Tracking Control of Autonomous Vehicles

Tianqi Yang

A Thesis

in

The Department

of

Mechanical, Industrial & Aerospace Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Mechanical Engineering) at

Concordia University

Montréal, Québec, Canada

September 2022

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Tianqi Yang**

Entitled: **Tracking Control of Autonomous Vehicles**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Mechanical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. Wenfang Xie

_____ External Examiner
Dr. Jun Yan

_____ Examiner
Dr. Wenfang Xie

_____ Supervisor
Dr. Youmin Zhang

Approved by _____
Martin D. Pugh, Chair
Department of Mechanical, Industrial & Aerospace Engineering

August 31, 2022
Date of Defense

_____
Dr. Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Tracking Control of Autonomous Vehicles

Tianqi Yang

This thesis intends to design new tracking schemes to enhance the performance and stability of general autonomous vehicles (AVs). Three main types of controllers used for tracking control are investigated.

The geometric controller cannot meet high tracking requirements, and control parameters significantly affect its performance. Therefore, an observer-based nonlinear control combined with a particle swarm optimization (PSO) algorithm is developed for low-speed vehicles to track the pre-determined trajectory accurately. A control law featured with self-tuning gains is designed using the backstepping control technique, for which global asymptotic stability is validated. The PSO evaluates tracking performance through the proposed fitness function and generates optimized tuning parameters with fewer iterations, reducing tuning efforts. Velocity and steering tracking could also be rapidly realized by modifying the error weights of the performance evaluation criterion. Based on the proposed yaw error observer (YEO), the problem of the angle measurements being temporarily inaccurate or unavailable is tackled effectively with the given information.

Further, existing methods can suffer from complex control algorithms and a lack of tracking stability at high speed. The vehicle's motion is decoupled by considering the Frenet frame. A lateral control law based on the linear-quadratic-regulator (LQR) imposes the tracking errors to converge to zero stably and quickly, providing the optimal solution

in real-time due to adaptive gains. Regarding the steady-state errors, they are eliminated through the correction of the feedforward term. Besides, the designed double proportional-integral-derivative (PID) controller realizes not only the longitudinal control but also the velocity tracking.

Acknowledgments

Words cannot express my gratitude to my supervisor Prof. Youmin Zhang for his continuous encouragement, invaluable patience, and insightful guidance throughout my master's study. I could not have accomplished this thesis research work without my supervisor, who also generously provided knowledge and expertise. Moreover, this endeavor would not have been possible without the generous support from my supervisor, who financed my research.

I would like to extend my sincere thanks to my two fellow lab mates, Juqi Hu and Linhan Qiao, for their help, suggestions, and cooperation on my control algorithm design and experimental tests. Thanks should also go to other laboratory members, namely Shun Li, Qiaomeng Qin, Jin Li, Xiaobo Wu, and Yufei Fu, who inspired and impacted me through the most difficult time.

Lastly, I would be remiss in not mentioning my parents. Their support and belief in me have kept my motivation and spirits high during my master's program.

Table of Contents

List of Figures

Chapter 1

Introduction and Literature Review

1.1 Tracking Control in Autonomous Vehicles

Car-like vehicles, characterized by excellent adaptability and high flexibility, have become more commonplace both in military and commercial domains with the rapid development of sensing and computing technologies [1, 2]. The car-like vehicle has become increasingly popular in academia and industry because it is a handy and applicable tool for verifying localization, perception, planning, prediction, control, artificial intelligence, machine learning, and other advanced self-driving concepts [3, 4]. It can be broadly composed of four main categories, namely a controller, power system, sensors, and actuators, of which the controller subsystem as a critical component enables the vehicle executes control commands in order to track a pre-defined trajectory provided from the trajectory planning stably. According to [5–7], under nonholonomic constraints, no doubt that utilizing different controller, control laws, or control gains may exhibit remarkable differences in tracking performance and results. Existing gain tuning methods can lead to some shortcomings for the vehicle, such as slow response, poor transient, robustness, etc. Control gains are supposed to upgrade automatically online for fitting in the current tracking state to get optimal tracking. Nonetheless, the adaptive control gains have not yet been widely implemented.

