

POLITECNICO DI TORINO

Corso di Laurea in Ingegneria dell'Autoveicolo

**The Design of Control Module of Autonomous
Driving Truck in Mining Area**



Advisor:
Ing. Massimiliana Carello

Candidate:
Dong Jianhao

December 2020

ABSTRACT

For a long time, the working environment in the mining area has been very harsh, and more and more young people are unwilling to work there. Autonomous driving technology in the mining area can replace drivers, greatly improving the production efficiency and reducing the occurrence of safety accidents.

As the final execution module in driverless vehicle technology, the control module is very important. This thesis will design the lateral and longitudinal controller respectively to enable the vehicle to track the planned trajectory stably and accurately.

The longitudinal controller adopts the dual closed-loop PID control of speed and acceleration. Different speed requirements have been tested in a simulation environment, and they have also been verified on actual vehicles.

The lateral controller adopts the lateral control algorithm based on geometric model like Pure Pursuit, Stanley, and the algorithm based on vehicle model such as Rear-Wheel Feedback, LQR and MPC. In the simulation environment, the reference trajectory under different conditions is tested, and the advantages and disadvantages of each algorithm are finally evaluated by comparing errors.

Keywords: Autonomous Vehicle, ROS, Control Algorithm

CONTENTS

1. Introduction.....	1
1.1 Background	1
1.2 General Autonomous Driving Technologies.....	1
1.3 The Autonomous Driving Vehicle in Mining Area.....	3
1.3.1 Mining Area.....	3
1.3.2 Characteristics of Autonomous Driving in Mining Area.....	4
1.4 Layout of The Thesis.....	5
2. Literature Review of Vehicle Control.....	6
2.1 Longitudinal Control	6
2.2 Lateral Control.....	7
2.3 Centralized Control.....	8
3. Simulation Environment	10
3.1 Robot Operation System ROS.....	10
3.2 Visualized Simulator—Gazebo	11
3.3 Test Trajectory	12
4. Longitudinal Controller Design.....	14
4.1 Control principle block diagram.....	14
4.2 Challenges	15
4.2.1 Slope Compensation	15
4.2.2 Inaccurate Acceleration	15
4.3 PID	16
4.3.1 Parameters Introduction	16
4.3.2 Determine PID type	18
4.4 Simulation Result.....	18
4.5 Real Truck Test	19
5. Lateral Controller Design	23
5.1 Control Methods Based on Geometric Model	23
5.1.1 Ackerman steering model	23
5.1.2 Pure Pursuit Control	24
5.1.3 Stanley Control.....	30
5.2 Vehicle Model.....	35
5.2.1 Vehicle Kinematic Modeling	35
5.2.2 Vehicle Dynamic Modeling	36
5.2.3 Model Linearization and Discretization.....	39
5.2.4 Dynamic Model in terms of error variables	43
5.3 Rear-Wheel Feedback Algorithm	44
5.4 LQR Control.....	49
5.4.1 Introduction.....	49
5.4.2 LQR Formula Proof.....	49
5.4.3 Control Block.....	50
5.4.4 Simulation Result	51
5.5 MPC.....	55

5.5.1	Introduction	55
5.5.2	Principle	55
5.5.3	Increment Model	55
5.5.4	Predictive Model	56
5.5.5	Cost Function Design	57
5.5.6	Constraints	58
5.5.7	Quadratic programming	59
5.5.8	Simulation Result	60
6.	Conclusions and Future Work	69
6.1	Simulation Result Comparison	64
6.2	Conclusions	69
6.3	Future Work	69
	References	71

1. Introduction

1.1 Background

Society is constantly evolving and technology is improving day by day. Our society is full of "machine intelligence", from vacuum cleaner robots that help clean our home, to airplanes controlled by computer landing safely in unpredictable weather. Across the last century, we have witnessed more and more devices used to replace human such as washing machines, microwave ovens, motor vehicles, and personal computers and information technology that can improve productivity and convenience of life.

As the most important mobility tool for human beings, vehicles not only improve people's travel efficiency, but also help many industrial activities. However, driving a car could be a boring, tedious, and dangerous activity. Every moment we must pay attention to the changes in the surrounding environment when driving. We will face the danger of extremely poor road conditions. We may question the skills or judgments of other drivers, and even we have to pay attention to our fatigue or mistakes. In fact, these driving behaviors and some skilled driving experience may cause the driver's attention to be negligent, and accidents may occur. Driver errors are the main cause of most accidents, and about half of these accidents are due to delays in driver response.

Since 2000, with the development of disciplines such as artificial intelligence, pattern recognition, computer vision, and electronics, vehicles are no longer just a complex mechanical structure. It combines many electronic control systems and integrates many advanced scientific research results. For example, the application of the ABS system improves the safety and stability of the vehicle, and the application of the active suspension reduces the discomfort when people ride in the vehicle, and makes the vehicle's driving more stable.

But what we hope more is that the car can have advanced functions such as automatic planning of driving paths, automatic recognition of roads, automatic driving, and so on, so that drivers can be released from boring driving behaviors, making driving safer. So now driverless vehicle technology gets more and more people's attention.

1.2 General Autonomous Driving Technologies

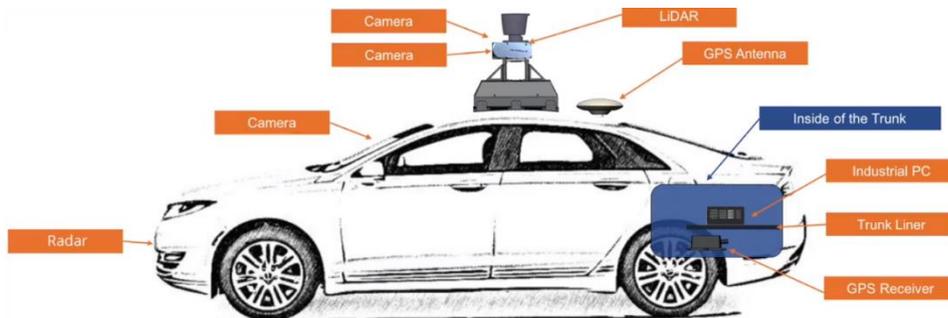


Figure 1.1 Hardware of Autonomous Vehicle

The autonomous driving vehicle is a kind of intelligent mobile robot that involves so many disciplines such as environment perception, localization and navigation, path and trajectory planning, and vehicle motion control[1]. Their research goal is to replace human drivers to conduct autonomous driving of the vehicle and thus improves driving safety and efficiency.

The hardware configuration of autonomous vehicle is shown in Figure 1.1.

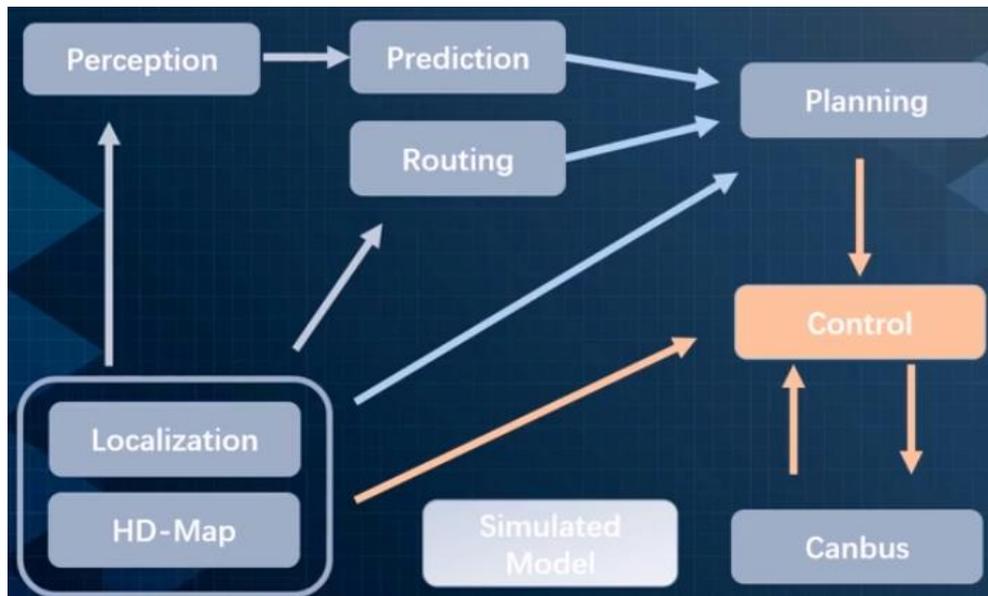


Figure 1.2 Software Structure

Figure 1.2 shows the software structure of autonomous driving system.

Perception: The driverless vehicle is equipped with sensors like Lidar, radar and camera to get the information of the surrounding environment and obstacles such as moving vehicles and people as well as other things. The information is used to construct an environment model to get available space for driving.

Prediction: When moving objects are detected by the perception module, the driverless system has to predict their future pose through their speed and direction to plan the reasonable behavior of ego vehicle to avoid collision.

Localization: localization provides the accurate position of the ego vehicle which is necessary for the planning and control module. The commonly used methods are the GNSS RTK, Lidar localization, IMU, and the combination of these sensors using sensor fusion technology.

Planning: It includes two parts: global planning (Routing) and local planning. The global planning refers to that after setting the endpoint of driving, the smart car obtains its position through the localization device, loads a high-definition map, and calculates the path from the current position to the endpoint through a path search algorithm. Local path planning refers to the real-time detection of road and obstacle information in front of the vehicle through the camera and lidar while the vehicle is running, and planning a reasonable path to bypass these obstacles.

Control: Vehicle motion control technology refers to making the vehicle track the target trajectory produced by planning module through appropriate control strategies. The motion control of the vehicle can be decomposed into longitudinal speed control and lateral steering control. There is a certain coupling relationship between the two, but they can be controlled separately in usual situation.

1.3 The Autonomous Driving Vehicle in Mining Area

1.3.1 Mining Area



Figure 1.3 Mining Area

One of the main jobs in the mining area is the transportation of ore materials. Each small or medium-sized coal mine needs more than 100 vehicles and more than 200 drivers, so the mining area needs to recruit a large number of truck drivers to complete the transportation work, which means that there is a vast demand for autonomous driving technologies.

But in recent years, fewer and fewer people are willing to work in the mining area, mainly for the following reasons:

- The working environment in the mining area is harsh, the air is full of dust, and even radioactive materials, which is very harmful to the human body.
- Poor road conditions and imperfect other infrastructures lead to high driving risk.
- The work place is far away from the city, people especially young people are not willing to work here.

Therefore, the recruitment of drivers in the mining area is very difficult, and the labor cost is very large. The entire mining industry uses tens of billions to recruit truck drivers every year, and they have not yet solved the transportation safety and efficiency issues. Safety accidents in mining areas occur from time to time, and each occurrence is a misfortune of a family, and enterprises need to pay a large amount of compensation to their families.

If the autonomous driving technology is used to carry out unmanned transformation of the mining area, the annual employment cost and accident rate of the mining area can be greatly reduced. At the same time, by intelligently scheduling driverless trucks, industrial activities can be reasonably and effectively arranged, and the operation efficiency of the mining area can be greatly improved.

1.3.2 Characteristics of Autonomous Driving in Mining Area

At present, many enterprises focus on the development of autonomous driving passenger cars, but for the current technical level, urban roads are still too complicated, and there are too many uncontrollable variables, which is difficult to guarantee safety.

As a closed scene, the mining area is more suitable for the deployment of autonomous driving technologies.

- The transportation roads in the mining area are almost determined. Each mine truck carries out point-to-point transportation on a relatively fixed route in a single shift, and the speed is often lower than 30km/h, which greatly reduces the real-time requirements for perception, planning and control modules.
- The mining area is not a road scenario, but an operation scenario. Unlike passenger cars, it won't be restricted by too many laws and regulations.
- Driverless trucks have no passengers during operation, so the safety requirements will be slightly lower than that of passenger cars.



Figure 1.4 Autonomous Minecart

1.4 Layout of The Thesis

Chapter 2 introduces the current research situation of longitudinal and lateral control algorithm, as well as the centralized control algorithm to control longitudinal and lateral motion simultaneously.

Chapter 3 introduces the Robot Operation System which can be used to construct the software frame of robot. Combined with Gazebo, which is a visualization simulation tool with physical engine and will be introduced detailed later, we can build a simulation platform to test and visualize the performance of control algorithm.

Chapter 4 explains the design of longitudinal controller using dual closed-loop PID. In this case, it is necessary to take into account the ramp occurring frequently, and the physical limitation of actuators.

Chapter 5 presents the design of lateral controller based on geometric, kinematic and dynamic vehicle model using Pure Pursuit, Stanley, Rear-Wheel Feedback, LQR and MPC algorithm. Then their simulation results are compared.

Chapter 6 gives the conclusion of this thesis and introduces the challenge in more extreme condition for control module.

2. Literature Review of Vehicle Control

2.1 Longitudinal Control

Longitudinal control is based on vehicle and road information, by controlling the driving or braking of the intelligent vehicle, and accelerating or decelerating according to the planned speed, thereby achieving accurate and fast tracking of the planned longitudinal behavior. In order to solve the longitudinal control problem of intelligent vehicles, scholars at home and abroad have used different methods to do some discussion and research.

CHOI S, et al.[2] uses intelligent PID method to compensate the unmodeled dynamic characteristics in the longitudinal control of vehicle start-stop conditions.

Kingston University SHAKOURI et al.[3] designed two controllers of gain scheduling PI and Linear Quadratic Regulator (LQR) to control the throttle opening. Among them, LQR depends on the accurate model.

At University of Michigan XU et al.[4] proposed a longitudinal vehicle speed tracking method based on optimal preview control, introducing multi-point preview road gradient increments and expected vehicle speed increments into new state vectors to reconstruct the nonlinear optimal control problem into Augment the LQR problem and get an analytical solution to the optimization problem while reducing the computational burden. Compared with PI control, the tracking accuracy of the optimal preview control is higher and the action is smoother.

In order to improve the system's robustness to parameter uncertainties and external disturbances, GERDES et al.[5] of the University of California at Berkeley adopted a sliding mode control method. First, the sliding mode control was used to calculate the speed or position closed-loop to obtain the desired acceleration, and then according to the inverse longitudinal dynamic model to calculate the driving/braking torque, and finally realize the tracking of the desired torque by the driving and braking actuators through sliding mode control.

The University of Pavia, FERRARA et al.[6] used the second-order sliding mode control method to realize the longitudinal control of the vehicle fleet through the second-order sliding mode control method to ensure the safety distance of adjacent vehicles.

Tongji University HANG et al.[7] combined RBF neural network with sliding mode control to design an adaptive vehicle speed control law.

NARANJO J E, et al.[8, 9] adopts the fuzzy logic control method in longitudinal control. Although fuzzy control does not require accurate system modeling, the establishment of a practical rule base relies on expert knowledge and usually requires a large number of test calibrations.

2.2 Lateral Control

The target of lateral control is to control the front wheel steering angle, so as to achieve accurate and fast tracking of planned path.

Pure pursuit algorithm is the easiest lateral control method proposed by Carnegie Mellon University scholars[10, 11].

In 2005, Stanley Racing at Stanford University applied the Stanley algorithm to win the US Defense Advanced Research Projects Agency (DARPA) Desert Challenge [12]. The Stanley algorithm is more suitable for relatively higher speed driving conditions than the pure pursuit control algorithm, but it requires smoother desired path. In the case of unsatisfactory road curvature smoothness, it is prone to vehicle overshoot. Due to the neglect of the dynamic characteristics of the vehicle and the dynamic characteristics of the steering actuator, the tracking performance is poor when the lateral acceleration of the vehicle is large.

At Stanford University HOFFMANN et al.[13] further improved the control law through yaw angular velocity deviation compensation and steering correction, which can improve the path tracking performance of Stanley algorithm to a certain extent.

The above all are control theory researches based on vehicle geometric model.

Beijing Institute of Technology Zhao Xijun et al.[14] adopted heading error feedback control, first determined the expected heading error from the kinematic relationship, and then designed a segmented fixed gain PID algorithm based on the vehicle-road dynamics model according to the driving speed. The method has good tracking effect on unstructured roads.

The second University of Rome, MARINO et al.[15] proposed a lateral control method based on cascade PID, in which the outer loop circuit obtains the desired yaw rate according to the lateral displacement error, and the inner loop circuit uses PI control to achieve the yaw rate tracking and passes The pole configuration adjusts the controller parameters to ensure the gradual stability of the entire system. However, the algorithm is designed based on the linearized model of the kinematics of the vehicle relative to the path. It cannot guarantee the progressive signing of the lateral displacement error within a large range, and the algorithm does not consider the effect of actuator constraints on the control effect.

University of Michigan XU et al.[16] proposed a path tracking algorithm based on optimal preview control, which introduces multi-point preview road curvature in a finite time window into the augmented state vector, and reconstructs the nonlinear optimal control problem into augmented LQR Problem, the optimal steering control law is composed of feedback control to stabilize tracking error and feedforward control to deal with future road curvature. Compared with the LQR control without preview, the optimal preview control can greatly reduce the tracking error and reduce the overshoot to make the steering operation smoother.

Based on the control framework of the combination of road curvature feedforward and LQR feedback, GOODARZI et al.[17] added direct yaw moment control on the basis of active steering of the front wheels to achieve the large lateral acceleration.. Introducing actuator conditions such as distributed drive to increase the control freedom of the system can not only improve the control margin of unmanned vehicles under extreme conditions, but also help to increase the response speed of lateral motion.

Falcone et al. [18] of Sanio University designed a lateral controller based on nonlinear model predictive control and linear time-varying model predictive control for the lane change scenario of driverless vehicles on low-adhesion roads. Constraints limit the tire force in the linear region to ensure the stability of the vehicle. The analysis gives the asymptotic stability conditions of the linear time-varying model predictive control closed-loop system [19].

2.3 Centralized Control

Lateral control and longitudinal control are always discussed separately in most case. But some experts and scholars at home and abroad have carried out relevant research about controlling longitudinal and lateral behavior coupled.

University of California, Berkeley, LIM et al.[20] analyzed the coupling mechanism of vehicle longitudinal and lateral motion, and designed a control law to compensate for the coupling impact based on dynamic surface sliding mode control. Simulation shows that considering the coupling impact can effectively improve the trajectory tracking performance. However, when the algorithm uses the tire inverse model to solve the front wheel steering angle, the analytical solution cannot be obtained, and the mapping relationship is not accurate in practical applications.

In order to expand the scope of application conditions, the Berkeley University of California LEE et al.[21] removed the above simplified assumptions and designed a robust adaptive control algorithm for the MIMO nonlinear system using the backstepping method.

Some authors [22, 23] compensate the influence of parameter uncertainty, strong nonlinearity and coupling effects of trajectory tracking system through neural network.

Massachusetts Institute of Technology PETERS et al.[24] proposed a trajectory tracking control method based on differential flat theory. The center position of the front axle is used as a differential flat output, and the stability analysis of the internal dynamic yaw dynamics during vehicle trajectory tracking is performed.

In [25] more attention to the coupling characteristics of vertical and lateral movements of vehicles has been paid. But these studies have only been verified by simulation.

Under the MPC framework, the longitudinal and lateral control can be transformed into the same constrained optimization problem to fully consider the vehicle motion coupling effect. The California Berkeley University GAO [26, 27] proposed two MPC-based control architectures for vehicle obstacle avoidance, one is planning and control

layered implementation, and the other is planning and control implemented in the same MPC controller. The advantage of hierarchical control is that the planning can use a simpler model, the calculation amount is relatively smaller, and the integrated control can prevent the planning from generating unfeasible trajectories, but the real-time application is more difficult.

3. Simulation Environment

3.1 Robot Operation System ROS

ROS is a highly flexible software architecture for robot programs. It contains a large number of tool software, library code and agreed protocols, and aims to simplify the difficulty and complexity of the process of creating complex and robust robot behaviors across robot platforms.



Figure 3.1 ROS Sign

The main goal of ROS is to provide code reuse support for robot research and development.

Figure 3.2 illustrates the communication mechanism of ROS.

ROS is a distributed process (also known as "node") framework. These processes are encapsulated in packages and function packages that are easy to share and publish. Each node is independent of each other, and even the computer language of operation can be different. All the user needs to do is to define the message format for communication between the two nodes, so that the communication between different nodes can be carried out.

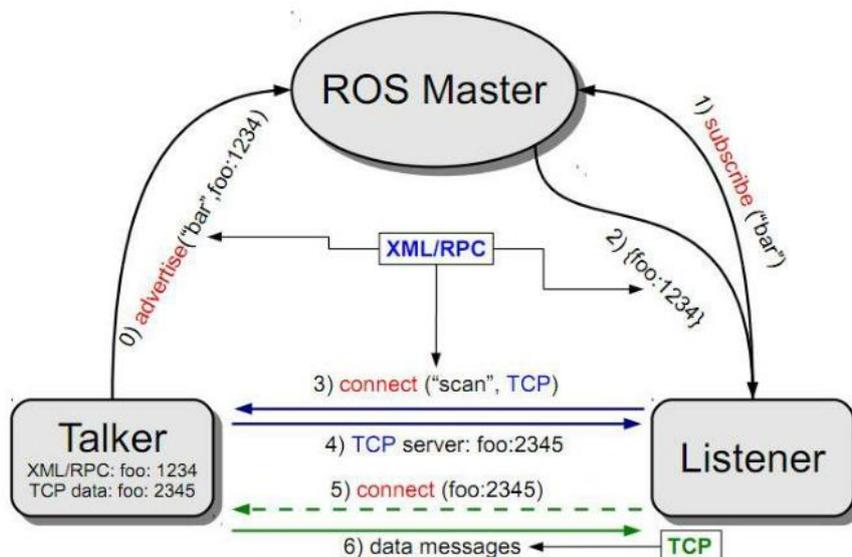


Figure 3.2 ROS Communication Mechanism

Another huge advantage of ROS is that it has a large number of open source libraries. When users only want to study a certain aspect of a system, they can use the open source function package to build other parts. For example, if you want to use ROS

to develop control algorithms for autonomous vehicles, you can first use the open source function package to implement the perception and planning module, and then implement the control module yourself.

3.2 Visualized Simulator—Gazebo

Gazebo is a 3D dynamic simulator that can accurately and effectively simulate robot in complex indoor and outdoor environments.

Gazebo provides high-fidelity physical simulation, which provides a world, a complete set of sensor models, robot models (shown in Figure 3.3), and very user-program-friendly interaction.



Figure 3.3 Gazebo Vehicle Model

Gazebo works well with ROS. Users can deploy software modules in ROS and then run the vehicle model in Gazebo.

The workflow is demonstrated in Figure 3.4. The software module sends control commands to the vehicle model in Gazebo. With the physics engine, the vehicle model will start to move and continuously feedback its own localization and movement information to ROS for error calculation, so as to achieve closed-loop control.

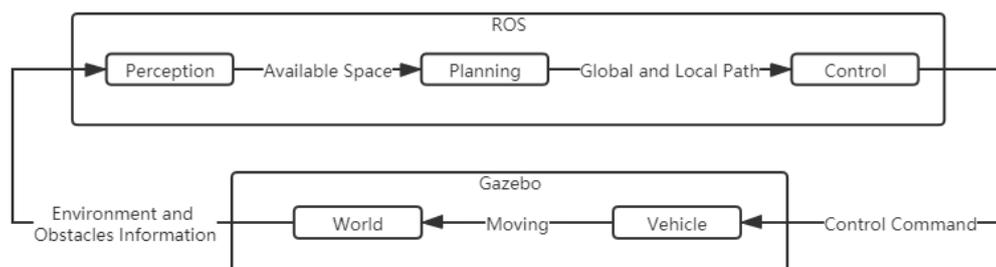


Figure 3.4 ROS-Gazebo Workflow

The right part of Figure 3.5 is real-time 3D simulation result, the left part is a plot which compares the reference trajectory and actual trajectory, and also shows the real time tracking error data.

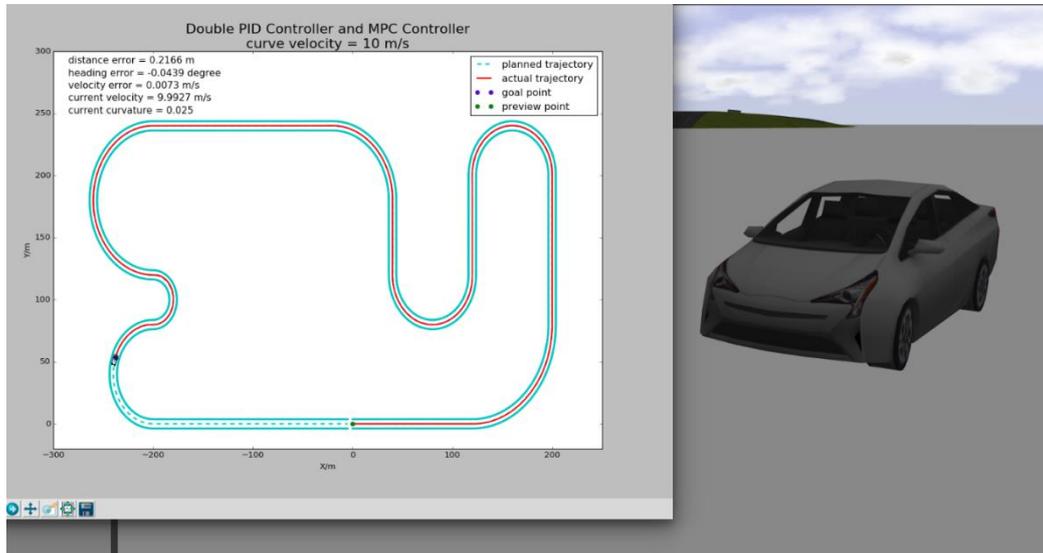


Figure 3.5 ROS-GAZEBO Simulation

3.3 Test Trajectory

Figure 3.6 is the reference trajectory, Figure 3.7 and 3.8 are the reference velocity and heading angle. They will be used to test the performance of control algorithms.

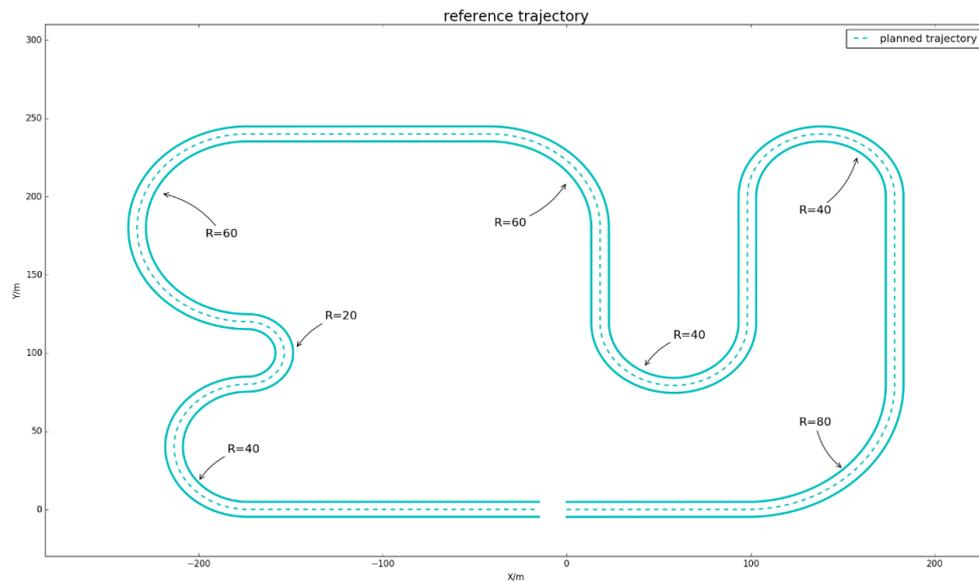


Figure 3.6 Reference Path

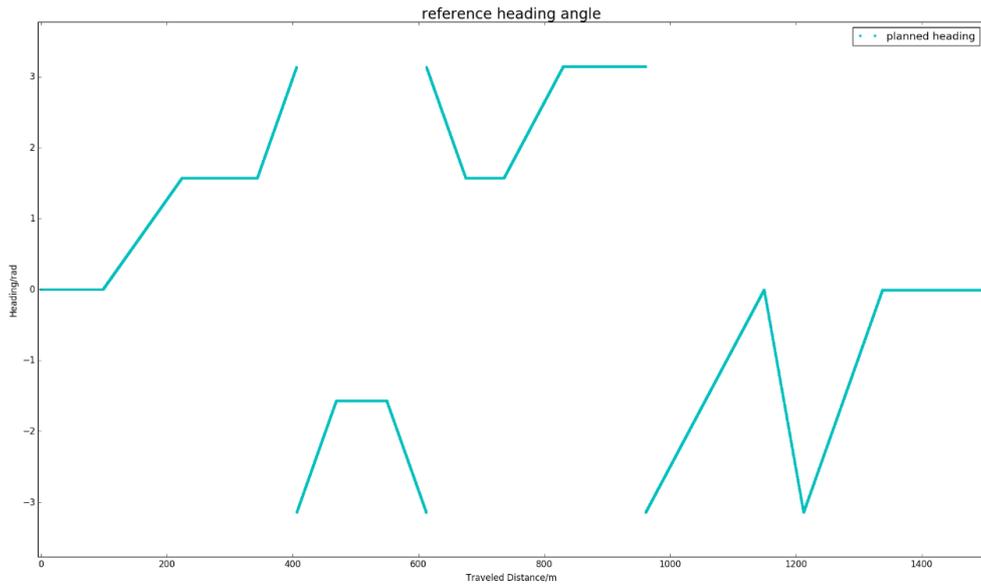


Figure 3.7 Reference Heading

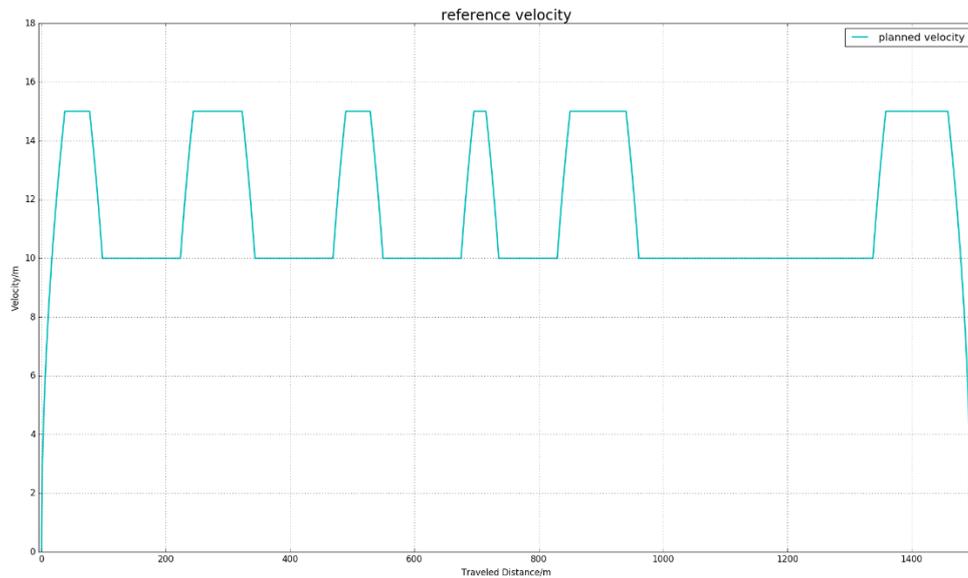


Figure 3.8 Reference Velocity

In Figure 3.8, the higher velocity means the velocity of vehicle running in straight road, and the lower one means the velocity in bend.

