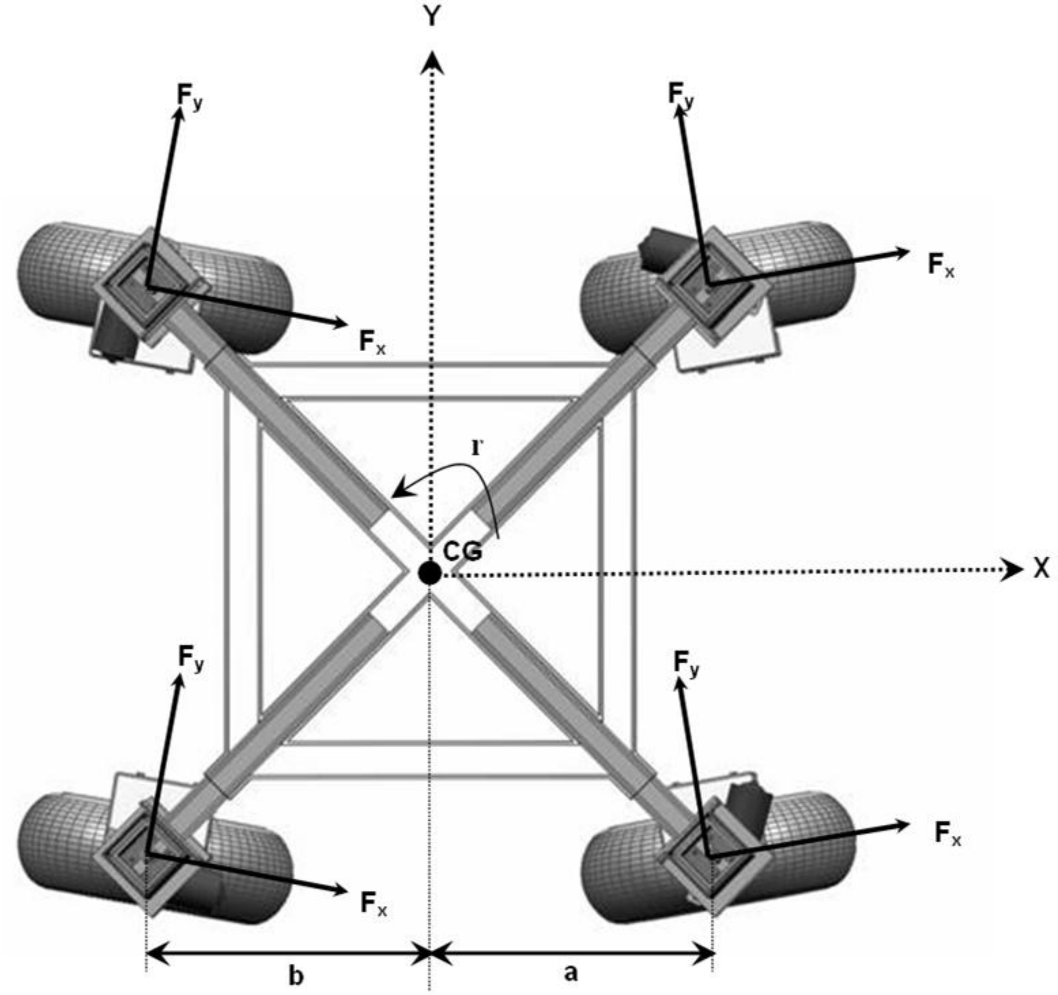
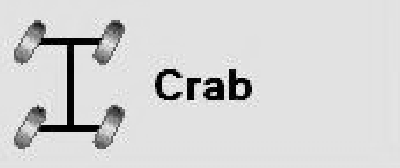
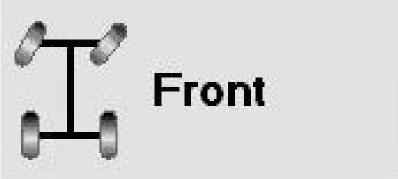
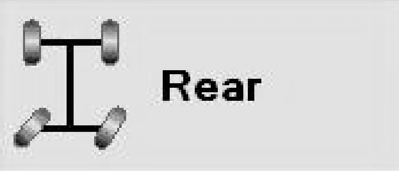
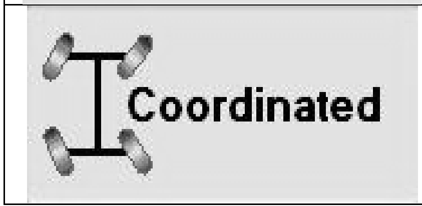


Figure 3.2: *Illustration of planar motions and forces of AgRover*

The force on each wheel can be decomposed into two sub-forces, F_x and F_y. Yaw rate “r” is the rotation rate of the whole vehicle (Figure 3.2).

Since AgRover is a four-wheel-steering (4WS) robotic vehicle, there are four steering modes available. Among the four steering modes listed in the Table 3.2, coordinated steering has the highest flexibility, under which each wheel can be individually controlled. In this study, coordinated steering mode is selected. To simplify the system model, the two front wheels are assumed to have the same steering angle. The same assumption is made for the two rear wheels.

Table 3.2: *List of four steering modes*

	All of four wheels have the same steering angle
	Only two front wheels can be steered.
	Only two rear wheels can be steered.
	Each wheel has individual steering angle and aiming to the same turning center.

The turning centre of coordinated steering is constrained to be on the central line paralleled with front and rear axle, where the centre of the gravity is. In other words, under this particular coordinated steering mode, though each wheel has independent steering angle, the front and rear steering angles have the same magnitude but to opposite ways of direction.

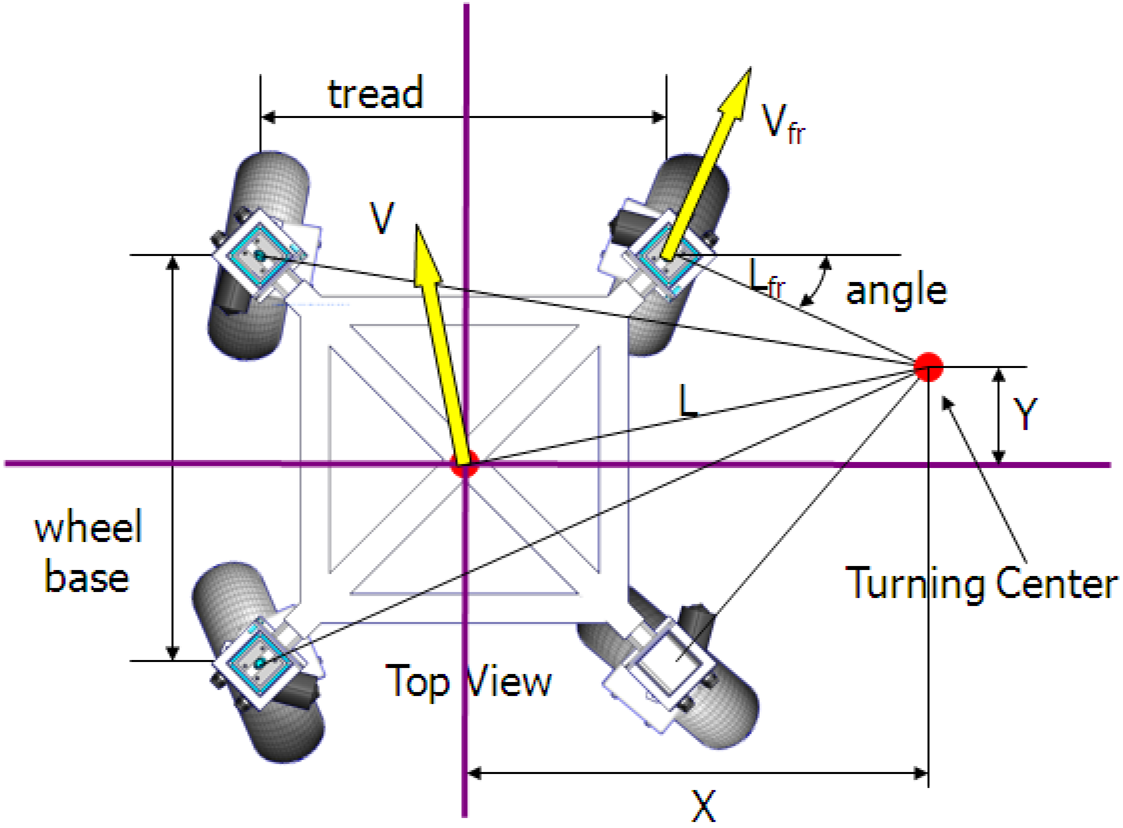


Figure 3.3: *Free way coordinated steering mode*

As displayed in Figure 3.3, the freeway steering mode, which belongs to coordinated steering mode, is one of the most flexible steering modes for this 4WS and 4WD platform. The turning center is not constrained on the central line, but can be flexible on the 2-D surface. Though it has not been adopted and studied in this paper, but can be a potential option for some difficult steering circumstances, such as headland turning within limited area. It should present capability to solve such issues.

3.4 System Modelling

3.4.1 Vehicle kinematic model

Vehicle kinematic model is to be studied specifically in Chapter 4, so the discussion in this section will be focused on vehicle dynamic model.

3.4.2 Vehicle dynamic model

At this research stage, roll and pitch factors such as load shift are not included in this paper. There are six variables including heading angle (ψ), location X and Y , yaw rate (r), longitudinal velocity (u) and lateral velocity (v) were chosen as the state variables to build the model. The relationships among them are presented below:

State Variables:

$$x = [\psi \quad X \quad Y \quad r \quad u \quad v]^T = [x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6]^T \tag{3.1}$$

X, Y, and ψ are chosen as system output :

$$y = [\psi \quad X \quad Y]^T = [x_1 \quad x_2 \quad x_3]^T \tag{3.2}$$

Input (control signal):

$$Control = [A_{cc} \quad \delta_f \quad \delta_r]^T \tag{3.3}$$

Because the vehicle kinematic control should be based on vehicle-fixed coordinate system, to unify the relationship equations, coordinates rotation and translation are necessary (Chen, et al, 2009).

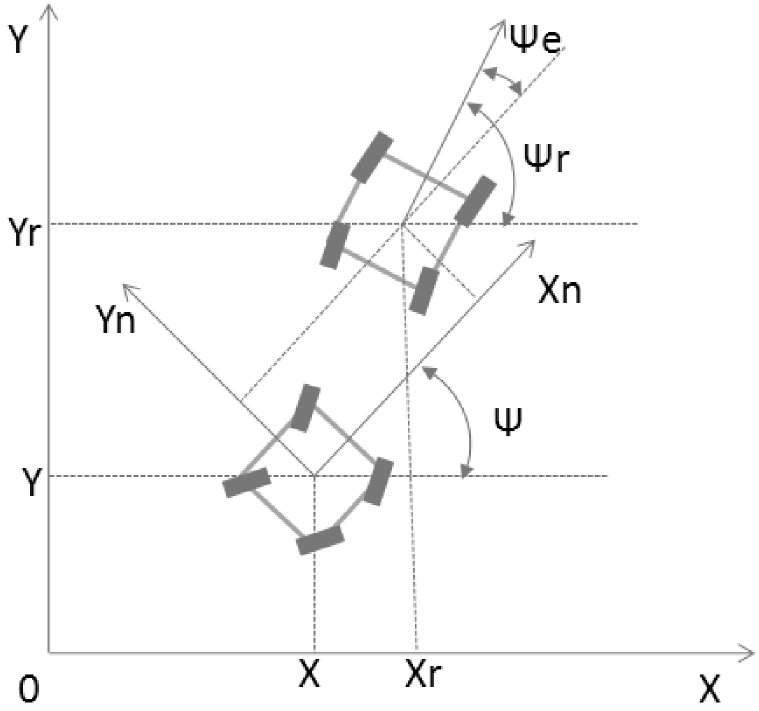


Figure 3.4, *Illustration of the global coordinate system and the vehicle-fixed coordinate system.*

As illustrated in Figure 3.3, X and Y are current global coordinates, and X_n and Y_n are vehicle coordinates. X_r and Y_r are the target location of the robotic vehicle in global coordinates. In this case, to simplify the relationship equations between different state vectors, the global coordinates should be converted into vehicle-fixed coordinates, which is a time-varying coordinate system associated with heading angle.

As indicated in Figure 3.3, coordinate rotation matrix is:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix} \tag{3.4}$$

Define pseudo variables:

$$q_1 = x_n = \begin{bmatrix} x_{1n} \\ x_{2n} \\ x_{3n} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = T * y \tag{3.5}$$

and $q_2 = [x_4 \quad x_5 \quad x_6]^T$ (3.6)

Then the state equations can be described as :

$$\begin{cases} \dot{q}_1 = q_2 \\ \dot{q}_2 = \begin{bmatrix} \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \begin{bmatrix} \frac{F_{yf}a - F_{yr}b}{I_{zz}} \\ A_{cc} \\ \frac{-Mur + F_{yf} + F_{yr}}{M} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -ur \end{bmatrix} + \begin{bmatrix} 0 & \frac{a}{I_{zz}} & -\frac{b}{I_{zz}} \\ 1 & 0 & 0 \\ 0 & \frac{1}{M} & \frac{1}{M} \end{bmatrix} \begin{bmatrix} A_{cc} \\ F_{yf} \\ F_{yr} \end{bmatrix} + \rho \end{cases} \quad (3.7)$$

Where ρ represents system uncertainty.

The system state can be described as:

$$\begin{cases} \dot{q}_1 = q_2 \\ \dot{q}_2 = A(q_1, q_2) + BU \end{cases} \quad (3.8)$$

$$A(q_1, q_2) = [0 \quad 0 \quad -ur]^T \quad (3.9)$$

$$B = \begin{bmatrix} 0 & \frac{a}{I_{zz}} & -\frac{b}{I_{zz}} \\ 1 & 0 & 0 \\ 0 & \frac{1}{M} & \frac{1}{M} \end{bmatrix} \quad (3.10)$$

$$U = [A_{cc} \quad F_{yf} \quad F_{yr}]^T \quad (3.11)$$

As defined, the control signals are:

$Control = [A_{cc} \quad \delta_f \quad \delta_r]^T$.

Based on vehicle dynamics theory,

if $|F_{yf}| < \frac{\mu N_f}{2}$, then $F_{yf} = -C_{\alpha f} \tan \alpha_f$. (3.12)

Otherwise, if$|F_{yf}| \geq \frac{\mu N_f}{2}$, then

$$F_{yf} = -\mu N_f sgn(\alpha_f) \left[1 - \frac{\mu N_f}{4C_{\alpha f} \tan \alpha_f}\right]. \quad (3.13)$$

F_{yf} is the lateral force of the front tires. Similarly F_{yr} is the lateral force of the rear tires. The lateral force is related to the side slip angle α and steering angle δ. The relationship is:

$$\alpha_f = \frac{v+ar}{u} - \delta_f,\ \alpha_r = \frac{v-br}{u} - \delta_r\ . \tag{3.14}$$

From the equations above, the control signals given in (3.3) can be achieved.

In this case, CG is assumed to be at the geometric centre. It is also assumed that the load of the front and rear tires satisfy the equation$N_f = N_r$. On the other hand, since the vehicle speed in this simulation is considered to be slow, load shift from side to side while turning is not considered.

3.5 Methods

3.5.1 Sliding Mode Control

The sliding mode control (SMC) approach, of variable structure control (VSC), is recognized as an efficient tool to design robust controllers for complex high-order nonlinear dynamic plant operating under uncertain conditions (Sabanovic, et al., 2004). It was initiated in the former Soviet Union about forty years ago. SMC methodology has subsequently received much attention from the international control community within the last two decades. The major advantage of SMC is low sensitivity to plant parameter variations and disturbances which eliminates the necessity of exact modeling.

3.5.1.1 The concept of a "sliding mode"

The "sliding mode" phenomenon may appear in dynamic systems governed by ordinary differential equations with discontinuous state functions in the right-hand sides (Sabanovic, et al., 2004).

Define $S(x)$ a sliding manifold $S(x) = [s_1(x),\ s_2(x), \dots s_m(x)]^T$ (3.15)

Where m is determined by the dimension of the system control inputs.

The domain of the sliding mode is $s_i(x) = 0, i = 1 \ldots m$ (3.16)

3.5.1.2 Design procedure

The design procedure of SMC consists of two independent sub problems of lower dimensions (Sabanovic, et al., 2004):

- Design of the desired dynamics for the system of the $(n - m)$th order by proper choice of a sliding manifold $S = 0$, where n is the dimension of the system.
- Enforce sliding motion in this manifold which is equivalent to a stability problem of the mth order system. The switching function $\dot{s}$ was designed to make sure $s\dot{s} \leq 0$.

For the second step, Lyapunov stability theory is usually used to solve this problem. The defined Lyapunov function $V(s)$ must be ensured as positive definite to maintain the system stability, and it is also the control target.

3.5.2 Backstepping Technique

In control theory, backstepping is a technique for designing stabilizing controls for a special class of nonlinear dynamic systems (Sabanovic, et al., 2004). This recursive control technique decomposes a higher-dimensional system to several lower-dimensional subsystems. Because of the recursive structure, the designer can start the process at the known stable system and progressively stabilize each outer subsystem.

The advantage of the combining SMC and backstepping methodology is that the controller has the merits from both of them and is able to provide the vehicle system stability without exact system model. Backstepping downgrades the MIMO system model to a lower dimensional virtual model and SMC provides robust control with uncertainties and disturbances.

3.5.3 Backstepping SMC Design

For this AgRover path following control application, since the system is a high dimensional MIMO nonlinear system with some varying system factors and parameters, conventional control methods such as PID control are not applicable. Backstepping-based sliding mode control offers a promising control strategy. The design work contains two backstepping steps (Liu, 2005):

3.5.3.1 Step I:

Define state error $\mathrm{z}_1 = \mathrm{q}_1 - \mathrm{x}_\mathrm{r}$, where x_r is the reference value of the state of vehicle.

$$\dot{z}_1 = \dot{q}_1 - \dot{x}_r = q_2 - \dot{x}_r \tag{3.17}$$

Define $\alpha = c_1 z_1$, where $c_1 > 0$ is a positive constant.

Define $z_2 = \dot{z}_1 + \alpha$, whose derivative is

$$\dot{z}_2 = \dot{q}_2 - \dot{\alpha}_1 = A(q_1, q_2) + BU + \rho + c_1\dot{z}_1 - \ddot{x}_r \tag{3.18}$$

And Lyapunov function is chosen as:

$$V_1 = \frac{1}{2} z_1^2, \tag{3.19}$$

Then

$$\dot{V}_1 = z_1\dot{z}_1 = z_1(q_2 - \dot{x}_r) = \mathrm{z}_1(\mathrm{z}_2 - \alpha) = \mathrm{z}_1\mathrm{z}_2 - \mathrm{c}_1\mathrm{z}_1^2 \tag{3.20}$$

To make sure $\dot{V}_1 \leq 0$, i.e., to ensure the system is stable and converges to $z_1 = 0$, $z_2 = 0$ is required.

3.5.3.2 Step II:

Define SMC switch function $\mathrm{s} = \mathrm{kz}_1 + \mathrm{z}_2, \mathrm{k} > 0.$ (3.21)

Then define Lyapunov function**:** $\mathrm{V}_2 = \mathrm{V}_1 + \frac{1}{2}\mathrm{s}^2$ (3.22).

Because of the positive definiteness of V_2 is required, the control law U can be designed:

$$U = B^{-1}\left[-k(z_2 - c_1 z_1) - A(q_1, q_2) - \rho sgn(s) - \dot{\alpha} + \ddot{x}_r - h\big(s + \varepsilon sgn(s)\big)\right] \tag{3.23}$$

where the control parameters $h, \varepsilon > 0$.

The control law design in (3.20) is not unique because, besides $\dot{s} = h\big(s + \varepsilon sgn(s)\big)$, other functions, such as saturation function, relay function etc., could be selected to replace sign function to realize the "sliding mode" of $s = 0$.

$$\dot{V}_2 = \dot{V}_1 + s\dot{s}$$

$$= -c_1 z_1^2 + z_1 z_2 + s[k(z_2 - c_1 z_1) + A(q_1, q_2) + BU + \rho + c_1 \dot{z}_1 - \ddot{x}_r]$$

$$= z_1 z_2 - c_1 z_1^2 - hs^2 - h\varepsilon|s| + \rho s - \bar{\rho}|s|$$

$$\leq z_1 z_2 - c_1 z_1^2 - hs^2 - h\varepsilon|s| + |\rho|(|s| - s)$$

$$\leq z_1 z_2 - c_1 z_1^2 - hs^2 - h\varepsilon|s| \qquad (3.24).$$

Define $Q = \begin{bmatrix} c_1 + hk^2 & hk - \frac{1}{2} \\ hk - \frac{1}{2} & h \end{bmatrix}$, and $z = [z_1 \quad z_2]^T$.

$$z^T Q z = [z_1 \quad z_2] \begin{bmatrix} c_1 + hk^2 & hk - \dfrac{1}{2} \\ hk - \dfrac{1}{2} & h \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$$= c_1 z_1^2 + hk^2 z_1^2 + 2hk z_1 z_2 - z_1 z_2 + h z_2^2$$

$$= c_1 z_1^2 - z_1 z_2 + hs^2, \tag{3.25}$$

Then $\dot{V}_2 \leq -z^T Q z - h\varepsilon|s|$ (3.26)

As long as $|Q| > 0$ (positive definite),$\dot{V}_2 \leq 0$, then the system is convergent. Thus h, c_1 and k should satisfy:

$$|Q| = h(c_1 + hk^2) - (hk - \frac{1}{2})^2 = h(c_1 + k) - \frac{1}{4} > 0 \tag{3.27}$$

As long as the control parameters meet the in-equation requirement above, the system is asymptotically stable.

3.6 Simulation Results

The AgRover modelling and control simulation was done by Matlab Simulink. In this simulation experiment, there are three input signals: acceleration, front steering angle, and rear steering angle and three output signals: heading angle, X-coordinate, and Y-coordinate. Any complex path can be decomposed into lines and curves, thus the path-following simulation comprises three types of reference paths: straight line, circle and inconsistent path.

3.6.1Tracking a straight line path

The reference path is $y = x$ with the velocity of 1.41 m/s, so the reference path function is:

$$[\psi \quad X \quad Y]^T = \left[\frac{\pi}{4} \quad t \quad t\right]^T \tag{3.28}$$

The original vehicle location is $[4, -1]$.

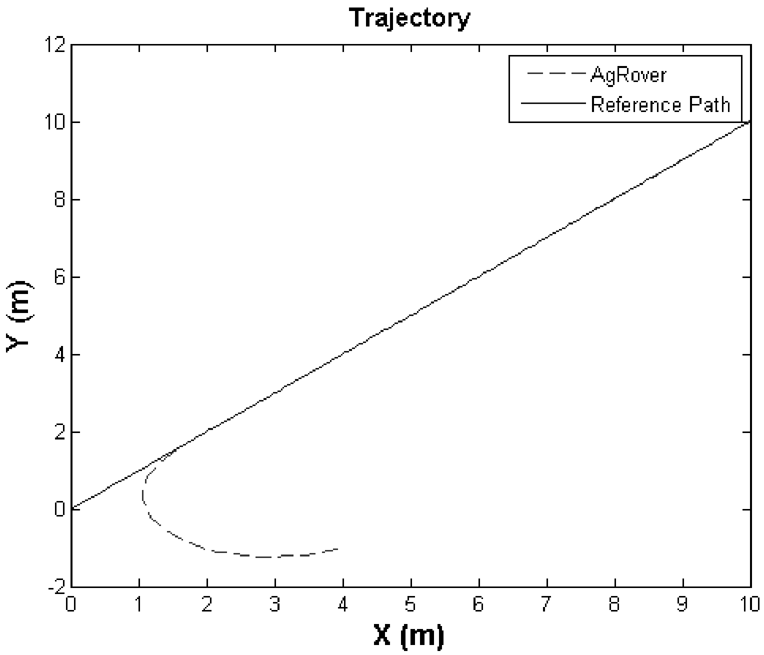


Figure 3.5: *Vehicle trajectory when tracking a straight line path*

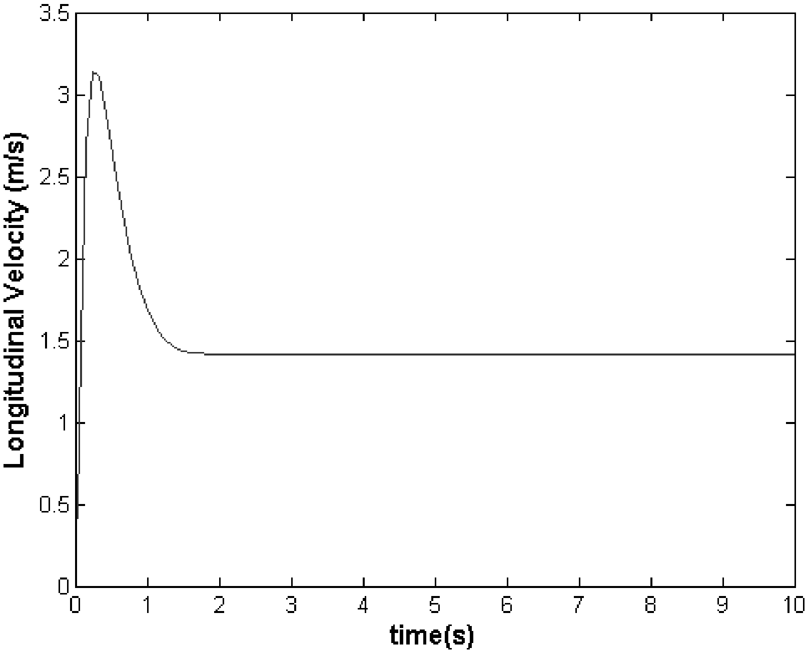


Figure 3.6: *Longitudinal velocity change when tracking a straight line path*

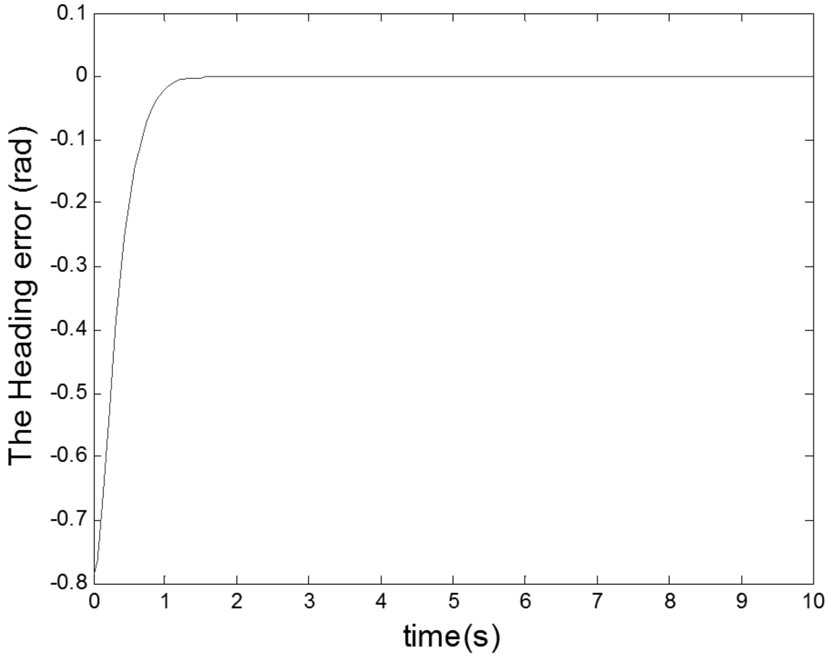


Figure 3.7: *Heading error when tracking a straight line path*

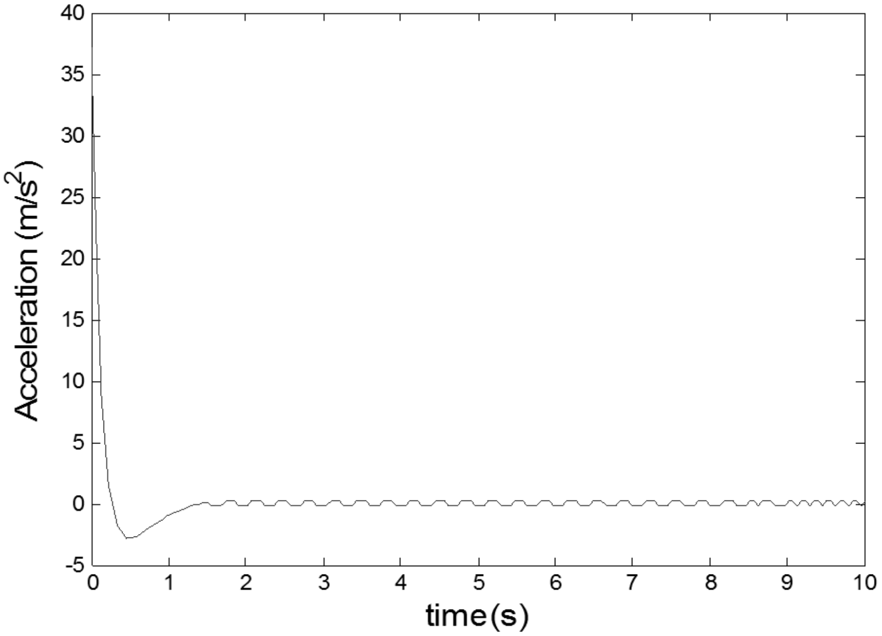


Figure 3.8: *Vehicle acceleration during the straight line path approaching and following processes*

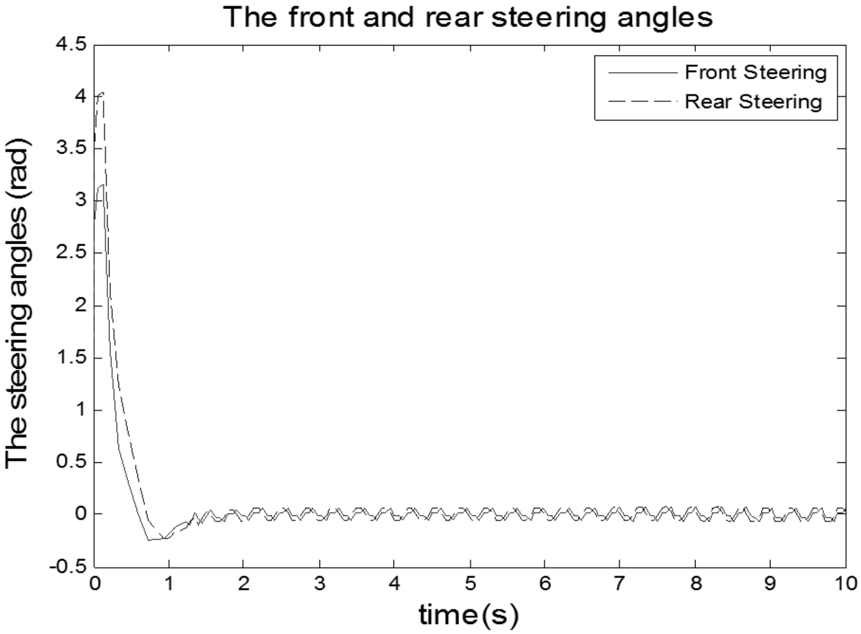


Figure 3.9: *Front and rear steering angles during the straight line path approaching and following processes*

From simulation results Figure 3.5-3.9, it is apparent that AgRover can approach a straight line reference path effectively and accurately.

3.6.2 Tracking an inconsistent path

In this experiment, the reference path has a step change from $y = 5$ to $y = 10$ at 10 sec. The original location is at $[4, -1]$ and the original heading is $-\pi$.

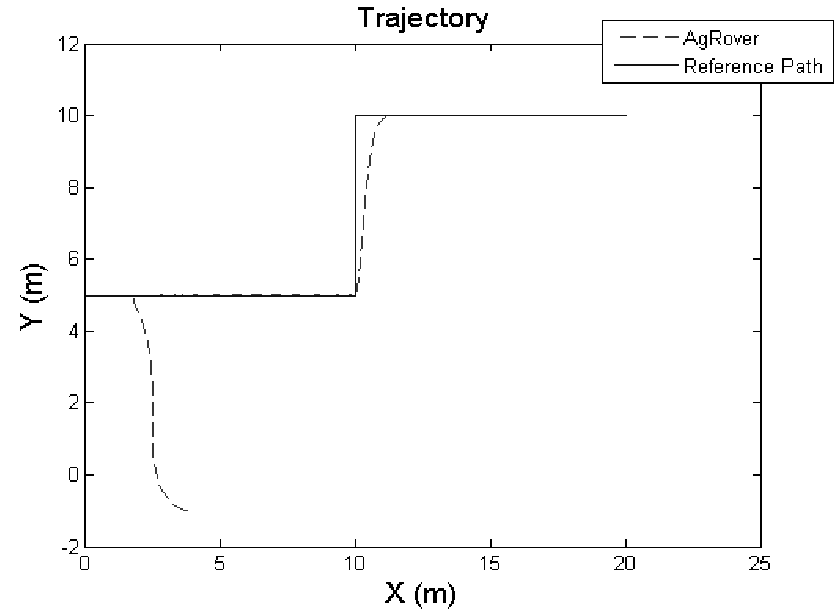


Figure 3.10: *Vehicle trajectory when tracking an inconsistent path*

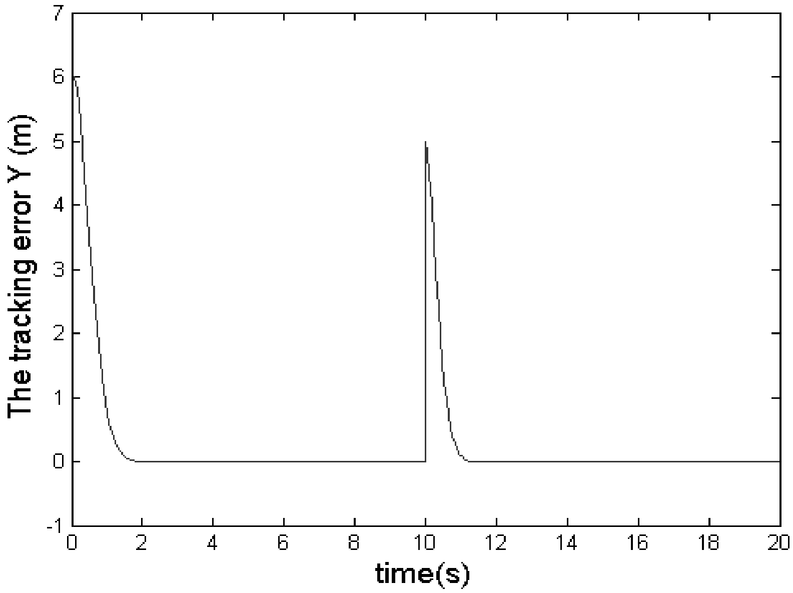


Figure 3.11: *Error in Y-coordinate when tracking an inconsistent path*

The tracking trajectory is presented in Fig. 10. When observing the tracking error in Y-coordinate (Figure 3.11), it is obvious that there was a pulse of the tracking error at 10 sec but the vehicle adjusted and corrected itself quickly.

3.6.3 Track a circle reference path

The origin of the circular path is located at [3, 1] with a radius of 1m.

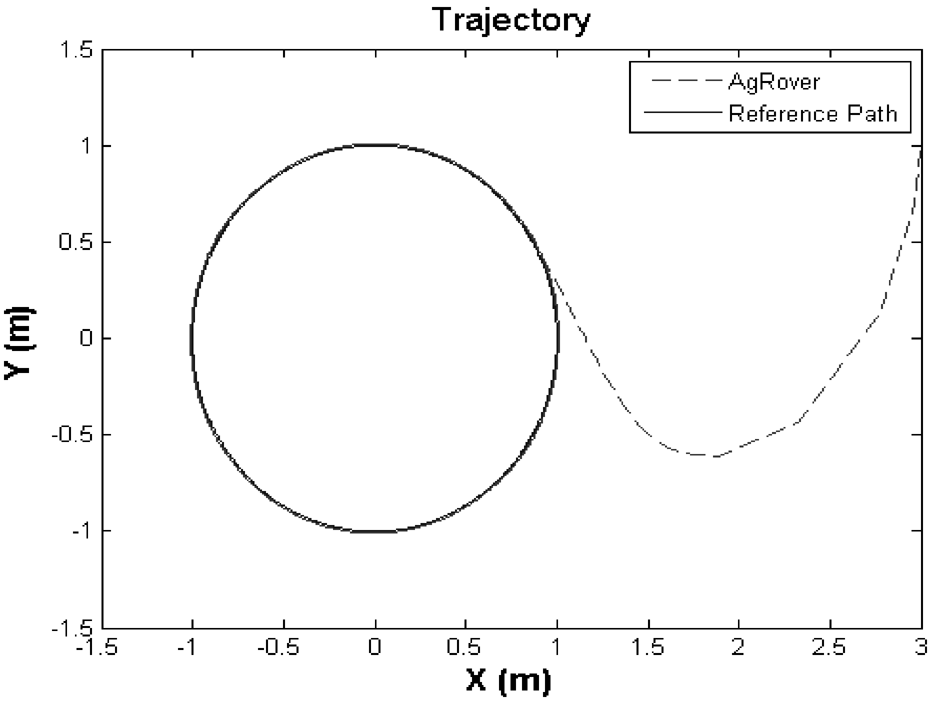


Figure 3.12: *Vehicle trajectory when tracking a circular path*

It is noticeable that AgRover does not simply “reach” the reference circle (Figure 3.12). It is adjusting the heading during approaching, which effectively avoids any large overshoot during the tracking process.

3.6.4 Robustness Test

3.6.4.1 Step perturbation test

To test the robustness of the controller, we intentionally changed the vehicle model parameters to examine the performance of the proposed controller when tracking an inconsistent path same as that given in section *3.6.2*.