Furthermore, without precise positioning, the data from sensors may disable the tracking. In practice, there are indeed some inevitable errors between sensor-based and real value, especially when only one type of positioning sensor is available. Motivations for using an observer to estimate unknown information are given in [8–10], where the observer-based controllers are able to compensate for disturbances and uncertainties. However, few studies have considered the situation when the yaw angle of car-like vehicles is temporarily unmeasurable.

In retrospect, although vehicle trajectory tracking has been a mature field emerging with a plethora of control methodologies, it remains hard to achieve a trade-off between tracking performance and stability. Another challenge stems from how to reach a compromise between control complexity and modeling error since the control law is derived from the vehicle's mathematical model, such as geometric, kinematic, and dynamic configuration. Some valuable properties and assessments of different tracking controllers are discussed in [11–14]. Two prevalent geometric controllers, namely Pure Pursuit (PP) and Stanley, are easy to implement due to simple configurations and less needed parameters [15,16]. Nevertheless, the unsystematic parameter tuning method hinders the reliability and efficiency of the algorithm, leading to cutting-corner behavior or oscillations during tracking [17–19]. Lee et al. [20] proposed a dynamic-based controller with an adaptive regulator that provides better tracking performance in decreasing lateral and heading offset than the PP and Stanley method. Despite increasing the robustness as vehicle speed changes, the relevant parameters and equations in adaptive design are selected manually. The LQR control is widespread in tracking control because it optimally provides feedback gains for the entire time domain to ensure better tracking performance [21]. To optimally tune the gains of the nonlinear controller, Alcala et al. [22] reformulated the closed-loop system in a linear parameter varying form. Despite presenting satisfactory results by a real test, the heuristic algorithm can only ensure a locally optimal solution instead of a global counterpart on a

kinematic level. It is noted that implementing online tuning for LQR is not straightforward due to the offline determination of optimal gains, and the linear structure of LQR imposes restrictions like linearization approximations on the utilized model. Recent research indicates a Frenet frame which enables the vehicle's motion to be decoupled into the longitudinal and lateral direction, simplifying tracking control complexity [23]. By incorporating LQR and PID control, trajectory tracking is achieved without the heavy control law's design work. The former, generally integrated with the feedforward control, regulates the steering and the latter for the velocity. The classical tracking controllers such as PID and sliding model controllers have the virtue of robustness and simpleness, whereas their parameters are not readily to tune [24–26]. To remedy this problem, the PSO based controllers were built [27–29]. The PSO, having a flexible and well-balanced mechanism, is extensively employed to optimize various controller parameters to enhance the tracking performance. The supervised controller tuned by PSO offered a slight deviation during the tracking, which was verified by a wheel mobile robot in [30]. Amer et al. [31] developed a PSO-tuned Stanley controller based on a sophisticated vehicle model which decreased lateral cross tracking error and acceleration. Although these PSO-tuned controllers reduce the tuning efforts and time, they are suffered from the problem of the extended computing time spent and premature convergence. Further, only is it reasonable to get optimal tuning values considering the complete state of the utilized vehicle, as well as the error posture dynamics. In contrast, the aforementioned studies did not take these elements into account. Dai et al. [32] explored a PSO-tuned backstepping controller to minimize posture errors. The simulation results revealed apparent oscillations, given that the optimized parameters as a fixed form may be inappropriate for the controller during the tracking period, and the fitness function has not associated with the control inputs of the kinematic model. Up till now, few studies have sought to link PSO with adaptive control gains.

Apart from the studies introduced above, there have been numerous notable controllers.