4. Longitudinal Controller Design

The longitudinal control of autonomous vehicles is mainly to study how to control the longitudinal movement of the vehicle, that is, to control the vehicle to drive at the desired speed, the distance between vehicles, the longitudinal acceleration of the vehicle, the throttle opening and the braking amount of the brake.

Longitudinal control usually adopts a hierarchical control structure. The upper controller produces the acceleration compensation based on the distance or speed in a closed loop, and the lower controller controls the throttle and brake to track the desired acceleration. [28]

The upper-level controller currently uses PID control. Its main advantage is that it does not rely on an accurate longitudinal model of the vehicle. Although the control effect under extreme conditions is average, the control conditions of the mining area are relatively normal, so it can also meet the control requirements.

The lower-level controller can realize the real-time tracking of the desired acceleration through methods such as the inverse solution of vehicle longitudinal dynamics, engine MAP look-up table or PID control.

4.1 Control principle block diagram

If the accurate performance parameters of the engine can be obtained, we can establish an accurate Engine Map to build a calibration table. The longitudinal control block diagram is shown following [29]

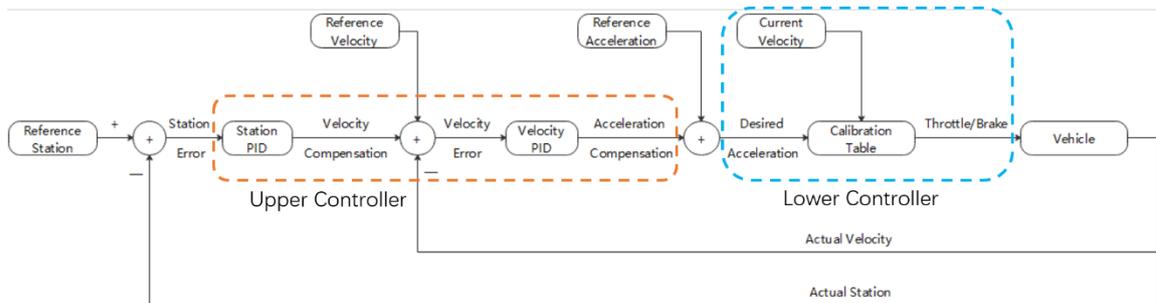


Figure 4.1 PID Control Block 1

In Figure 4.1, the lower-level controller is implemented through a calibration table, which establishes the relationship between expected acceleration and throttle opening at the current vehicle speed, so that the vehicle can quickly and accurately track speed at any vehicle speed.

The upper controller is the dual closed-loop control of position and speed, so that not only speed tracking can be carried out, but also position tracking can be realized.

In fact, the control requirements of unmanned driving in the mining area are different from the urban road environment. There is no need for position tracking, and

only the task of speed tracking is required. At the same time, due to limited conditions, there is no way to get all the power-related parameters of the mincart, and it is impossible to establish an accurate vehicle longitudinal dynamics model and Engine Map.

However, in actual driving behavior, human drivers do not understand the longitudinal dynamics of the vehicle and can also control the vehicle well. Therefore, both the upper and lower controllers can be designed with PID. [30]

If it is under the full vehicle speed range, the lower-level controller using PID cannot achieve the best effect, because the optimal gains of PID vary with vehicle speed. Fortunately, the speed of the vehicle in the mining area is slow, and the speed range of the vehicle is also small. The PID can achieve relatively good results even without gain scheduling, so the following simplified version of the control block diagram is obtained, shown in Figure 4.2.

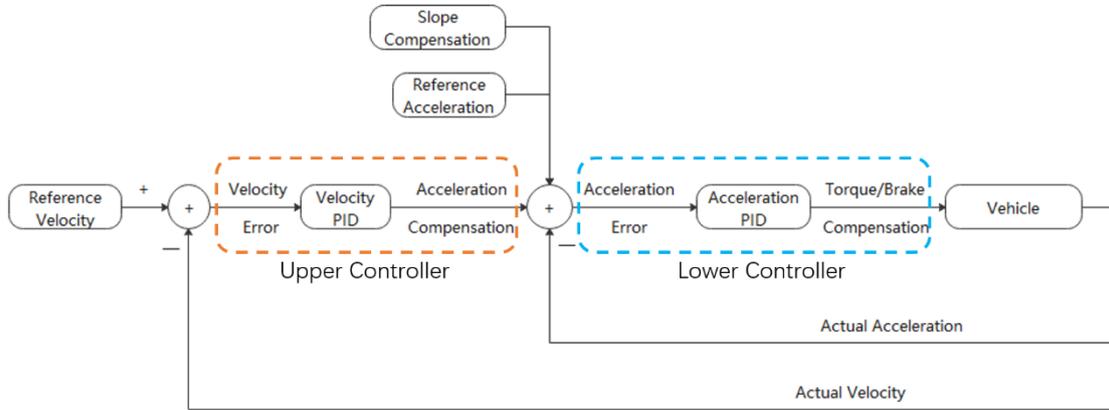


Figure 4.2 PID Control Block 2

4.2 Challenges

Due to poor conditions in the mining area, there are the following problems.

4.2.1 Slope Compensation

There are long uphill and downhill in the mining area. When the mine truck is uphill, it will be affected by the component force of gravity along the ramp, resulting in an increase in resistance. This means that the vehicle needs to output more power to compensate, and the downhill is in contrast. The component force of the vehicle's gravity along the ramp can be calculated from the pitch angle α of vehicle. [31]

$$\text{slope compensation} = g * \sin\alpha$$

However, these ramp roads are not smooth. It is not accurate to calculate the slope compensation directly using the current vehicle pitch angle data provided by IMU. Therefore, it is necessary to deal with the obtained slope data for a period of time before the current time using median filter processing.

4.2.2 Inaccurate Acceleration

GPS and IMU are installed on the mine truck. IMU can provide acceleration data, but due to the poor road conditions in the mining area, the vehicle is very bumpy during driving, so the acceleration can only be obtained by differential using speed data provided by GPS. But since the frequency of GPS is very high, and the velocity signal has error, differential processing will amplify the error.

Therefore, a timer is used in the program, using the speed data subscribed from GPS every 0.01 s, and calculate the approximate value of the actual acceleration.

4.3 PID

PID controller is the most widely used automatic controller. It is suitable for occasions where the controlled object model is not well established. The principle is demonstrated in Figure 4.3.

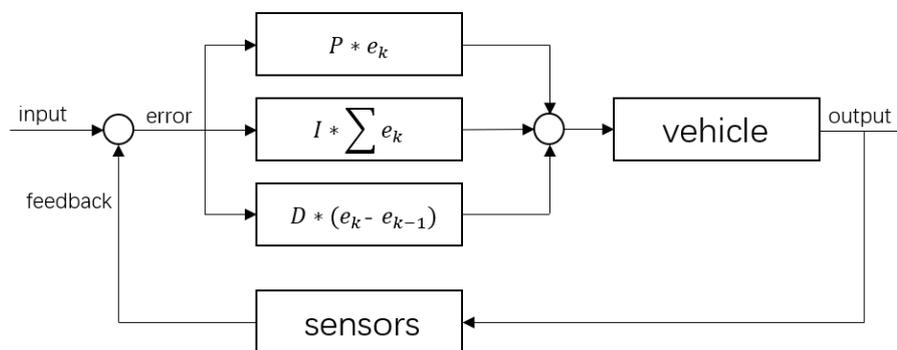


Figure 4.3 PID Principle

It has the following advantages:

- no requirement for model
- simple principle
- easy implementation
- wide application
- independent control parameters

4.3.1 Parameters Introduction

a) Proportional gain

Figure 4.4 shows the effect of proportional gain.

- As proportional gain increases, the response will be faster, but the overshoot also increase.
- If there is a steady-state error under proportional control, increasing proportional gain can reduce the steady-state error.

- Proportional gain controller is only suitable for occasions where there is no steady state error.

Figure 4.4 shows the effect of proportional gain.

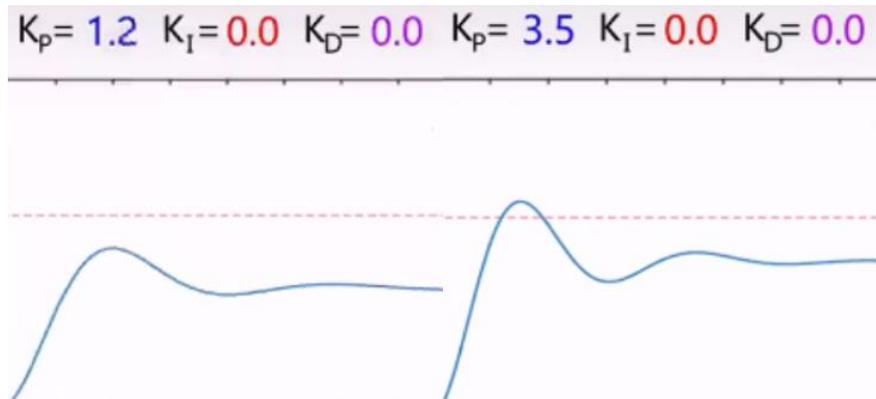


Figure 4.4 Effect of Proportional Gain

b) Integral gain

Figure 4.5 shows the effect of integrative gain.

- Increasing the integral term can eliminate the steady-state error, but it also enlarges the overshoot and reduce the system stability. The magnitude of integral gain affects the speed of elimination of steady-state errors.
- Due to the limitation of the actual actuator, special treatment is required to avoid integral saturation.
- Integral term cannot be used alone, PI controller is suitable for controlled objects with high requirement for steady-state accuracy.

Figure 4.5 shows the effect of integrative gain.

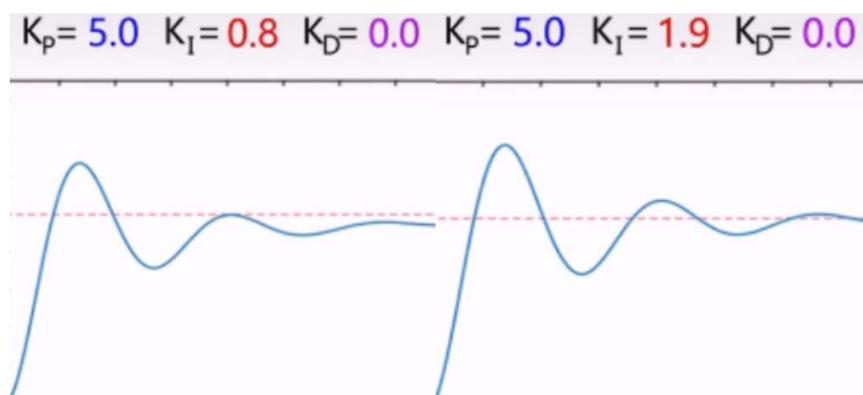


Figure 4.5 Effect of Integrative Gain

c) Differential gain

Figure 4.6 shows the effect of differential gain.

- Increasing the differential term can effectively reduce the oscillation, but it should be noted that if there are noise or high frequency components in the signal, it will be amplified by the differentiator.
- Differential term cannot be used alone too, the PD controller is mainly used for controlled objects that require high dynamic performance and are not affected by high-frequency noise.

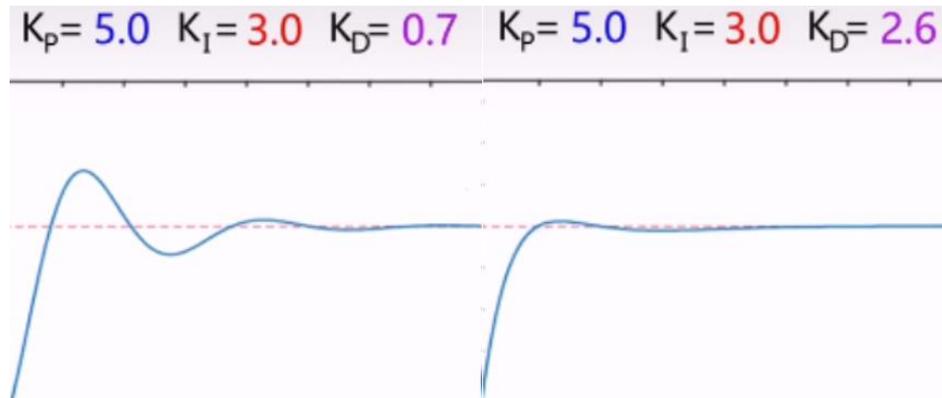


Figure 4.6 Effect of Differential Gain

4.3.2 Determine PID type

For the upper controller, controller with proportional gain is enough.

For the lower controller, the differential term is not allowed. Since the acceleration used is obtained by differential calculation of velocity, the difference between adjacent accelerations signal will be very large. If the differential term is used in this case, the error signal will be amplified largely.

During the driving of the vehicle, the friction between road surface and tire, and the internal resistance of the vehicle, as well as the air resistance will cause steady-state error in the speed control process. Therefore, it is not enough to use the controller with only proportional gain, integral term can eliminate steady state error effectively.

Therefore, PI controller is used.

4.4 Simulation Result

The longitudinal controller is tested in the scenario with velocity equal to 10m/s and 15m/s in simulation platform.

Figure 4.7 and 4.8 indicate the result of velocity tracking with 10m/s and 15m/s velocity in curve.

The average errors are 0.0456m/s and 0.221m/s respectively.

The tracking error under higher velocity is larger.

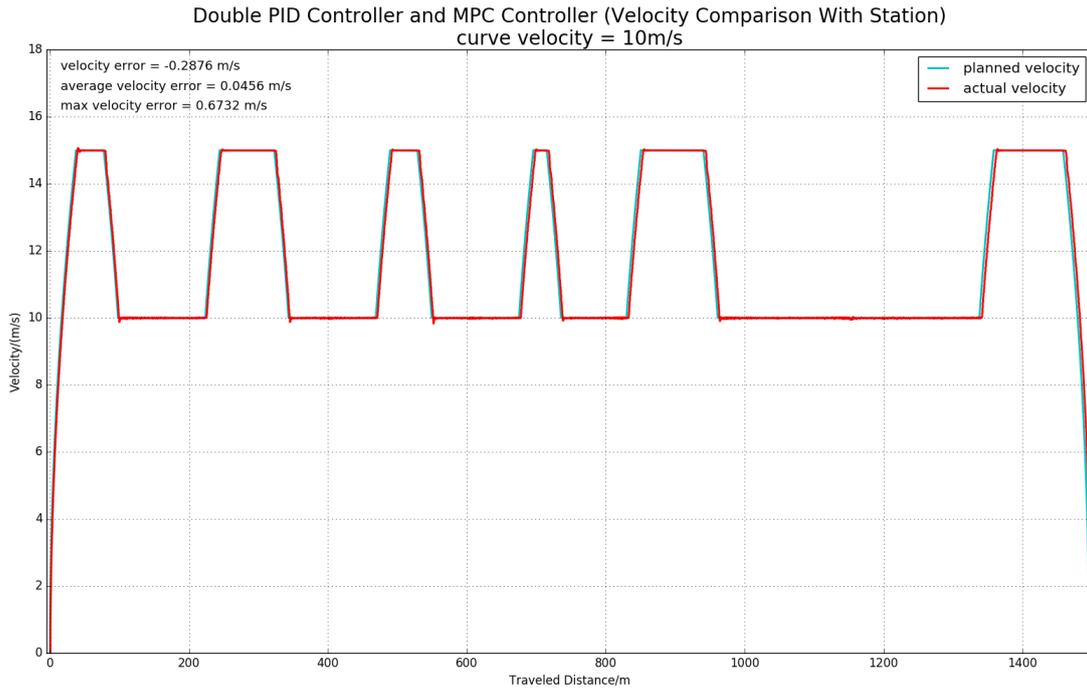


Figure 4.7 Simulation Result of Vehicle Tracking at V=10 m/s

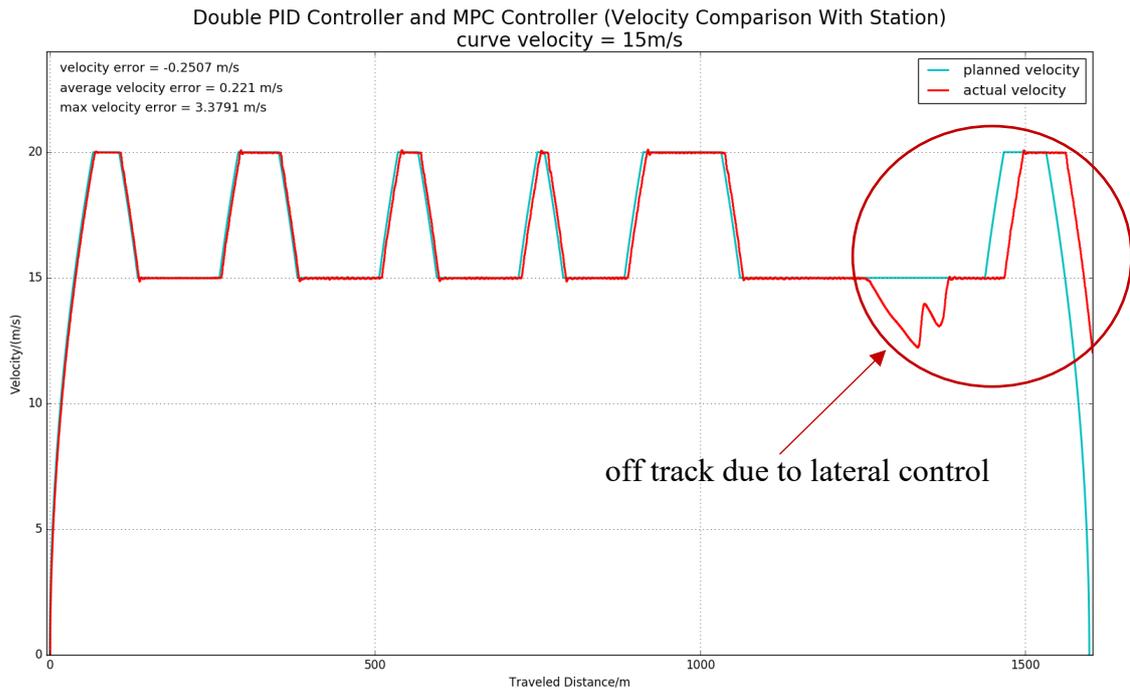


Figure 4.8 Simulation Result of Vehicle Tracking at V=15m/s

4.5 Real Truck Test

The longitudinal controller is also tested in real truck in mining area for uphill and downhill.

Figure 4.9 is the test route including uphill and downhill.



Figure 4.9 Real Vehicle Test Environment

Figure 4.10 shows the road condition in the flat road which is the first part of test route.



Figure 4.10 Flat Road

Figure 4.11, 4.12, 4.13 show the downhill condition.



Figure 4.11 Downhill 1



Figure 4.12 Downhill 2



Figure 4.13 Downhill 3

Figure 4.14 is the uphill road part.



Figure 4.14 Uphill

Figure 4.15 and 4.16 indicate the velocity tracking result in uphill and downhill condition.

The average error is 0.09m/s and 0.04m/s respectively.

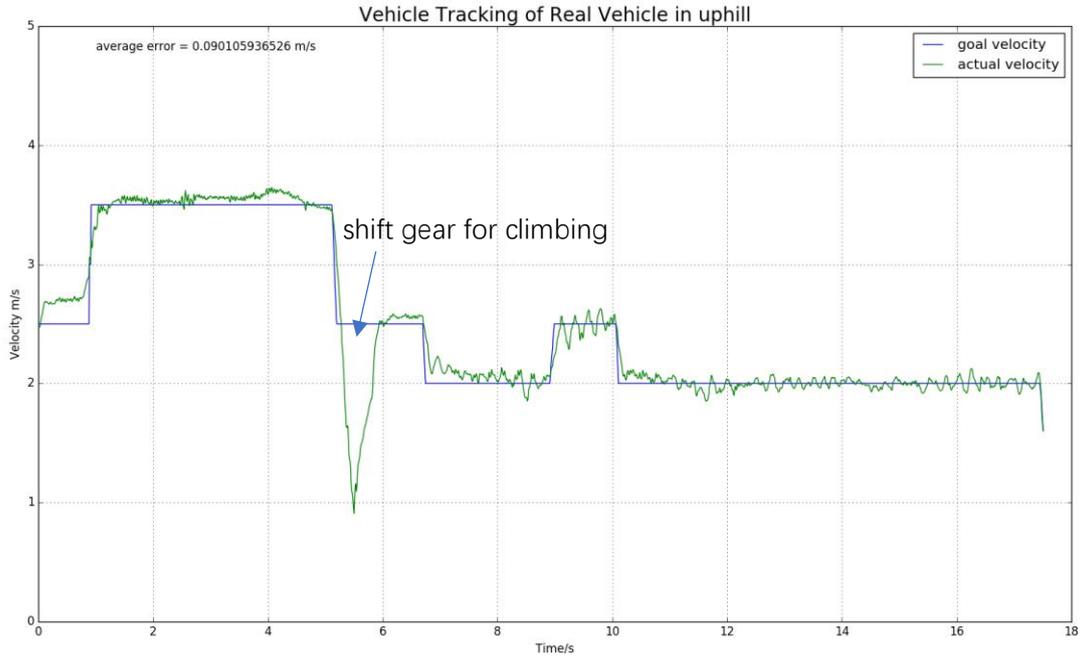


Figure 4.15 Real Vehicle Test Results on Uphill

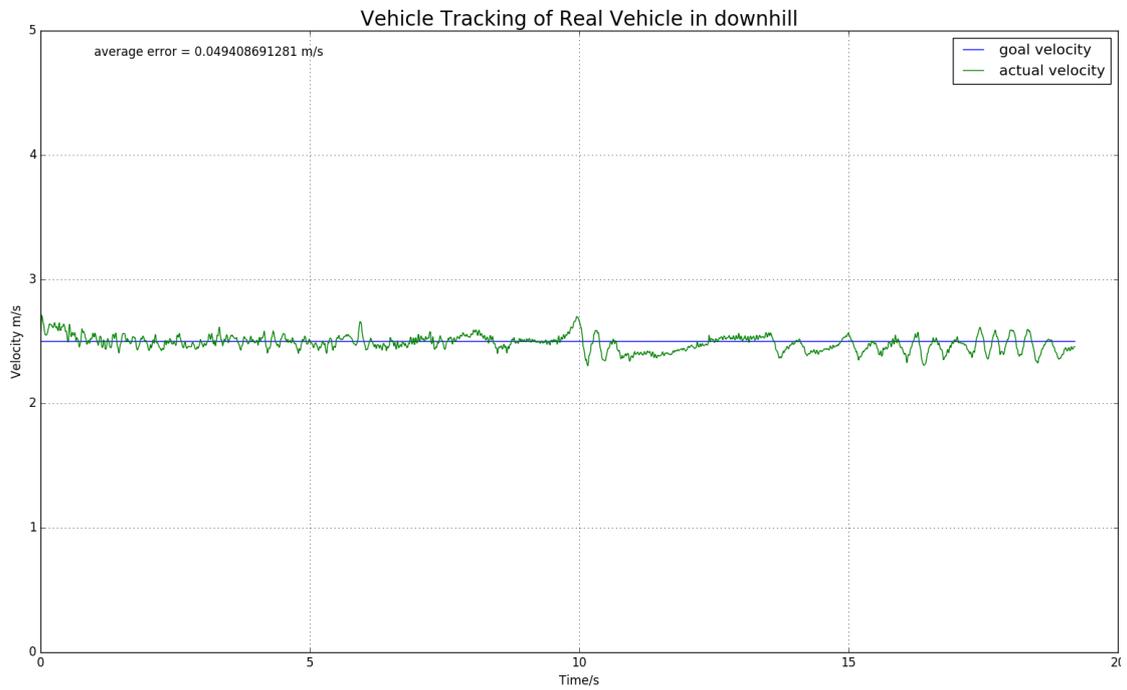


Figure 4.16 Real Vehicle Test Results on Downhill

5. Lateral Controller Design

The purpose of lateral control is to calculate an appropriate front wheel steering angle δ to control the vehicle to travel accurately along the trajectory output by the planning module. It is usually assumed that the longitudinal vehicle speed is constant and then the lateral motion controller is designed so that the lateral displacement error and heading angle error between the actual position of vehicle and the desired path gradually converge to zero.

Lateral control methods can be divided into:

- control algorithm based on geometric model
- control algorithm based on kinematic model
- control algorithm based on dynamic model

5.1 Control Methods Based on Geometric Model

The geometric model refers to the geometric relationship of the automated vehicle when steering, including the relative pose relationship between the vehicle and the reference path and the Ackerman steering geometric model.

The common algorithms for path tracking control of driverless vehicles based on geometric models include pure pursuit algorithm, and Stanley algorithm.

5.1.1 Ackerman steering model

The Ackerman steering model which is shown in Figure 5.1.

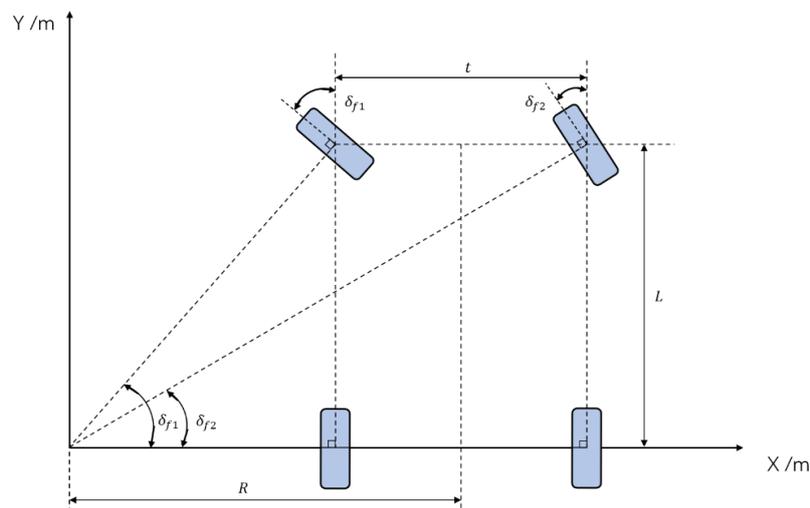


Figure 5.1 Ackerman Steering Model

Ackerman steering model describes the behavior of vehicle during turning.

The relation between steering radius and steering angle of front wheel are:

$$\begin{cases} \tan\delta_{f1} = \frac{L}{R-\frac{l}{2}} \\ \tan\delta_{f2} = \frac{L}{R+\frac{l}{2}} \end{cases} \quad (5.1)$$

In order to simplify the vehicle motion behavior, an effective method is to use the single-track model shown in Figure 5.2, which assumes that the two front wheels have same steering angle all the time, so that we can use one front wheel and rear-wheel to represent the vehicle, like a bicycle.

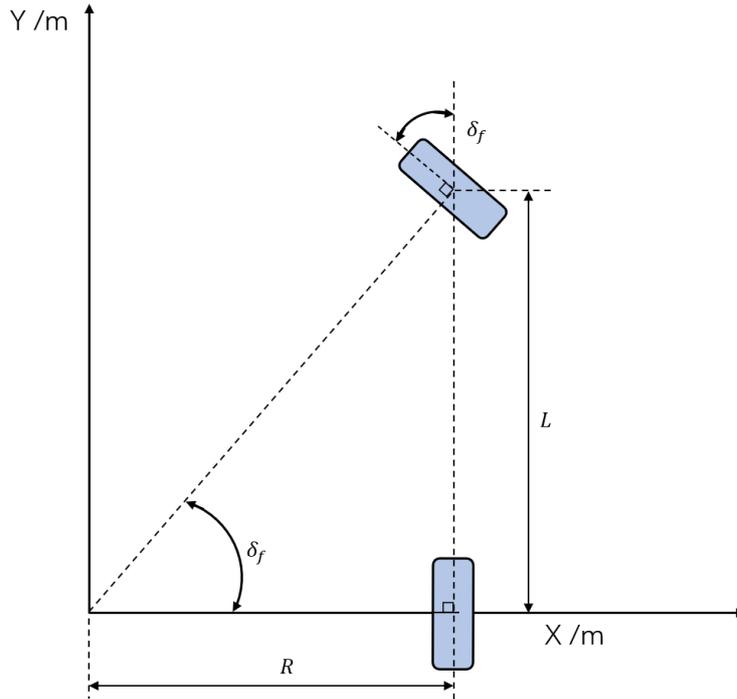


Figure 5.2 Ackerman Steering Single-Track Model

So that Equation (5.1) becomes:

$$\tan\delta_f = \frac{L}{R} \quad (5.2)$$

5.1.2 Pure Pursuit Control

a) Principle

Pure pursuit algorithm [11] is a path tracking control strategy proposed by scholars of Carnegie Mellon University. Figure 5.3 shows a schematic diagram of the geometric relationship of pure pursuit control.

The basic principle is to calculate the steering radius R of the vehicle to make the center point of the rear-axle of the vehicle follow a circular arc to reach the reference path target point $G: (g_x, g_y)$ with a forward-looking distance l_d . And then calculate the steering angle of the front wheels δ_f required for the steering radius R based on the Ackerman steering model (Equation (5.2)).