Assume parameter perturbation happened at 5sec as follows:

Vehicle mass (M): 270 → 540 kg

Turning moment of inertia (Izz): 157 → 100 Nms^2/rad

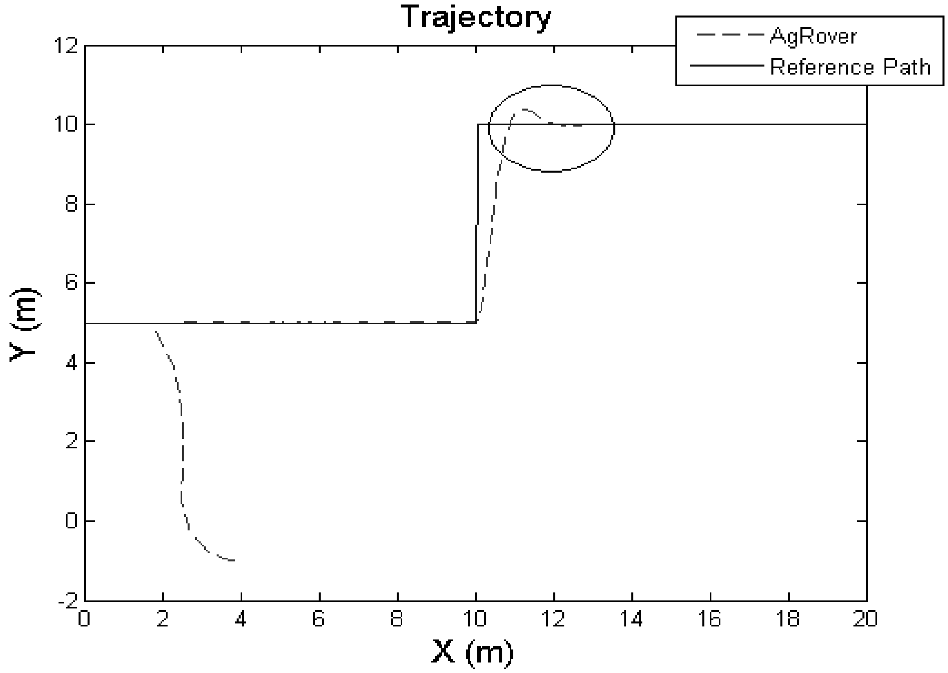


Figure 3.13: *Vehicle trajectory when tracking an inconsistent path during parameter perturbation test with mass and turning moment of inertia changed.*

The system parameter change does not affect the performance of AgRover along the original reference path. However, it causes an apparent overshoot (marked with an oval) at where the step change of the reference path occurs. The controller adjusts responsively and eliminates the error within 2 seconds.

Under off-road circumstances, friction coefficient cannot maintain as a constant, but a time-varying variable, which affects the cornering stiffness. For an off-road vehicle controller, it should have capability of overcoming cornering stiffness perturbation. Another robustness test was done by adding a perturbation of $0.2\sin(50t)$ on the cornering stiffness $C_{\alpha f}$, $C_{\alpha r}$, which simulates the circumstance that AgRover is working with variable friction coefficient:

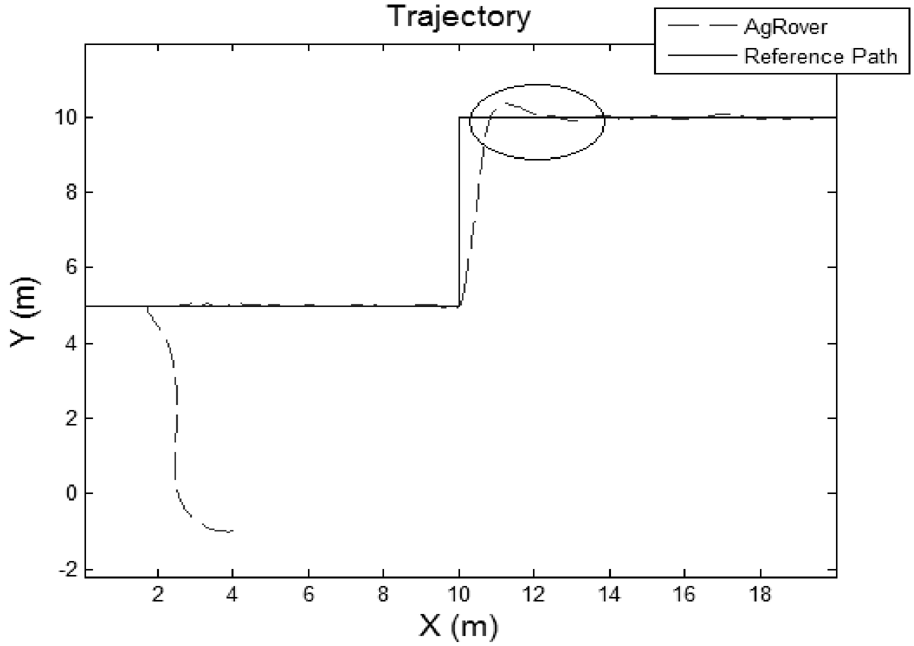


Figure 3.14: *Vehicle trajectory during parameter perturbation test with cornering stiffness changed.*

From Figure 3.14 it can be observed that the disturbance from corner stiffness does not significantly affect the system tracking performance. When compared with the result in Figure 3.10, where no disturbance is introduced when tracking an inconsistent path, this sinusoidal perturbation in the cornering stiffness causes an overshoot at 11 sec (marked by an oval). Similar to the result presented in Figure 3.14, this overshooting error is corrected quickly within 2 seconds, but a small ripple type tracking error persists due to the sinusoidal characteristics of the disturbance in cornering stiffness.

3.6.4.2 High frequency disturbance test

To test the robustness to high frequency disturbance on the feedback, a feedback disturbance 0.5 $sin(\omega t)$ was added to both X-coordinate and Y-coordinate signals, where $\omega = 1000\pi\ rad/s$ and the following results (Figure 3.15, 3.16) have been achieved:

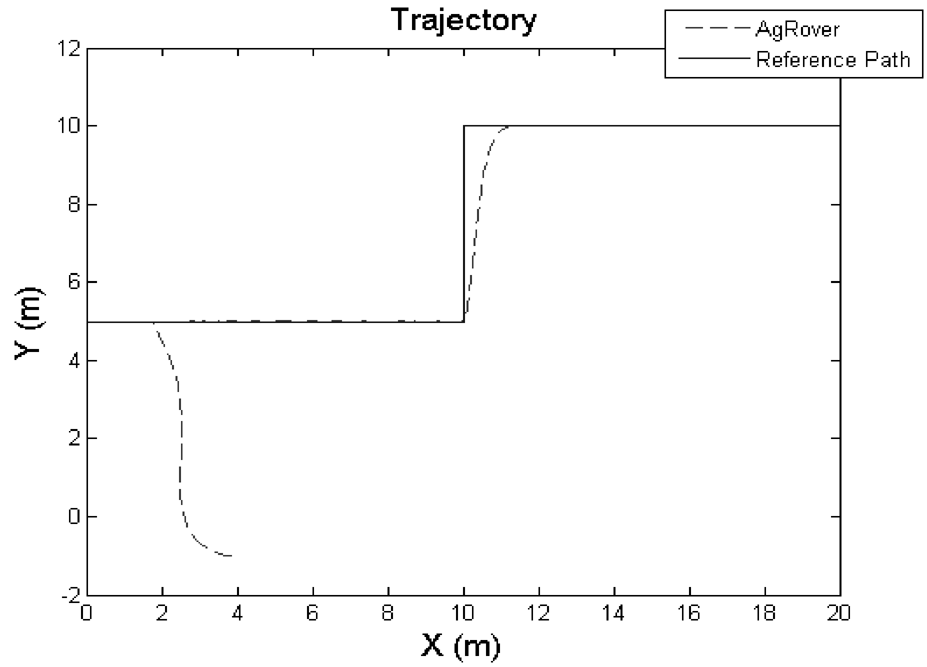


Figure 3.15: *Vehicle trajectory when tracking an inconsistent path with high frequency disturbance in feedback.*

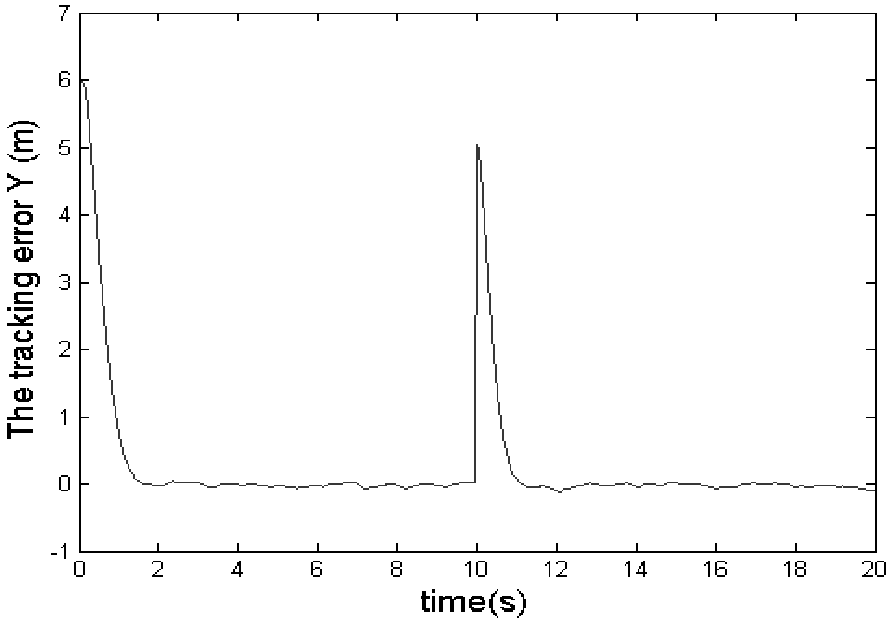


Figure 3.16: *Tracking error in Y-coordinate.*

The results show that though the feedback signal is contaminated with external disturbance, the system sustains well. Figure 3.15 and Figure 3.16 depict that the backstepping-based SMC controller has great robustness to external disturbance. The influence from disturbance-caused “chattering” was observed but with a small magnitude (Figure 3.16).

3.7 Conclusion

The results from different path following simulation tests have revealed that backstepping-based SMC is an effective controller design method for the path following control of AgRover, which is a nonholonomic MIMO high-order nonlinear system. The proposed basckstepping-based SMC control does not require precise plant model and is robust to system uncertainties and external disturbances. Under various operational circumstances, the backstepping-based SMC demonstrated a great effectiveness and accuracy in navigational control. Through dividing the control process into two backstepping stages, the reference paths of various shapes can be approached correctly and accurately. This work has built a valuable foundation for controlling the real hardware system, which will be the next step of this research. Besides, the system complexity can be increased to include roll and pitch factors and the capability of the controller will be further examined.

References

Chen, C., Li, T., Yeh, Y., Chang, C. (2009) Design and implementation of an adaptive sliding-mode dynamic controller for wheeled mobile robots. Mechatronics; 19(2): 156-166.

Cheng, J., Yi, J., Zhao, D., (2007) Design of a sliding mode controller for a trajectory tracking problem of marine vessels. Control Theory & Applications; 1(1): 233-237.

Dixon, W., De Queiroz, M., Dawson, D., Flynn, T., (2004) Adaptive tracking and regulation of a wheeled mobile robot with controller/update law modularity. IEEE Transactions on Control Systems Technology; 12(1): 138-147.

Hagras, H., Callaghan, V., Colley, M., Carr-West, M., (1999) A fuzzy-genetic based embedded-agent approach to leaning and control in agricultural autonomous vehicles. Proceeding of the IEEE International Conference Robotics and Automation. p. 1005-1010.

Gillespie, T. D., (1992) Fundamentals of vehicle dynamics. Society of Automotive Engineers, Inc. Warrendale.

Hagras, H., Colley, M., Callaghan, V., Carr-West, M.,(2002) Online Learning and adaptation of autonomous mobile robots for sustainable agriculture. Autonomous Robot; 13(1): 37-52.

Kanayama, Y., Kimura, Y., Miyazaki, F., Noguchi, T., (1990) A stable tracking control method for an autonomous mobile robot. Proceedings of the IEEE conference on Robotics and Automation, p. 384-389.

Katupitiya, J., Eaton, R., Cole, A., Meyer, C., Rodnay, G., (2005) Automation of an Agricultural Tractor for Fruit Picking. Robotics and Automation: Proceedings of the IEEE International Conference; p. 3201-3206.

Kim, D., Oh, T., (1999) Tracking control of a two-wheeled mobile robot using input-output linearization. Control Engineer Prac; 7: 369-73.

Kondo, N., (2005) Latest agricultural robots and traceability information based on robotic agriculture. Resource: Engineering and Technology for Sustainable World; 12(7): 3-4.

Lin, F., Shen, P., (2002) Adaptive backstepping sliding mode control for linear induction motor drive. IEE Proceedings- Electric Power Application; 149(3): 184-194.

Liu, J., (2005) MATLAB Simulation for Sliding Mode Control: Tsinghua University Press.

Nejati, H., Azimifar, Z., Zamani, M., (2008) Using fast fourier transform for weed detection in corn fields. Proceeding of the IEEE International Conference Systems, Man and Cybernetics. p. 1215-1219.

Sabanovic, A., Fridman, E.M., Spurgeon, S., (2004) Variable Structure Systems: from Principles to Implementation, IEE Control Series 66.

Shen, P., Lin, F., (2005) Intelligent backstepping sliding-mode control using RBFN for two-axis motion control system. IEE Proceeding electrical Power Application; 152(5): 1321-1342.

Wang, S., Wang, P., (2005) Nonlinear Modeling and Analysis of Vehicle Planar Motion Dynamics. IEEE International Conference on Mechatronics; p. 90-95.

Watanabe, T., (2004) A duck robot for weeding work on the paddy field. Proceeding of the First IEEE Technical Exhibition Based Conference on Robotics and Automation. p. 81-82

Zhou, J., Zhang, M., Liu, G., Li, S., (2008) Fuzzy Control for Automatic Steering and Line Tracking of Agricultural Robot. Proceeding of the International Conference on Computer Science and Software Engineering, p. 1094-1097.

CHAPTER 4: ROBUST NAVIGATION CONTROL IMPLEMENTATION AND EXPERIMENTS ON A 4WD/4WS ROBOTIC VEHICLE

A paper to be submitted to *Autonomous Robots*

Xuyong Tu, Lie Tang

4.1 Abstract

This chapter demonstrates how to develop and implement a backstepping sliding mode control (SMC) controller on a 4WS and 4WD agricultural robotic vehicle. The experimental results demonstrated capability and robustness of the developed SMC navigation controller when dealing with a nonholonomic system that has a high degree of freedom. This Lyapunov-based controller design method has been proven using mathematical computations, and it performed robustly under off-road conditions. Diverse reference trajectories were chosen to validate the controller performance, including both static and dynamic performance evaluation.

Keywords: agricultural robotic vehicle, robust control, navigation control, backstepping, SMC, trajectory tracking

4.2 Introduction

Sliding mode control has been widely applied in industrial and research engineering control, and it has gained considerable attention for its robustness when parameters are uncertain and external disturbances exist (Utkin, 1992; Hung et al., 1993; Young et al., 1999; Huang et al., 2003). Robustness of a control system is critical because of the complexity of real-world applications. There are uncertainties and time-varying parameters in the systems, as well as unpredictable external noise and disturbances, which the controller needs to overcome. For instance, the dynamics of agricultural robotic vehicles are highly non-linear, coupled, and time varying. Off-road working conditions continuously present challenges and difficulties for the controllers to accommodate and cope with. Therefore, a navigation controller that is resistant to system uncertainties and outside disturbances is essential for an autonomous agricultural robotic vehicle to perform under off-road conditions. In Chapter 3, SMC was theoretically proved to be an effective and robust control method, but its performance has to be validated by field experiments.

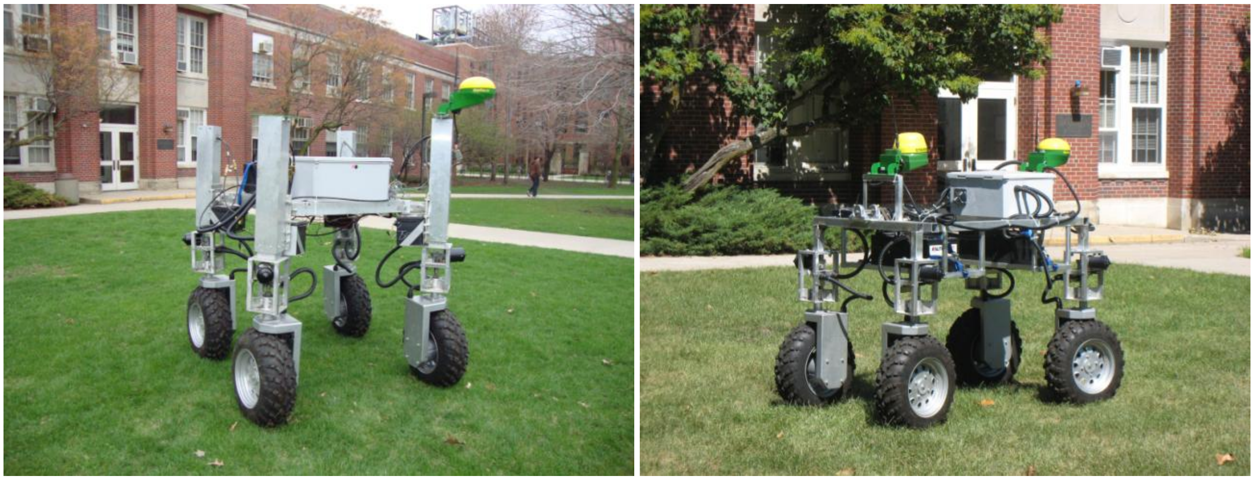
There have been many examples of successful outcomes in the literature. Vessel trajectory tracking control (Cheng et al., 2007) was accomplished using SMC. In the literature, some adaptive control algorithms for trajectory tracking and dynamic positioning of underwater robots have been proposed, such as PD adaptive control (Gianluca et al., 2001; Li and Lee, 2005a; Nguyen and Edwin, 2007).

However, results of field experiments conducted with a vehicle dynamic model suggest that using SMC with a vehicle dynamic model requires high feedback frequency to reflect dynamic performance, which cannot be met by the realistic condition of the equipment. This issue will be discussed, with examples, later in this chapter. Therefore, to accommodate realistic GPS signal sampling frequency, a vehicle kinematic model was adopted and used in a system upper level controller design. With the revised model, the SMC controller presented effective and robust performance, as well as smooth and accurate approaching precision. Quantification error calculation and analysis was conducted; it is discussed later in the chapter.

4.3 Vehicle Modeling

4.3.1 AgRover platform evolution overview

As described in Chapter 1, AgRover has experienced two development phases. The main reason for the evolution from Gen. I to Gen. II was to lighten the weight of the platform and lower the center of gravity. The purpose of this redesign was to find a compromise between the working power provided by the steering and driving motors, and the mobility and maneuverability of the whole vehicle system.



(4.1a) *(4.1b)*

Figure 4.1: *AgRover Gen. I (4.1a) and Gen. II (4.1b)*

After the modification, as briefly mentioned in Chapter 2, the heavy pneumatic leveling system was abandoned to lower the gravity center. This measure improved the stability of the vehicle body. The changes from Gen. I to Gen. II can be seen in Figure 4.1. All future references to “AgRover” mentioned in the paper refer to AgRover Gen. II, unless clearly marked as AgRover Gen.I.

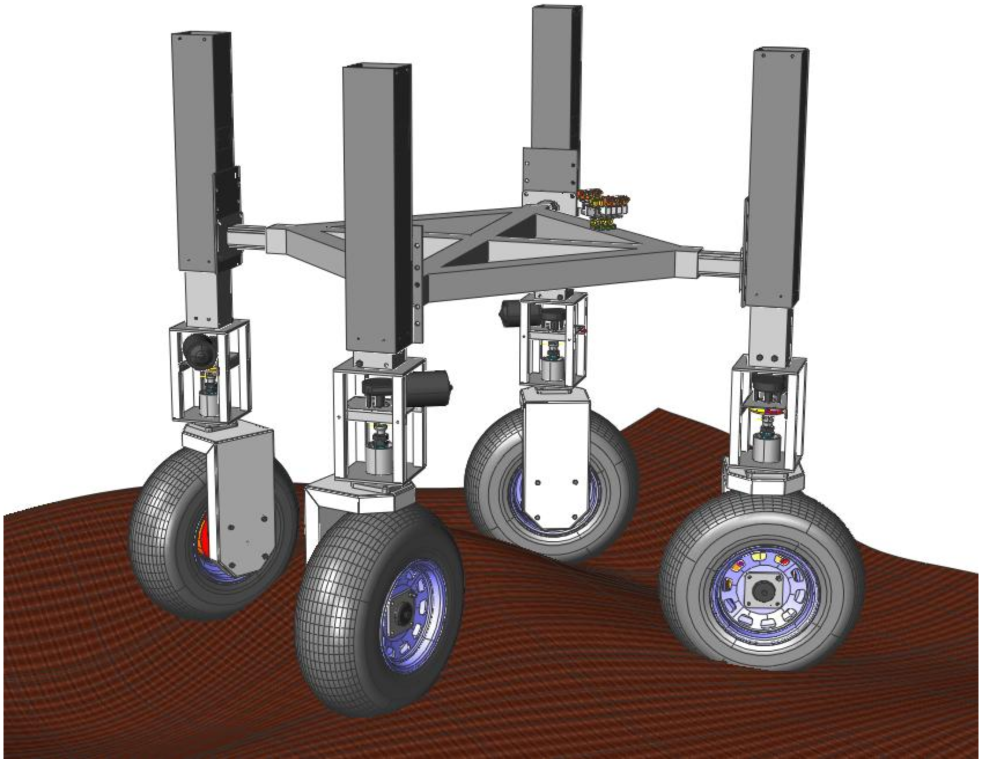


***Figure 4.2**: AgRover Gen. I*

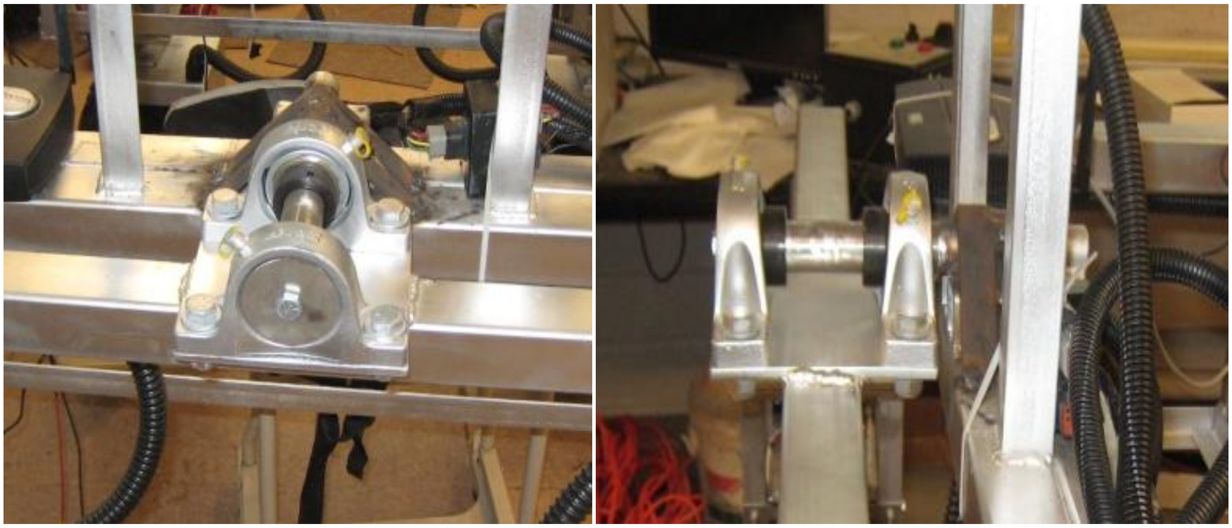


***Figure 4.3:** AgRover Gen. II three-point chassis design with pivot junction*

To maintain off-road capability, AgRover has a three-point chassis design (Figure 4.3), as introduced in Chapter 2, to accommodate off-road field surfaces and ensure that all the wheels are able to provide traction on uneven terrain.

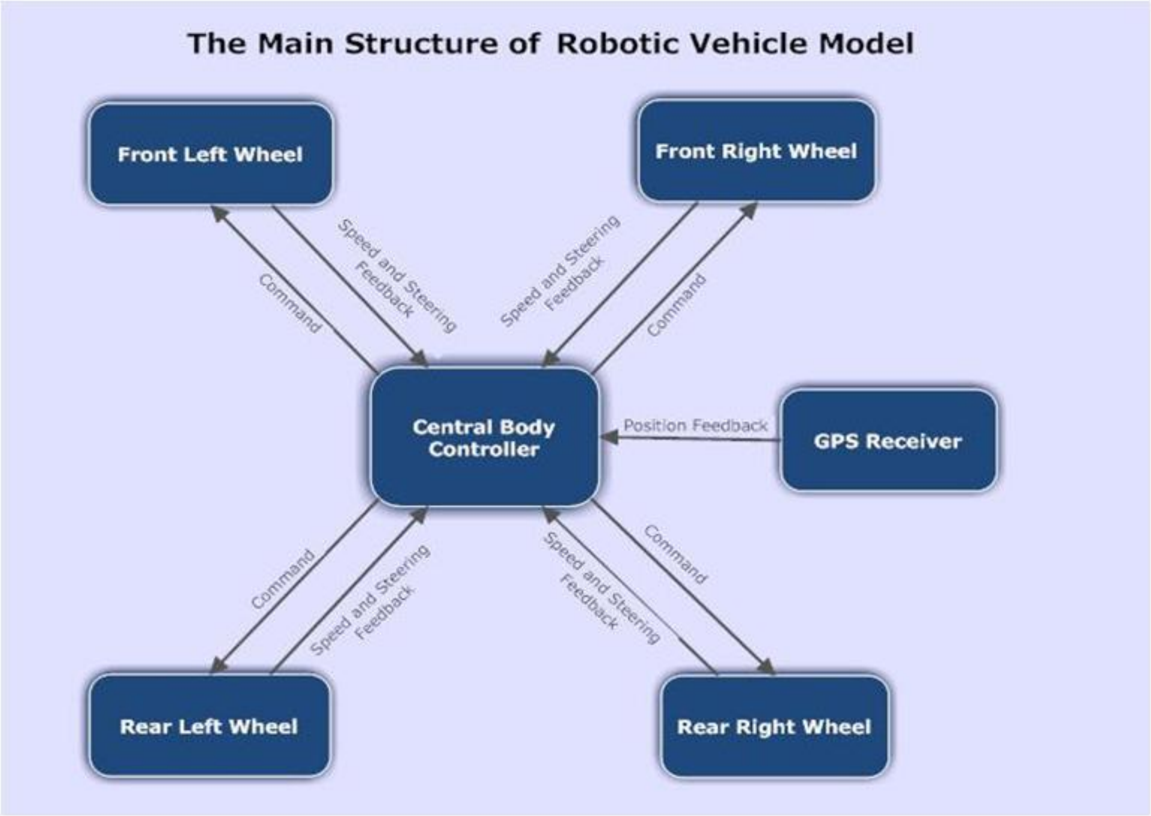

The Main Structure of Robotic Vehicle Model
Front Left Wheel
Front Right Wheel
Command
Speed and Steering Feedback
Speed and Steering Feedback
Command
Central Body Controller
Position Feedback
GPS Receiver
Command
Speed and Steering Feedback
Speed and Steering Feedback
Command
Rear Left Wheel
Rear Right Wheel

Figure 4.4: *AgRover Gen. II controller structure*

As introduced in Chapter 1, the system control architecture is as shown in Figure 4.4. The central controller does not need to access the motor control directly, but instead sends commands to each lower level controller. This architecture is able to ensure that the lower level control, or motor level control, is relatively fast (50 Hz) and the critical hardware control time is efficient.

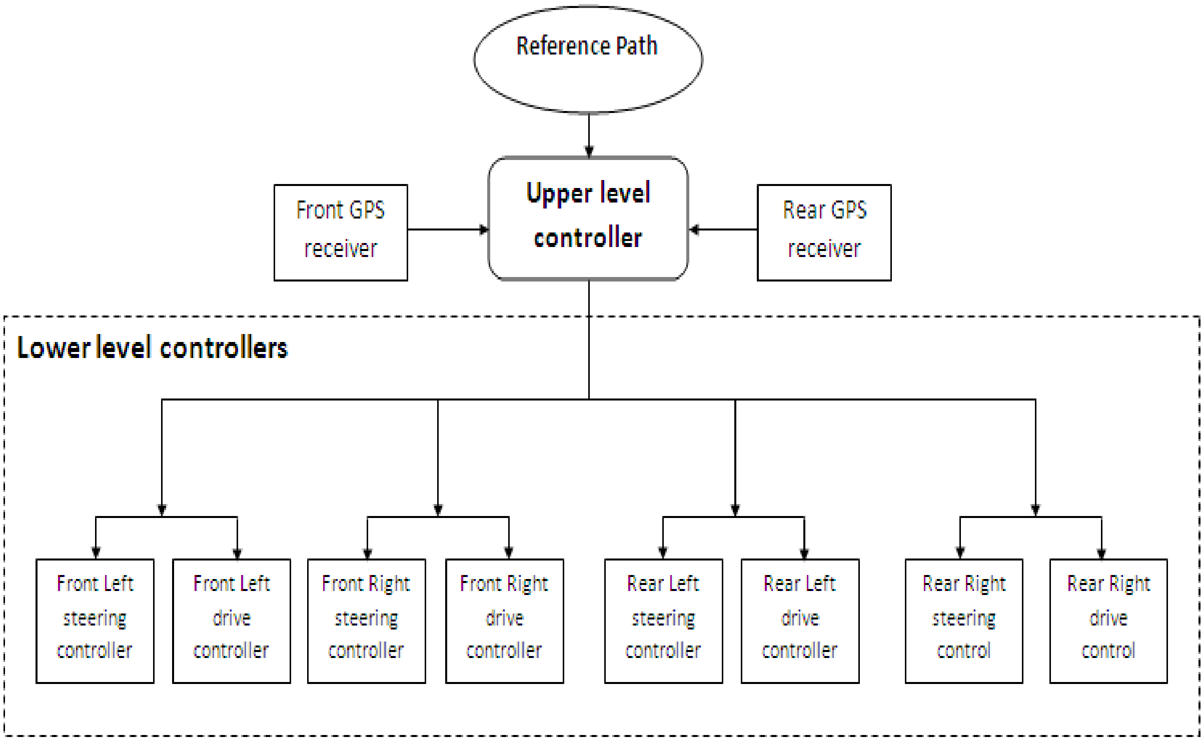

Reference Path
Upper level controller
Front GPS receiver
Rear GPS receiver
Lower level controllers
Front Left steering controller
Front Left drive controller
Front Right steering controller
Front Right drive controller
Rear Left steering controller
Rear Left drive controller
Rear Right steering control
Rear Right drive control

Figure 4.5: *The AgRover controller architecture*

Because the upper level controller focuses on sampling and communication with RTK-GPS receivers, it can guarantee that generating control signals for the motors is in time. In this chapter, the upper level controller, or the central controller, is introduced in detail.

4.3.2 Vehicle kinematic model and local coordinate conversion

Based on the preliminary experimental tests, the results of the dynamic vehicle model were not satisfactory; therefore, the kinematic model was adopted. According to the controller architecture, the upper level controller was no longer responsible for acceleration control, which was handled by the lower level controllers.

Table 4.1: *Parameters and descriptions of the vehicle kinematic model*

Parameter	Description
x	X coordinate in the global 2D coordinate system
y	Y coordinate in the global 2D coordinate system
ψ	Heading angle between the orientation of the vehicle body and X axle in the global 2D coordinate system
x_r	Reference X coordinate in the global 2D coordinate system
y_r	Reference Y coordinate in the global 2D coordinate system
ψ_r	Reference heading angle of the vehicle body in the global 2D coordinate system
x_e	X coordinate in the vehicle 2D coordinate system
y_e	Y coordinate in the vehicle 2D coordinate system
ψ_e	Vehicle heading angle in the vehicle 2D coordinate system
δ	Vehicle steering angle, control variable
u	Vehicle longitudinal velocity, control variable
r	Vehicle yaw rate
δ_r	Reference steering angle, control variable
u_r	Reference longitudinal velocity, control variable
r_r	Reference yaw rate
J	System state vector in the global coordinate system
J_e	System state vector in the vehicle coordinate system
q	Control input vector

The first step of the controller design was to convert the global vehicle position coordinates into a vehicle coordinate system (see Chapter 3). As stated previously, front steering and coordinated steering are the selected areas to be studied. Further examination regarding how to set up the vehicle system model and realize the conversion for these two steering modes will be presented. A “bicycle” two-wheel model will be used to simplify the modeling procedure and present the conversion in a straightforward manner.

4.3.2.1 Front-steering kinematic model conversion

Under front-steering mode, the robot actually worked like a conventional front-steering car, and the yaw motion center was at the middle point of the rear axle, which was the same case for the robot.

Table 4.2: *Subscript in vehicle modeling*

Subscript	Description
f	Variables referring to the front wheels
r	Variables referring to the rear wheels

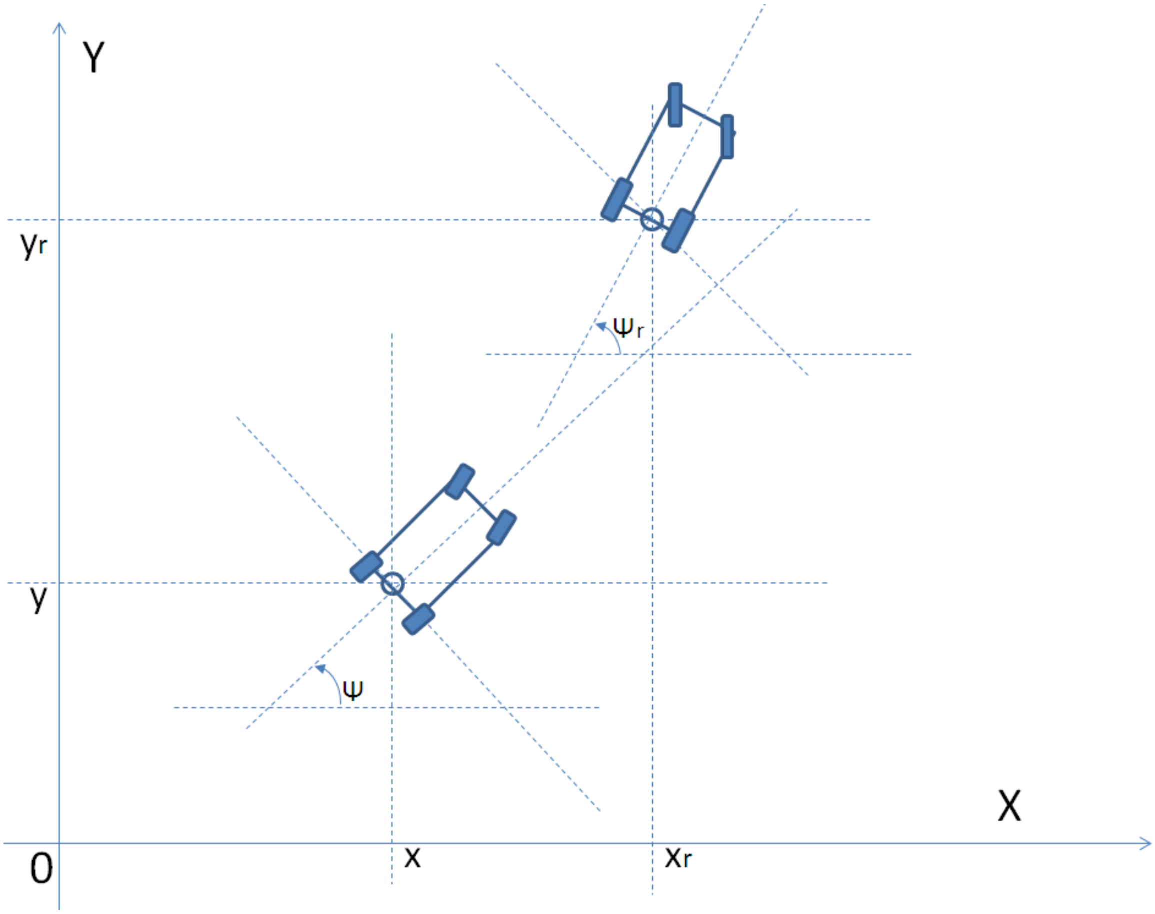


Figure 4.6: *AgRover under front steering mode in the global 2D coordinate system, with its reference coordinates and orientation. The lower figure is the current location and gesture of AgRover, and the upper one is the target location and gesture.*

As previously stated, the robot can be simplified by a two-wheel bicycle model, from which it is more straightforward to attain coordinate conversion from the global 2D coordinate system to the local vehicle coordinate system (see Figure 4.6).