Using the Lyapunov theorem to prove the global asymptotic stability of the tracking system was initially raised and verified on a non-holonomic vehicle by Kanayama et al. [7]. Through this means, massive nonlinear backstepping tracking controllers have emerged, refer to [33–36]. However, the choice of control gains or parameters poses an intractable matter which would be detrimental to the development and stability verification of the control law. Hu [37] proposed an adaptive backstepping controller with the consideration of input constraints to realize an accurate tracking of the car-like vehicle, where partial control gains were automatically adjusted. When slipping occurs, pure rolling constraints are no longer satisfied. Thereby, a robust adaptive controller which integrated the sliding effects as extra unknown arguments was constructed against deterioration of smooth movement and guidance [38]. Because of fast decision ability, fuzzy sets and neural networks (NN) are also applied to adaptive controllers. A fuzzy logic control strategy for trajectory tracking was presented in [39], in which the tracking errors were redefined and encoded simply. In [40], the position's convergence rate of the car-like robot running two NNs was speeded up while gains were upgraded online. Nonetheless, these two kinds of controllers need to be programmed or trained more to make suitable decisions with their expertise to improve tracking performance. Traditional control methods through an offline manner lead to a fixed control law which may not be suitable for all driving scenarios or the whole tracking time. In contrast, the basic idea of model predictive controller (MPC) is to solve the issue of online open-loop optimization and obtain the optimal solution for tracking. An additional advantage of MPC is that it can add constraints to the future input, output, and state variables of the tracking system, which complies with the requirements in real applications [41–43]. However, MPC would lead to a more complex control algorithm, adding the computing burden [44].

Furthermore, observer-based controllers are becoming increasingly popular in trajectory tracking control because uncertainties, disturbances, modeling errors, sensor faults,

and noise are inevitable in real-time tracking control [45–48]. Cui [49] designed an improved linear extended state observer integrating with a sliding mode controller to estimate the total uncertainties of the differential-driving mobile robots. The experimental results show that the proposed method significantly compensates for external disturbance and parameter perturbation. Huang et al. [50] regarded unknown parameters in the dynamic model and external disturbance as a lumped disturbance and thus designed an observer-based adaptive torque controller to ensure a feedforward compensation. The simulation results indicate that the wheeled mobile robot employing this control scheme can converge to the intended trajectory with zero steady-state error. Zou [51] proposed an output feedback scheme together with an observer and estimator to compress the effects resulting from sensor faults and realize formation control with uniformly ultimately bounded stability.

Autonomous vehicles, also known as self-driving cars, have become an attainable reality because of the rapid development of sensors, high precision maps, computing technologies, etc [11, 52]. The global Cartesian coordinate frame is commonly used to describe the position of the vehicle. Nevertheless, this coordinate frame does not facilitate trajectory planning and tracking, given that it is tough to know how far the vehicle has driven and whether it deviates from the center of the lane. By contrast, the Frenet frame, as an alternative method, can use the position based on the road center lane to determine the longitudinal displacement along the road and the lateral offset perpendicular to the road center [53, 54]. Another merit is that the motions of the vehicle based on the dynamic model are able to be decoupled in the Frenet frame, which signifies that the control difficulties are remarkably reduced [55]. Furthermore, there are numerous path and trajectory planning methods by means of the Frenet frame to optimize trajectory smoothness and promote passenger comfort [56–58]. The Frenet-based method can treat the trajectory planning problem as a quadratic program, meeting the real-time and computational requirements [59].

In retrospect, Duan et al. [23] implemented the vehicle trajectory tracking by utilizing

the LQR and MPC, where the tracking errors were defined in the Frenet frame. However, the PID control offers apparent delays and unstable velocity tracking for the actual vehicle. The oscillations emerging from lateral and orientational errors may also deteriorate the system stability and even automatic guidance performance. For the lateral control of the vehicle, extensive literature cooperated LQR with the feedforward control for the purpose of acquiring superior steering performance [60–63]. Even though this combined control approach ensures that the steady-state error is removed, the optimal solutions are only derived in an offline manner [64]. The offline gains lead to fixed control law, which is unfavorable to all the driving scenarios. To overcome this issue, Lee et al. [20] investigated an adaptive regulator on the basis of a dynamic bicycle model. In this way, the self-tuning gains could be updated online to boost the tracking accuracy without overshooting. Still, excessive control parameters need to be determined, adding the control uncertainties and human intervention. Moreover, some valuable assessments and properties of multiple tracking controllers are discussed in [5, 14, 21, 65], from which optimized tracking results are presented. However, it is still hard to tackle the problem between tracking performance and system stability. Another challenge results from the selection of the weight of Q and R.