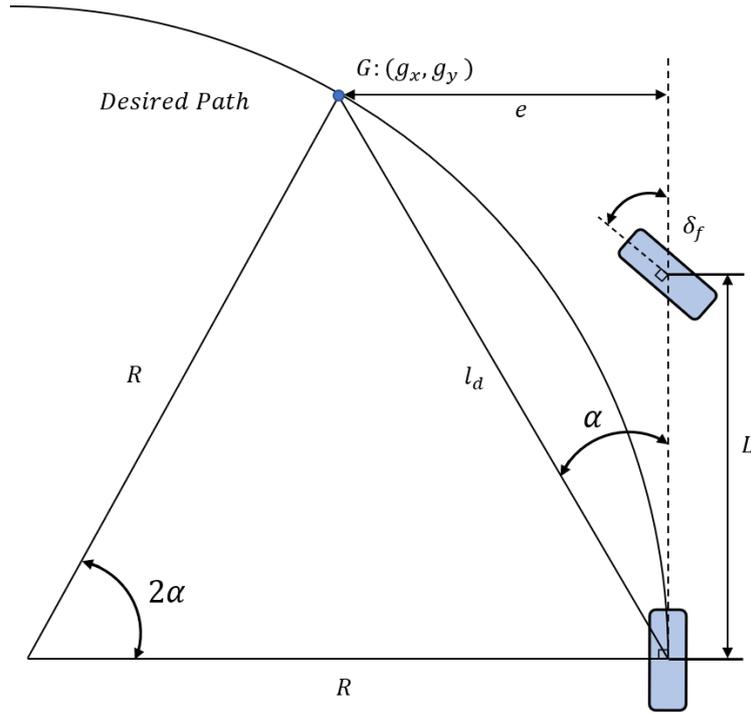


Figure 5.3 Pure Pursuit Control

According to the equation:

$$\frac{l_d}{\sin(2\alpha)} = \frac{R}{\sin\left(\frac{\pi}{2}-\alpha\right)} \quad (5.3)$$

The steering radius R and curvature ρ are calculated by the following equations:

$$R = \frac{l_d}{2\sin\alpha} = \frac{l_d^2}{2e} \quad (5.4)$$

$$\rho = \frac{1}{R} = \frac{2}{l_d^2} e \quad (5.5)$$

Where: e is the lateral displacement error between goal point and actual point.

Combined with Ackerman steering model (Equation (5.2)), the steering angle of front wheel required by the radius is:

$$\tan\delta_f = \frac{2L\sin\alpha}{l_d} = \frac{2L}{l_d^2} e \quad (5.6)$$

where L is wheelbase of the vehicle.

From Equation (5.6), pure pursuit can be regarded as a proportional control method [10] whose input is the lateral displacement error between goal and actual point e , and proportional gain is $\frac{2L}{l_d^2}$.

Therefore, the effect of algorithm depends on the choice of l_d .

Small l_d means a large proportional gain which causes overshoot and oscillation during turning, and on the contrary, large l_d will lead to the low response speed.

So that it is recommended to set the forward-distance l_d related to the velocity:

$$l_d = V * T \quad (5.7)$$

Where: V is the current speed of vehicle and T is a predetermined time.

b) Pros and Cons

Pros:

- It is easy to implement.
- It has good robust for the disturbance in road curvature.

Cons:

- The effect of algorithm seriously depends on the choice of l_d , but it is actually difficult to get the optimal value.
- It doesn't take into account the dynamic properties of vehicle and steering actuator, so that it is only applied in the case of low velocity and small lateral acceleration.

c) Simulation Results

Figure 5.4 and 5.6 is the result of path tracking and heading angle tracking of pure pursuit controller with velocity equal to 10m/s. The error of displacement and heading angle are plotted in Figure 5.4 and 5.6.

The average lateral displacement error is 4.2737cm, the average heading error is 0.2406deg which can be ignored.

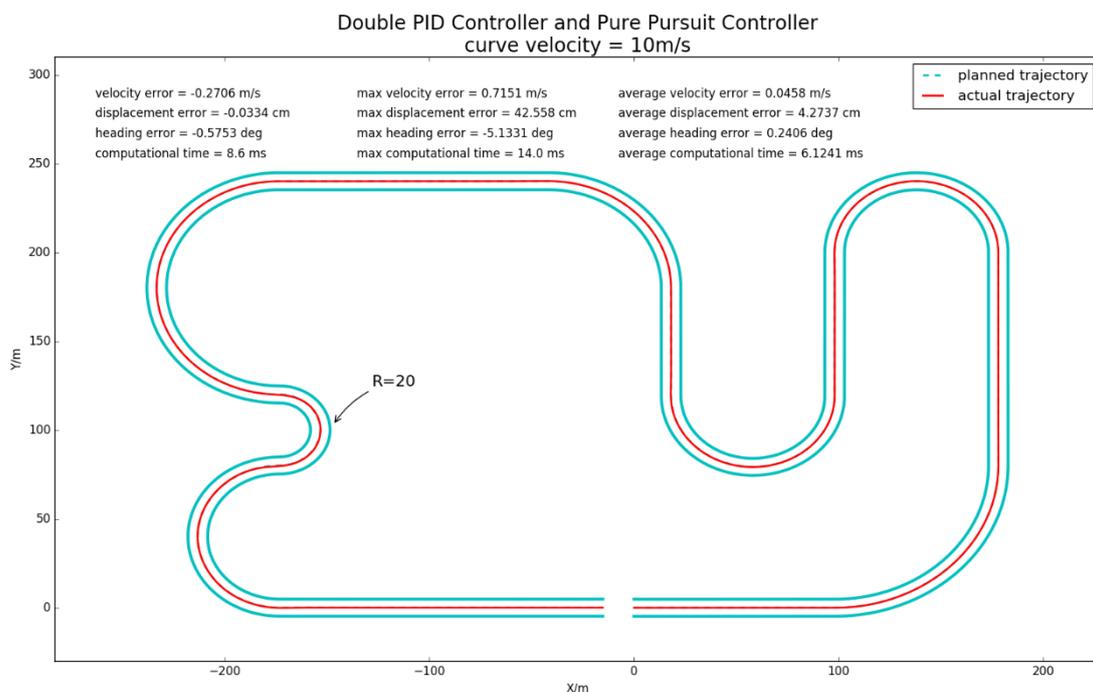


Figure 5.4 Path Tracking of Pure Pursuit at V=10 m/s

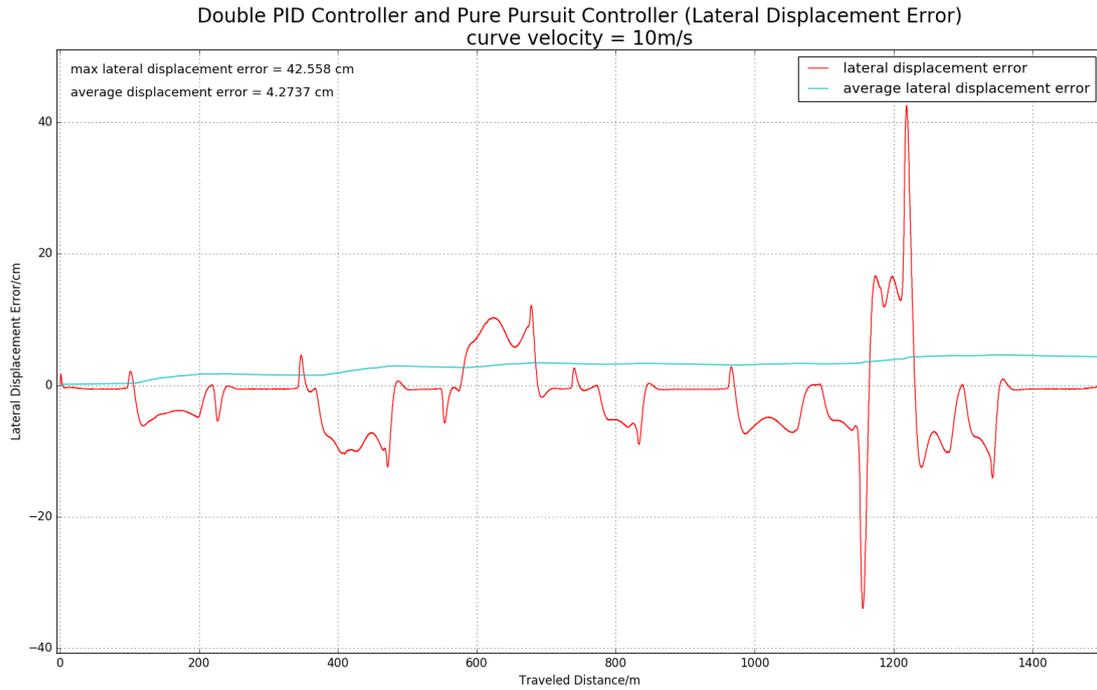


Figure 5.5 Lateral Displacement Error of Pure Pursuit at V=10 m/s

From the error plot Figure 5.5 and 5.7, the lateral displacement error gets large when turning, and converge in straight road.

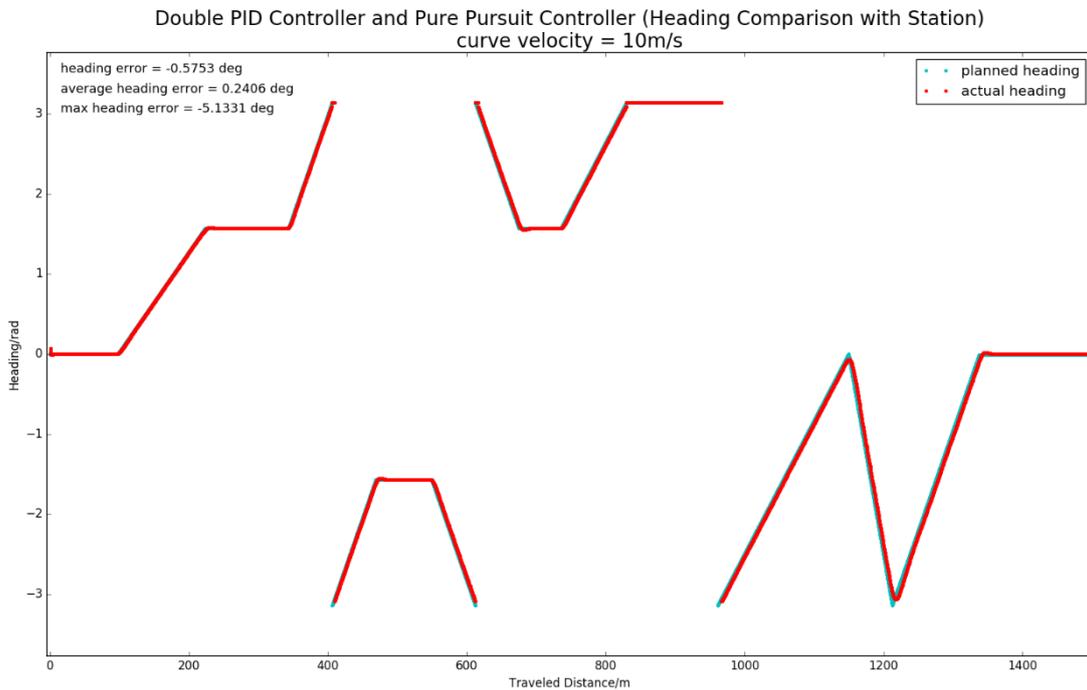


Figure 5.6 Heading Tracking of Pure Pursuit at V=10 m/s

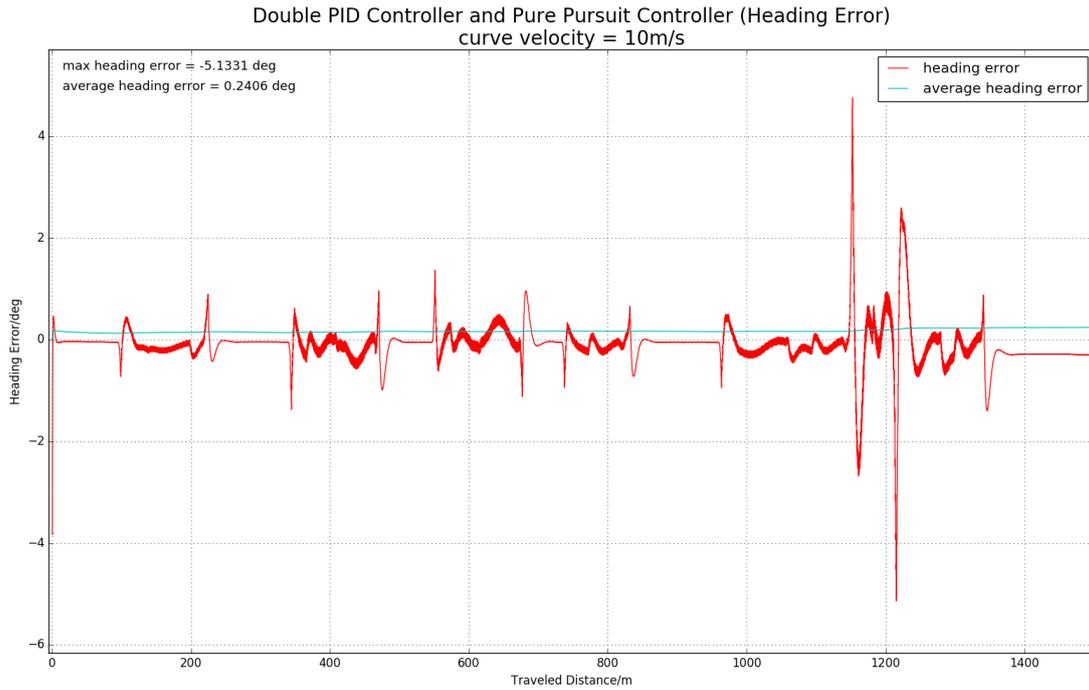


Figure 5.7 Heading Error of Pure Pursuit at V=10 m/s

Figure 5.8 and 5.10 is the result of path tracking and heading angle tracking of pure pursuit controller with velocity equal to 15m/s. The error of displacement and heading angle are plotted in Figure 5.9 and 5.11.

The average displacement error and heading error is 270.145cm and 4.1659deg.

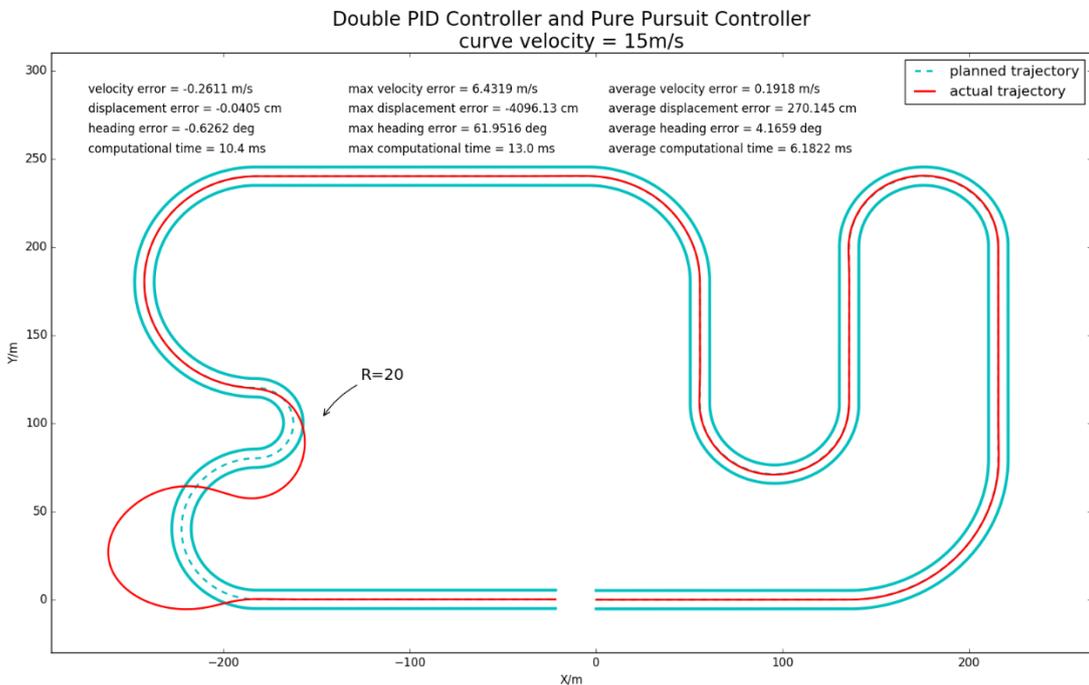


Figure 5.8 Path Tracking of Pure Pursuit at V=15 m/s

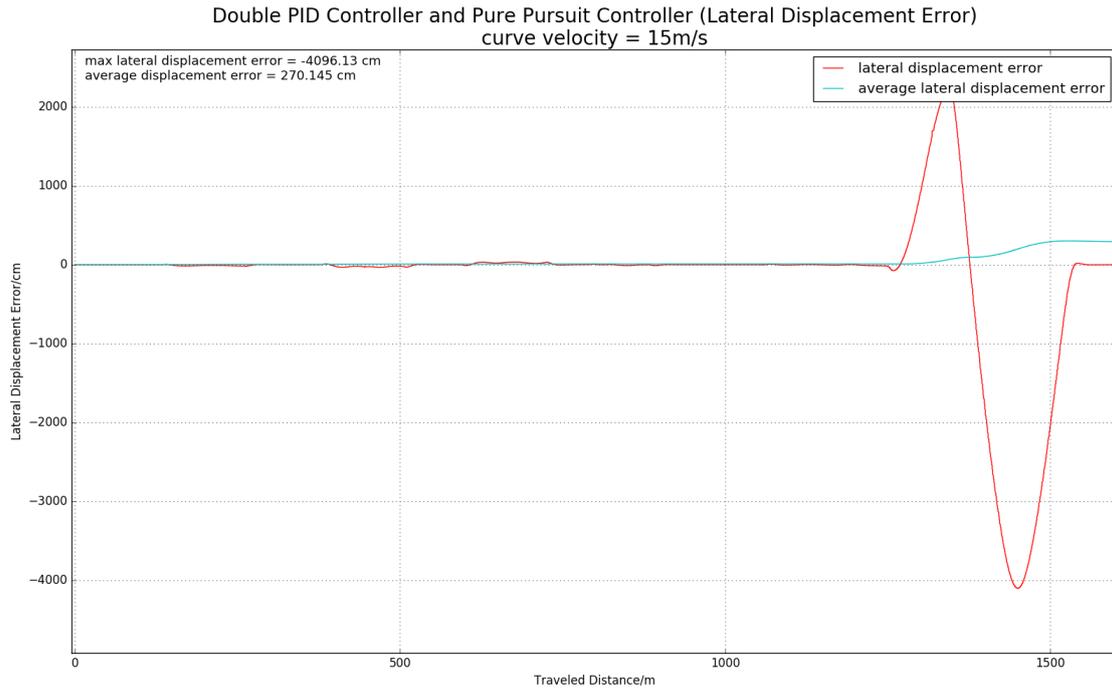


Figure 5.9 Lateral Displacement Error of Pure Pursuit at V=15 m/s

The maximum error occurs during the turning on curve with radius equal to 20 which exceeds the steering limitation of vehicle model. But from the Figure 5.9 and 5.11, the error in other parts is very small.

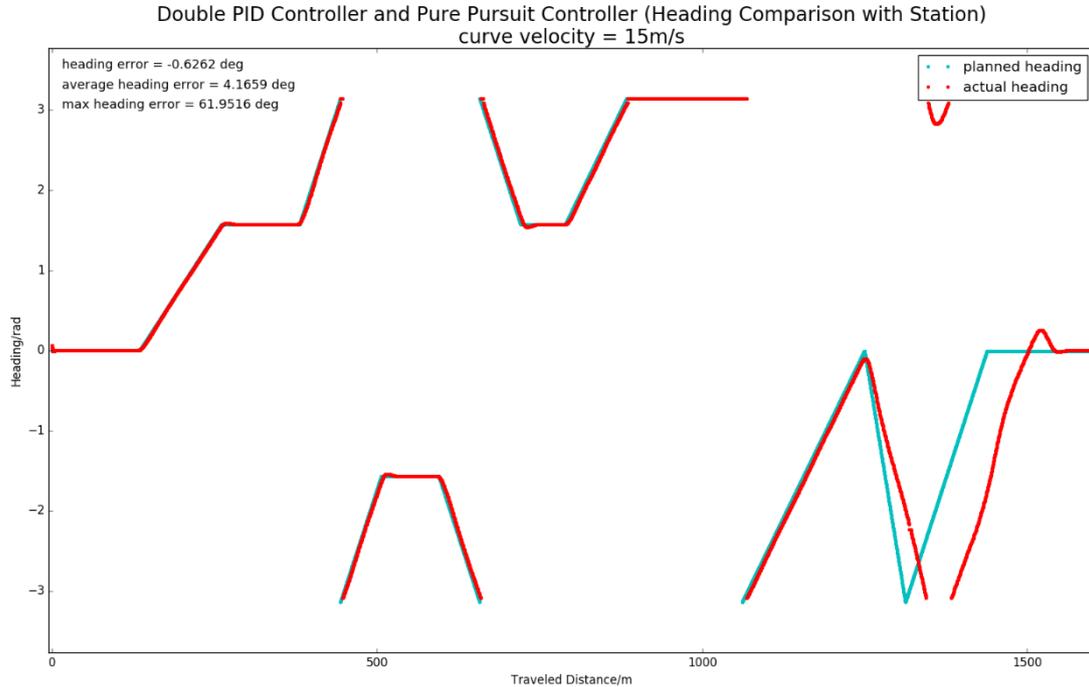
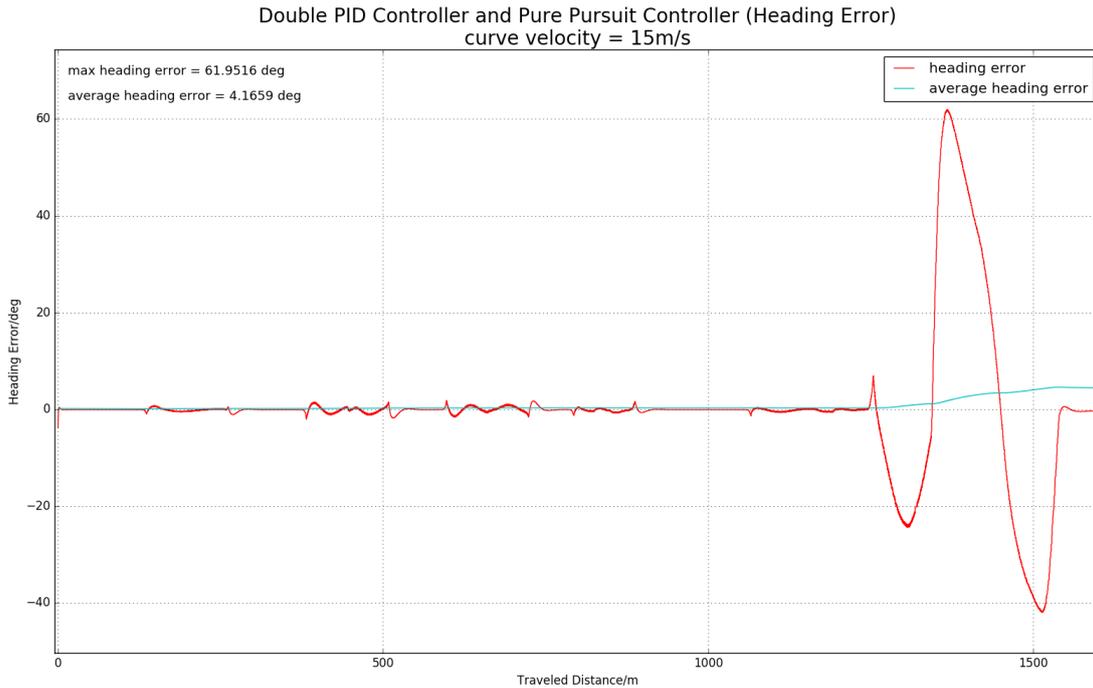


Figure 5.10 Heading Tracking of Pure Pursuit at V=15 m/s

Figure 5.11 Heading Error of Pure Pursuit at $V=15$ m/s

5.1.3 Stanley Control

a) Principle

In 2005, Stanley University Stanley won the United States Defense Advanced Research Projects Agency (DARPA) Desert Challenge using Stanley algorithm.

As shown in Figure 5.12, the Stanley algorithm[13] designs the nonlinear feedback control law according to the lateral displacement error y_e and heading angle error φ_e from the front axle center control point to the nearest desired path target point $G: (g_x, g_y)$. The nonlinear controller can ensure that the lateral displacement error y_e exponent converges to 0.

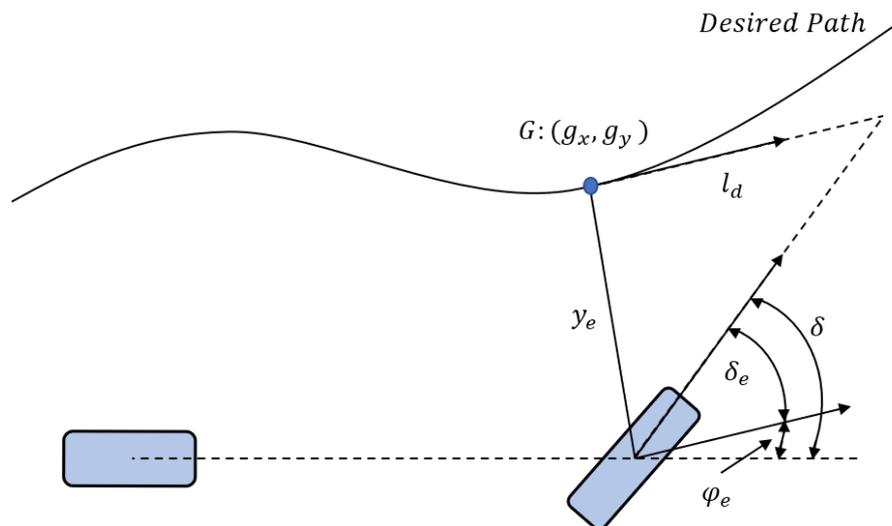


Figure 5.12 Stanley Control

Considering the symbol of Figure 5.12 it is possible to write the following equations:

$$\delta = \varphi_e + \delta_e \quad (5.8)$$

$$\delta_e = \text{atctan} \frac{y_e}{l_d} \quad (5.9)$$

The effect of algorithm also depends on the choice of l_d , which is set to:

$$l_d = V * T \quad (5.10)$$

Therefore:

$$\delta = \varphi_e + \arctan \frac{y_e}{VT} \quad (5.11)$$

b) Pros and Cons

Pros:

- Compared to the pure pursuit algorithm, Stanley algorithm is more suitable for the relatively higher speed driving.
- Stanley algorithm responses faster than pure pursuit algorithm because it compares the center point of front axle to the reference goal point, and most vehicles are front-wheel steering.
- There is only one parameter, which is easy to adjust.

Cons:

- Stanley algorithm has higher requirement for the smooth of reference trajectory.

c) Simulation Result

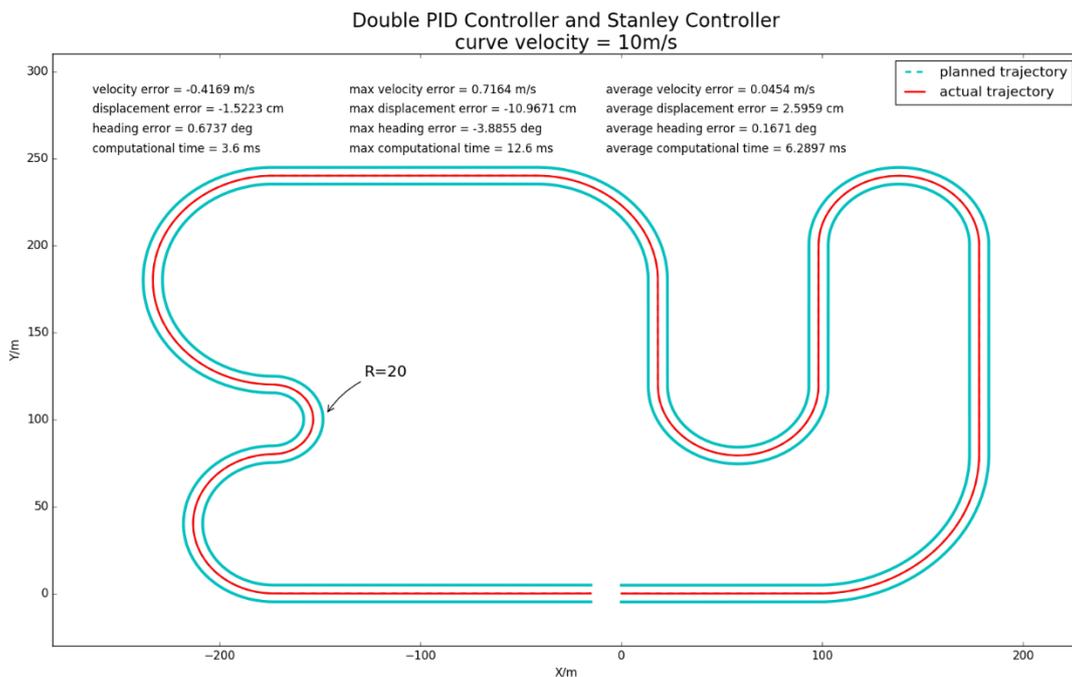


Figure 5.13 Path Tracking of Stanley at V=10 m/s

Figure 5.13 and 5.15 is the result of path tracking and heading angle tracking of Stanley controller with velocity equal to 10m/s. The error of displacement and heading angle are plotted in Figure 5.14 and 5.16.

The average lateral displacement error is 2.59cm, the average heading error is 0.1671deg which is better than the result of pure pursuit controller.

Besides, its maximum error of lateral displacement(-10.96cm) and heading angle(-3.88deg) are also smaller than that of pure pursuit controller.