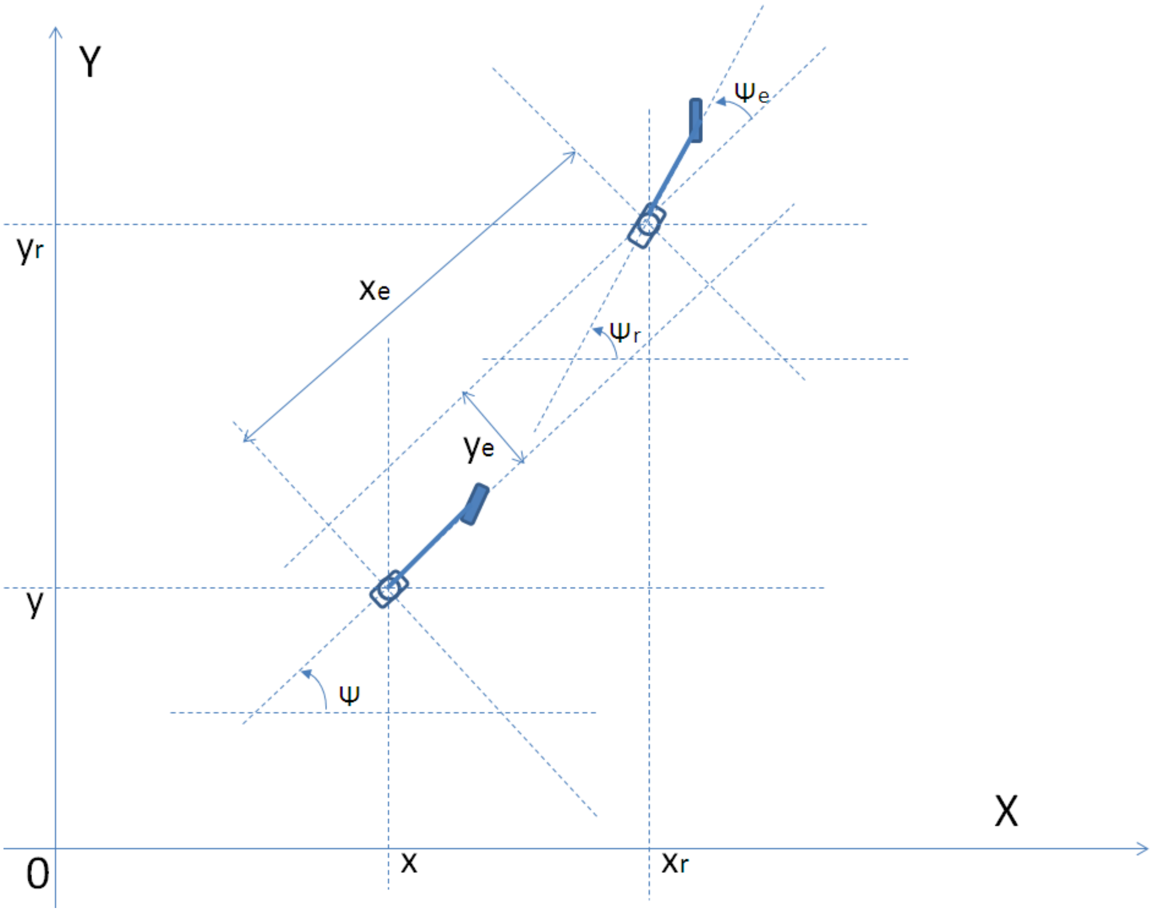


Figure 4.7 *AgRover two-wheel bicycle model under front-steering mode in the global 2D coordinate system, with its reference coordinates and orientation. x_e and y_e represent the coordinates of the reference point in the current vehicle coordinate system.*

In Figure 4.7, the angle between the vehicle local coordinate system $X_e - Y_e$ and the global coordinate system $X - Y$ is equal to the current heading angle ψ. The following relationships are obvious from the diagram (Figure 4.7):

$$\psi_e = \psi_r - \psi \ , \tag{4.1}$$

$$x_e = (x_r - x)cos\psi + (y_r - y\ )sin\psi \ , \tag{4.2}$$

$$y_e = (x_r - x)(-sin\psi) \ + (y_r - y\ )\ cos\psi. \tag{4.3}$$

Further, the transformation matrix is defined as

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\psi & sin\psi \\ 0 & -sin\psi & cos\psi \end{bmatrix} \quad . \tag{4.4}$$

Define $J = [\psi \quad x \quad y]^T$ and $J_e = [\psi_e \quad x_e \quad y_e]^T$, and $q = \ [\delta \quad u]^T$; the conversion equation is derived as follows:

$$J_e = \begin{bmatrix} \psi_e \\ x_e \\ y_e \end{bmatrix} = \begin{bmatrix} \psi_r - \psi \\ (x_r - x)cos\psi + (y_r - y\ )sin\psi \\ (x_r - x)(-sin\psi)\ + (y_r - y\ )\ cos\psi \end{bmatrix} \tag{4.5}$$

and

$$= \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\psi & sin\psi \\ 0 & -sin\psi & cos\psi \end{bmatrix} \begin{bmatrix} \psi_r - \psi \\ x_r - x \\ y_r - y \end{bmatrix} . \tag{4.6}$$

The system state equation was attained in the vehicle local coordinate system as follows:

$$\because \begin{cases} \dot{\psi} = r \\ \dot{x} = ucos\psi \\ \dot{y} = usin\psi \end{cases} \tag{4.7}$$

$$\therefore u = \dot{x}cos\psi + \dot{y}sin\psi \tag{4.8}$$

$$0 = \dot{x}sin\psi - \dot{y}cos\psi \tag{4.9}$$

$$\text{And} \because \begin{cases} \dot{\psi_r} = r_r \\ \dot{x_r} = u_r cos\psi_r \\ \dot{y_r} = u_r sin\psi_r \end{cases} \tag{4.10}$$

$$\therefore u_r = \dot{x_r}cos\psi_r + \dot{y_r}sin\psi_r \tag{4.11}$$

$$0 = \dot{x_r}sin\psi_r - \dot{y_r}cos\psi_r \tag{4.12}$$

Therefore, the following simplification can be derived:

$$\dot{\psi_e} = \dot{\psi_r} - \dot{\psi} = \ r_r - r \tag{4.13}$$

$$\dot{x_e} = (\dot{x_r} - \dot{x})cos\psi - (x_r - x)sin\psi\dot{\psi} + (\dot{y_r} - \dot{y})sin\psi + (y_r - y)cos\psi\dot{\psi}$$

$$= (-(x_r - x)sin\psi + (y_r - y)cos\psi)\dot{\psi} -\ (\dot{x}cos\psi + \dot{y}sin\psi) + (\dot{x_r}cos\psi + \dot{y_r}sin\psi)$$

$$= y_e r - u + \dot{x_r} \cos(\psi_r - \psi_e) + \dot{y_r} sin(\psi_r - \psi_e)$$

$$= y_e r - u + (\dot{x_r} cos\psi_r + \dot{y_r} sin\psi_r) cos\psi_e + (\dot{x_r} sin\psi_r - \dot{y_r} cos\psi_r) sin\psi_e$$

$$= y_e r - u + \ u_r cos\psi_e \tag{4.14}$$

Similarly,

$$\dot{y_e} = -(\dot{x_r} - \dot{x}) \sin\psi - (x_r - x) \cos\psi \,\dot{\psi} + (\dot{y_r} - \dot{y}) \cos\psi - (y_r - y) \sin\psi \,\dot{\psi}$$

$$= \ -\dot{x_r} sin\psi + \dot{y_r} cos\psi + (\dot{x} sin\psi - \dot{y} cos\psi) - (x_r cos\psi + y_r sin\psi)\dot{\psi}$$

$$+ (x cos\psi + y sin\psi)\dot{\psi}$$

$$= -\dot{x_r} \sin(\psi_r - \psi_e) + \dot{y_r} cos(\psi_r - \psi_e) - \big((x_r - x) cos\psi + (y_r - y) sin\psi\big)\dot{\psi}$$

$$= -\dot{x_r} \sin(\psi_r - \psi_e) + \dot{y_r} cos(\psi_r - \psi_e) - x_e r$$

$$= \ -\dot{x_r} (\sin\psi_r cos\psi_e - cos\psi_r sin\psi_e) + \dot{y_r} (\cos\psi_r cos\psi_e + \sin\psi_r sin\psi_e)$$

$$- x_e r$$

$$= (-\dot{x_r} sin\psi_r + \dot{y_r} \cos\psi_r) cos\psi_e + (\dot{x_r} cos\psi_r + \dot{y_r} \sin\psi_r) sin\psi_e - x_e r$$

$$= u_r sin\psi_e - \ x_e r \tag{4.15}$$

Given the above equations, the state function in the vehicle coordinate system was derived as follows:

$$\dot{J}_e = \begin{bmatrix} \dot{\psi}_e \\ \dot{x}_e \\ \dot{y}_e \end{bmatrix} = \begin{bmatrix} r_r - r \\ y_e r - u + \ u_r cos\psi_e \\ u_r sin\psi_e - \ x_e r \end{bmatrix} = \begin{bmatrix} r_r \\ u_r cos\psi_e \\ u_r sin\psi_e \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ y_e & -1 \\ -x_e & 0 \end{bmatrix} \begin{bmatrix} r \\ u \end{bmatrix}. \tag{4.16}$$

In front-steering mode, the relationship between r and δ was

$$r = \ \frac{utan\delta}{L} \ , \tag{4.17}$$

where L was the robot wheelbase in (3.17). Therefore, the control signal for the steering wheel was

$$\delta = \arctan\left(\frac{rL}{u}\right) \quad . \tag{4.18}$$

4.3.2.2 Coordinated steering kinematic model conversion

The difference in coordinated steering is that, because the front and rear wheels are both steering, the yaw motion center changes from the rear axle to the geometric center of the chassis (Figure 4.8).

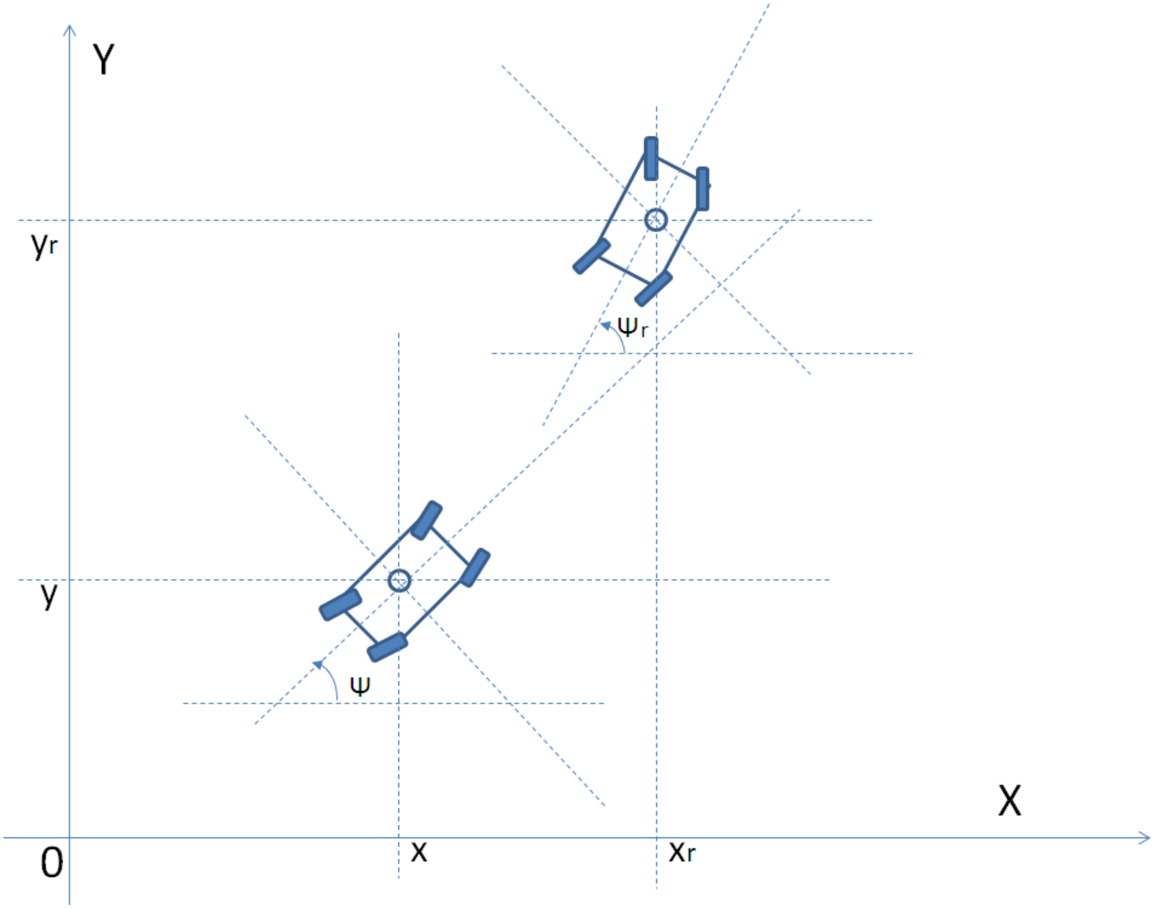


Figure 4.8: *AgRover under coordinated steering mode in the global 2D coordinate system, with its reference coordinates and orientation. The lower figure is the current location and gesture of AgRover, and the upper one is the target location and gesture.*

The simplified bicycle two-wheel steering model needed be revised to account for the differences in steering (Figure 4.9).

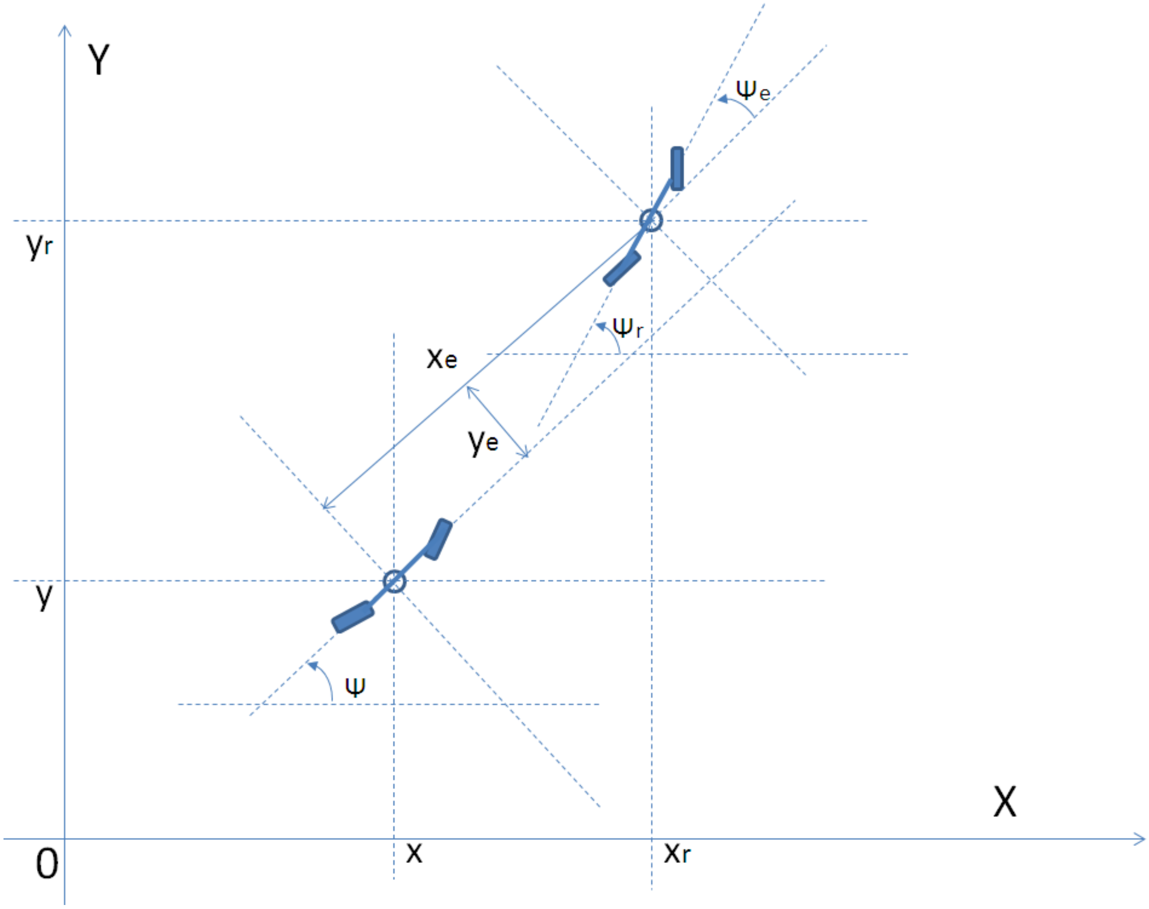


Figure 4.9: *AgRover two-wheel bicycle model under coordinated steering mode in the global 2D coordinate system, with its reference coordinates and orientation.* x_e *and* y_e *represent the coordinates of the reference point in the current vehicle coordinate system.*

The modeling process for J_e was exactly the same as for the front-steering mode; the state equations were

$$J_e = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\psi & \sin\psi \\ 0 & -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} \psi_r - \psi \\ x_r - x \\ y_r - y \end{bmatrix} \tag{4.19}$$

and

$$\dot{J_e} = \begin{bmatrix} \dot{\psi_e} \\ \dot{x_e} \\ \dot{y_e} \end{bmatrix} = \begin{bmatrix} r_r - r \\ y_e r - u + u_r \cos\psi_e \\ u_r \sin\psi_e - x_e r \end{bmatrix} = \begin{bmatrix} r_r \\ u_r \cos\psi_e \\ u_r \sin\psi_e \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ y_e & -1 \\ -x_e & 0 \end{bmatrix} \begin{bmatrix} r \\ u \end{bmatrix} . \qquad (4.20)$$

Because the front and rear steering angles have the same magnitude, but in opposite directions, the steering angles can specified as

$$r = \frac{u \tan\delta_f}{L/2} = \frac{u \tan(-\delta_r)}{L/2} \quad . \qquad (4.21)$$

The control signal for the front and rear steering wheels is

$$\delta_f = -\delta_r = \tan^{-1}(\frac{rL}{2u}) \quad . \qquad (4.22)$$

4.4 Backstepping SMC Controller Design

The purpose of the backstepping SMC controller was to compute the solution for $[r \quad u]^T$, as well as to ensure system convergence. In other words, the ultimate goal of the SMC controller is to realize that in finite time, $x_e \to 0$, $y_e \to 0$ and $\psi_e \to 0$. In this controller design process, the switch function design is critical and challenging. The procedure to design the switch function requires a detailed discussion.

4.4.1 An important theorem

Theorem 1 (Wu et al., 2001; Liu, 2005). For $\forall x \epsilon R$ and $|x| < \infty$,

$\Phi(x) = x \sin(\tan^{-1} x) \geq 0$ and $\Phi(x) = 0 \Leftrightarrow x = 0$.

Proof. It can be proved under the following three circumstances:

1) If $x = 0$, $\Phi(x) = 0$.

2) If $x \epsilon (-\infty, 0)$, $\tan^{-1} x \, \epsilon (-\frac{\pi}{2}, 0)$, so $\sin(\tan^{-1} x) < 0$, $\Phi(x) > 0$.

3) If $x \epsilon (0, +\infty)$, $\tan^{-1} x \, \epsilon (0, \frac{\pi}{2})$, so $\sin(\tan^{-1} x) > 0$, $\Phi(x) > 0$.

The proved theorem plays an important role in the design of the switch function.

4.4.2 Switch function design

Sliding mode control is a Lyapunov stability theory-based controller design methodology, and the generation of the controller and the convergence of the system are both related to Lyapunov theory.

When $x_e = 0$, define the Lyapunov function as

$$V = \frac{1}{2} y_e^2 \ . \tag{4.23}$$

Assume $\psi_e = -\tan^{-1}(u_r y_e)$, and by using Eq. (4.15), we get following equation:

$$\dot{V} = y_e \dot{y}_e = \ y_e(u_r \sin \psi_e - x_e r) = y_e u_r \sin(-\tan^{-1}(u_r y_e)) - y_e x_e r$$

$$= -y_e u_r \sin(\tan^{-1}(y_e u_r)) - y_e x_e r \tag{4.24}$$

$$\because x_e = 0 \tag{4.25}$$

$$\therefore y_e x_e r = 0 \quad . \tag{4.26}$$

From Theorem 1, it can be concluded that

$$y_e u_r \sin(\tan^{-1}(y_e u_r)) \geq 0, \text{ and } y_e u_r \sin(\tan^{-1}(y_e u_r)) = 0 \ \Leftrightarrow \ y_e u_r = 0. \tag{4.27}$$

Therefore, it is guaranteed that

$$\dot{V} \leq 0 \ \text{ and } y_e \rightarrow 0. \tag{3.28}$$

Thus, if $x_e = 0$ and ψ_e is contingent to $-\tan^{-1}(u_r y_e)$, the state variable $y_e$is contingent to 0. According to this conclusion, two switching functions can be designed as

$$\boldsymbol{S} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = \begin{bmatrix} x_e \\ \psi_e + \tan^{-1}(u_r y_e) \end{bmatrix} . \tag{4.29}$$

The next step is to design the SMC controller and let $s_1 \to 0$ and $s_2 \to 0$, which may lead to $x_e \to 0$. Meanwhile, ψ_e is contingent to $-\tan^{-1}(u_r y_e)$, and $y_e \to 0$ and $\psi_e \to 0$.

4.4.3 SMC controller signals

The tasks for this step were to design the approaching law and attain controller output. In the SMC approaching law selection, there are several popular options, one of which is sign function:

$$\dot{s} = -k \operatorname{sgn} s \ , k > 0 \, . \tag{4.30}$$

However, sign function approaching is fast, but it is not continuous and usually causes chattering issues. It does not account for fast robot motion control scenarios. Thus, an improved continuous approaching law was adopted:

$$\dot{s} = -k \frac{s}{|s|+\sigma} \, , k, \sigma > 0. \tag{4.31}$$

This method maintained rapid approaching speed, as sign function does, and it eases overshooting while penetrating the sliding surface $s = 0$.

With $\dot{s_1} = -k_1 \frac{s_1}{|s_1|+\sigma_1}$ and $\dot{s_2} = -k_2 \frac{s_2}{|s_2|+\sigma_2}$

$$\dot{\boldsymbol{S}} = \begin{bmatrix} \dot{s_1} \\ \dot{s_2} \end{bmatrix} = \begin{bmatrix} -k_1 \frac{s_1}{|s_1|+\sigma_1} \\ -k_2 \frac{s_2}{|s_2|+\sigma_2} \end{bmatrix} = \begin{bmatrix} \dot{x_e} \\ \dot{\psi_e} + \frac{\partial \tan^{-1}(u_r y_e)}{\partial u_r} + \frac{\partial \tan^{-1}(u_r y_e)}{\partial y_e} \end{bmatrix}, \tag{4.32}$$

substitute (4.13–4.15) for (4.32):

$$\dot{\boldsymbol{S}} = \begin{bmatrix} -k_1 \frac{s_1}{|s_1|+\sigma_1} \\ -k_2 \frac{s_2}{|s_2|+\sigma_2} \end{bmatrix} = \begin{bmatrix} y_e r - u + u_r cos\psi_e \\ r_r - r + \frac{\mathrm{y_e}}{1+(\mathrm{u_r y_e})^2} \dot{\mathrm{u}}_\mathrm{r} + \frac{\mathrm{u_r}}{1+(\mathrm{u_r y_e})^2} (u_r sin\psi_e - x_e r) \end{bmatrix} \tag{4.33}$$

Then, the control output can be attained:

$$q = \begin{bmatrix} u \\ r \end{bmatrix} = \begin{bmatrix} y_e r + \ u_r cos\psi_e + k_1 \frac{s_1}{|s_1| + \sigma_1} \\ \frac{r_r + \frac{y_e}{1+(u_r y_e)^2}\dot{u}_r + \frac{u_r}{1+(u_r y_e)^2} u_r sin\psi_e + k_2 \frac{s_2}{|s_2| + \sigma_2}}{1 + \frac{u_r}{1+(u_r y_e)^2} x_e} \end{bmatrix}. \tag{4.34}$$

If the vehicle is under front-wheel steering, the steering angle for the bicycle model is

$$\delta = \arctan\left(\frac{rL}{u}\right). \tag{4.35}$$

If coordinated steering is the case, the front and rear steering angles are

$$\delta_f = \ -\delta_r = \tan^{-1}(\frac{rL}{2u}).$$

(4.36)

From the development, it is proven and known that the system is contingent to the origin, where $x_e \rightarrow 0, y_e \rightarrow 0$, and $\psi_e \rightarrow 0$. The four control parameters, k_1, σ_1, k_2, and σ_2, affect the dynamic performance of the controller; this is investigated in the following discussion.

4.5 Controller Implementation

The control signals, the steering angle, and the reference velocity are calculated and generated by the upper level controller. However, these are not the signals that can directly communicate with the motors. Before talking to the lower level controllers, the signals have to be translated into specific control signals for all four steering motors and the four driving motors. Further, $[u \quad \delta]$ are considered as known parameters in this discussion.

Table 4.3: *AgRover physical information*

Description (unit)	Symbol	Value
Wheelbase (cm)	L	138.43
Tread (cm)	d	91.44
Tire Diameter (cm)	d_t	55.88

The basic physical specifications of AgRover, listed in Table 4.3, were used in the conversion from central control signals to the wheel control information. Because AgRover is a four-wheel-steering and four-wheel-driving platform, there are four available steering modes, which were introduced in Chapter 2. In the practical experiments presented in this paper, front steering and coordinated steering were examined and tested.

Table 4.4: *AgRover subscript notation*

Subscript	Definition
l	Left wheels
r	Right wheels
fl	Front left wheel
fr	Front right wheel
rl	Rear left wheel
rr	Rear right wheel

4.5.1 Controller implementation under front steering mode

Under front steering mode, the turning center is on the extension of the rear axle, and the steering angle and velocity signal for each wheel can be calculated using the diagram shown in Figure 4.10.

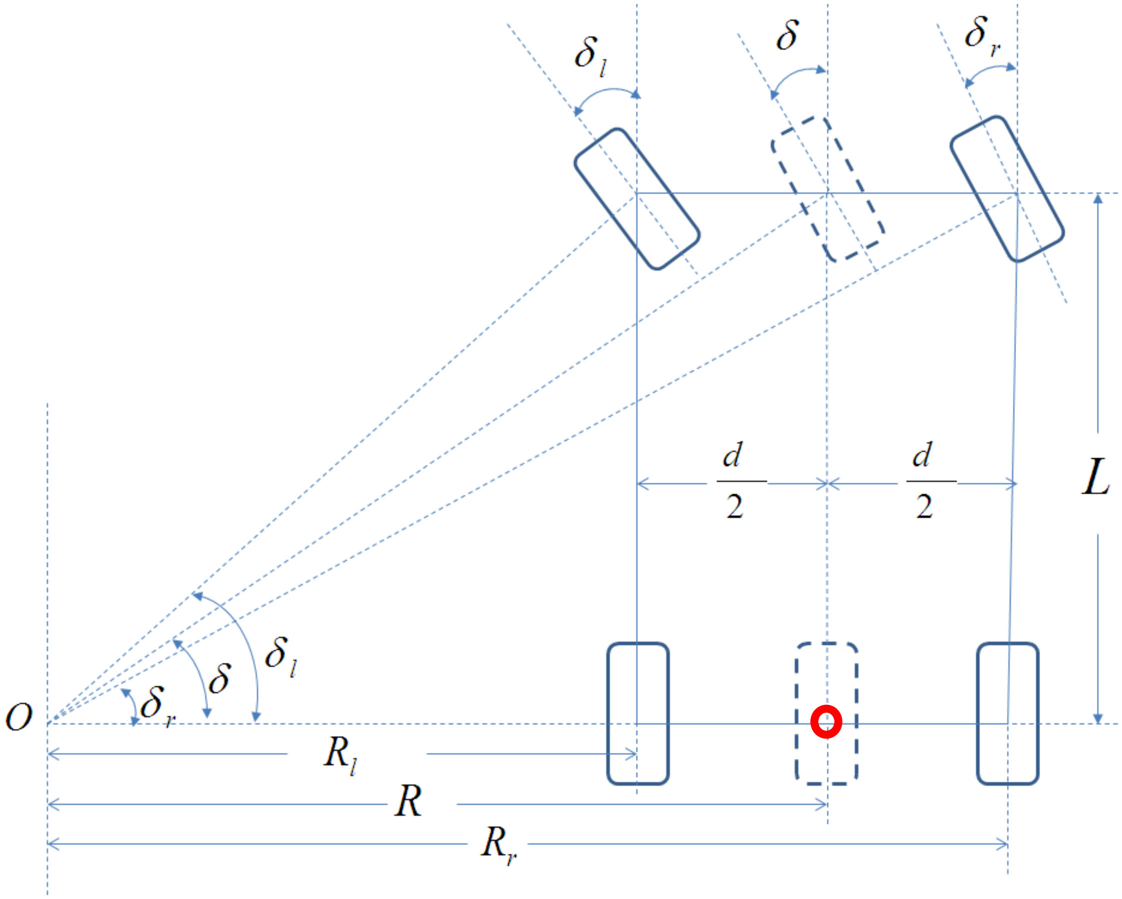


Figure 4.10: *Signal conversion diagram: front wheel steering mode*

With steering signal δ given by the SMC controller, the turning center can be located as

$$R = \frac{L}{\tan\delta}. \tag{4.37}$$

Then, the left and right wheel turning radii can be derived by

$$\mathrm{R_l} = \mathrm{R} - \frac{\mathrm{d}}{2} = \frac{\mathrm{L}}{\tan\delta} - \frac{\mathrm{d}}{2} \tag{4.38}$$

and

$$\mathrm{R_r} = \mathrm{R} + \frac{\mathrm{d}}{2} = \frac{\mathrm{L}}{\tan\delta} + \frac{\mathrm{d}}{2}\,. \tag{4.39}$$

The left steering angle is attained by

$$\delta_\mathrm{l} = \arctan\left(\frac{\mathrm{L}}{\mathrm{R_l}}\right) = \arctan\left(\frac{\mathrm{L}}{\mathrm{R} - {}^{\mathrm{d}}/_{2}}\right) = \arctan\,(\frac{\mathrm{L}}{\frac{\mathrm{L}}{\tan\delta} - \frac{\mathrm{d}}{2}}) \tag{4.40}$$

and the right steering angle can also be provided:

$$\delta_\mathrm{r} = \arctan\left(\frac{\mathrm{L}}{\mathrm{R_r}}\right) = \arctan\left(\frac{\mathrm{L}}{\mathrm{R} + {}^{\mathrm{d}}/_{2}}\right) = \arctan\left(\frac{\mathrm{L}}{\frac{\mathrm{L}}{\tan\delta} + \frac{\mathrm{d}}{2}}\right). \tag{4.41}$$

The velocity control signals for every driving motor can be attained accordingly:

$$u_{fl} = u\frac{R_l}{R\cos\delta_\mathrm{l}}\,, \tag{4.42}$$

$$u_{fr} = u\frac{R_r}{R\cos\delta_\mathrm{r}}\,, \tag{4.43}$$

$$u_{rl} = u\frac{R_l}{R}, \tag{4.44}$$

$$u_{rr} = u\frac{R_r}{R}. \tag{4.45}$$

4.5.2 Controller implementation under coordinated steering mode

Similarly, under coordinated steering mode, the turning center is on the extension of the line pertaining to the geometric center, parallel with the front and rear axles. Thus, the steering angle and velocity signal for each wheel can be calculated using the diagram shown in Figure 4.11.

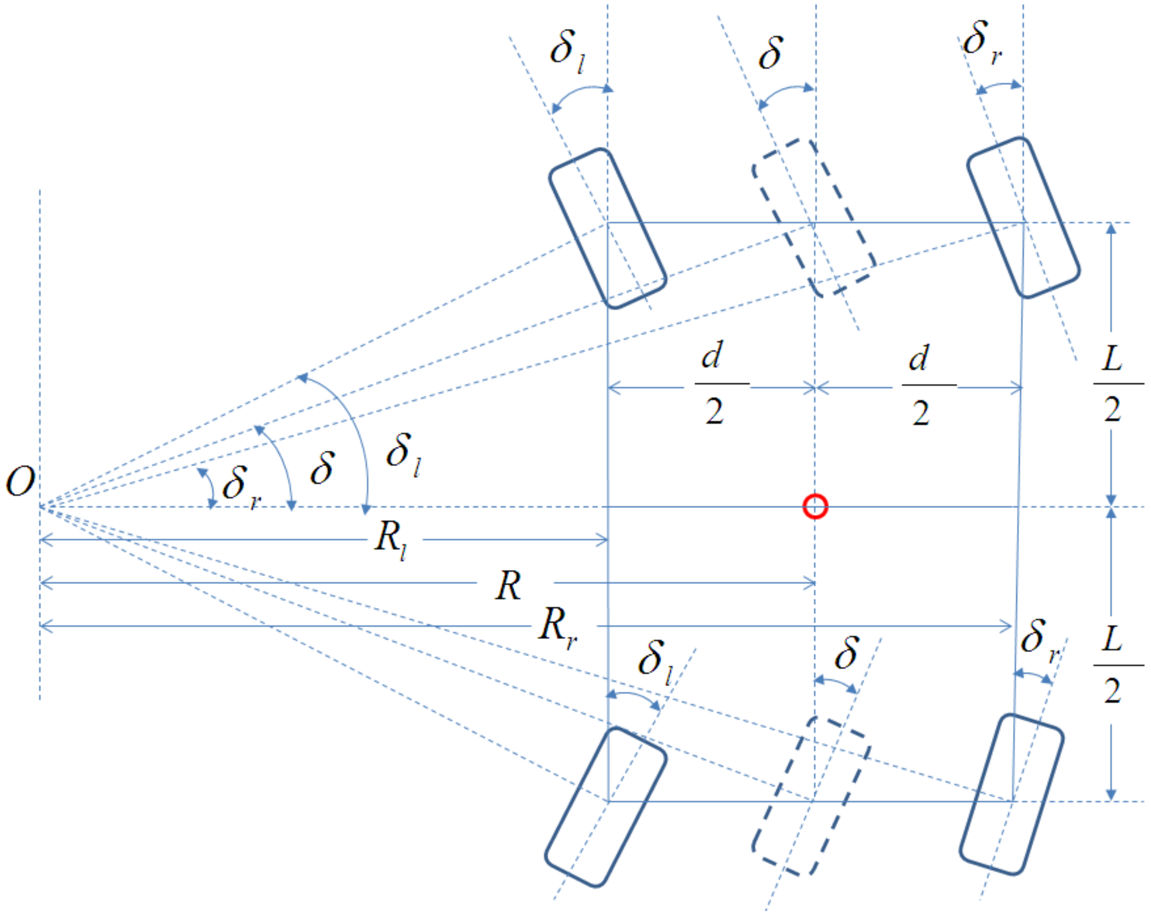


Figure 4.11: *Signal conversion diagram: coordinated steering mode*

With steering signal δ given by the SMC controller, the turning center can be located as

$$\mathrm{R} = \frac{{}^{\mathrm{L}}\!/_{2}}{\tan\delta} = \frac{\mathrm{L}}{2\tan\delta}\,. \tag{4.46}$$

The turning radii of the left and right wheels can be calculated by

$$\mathrm{R_l} = \mathrm{R} - \frac{\mathrm{d}}{2} = \frac{\mathrm{L}}{2\tan\delta} - \frac{\mathrm{d}}{2} \tag{4.47}$$

and

$$\mathrm{R_r} = \mathrm{R} + \frac{\mathrm{d}}{2} = \frac{\mathrm{L}}{2\tan\delta} + \frac{\mathrm{d}}{2}\ \ , \tag{4.48}$$

leading to the steering angles of each wheel, as follows:

$$\delta_{\mathrm{fl}} = \arctan\left(\frac{{}^{\mathrm{L}}\!/_{2}}{\mathrm{R_l}}\right) = \arctan\left(\frac{\mathrm{L}}{2\mathrm{R}-\mathrm{d}}\right) = \arctan\,(\frac{\mathrm{L}}{\frac{\mathrm{L}}{\tan\delta}-\mathrm{d}})\ , \tag{4.49}$$

$$\delta_{\mathrm{rl}} = -\arctan\left(\frac{{}^{\mathrm{L}}\!/_{2}}{\mathrm{R_l}}\right) = -\arctan\left(\frac{\mathrm{L}}{2\mathrm{R}-\mathrm{d}}\right) = -\arctan\,(\frac{\mathrm{L}}{\frac{\mathrm{L}}{\tan\delta}-\mathrm{d}})\ , \tag{4.50}$$

$$\delta_{\mathrm{fr}} = \arctan\left(\frac{{}^{\mathrm{L}}\!/_{2}}{\mathrm{R_r}}\right) = \arctan\left(\frac{\mathrm{L}}{2\mathrm{R}+\mathrm{d}}\right) = \arctan\,(\frac{\mathrm{L}}{\frac{\mathrm{L}}{\tan\delta}+\mathrm{d}})\ , \tag{4.51}$$

$$\delta_{\mathrm{rr}} = -\arctan\left(\frac{{}^{\mathrm{L}}\!/_{2}}{\mathrm{R_r}}\right) = -\arctan\left(\frac{\mathrm{L}}{2\mathrm{R}+\mathrm{d}}\right) = -\arctan\,(\frac{\mathrm{L}}{\frac{\mathrm{L}}{\tan\delta}+\mathrm{d}})\ . \tag{4.52}$$

The velocities of all the wheels are attained easily:

$$u_{fl} = u_{rl} = u\frac{R_l}{R\cos\delta_{\mathrm{l}}}\,, \tag{4.53}$$

$$u_{fr} = u_{rr} = u\frac{R_r}{R\cos\delta_{\mathrm{r}}}\,. \tag{4.43}$$

4.5.3 Controller working diagram

Thus far, then, the controller design and all motor control command signals have been finished. In the vehicle operating software, a multi-threading technique is used to accommodate the large amount of the hardware control tasks and to handle the rapid sampling frequency.

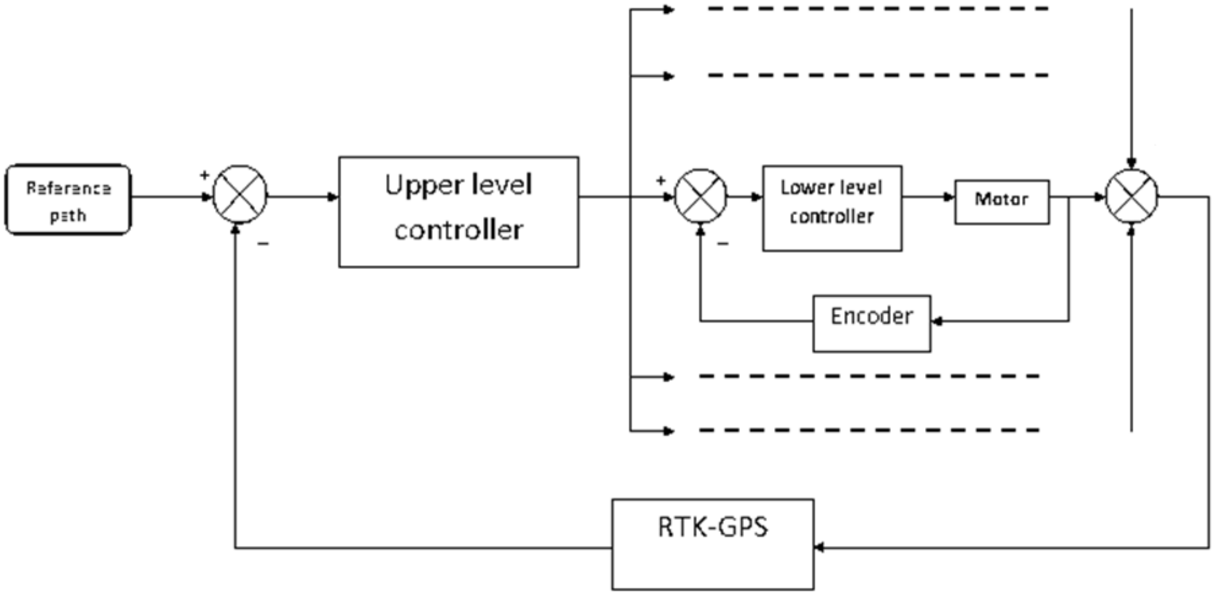


Figure 4.12: *AgRover control loops working flow diagram*

4.6 Automatic Guidance Working Procedure

As introduced in Chapter 1, the shift between manual mode and automatic mode on the AgRover has a safety redundancy design. The details regarding the automatic guidance working procedure are demonstrated in this section (Figure 4.13). This process involves a lot of realistic and practical considerations, which are critical for the AgRover itself and important for the operational safety of personnel and objects at the working site.