1.2 Scope and Objectives of the Thesis Research

Based on the review of the reported studies, it is clear that the developments in the control module for car-like vehicles and AVs can suffer from undesirable transient and steady-state performance, as well as the unreasonable tuning of control parameters, which directly results in the effectiveness, applicability, and practicability of the designed controller. The scope of this thesis research mainly concentrates on designing and developing trajectory tracking strategies to enhance the tracking performance of the vehicle. In this work, specific research objectives can be summed up below:

(1)Deploy the geometric-based tracking control algorithm (PP and Stanley) into the practical application and observe how different control parameters affect tracking performance under different driving scenarios and driving speed.

(2)Develop a backstepping control law on a kinematic level for low-speed vehicles so as to enhance the tracking performance and system stability.

(3)Develop a yaw error observer for low-speed vehicles to resolve the problem of sensor faults.

(4)Design a systematic way to optimize the control parameters, aiming at reducing human intervention and tuning efforts.

(5)Design a new performance evaluation criterion for the tracking performance of AVs.

(6)Decouple the motion of AVs based on a dynamic vehicle model with the aim of decreasing control complexity.

(7)Develop a dynamic-based controller to ensure steering tracking while designing a longitudinal tracking controller to guarantee forward speed tracking for wide-speed range AVs.

1.3 Thesis Layout

This thesis is prepared based on the manuscript-based format described in the "Thesis Preparation, Examination and Regulation" guidelines of the School of Graduate Studies, Concordia University. The thesis research is organized into 6 chapters, solving the research objectives mentioned above and the problems in the introduction and literature review (Chapter 1). Chapter 2 and Chapter 3 demonstrate geometric-based controllers, namely PP and Stanley strategies, used for path tracking control. Chapter 4 presents a kinematic-based

controller for low-speed car-like vehicles to realize trajectory tracking control. Chapter 5 describes a dynamic-based controller for wide-speed range AVs to achieve trajectory tracking. The major contributions and conclusions of this thesis research are summarized in Chapter 6 together with recommendations for future research direction.

Chapter 4 presents the following article:

- Tianqi Yang, Youmin Zhang, and Juqi Hu, "Observer-based Adaptive Trajectory Tracking Control of Car-Like Vehicles Using PSO," *IEEE Transactions on Industrial Electronics*. (To be submitted).

This study proceeds by first giving the problem statements in Section 4.1, including the kinematic bicycle model, reference trajectory, error dynamics, and control gains. Next, the detailed control strategy will be introduced in Section 4.2; a backstepping control law based on YEO and PSO is derived. Global asymptotic stability is proven for the backstepping control law, as well as YEO. Section 4.3 presents the simulation results. Comprehensive comparisons and discussions of the experimental results are offered in Section 4.4.

Chapter 5 presents the following article:

- Tianqi Yang, Juqi Hu, and Youmin Zhang, "Trajectory Tracking for Autonomous Vehicles based on Frenet Frame," in *37th Youth Academic Annual Conference of Chinese Association of Automation (YAC'22)*, May 27-29, 2022, Beijing, China.

This study investigates a closed-loop control system including the LQR, double PID, and feedforward control to realize automatic tracking. A detailed dynamic bicycle model, coordinate transformation, and error dynamics will be introduced in Section 5.1. The planning of the reference trajectory will be presented in Section 5.2. Section 5.3 contains the procedure of the tracking control strategy, followed by the comprehensive simulation results in Section 5.4.

Chapter 2

Tracking Control based on Pure Pursuit Method

2.1 Introduction

The PP is an important precedent and the most basic control strategy for solving the tracking control problem due to its simplicity and practicality. By this geometric path tracker, the computed steering commands are able to drive the control target from its current position to the look-ahead point ahead of the vehicle. With respect to the velocity command, it could be regarded as a constant that can be changed at any waypoint. These two control commands are calculated according to the vehicle's pose (position and orientation). In order to comprehend this tracking method, we can imagine that the vehicle constantly pursues a moving goal point (look-ahead point) based on its current position until the last point on the reference path.