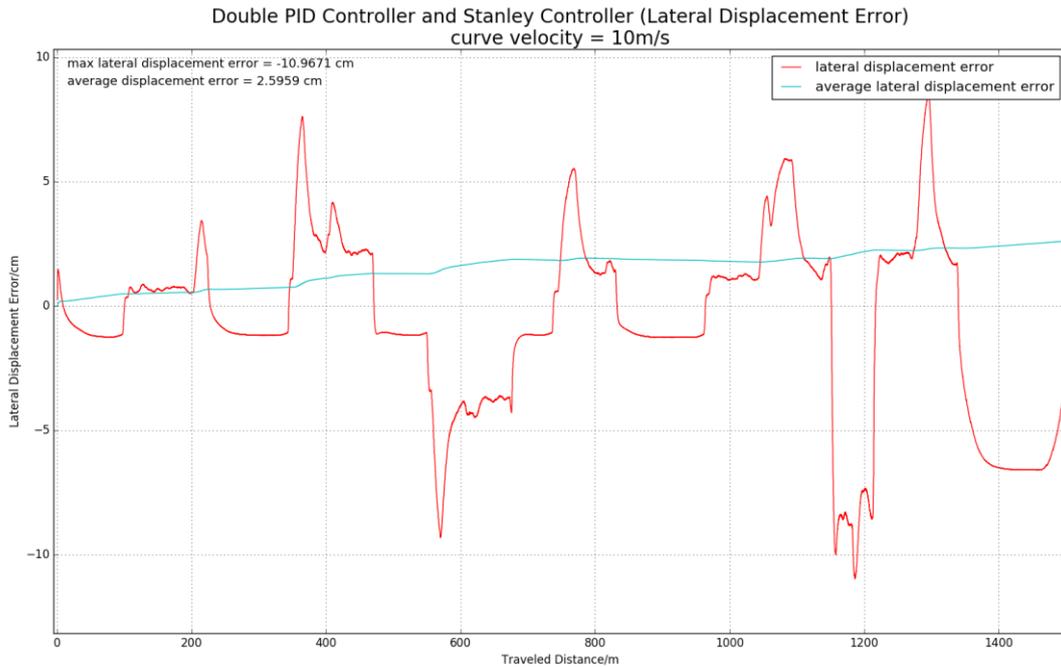


Figure 5.14 Lateral Displacement Error of Stanley at V=10 m/s

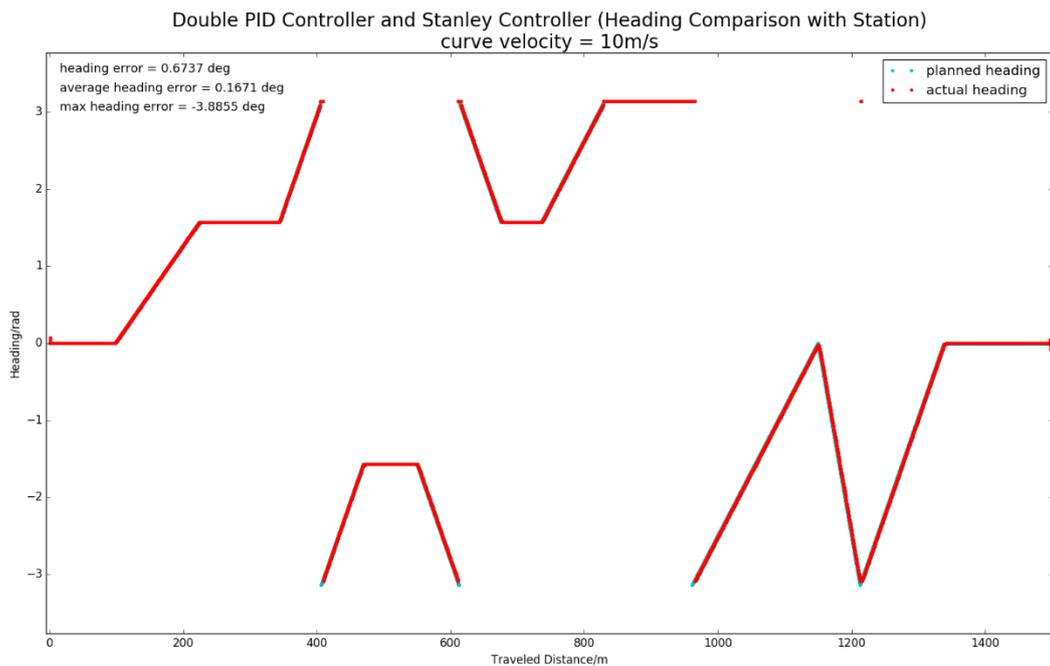


Figure 5.15 Heading Tracking of Stanley at V=10 m/s

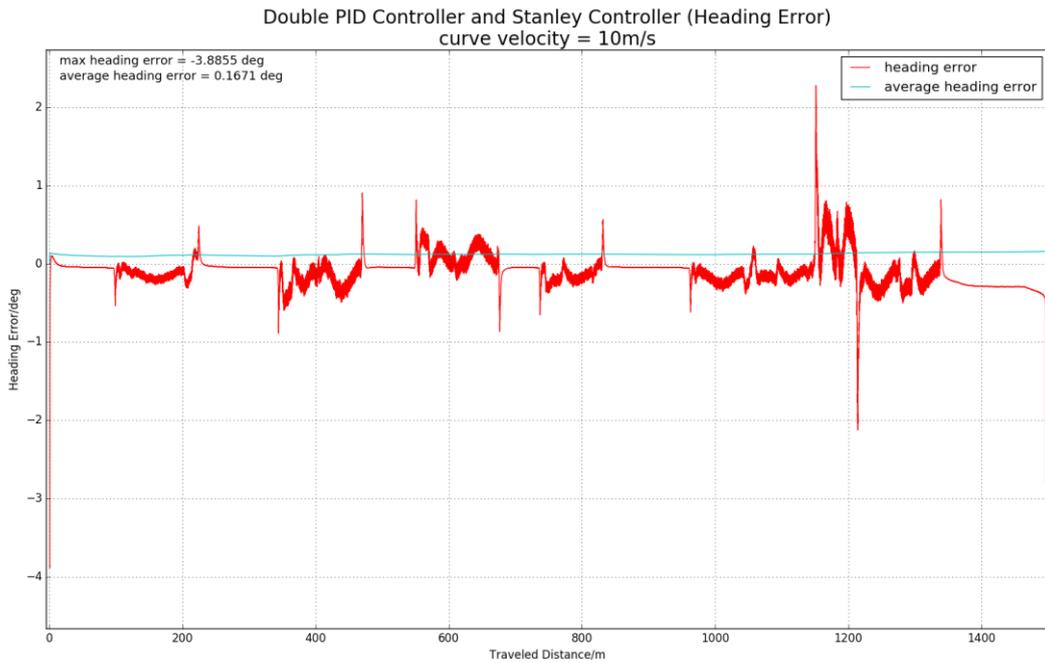


Figure 5.16 Heading Error of Stanley at V=10 m/s

Figure 5.17 and 5.19 is the result of path tracking and heading angle tracking of Stanley controller with velocity equal to 15m/s. The error of displacement and heading angle are plotted in Figure 5.18 and 5.20.

The average lateral displacement error is 79.99cm, the average heading error is 1.85deg which is much less than the result of pure pursuit controller in same condition.

The maximum error also occurs during the turning on curve with radius equal to 20. Compared to the performance of pure pursuit controller, Stanley controller has shorter response time and smaller error on all the path.

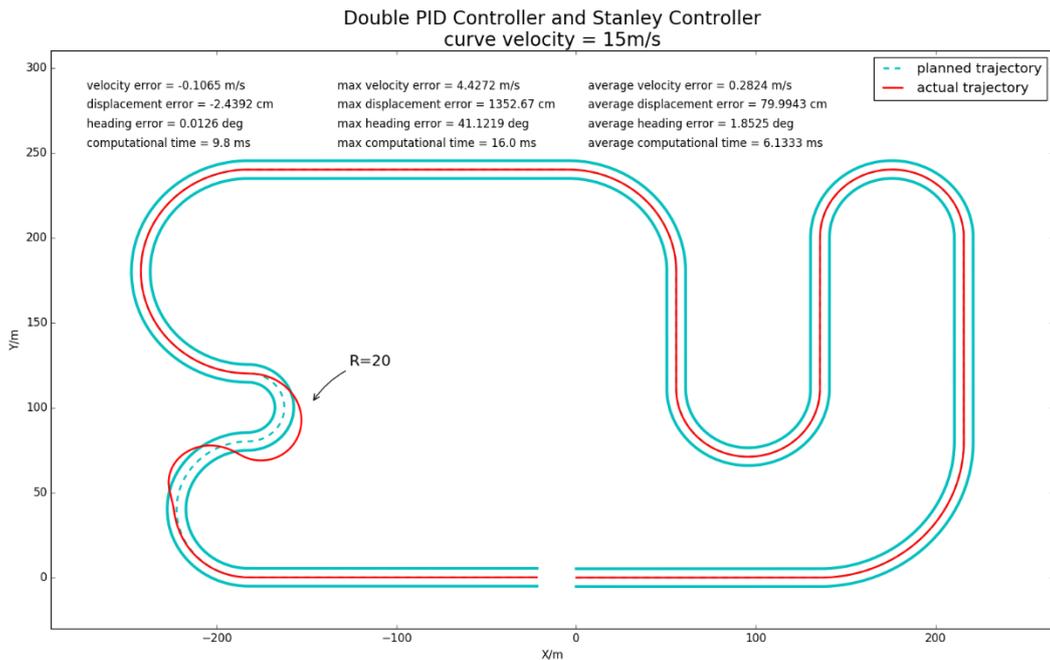


Figure 5.17 Path Tracking of Stanley at V=15 m/s

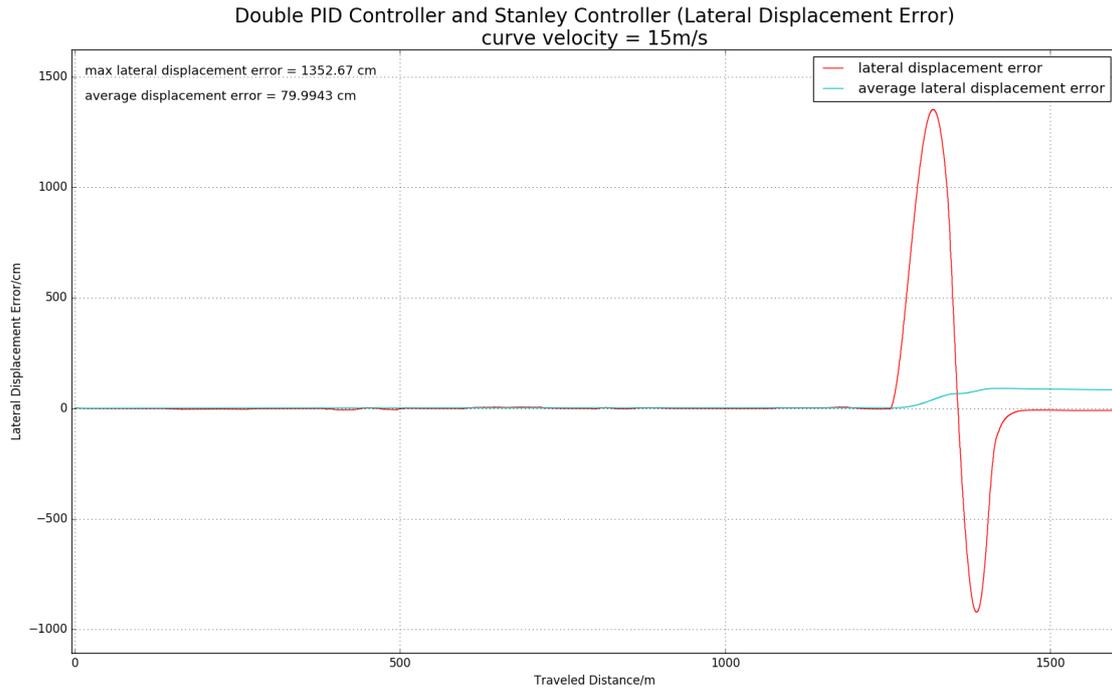


Figure 5.18 Lateral Displacement Error of Stanley at V=15 m/s

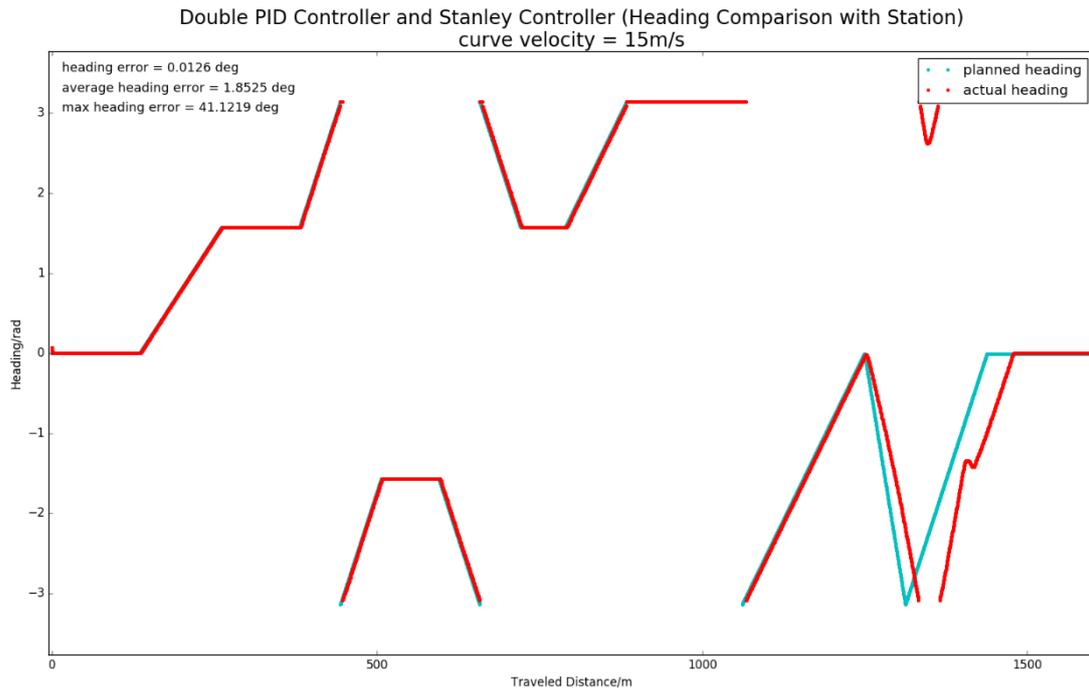
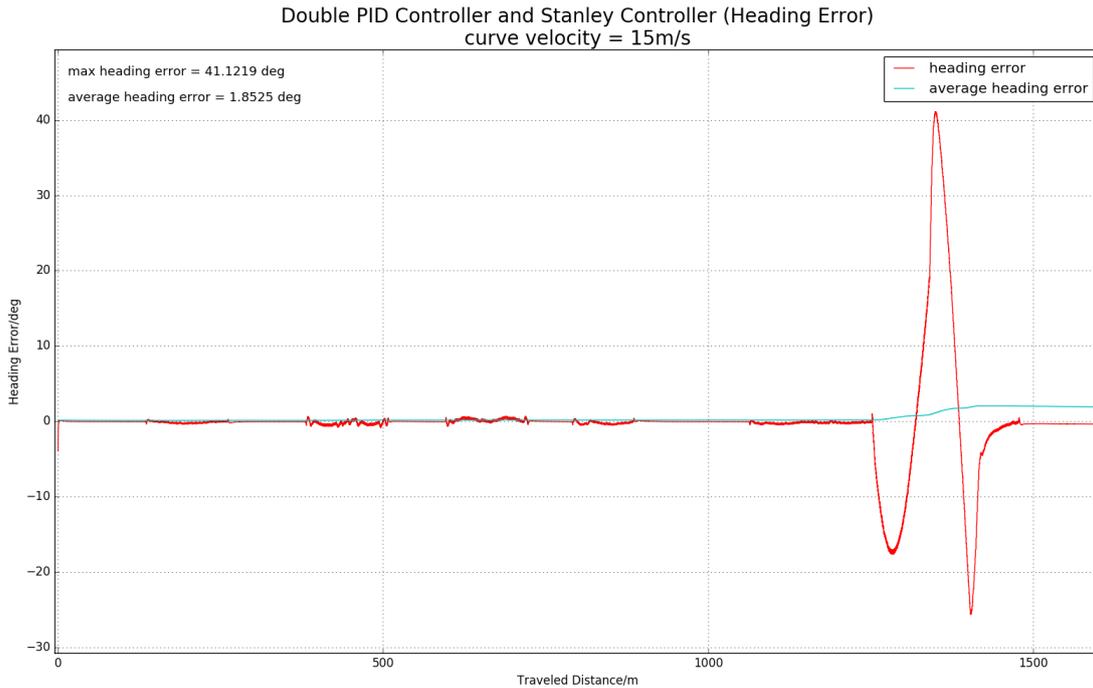


Figure 5.19 Heading Tracking of Stanley at V=15 m/s

Figure 5.20 Heading Error of Stanley at $V=15$ m/s

5.2 Vehicle Model

Different from the geometric model which depends on the vehicle pose and Ackerman steering model, the vehicle kinematic model further considers the motion of vehicle, and the dynamic model takes into account the physical properties of the vehicle itself (such as mass, etc.) and the forces acting on the vehicle.

5.2.1 Vehicle Kinematic Modeling

When studying vehicle motion control problems based on kinematics model [32], it is usually assumed that the vehicle does not have sideslip, in another word, slip angle of the center of gravity is zero.

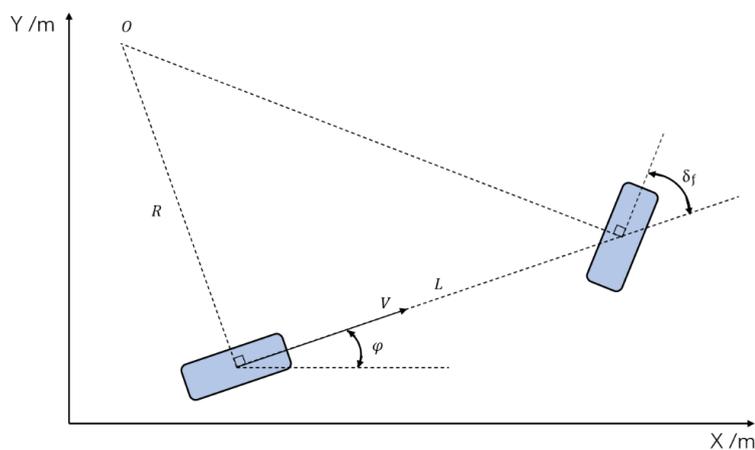


Figure 5.21 Vehicle Kinematic Model

Figure 5.21 shows the schematic diagram of the kinematic model.

The path-tracking control method based on the kinematic model does not consider the dynamic characteristics of the vehicle, and is simpler and more practical.

It usually has a better control effect under the conditions of medium and low speed and small curvature than the algorithms based on geometric model.

However, the actual vehicle has obvious sideslip and large lateral acceleration when turning under high speed, so the method based on kinematic model is not suitable for the conditions.

The derivation of the vehicle kinematic model is shown followed.

The velocity components along X and Y axis (Global Frame) are:

$$\begin{cases} \dot{X} = V \cos \varphi \\ \dot{Y} = V \sin \varphi \end{cases} \quad (5.12)$$

The derivative of heading angle are:

$$\begin{cases} \dot{\varphi} = \frac{V}{R} \\ \tan \delta_f = \frac{L}{R} \end{cases} \quad (5.13)$$

Therefore, the differential equations of nonlinear vehicle kinematic model are:

$$\begin{cases} f_1 = \dot{X} = V \cos \varphi \\ f_2 = \dot{Y} = V \sin \varphi \\ f_3 = \dot{\varphi} = \frac{V \tan \delta_f}{l} \end{cases} \quad (5.14)$$

5.2.2 Vehicle Dynamic Modeling

a) Introduction

Because the path tracking control method based on geometric or kinematic model ignores the dynamic characteristics of the vehicle, the applicable working conditions have limitations.

Therefore, in order to obtain a more accurate tracking control effect, especially in high-speed and large curvature conditions, it is necessary to consider the dynamic properties of the vehicle when designing the path tracking control algorithm.

Autonomous driving vehicles are highly non-linear and complex dynamic systems with strong coupling characteristics. Although expanding the model dimensions can improve the accuracy of the model, it will also increase the difficulty of modeling and also bring huge challenges to the rapid computation. When modeling, it is necessary to make compromise between complexity and accuracy of the model. Therefore, the vehicle dynamics model with 3 degrees of freedom is commonly used [33].

The dynamic model [34] is established according to the external force and torque acting on the vehicle which mainly come from tires and air resistance. In the mining

area, the vehicle speed is relatively slow, so the influence of air resistance is so small that can be ignored.

Therefore, under the following idealized assumptions, a 3DOF dynamic model describing vehicle motion is established:

- Assuming that the road is flat, so the influence caused by the vertical displacement of the road is ignored.
- Assuming the vehicle is rigid which ignores the influence of elastic elements like suspensions.
- Assuming that the input signal directly acts on the front wheels of the vehicle, ignoring the dynamic characteristics of the steering mechanism.
- Ignore the influence of air resistance.
- Assuming that the front wheel side slip angle is small ($-4^\circ \sim 4^\circ$), the longitudinal and lateral tire force model are linear. [35]
- Ignoring the coupling characteristics of the lateral and longitudinal motion and load transfer of vehicle.

b) Establishment of vehicle dynamic model

Figure 5.22 shows the schematic diagram of the dynamic model.

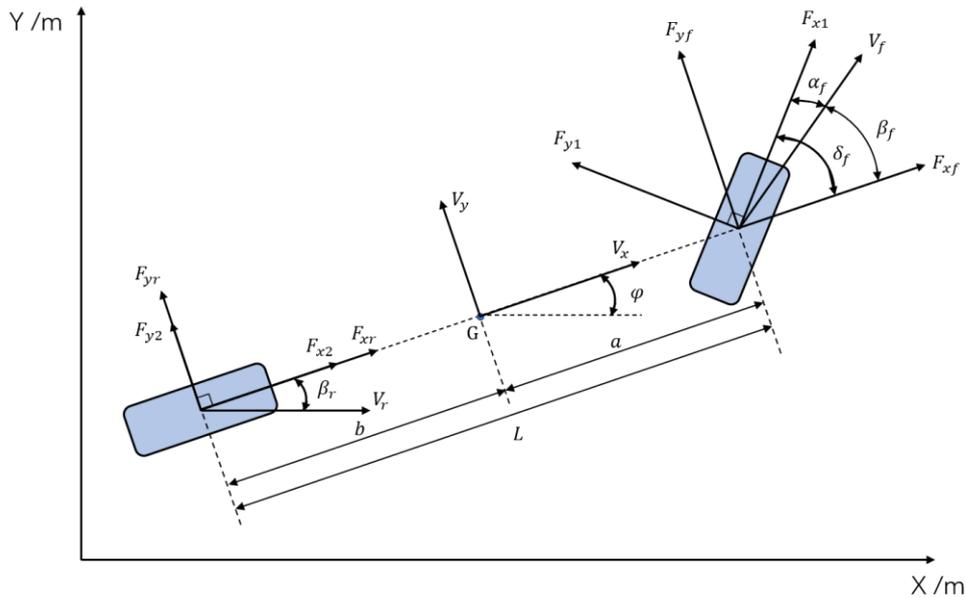


Figure 5.22 Vehicle Dynamic Model

The forces and moment acting on vehicle are:

$$\begin{cases} ma_x = 2F_{xf} + 2F_{xr} \\ ma_y = 2F_{yf} + 2F_{yr} \\ I\ddot{\varphi} = 2aF_{yf} - 2bF_{yr} \end{cases} \quad (5.15)$$

When vehicle is turning, the acceleration can be divided into the linear and angular acceleration around center of gravity G.

$$\begin{cases} a_x = \ddot{x} - V_y \dot{\phi} \\ a_y = \ddot{y} + V_x \dot{\phi} \end{cases} \quad (5.16)$$

All the force is provided by tire are:

$$\begin{cases} F_{xf} = F_{x1} \cos \delta_f - F_{y1} \sin \delta_f \\ F_{yf} = F_{x1} \sin \delta_f + F_{y1} \cos \delta_f \\ F_{xr} = F_{x2} \\ F_{yr} = F_{y2} \end{cases} \quad (5.17)$$

Where F_{xi} and F_{yi} are the longitudinal and lateral force of tire.

Due to the assumption of small steering angle δ_f , the force can be simplified:

$$\begin{cases} F_{xf} = F_{x1} \\ F_{yf} = F_{y1} \\ F_{xr} = F_{x2} \\ F_{yr} = F_{y2} \end{cases} \quad (5.18)$$

The direction of velocity of front and rear-wheel are:

$$\begin{cases} \tan \beta_f = \frac{V_{yf}}{V_{xf}} = \frac{V_y + a\dot{\phi}}{V_x} \\ \tan \beta_r = \frac{V_{yr}}{V_{xr}} = \frac{V_y - b\dot{\phi}}{V_x} \end{cases} \quad (5.19)$$

Because β_f and β_r are small, so the Equation(5.19) can be transformed into:

$$\begin{cases} \beta_f = \frac{V_y + a\dot{\phi}}{V_x} \\ \beta_r = \frac{V_y - b\dot{\phi}}{V_x} \end{cases} \quad (5.20)$$

The lateral forces are:

$$\begin{cases} F_{y1} = C_{\alpha f} \alpha_f = C_{\alpha f} (\delta_f - \beta_f) = C_{\alpha f} \left(\delta_f - \frac{V_y + a\dot{\phi}}{V_x} \right) \\ F_{y2} = C_{\alpha r} \alpha_r = C_{\alpha r} (\delta_r - \beta_r) = -C_{\alpha r} \beta_r = C_{\alpha r} \frac{b\dot{\phi} - V_y}{V_x} \end{cases} \quad (5.21)$$

The longitudinal forces are:

$$\begin{cases} F_{x1} = C_{\sigma f} \sigma_f \\ F_{x2} = C_{\sigma r} \sigma_r \end{cases} \quad (5.22)$$

Therefore:

$$\begin{cases} F_{xf} = F_{x1} = C_{\sigma f} \sigma_f \\ F_{xr} = F_{x2} = C_{\sigma r} \sigma_r \\ F_{yf} = F_{y1} = C_{\alpha f} \left(\delta_f - \frac{V_y + a\dot{\phi}}{V_x} \right) \\ F_{yr} = F_{y2} = C_{\alpha r} \frac{b\dot{\phi} - V_y}{V_x} \end{cases} \quad (5.23)$$

Combined with Equation (5.15) and (5.16), the final nonlinear dynamic model is:

$$\begin{cases} \dot{X} = \dot{x} \cos \varphi - \dot{y} \sin \varphi \\ \dot{Y} = \dot{x} \sin \varphi + \dot{y} \cos \varphi \\ \dot{\phi} = \dot{\phi} \\ \ddot{\phi} = \frac{2C_{\alpha r} * b - 2C_{\alpha f} * a}{I_z \dot{x}} \dot{y} - \frac{2C_{\alpha r} * b^2 + 2C_{\alpha f} * a^2}{I_z \dot{x}} \dot{\phi} + \frac{2C_{\alpha f} a}{I_z} \delta_f \\ \ddot{x} = \dot{y} \dot{\phi} + \frac{2(C_{s f} s_f + C_{s r} s_r) + 2C_{\alpha f} \delta_f^2}{m} - 2C_{\alpha f} \delta_f \frac{\dot{y}}{m \dot{x}} - 2C_{\alpha f} \delta_f \frac{a \dot{\phi}}{m \dot{x}} = acc \\ \ddot{y} = -\frac{2C_{\alpha r} + 2C_{\alpha f}}{m \dot{x}} \dot{y} - \left(\dot{x} + \frac{-2C_{\alpha r} * b + 2C_{\alpha f} * a}{m \dot{x}} \right) \dot{\phi} + \frac{2C_{\alpha f}}{m} \delta_f \end{cases} \quad (5.24)$$

Equation (5.24) is built based on the global frame, but the model based on the vehicle coordinates frame system is more common:

$$\begin{cases} \dot{x} = \dot{x} \\ \dot{y} = \dot{y} \\ \dot{\phi} = \dot{\phi} \\ \ddot{\phi} = \frac{2C_{\alpha r} * b - 2C_{\alpha f} * a}{I_z \dot{x}} \dot{y} - \frac{2C_{\alpha r} * b^2 + 2C_{\alpha f} * a^2}{I_z \dot{x}} \dot{\phi} + \frac{2C_{\alpha f} a}{I_z} \delta_f \\ \ddot{x} = \dot{y} \dot{\phi} + \frac{2(C_{s f} s_f + C_{s r} s_r) + 2C_{\alpha f} \delta_f^2}{m} - 2C_{\alpha f} \delta_f \frac{\dot{y}}{m \dot{x}} - 2C_{\alpha f} \delta_f \frac{a \dot{\phi}}{m \dot{x}} = acc \\ \ddot{y} = -\frac{2C_{\alpha r} + 2C_{\alpha f}}{m \dot{x}} \dot{y} - \left(\dot{x} + \frac{-2C_{\alpha r} * b + 2C_{\alpha f} * a}{m \dot{x}} \right) \dot{\phi} + \frac{2C_{\alpha f}}{m} \delta_f \end{cases} \quad (5.25)$$

5.2.3 Model Linearization and Discretization

a) Linearization

Whether it is kinematic or dynamic modeling, the resulting model is nonlinear, although it can also be used for control, but the computational consumption will be very large, so in general, we will linearize the nonlinear system to get a linear system in the form of state space.

There are many linearization methods, which can be roughly divided into approximate linearization and precise linearization. The approximate linearization method is simple and has a wide range of applications. The disadvantage is that it cannot be applied to occasions with very high control accuracy requirements. Precise linearization is often not universal and requires analysis for one specific systems.

If there is a reference system:

$$\dot{\xi}_r = f(\xi_r, u_r) \quad (5.26)$$

The small deviation linearization method can be used.

The state and control quantity of the vehicle at any time satisfy the following relationship:

$$\dot{\xi} = f(\xi, u) \quad (5.27)$$

Perform Taylor's first-order expansion at any point (ξ_r, u_r) to get:

$$\dot{\xi} = f(\xi_r, u_r) + J_f(\xi)(\xi - \xi_r) + J_f(u)(u - u_r) \quad (5.28)$$

Where $J_f(\xi)$ and $J_f(u)$ are the Jacobian matrix of $f(\xi_r, u_r)$ with respect to ξ and u .