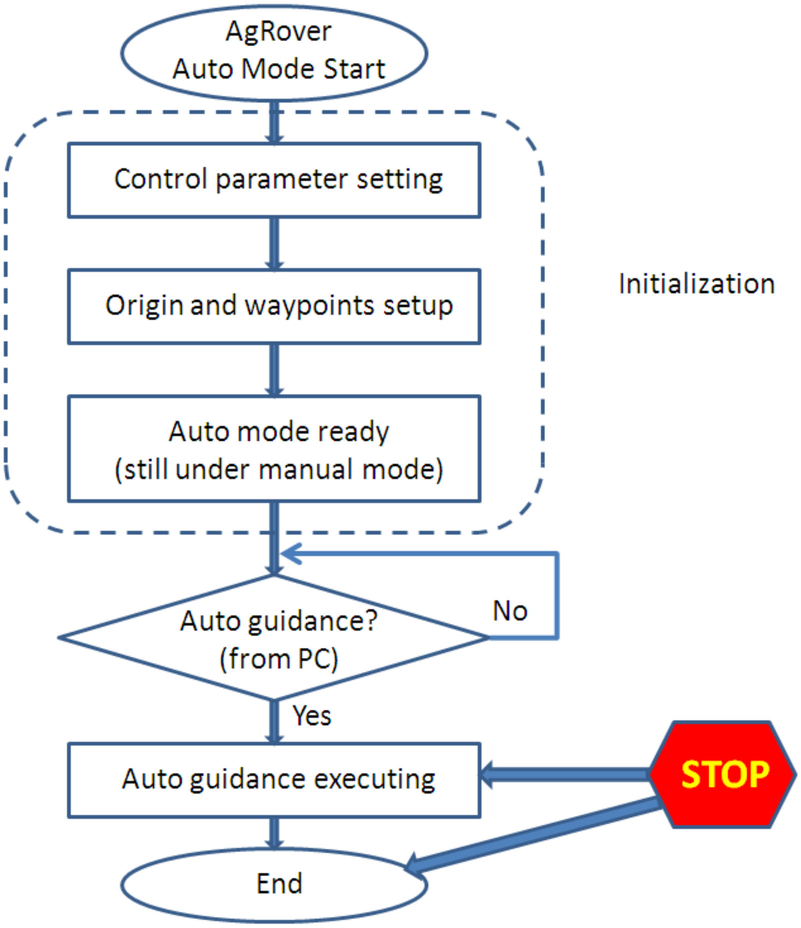
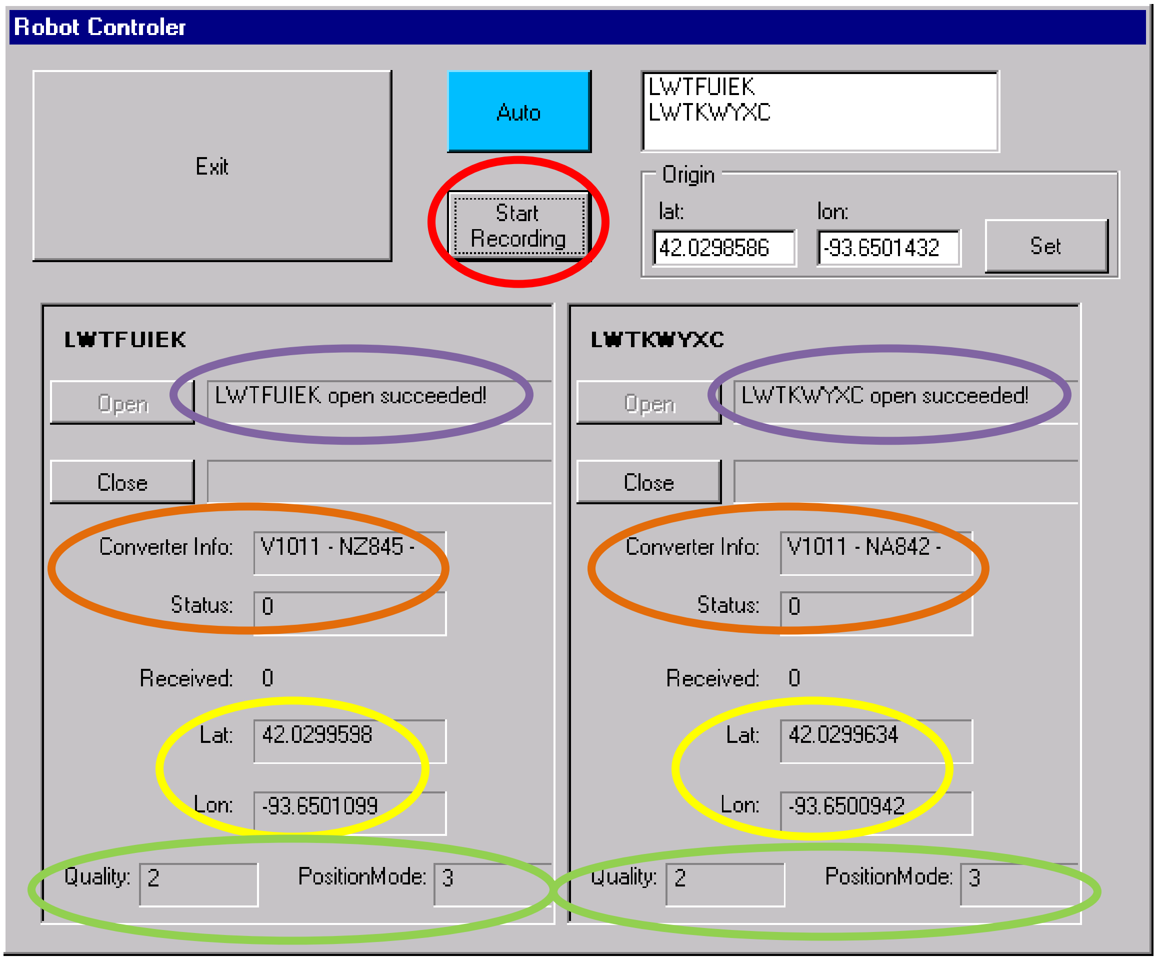


Figure 4.13: *AgRover automatic guidance working procedure*

4.6.1 GPS-related hardware checking

The initialization procedure, shown in Figure 4.13, takes several steps. The automatic guidance mode can be started after the AgRover has been fully started in a normal manner and it is working flawlessly, controlled by the joystick. Another precondition for determining whether the experiment can be executed is the status of the two GPS receivers. The items on the checklist for GPS receiver status is shown in Figure 4.14.



LWTFUIEK
LWTKWYXC

Origin
lat: 42.0298586
lon: -93.6501432

LWTFUIEK

LWTFUIEK open succeeded!

Converter Info: V1011 - NZ845 -
Status: 0
Received: 0
Lat: 42.0299598
Lon: -93.6501099
Quality: 2
PositionMode: 3

LWTKWYXC

LWTKWYXC open succeeded!

Converter Info: V1011 - NA842 -
Status: 0
Received: 0
Lat: 42.0299634
Lon: -93.6500942
Quality: 2
PositionMode: 3

***Figure 4.14** Checklist for the status of the GPS receivers*

1) Interface status, as marked by purple circles. It indicates whether the USB interface is properly open. Any further procedures can be conducted only when both of the receivers indicate “open succeeded!” Otherwise, hardware troubleshooting is required.
2) Converter status, as marked with orange circles. Each of the LAWICEL (Sweden) CAN/USB converters has its unique serial ID, and each has hardware self-checking capability. If an error code comes back and displays “Status,” a malfunction or disoperation has occurred.
3) Signal quality and working mode, as marked with green circles. Because it is a precise automatic navigation experiment, both of the GPS receivers are required to be working

under “3D RTK” mode, as indicated in Figure 4.14. Multiple factors could cause the GPS receiver not to work properly. One of the common possibilities is that the mobile GPS signal is shadowed, or the communication from the base station is blocked.

4) Latitude and longitude data, as marked with yellow circles. The data from the base station is readable and can be compared with the two receivers from the AgRover. With the 1.38m wheelbase, it can be roughly determined whether they are collecting reasonable data.

After these checks are completed, it is safe to start collecting and using the data from the two GPS receivers by clicking the button “Start recording” (marked in red). The button text changes to “Stop recording” during the trial, which can be used as an emergency button while the trajectory tracking is proceeding. The GPS data is stored temporarily in the computer’s RAM memory. The program creates and writes the data to a log file after the trial is completed.

4.6.2 Origin setting and initial vehicle information collection

Although the AgRover was powered by deep-cycle batteries, the power was not sufficient for the robot to work continuously for several hours. Due to the drifting GPS signal issue, which happened occasionally, it was helpful to survey the initial position of the robot at the beginning of each trial to ensure experimental accuracy. In the experiments presented in this paper, the initial location of the AgRover on the GPS UTM coordinate system was considered the origin of the virtual coordinate system, which was also used for reference path and waypoint creation. This coordinate survey directly affects experimental accuracy.

4.6.2.1 Origin survey under front-steering mode

Under front-steering mode, the origin was set at the center of the real axle; therefore, the stabilized location of the rear GPS receiver was used as the origin. The X axis and Y axis were parallel to their counterparts in the UTM coordinate system. Thus, initial heading angle $\psi_o$can be derived from the data received by the front and rear GPS receivers (Figure 4.15).

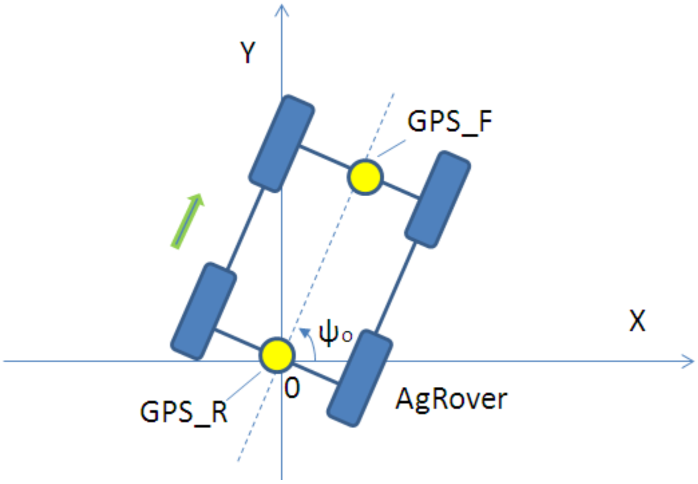


Figure 4.15: *Initialization of the origin and the vehicle's original heading under front steering mode*

4.6.2.2 Origin survey under coordinated steering mode

The survey process was similar under coordinated steering mode, and the heading calculation is the same as that of front-wheel steering. However, the origin is set at the geometric center of the vehicle, rather than the rear (Figure 4.16). This is due to the ability of both the front and rear wheels to steer at the same time and the center of the vehicle being the yaw center.

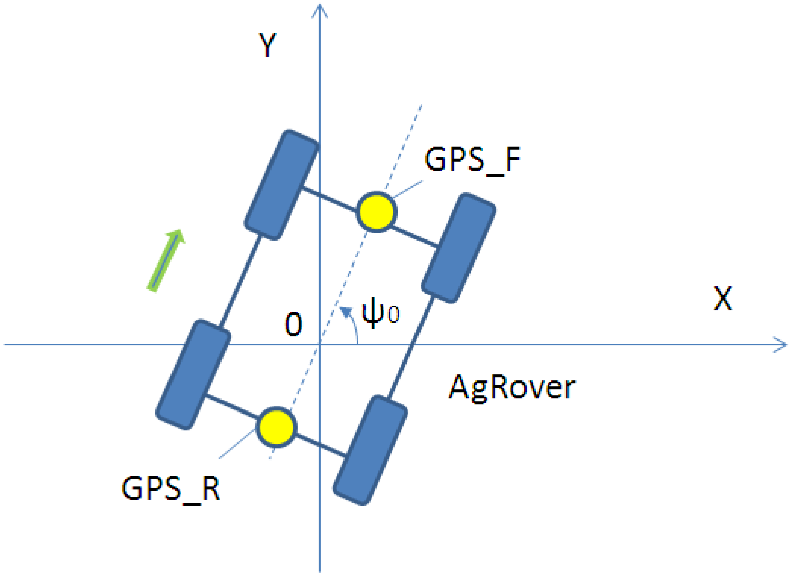


Figure 4.16: *Initialization of the origin and the vehicle original heading under coordinated steering mode*

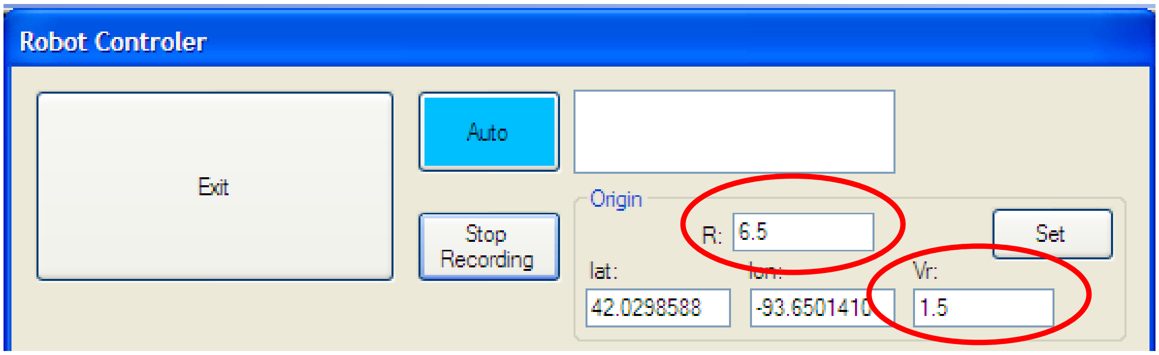
4.6.2.3 Origin setting

After the GPS data is ready and settled, the origin can be manually set by clicking the “Set” button on the interface window.

4.6.3 Reference path setting and waypoint selection

4.6.3.1 Reference path creation

Under the newly created local coordinate system, it was possible to map the reference path. Basic information about the reference path was defined on the robot control software interface. The software provided flexibility of the reference path setting, which improved working efficiency while working in the field. The example shown in Figure 4.17 demonstrates how to define a circle-shaped reference path with radius and velocity.



R: 6.5
lat: 42.0298588
lon: -93.6501410
Vr: 1.5

Figure 4.17: *An example of a circle reference path setting on the AgRover: "R" denotes the turning radius of the reference circle and "Vr" denotes the longitudinal velocity of the AgRover on the circle.*

4.6.3.2 Waypoint selection

As the vehicle's initial location and heading were defined in a new vehicle-based coordinate system, the reference path could be determined from the basic information offered by the software interface window. However, for practical trajectory tracking, it is not realistic to select the closest point on the reference path as the target. Because the designed SMC controller is sensitive to the system errors, x_e, y_e, and ψ_e, it would try hard to reach the target with all of the error contingent to zero. However, the realistic hardware working limit and delay might lead to an endless vibration in the vehicle system, and the target would never be reached in finite time. To accommodate this issue and provide a reasonable approach and tracking plan, a waypoint selection mechanism was proposed.

With the provided reference path information, the waypoints were generated and used by the SMC controller algorithm. A mechanism regarding how and when to select an appropriate waypoint to provide promising trajectory tracking performance was developed; Figure 4.18 helps explain this procedure.

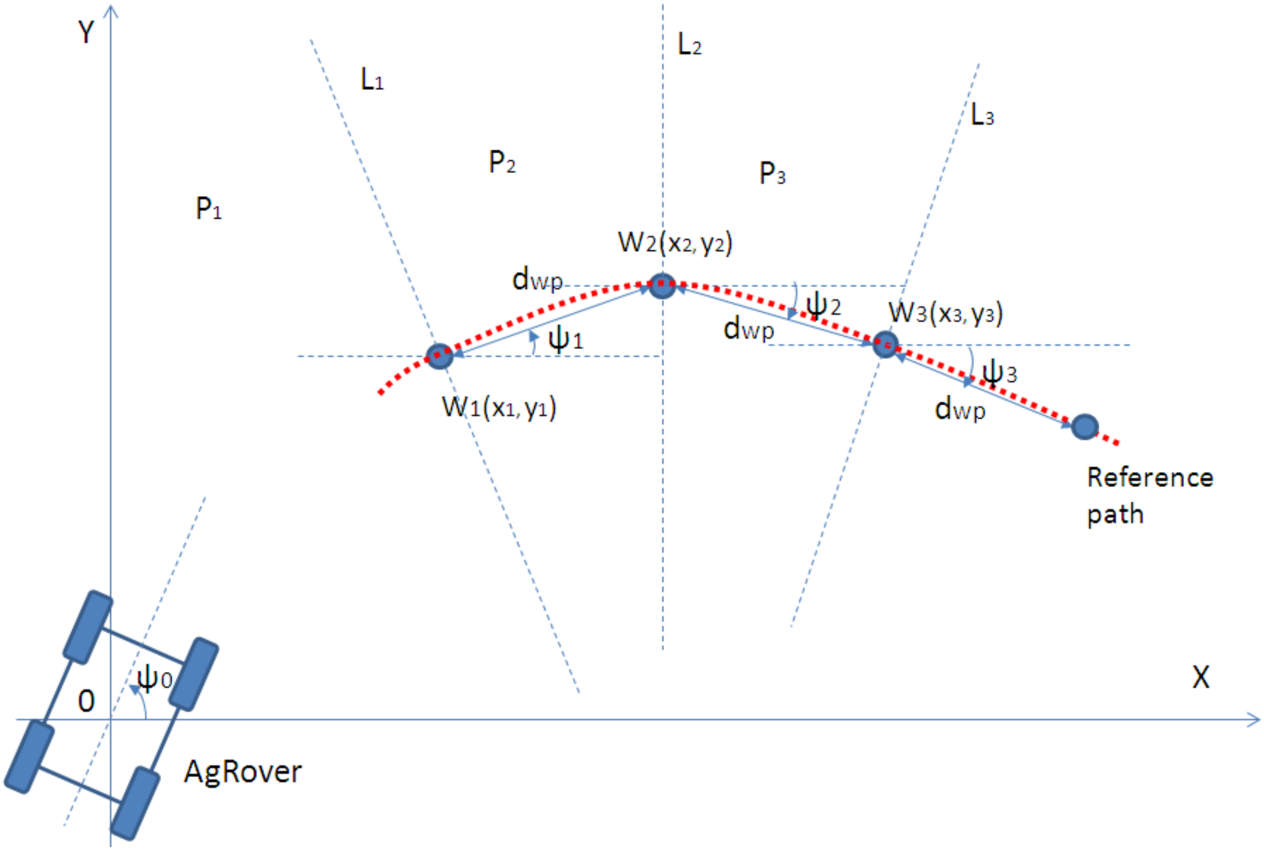


Figure 4.18: *Waypoint generations and selection diagram: the red dashed curve is the reference path; there are several waypoints (W_1, W_2, and W_3) on the red path.*

As shown in Figure 4.18, the dashed L_1, L_2, and L_3 lines are vertical to the connections between the waypoints, and they divide the area into P_1, P_2, and P_3 (three phases). The step length, d_{wp}, between the waypoints was $1m$, and it should be $1.5m$ on the straight-line reference path. If the AgRover was approaching the reference path, it started from approaching W_1. However, during this process, once the controller detected that the vehicle already exceeded L_1, and no longer stayed in P_1 but entered into P_2, $W_1$was replaced by W_2 as the target waypoint. This algorithm is further explained in the following section.

4.6.4 Implementation of waypoint updating mechanism

To realize the mechanism defined above in the AgRover controller, a corresponding algorithm was developed. Based on each waypoint as an origin, a sub-coordinate system was built up. While the robot approaches one of the waypoints, the coordinates of the robot location in that coordinate system are calculated. Once the robot has been detected across the vertical line, even though the target waypoint has not been reached, it has to be replaced with the next waypoint instead. The procedure is explained further in Figure 4.19.

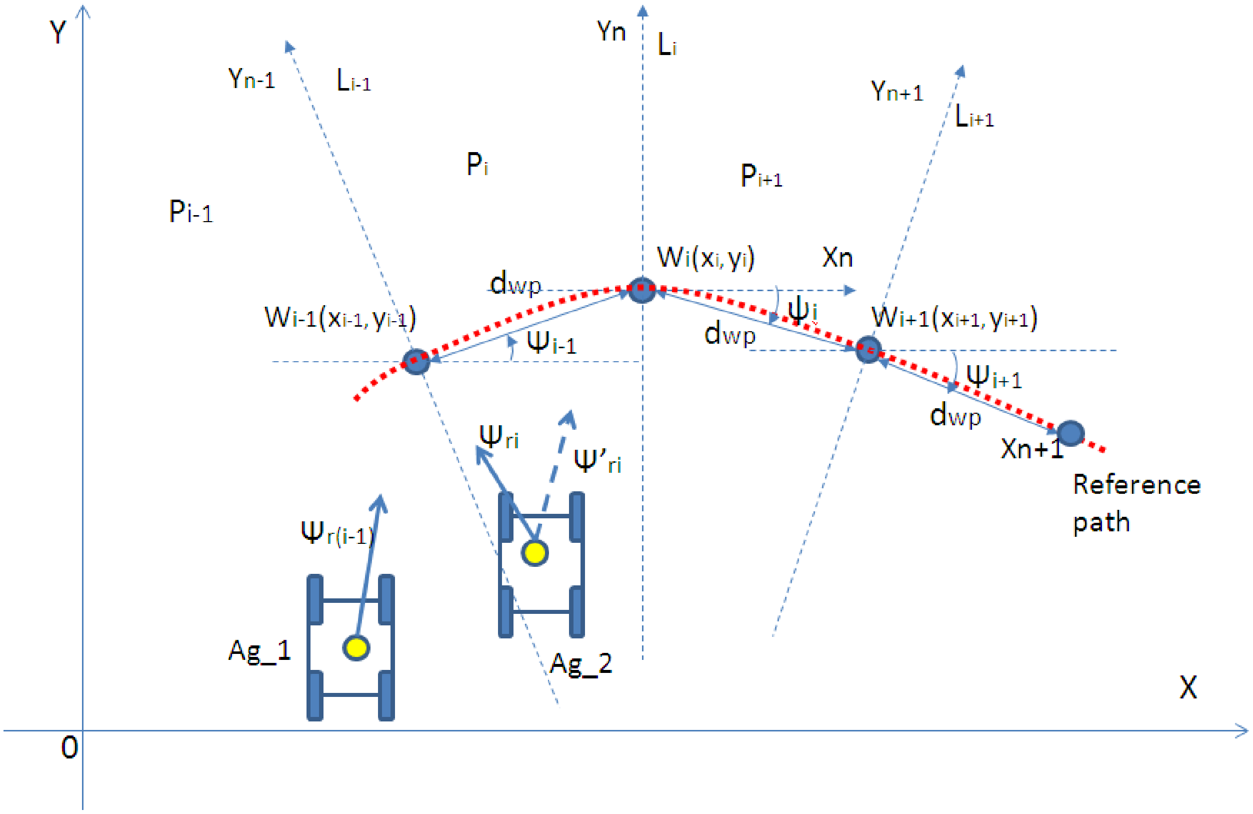


Figure 4.19: *Waypoint update mechanism diagram: Ag_1 and Ag_2 represent the two statuses of the AgRover while it approaches* W_{i-1}. $\psi_{r(i-1)}$, ψ_{ri}, *and*$\psi_{r(i+1)}$ *are the calculated heading angles that will be implemented in the lower level controllers.*

As AgRover was at Ag_1 and operating in the P_{i-1} phase, the vehicle was located at the left side of the virtual Y_{n-1} axis, and there is no doubt that the x_{n-1} coordinate of Ag_1 at the $W_{i-1}(x_{i-1}, y_{i-1})$ coordinate system is negative. The target waypoint continued as W_{i-1} in this case.

When AgRover moved to Ag_2, however, it was operating under another condition. To clarify the problem, it was necessary to calculate the x_{n-1} coordinate under the $X_{n-1} - Y_{n-1}$ coordinate system:

$$x_{n-1} = (x_{Ag\_2} - x_{i-1})\cos\psi_{i-1} + \left(y_{Ag\_2} - y_{i-1}\right)\sin\psi_{i-1} > 0\ . \tag{4.44}$$

It could be concluded from the last equation that the AgRover was at the right side of the virtual Y_{n-1} axis, so the target waypoints needed to be updated from W_{i-1} to W_i. Likewise, target heading angle ψ_{ri}, for the lower level controllers to execute, had to be updated as ψ'_{ri}. Thus, the same calculation was performed again to make sure $x_n \leq 0$:

$$x_n = (x_{Ag\_2} - x_i)\cos\theta_i + \left(y_{Ag\_2} - y_i\right)\sin\theta_i \tag{4.45}$$

$$\psi_{ri} = \psi'_{ri}\ . \tag{4.46}$$

If condition $x_n > 0$ still existed, then the target waypoint kept updating by W_{i+1}. The difference between (3.44) and (3.45) is that the slope angle of X_n axis

$$\theta_i = \frac{\psi_{i-1}+\psi_i}{2}\quad . \tag{4.47}$$

However, to apply to the first and last waypoints, the slope angle is consistent as the reference heading angle:

$$\theta_0 = \psi_0, \theta_f = \psi_f. \tag{4.48}$$

Thus, the preparation at the reference path and waypoints has been accomplished.

4.6.5 Safety redundancy design

The scope of the safety design is a large one, which is very important for experimental platform design. It includes mechanical safety design, electrical safety design, and operational and maintenance safety design. As introduced in Chapter 2, mechanical and electrical safety was involved in the AgRover design. In this section, operational safety, especially operation emergency stop under automatic navigation mode, will be presented and discussed. It was mentioned in the automatic navigation working procedure, represented by the stop sign in Figure 4.13.

4.6.5.1 Self-diagnosed emergency stop

As introduced in Chapter 2, the AgRover had self-diagnosis capability under the manual control mode, and it was also valid under the automatic navigation mode. To enhance hardware safety, an RTK-GPS signal was involved and contributed to safety design. The circumstances that can cause emergency stop are summarized in Table 4.5.

Table 4.5: *Self-diagnosed situations that might lead to emergency stop*

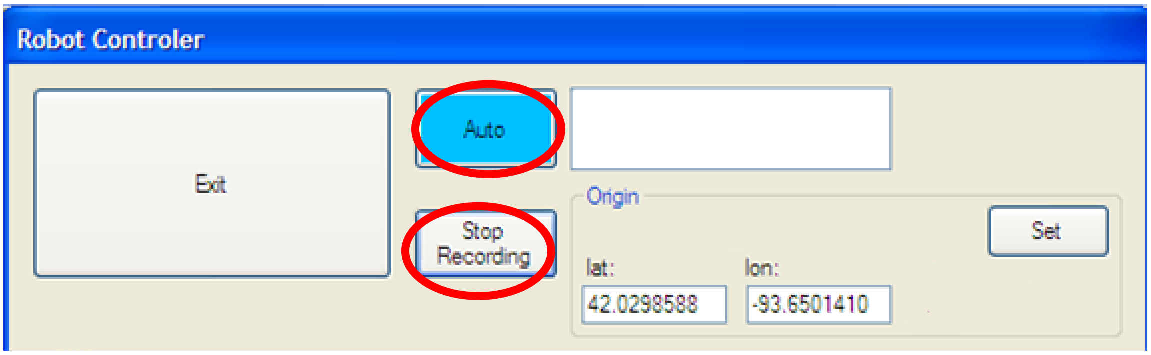
No.	Description	Reaction
1)	Steering motor or encoder failure	AgRover stop and alarm on
2)	Driving motor or encoder failure	AgRover stop and alarm on
3)	One or more wheels stuck	AgRover stop and alarm on
4)	Battery power low	AgRover stop and alarm on
5)	Wireless communication with PC dropped	AgRover stop and alarm on
6)	GPS signal shaded or quality fails	AgRover stop
7)	Waypoint is invalid as defined	AgRover stop

The circumstances listed in Table 4.5 are all realistic situations that occurred during the previous study and experiments conducted by the author. All of them played important roles in protecting the AgRover and nearby personnel.

4.6.5.2 Mandatory emergency stop

The AgRover had a strong self-protection mechanism. However, there were still numerous possibilities wherein unpredictable mistakes or external interrupts could occur under automatic navigation, which needed to be prevented. Especially while in controller troubleshooting and debugging mode, the robot did not have the ability to indicate any setting mistakes. Moreover, when the hardware limit was at risk of being exceeded, the AgRover had to be forced to stop before anything worse could occur. There were multiple options to stop the robot, which are introduced in this section.

4.6.5.2.1 Emergency stop on AgRover operation interface



lat: 42.0298588
lon: -93.6501410

Figure 4.20: *Emergency buttons on the AgRover operation interface*

Either “Auto” or “Stop Recording” could stop the automatic navigation mode from “working” to “idle.” However, these two options were only able to work if the wireless remote desktop was still accessible and the software was working properly.

4.6.5.2.2 Emergency stop on remote monitoring station

The stop buttons on the joysticks were usually handled by the index finger; this was actually the easiest way to stop the vehicle. After the vehicle fully stopped, the working mode and the motors could all be reset with the “Motor reset” button.

There were also two buttons on the remote monitoring software interface that could perform an emergency stop—“Emergency Brake” and “Automatic” (marked on Figure 4.21). Either of these buttons could stop the automatic working cycle and return the robot to manual control mode.

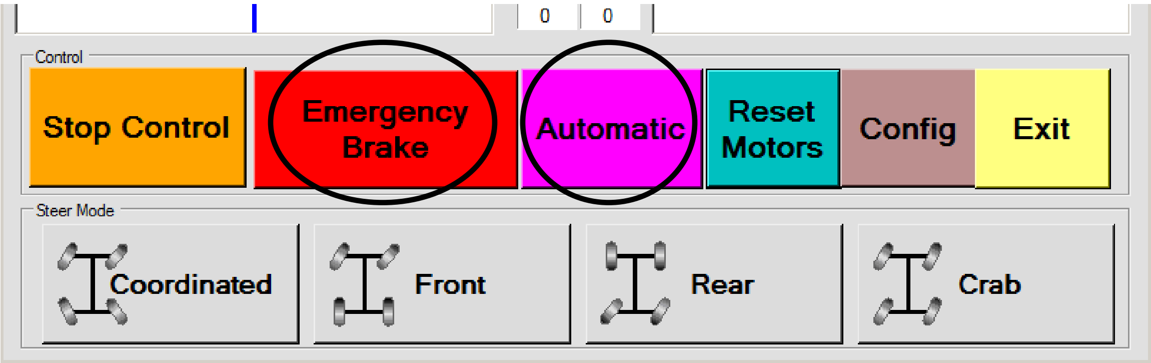


Figure 4.21: *Emergency stop buttons on the PC remote monitoring software*

4.7 Results and Discussion

4.7.1 Controller Parameter Setting

As derived in Section 4.4, the control signal output of the SMC controller was

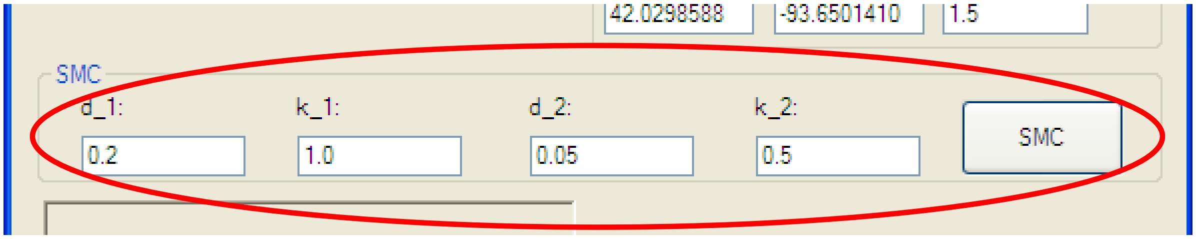
$$q = \begin{bmatrix} u \\ r \end{bmatrix} = \begin{bmatrix} y_e r + \ u_r cos\psi_e + k_1 \frac{s_1}{|s_1|+\sigma_1} \\ \frac{r_r + \frac{y_e}{1+(u_r y_e)^2}\dot{u}_r + \frac{u_r}{1+(u_r y_e)^2} u_r sin\psi_e + k_2 \frac{s_2}{|s_2|+\sigma_2}}{1+\frac{u_r}{1+(u_r y_e)^2} x_e} \end{bmatrix}. \qquad (4.49)$$

The functionalities of the four control parameters were different, but they were coupled, affecting the dynamic performance of the control result.

Table 4.6: *Functionalities of control parameters*

Affected signal	Parameter	Functionality
u	k_1	Determine approaching speed to $s_1 = 0 \Leftrightarrow x_e = 0$
	σ_1	Consistency when penetrating sliding surface $s_1 = 0$
r	k_2	Determine approaching speed to $s_2 = 0 \Leftrightarrow \psi_e = -\tan^{-1} u_r y_e \Leftrightarrow \psi_e = 0, y_e = 0$
	σ_2	Consistency when penetrating sliding surface $s_2 = 0$

From Table 4.6, it can be concluded that k_1 and k_2 were critical to the system approaching speed to the zero point, and that σ_1 and σ_2 were sensitive to the system dynamic performance while the robot was working within the target trajectory neighborhood. Because the convergence of the system was proven in Section 4.4, there was no concern that the value of the control parameters might destroy the system's global stability. However, the dynamic performance variance within finite time by different control parameters still needed to be investigated.



42.0298588 | -93.6501410 | 1.5

SMC

d_1: 0.2

k_1: 1.0

d_2: 0.05

k_2: 0.5

Figure 4.22: *Control parameter setting on AgRover operation interface*

Although the relationship between the control signal and parameters appeared to be independent in Equation (4.49), the coupled relationship among the system state variables made the system a high-dimensional MIMO nonlinear system. It was difficult to study the sensitivity of each parameter to the system dynamic performance theoretically. SMC had already proved its robust capability, which instilled confidence in the subsequent off-road experiments.

All of the trajectories could be decomposed by straight lines, curves, and circles, where curves could be treated as special circle parts. Therefore, straight lines, circle curves, and their combinations were usually chosen as reference trajectories to validate the effectiveness of a controller. In the following discussion, each of the cases is studied and results are presented.

4.7.2 Straight-line trajectory tracking

Straight-line trajectory was a common case if the AgRover was required to follow a crop row in a field. As assumed and shown in Figure 4.27, on the virtual coordinate system, the robot was parked at the origin $(0, 0)$, and the initial heading angle was -1.73. The start point of the straight line was$(-15.45, -22.84)$, and the slope angle was -1.77. Therefore, the initial orientation of the AgRover was almost parallel to the path. In this case, the AgRover was working under front-steering mode.

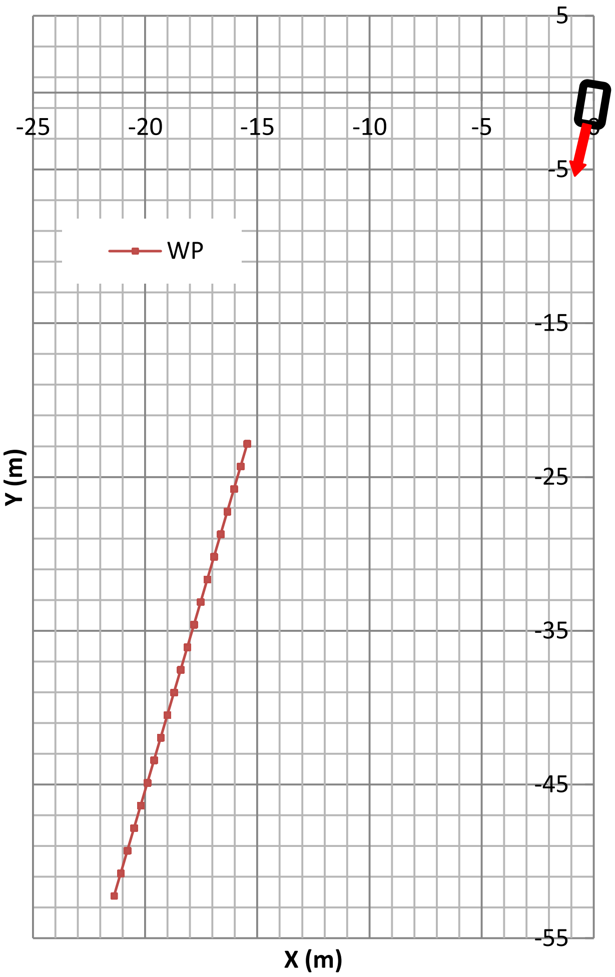


Figure 4.23: *Straight-line reference trajectory and initial location and heading of AgRover*

After running SMC automatic navigation with the AgRover, the tracking trajectory shown in Figure 4.24 was attained.

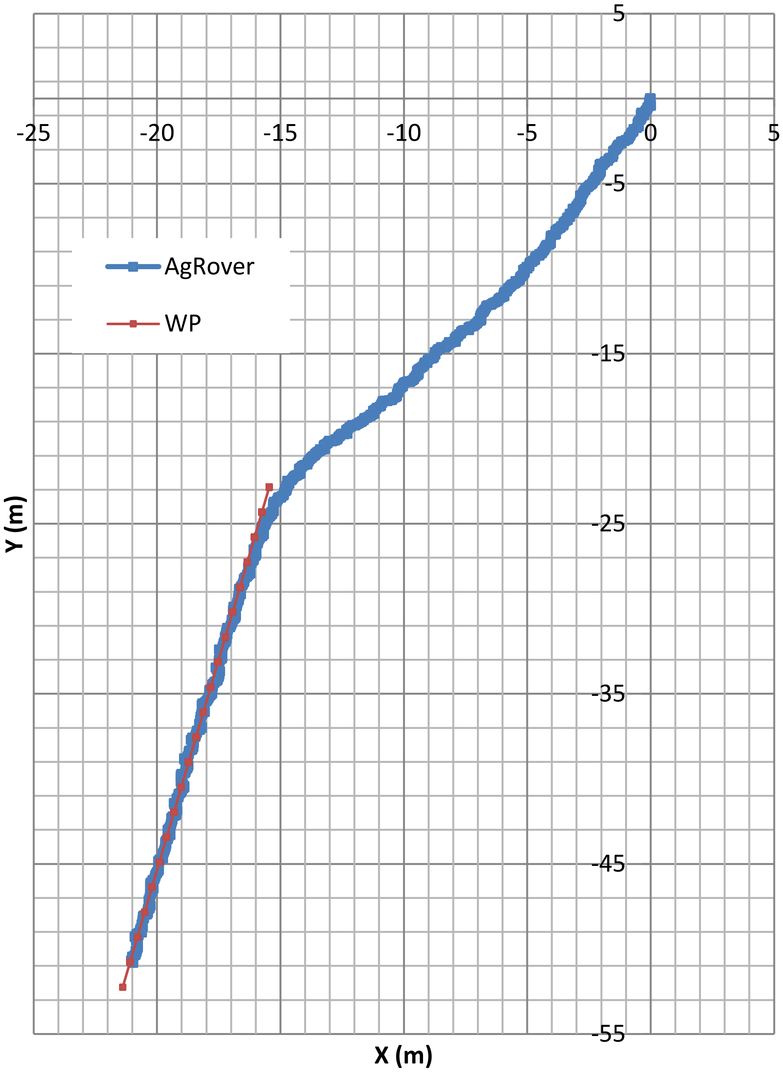


Figure 4.24: *Straight-line trajectory tracking result under front-steering mode*

The average tracking velocity on the line was $0.72\ m/s$. To investigate the dynamic performance, the offset error shown in Figure 4.25 was observed.