Equation (5.28) subtract Equation (5.26):

$$\dot{\xi} - \dot{\xi}_r = J_f(\xi)(\xi - \xi_r) + J_f(u)(u - u_r) \quad (5.29)$$

So that the linear system is:

$$\dot{\tilde{\xi}} = A(t)\tilde{\xi} + B(t)\tilde{u} \quad (5.30)$$

Where:

$$\begin{cases} \tilde{\xi} = \xi - \xi_r \\ \tilde{u} = u - u_r \\ A(t) = J_f(\xi) \\ B(t) = J_f(u) \end{cases} \quad (5.31)$$

b) Discretization

The model obtained above is continuous, which cannot be used to control the system by a digital controller. The model must be discrete, so the above continuous system needs to be discretized.

The approximate discretization method of the continuous state space system is applied, which is suitable for the case where the sampling period is small and the accuracy of the discretization is not so high.

Continuous system is:

$$\dot{\xi} = A\xi + Bu \quad (5.32)$$

$$\dot{\xi}(kT) = \lim_{T \rightarrow 0} \frac{\xi((k+1)T) - \xi(kT)}{T} \quad (5.33)$$

When $T \rightarrow 0$:

$$\dot{\xi}(kT) \approx \frac{\xi((k+1)T) - \xi(kT)}{T} \quad (5.34)$$

So that:

$$\frac{\xi((k+1)T) - \xi(kT)}{T} = A\xi(kT) + Bu(kT) \quad (5.35)$$

$$\xi((k+1)T) = (I + AT)\xi(kT) + BTu(kT) \quad (5.36)$$

Finally:

$$\begin{cases} \xi((k+1)T) = A'\xi(kT) + B'u(kT) \\ A' = I + AT \\ B' = BT \end{cases} \quad (5.37)$$

c) Linear vehicle kinematic model

The nonlinear vehicle dynamic model is:

$$\begin{cases} f_1 = \dot{X} = V\cos\varphi \\ f_2 = \dot{Y} = V\sin\varphi \\ f_3 = \dot{\varphi} = \frac{V\tan\delta_f}{l} \end{cases} \quad (5.38)$$

The state is:

$$\xi = [X \quad Y \quad \varphi]^T \quad (5.39)$$

The control quantity is:

$$u = \delta_f \quad (5.40)$$

The state space matrices are:

$$A(t) = J_f(\xi) = \begin{bmatrix} \frac{\partial f_1}{\partial X} & \frac{\partial f_1}{\partial Y} & \frac{\partial f_1}{\partial \varphi} \\ \frac{\partial f_2}{\partial X} & \frac{\partial f_2}{\partial Y} & \frac{\partial f_2}{\partial \varphi} \\ \frac{\partial f_3}{\partial X} & \frac{\partial f_3}{\partial Y} & \frac{\partial f_3}{\partial \varphi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & -V_r \sin\varphi_r \\ 0 & 0 & V_r \cos\varphi_r \\ 0 & 0 & 0 \end{bmatrix}$$

$$B(t) = J_f(u) = \begin{bmatrix} \frac{\partial f_1}{\partial \delta_f} \\ \frac{\partial f_2}{\partial \delta_f} \\ \frac{\partial f_3}{\partial \delta_f} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{V_r}{l \cos^2(\delta_{fr})} \end{bmatrix}$$

The linear state space model is:

$$\begin{bmatrix} \dot{X} - \dot{X}_r \\ \dot{Y} - \dot{Y}_r \\ \dot{\varphi} - \dot{\varphi}_r \end{bmatrix} = \begin{bmatrix} 0 & 0 & -V_r \sin\varphi_r \\ 0 & 0 & V_r \cos\varphi_r \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} X - X_r \\ Y - Y_r \\ \varphi - \varphi_r \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{V_r}{l \cos^2(\delta_{fr})} \end{bmatrix} [\delta_f - \delta_{fr}] \quad (5.41)$$

The discrete state space matrices are:

$$A' = I + TA \quad (5.42)$$

$$B' = TB \quad (5.43)$$

The final discrete kinematic model is:

$$\xi((k+1)T) = A'\xi(kT) + B'u(kT) \quad (5.44)$$

$$A' = \begin{bmatrix} 1 & 0 & -TV_r \sin \varphi_r \\ 0 & 1 & TV_r \cos \varphi_r \\ 0 & 0 & 1 \end{bmatrix}$$

$$B' = \begin{bmatrix} 0 \\ 0 \\ V_r \\ l \cos^2(\delta_{fr}) \end{bmatrix}$$

d) Linear vehicle dynamic model

The nonlinear vehicle dynamic model is:

$$\left\{ \begin{array}{l} \dot{x} = \dot{x} \\ \dot{y} = \dot{y} \\ \dot{\varphi} = \dot{\varphi} \\ \ddot{\varphi} = \frac{2C_{\alpha r} * b - 2C_{\alpha f} * a}{I_z \dot{x}} \dot{y} - \frac{2C_{\alpha r} * b^2 + 2C_{\alpha f} * a^2}{I_z \dot{x}} \dot{\varphi} + \frac{2C_{\alpha f} a}{I_z} \delta_f \\ \ddot{x} = \dot{y} \dot{\varphi} + \frac{2(C_{sf} s_f + C_{sr} s_r) + 2C_{\alpha f} \delta_f^2}{m} - 2C_{\alpha f} \delta_f \frac{\dot{y}}{m \dot{x}} - 2C_{\alpha f} \delta_f \frac{a \dot{\varphi}}{m \dot{x}} = acc \\ \ddot{y} = -\frac{2C_{\alpha r} + 2C_{\alpha f}}{m \dot{x}} \dot{y} - \left(\dot{x} + \frac{-2C_{\alpha r} * b + 2C_{\alpha f} * a}{m \dot{x}} \right) \dot{\varphi} + \frac{2C_{\alpha f}}{m} \delta_f \end{array} \right. \quad (5.45)$$

The state is:

$$\xi = [x \quad \dot{x} \quad y \quad \dot{y} \quad \varphi \quad \dot{\varphi}]^T \quad (5.46)$$

The control quantity is:

$$u = [acc \quad \delta_f]^T \quad (5.47)$$

The state space matrices are:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & \frac{\partial f_{\dot{y}}}{\partial \dot{x}} & 0 & -\frac{2C_{\alpha r} + 2C_{\alpha f}}{m \dot{x}} & 0 & -\left(\dot{x} + \frac{-2C_{\alpha r} b + 2C_{\alpha f} a}{m \dot{x}} \right) \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & \frac{\partial f_{\dot{\varphi}}}{\partial \dot{x}} & 0 & \frac{2C_{\alpha r} b - 2C_{\alpha f} a}{I_z \dot{x}} & 0 & -\frac{2C_{\alpha r} b^2 + 2C_{\alpha f} a^2}{I_z \dot{x}} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & \frac{2C_{\alpha f}}{m} \\ 0 & 0 \\ 0 & \frac{2C_{\alpha f} a}{I_z} \end{bmatrix}$$

If the model is only used for lateral control, the state space matrices be simplified into:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha r} + 2C_{\alpha f}}{m\dot{x}} & 0 & -\left(\dot{x} + \frac{-2C_{\alpha r}b + 2C_{\alpha f}a}{m\dot{x}}\right) \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2C_{\alpha r}b - 2C_{\alpha f}a}{I_z\dot{x}} & 0 & -\frac{2C_{\alpha r}b^2 + 2C_{\alpha f}a^2}{I_z\dot{x}} \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}a}{I_z} \end{bmatrix}$$

The final discrete dynamic model is:

$$\begin{bmatrix} \dot{y} - \dot{y}_r \\ \dot{y} - \dot{y}_r \\ \dot{\varphi} - \dot{\varphi}_r \\ \dot{\varphi} - \dot{\varphi}_r \end{bmatrix} = \begin{bmatrix} 1 & T & 0 & 0 \\ 0 & 1 - 2T\frac{C_{\alpha r} + C_{\alpha f}}{m\dot{x}} & 0 & -T\left(\dot{x} + \frac{-2C_{\alpha r}b + 2C_{\alpha f}a}{m\dot{x}}\right) \\ 0 & 0 & 1 & T \\ 0 & 2T\frac{C_{\alpha r}b - C_{\alpha f}a}{I_z\dot{x}} & 0 & 1 - 2T\frac{C_{\alpha r}b^2 + C_{\alpha f}a^2}{I_z\dot{x}} \end{bmatrix} \begin{bmatrix} y - y_r \\ \dot{y} - \dot{y}_r \\ \varphi - \varphi_r \\ \dot{\varphi} - \dot{\varphi}_r \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2TC_{\alpha f}}{m} \\ 0 \\ \frac{2TC_{\alpha f}a}{I_z} \end{bmatrix} [\delta_f - \delta_{fr}] \quad (5.48)$$

5.2.4 Dynamic Model in terms of error variables

The vehicle dynamics model obtained above is based on the vehicle coordinate frame, but for trajectory tracking, it is better to establish the dynamics model in terms of error with respect to road or reference trajectory[29].

$$a_{ydes} = \frac{\dot{x}^2}{R} = \dot{x}\dot{\varphi}_{des} \quad (5.49)$$

$$\ddot{e}_1 = a_y - a_{ydes} = \ddot{y} + \dot{x}(\dot{\varphi} - \dot{\varphi}_{des}) \quad (5.50)$$

Where:

e_1 : the distance of the center of gravity of the vehicle from the center line of the lane, so the \ddot{e}_1 means the error between the desired and actual lateral acceleration.

a_{ydes} : the desired lateral acceleration.

So that the error between the desired and actual lateral velocity is:

$$\dot{e}_1 = \dot{y} + \dot{x}(\varphi - \varphi_{des}) \quad (5.51)$$

The error between the desired and actual lateral heading angle is:

$$e_2 = \varphi - \varphi_{des} \quad (5.52)$$

Therefore:

$$\begin{cases} \dot{y} = \dot{e}_1 - \dot{x}e_2 \\ \dot{\varphi} = \dot{e}_2 + \dot{\varphi}_{des} \end{cases} \quad (5.53)$$

Substituting Equation (5.53) into Equation (5.45), we get:

$$\begin{aligned}
 \begin{bmatrix} \dot{e}_1 \\ \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_2 \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -2\frac{C_{\alpha r}+C_{\alpha f}}{m\dot{x}} & 2\frac{C_{\alpha r}+C_{\alpha f}}{m} & 2\frac{C_{\alpha r}b-C_{\alpha f}a}{m\dot{x}} \\ 0 & 0 & 0 & 1 \\ 0 & 2\frac{C_{\alpha r}b-C_{\alpha f}a}{I_z\dot{x}} & 2\frac{C_{\alpha f}a-C_{\alpha r}b}{I_z} & -2\frac{C_{\alpha r}b^2+C_{\alpha f}a^2}{I_z\dot{x}} \end{bmatrix} \begin{bmatrix} e_1 \\ e_1 \\ e_2 \\ e_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}a}{I_z} \end{bmatrix} \delta_f \\
 &+ \begin{bmatrix} 0 \\ 2\frac{C_{\alpha r}b-C_{\alpha f}a}{m} - \dot{x} \\ 0 \\ -2\frac{C_{\alpha r}b^2+C_{\alpha f}a^2}{I_z\dot{x}} \end{bmatrix} \dot{\varphi}_{des} \quad (5.54)
 \end{aligned}$$

5.3 Rear-Wheel Feedback Algorithm

a) Principle

Establish vehicle kinematic model in Frenet coordinate system:

$$\begin{cases} \dot{s} = \frac{v \cos \varphi_e}{1 - \rho(s)y_e} \\ \dot{y}_e = v \sin \varphi_e \\ \dot{\varphi}_e = \frac{v \tan \delta_f}{L} - \frac{v \rho(s) \cos \varphi_e}{1 - \rho(s)y_e} \end{cases} \quad (5.55)$$

Where $\rho(s)$ is the curvature of road on s .

For the Rear-Wheel Feedback algorithm[36], in order to guarantee the stability of vehicle motion, the angular velocity is:

$$\omega = \frac{v \rho(s) \cos \varphi_e}{1 - \rho(s)y_e} - (k_\varphi |v|) \varphi_e - \left(k_e v \frac{\sin(\varphi_e)}{\varphi_e} \right) y_e \quad (5.56)$$

Where k_φ and k_e are the gains for the heading error and lateral displacement error.

Combined with Ackerman steering model Equation (5.2):

$$\tan \delta_f = \frac{L}{R} = \frac{\omega L}{v} \quad (5.57)$$

b) Simulation Result

Figure 5.23 and 5.25 is the result of path tracking and heading angle tracking of Rear-Wheel Feedback controller with velocity equal to 10m/s. The error of displacement and heading angle are plotted in Figure 5.24 and 5.26.

The average lateral displacement error is 4.3668cm, the average heading error is 0.2714deg which are worse slightly than the pure pursuit and Stanley controller, but still not bad.

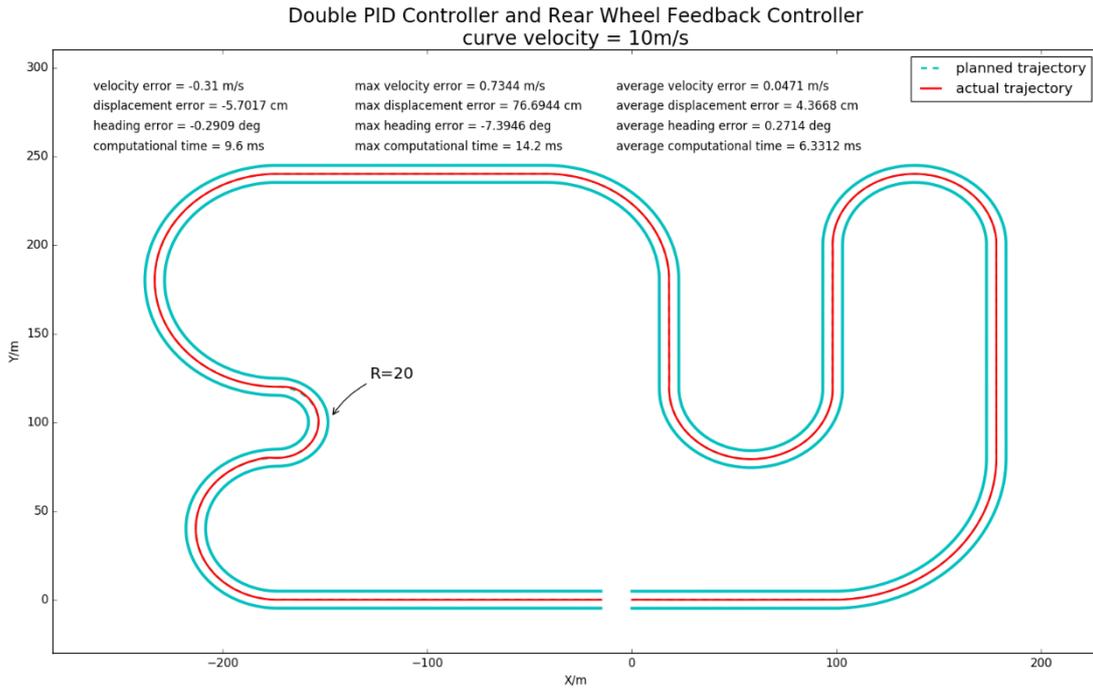


Figure 5.23 Path Tracking of Rear-Wheel Feedback at V=10 m/s

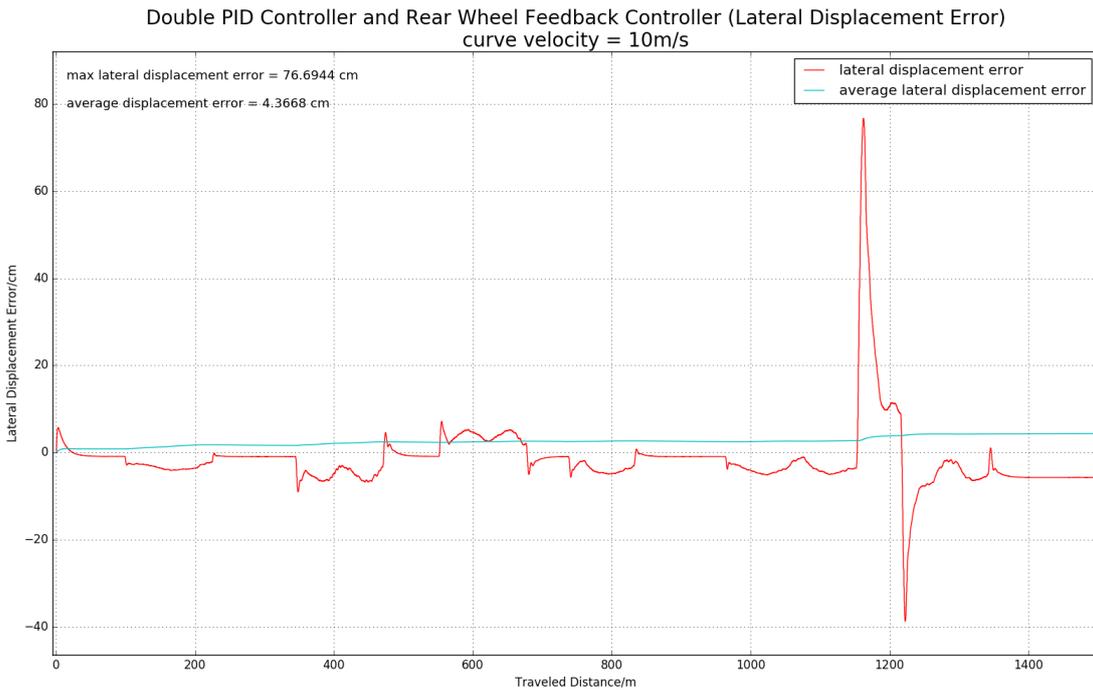


Figure 5.24 Lateral Displacement Error of Rear-Wheel Feedback at V=10 m/s

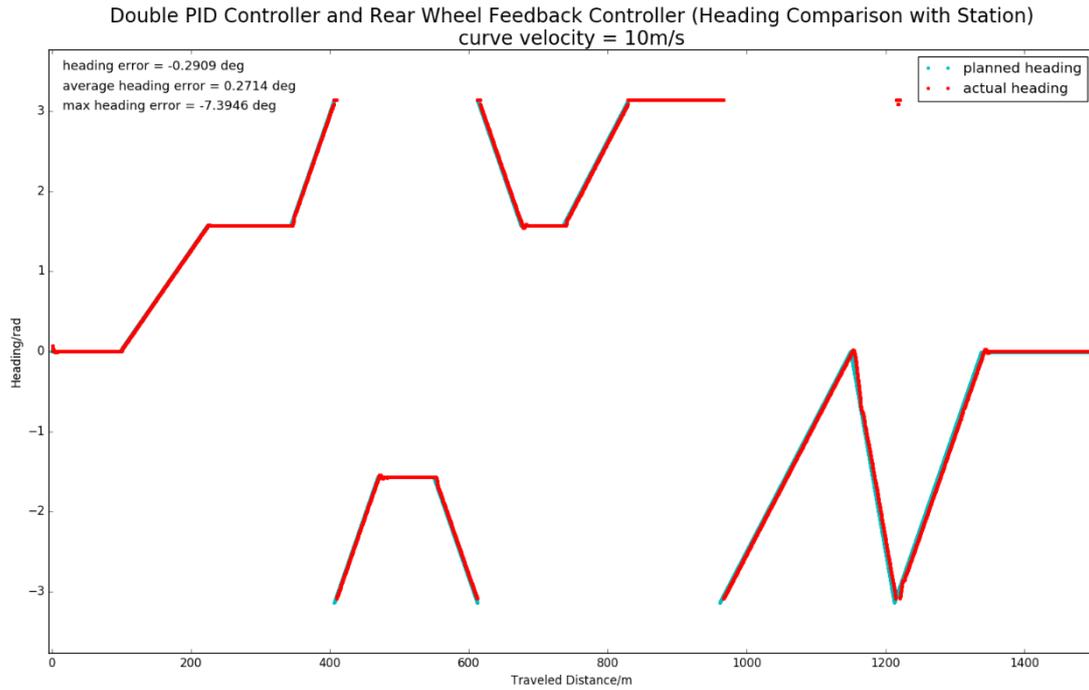


Figure 5.25 Heading Tracking of Rear-Wheel Feedback at V=10 m/s

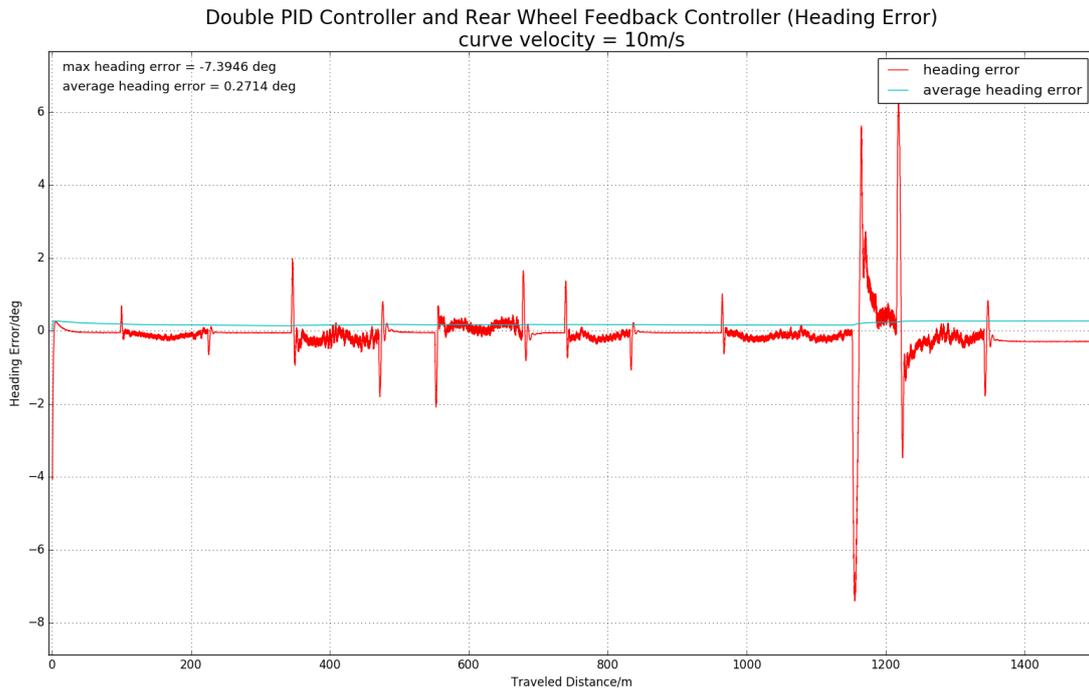


Figure 5.26 Heading Error of Rear-Wheel Feedback at V=10 m/s

Figure 5.27 and 5.29 is the result of path tracking and heading angle tracking of Rear-Wheel Feedback controller with velocity equal to 15m/s.

The error of displacement and heading angle are plotted in Figure 5.28 and 5.30.

The average lateral displacement error is 75.06cm, the average heading error is 1.9662deg which is close to the result of Stanley controller, and better than the result of pure pursuit controller.

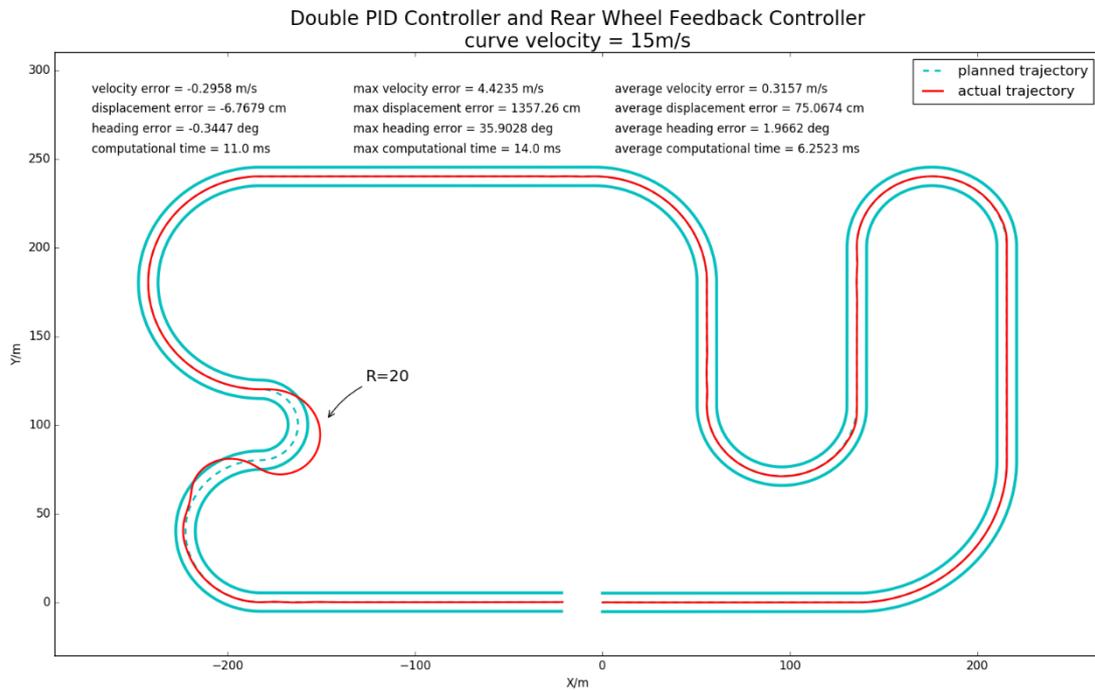


Figure 5.27 Path Tracking of Rear-Wheel Feedback at V=15 m/s

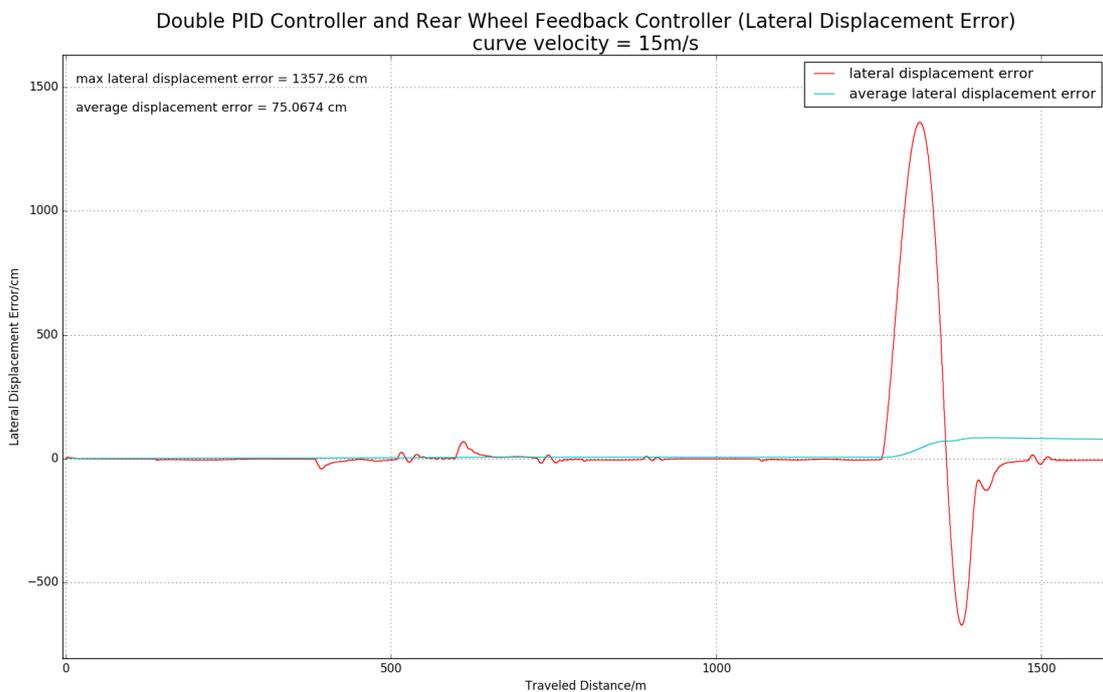


Figure 5.28 Lateral Displacement Error of Rear-Wheel Feedback at V=15 m/s

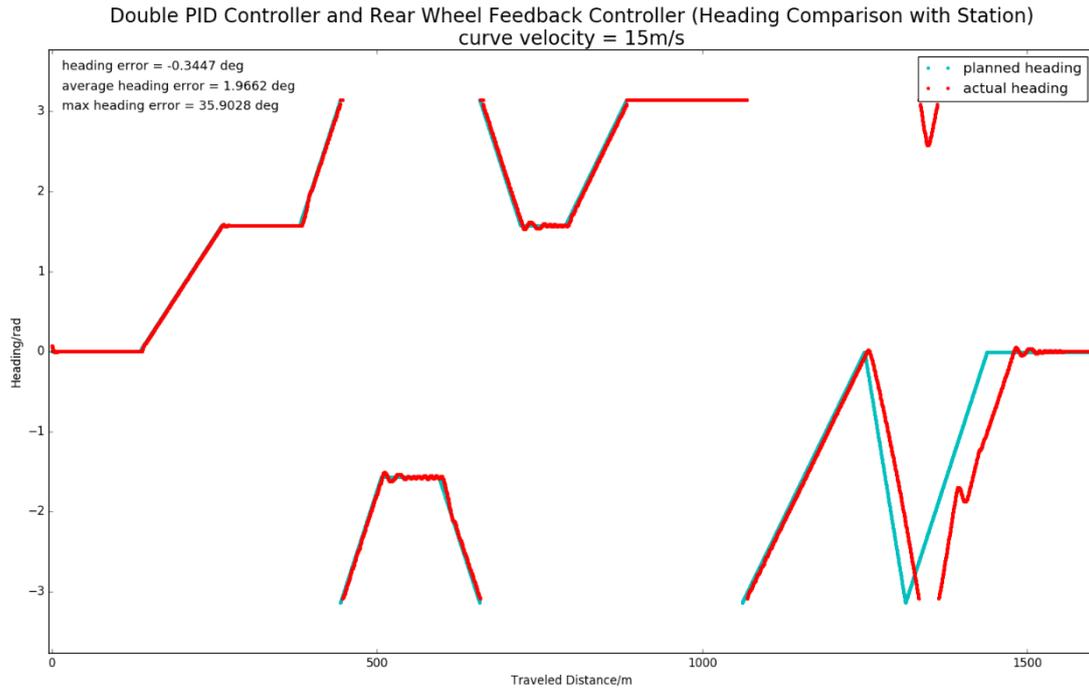


Figure 5.29 Heading Tracking of Rear-Wheel Feedback at V=15 m/s

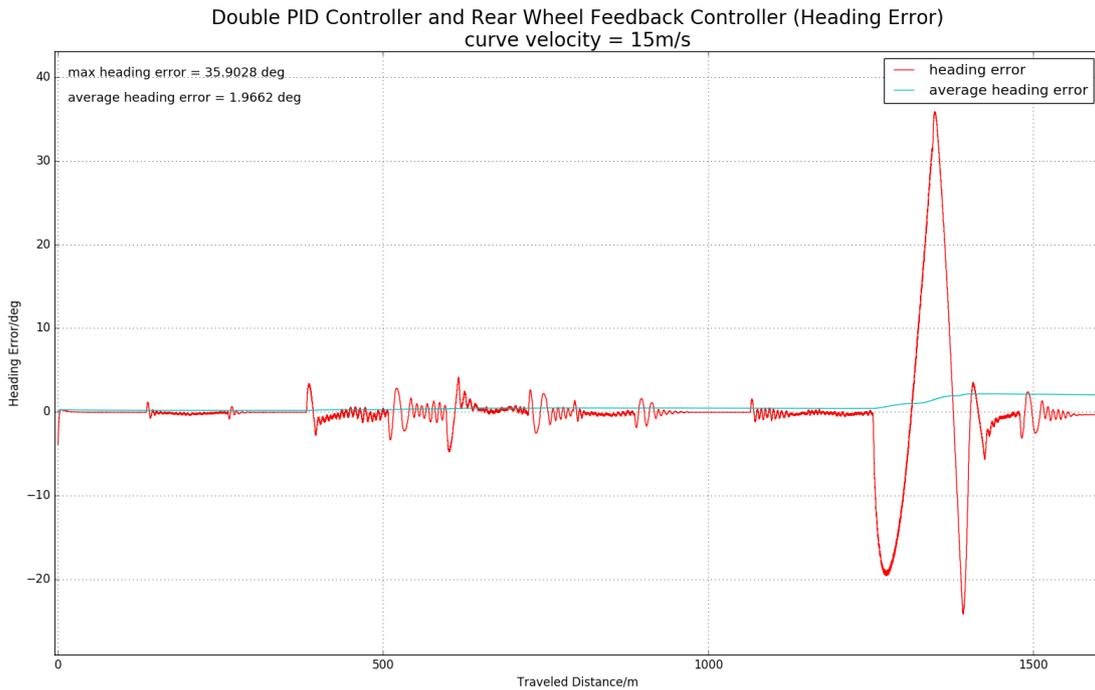


Figure 5.30 Heading Error of Rear-Wheel Feedback at V=15 m/s

5.4 LQR Control

5.4.1 Introduction

LQR is a linear quadratic regulator. Its control object is a linear system given in the form of state space in modern control theory, and the cost function is a quadratic function of state and control quantity[37].

The LQR optimal design means that the designed state feedback controller K should make the quadratic cost function J take the minimum value, and K is uniquely determined by the weight matrix Q and R , so the choice of Q and R is particularly important.

LQR theory is the earliest and most mature state space design method in modern control theory. Especially, LQR can obtain the optimal control law of state linear feedback, which is easy to form a closed-loop optimal control.

5.4.2 LQR Formula Proof

The cost function is:

$$J = \frac{1}{2} (\sum_{t=0}^{N-1} (x^T Q x + u^T R u) + x_N^T Q_0 x_N) \quad (5.58)$$

Where x is the state error with respect to the reference state, u is the feedback control quantity, x_N is the final state error with respect to the reference state of the vehicle.

$V_{\tau T_0 N}(z)$ is defined as the minimum cost from state z at $t = \tau$ to the final state:

$$V_{\tau T_0 N}(z) = \min_{u_{\tau} \dots u_{N-1}} \sum_{t=\tau}^{N-1} (x_t^T Q x_t + u_t^T R u_t) + x_N^T Q_0 x_N \quad (5.59)$$

When $\tau = N$, $x_N = z$, so:

$$V_{\tau T_0 N}(z) = x_N^T Q_0 x_N \quad (5.60)$$

When $\tau = N - 1$, $x_{N-1} = z$, $u_{N-1} = \omega$:

$$V_{\tau T_0 N}(z) = \min_{\omega} (x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1}) + x_N^T Q_0 x_N \quad (5.61)$$

When $\tau = N - 2$, $x_{N-2} = z$, $u_{N-2} = \omega$:

$$V_{\tau T_0 N}(z) = \min_{\omega} (x_{N-1}^T Q x_{N-1} + u_{N-1}^T R u_{N-1} + x_{N-2}^T Q x_{N-2} + u_{N-2}^T R u_{N-2}) + x_N^T Q_0 x_N \quad (5.62)$$

Therefore, the relation between the adjacent $V_{\tau T_0 N}(z)$ is:

$$V_{\tau T_0 N}(z) = \min_{\omega} (z_t^T Q z_t + \omega^T R \omega + V_{\tau+1 T_0 N}(A z_t + B \omega)) \quad (5.63)$$

If the overall cost is minimum, the cost of every step must be minimum too. So, the problem has been transformed into a dynamic programming problem.

We suppose there is a matrix P , which satisfies:

$$V_{\tau T o N}(z) = z_t^T P_t z_t \quad (5.64)$$

So that:

$$\begin{aligned} V_{\tau T o N}(z) &= \min_{\omega} (z_t^T Q z_t + \omega^T R \omega + V_{\tau+1 T o N}(A z_t + B \omega)) \\ &= z_t^T Q z_t + \min_{\omega} (\omega^T R \omega + (A z_t + B \omega)^T P_{t+1} (A z_t + B \omega)) = z_t^T P_t z_t \end{aligned} \quad (5.65)$$

The ω where the derivation = 0 is corresponding to the minimum cost.

After derivation:

$$2\omega^T R + 2(A z_t + B \omega)^T P_{t+1} B = 0 \quad (5.66)$$

$$\omega^* = -(R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A z_t \quad (5.67)$$

Substituting it into $V_{\tau T o N}(z)$:

$$V_{\tau T o N}(z) = z_t^T (Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A) z_t = z_t^T P_t z_t \quad (5.68)$$

Therefore:

$$P_t = Q + A^T P_{t+1} A - A^T P_{t+1} B (R + B^T P_{t+1} B)^{-1} B^T P_{t+1} A \quad (5.69)$$

Solve step:

- Let $P = Q$.
- Iterating forward through Equation (5.69) to get P_t until the difference between adjacent P_t is lower than threshold, the steady state P_t is the final P .
- $K = -(R + B^T P B)^{-1} B^T P A$.
- Feedback control quantity $u = -K * x$.

5.4.3 Control Block

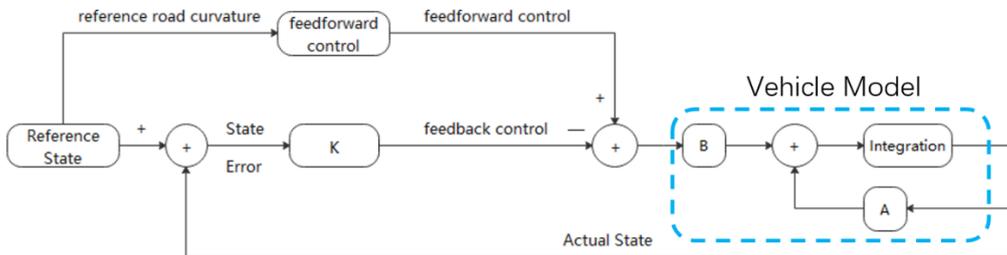


Figure 5.31 LQR Control Block

Figure 5.31 shows the control block of control module using LQR controller which includes the feedforward and feedback parts.

Feedback control quantity:

$$u_{fb} = -K * x \tag{5.70}$$

Feedforward control quantity[38]:

$$u_{ff} = \arctan(\kappa L) \tag{5.71}$$

where κ is the curvature of preview point in reference path.

The final control input for the vehicle is:

$$u = u_{fb} + u_{ff} \tag{5.72}$$

5.4.4 Simulation Result

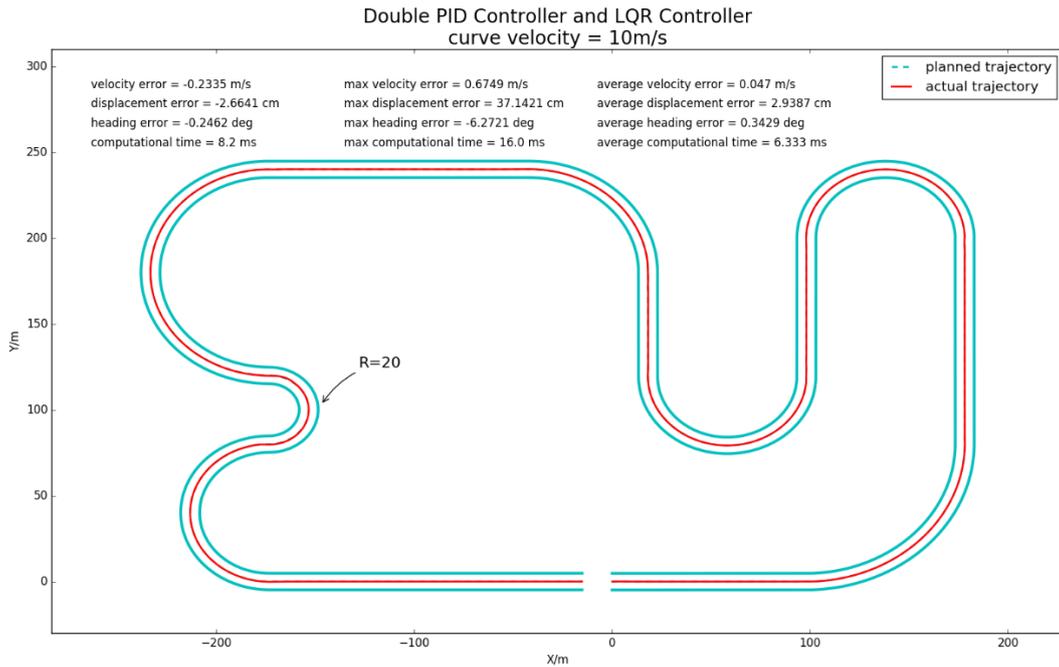


Figure 5.32 Path Tracking of LQR at V=10 m/s

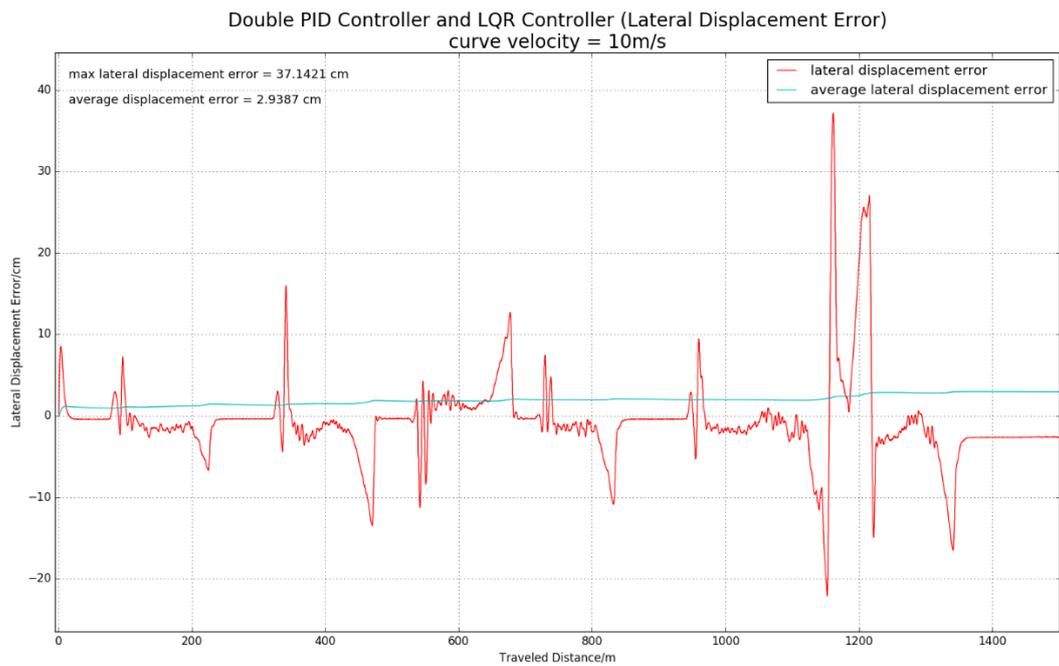


Figure 5.33 Lateral Displacement Error of LQR at V=10 m/s

Figure 5.32 and 5.34 is the result of path tracking and heading angle tracking of LQR controller with velocity equal to 10m/s. The error of displacement and heading angle are plotted in Figure 5.33 and 5.35.

The average lateral displacement error is 2.9387cm, the average heading error is 0.3429deg which are close to the result of Stanley controller and better than pure pursuit and rear wheel feedback controller.

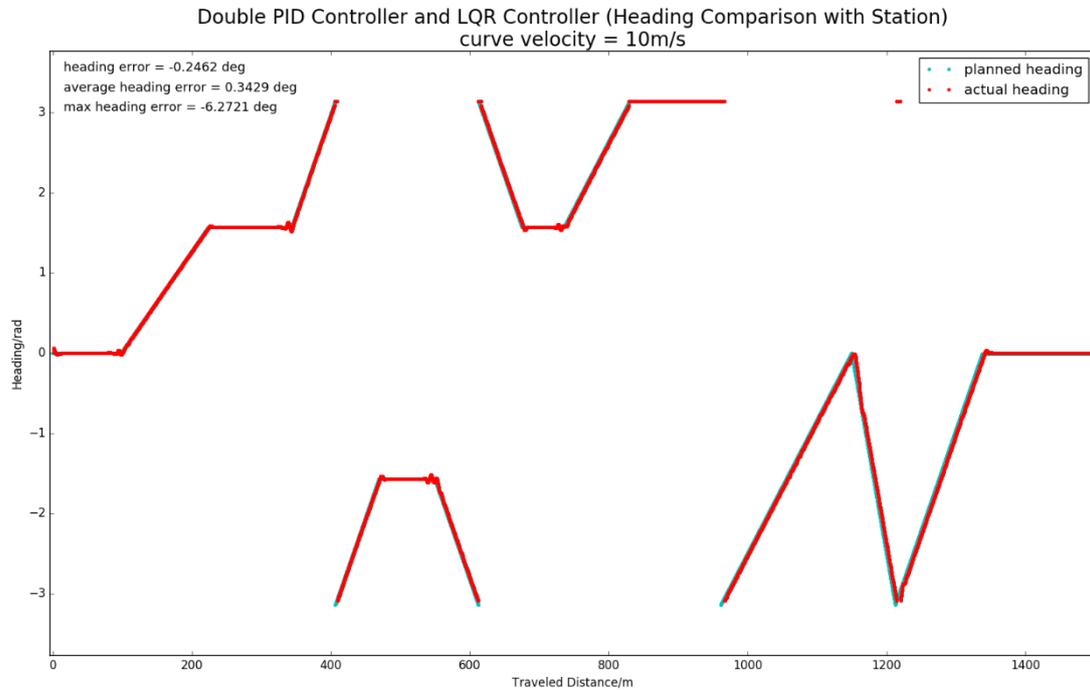


Figure 5.34 Heading Tracking of LQR at V=10 m/s

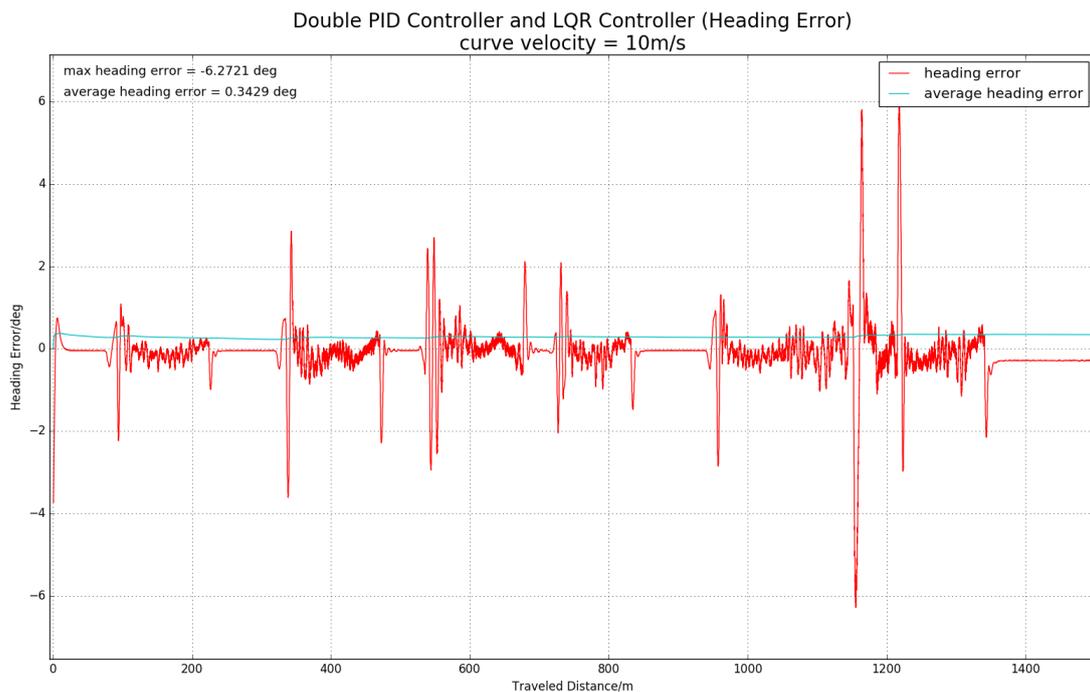


Figure 5.35 Heading Error of LQR at V=10 m/s

Figure 5.36 and 5.38 is the result of path tracking and heading angle tracking of LQR controller with velocity equal to 15m/s.

The error of displacement and heading angle are plotted in Figure 5.37 and 5.39.

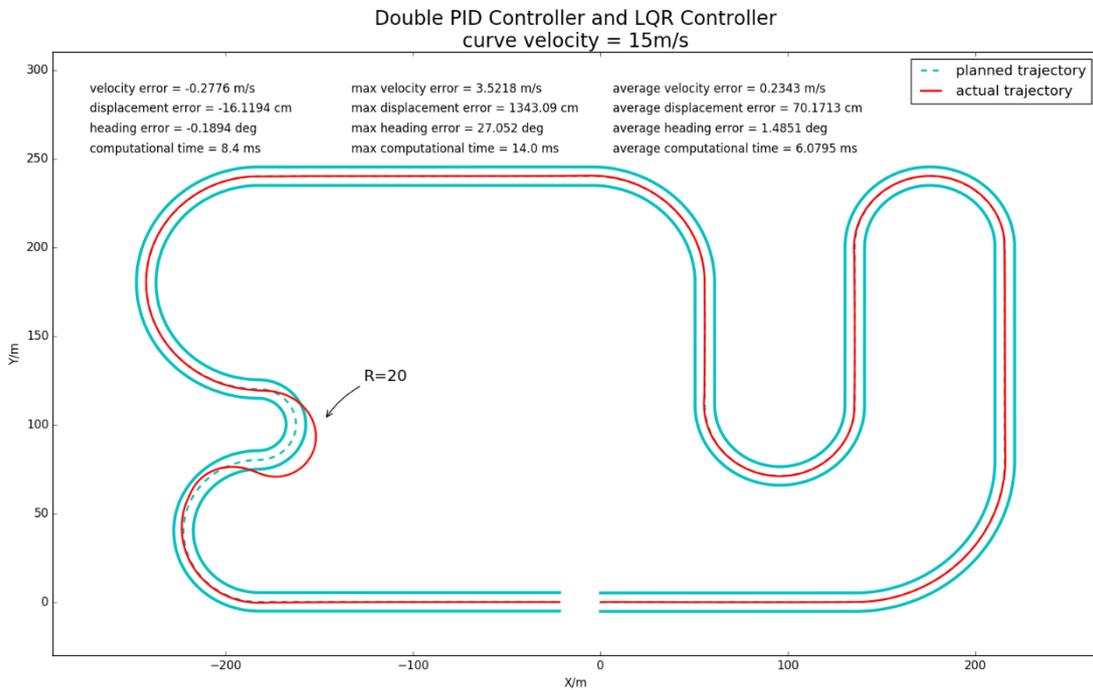


Figure 5.36 Path Tracking of LQR at V=15 m/s

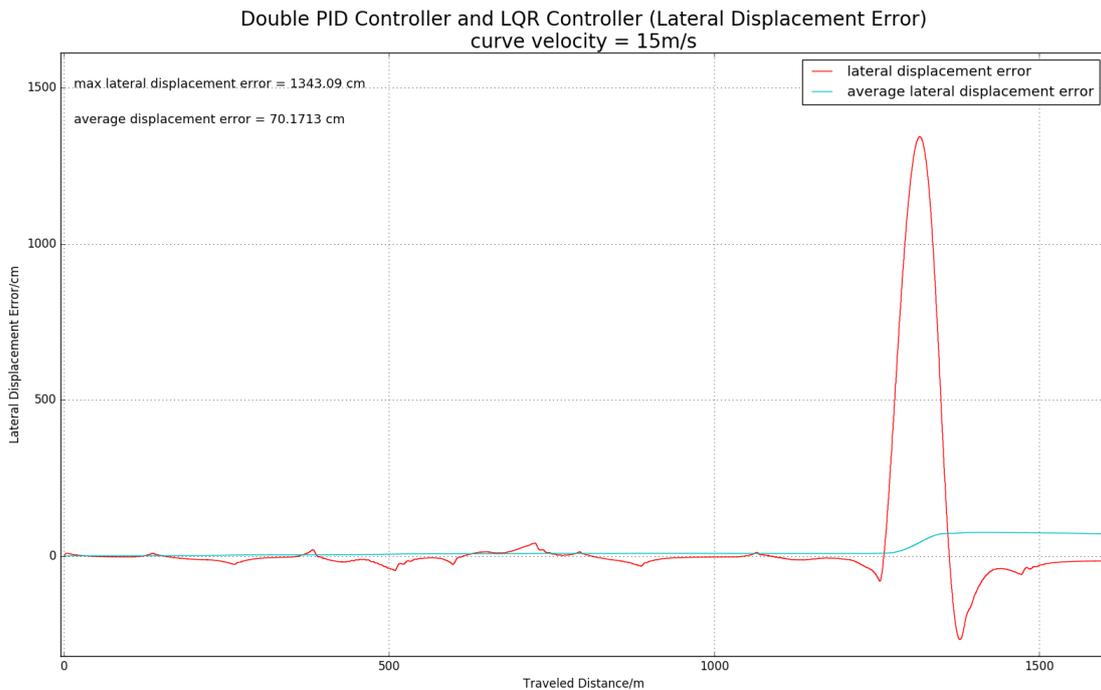


Figure 5.37 Lateral Displacement Error of LQR at V=15 m/s

The average lateral displacement error is 70.1713cm, the average heading error is 1.4851deg which is best one compared to the pure pursuit, Stanley and rear wheel feedback controllers.

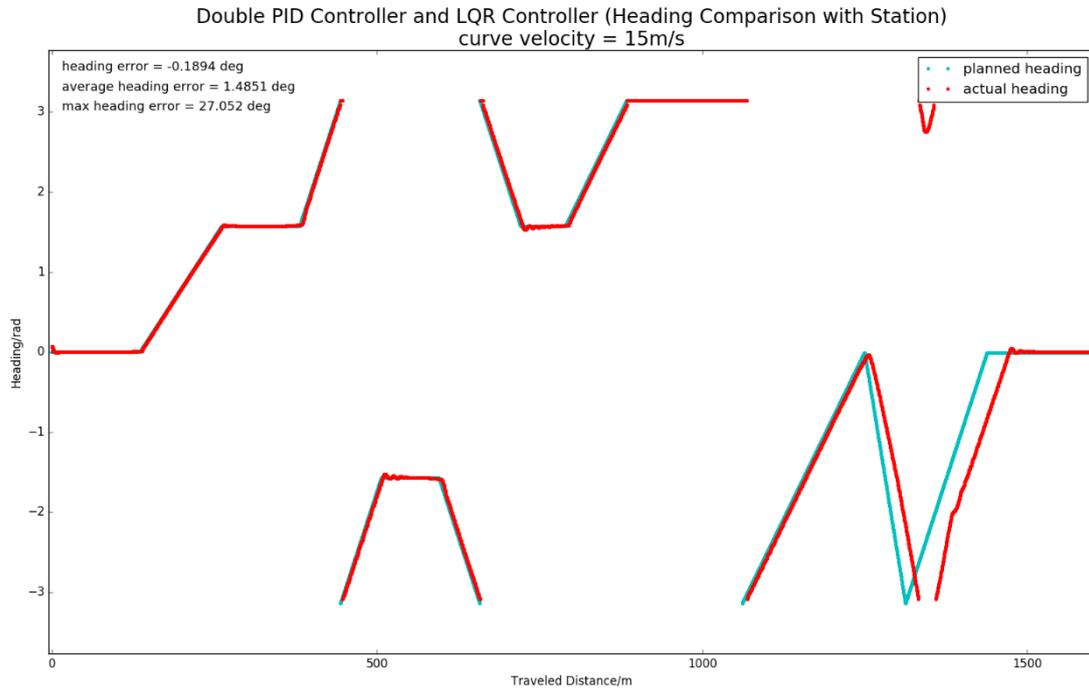


Figure 5.38 Heading Tracking of LQR at V=15 m/s

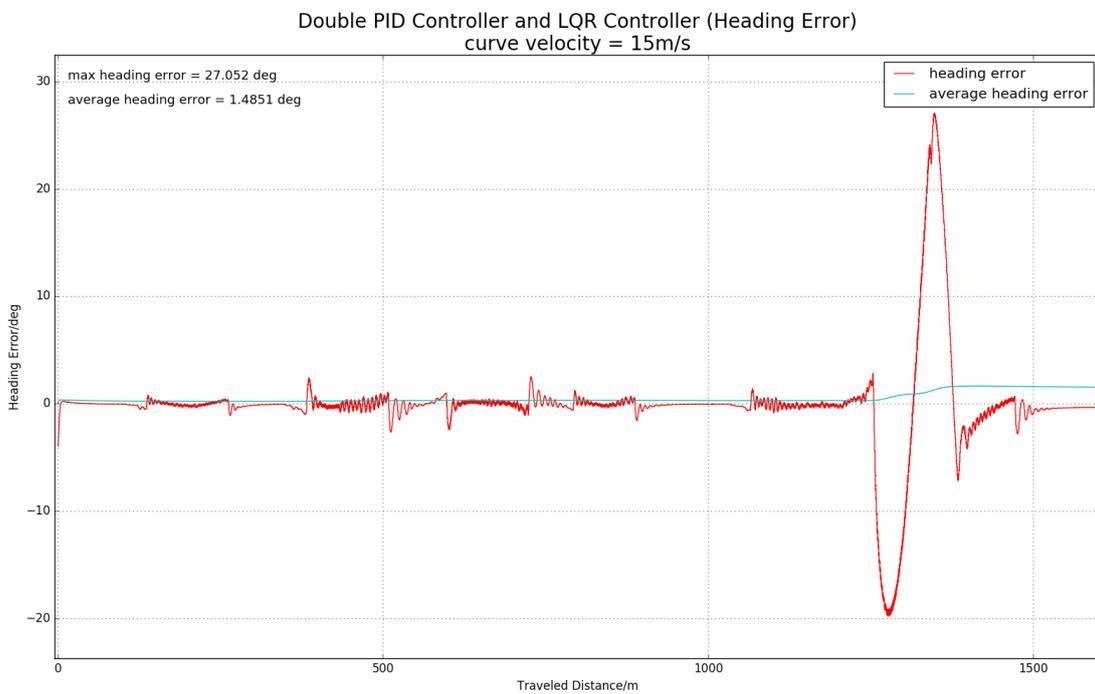


Figure 5.39 Heading Error of LQR at V=15 m/s

5.5 MPC

5.5.1 Introduction

Autonomous driving vehicle faces a dynamically changing complex environment and must meet safety and actuator constraints. The model predictive control algorithm has the ability to systematically consider predictive information and deal with multi-constraints optimization problems.

Linear time-varying model predictive control (MPC)[39] uses a linear time-varying model as the predictive model. Its biggest advantage is that the calculation is relatively simple and the real-time performance is good which is crucial for the motion control of autonomous vehicles.

5.5.2 Principle

The basic principle of model predictive control is shown in Figure 5.40.

During the control process, there is a reference trajectory, as shown in the figure. Taking k as the current timestamp, the controller combines the current control quantity and predictive model to predict the output state of the vehicle in the future time domain $[k, k + N_p]$ which is also called the prediction horizon.

By solving the optimization problem that satisfies the cost function and constraints, a control quantity sequences in the control time domain $[k, k + N_c]$ are obtained, as shown by the green rectangular wave in the figure. Then the first element of the control quantity sequences is applied to the controlled object as the actual control quantity.

When it comes to the next moment $k + 1$, repeat the above steps, so rolling to complete the optimization problems with constraints, in order to achieve continuous control of the controlled object.