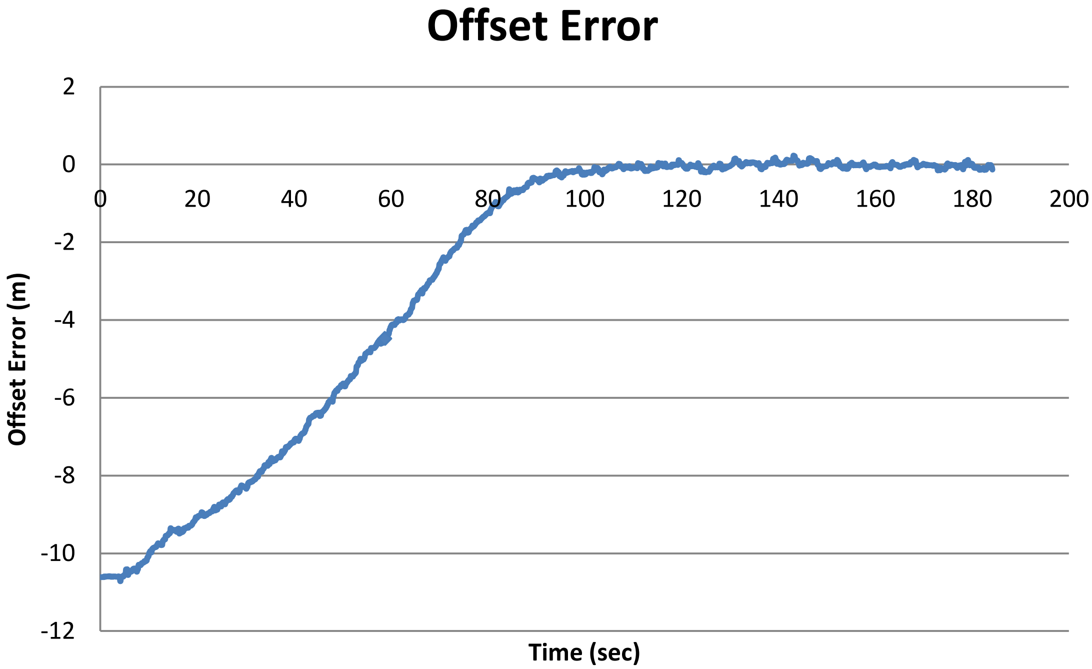


Figure 4.25: *Offset error of straight-line trajectory tracking under front-steering mode*

The root mean square error (RMSE) of this straight-line trajectory tracking was $0.079\ m$. The working performance of this tracking trial was promising.

4.7.3 Circle trajectory tracking

Circle trajectory was another common example of field operation. It existed in various turning scenarios, such as on-road/off-road turning and headland turning. Because any complex curve can be decomposed into combinations of circles, a satisfactory circle-turning performance was required for the capability of complex trajectories.

As illustrated in Figure 4.30, on the virtual coordinate system, the robot was assumed parked at the origin $(0, 0)$, and the initial heading angle was 1.26 rad. The start point of the circle was$(-8.5, 4.0)$, with a turning radius of $6.5m$, and the center of the circle was $(-13.04, -0.5)$. The AgRover was working under coordinated steering mode, which had an advantage on the sensitivity of the heading angle.

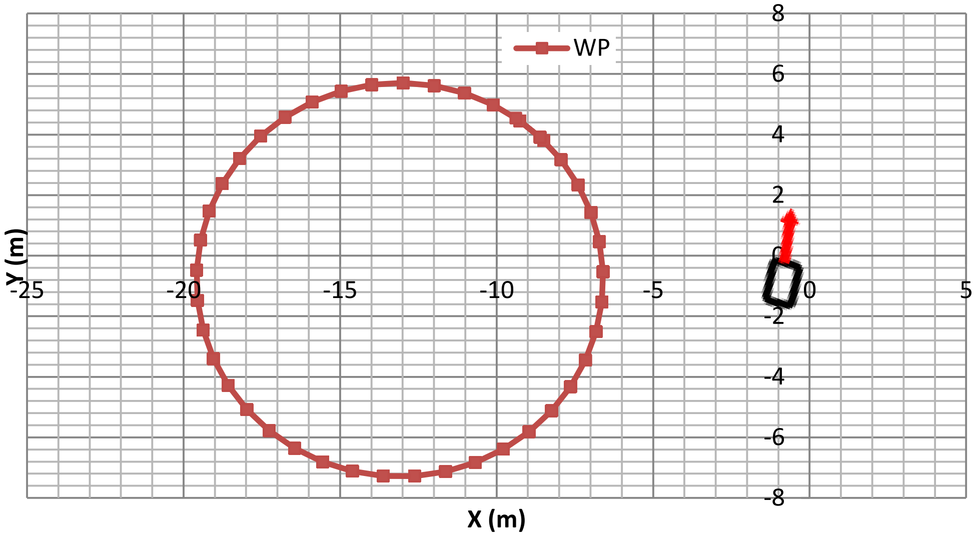


Figure 4.26: *Circle reference trajectory and initial location and heading of AgRover*

By running the SMC controller with the coordinated steering model, the tracking trajectory showed in Figure 4.27 was achieved.

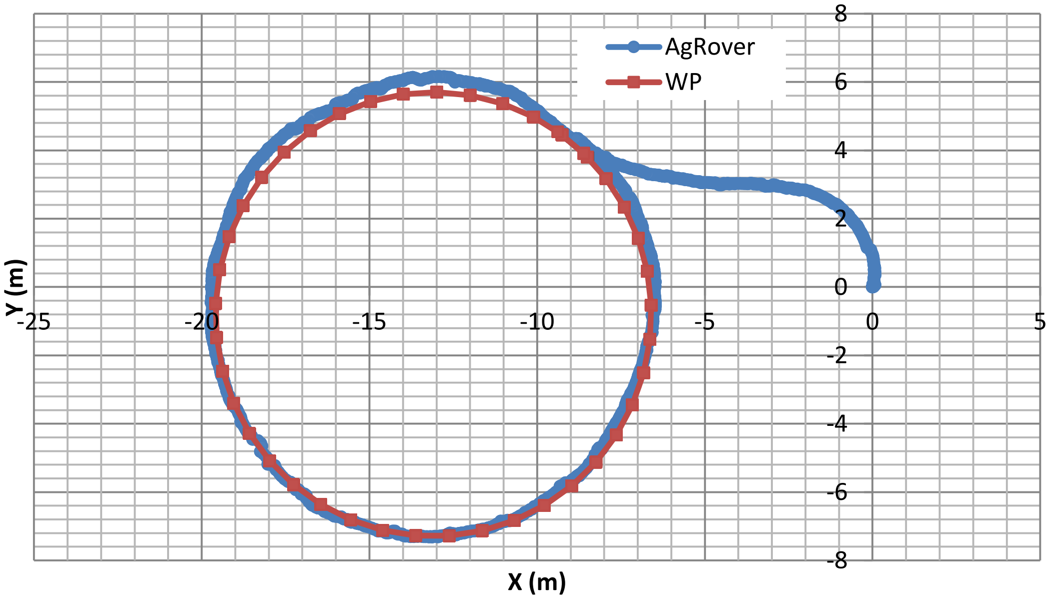


Figure 4.27: *Circle trajectory tracking result under coordinated steering mode*

The approaching process of the coordinated steering was smooth and accurate. To be able to assess the details of the dynamic approaching performance, the offset error, shown in Figure 4.32, was observed as well.

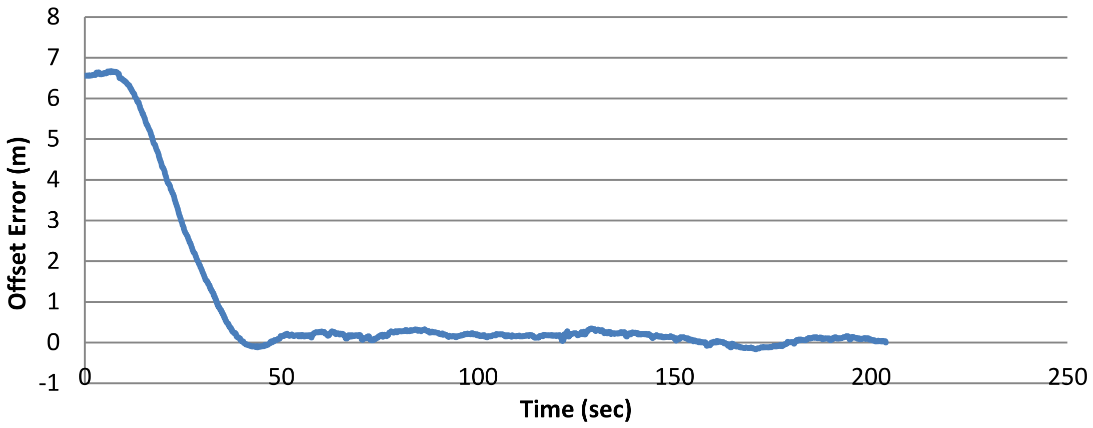


Figure 4.28: *Offset error of circle trajectory tracking under coordinated steering mode.*

The average velocity on the circle was $0.5\ m/s$, and the RMSE was $0.17\ m$.

4.7.4 "∞"-shaped trajectory tracking

Based on the experience of circle trajectory tracking, it was reasonable to see the tracking performance when the reference trajectory had two circles side by side. Under the assumption, the AgRover was originally at origin $(0, 0)$, and the initial heading angle was 1.51. The start point of the circle was $(-15.9, 4.58)$, with a radius of $6.5m$, and the center of the circle was $(-20.5, 0)$. The other circle was tangential to the first one, with the same radius $(6.5m)$, and the center was at $(-7.5, 0)$. The AgRover was working under coordinated steering mode.

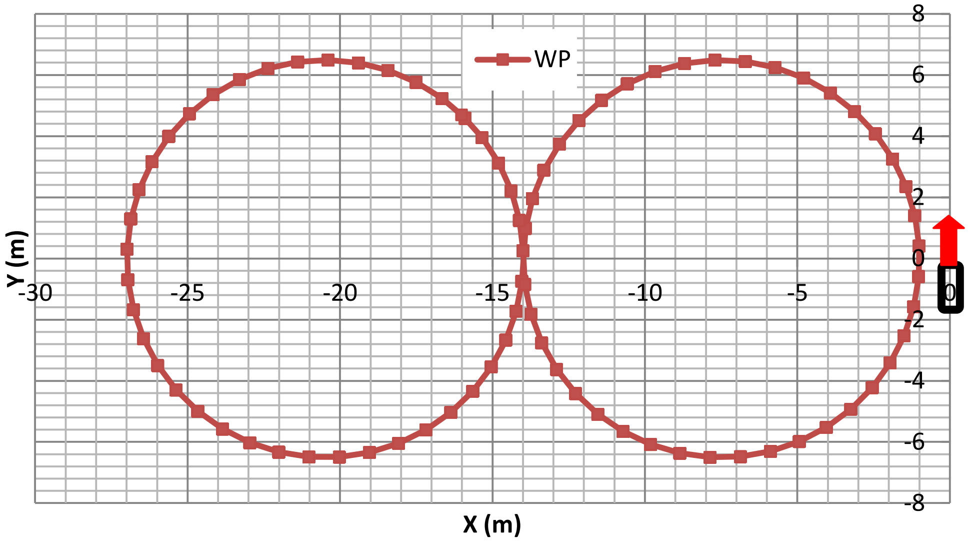


Figure 4.29: *"∞"reference trajectory and initial location and heading of AgRover*

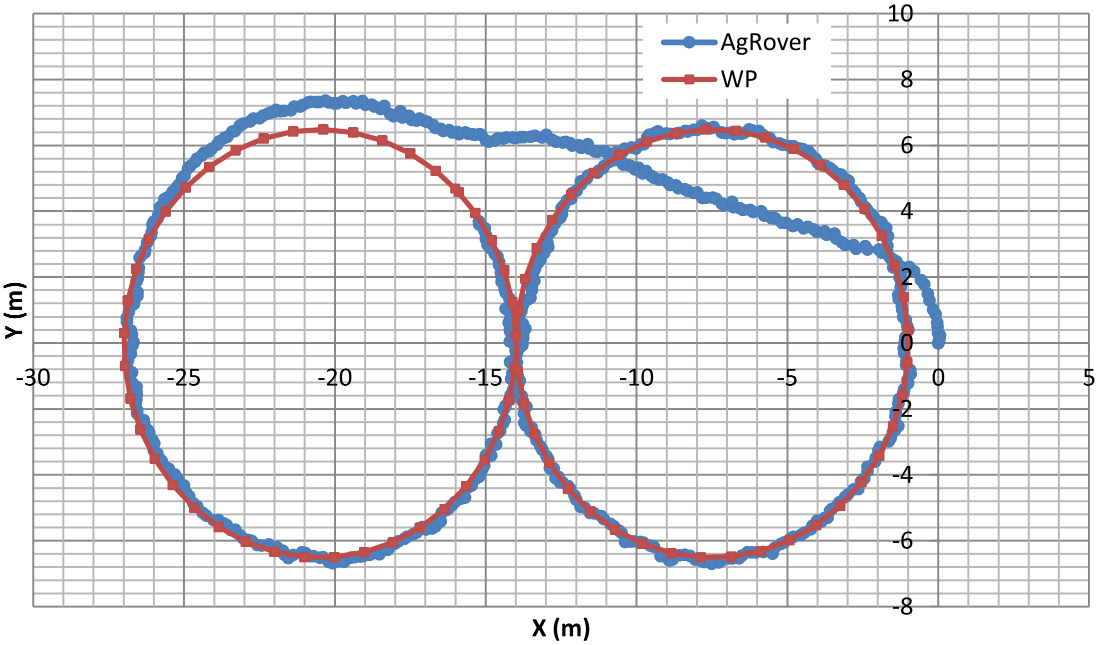


Figure 4.30: *"∞" trajectory tracking result under coordinated steering mode*

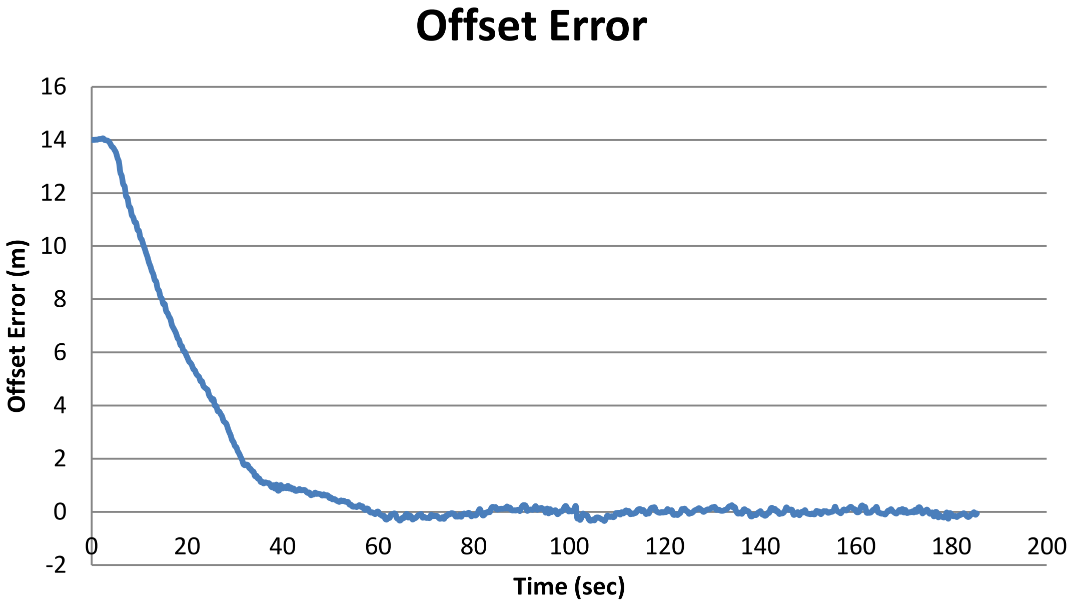


Figure 4.31: *Offset error of "∞" trajectory tracking under coordinated steering mode*

The RSME on the trajectory was $0.12\ m$, and the average velocity was $0.58\ m/s$.

4.7.5 Complex trajectory tracking

The "U" turn is a common headland turning circumstance that contains straight lines and curves. To finish this type of complex trajectory, a "hybrid" steering mode was proposed and adopted, in which the AgRover worked under front steering on the straight lines and under coordinated steering on the curves. The AgRover was originally at origin $(0, 0)$, and the initial heading angle was -2.74.

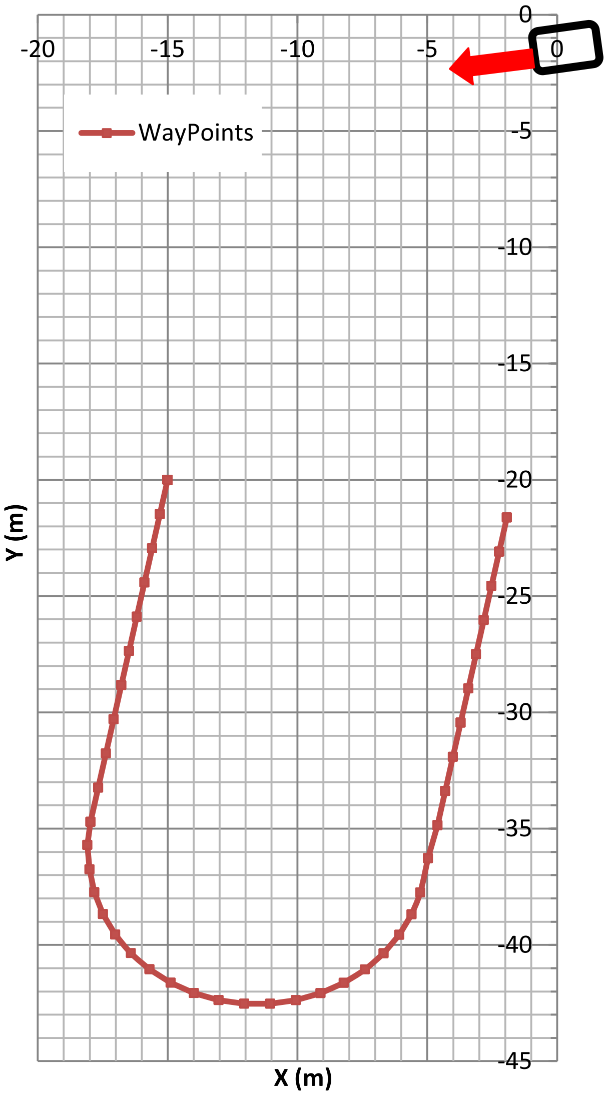


Figure 4.32: *"∞" reference trajectory and initial location and heading of AgRover*

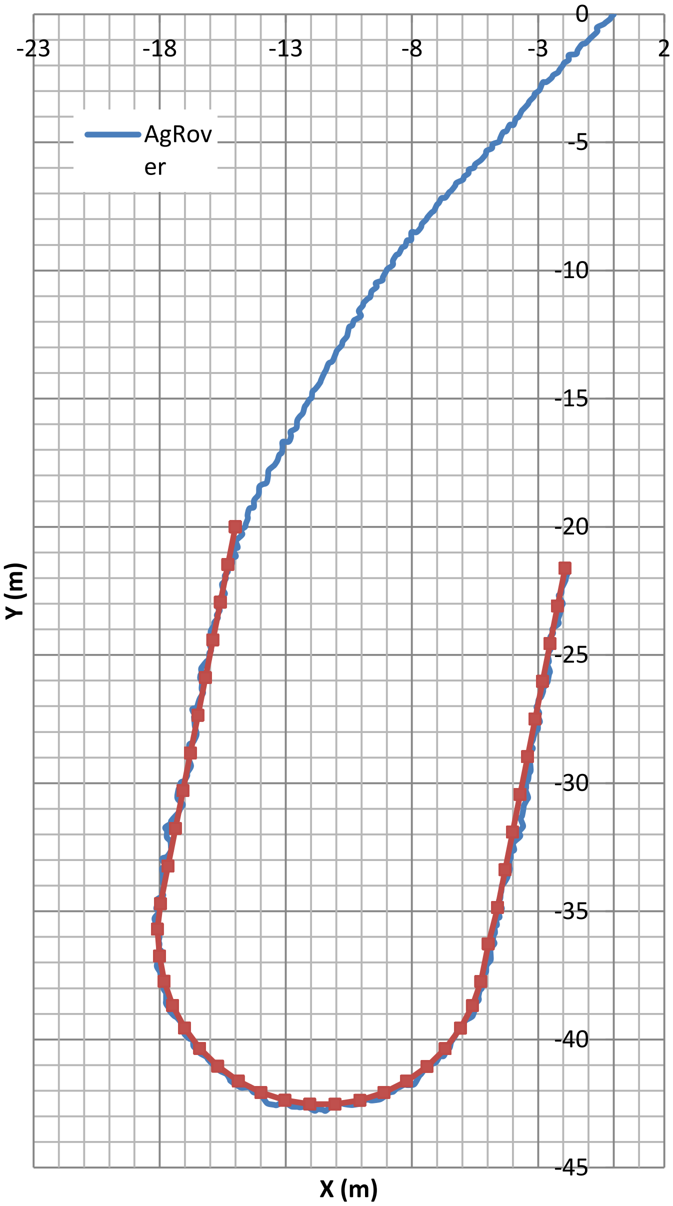


***Figure 4.33**: "Hybrid" turning mode performance in a U-turn trajectory*

The curve turning radius was 6.5 m. The RSME on the straight lines were 0.11 m and 0.12 m, respectively, and it was 0.13 m on the curve. The average speed was 0.54 m/s on the reference trajectory.

4.8 Conclusions

SMC was proven to be an effective tool, both theoretically and practically. It was studied and investigated by previous research work and in this chapter as well. It was successfully implemented on the AgRover to accomplish automatic navigation control of trajectory tracking tasks. The quantified experimental results of the field experiments were promising and impressive.

However, SMC is not perfect. The error rate was consistently between $0.1m$ and $0.2m$, which could be improved. It is expected that with improved hardware responsiveness and sampling frequency, the controller performance could be improved correspondingly. The “chattering” issue existed in most of the experiment results, which was shown in the tracking trajectory. Improved adaptive control and fuzzy logic control can help and it can be implemented with an SMC controller to solve and ease this issue. The control parameters were sensitive to the system dynamic performance. In addition, computing the sensitivity and the relationship among individual control parameters could be a potential research interest.

In conclusion, SMC performed well in AgRover off-road tests. It showed the potential to adopt a vehicle dynamic model to the controller design if the hardware and feedback capability allows. It is predictable that with computer and electronic technology development, SMC will be able to play an important role in autonomous agricultural vehicle navigation control.

References

Cheng, J., Yi, J., Zhao, D., (2007) Design of a sliding mode controller for a trajectory tracking problem of marine vessels. Control Theory & Applications; 1(1): 233-237.

Huang, Y. J., Kuo, T. C., and Way, H. K., (2003) Robust vertical takeoff and landing aircraft control via integral sliding mode, Proc. Inst. Electr. Eng.—Control Theory Appl., vol. 150, no. 4, pp. 383–388.

Hung, J.Y., Gao, W., and Hung, J. C., (1993) Variable structure control: A survey, IEEE Trans. Ind. Electron., vol. 40, no. 1, pp. 2–22.

Liu, J., (2005) MATLAB Simulation for Sliding Mode Control: Tsinghua University Press.

Shen, P., Lin, F., (2005) Intelligent backstepping sliding-mode control using RBFN for two-axis motion control system. IEE Proceeding electrical Power Application; 152(5): 1321-1342.

Utkin, V.I.,(1992) Sliding Modes in Control and Optimization.NewYork: Springer-Verlag.

Wu, W., Chen, H., and Wang, Y., (2001) Global trajectory tracking control of mobile robots, ACTA Automatica SNICA, vol. 27, no.3.

CHAPTER 5: HEADLAND TURNING OPTIMIZATION FOR AGRICULTURAL VEHICLES

A paper to be submitted to *the Transactions of the ASABE*

Xuyong Tu, Lie Tang

5.1 Abstract

An essential part of agricultural vehicle navigation control and field coverage path-planning process, agricultural vehicle headland turning is a process that has given initial and final conditions. It should be conducted in a manner that minimizes time and travel, in order to maximize operational efficiency. To optimize and automate the turning process, vehicle technical specifications and headland topographic features should be considered and respected. However, it is challenging to solve this dynamic nonlinear optimization problem with traditional indirect numerical methods. Improved computer technologies provide the possibility and capability to use direct numerical methods to solve such a problem in restricted parameter spaces with constraints. By using optimization software such as TOMLAB®, simulation results were generated for diverse circumstances of the tractor/trailer headland-turning scenario.

Keywords: headland turning optimization, trajectory optimization, direct numerical methods, agricultural vehicles

5.2 Introduction

By January 2013, there were 2,286,736 farms with 400,830,068 planted acres across the United States (Farm Service Agency, USDA, Jan. 2013), which is a dramatic increase from 242,609,961 acres in 2007. However, the number of hired farm workers (which make up one-third of all those working on farms) has held relatively steady, from

1,032,000 in 2007 to 1,062,000 in 2012 (Farm Labor Survey [FLS], USDA). To improve operational efficiency and realize better farm economy, agricultural automation is absolutely required and expected.

With the tremendous development of unmanned technology in modern agriculture, automated path planning plays an important role in agricultural field operation, and agricultural vehicle headland turning is a significant part of this process. The headland turning process should be conducted in a minimum amount time and travel within restricted off-road conditions (Jin, 2009). The optimization of headland-turning control can help substantially improve time and fuel efficiency and maximize field use efficiency, and it might simply affect the generation of the planned path. Therefore, to improve the field efficiency of auto-steer mobile agricultural field equipment, optimization of their headland-turning trajectory is of great interest to agricultural equipment manufacturers. However, the difficulties of headland-turning optimization are also obvious. The off-road vehicle model is complex and nonlinear; therefore, well-developed linear optimization algorithms are not able to solve the problem (Oksanen, 2004). In addition, attached trailers increase the complexity of the model and make the dynamic nonlinear problem more challenging to solve.

The numerical solutions to the trajectory optimization problem can be categorized into two approaches—indirect methods and direct methods (Conway, 2012). Indirect solutions use analytical necessary conditions from the calculus of variations, which requires the addition of the co-state variables of the problem. However, when using indirect methods, the derivation of the necessary conditions, including differential equations, boundary conditions, and path constraints, is usually a complicated mathematical procedure. A minor change in any of these constraints can cause considerable amounts of computation, even with the help of software such as POST® (Program to Optimize Simulated Trajectories; Sierra Engineering Inc., Carson City, NV) or GTS (Geostatistical Temporal-Spatial algorithm; The Aerospace Corporation, El Segundo, CA). Conversely, direct solutions convert such types of optimization problems into a nonlinear programming (NLP) problem, which transcribes a continuous optimal control problem into a parameter optimization problem (Conway and Paris, 2011; Ross and Fahroo, 2003; Fahroo and Ross, 2002). An NLP method integrates the system equations stepwise, using either implicit or explicit rules. Compared to indirect methods, considerable advancements have been made in direct methods over the last decade (Conway, 2012). Commercial software packages, such as TOMLAB® (Tomlab, Stockholm, Sweden), DIDO® (Elissar Global, Carmel, CA), and SOCS® (Boeing, Seattle, WA), are available to facilitate the implementation of direct methods for solving optimal control problems. Therefore, the exploration of applying direct methods to agricultural field equipment headland-turning trajectory optimization represents a potential significant contribution toward this long-standing problem, which is important to both our research community and the equipment industry.

In this paper, a framework for tractor headland-turning optimal control is provided. The presented optimal headland-turning research was restricted to a 2-D plane. This study investigated the following three models: 1) tractor only, 2) tractor-implement, and 3) tractor-implement-implement. The minimum-time optimal control problem of headland turning in the three different models was studied and discussed. The optimization simulation was conducted using MATLAB with a TOMLAB toolbox.

5.3 Research Question Statement

The headland-turning problem could be described by the schematic shown in Figure 5.1, where the areas outside of the headland are considered a "prohibited area." Because it was assumed that the situation was known, the basic information about the environment and the vehicle was also considered as known, and it was the basis of the future discussion.

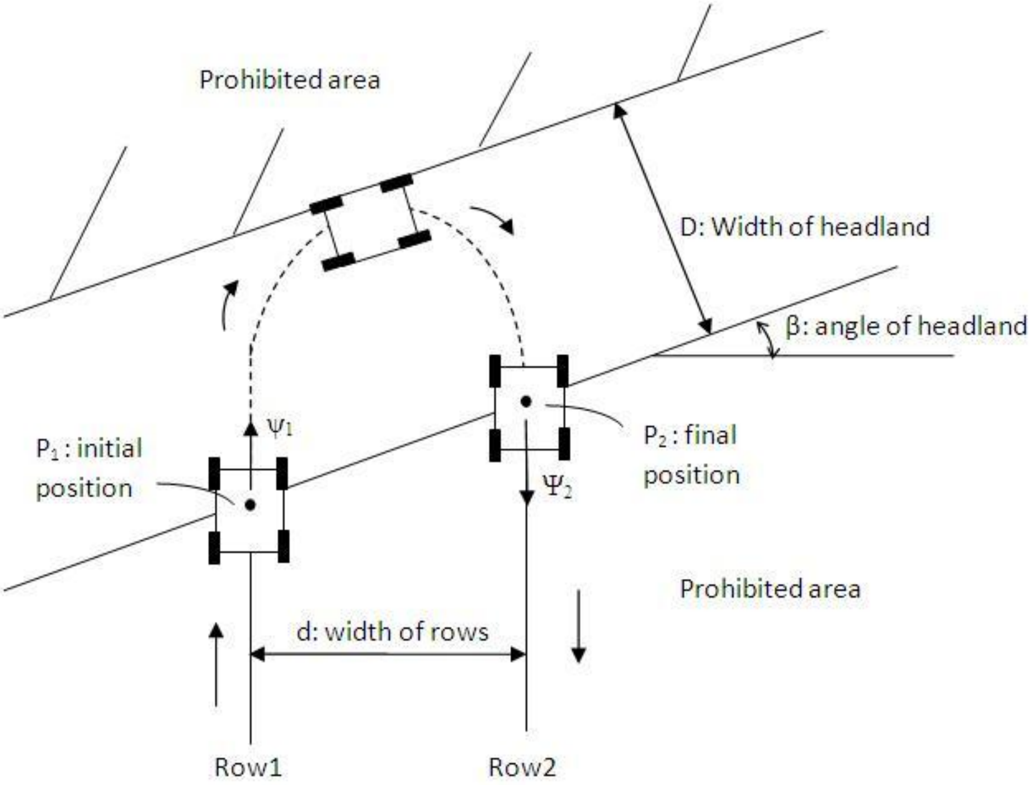


Figure 5.1: *Headland turning schematic*

As illustrated in the schematic (Figure 5.1), the headland was located between the two prohibited areas, where the borders were assumed to be parallel straight lines. D denotes the width, and the angle between the border and the orthogonal direction of the path is denoted as β. The vehicle was supposed to go from the initial point, $P1$, to the final point, $P2$, both located on the lower border line. The initial heading angle, $\psi1$, and the final heading angle, $\psi2$, were lined up with the vehicle infield paths, but they were of opposite orientations. During the turning process, the trajectory was restricted to the headland area, which meant that the vehicle was not allowed to traverse into the prohibited areas.

Some of the diverse optimization objectives associated with this scenario are:

- Minimum time with fixed velocity
- Minimum time with variable velocity
- Minimum fuel consumption with open time
- Shortest trajectory with fixed time

In this study, minimum time problems with fixed and variable velocities were selected as the optimization objectives and investigated. A time-optimal control framework was established, and the steering control sequence was designed accordingly.

5.4 System Modeling

5.4.1 Optimal Headland-Turning Problem Formulation

Based on the description in Section 5.3, the initial and final conditions were mapped on a two-dimensional coordinate system, where the initial point was set as the origin of the coordinate system, and the initial heading, $\psi1$, was considered the direction of the Y axis.

Table 5.1*: Parameters used in Headland Turning Optimization modeling.*

Parameters	Description (unit)
D	Headland width (m)
d	Row width (m)
β	Angle of headland boundary (rad)
Ψ	Heading angle of vehicle (rad)

All parameters described in Table 5.1 are further depicted in Figure 5.2. One of the assumptions of the model was that the edges of the headland were considered to be two parallel straight lines. This concept is applied and discussed throughout this paper.

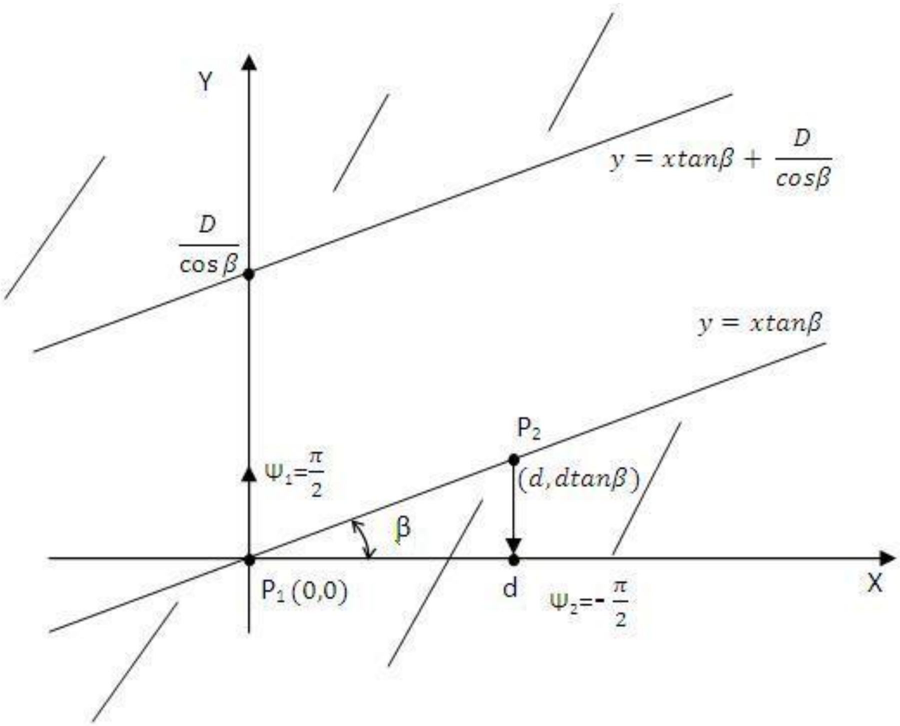


Figure 5.2: *Virtual coordinate system*

The original position of the headland-turning process was set at the origin of the XY coordinate system. Thus, the initial condition could be stated as

$$\begin{cases} \psi_o = \psi_1 = \frac{\pi}{2} \\ P_o = P_1 = (x_o, y_o) = (0,0) \end{cases} , \tag{5.1}$$

and the final condition could be stated as

$$\begin{cases} \psi_f = \psi_2 = -\frac{\pi}{2} \\ P_f = P_2 = (x_f, y_f) = (d, d\tan\beta) \end{cases} . \tag{5.2}$$

As indicated in Figure 4.2, the working area was restricted by two parallel straight lines, which were defined as

$$\begin{cases} x\tan\beta - y \le 0 \\ -x\tan\beta + y - \frac{D}{\cos\beta} \le 0 \end{cases} . \tag{5.3}$$

These inequalities were considered to be constraint conditions. Some other constraints, such as steering angle limit ($-\delta_{max} \le \delta \le \delta_{max}$) and constant velocity ($v = v_o = v_f$), were also considered in the optimization process.

Further, the constraints were summarized as

$$\begin{cases} -\delta_{max} \le \delta \le \delta_{max} \\ x\tan\beta - y \le 0 \\ -x\tan\beta + y - \frac{D}{\cos\beta} \le 0 \end{cases} . \tag{5.4}$$

Thus, all initial, final conditions and constraints of this headland-turning trajectory optimization problem were formulated mathematically. The next step is vehicle modeling.

5.4.2 Tractor Kinematic Modeling

The farm tractor is one of the most common agricultural vehicles in daily field operations. In this study, the tractor and its implements were chosen as the prototypical example for the headland-turning optimization problem. Thus, front-wheel steering was the designated steering mode in the vehicle modeling. A schematic tractor model was mapped onto a two-dimensional world coordinate system, shown in Figure 5.3.

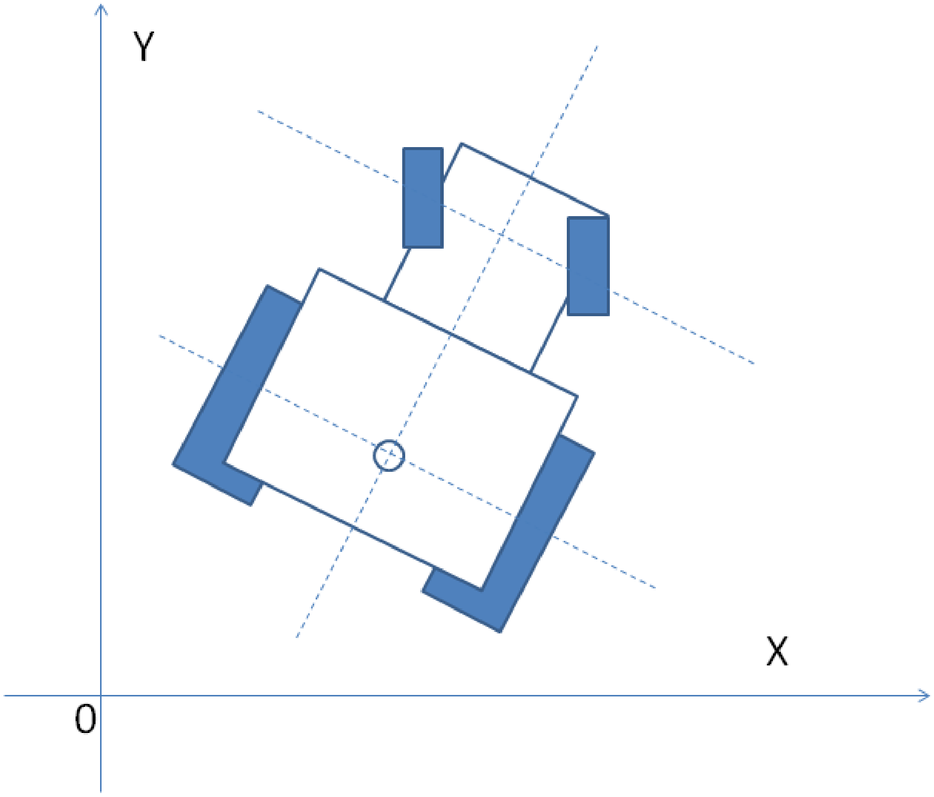


Figure 5.3: *Schematic representation of a tractor in a two-dimensional coordinate system*

In the headland-turning scenario, the tractor is usually driving at a relatively low speed to avoid an under-steering situation and reduce vibration. With this assumption, the system states had little transition during the turning process, and the fidelity of vehicle dynamics could be compromised and ignored. According to the vehicle system modeling, the kinematic model was the most appropriate model to be adopted and applied. In a vehicle kinematic model, the steering tires are assumed to be traveling in the direction they are facing. In addition, the responses of a vehicle to a steering input are determined only by the geometric parameters. The four-wheel farm tractor model can be simplified and converted to a two-wheel “bicycle” model (Figure 5.4).