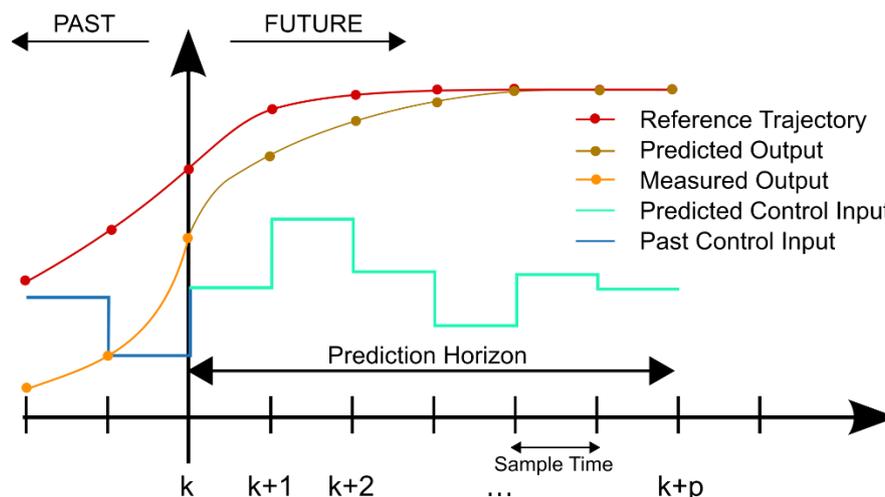


Figure 5.40 MPC Principle

5.5.3 Increment Model

Linear state space model is necessary for linear model predictive control.

There is a discrete linear vehicle model:

$$x(k+1) = Ax(k) + Bu(k) \quad (5.73)$$

In this case, the control quantity $u(k)$ is the feedback control quantity, and all the constraints are all about it.

If $u(k)$ is control increment, by applying constraints on it, the change rate of the control quantity can be limited to achieve smoother and stable tracking.

Let:

$$\xi(k|t) = \begin{bmatrix} x(k|t) \\ u(k-1|t) \end{bmatrix} \quad (5.74)$$

$$\Delta u(k|t) = u(k|t) - u(k-1|t) \quad (5.75)$$

So that:

$$\xi(k+1|t) = A'_t \xi(k|t) + B'_t \Delta u(k|t) \quad (5.76)$$

$$\eta(k|t) = C'_t \xi(k|t) \quad (5.77)$$

Where:

$$A'_t = \begin{bmatrix} A_t & B_t \\ 0_{m \times n} & I_m \end{bmatrix}$$

$$B'_t = \begin{bmatrix} B_t \\ I_m \end{bmatrix}$$

$$C'_t = [C_t \quad 0]$$

5.5.4 Predictive Model

Predictive model is the basis of model predictive control. Its main function is to predict the future output of the system based on the historical information and future input of the object.

$$\xi(k+1) = A\xi(k) + Bu(k) \quad (5.78)$$

The state of next sampling time is:

$$\xi(k+2) = A\xi(k+1) + Bu(k+1) \quad (5.79)$$

$$\xi(k+2) = A(A\xi(k) + Bu(k)) + Bu(k+1) \quad (5.80)$$

So that as long as the initial state and all control sequences are known, the future state of the vehicle can be predicted, and the prediction model can be obtained.

Let the prediction horizon is N_p and control horizon is N_c , so that the future state of vehicle can be calculated as:

$$\xi(t+N_p|t) = A_t'^{N_p} \xi(t|t) + A_t'^{N_p-1} B_t' \Delta u(t|t) + \dots + A_t'^{N_p-N_c-1} B_t' \Delta u(t+N_c|t) \quad (5.81)$$

$$\eta(t+N_p|t) = C_t' (A_t'^{N_p} \xi(t|t) + A_t'^{N_p-1} B_t' \Delta u(t|t) + \dots + A_t'^{N_p-N_c-1} B_t' \Delta u(t+N_c|t))$$

(5.82)

The predictive model can be represented in the form of matrix:

$$Y(t) = \Psi_t \xi(k|t) + \Theta \Delta U(t) \quad (5.83)$$

Where:

$$Y(t) = \begin{bmatrix} \eta'(k+1|t) \\ \eta'(k+2|t) \\ \dots \\ \eta'(k+N_c|t) \\ \dots \\ \eta'(k+N_p|t) \end{bmatrix} \quad \Psi_t = \begin{bmatrix} C'_t A'_t \\ C'_t A'^2_t \\ \dots \\ C'_t A'^{N_c}_t \\ \dots \\ C'_t A'^{N_p}_t \end{bmatrix}$$

$$\Theta = \begin{bmatrix} C'_t B'_t & 0 & 0 & 0 \\ C'_t A'_t B'_t & C'_t B'_t & 0 & 0 \\ \dots & \dots & \ddots & \dots \\ C'_t A'^{N_c-1}_t B'_t & C'_t A'^{N_c-2}_t B'_t & \dots & C'_t B'_t \\ C'_t A'^{N_c}_t B'_t & C'_t A'^{N_c-1}_t B'_t & \dots & C'_t A'_t B'_t \\ \vdots & \vdots & \ddots & \vdots \\ C'_t A'^{N_p-1}_t B'_t & C'_t A'^{N_p-2}_t B'_t & \dots & C'_t A'^{N_p-N_c}_t B'_t \end{bmatrix} \quad \Delta U(t) = \begin{bmatrix} \Delta u(k|t) \\ \Delta u(k+1|t) \\ \dots \\ \Delta u(k+N_c|t) \end{bmatrix}$$

5.5.5 Cost Function Design

Actually, the control sequence $\Delta U(t)$ is unknown. Only by setting reasonable optimization goals and solving them, can the control sequence be obtained.

The purpose of designing the cost function is to obtain the control quantity acting on the autonomous vehicle at the moment when the value of cost function is minimum which is also the optimization goal.

First, in order to guarantee the path tracking accuracy and stability of the driverless vehicle in the lateral control process, the difference between the predicted value and the measured value of the output must be considered.

Secondly, in order to ensure that the smooth change of the steering angle of front wheel which ensure the stable driving of the automated vehicle, an item related to the control increment has to be took into account.

Therefore, the cost function is:

$$J = \sum_{i=0}^{N_p} \|y'(k+i) - y'_r(k+i)\|_Q^2 + \sum_{i=0}^{N_c} \|\Delta u(k+i)\|_R^2 \quad (5.84)$$

The first term means the ability of algorithm to track reference trajectory in prediction horizon. The second term means the requirement of smooth control. The entire expression is used to evaluate the performance of MPC algorithm to track desired trajectory fast and stably.

However, the state space matrices are time-varying, the feasible solution cannot be obtained for the optimization goal sometimes. Therefore, it is necessary to add a relaxation factor into the cost function:

$$J = \sum_{i=0}^{N_p} \|y'(k+i) - y_r'(k+i)\|_Q^2 + \sum_{i=0}^{N_c} \|\Delta u(k+i)\|_R^2 + \rho \varepsilon^2 \quad (5.85)$$

where ρ is the weight, and ε is a relaxation factor.

5.5.6 Constraints

In the actual control system, it is often necessary to meet some constraints of the system state quantity and the control quantity, shown following:

Constraints for increment of control quantity:

$$\Delta u_{\min}(t+k) \leq \Delta u(t+k) \leq \Delta u_{\max}(t+k) \quad k = 0, 1, \dots, N_c - 1 \quad (5.86)$$

Constraints for control quantity:

$$u_{\min}(t+k) \leq u(t+k) \leq u_{\max}(t+k) \quad k = 0, 1, \dots, N_c - 1 \quad (5.87)$$

Since in the vehicle model applied, the control quantity is the increment of steering angle with respect to the amount at last timestamp, so in the quadratic programming solver, it is necessary to transform the constraints for the steering angle, the Equation (5.87) into the form of the increment of steering angle.

There is the relationship between them:

$$\left\{ \begin{array}{l} u(k|k) = \Delta u(k|k) + u(k-1|k) \\ u(k+1|k) = \Delta u(k+1|k) + \Delta u(k|k) + u(k-1|k) \\ \vdots \\ u(k+N_c-1|k) = \Delta u(k+N_c-1|k) + \dots + \Delta u(k+1|k) + \Delta u(k|k) + u(k-1|k) \end{array} \right.$$

which can be represented in the form of matrix:

$$U(k) = G\Delta U(k) + \tilde{U} \quad (5.88)$$

$$\begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N_c-1|k) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 1 & \dots & \dots & 1 \end{bmatrix} \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+N_c-1|k) \end{bmatrix} + \begin{bmatrix} u(k-1|k) \\ u(k-1|k) \\ \vdots \\ u(k-1|k) \end{bmatrix}$$

So, the constraints for control quantity, Equation (5.87) is updated:

$$U_{\min}(k) < G\Delta U(k) + \tilde{U} < U_{\max}(k) \quad (5.89)$$

Constraints for output:

$$y_{\min}(t+k) \leq y(t+k) \leq y_{\max}(t+k) \quad k = 0, 1, \dots, N_c - 1 \quad (5.90)$$

Combined with Equation (5.83) and (5.85):

$$\begin{aligned} Y_{\min}(k) - \varepsilon < \Psi_t \xi(k|t) + \Theta \Delta U(t) < Y_{\max}(k) + \varepsilon \\ A^*(k) \Delta u(k) < b^*(k) \end{aligned} \quad (5.91)$$

$$A^*(k) = \begin{bmatrix} G & 0 \\ -G & 0 \\ \theta & 0 \\ -\theta & 0 \end{bmatrix}$$

$$b^*(k) = \begin{bmatrix} U_{max}(k) - \tilde{U} \\ -U_{min}(k) + \tilde{U} \\ Y_{max}(k) + \varepsilon - \Psi_t \xi(k|t) \\ -Y_{min}(k) + \varepsilon + \Psi_t \xi(k|t) \end{bmatrix}$$

5.5.7 Quadratic programming

The cost function shown as Equation (5.85) can be converted into a quadratic programming problem in terms of the control sequence $\Delta U(t)$ through proper processing.

$$J = (\Psi_t \xi + \theta \Delta U - Y_{ref})^T Q (\Psi_t \xi + \theta \Delta U - Y_{ref}) + \Delta U^T R \Delta U + \rho \varepsilon^2$$

$$E = \Psi_t \xi - Y_{ref}$$

$$Y_{ref}(t) = \begin{bmatrix} \eta'_{ref}(k+1|t) \\ \eta'_{ref}(k+2|t) \\ \dots \\ \eta'_{ref}(k+N_c|t) \\ \dots \\ \eta'_{ref}(k+N_p|t) \end{bmatrix}$$

So that:

$$\begin{aligned} J &= (E + \theta \Delta U)^T Q (E + \theta \Delta U) + \Delta U^T R \Delta U + \rho \varepsilon^2 \\ &= E^T Q E + (\theta \Delta U)^T Q (\theta \Delta U) + 2E^T Q (\theta \Delta U) + (\Delta U)^T R \Delta U + \rho \varepsilon^2 \\ &= (\Delta U)^T (\theta^T Q \theta + R) \Delta U + (2E^T Q \theta) \Delta U + E^T Q E + \rho \varepsilon^2 \end{aligned}$$

Therefore, the optimization goal can be transformed into:

$$\min_{\Delta U(t)} [\Delta U(t)^T, \varepsilon]^T H [\Delta U(t)^T, \varepsilon] + G [\Delta U(t)^T, \varepsilon] \quad (5.92)$$

$$H = \begin{bmatrix} \theta^T Q \theta + R & 0 \\ 0 & \rho \end{bmatrix}$$

$$G = [2E^T Q \theta \quad 0]$$

$$\Delta U_{min} \leq \Delta U(k) \leq \Delta U_{max}, \quad k = t, \dots, t + N_c - 1$$

$$U_{min}(k) < G \Delta U(k) + \tilde{U} < U_{max}(k)$$

$$A^*(k) \Delta U(k) < b^*(k)$$

The result of the quadratic programming problem is the control sequence:

$$\Delta U_t^* = [\Delta u_t^* \quad \Delta u_{t+1}^* \quad \dots \quad \Delta u_{t+N_c-1}^*]$$

Taking the first element as the feedback control increment, and the feedback control quantity is:

$$u_{fb} = u(k-1) + \Delta u_t^* \quad (5.93)$$

Feedforward control quantity[40]:

$$u_{ff} = \arctan(\kappa L) \tag{5.94}$$

The final control input for the vehicle is:

$$u = u_{fb} + u_{ff} \tag{5.95}$$

5.5.8 Simulation Result

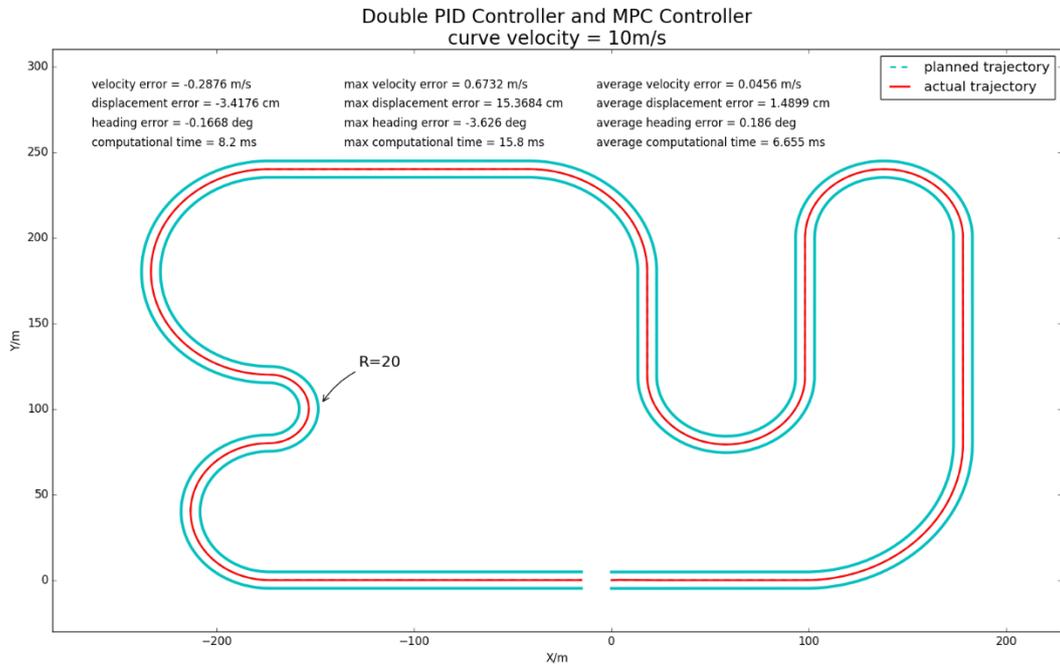


Figure 5.41 Path Tracking of MPC at V=10 m/s

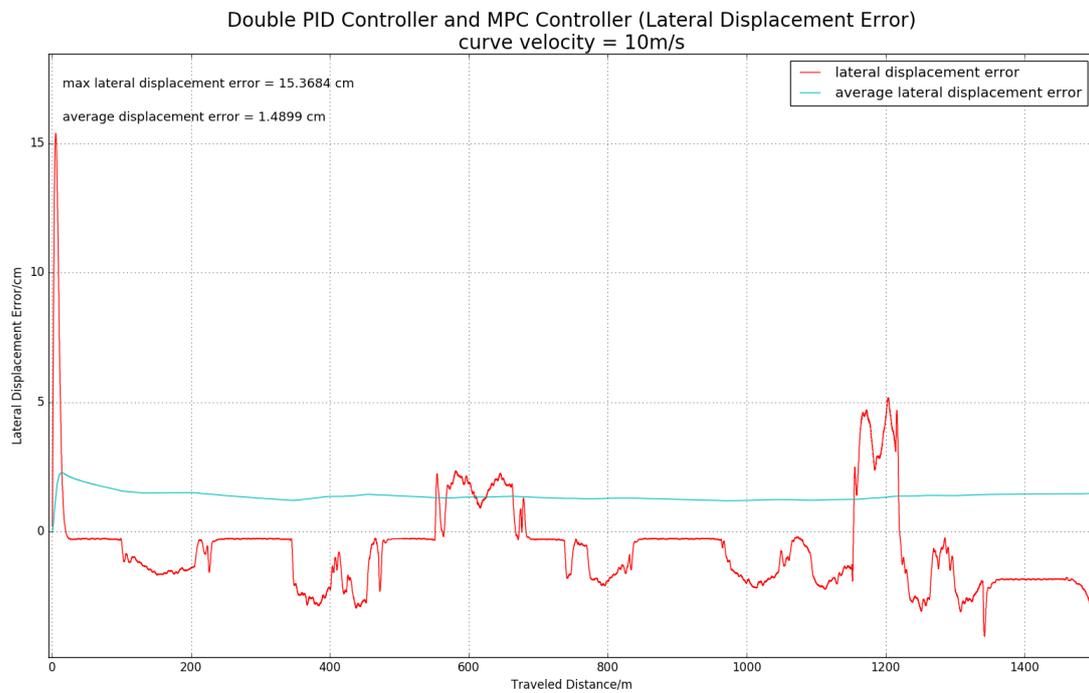


Figure 5.42 Lateral Displacement Error of MPC at V=10 m/s

Figure 5.41 and 5.43 is the result of path tracking and heading angle tracking of MPC controller with velocity equal to 10m/s. The error of displacement and heading angle are plotted in Figure 5.42 and 5.44.

The average lateral displacement error is 1.4899cm, the average heading error is 0.186deg which are the smallest.

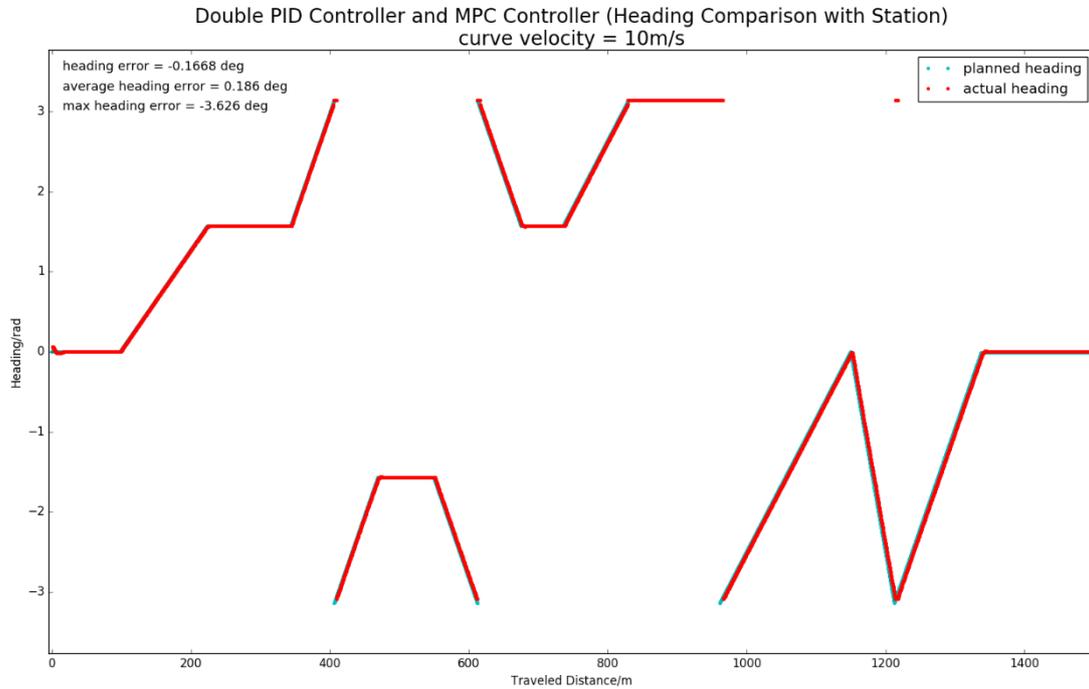


Figure 5.43 Heading Tracking of MPC at V=10 m/s

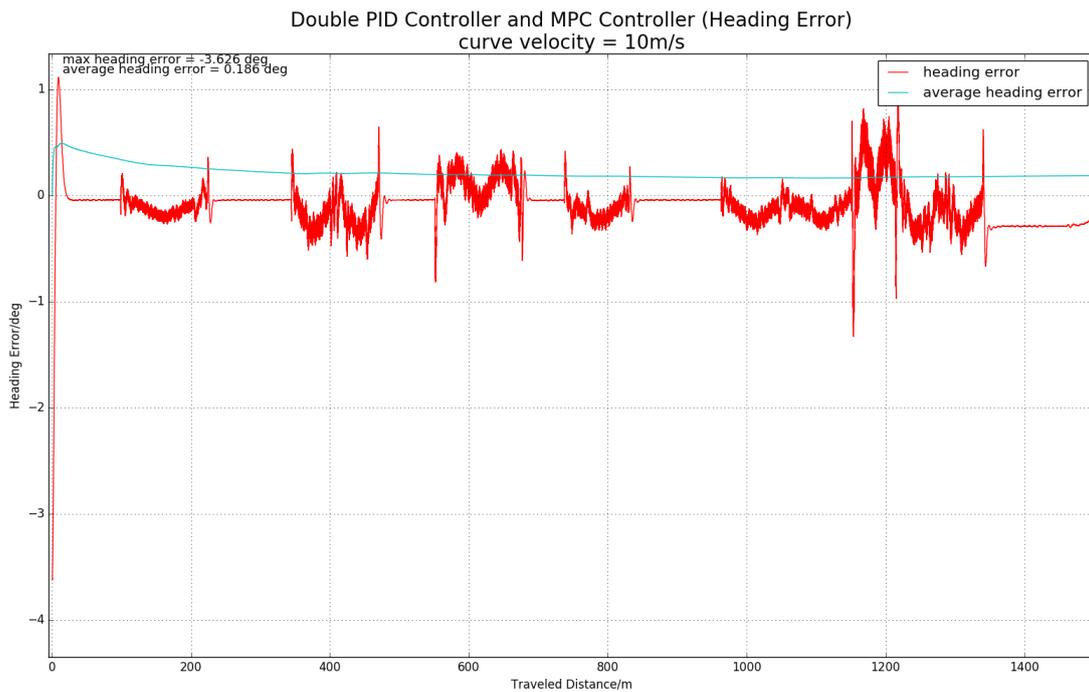


Figure 5.44 Heading Error of MPC at V=10 m/s

Figure 5.45 and 5.47 is the result of path tracking and heading angle tracking of MPC controller with velocity equal to 15m/s.

The error of displacement and heading angle are plotted in Figure 5.46 and 5.48.

The average lateral displacement error is 56.835cm, the average heading error is 1.3639deg which are also the best among all the controllers.

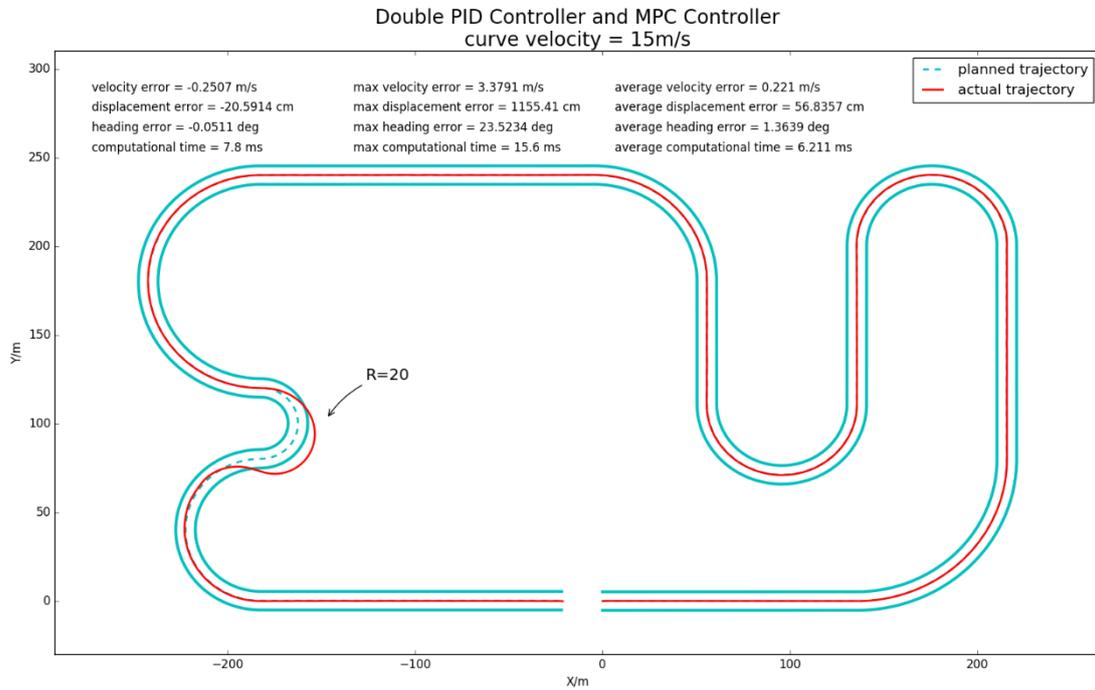


Figure 5.45 Path Tracking of MPC at V=15 m/s

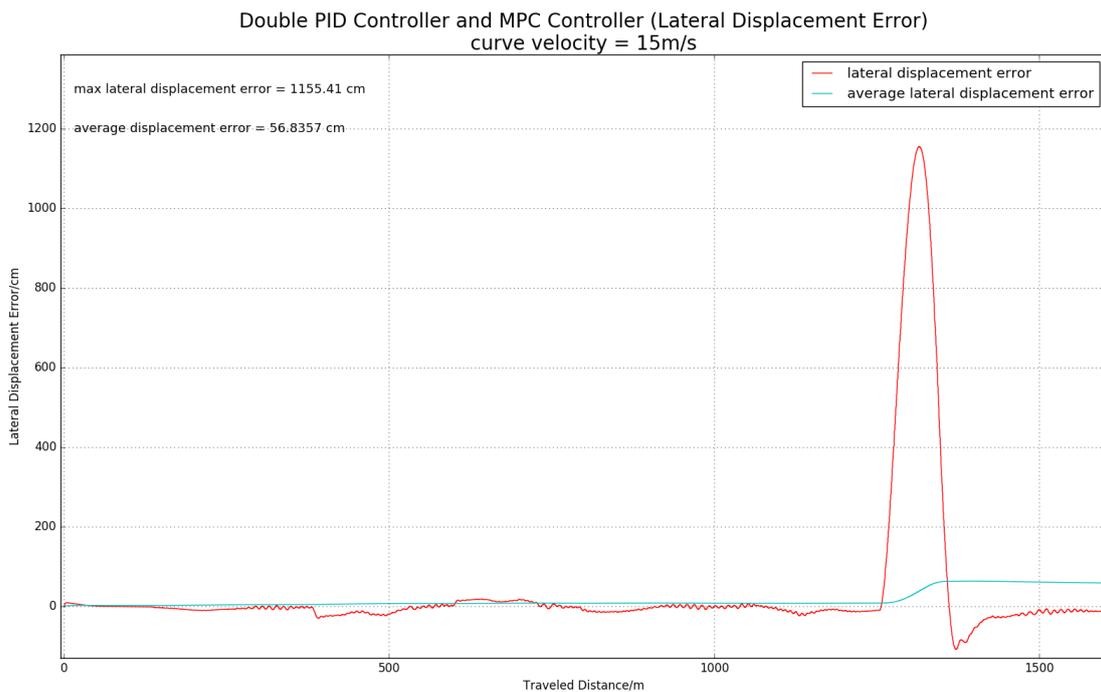


Figure 5.46 Lateral Displacement Error of MPC at V=15 m/s

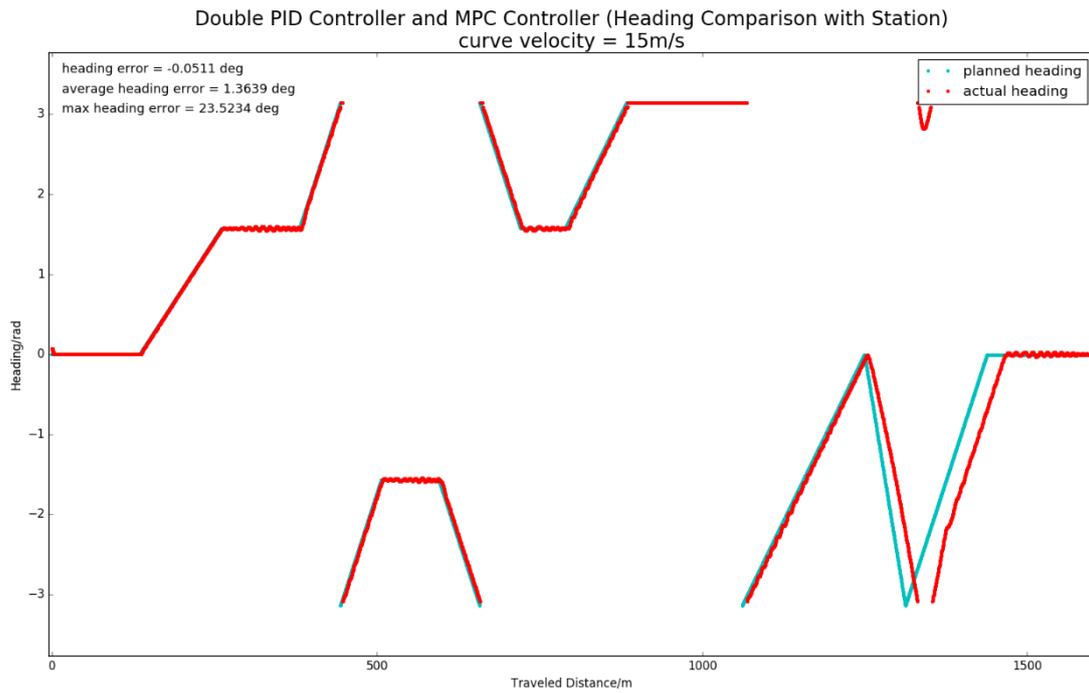


Figure 5.47 Heading Tracking of MPC at V=15 m/s

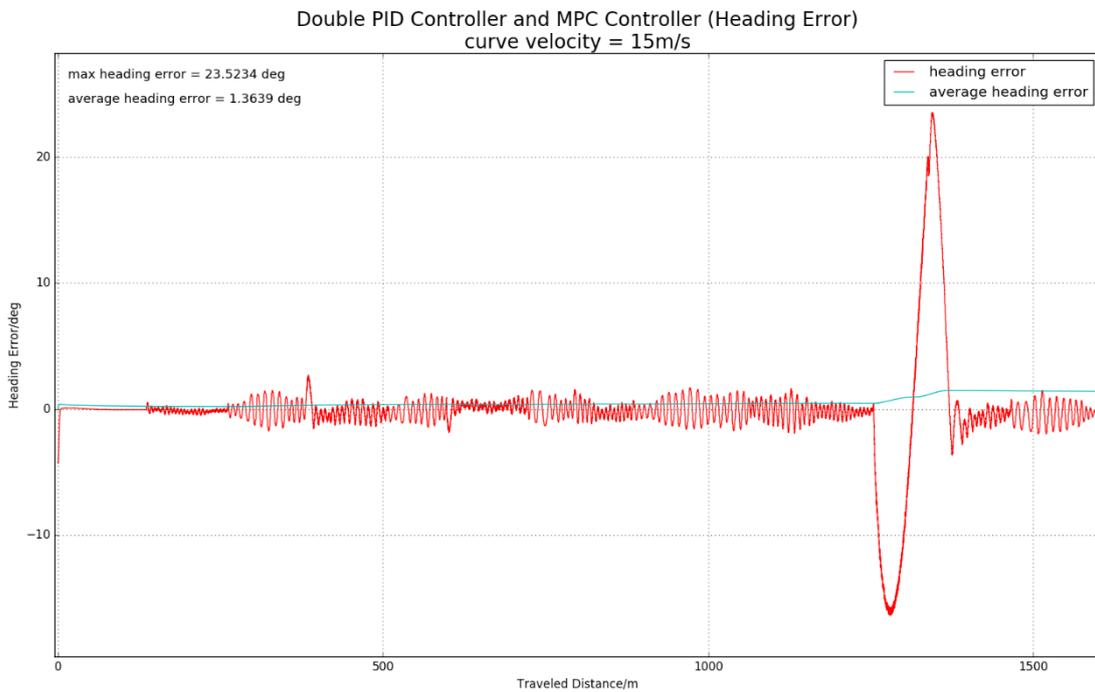


Figure 5.48 Heading Error of MPC at V=15 m/s

5.6 Simulation Result Comparison

Since the velocity limitation in mining area is $30\text{km/h} = 8.33\text{m/s}$, so the velocity of test trajectory is 10m/s . In order to figure out the potential of control algorithms, the trajectory with $V=15\text{m/s}$ is also tested.

The following figures show the comparison of the performance of various control methods at different velocity.

The table of error data are shown in Figure 5.49.

		Stanley	Pure Pursuit	RWF	LQR	MPC
10	Max Velocity Error (m/s)	0.716	0.715	0.734	0.675	0.673
	Average Velocity Error (m/s)	0.045	0.046	0.047	0.047	0.046
	Max Lateral Distance Error (cm)	-10.967	42.558	76.694	37.142	15.368
	Average Lateral Distance Error (cm)	2.596	4.274	4.367	2.939	1.490
	Max Heading Error (deg)	-3.886	-5.133	-7.395	-6.272	-3.626
	Average Heading Error (deg)	0.167	0.241	0.271	0.343	0.186
	Max Computational Time (ms)	12.600	14.000	14.200	16.000	15.800
	Average Computational Time (ms)	6.290	6.124	6.331	6.333	6.655
15	Max Velocity Error (m/s)	4.427	6.432	4.424	3.522	3.379
	Average Velocity Error (m/s)	0.282	0.192	0.316	0.234	0.221
	Max Lateral Distance Error (cm)	1352.670	-4096.130	1357.260	1343.090	1155.410
	Average Lateral Distance Error (cm)	79.994	270.145	75.067	70.171	56.836
	Max Heading Error (deg)	41.122	61.952	35.903	27.052	23.523
	Average Heading Error (deg)	1.853	4.166	1.966	1.485	1.364
	Max Computational Time (ms)	16.000	13.000	14.000	14.000	15.600
	Average Computational Time (ms)	6.133	6.182	6.252	6.080	6.211

Figure 5.49 Error Data Comparison of Different Controller

Figure 5.50 shows the comparison of reference and actual path with velocity equal to 10m/s .

In the case of $V=10\text{m/s}$, all the controllers can track reference path well.

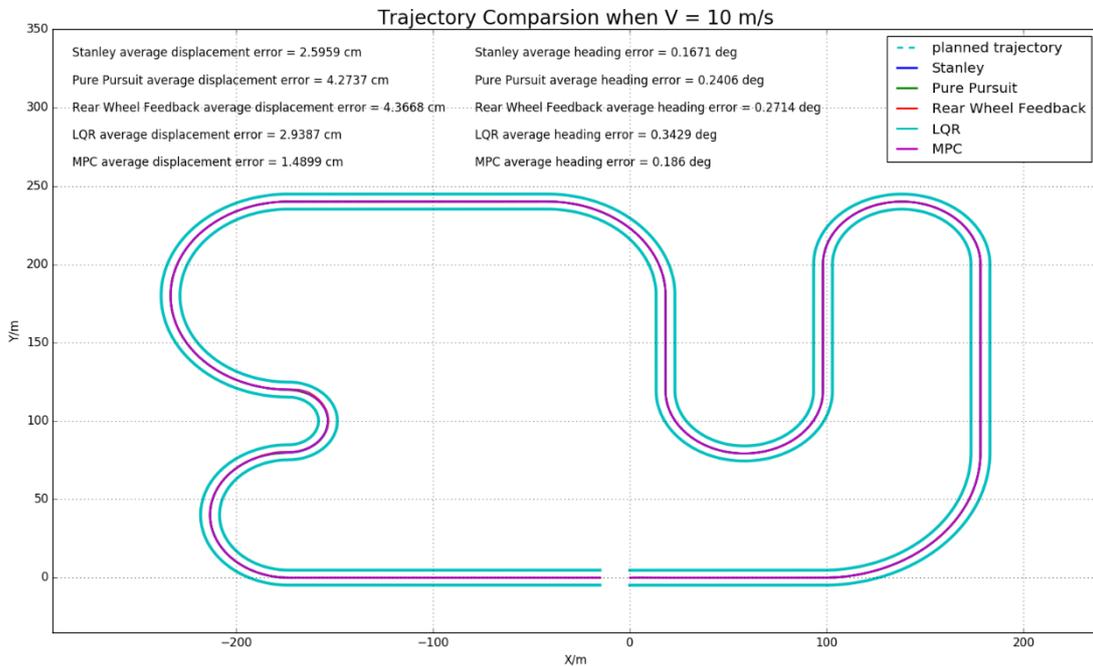


Figure 5.50 Path Tracking Comparison at $V=10\text{m/s}$

Figure 5.51 and 5.52 show the detailed error data.

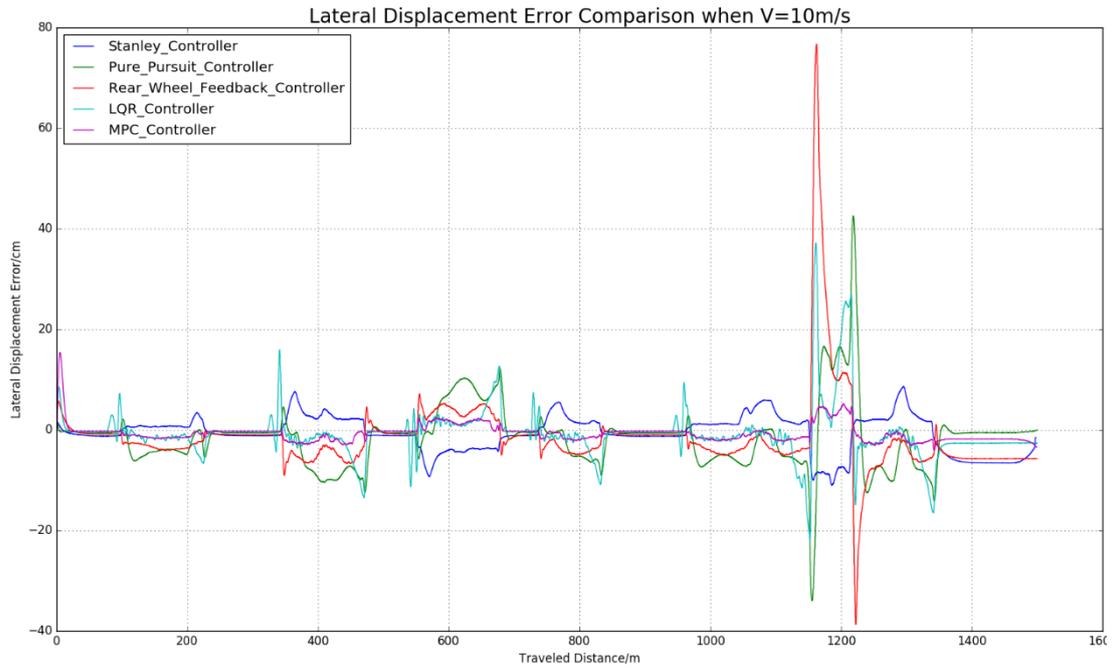


Figure 5.51 Lateral Displacement Error Comparison at V=10m/s

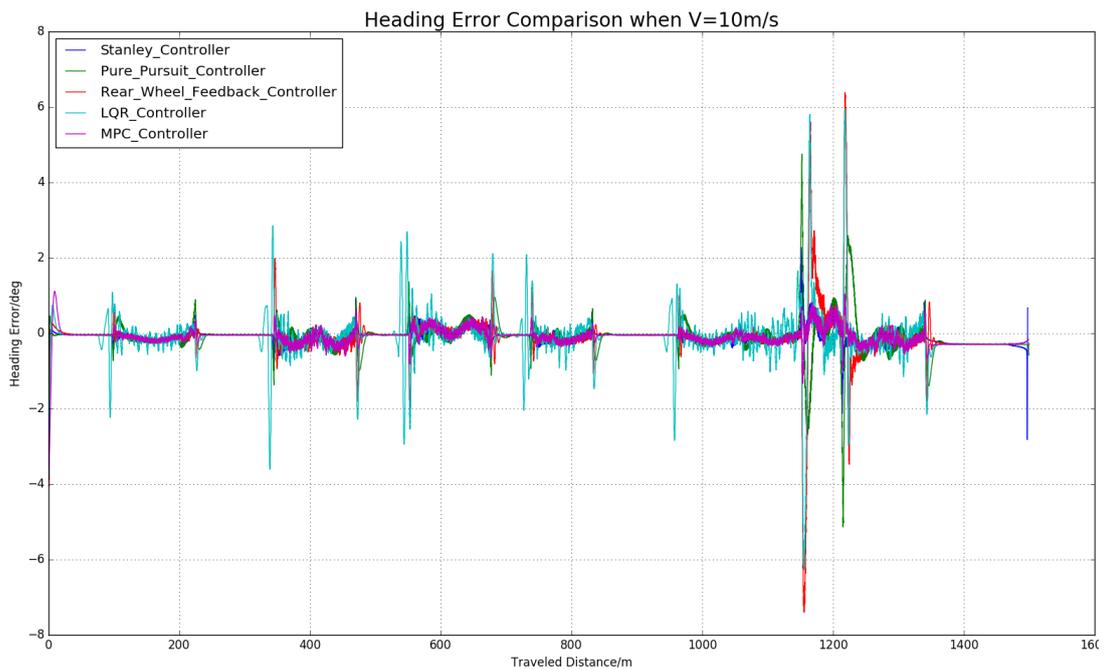


Figure 5.52 Heading Error Comparison at V=10m/s

As shown in the comparison plots of lateral displacement error and heading error, the largest error occurs at the bend where R is equal to 20m.

The maximum average lateral displacement error is 4.367cm (Rear-Wheel Feedback), and the maximum average heading angle error is 0.343 degree (LQR).

The MPC Controller has the best performance. The average trajectory tracking error is about 1.49cm, the maximum error is 15.36cm, the average heading error is 0.186 degree, and the maximum error is only 3.626 degree.

The overall performance of the Stanley Controller is also very good, with an average tracking error of 2.59cm and a maximum deviation of 10.96cm. This is because the vehicle model used has a front wheel steering mechanism, and Stanley just uses the front wheel axle center and the target point for error analysis, which leads to faster response.

The tracking error of the LQR Controller is also really small, but every time the road curvature changes, a large error will occur, which decreases overall performance.

Although the overall data of the Rear-Wheel Feedback Controller and the Pure Pursuit Controller are relatively poor, the main error appears in the bend with the radius of 20m, and the error in other places is very small. It also reflects that the tracking ability of these two controllers are relatively weak in the case of large lateral acceleration.

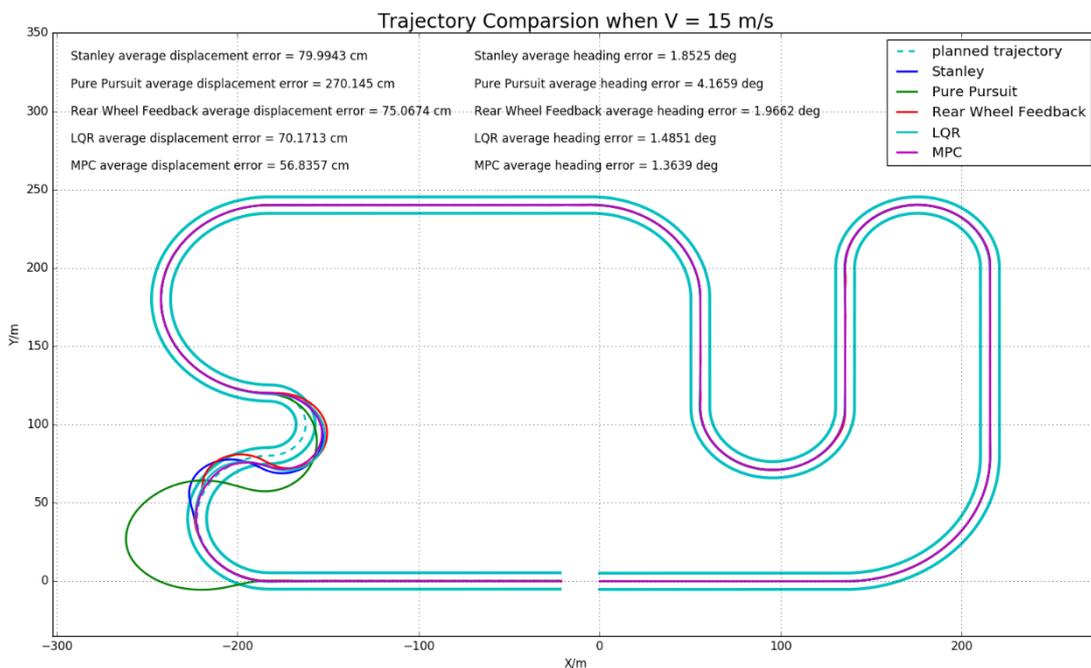


Figure 5.53 Path Tracking Comparison at $V=15\text{m/s}$

Figure 5.53 illustrates the comparison of reference and actual path with velocity equal to 15m/s.

All control algorithms are unable to track the curve with the radius of 20m precisely.

MPC and LQR controllers can response faster than other controllers.

The error of lateral displacement and heading angle are also plotted in Figure 5.54 and 5.55.

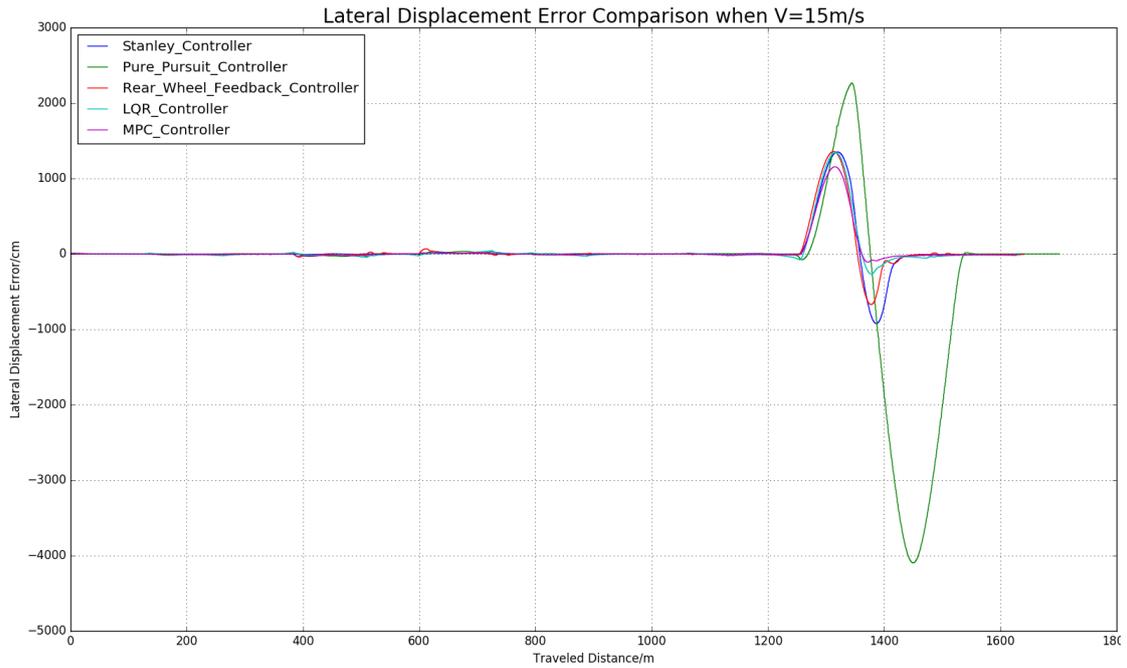


Figure 5.54 Lateral Displacement Error Comparison at V=15m/s

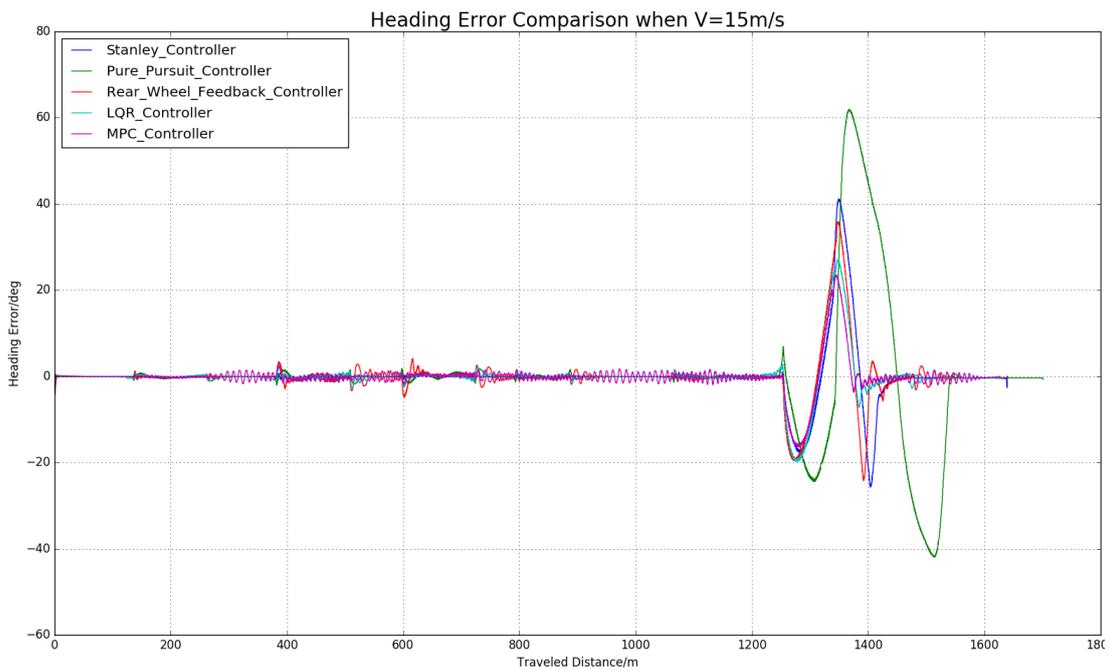


Figure 5.55 Heading Error Comparison at V=15m/s

MPC Controller is still the best one among them, which can quickly return to the reference trajectory smoothly. The average trajectory tracking error is 56.836cm, and the average heading angle error is 1.364 degree.

LQR Controller also has good result in this test, the average trajectory tracking error is 70.171cm, and the average heading angle error is 1.485 degree.

The tracking error of Stanley Controller, the Rear-Wheel Feedback Controller are higher slightly than LQR.

The Pure Pursuit Controller has the worst effect in this test.

6. Conclusions and Future Work

6.1 Conclusions

About longitudinal controller design, the dual closed-loop PID control of speed and acceleration can track the reference trajectory well under the velocity equals to 10m/s and 15m/s.

In the lateral controller part, the MPC Controller has the best ability to track the planned trajectory in two different velocity conditions. The performance of LQR Controller is reduced slightly with the increasing velocity.

Stanley Controller, Rear-Wheel Feedback Controller and the Pure Pursuit Controller are more suitable for the lateral control in low velocity.

6.2 Future Work

In summary, most of the current motion control researches is conducted in relatively simple conditions, and therefore rarely consider the nonlinear characteristics of tires, the coupling characteristics of the vehicle under large lateral acceleration, and the influence of tire force constraints. In addition, there are time-varying uncertainties in the road gradient, vehicle quality, tire-road adhesion coefficient and other parameters in the vehicle longitudinal dynamic model, which is an important challenge for the longitudinal motion control design.

Based on this, further research on longitudinal motion control can be carried out around the following two aspects.

On the one hand, consider the mechanism of the vehicle's nonlinear factors and coupling characteristics on the longitudinal control under extreme conditions, and establish a longitudinal and lateral tire force coordination mechanism to ensure the stability of the vehicle when controlling the longitudinal movement.

On the other hand, in automated vehicles, multi-source sensor information fusion technology is used to estimate the parameters of the environment, and a closed-loop system composed of sensors and controllers is constructed, which makes the motion control system running in a complex dynamic environment has better adaptive capability and control accuracy.

As for lateral control, for conventional working conditions with small lateral acceleration, simple geometric/kinematic models or linear dynamics models can usually meet the requirements. In extreme conditions like large lateral acceleration, the inaccuracy of the model may lead to deterioration of tracking performance. Therefore, it is necessary to consider factors such as tire nonlinearity, slip, roll/pitch motion, and actuator dynamic characteristics to establish high fidelity nonlinear dynamics model. However, the increase in the complexity of the model will inevitably bring about the increase in the difficulty of designing the control law and the increase in the amount of

calculation. How to construct a mathematical model with high fidelity for extreme conditions and easy to control realization needs further research.

Beside there is no sufficient consideration about the complex working conditions such as discontinuous curvature and sudden changes in the road surface. The applicability and robustness still lack sufficient verification.

References

- [1] L. L. Shaoshan Liu, "Creating Autonomous Vehicle Systems," *A Publication in the Morgan & Claypool Publishers series SYNTHESIS LECTURES ON COMPUTER SCIENCE*, #9, 2018.
- [2] S. Choi, B. d'Andrea-Novell, M. Fliess, H. Mounier, and J. Villagra, "Model-free control of automotive engine and brake for Stop-and-Go scenarios," in 2009 European Control Conference (ECC), 2009.
- [3] P. Shakouri, A. Ordys, D. S. Laila, and M. Askari, "Adaptive Cruise Control System: Comparing Gain-Scheduling PI and LQ Controllers," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 12964-12969, 2011/01, 2011.
- [4] S. Xu, H. Peng, Z. Song, K. Chen, and Y. Tang, "Accurate and Smooth Speed Control for an Autonomous Vehicle," in 2018 IEEE Intelligent Vehicles Symposium (IV), 2018.
- [5] J. C. Gerdes, and J. K. Hedrick, "Vehicle speed and spacing control via coordinated throttle and brake actuation," *Control Engineering Practice*, vol. 5, no. 11, pp. 1607-1614, 1997/11, 1997.
- [6] A. Ferrara, and C. Vecchio, "Second order sliding mode control of vehicles with distributed collision avoidance capabilities," *Mechatronics*, vol. 19, no. 4, pp. 471-477, 2009/06, 2009.
- [7] P. Hang, X. Chen, B. Zhang, and T. Tang, "Longitudinal Velocity Tracking Control of a 4WID Electric Vehicle," *IFAC-PapersOnLine*, vol. 51, no. 31, pp. 790-795, 2018.
- [8] H. Nam, W. Choi, and C. Ahn, "Model Predictive Control for Evasive Steering of an Autonomous Vehicle," *International Journal of Automotive Technology*, vol. 20, no. 5, pp. 1033-1042, 2019/08/10, 2019.
- [9] J. E. Naranjo, C. Gonzalez, R. Garcia, and T. de Pedro, "Cooperative Throttle and Brake Fuzzy Control for ACC\$+\$ Stop&Go Maneuvers," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 4, pp. 1623-1630, 2007/07, 2007.
- [10] C. R. C, "Implementation of the pure pursuit path tracking algorithm," *Robotics Institute Carnegie Mellon University Pittsburgh, Pennsylvania*, 1992.
- [11] J. M. Snider, "Automatic Steering Methods for Autonomous Automobile Path Tracking," *Robotics Institute Carnegie Mellon University Pittsburgh, Pennsylvania*, 2009.
- [12] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney, "Stanley: The robot that won the DARPA Grand Challenge," *Journal of Field Robotics*, vol. 23, no. 9, pp. 661-692, 2006.
- [13] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo, and S. Thrun, "Autonomous Automobile Trajectory Tracking for Off-Road Driving: Controller Design,

- Experimental Validation and Racing,” in 2007 American Control Conference, 2007.
- [14] Z. X. C. Huiyan, “A study on lateral control method for the path tracking of intelligent vehicles,” *Automotive Engineering*, 2011.
- [15] R. Marino, S. Scalzi, and M. Netto, “Nested PID steering control for lane keeping in autonomous vehicles,” *Control Engineering Practice*, vol. 19, no. 12, pp. 1459-1467, 2011/12, 2011.
- [16] S. Xu, and H. Peng, “Design, Analysis, and Experiments of Preview Path Tracking Control for Autonomous Vehicles,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 48-58, 2020/01, 2020.
- [17] A. Goodarzi, A. Sabooteh, and E. Esmailzadeh, “Automatic path control based on integrated steering and external yaw-moment control,” *Proceedings of the Institution of Mechanical Engineers, Part K: Journal of Multi-body Dynamics*, vol. 222, no. 2, pp. 189-200, 2008/06, 2008.
- [18] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, “Predictive Active Steering Control for Autonomous Vehicle Systems,” *IEEE Transactions on Control Systems Technology*, vol. 15, no. 3, pp. 566-580, 2007/05, 2007.
- [19] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat, “Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation,” *International Journal of Robust and Nonlinear Control*, vol. 18, no. 8, pp. 862-875, 2008.
- [20] L. E. H. M. H. J. K., “Lateral and longitudinal vehicle control coupling for automated vehicle operation,” in Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251), 1999.
- [21] H. Lee, and M. Tomizuka, “Coordinated Longitudinal and Lateral Motion Control of Vehicles for IVHS,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 123, no. 3, pp. 535-543, 1998/03/24, 1998.
- [22] J. Guo, Y. Luo, and K. Li, “Adaptive coordinated collision avoidance control of autonomous ground vehicles,” *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 232, no. 9, pp. 1120-1133, 2018/05/22, 2018.
- [23] S. Kumarawadu, and L. Tsu-Tian, “Neuroadaptive Output Tracking of Fully Autonomous Road Vehicles With an Observer,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 335-345, 2009/06, 2009.
- [24] S. C. Peters, E. Frazzoli, and K. Iagnemma, “Differential flatness of a front-steered vehicle with tire force control,” in 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2011.
- [25] S. Fergani, L. Menhour, O. Sename, L. Dugard, and B. D'Andréa-Novel, “Integrated vehicle control through the coordination of longitudinal/lateral and vertical dynamics controllers: Flatness and LPV/ H_∞ -based design,” *International Journal of Robust and Nonlinear Control*, vol. 27, no. 18, pp. 4992-5007, 2017/05/25, 2017.
- [26] G. Y., “Model Predictive Control for Autonomous and Semiautonomous

- Vehicles,” *San Francisco: University of California, Berkeley*, 2014.
- [27] Y. Gao, T. Lin, F. Borrelli, E. Tseng, and D. Hrovat, “Predictive Control of Autonomous Ground Vehicles With Obstacle Avoidance on Slippery Roads,” in *ASME 2010 Dynamic Systems and Control Conference*, Volume 1, 2010.
- [28] J. K. H. Lu X, “longitudinal control design and experiment for heavyduty trucks,” *Proc Am Control Conf, Colorado*, pp. 36-41, 2003.
- [29] R. Rajamani, “Vehicle Dynamics and Control 2ed.pdf,” 2012.
- [30] J. H. a. Y. D. K. K Yi*, “A vehicle control algorithm for stop-and-go cruise control,” *Proc Ins Mech Eng D-J Automobile Eng*, pp. 1099-1115, 2001.
- [31] L. Bainan, “Research on longitudinal speed control for autonomous vehicles adaptive with road grade and vehicle mass,” *Changchun: Jilin University*, 2015.
- [32] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, “A stable tracking control method for an autonomous mobile robot,” in *Proceedings., IEEE International Conference on Robotics and Automation*.
- [33] J. Liu, P. Jayakumar, J. L. Stein, and T. Ersal, “A study on model fidelity for model predictive control-based obstacle avoidance in high-speed autonomous ground vehicles,” *Vehicle System Dynamics*, vol. 54, no. 11, pp. 1629-1650, 2016/08/31, 2016.
- [34] M. T. Huei Peng, “Vehicle Lateral Control for Highway Automation,” *1990 American Control Conference*, pp. 788 – 794, 1990.
- [35] P. E. N. Gwanghun Gim, “An Analytical Model of Pneumatic Tyres for Vehicle Dynamics Simulations,” *International Journal of Vehicle Design*, pp. 589 – 618, 1990/11, 1991.
- [36] B. Paden, M. Cap, S. Z. Yong, D. Yershov, and E. Frazzoli, “A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles,” *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33-55, 2016/03, 2016.
- [37] K. Kritayakirana, and J. C. Gerdes, “Using the centre of percussion to design a steering controller for an autonomous race car,” *Vehicle System Dynamics*, vol. 50, no. sup1, pp. 33-51, 2012/01, 2012.
- [38] M. T. Huei Peng, “Preview control for vehicle lateral guidance in highway automation,” *1991 American Control Conference (ACC)*, pp. 3084- 3089, 1991.
- [39] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, “A Predictive Controller for Autonomous Vehicle Path Tracking,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 92-102, 2009/03, 2009.
- [40] J. C. G. Kritayakirana K, “Controlling an Autonomous Racing Vehicle Using Feedforward and Feedback to Control Steering and Speed,” *Proceedings of the*

ASME 2009 Dynamic Systems and Control Conference, pp. 1-8, 2009.