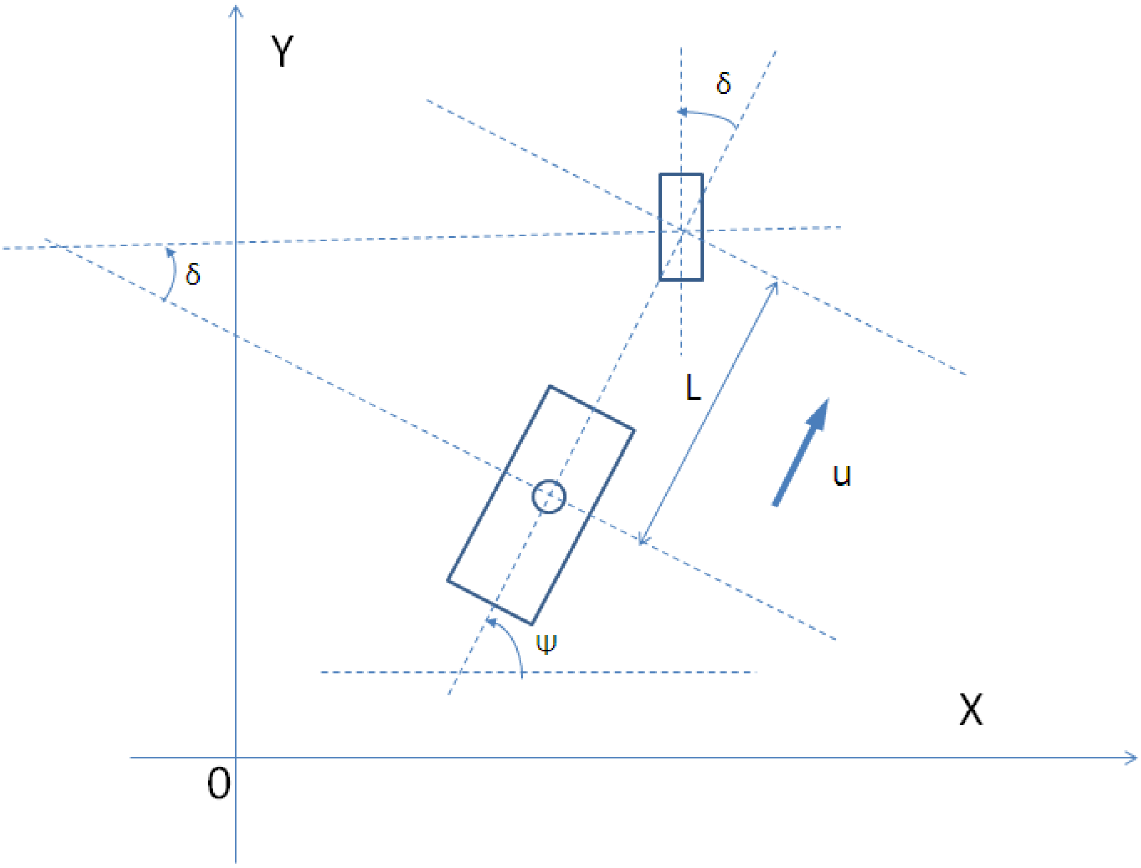


Figure 5.4: *Top view of a tractor modeled as a bicycle, with geometric variables labeled*

Given that there were no vehicle dynamic factors in this modeling,

$$\dot{x} = u\cos\psi \qquad (5.5)$$

and

$$\dot{y} = u\sin\psi \quad , \qquad (5.6)$$

where u was the vehicle longitudinal velocity and Ψ was the vehicle heading angle (the angle between the world coordinate x-axis and the vehicle longitudinal axis, as depicted in Figure 5.4).

Because front steering was the only steering mode, the yaw rate (derivative of heading angle) of the vehicle had a triangular function relationship with the steering angle:

$$\dot{\psi} = \mathrm{r} = \frac{\mathrm{utan}\delta}{\mathrm{L}} \;. \tag{5.7}$$

The steering angle δ had to stay within the range of hardware limit:

$$-\delta_{max} \leq \ \delta \leq \delta_{max} \quad . \tag{5.8}$$

In this study, it was defined that turning left is positive steering and turning right is negative steering. Besides steering, acceleration of the vehicle’s velocity was considered another system control input:

$$\dot{\mathrm{u}} = \mathrm{a} \quad . \tag{5.9}$$

The velocity and acceleration had boundaries and could be described as

$$u_{min} \leq \ u \leq u_{max} \tag{5.10}$$

and

$$a_{min} \leq \ a \leq a_{max} \quad\quad . \tag{5.11}$$

Thus, there were two control signals—steering angle, δ, and longitudinal acceleration, a. Thus, it was a system that had two degrees of freedom. The state variables, $[\mathrm{x} \quad \mathrm{y} \quad \psi]^T$, were considered to be system output. The system vector, X, was defined as

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x \\ y \\ \Psi \\ u \end{bmatrix} \quad\quad . \tag{5.12}$$

Then, the system state function was derived as

$$\dot{X} = f(X) = \begin{bmatrix} x_4 \cos x_3 \\ x_4 \sin x_3 \\ x_4 \tan u_1 / L \\ u_2 \end{bmatrix} \quad , \tag{5.13}$$

assuming that velocity, u , was a constant. Because this optimization problem was a minimum time problem, the cost function should be

$$J_{min}(X) = \int_{t_o}^{t_f} 1 \; dt = \int_{x_o}^{x_f} \frac{1}{u\cos\psi} \, dx \quad . \tag{5.14}$$

Thus, the problem frame has been established.

5.4.2 Tractor-implement Kinematic Modeling

Similar to the tractor model, tractor-implement models have been proposed in the literature for either off-road (Bell, 1999; Feng et al., 2005) or on-road (El-Gindy, 1989; Torishu et al. 1992; Deng and Kang, 2003) operations. In this study, a single-axle towed implement was considered the only implement model in further research. A top view of a generic tractor-implement combination is presented in Figure 5.5.

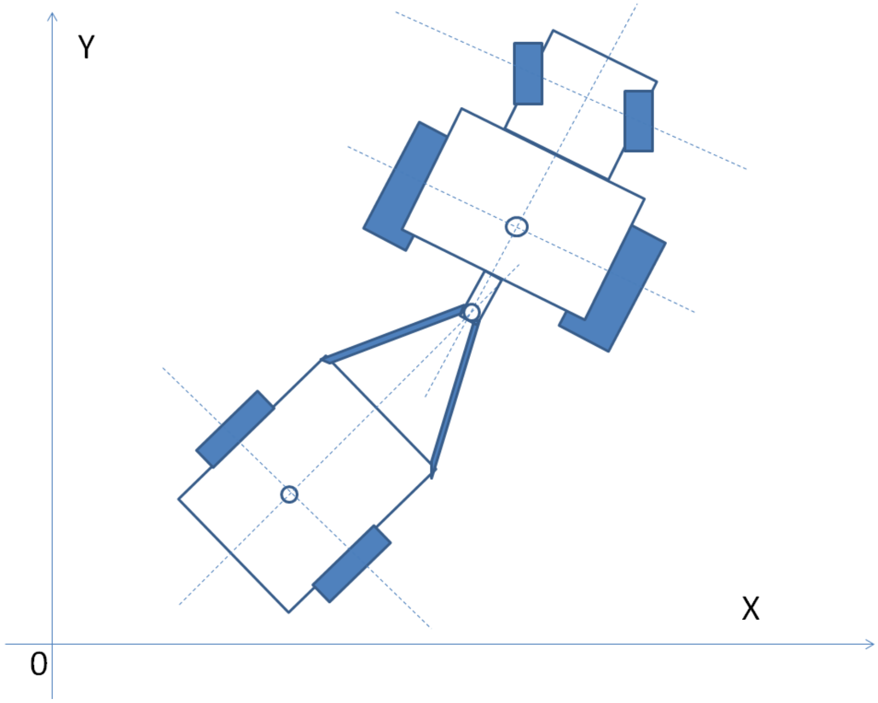


Figure 5.5: *A tractor-implement combination in a two-dimensional coordinate system*

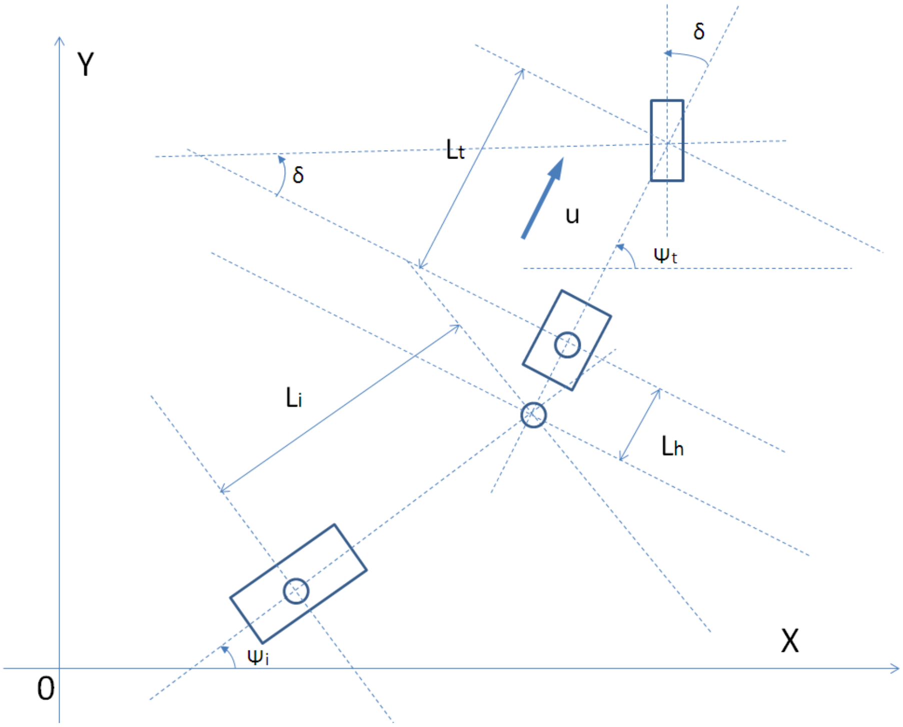


Figure 5.6: *Top view of a tractor-implement model with geometric variables labeled in a two-dimensional coordinate system*

A top view of a tractor-implement model with geometric variables labeled in a 2D coordinate system is presented in Figure 5.7. For the convenience of further discussion, subscript notations were used to specify whether the variable was related to the tractor or to the implement, specifying the coordinate axis to which the variable corresponds.

Table 5.2*: Subscript notations for the vehicle dynamics variables.*

Subscript	Description
t	Tractor
i	Implement

The velocity of the tractor affects the tractor's GC coordinates:

$$\dot{x}_t = u_t \cos\psi_t \quad , \tag{5.15}$$

$$\dot{y}_t = u_t \sin\psi_t \quad , \tag{5.16}$$

where u is the vehicle longitudinal velocity and ψ_t is the tractor heading angle. Similar to the tractor-only model, the yaw rate is

$$\dot{\psi}_t = r_t = \frac{u_t \tan\delta_t}{L_t}, \tag{5.17}$$

where steering angle δ_t is one of the system control inputs. Tractor acceleration is considered to be the other control input of the system:

$$\dot{u}_t = a_t \qquad . \tag{5.18}$$

Because the single-axle implement is a passive object in this combination, the velocity and heading angle of the implement are passive variables, depending on tractor velocity u_t, heading ψ_t, and having a one-order relationship with tractor steering angle δ_t:

$$\delta_i = \psi_t - \psi_i \quad , \tag{5.19}$$

$$u_i = u_t \cos(\psi_t - \psi_i) \ . \tag{5.20}$$

Thus, the model of the implement could be described as

$$\dot{x}_i = u_i\cos\psi_i = u_t\cos\,(\psi_t - \psi_i)\cos\psi_i, \tag{5.21}$$

$$\dot{y}_\imath = u_i\sin\psi_i = u_t\cos\,(\psi_t - \psi_i)\sin\psi_i, \tag{5.22}$$

$$\dot{\psi}_\imath = r_i = \frac{u_i\tan\delta_i}{L_i} = \left. u_t\cos\,(\psi_t - \psi_i)\tan\,(\psi_t - \psi_i) \middle/ _{L_i} \right. = \left. u_t\sin\,(\psi_t - \psi_i) \middle/ _{L_i}. \right. \tag{5.23}$$

Therefore ,

$$\dot{u}_\imath = a_t\cos(\psi_t - \psi_i) - u_t\left(\frac{u_t\tan\delta_t}{L_t} - \frac{u_t\sin(\psi_t - \psi_i)}{L_i}\right)\sin(\psi_t - \psi_i) = a_t\cos(\psi_t - \psi_i) - \frac{{u_t}^2\tan\delta_t\sin(\psi_t - \psi_i)}{L_t} + \frac{{u_t}^2\sin^2(\psi_t - \psi_i)}{L_i} \quad . \tag{5.24}$$

In summary, there are two control signals—steering angle $u_1 = \delta_t$ and longitudinal acceleration $u_2 = a_t$. The state variables $[x_t \;\; y_t \;\; \psi_t \;\; u_t \;\; x_i \;\; y_i \;\; \psi_i \;\; u_i]^T$ are considered to be system output. The system vector, X, is defined as

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \psi_t \\ u_t \\ x_i \\ y_i \\ \psi_i \\ u_i \end{bmatrix} \quad . \tag{5.25}$$

Then, the system state equations are derived as

$$\dot{X} = f(X) = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \\ \dot{x_6} \\ \dot{x_7} \\ \dot{x_8} \end{bmatrix} = \begin{bmatrix} \dot{x_t} \\ \dot{y_t} \\ \dot{\psi_t} \\ \dot{u_t} \\ \dot{x_\imath} \\ \dot{y_\imath} \\ \dot{\psi_\imath} \\ \dot{u_\imath} \end{bmatrix} \tag{5.26}$$

and

$$\dot{X} = \begin{bmatrix} u_t \cos\psi_t \\ u_t \sin\psi_t \\ u_t \tan\delta_t / L_t \\ a_t \\ u_t \cos(\psi_t - \psi_i)\cos\psi_i \\ u_t \cos(\psi_t - \psi_i)\sin\psi_i \\ u_t \sin(\psi_t - \psi_i) / L_i \\ a_t \cos(\psi_t - \psi_i) - u_t^2 \tan\delta_t \sin(\psi_t - \psi_i) / L_t + u_t^2 \sin^2(\psi_t - \psi_i) / L_i \end{bmatrix} =$$

$$\begin{bmatrix} x_4 \cos x_3 \\ x_4 \sin x_3 \\ x_4 \tan u_1 / L_t \\ u_2 \\ x_4 \cos(x_3 - x_7)\cos x_7 \\ x_4 \cos(x_3 - x_7)\sin x_7 \\ x_4 \sin(x_3 - x_7) / L_i \\ u_2 \cos(x_3 - x_7) - x_4^2 \tan u_1 \sin(x_3 - x_7) / L_t + x_4^2 \sin^2(x_3 - x_7) / L_i \end{bmatrix}. \qquad (5.27)$$

The model indicates that the tractor-trailer combination was a two-order and eight-dimensional system. The description of $\dot{\psi}_i$ and $\dot{u}_i$ implied that the motion of the implement was affected by the heading angles and velocities of both the tractor and the implement itself. Although it was still a two-DOF system, the complexity of the system was increased.

5.4.3 Tractor-implement-implement Kinematic Modeling

In agricultural farm operation, two implements might be needed to do individual tasks in series, such as pre-seeding processing and fertilizing. The tractor-implement-

implement scenario (Figure 5.7) was considered in this headland-turning optimization research.

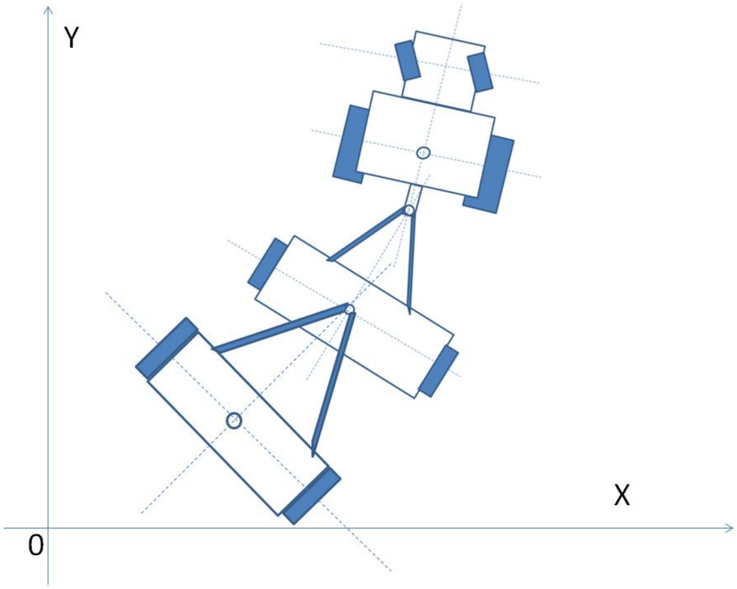


Figure 5.7: *A tractor-implement-implement combination in a two-dimensional coordinate system*

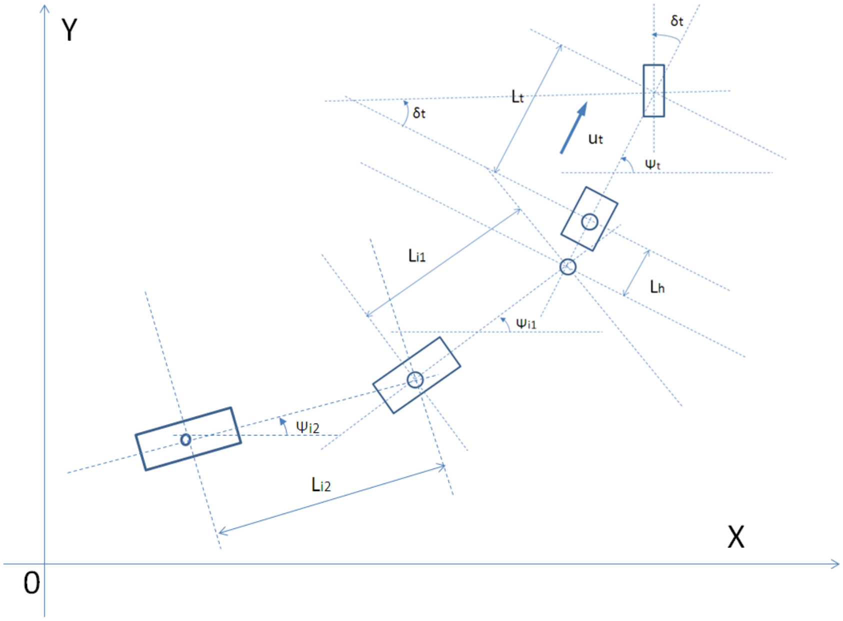


Figure 5.8: *Top view of a tractor-implement-implement model, with geometric variables labeled, in a two-dimensional coordinate system*

Table 5.3*: Subscript notations for the vehicle dynamics variables.*

Subscript	Description
t	Tractor
i1	Implement 1
i2	Implement 2

In summary, there were still two control signals for the vehicle system—steering angle $\mathrm{u_1 = \delta_t}$ and longitudinal acceleration $\mathrm{u_2 = a_t}$. However, the system dimension was increased from 8 to 12, due to the addition of a second implement. The state variables $[x_t \;\; y_t \;\; \psi_t \;\; u_t \;\; x_{i1} \;\; y_{i1} \;\; \psi_{i1} \; u_{i1} \; x_{i2} \;\; y_{i2} \;\; \psi_{i2} \;\; u_{i2}]^T$ were considered to be system state variables.

From single implement modeling, it was known that:

$$\mathrm{\delta_{i1} = \psi_t - \psi_{i1}} \qquad , \tag{5.28}$$

$$\mathrm{u_{i1} = u_t \cos(\psi_t - \psi_{i1})} \qquad , \tag{5.29}$$

$$\mathrm{\dot{x}_{i1} = u_{i1} cos\psi_{i1} = u_t cos\,(\psi_t - \psi_{i1}) cos\psi_{i1}} \;\; , \tag{5.30}$$

$$\mathrm{\dot{y}_{i1} = u_{i1} sin\psi_{i1} = u_t cos\,(\psi_t - \psi_{i1}) sin\psi_{i1}} \;\; , \tag{5.31}$$

$$\mathrm{\dot{\psi}_{i1} = r_{i1} = \frac{u_{i1} tan\delta_{i1}}{L_{i1}} = \frac{u_t cos\,(\psi_t - \psi_{i1}) tan\,(\psi_t - \psi_{i1})}{L_{i1}} = \frac{u_t sin\,(\psi_t - \psi_{i1})}{L_{1i}}}, \tag{5.32}$$

and

$$\mathrm{\dot{u}_{i1} = a_t \cos(\psi_t - \psi_{i1}) - u_t \left(\frac{u_t tan\delta_t}{L_t} - \frac{u_t \sin(\psi_t - \psi_{i1})}{L_{i1}} \right) \sin(\psi_t - \psi_{i1}) = \; a_t \cos(\psi_t - \psi_{i1}) - \frac{{u_t}^2 tan\delta_t sin(\psi_t - \psi_{i1})}{L_t} + \frac{{u_t}^2 \sin^2(\psi_t - \psi_{i1})}{L_{i1}}} \quad . \tag{5.33}$$

Based on the known equations, it was possible to solve the derivatives of the second implement as follows:

$$\mathrm{\delta_{i2} = \psi_{i1} - \psi_{i2}} \tag{5.34}$$

$$\mathrm{u}_{\mathrm{i}2} = \mathrm{u}_{\mathrm{i}1}\cos(\psi_{\mathrm{i}1} - \psi_{\mathrm{i}2}) \tag{5.35}$$

$$\dot{\mathrm{x}}_{\mathrm{i}2} = \mathrm{u}_{\mathrm{i}2}\cos\psi_{\mathrm{i}2} = \mathrm{u}_{\mathrm{i}1}\cos(\psi_{\mathrm{i}1} - \psi_{\mathrm{i}2})\cos\psi_{\mathrm{i}2} \tag{5.36}$$

$$\dot{\mathrm{y}}_{\mathrm{i}2} = \mathrm{u}_{\mathrm{i}2}\sin\psi_{\mathrm{i}2} = \mathrm{u}_{\mathrm{i}1}\cos(\psi_{\mathrm{i}1} - \psi_{\mathrm{i}2})\sin\psi_{\mathrm{i}2} \tag{5.37}$$

$$\dot{\psi}_{\mathrm{i}2} = \mathrm{r}_{\mathrm{i}2} = \frac{\mathrm{u}_{\mathrm{i}2}\tan\delta_{\mathrm{i}2}}{\mathrm{L}_{\mathrm{i}2}} = \frac{\mathrm{u}_{\mathrm{i}1}\cos(\psi_{\mathrm{i}1}-\psi_{\mathrm{i}2})\tan(\psi_{\mathrm{i}1}-\psi_{\mathrm{i}2})}{\mathrm{L}_{\mathrm{i}2}} = \frac{\mathrm{u}_{\mathrm{i}1}\sin(\psi_{\mathrm{i}1}-\psi_{\mathrm{i}2})}{\mathrm{L}_{12}} \tag{5.38}$$

$$\dot{\mathrm{u}}_{\mathrm{i}2} = \dot{\mathrm{u}}_{\mathrm{i}1}\cos(\psi_{\mathrm{i}1} - \psi_{\mathrm{i}2}) - \mathrm{u}_{\mathrm{i}1}\left(\dot{\psi}_{\mathrm{i}1} - \dot{\psi}_{\mathrm{i}2}\right)\sin(\psi_{\mathrm{i}1} - \psi_{\mathrm{i}2}) \tag{5.39}$$

$$= (\mathrm{a}_{\mathrm{t}}\cos(\psi_{\mathrm{t}} - \psi_{\mathrm{i}1}) - \frac{\mathrm{u}_{\mathrm{t}}{}^{2}\tan\delta_{\mathrm{t}}\sin(\psi_{\mathrm{t}}-\psi_{\mathrm{i}1})}{\mathrm{L}_{\mathrm{t}}} + \frac{\mathrm{u}_{\mathrm{t}}{}^{2}\sin^{2}(\psi_{\mathrm{t}}-\psi_{\mathrm{i}1})}{\mathrm{L}_{\mathrm{i}1}})\cos(\psi_{\mathrm{i}1} - \psi_{\mathrm{i}2}) - \\ \mathrm{u}_{\mathrm{i}1}(\frac{\mathrm{u}_{\mathrm{t}}\sin(\psi_{\mathrm{t}}-\psi_{\mathrm{i}1})}{\mathrm{L}_{1\mathrm{i}}} - \frac{\mathrm{u}_{\mathrm{i}1}\sin(\psi_{\mathrm{i}1}-\psi_{\mathrm{i}2})}{\mathrm{L}_{12}})\sin(\psi_{\mathrm{i}1} - \psi_{\mathrm{i}2}) \tag{5.40}$$

System vector X was defined as

$$X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \psi_t \\ u_t \\ x_{i1} \\ y_{i1} \\ \psi_{i1} \\ u_{i1} \\ x_{i2} \\ y_{i2} \\ \psi_{i2} \\ u_{i2} \end{bmatrix} \tag{5.41}$$

Then, the system state function was derived:

$$\dot{\mathrm{X}} = \mathrm{f(X)} = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \\ \dot{x_6} \\ \dot{x_7} \\ \dot{x_8} \\ \dot{x_9} \\ \dot{x_{10}} \\ \dot{x_{11}} \\ \dot{x_{12}} \end{bmatrix} = \begin{bmatrix} \dot{x_t} \\ \dot{y_t} \\ \dot{\psi_t} \\ \dot{u_t} \\ \dot{x_{i1}} \\ \dot{y_{i1}} \\ \dot{\psi_{i1}} \\ \dot{u_{i1}} \\ \dot{x_{i2}} \\ \dot{y_{i2}} \\ \dot{\psi_{i2}} \\ \dot{u_{i2}} \end{bmatrix} \tag{5.42}$$

$$= \begin{bmatrix} u_t \cos\psi_t \\ u_t \sin\psi_t \\ u_t \tan\delta_t \big/ L_t \\ a_t \\ u_t \cos(\psi_t - \psi_{i1}) \cos\psi_{i1} \\ u_t \cos(\psi_t - \psi_{i1}) \sin\psi_{i1} \\ u_t \cos(\psi_t - \psi_{i1}) \tan(\psi_t - \psi_{i1}) \big/ L_{i1} \\ a_t \cos(\psi_t - \psi_{i1}) - \dfrac{{u_t}^2 \tan\delta_t \sin(\psi_t - \psi_{i1})}{L_t} + \dfrac{{u_t}^2 \sin^2(\psi_t - \psi_{i1})}{L_{i1}} \\ a_t \cos(\psi_t - \psi_i) - {u_t}^2 \tan\delta_t \sin(\psi_t - \psi_i) \big/ L_t + {u_t}^2 \sin^2(\psi_t - \psi_i) \big/ L_i \\ u_t \cos(\psi_t - \psi_{i1}) \cos(\psi_{i1} - \psi_{i2}) \cos\psi_{i2} \\ u_t \cos(\psi_t - \psi_{i1}) \cos(\psi_{i1} - \psi_{i2}) \sin\psi_{i2} \\ u_t \cos(\psi_t - \psi_{i1}) \cos(\psi_{i1} - \psi_{i2}) \tan(\psi_{i1} - \psi_{i2}) \big/ L_{i2} \\ \left(a_t \cos(\psi_t - \psi_{i1}) - \dfrac{{u_t}^2 \tan\delta_t \sin(\psi_t - \psi_{i1})}{L_t} + \dfrac{{u_t}^2 \sin^2(\psi_t - \psi_{i1})}{L_{i1}}\right) \cos(\psi_{i1} - \psi_{i2}) \\ -u_{i1}(\dfrac{u_t \sin(\psi_t - \psi_{i1})}{L_{1i}} - \dfrac{u_{i1} \sin(\psi_{i1} - \psi_{i2})}{L_{12}}) \sin(\psi_{i1} - \psi_{i2}) \end{bmatrix}$$

$$= \begin{bmatrix} x_4 \cos x_3 \\ x_4 \sin x_3 \\ x_4 \tan u_1 \big/ L_t \\ u_2 \\ x_4 \cos(x_3 - x_7) \cos x_7 \\ x_4 \cos(x_3 - x_7) \sin x_7 \\ x_4 \cos(x_3 - x_7) \tan(x_3 - x_7) \big/ L_{i1} \\ u_2 \cos(x_3 - x_7) - {x_4}^2 \tan u_1 \sin(x_3 - x_7) \big/ L_t + {x_4}^2 \sin^2(x_3 - x_7) \big/ L_{i1} \\ x_4 \cos(x_3 - x_7) \cos(x_7 - x_{10}) \cos x_{10} \\ x_4 \cos(x_3 - x_7) \cos(x_7 - x_{10}) \sin x_{10} \\ x_4 \cos(x_3 - x_7) \cos(x_7 - x_{10}) \tan(x_7 - x_{10}) \big/ L_{i2} \\ \left(u_2 \cos(x_3 - x_7) - \frac{{u_2}^2 \tan\delta_t \sin(x_3 - x_7)}{L_t} + \frac{{u_2}^2 \sin^2(x_3 - x_7)}{L_{i1}}\right) \cos(x_7 - x_{10}) \\ - x_8(\frac{x_4 \sin(x_3 - x_7)}{L_{1i}} - \frac{x_8 \sin(x_7 - x_{10})}{L_{12}}) \sin(x_7 - x_{10}) \end{bmatrix} \tag{5.43}$$

According to the equations, the combination was a 12-dimensional system.

5.5 Results and Discussion

The philosophy behind direct methods was to discrete the problem in time and approximate the states and controls in these time intervals. The optimal control problem (OCP) then was converted to a nonlinear programming (NLP) problem, which could be solved effectively with numerical tools. In this chapter, the tractor and implement models mentioned previously were used in various headland-turning scenarios, and the simulation results are presented. The TOMLAB/SNOPT toolbox was adopted to generate these regional optimal solutions. TOMLAB/SNOPT is a direct collocation nonlinear programming (DCNLP) toolbox, which can be executed with MATLAB, Visual Studio, SQL, and other compiling environments. The specialty of this toolbox is that it is able to cope with high-dimensional nonlinear systems and provide local optimization solutions. The discussion is based on a simple example of a symmetric light bulb turn and is extended to more complicated scenarios, such as a fishtail turn (Figure 5.9). Tractor-implement combinations were considered and investigated in the optimization as well.

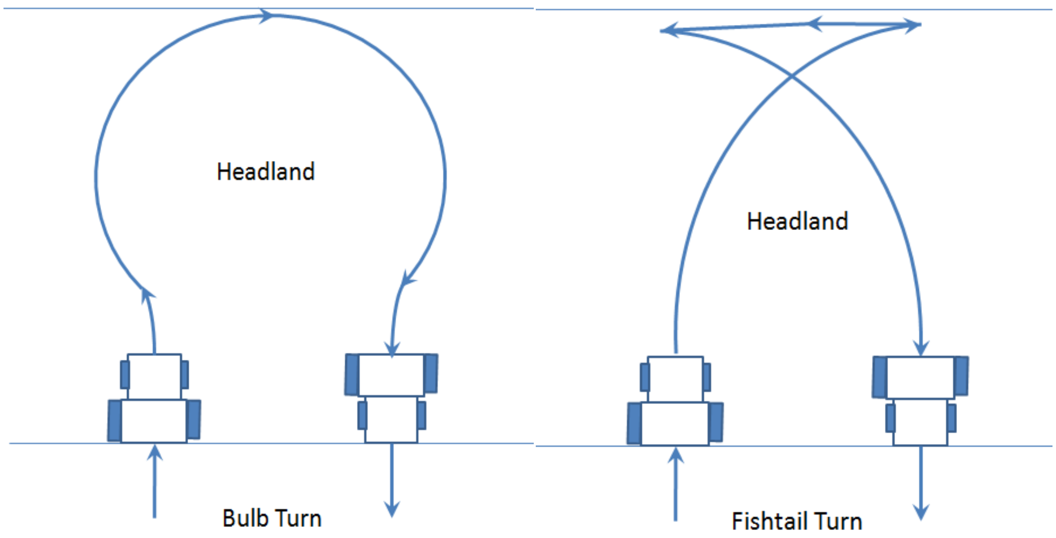


Figure 5.9: *Two different headland-turning modes—bulb turn and fishtail turn*

5.5.1 An Example: Symmetric Bulb Turn

Agricultural vehicles are commonly built in large dimensions to carry big and heavy motors and implements, to accommodate tough off-road environments. However, to improve land efficiency, the width between the working rows are sometimes narrow, which makes it difficult for vehicles to make a "U" turn to get to the next path. The "bulb" turn is a usual turning mode used when the minimum turning diameter of the tractor is greater than the path width. The turning process can be divided into three phases. At the beginning of the research, a tractor-only model was chosen for further study. The fixed vehicle parameters and the restricted area constraints are shown in Table 4.4.

Table 5.4*: Fixed vehicle parameters and restricted area constraints*

Vehicle parameters		
Description (Unit)	Symbol in the model	Value/Range
Wheelbase (m)	L	1.85
Initial heading of the tractor (rad)	H_o	$\pi/2$
Final heading of the tractor (rad)	H_f	$-\pi/2$
Steering angle (rad)	δ	$[-\pi/6 \quad \pi/6]$
Headland width (m)	D	6.5
Headland boundary angle (rad)	β	0
Row width (m)	d	5.0
Constant vehicle velocity (m/s)	u	2.0

As shown in Table 5.4, to simplify the complexity and prove the effectiveness of the toolbox, the vehicle was working under a constant velocity during the complete turning process, from the initial position to the final position. The headland boundary was vertical to the reference path, which implied that the headland angle was 0. Thus, the optimal time cost problem was equivalent to a shortest trajectory problem. The simulation results are presented in Figures 5.10 and 5.11.

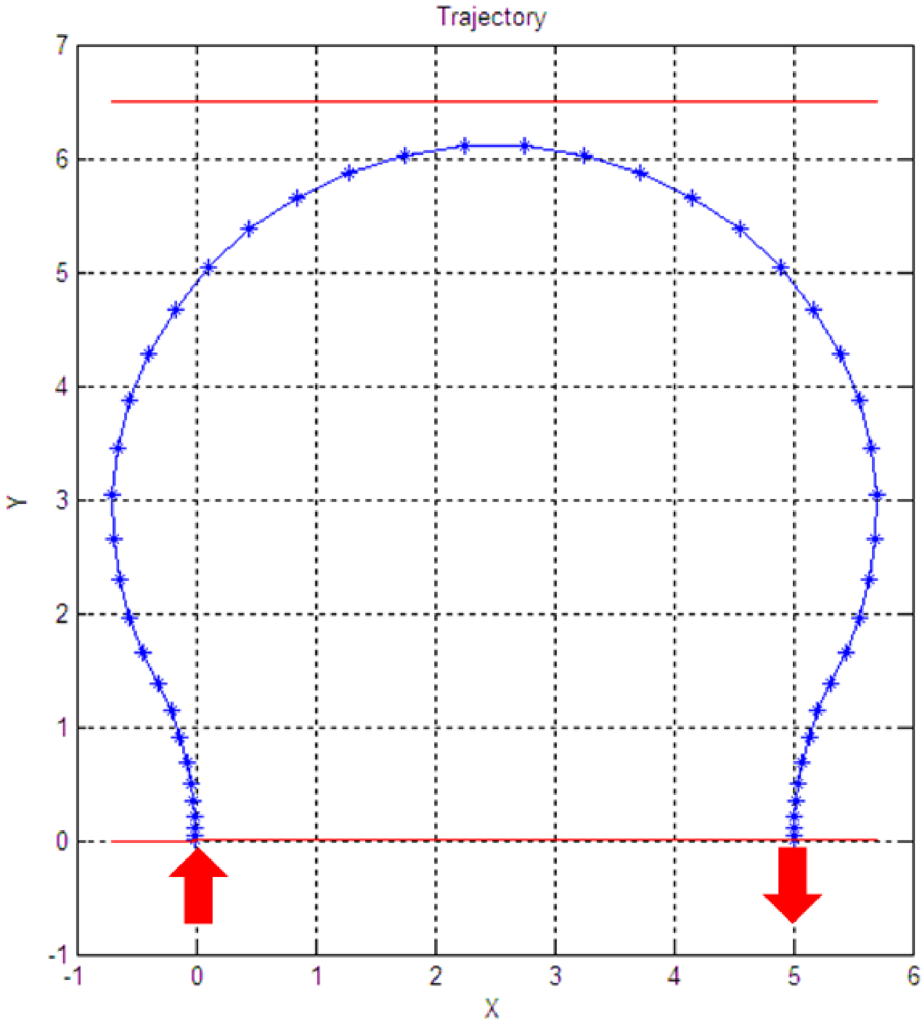


Figure 5.10: *Trajectory of a tractor bulb turn in minimum-time headland-turning optimization with constant velocity*

In this simplified scenario, there was no implement attached, so the trajectory of the tractor is the only focus. Because the velocity was considered constant, the steering angle was the only control signal of the system. The system states and control during the turning process are displayed in Figure 5.11.

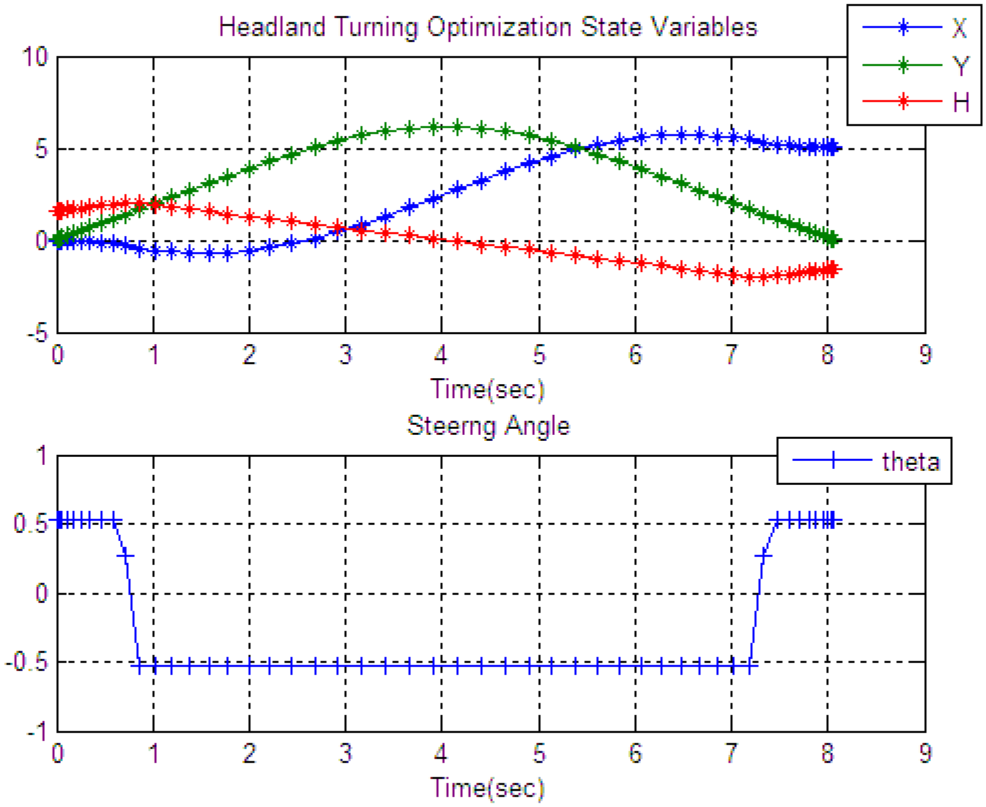


***Figure 5.11:** States and control of a tractor bulb turn in minimum-time headland-turning optimization with constant velocity*

The optimal time cost obtained from the simulation was 8.048 seconds. The three "phases" can be seen clearly by the steering angle control plot. The tractor was steering left at the beginning, then right, following a "bulb" shape, turning and finally steering left, finishing the trajectory. Because the problem could be considered to be a shortest trajectory problem with constant velocity, it could be computed by geometric analytical methods.

The minimum turning radius was obtained from the wheelbase and maximum steering angle of the tractor:

$$R = \frac{L}{\tan|\delta|} = 3.2. \qquad (5.44)$$

The minimum turning diameter could be derived as $2R = 6.4 > 5$, which means that a U-turn could not be accomplished, and the tractor had to make a bulb turn. Based on the known and derived information, the geometric solution was applied (Figure 5.12).

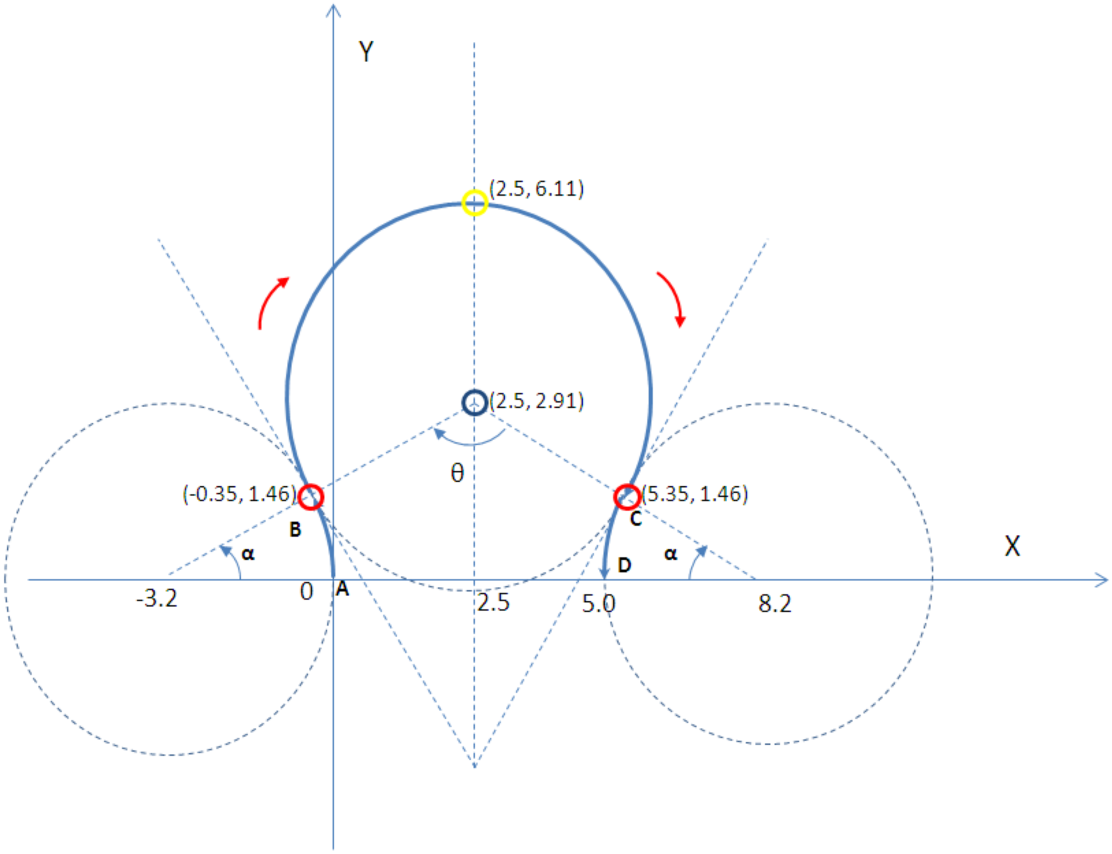


Figure 5.12: *A geometric solution of shortest trajectory of a tractor symmetric bulb turn*

As presented in Figure 5.12, the optimal trajectory was composed of three curves: $\widehat{AB}$, $\widehat{BC}$, and $\widehat{CD}$. They were from three circles with radii of 3.2 m, which were tangent to one another. The proof of why this trajectory was the optimal solution for the minimum time problem is omitted. Because the initial heading and final heading are $\frac{\pi}{2}$ and $-\frac{\pi}{2}$, two of the circles were tangent to the lines $x = 0$ and $x = 5$, and the other one was tangent to both of them; the center was on the line of $x = 2.5$. By geometric calculation, the centers of $\widehat{AB}$, $\widehat{BC}$, and $\widehat{CD}$ were computed as $(-3.2, 0)$, $(2.5, 2.91)$, and $(8.2, 0)$, respectively.

Due to the tangential relationship of the three circles, the tangent points were located at $(-0.35, 1.46)$ and $(5.35, 1.46)$, respectively (as denoted by red circles in Figure 5.12). The curvature of $\widehat{AB}$ and $\widehat{CD}$ was derived by

$$\alpha = \widehat{AB} = \widehat{CD} = sin^{-1}\left({}^{1.46}/_{3.2}\right) = 0.472\, rad. \tag{5.45}$$

The arc length of $\widehat{AB}$ and $\widehat{CD}$ was

$$L_{\widehat{AB}} = L_{\widehat{CD}} = R * \alpha = 1.5104\, m. \tag{5.46}$$

Obviously, $\theta = \pi - 2\alpha = 2.1976$, so the curvature of $\widehat{BC}$ was

$$2\pi - \theta = \pi + 2\alpha = 4.0856\, rad \tag{5.47}$$

and the arc length of $\widehat{BC}$ was $L_{\widehat{BC}} = R * (2\pi - \theta) = 13.0839\, m.$ (5.48)

Hence, the length of the trajectory could be achieved:

$$L_{\widehat{AD}} = L_{\widehat{AB}} + L_{\widehat{BC}} + L_{\widehat{CD}} = 16.1047\, m. \tag{5.49}$$

Because the tractor velocity was constantly 2 m/s, the analytical optimal turning time for this problem was

$$t_{min} = {}^{L_{\widehat{AD}}}/_{u} = {}^{(L_{\widehat{AB}} + L_{\widehat{BC}} + L_{\widehat{CD}})}/_{u} = 8.052\, sec. \tag{5.50}$$

The coordinates of the top point (denoted by the yellow circle in Figure 5.12) were derived as $(2.5, 6.11)$. This implied that under the setting scenario, one of the essential conditions was that the headland width must meet $D > 6.11\ m$.

***Table 5.5**: Comparison of Minimum Time Optimization results of DCNLP and analytical methods*

DCNLP result (sec)	Analytical result (sec)	Error (sec)	Error percentage
8.048	8.052	0.004	0.05 %

Based on Table 5.5, the DCNLP result is extremely close to what was achieved by the traditional analytical method. The error was 0.05%, which was relatively low within this short-range trajectory optimization. The performance of TOMLAB/SNOPT was satisfactory. The simulation was completed within 0.4843 seconds (according to the indicated CPU computation time), which indicated the potential of using this toolbox to solve NLP problems.

5.5.2 Asymmetric Bulb Turn

As the effectiveness and accuracy of DCNLP have been proven, a more realistic scenario is proposed. In this scenario, the headland boundary has an angle β with the field, and the velocity of the tractor is controllable. The vehicle parameters and headland information are listed in Table 5.6.

***Table 5.6**: Fixed vehicle parameters*

Vehicle parameters		
Description (Unit)	Symbol in the model	Value/Range
Wheelbase (m)	L	1.8
Initial heading of the tractor (rad)	ψ_o	$\pi/2$
Final heading of the tractor (rad)	ψ_f	$-\pi/2$
Initial tractor velocity (m/s)	u_o	0
Final tractor velocity (m/s)	u_f	0
Steering angle (rad)	δ	$[-\pi/6 \;\; \pi/6]$
Vehicle velocity (m/s)	u	$[0 \;\; 3]$
Vehicle acceleration (m/s^2)	a	$[-4 \;\; 2]$

***Table 5.7**: Area boundary constraints*

Headland width (m)	D	6.0
Headland boundary angle (rad)	β	$\pi/6$
Row width (m)	d	5.0 m

The headland angle (the boundary of the headland versus the field) was assumed to be 30 °, and the headland width became narrower. Furthermore, the initial and final vehicle velocities were both 0, but velocity could be a variable from 0 to 3 m/s on the trajectory. The maximum acceleration was $2\ m/s^2$, and the maximum brake deceleration was $-4\ m/s^2$, with the consideration of the safety concern during the turning process. The steering angle limit remained the same. By running this model in MATLAB with TOMLAB/SNOPT, the trajectory shown in Figure 5.13 was attained.

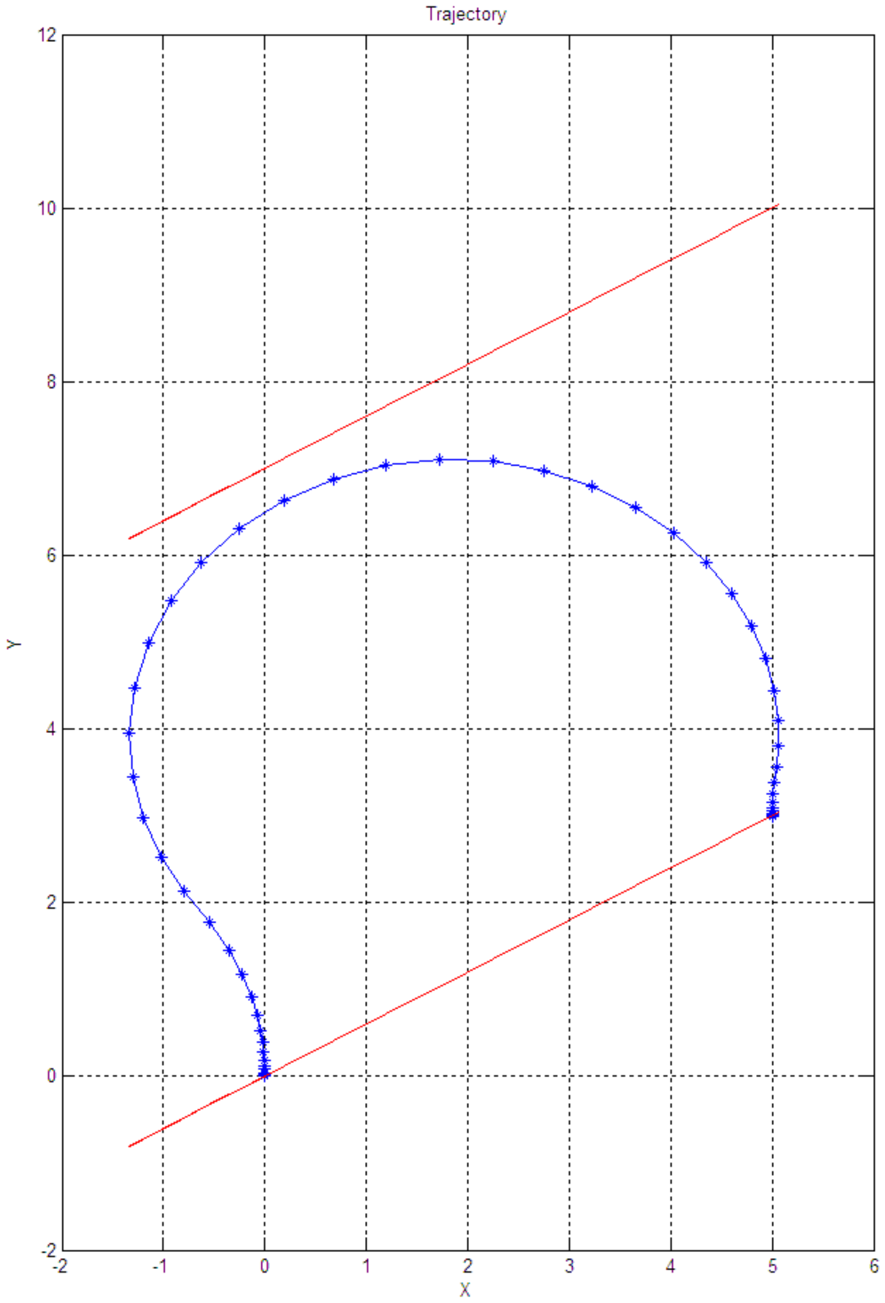


Figure 5.13: *Trajectory of a tractor bulb turn in minimum-time headland-turning optimization with variable velocity and revised headland angle*

The upper boundary of the headland was $-xcos\left(\frac{\pi}{6}\right)+y=6.93$, and the lower boundary was $-xcos\left(\frac{\pi}{6}\right)+y=0$. Thus, the headland boundary condition could be described as $0\leq -xcos\left(\frac{\pi}{6}\right)+y\leq 6.93$. (5.51)

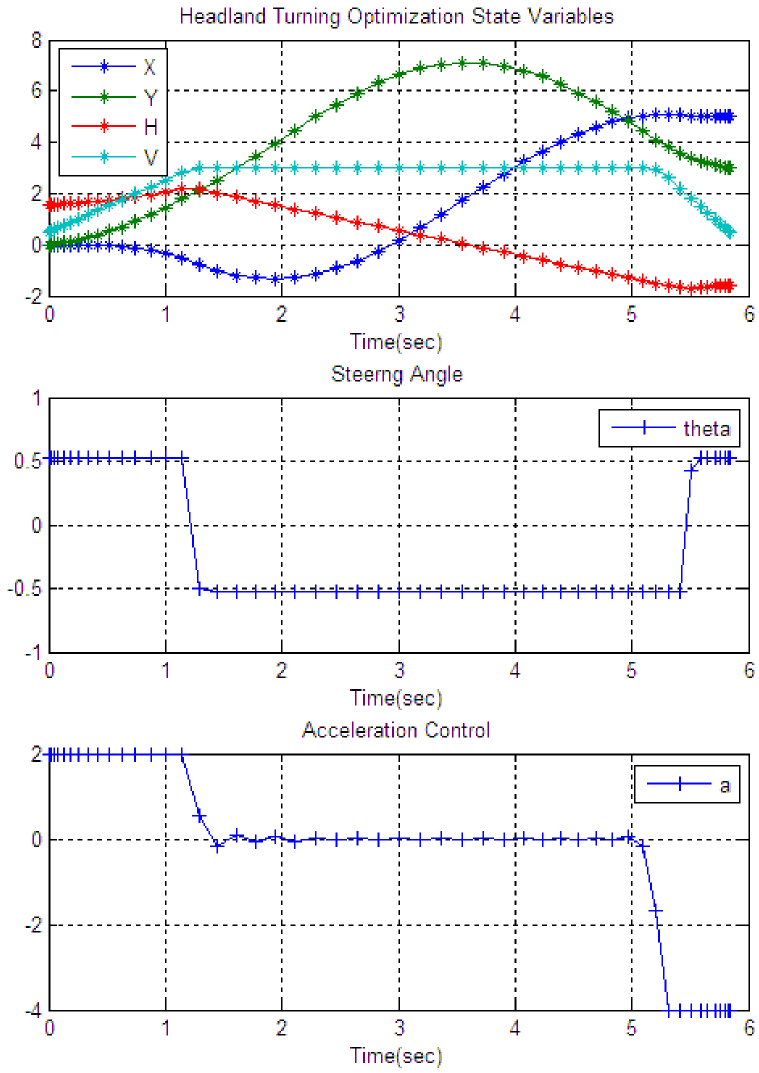


Figure 5.14: *States and control of a tractor bulb turn in minimum time headland turning optimization with variable velocity and revised headland angle*

The optimal time cost given from the simulation was 5.8359 seconds. From the control signal plots, the turning process could be divided into three phases as well. The tractor was accelerating hard from the initial position and kept turning left in the first phase (Figure 5.14). Then it was steering right and kept running with constant velocity. Finally the tractor slowed down and adjusted the tractor heading straight to the path and was ready to enter. It was easy to locate the extreme point (blue circle illustrated in

Figure 5.15) on the trajectory to make a parallel straight line to the lower boundary. From which it could be calculated that the minimum headland width required with the tractor and headland angle parameters was 5.57 m.

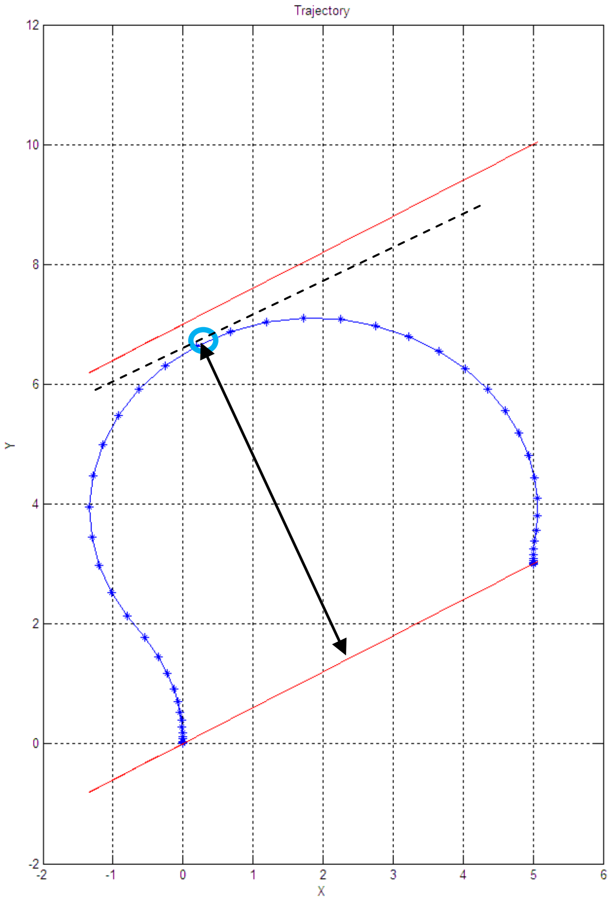


Figure 5.15: *Minimum headland width calculations with the headland angle at* $\frac{\pi}{6}$.

In this trial, the minimum turning radius remained the same as in the Symmetric Bulb Turn trial because of the same front steering angle limit, as well as the in-field path width. Though the vehicle velocity became a variable, it did not affect the turning radius or the trajectory. But the required headland width was changed, due to the impact of the changed headland angle. This would imply the necessity of headland turning optimization in global path planning research.

***Table 5.8**: Comparison of Minimum Time Optimization results from DCNLP and analytical methods*

Parameter	Symmetric Bulb Turn	Revised Bulb Turn	Difference?
In-field path width (m)	5.0	5.0	No
Tractor turning radius (m)	3.2	3.2	No
Headland angle (rad)	0	$\pi/6$	Yes
Required headland width (m)	**6.11**	**5.57**	**Yes**

The impact of the headland angle to the headland width was presented in Table 5.7. Further, it could be concluded that the angle between the path and the headland impacted the required width on the headland. If the headland width was restricted in global path planning scenario and bulb turn was the only turning mode, changing approaching angle to the reference path is a potential way to solve this issue.

5.5.3 Fishtail Turn

According to the previous discussion, it would be interesting to see what could be done if the headland width did not meet the requirement of bulb turn. In other words, if the headland was too narrow for any bulb turn, compared to the steering limit of the tractor, for the tractor to accomplish a bulb turn, some other turning mode should be adopted. And “fishtail” turn was one of the options.

Similar to bulb turn, the procedure of fishtail turn also could be divided into three phases. There is one main difference in the second phase that the tractor is allowed to reverse. It provides the possibility that the tractor was able to adjust the orientation within a relatively limited space, which improved the flexibility and mobility of the tractor in the field operation.

***Table 5.9**: The fixed vehicle parameters and restricted area constraints*

Vehicle parameters		
Description (Unit)	Symbol in the model	Value/Range
Wheelbase (m)	L	1.85 m
Initial heading of the tractor (rad)	ψ_o	$\pi/2$
Final heading of the tractor (rad)	ψ_f	$-\pi/2$
Initial tractor velocity (m/s)	u_o	0.5
Final tractor velocity (m/s)	u_f	0.5
Headland width (m)	**D**	**4.33 m**
Headland boundary angle (rad)	β	$\pi/6$
Row width (m)	d	5.0 m
Steering angle (rad)	**δ**	$[-\boldsymbol{\pi}/\mathbf{10}\quad \boldsymbol{\pi}/\mathbf{10}]$
Vehicle velocity (m/s)	u	$[-3\quad 3]$
Vehicle acceleration (m/s^2)	a	$[-4\quad 2]$

As illustrated in the table above, the difference on the parameter was the tractor maximum steering angle is limited at

$$\delta_{max} = \pi/10 \tag{5.52}$$

$$-\pi/10 = -\delta_{max} \leq \delta \leq \delta_{max} = \pi/10 \tag{5.53}$$

Thus the turning radius could be derived.

$$R = L/sin\delta_{max} = 1.6/\sin(\pi/10) = 5.18\, m \tag{5.54}$$

The width of the headland was set to be as low as 4.33 m, which was quite small compared to the increased turning radius. And it is impossible for the tractor to make a bulb turn in this restricted headland. Thus fishtail turn was required under this circumstance. The simulation results of the optimal solution were achieved by running DCNLP methods.

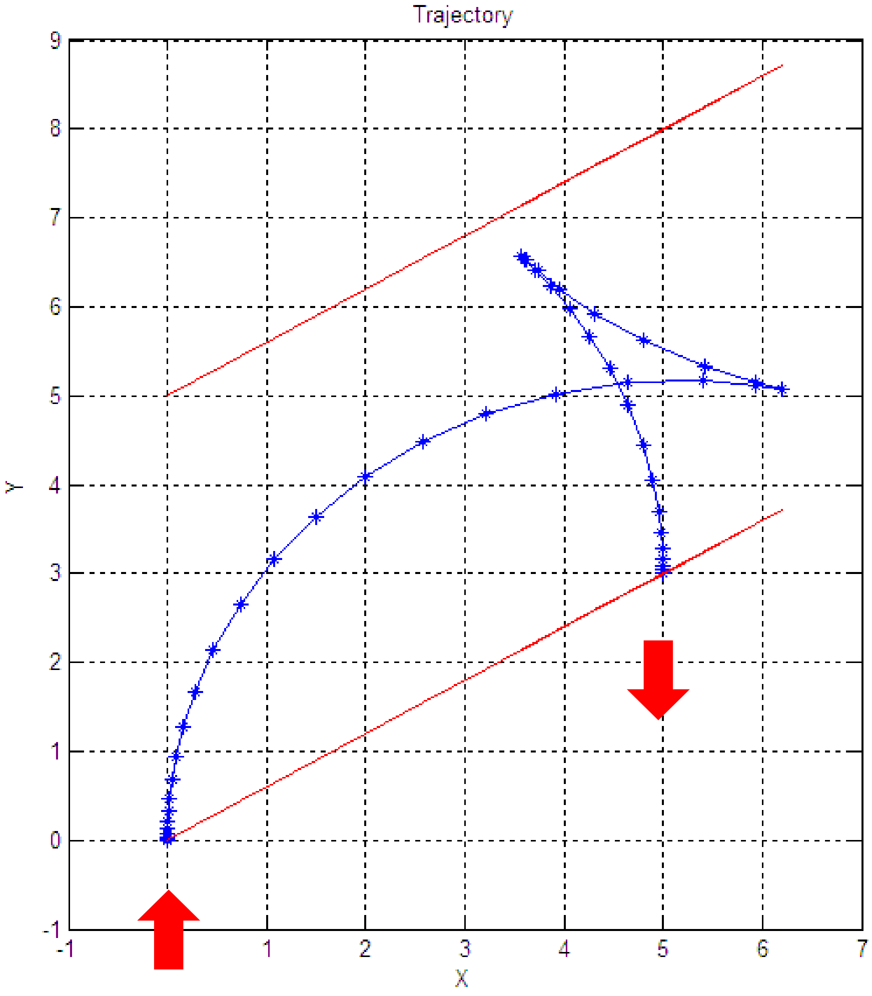


Figure 5.16: *Trajectory of a tractor fishtail turn of minimum time headland turning optimization with variable velocity and revised headland angle*

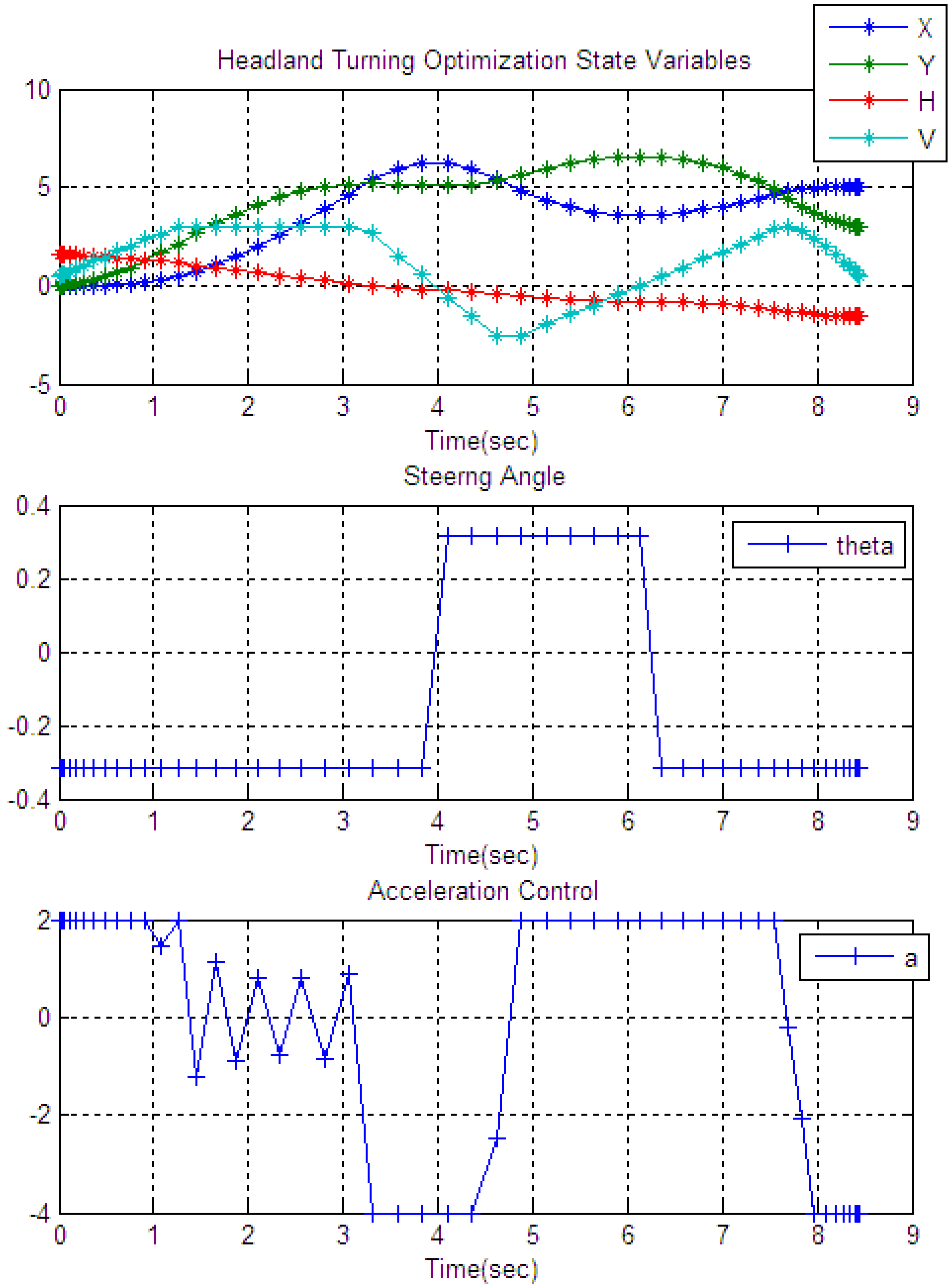


Figure 5.17: *States and control of a tractor fishtail turn in minimum-time headland-turning optimization with variable velocity and revised headland angle*

The optimal time cost of the fishtail turn was 8.4524 seconds. From the trajectory result shown in Figure 5.16, it was apparent that the fishtail turn could be accomplished even if the headland width was narrower, indicating an absolute advantage over the bulb turn. From a time efficiency aspect, and given the existence of the reverse phase, the headland-turning process was relatively slow, compared to the U-turn and bulb turn.

5.5.4 Tractor-implement headland turning optimization

Tractor-implement is a common scenario in agricultural field operation. The system model, discussed in Section 5.4.2, was a tractor with a single-axle implement. It will be studied and discussed further in this section.

Table 5.10*: Fixed vehicle and implement parameters and area constraints*

Vehicle parameters		
Description (Unit)	Symbol in the model	Value/Range
Wheelbase (m)	L_t	1.85
Implement length (m)	L_i	2.5
Initial heading of the tractor (rad)	ψ_{to}	${}^{\pi}/_{2}$
Final heading of the tractor (rad)	ψ_{tf}	$-{}^{\pi}/_{2}$
Initial heading of implement (rad)	ψ_{io}	${}^{\pi}/_{2}$
Final heading of implement (rad)	ψ_{if}	$-{}^{\pi}/_{2}$
Initial tractor velocity (m/s)	u_o	0.5
Final tractor velocity (m/s)	u_f	0.5
Headland width (m)	D	9.0
Headland boundary angle (rad)	β	0
Row width (m)	d	5.0
Steering angle (rad)	δ	$[-\,{}^{\pi}/_{6} \quad {}^{\pi}/_{6}]$
Vehicle velocity (m/s)	u	$[-3 \quad 3]$
Vehicle acceleration (m/s^2)	a	$[-4 \quad 2]$

By loading the headland constraints and tractor-implement information into MATLAB and running SNOPT, the trajectory shown in Figure 5.18 was attained.

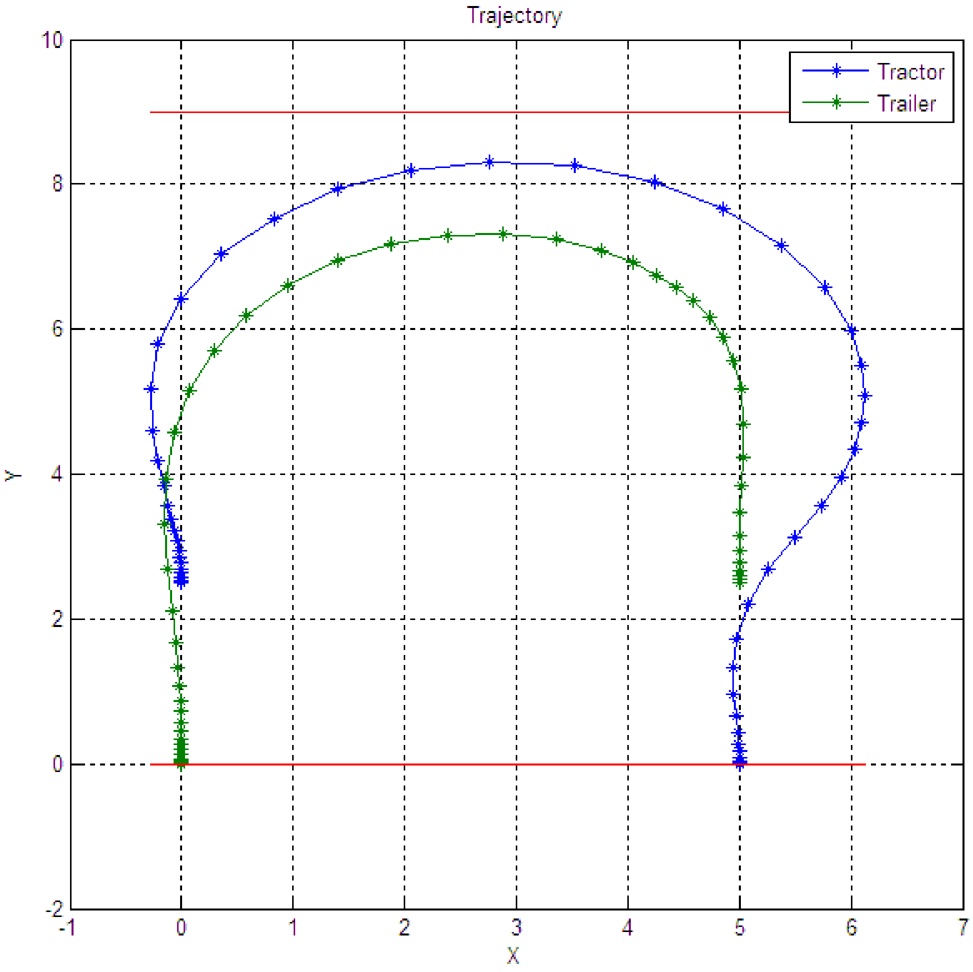


Figure 5.18: *Trajectory of a tractor-trailer combination in minimum-time headland-turning optimization with variable velocity*

In the trajectory, it can be seen that the tractor was performing a bulb turn to accommodate the turning of the implement.

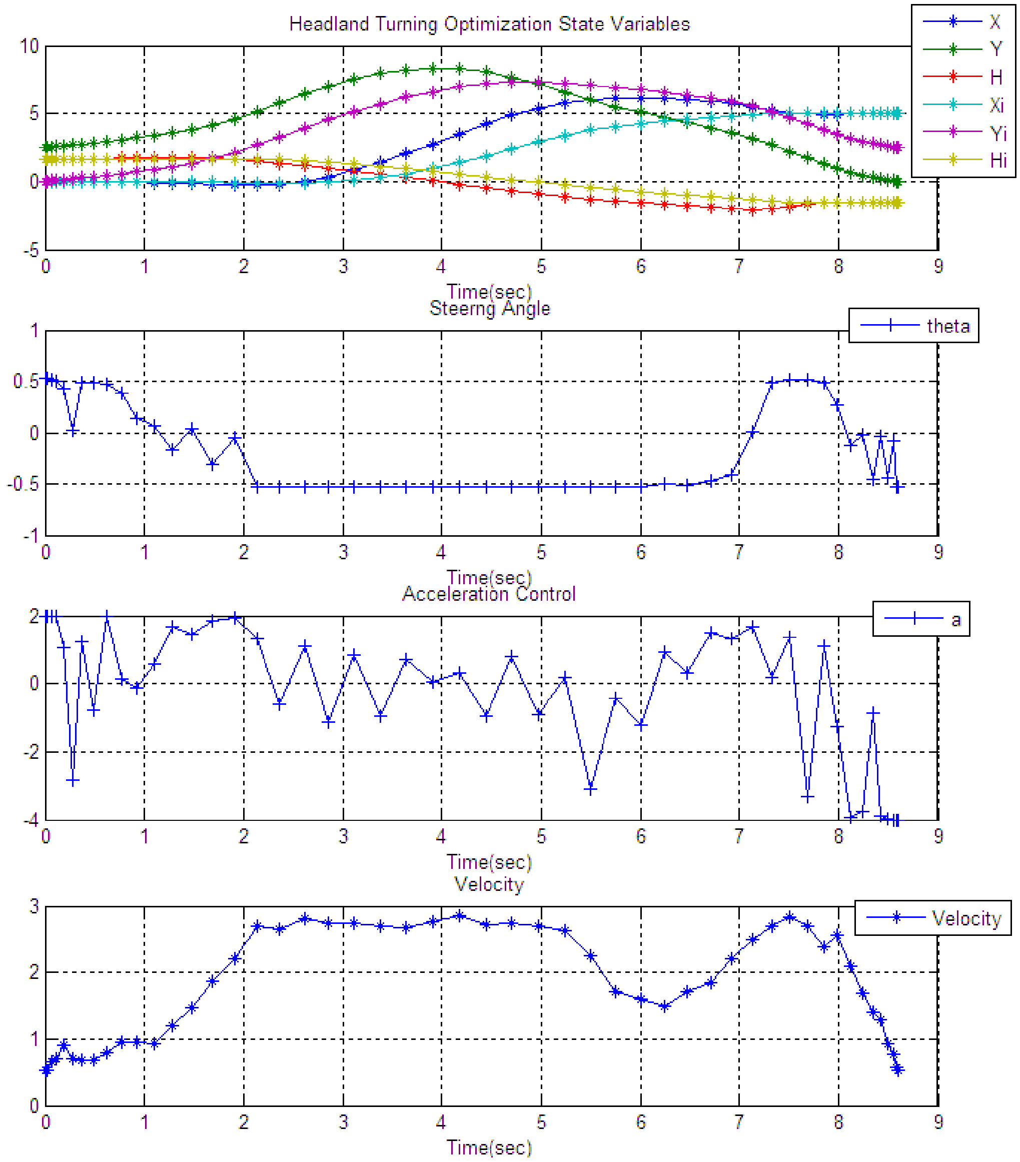


Figure 5.19: *States and controls of a tractor-trailer in minimum-time headland-turning optimization with variable velocity and steering limit*

The optimal time cost given by the simulation was 8.6093 seconds. From the variable plots in Figure 5.19, the results suggest that because the row width was not spacious enough to make a U-turn, the tractor steered first to the left to make a bulb turn and then accelerated toward the right. Once the implement approached the line of $x = 5$, the tractor slowed down, adjusted the body orientation, and was ready to enter the reference path.

There is a possible situation that with a larger implement, the tractor would not be able to turn hard to shorten the trajectory. In other words, the tractor steering angle limit is extremely small. It would be interesting to compute the minimum headland width to provide best-time efficiency under this scenario. The following simulation provides an example with most of the vehicle parameters remaining the same and the steering limit changing:

$$-\frac{\pi}{10} \leq \delta \leq \frac{\pi}{10}. \tag{5.55}$$

Table 5.11*: Fixed vehicle and implement parameters and restricted area constraints*

Vehicle parameters		
Description (Unit)	Symbol in the model	Value/Range
Wheelbase (m)	L_t	1.85
Implement length (m)	L_i	2.5
Initial heading of the tractor (rad)	ψ_{to}	${}^{\pi}/_{2}$
Final heading of the tractor (rad)	ψ_{tf}	$-{}^{\pi}/_{2}$
Initial heading of implement (rad)	ψ_{io}	${}^{\pi}/_{2}$
Final heading of implement (rad)	ψ_{if}	$-{}^{\pi}/_{2}$
Initial tractor velocity (m/s)	u_o	0.5
Final tractor velocity (m/s)	u_f	0.5
Headland width (m)	D	9.0
Headland boundary angle (rad)	β	0
Row width (m)	d	5.0
Steering angle (rad)	δ	$[-{}^{\pi}/_{10}\quad {}^{\pi}/_{10}]$
Vehicle velocity (m/s)	u	[0 3]
Vehicle acceleration (m/s^2)	a	$[-4\quad 2]$

The trajectory shown in Figure 5.20 was attained.

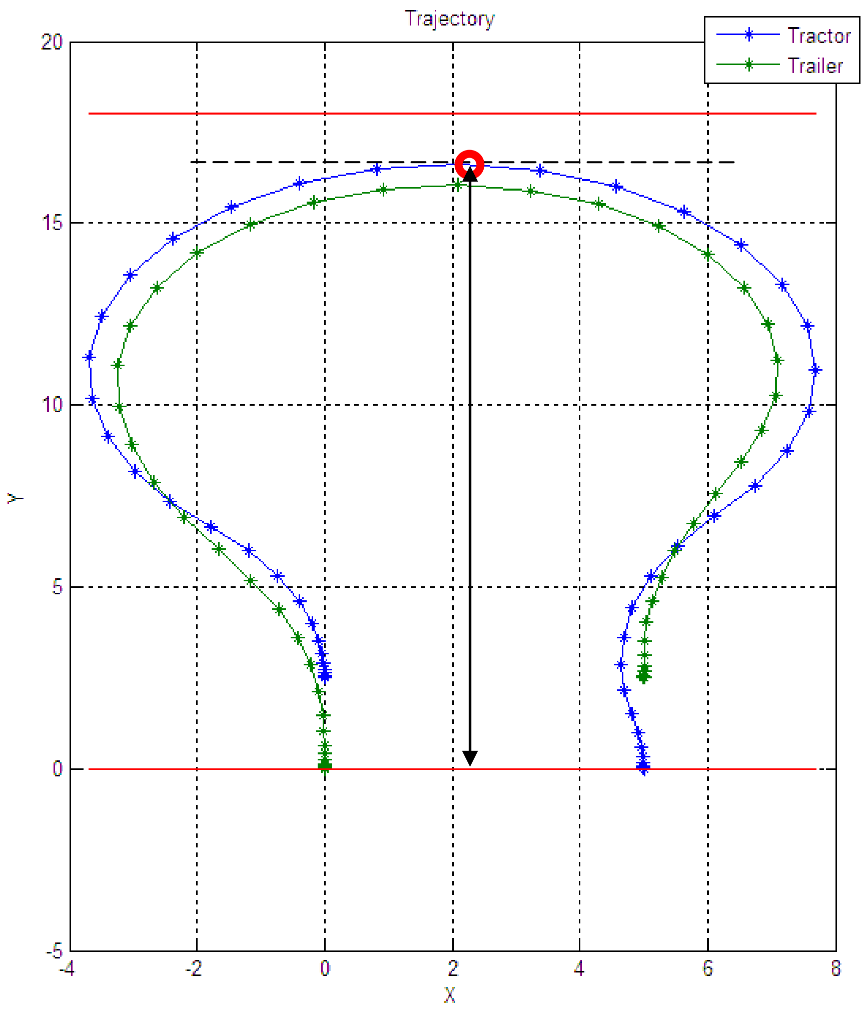


Figure 5.20: *Trajectory of a tractor-trailer combination in minimum-time headland-turning optimization with small steering limit*

From the generated minimum time trajectory, with the steering limit $\frac{\pi}{10}$ (18 °), the headland width needed to be at least 16.63 m to offer generous space to obtain optimal time efficiency. The given minimum turning time was 13.8 seconds.

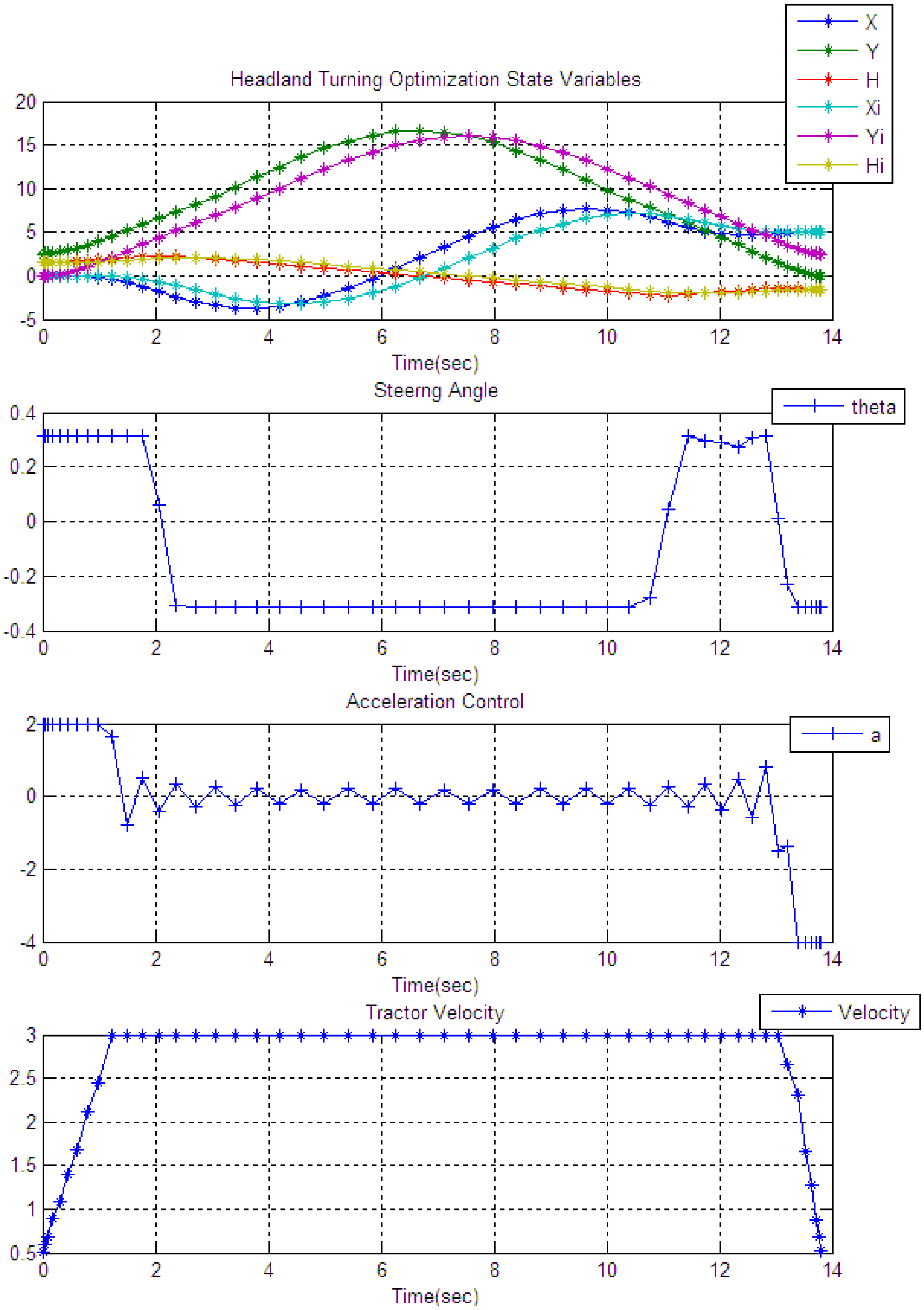


Figure 5.21: *States and controls of a tractor-trailer in minimum time headland turning optimization with small steering limit*

To test the effectiveness of SNOPT for solving realistic problems, another example was proposed. It was realistic to assume that the headland was located on a hill, and the altitude at the right side was higher than the left side. Furthermore, if the tractor was hauling a large, heavy trailer, the tractor would not be able to make a sharp turn in the uphill direction compared to the downhill direction. In other words, when the tractor entered the headland, the steering limit to the left would be greater than that to the right $|\delta_{max}^{left}| > |\delta_{max}^{right}|$.

***Table 5.12**: Fixed vehicle and implement parameters and area constraints*

Vehicle parameters		
Description (Unit)	Symbol in the model	Value/Range
Wheelbase (m)	L_t	1.85
Implement length (m)	L_i	2.5
Initial heading of the tractor (rad)	ψ_{to}	${}^{\pi}/_{2}$
Final heading of the tractor (rad)	ψ_{tf}	$-{}^{\pi}/_{2}$
Initial heading of implement (rad)	ψ_{io}	${}^{\pi}/_{2}$
Final heading of implement (rad)	ψ_{if}	$-{}^{\pi}/_{2}$
Initial tractor velocity (m/s)	u_o	0.5
Final tractor velocity (m/s)	u_f	0.5
Headland width (m)	D	15.58
Headland boundary angle (rad)	β	${}^{\pi}/_{6}$
Row width (m)	d	5.0
Steering angle (rad)	δ	$[-{}^{\pi}/_{10} \quad {}^{\pi}/_{8}]$
Vehicle velocity (m/s)	u	$[0 \quad 3]$
Vehicle acceleration (m/s^2)	a	$[-4 \quad 2]$

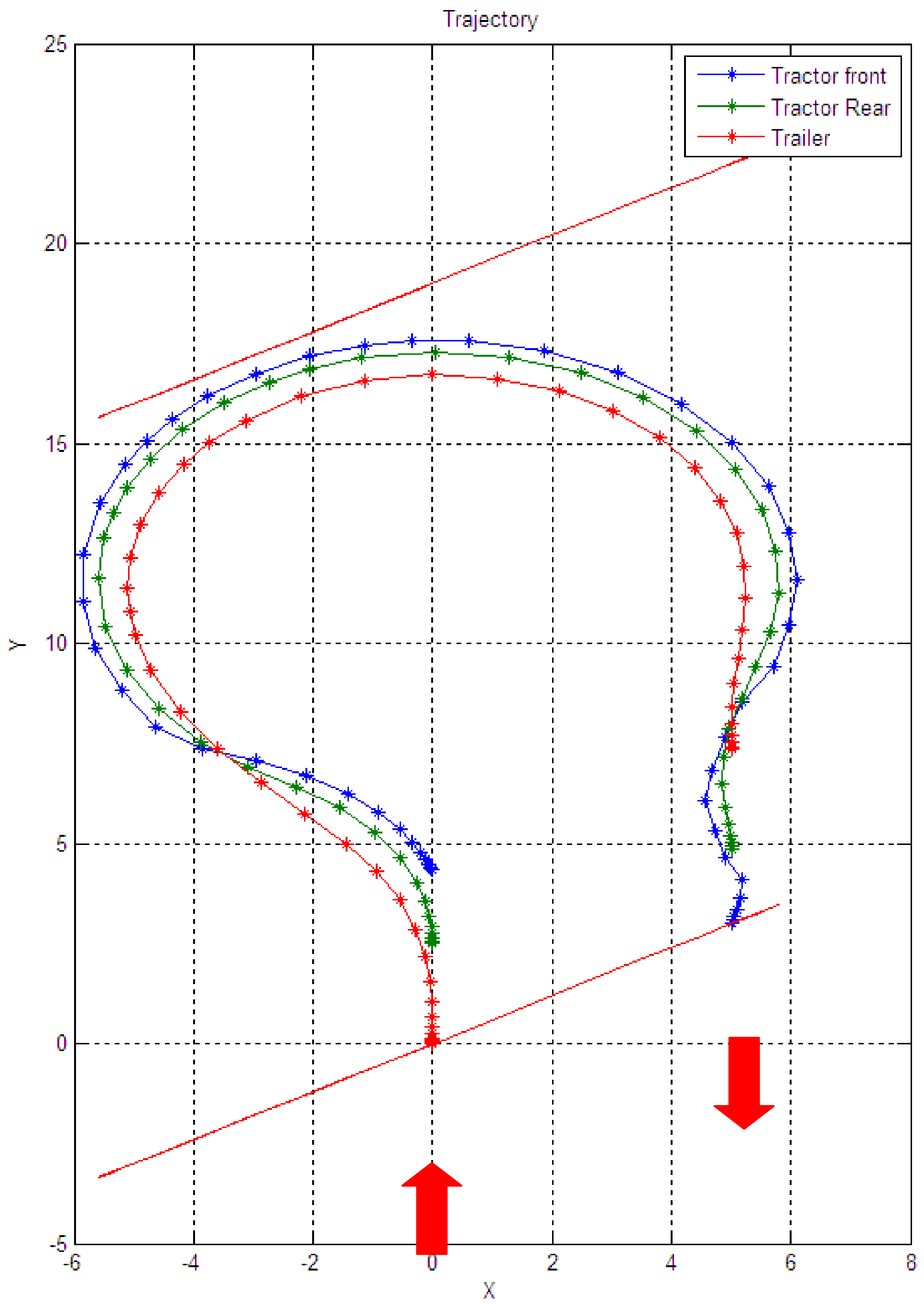


***Figure 5.22:** Trajectory of a tractor-trailer combination under different steering limits*

An asymmetric bulb turn was generated (Figure 5.22). It shows that there is a possibility that the DCNLP method was able to solve headland-turning optimization problems with complicated terrain constraints. For a comprehensive overview of individual wheel movements, there is a detailed wheel trajectory provided. In this trial, the width of the tractor was 2.45 m, and the width for the trailer was 5.0 m.

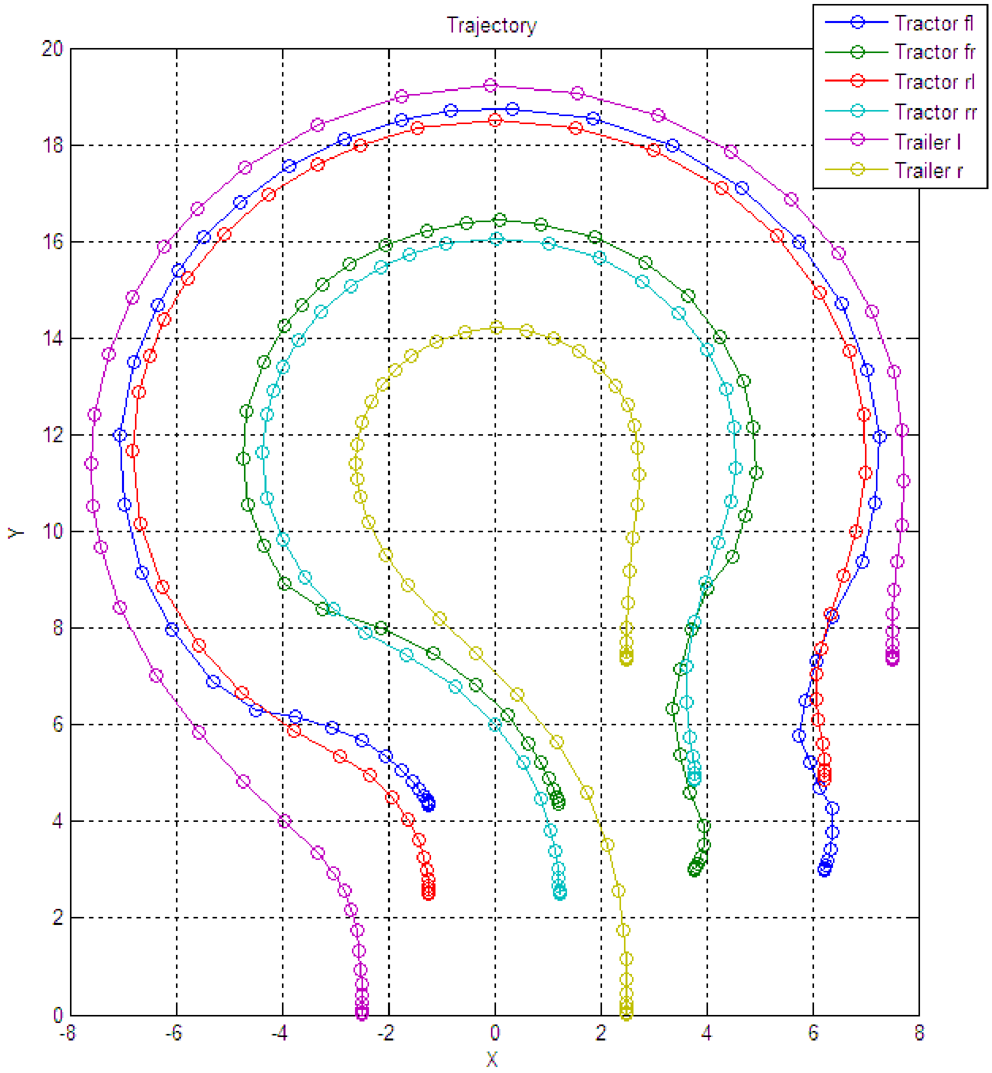


Figure 5.23: *Trajectory of all wheels of a tractor-trailer combination on a hill headland. In the tractor wheel plots, "fl" means front left wheel, "fr" means front right wheel, "rl" means rear left wheel, and "rr" means rear right wheel. In the trailer wheel plots, "l" means left wheel and "r" means right wheel.*

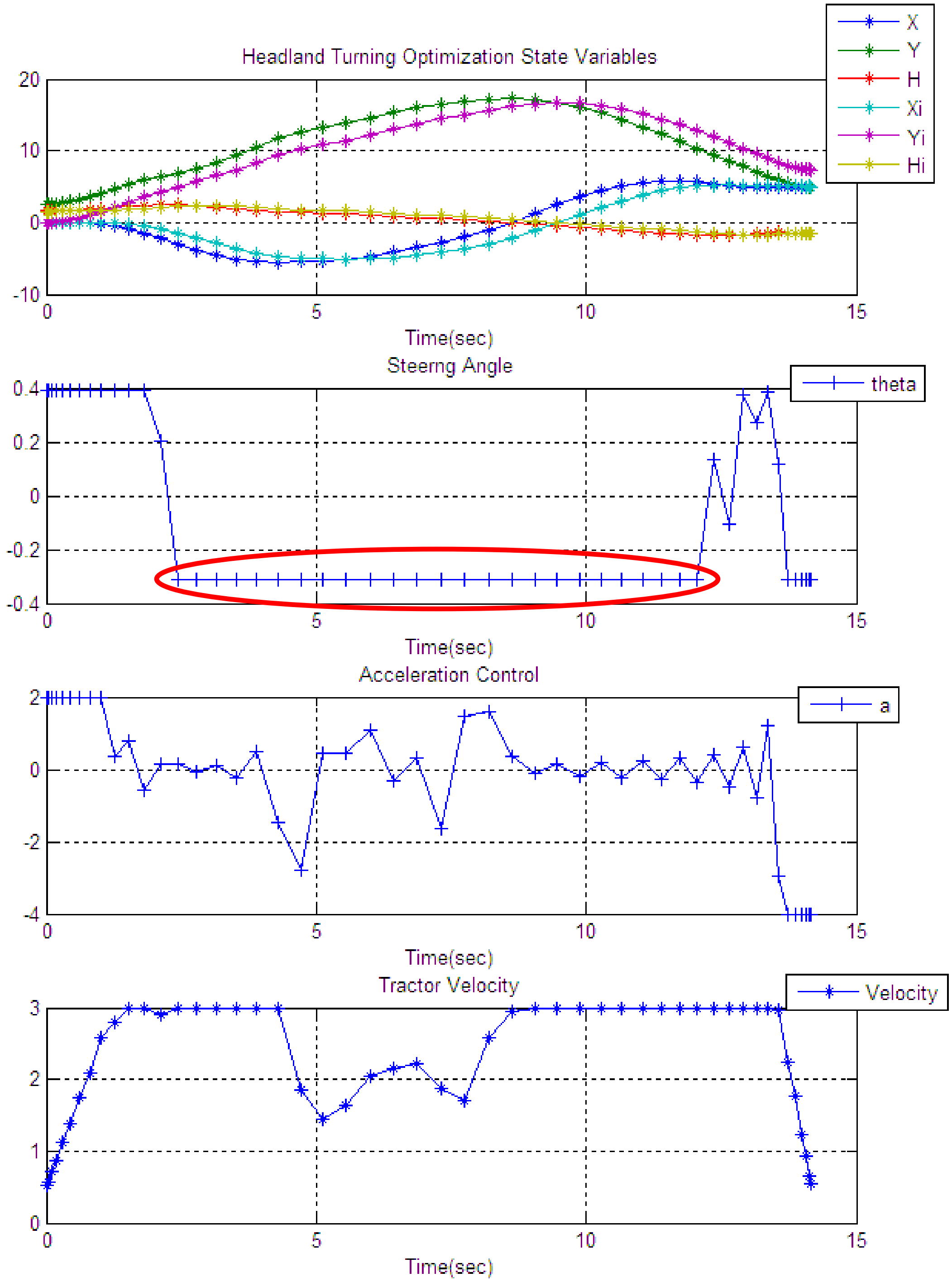


Figure 5.24: *States and controls of a tractor-trailer in headland turning on a hill*

The steering difference to the left and right can be seen in the plots in Figure 5.24. As indicated by the red circles, the right steering (negative steering angle) could not be large; therefore, the steering control required more time to finish the bulb turn. The optimal headland turning time for this case was 14.2 seconds.

5.5.5 Tractor with two implements headland-turning optimization

After discussing single-trailer scenarios, it is reasonable to consider a case where the tractor has two single-axle implements. An assumed working scenario is proposed below. However, due to the complexity of the two-implement model, which was discussed in Section 5.4.3, this simulation result needs further investigation.

***Table 5.13**: Fixed vehicle and implement parameters and restricted area constraints*

Vehicle parameters		
Description (Unit)	Symbol in the model	Value/Range
Wheelbase (m)	L_t	1.85
Length of Implement 1 (m)	L_{i1}	2.5
Length of Implement 2 (m)	L_{i2}	2.5
Initial heading of the tractor (rad)	ψ_{to}	$\pi/2$
Final heading of the tractor (rad)	ψ_{tf}	$-\pi/2$
Initial heading of Implement 1 (rad)	ψ_{i1o}	$\pi/2$
Final heading of Implement 1 (rad)	ψ_{i1f}	$-\pi/2$
Initial heading of Implement 2 (rad)	ψ_{i2o}	$\pi/2$
Final heading of implement 2 (rad)	ψ_{i2f}	$-\pi/2$
Initial tractor velocity (m/s)	u_o	0.5
Final tractor velocity (m/s)	u_f	0.5
Headland width (m)	D	28.0
Headland boundary angle (rad)	β	0
Row width (m)	d	10.0
Steering angle (rad)	δ	$[-\pi/6 \quad \pi/6]$
Vehicle velocity (m/s)	u	[0 3]
Vehicle acceleration (m/s^2)	a	$[-4 \quad 2]$

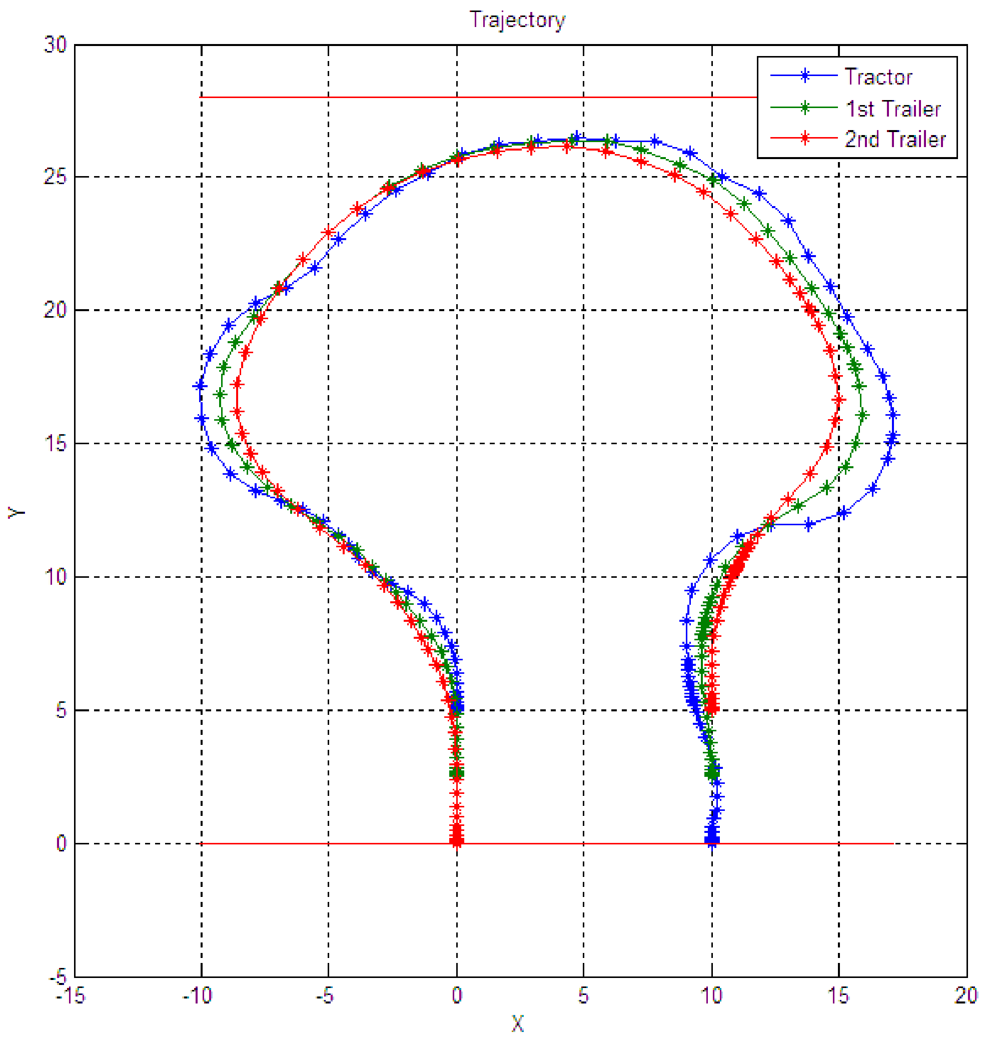


Figure 5.25: *Trajectory of headland turning by a tractor with two trailers hitched sequentially*

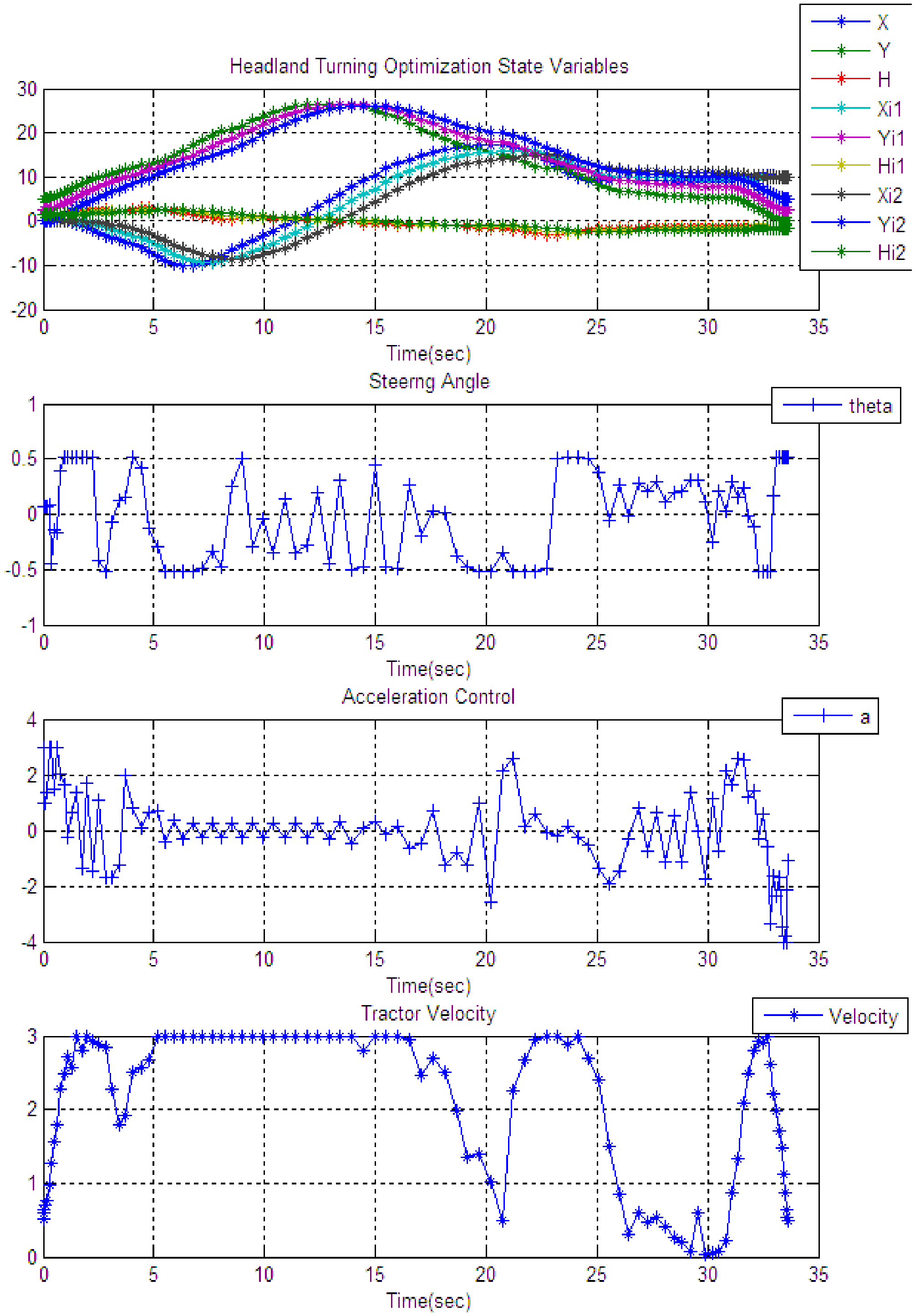


Figure 5.26: *States and controls of headland turning by a tractor with two trailers hitched sequentially*

The optimal headland turning time of this trial was 33.6102 seconds.

5.6 Conclusions

This research investigated the possibility and potential of using direct methods to solve the agricultural vehicle headland-turning optimization problem. A fresh research framework for the objective of minimum-time optimization was formulated. In this established framework, different tractor and tractor-implement vehicle models were created and applied. A direct collocation nonlinear programming toolbox, TOMLAB, was adopted in the MATLAB program development. The effectiveness and accuracy of the algorithm was evaluated and examined by a symbolic bulb turn example, and the result was promising. A series of simulation tests were carried out for further study and investigation. The results indicated that agricultural headland turning trajectory optimization can be accomplished by using numerical direct methods.

Based on this novel exploration, some helpful conclusions and thoughts were achieved for future work:

- The presented work was conducted using a vehicle kinematic model. To further the study, it is recommended that a vehicle dynamic model be adopted in the framework. A comparison will be conducted under the same boundary conditions and hardware constraints.
- The fishtail turn exhibited maneuverability advantages within a tighter headland space. However, the feasibility of a fishtail turn with towed implements needs to be further investigated and validated in field experiments.
- The study of on-hill headland turning implied the influence of terrain features on optimal trajectory generation. It also indicated the possibility of using direct methods to cope with complicated headland-turning scenarios in 3-D terrain.

References

Bell, T. (1999). Precision robotic control of agricultural vehicles on realistic farm trajectory.PhD dissertation, Stanford University.

Bchtis, D., Vougioukas, S., Tsatsarelis, C., and Ampatzidis, Y., (2007) Optimal Dynamic Motion Sequence Generation for Multiple Harvesters. Agricultural Engineering International: the CIGR Ejournal. Manuscript ATOE 07001. Vol. IX.

Betts, J. T..(1998) Survey of Numerical Methods for Trajectory Optimi-zation, Journal of Guidance, Control, and Dynamics, Vol. 21, No. 2, pp. 193–207. doi: 10.2514 / 2.4231

Conway, B.A., Paris, S.W.(2011). Spacecraft trajectory optimization using direct transcription and nonlin-ear programming. In: Conway, B.A. (ed.) Spacecraft Trajectory Optimization. Cambridge University Press, Cambridge.

Conway, B.A. (2012). A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems. J. Optim. Theory Appl. 152:271-306

Deng, W., and X. Kang (2003). Parametric study on vehicle-trailer dynamics for stability control. SAE Transactions, Journal of Passenger Cars, p. 1411-1419.

El-Gindy, M. (1989). Directional response of a tractor towing a semitrailer. International Journal of Vehicle Design, 10(2): 211-226.

Enright, P.J., Conway, B.A.(1992) Discrete approximations to optimal trajectories using direct transcription and nonlinear programming. J. Guid. Control Dyn. 15, 994–1002.

Fahroo, F., Ross, I.M.(2002) Direct trajectory optimization by a Chebyshev pseudospectral method. J. Guid. Control Dyn. 25, 160–166.

Feng, L., Y. He, Y. Bao, and H. Fang (2005). Development of trajectory model for a tractorimplement system for automated navigation applications. Instrumentation and Measurement Technology Conference, Ottawa, Canada, May 17-19.

Hargraves, C.R., Paris, S.W.(1987). Direct trajectory optimization using nonlinear programming and collo-cation. J. Guid. Control Dyn. 10, 338–342.

Jin, J. (2009). Optimal field coverage path planning on 2D and 3D surfaces. PhD dissertation, Iowa State University.

Oksanen, T., Visala, A. (2009) Coverage Path Planning Algorithms for Agricultural Field Machines. Journal of Field Robotics. 26(8), 651-668.

Oksanen, T., Visala, A. (2004). Optimal Control of Tractor-Trailer System in Headlands. Proceedings of the 7-8 October Conference. ASAE Publication Number 701P1004.

Ross, I.M., Fahroo, F.(2003) Legendre pseudospectral approximations of optimal control problems. In: Lec-ture Notes in Control and Information Sciences, vol. 295, pp. 327–342. Springer, New York.

Snider, J.M., (2009). Automatic Steering Methods for Autonomous Automobile Path Tracking. Dissertation. Robotics Institute, Carnegie Mellon University.

Torishu, R, S. W. Mugucia, and J. Takeda (1992). The kinematics and open-loop characteristics of tractor-trailer combinations. Journal of Faculty of Agriculture, Iwate University 20: 299-314.

CHAPTER 6: GENERAL CONCLUSIONS AND RECOMMENDATIONS

6.1 General Conclusions

This research investigated navigation control of autonomous agricultural vehicles under off-road conditions. The research work consisted of two main objectives, robust navigation control and headland turning optimization. For the first objective, an autonomous 4WS/4WD robot, AgRover, was designed and developed, and a robust backstepping-based SMC navigation controller was developed and implemented. For the second objective, an innovative research framework of headland turning optimization problem for agricultural vehicles was formulated and proposed. Direct collocation numerical method was investigated to generate optimized solutions under various assumed headland turning scenarios.

SMC was proven to be an effective tool, both theoretically and practically. It was successfully implemented on the AgRover to accomplish automatic navigation control of trajectory tracking tasks. The quantified experimental results of the field experiments were promising and impressive. The controller performed well in AgRover off-road tests.

The headland turning optimization research investigated the possibility and potential of using direct methods to solve the agricultural vehicle headland-turning optimization problem. In the proposed framework, different tractor and tractor-implement vehicle models were created and applied. A direct collocation nonlinear programming toolbox, TOMLAB, was adopted in the MATLAB program development. The effectiveness and accuracy of the algorithm was evaluated and examined by a symbolic bulb turn example, and the result was promising. A series of simulation tests were carried out for further study and investigation. The results indicated that agricultural headland turning trajectory optimization can be accomplished by using numerical direct methods.

6.2 Recommendations for Future Work

There are several potential areas where this work can be improved and/or expanded. Some of the important areas are listed as follows.

- The SMC controller is not perfect. The error rate was consistently between 0.1m and 0.2m, which could be improved. It is expectable that with improved hardware responsiveness and sampling frequency, the controller performance could be improved correspondingly. The “chattering” issue existed in most of the experiment results, which was shown in the tracking trajectory. Improved adaptive control and fuzzy logic control can help and it can be implemented with an SMC controller to solve and ease this issue.
- The control parameters were sensitive to the system dynamic performance. In future, computing the sensitivity and the relationship among individual control parameters could be a potential research interest.
- The study showed the potential to adopt a vehicle dynamic model to the controller design if the hardware capability allows.
- Also, the presented headland turning optimization work was conducted using a vehicle kinematic model. To further the study, it is recommended that a vehicle dynamic model be adopted in the framework. A comparison will be conducted under the same boundary conditions and hardware constraints.

- The fishtail turn exhibited maneuverability advantages within a tighter headland space. However, the feasibility of a fishtail turn with towed implements needs to be further investigated and validated in field experiments.
- The study of on-hill headland turning implied the influence of terrain features on optimal trajectory generation. It also indicated the possibility of using direct methods to cope with complicated headland-turning scenarios in 3-D terrain.