There are several noteworthy tips before we explain how the PP algorithm enables the vehicle to achieve tracking control. The PP controller acts only for the path tracking purpose in comparison to traditional controllers, and this controller is unique to the predefined path. Additionally, a critical precondition for successful tracking control is that the

path should be designed reasonably, ensuring acceptable steering maneuvers and reducing the oscillatory nature of the PP algorithm. By way of illustration, waypoint paths should provide a series of tightly spaced waypoints, whose position also needs to be restricted to satisfy approximate curvature constraints so as to generate a feasible path. According to the specification of the used vehicle, the desired linear velocity and the saturation value of the steering angle should also be taken into consideration. In this chapter, the controlled vehicle is considered to be the absence of side slip.

2.2 Control Algorithm

The schematic of the PP method and the kinematic model are shown in Figure 2.1. The core idea of this algorithm is to match a look-ahead point on the reference path. The length between the look-ahead point C and the central control point A (center of the rear wheel) is called the look-ahead distance L_d, which indicates that the vehicle how far along on the reference path to track towards. In this case, the center of the rear wheel of the vehicle will achieve the goal point C with the specific turning radius R. Note that L_d is the main tuning parameter for the PP tracker. Two reference coordinate frames, namely the global Cartesian coordinate frame and vehicle body counterpart, are built with a view to obtaining the vehicle's posture and deriving control law solutions. Hence, based on the geometric relationships of the triangle OAC and OAB, the steering angle δ_c can be attained.

In order to enable the rear wheel to precisely track the path (red-dotted line) to the point C, applying the law of sines to Figure 2.1 results in:

$$\frac{L_d}{R} = \frac{\sin 2\alpha}{\sin(\pi/2 - \alpha)} \tag{2–1}$$

By simplifying, we have:

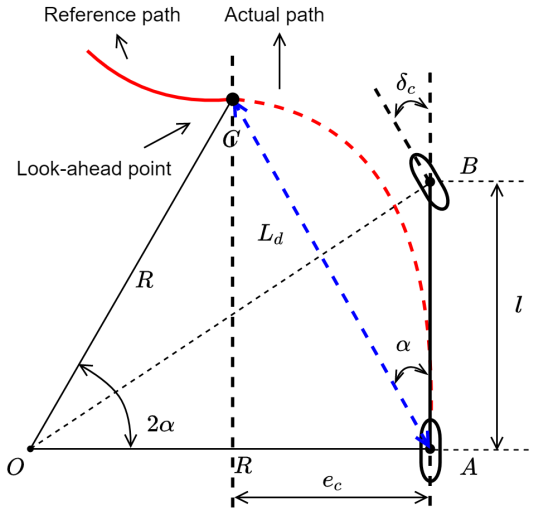


Figure 2.1: PP method.

$$R = \frac{L_d}{2\sin\alpha} \tag{2–2}$$

With the Ackerman steering configuration, the simple mathematical equation describes the geometric relations between the front wheel steering angle and the turning radius, which can be formulated as:

$$R = \frac{l}{\tan\delta_c} \tag{2–3}$$

Therefore, substituting (2–3) into (2–2) yields:

$$\delta_c(t) = \arctan\frac{2L\sin(\alpha(t))}{L_d} \tag{2–4}$$

Next, the cross tracking error is defined as follows:

$$e_c = L_d\sin\alpha \tag{2–5}$$

Using the small angle approximation together with (2–4) and (2–5), the expression for e_c can be rewritten as:

$$e_c \approx \frac{{L_d}^2}{2L}\delta(t) \tag{2–6}$$

Once the value of the steering command remains stable, it can be identified that cross tracking errors will be directly determined by the look-ahead distance. Consequently, the PP is essentially a proportional controller, and the intuition convinces one that tracking performance depends on the selection of the L_d. With the purpose of enhancing the tracking performance, the varying look-ahead distance is defined as a first-order polynomial of the vehicle's velocity, as:

$$L_d = k_v V + L_{dl} \tag{2–7}$$

where k_v and L_{dl} are called the coefficient of look-ahead distance and the minimum threshold of the look-ahead distance, respectively. A better understanding of the property of the look-ahead distance can be gained by referring to Figure 2.2. It should be stressed that the actual trajectory commonly does not match the straight line between waypoints due to the nature of the PP tracker.

The look-ahead distance is a critical factor for the tracking performance and results, thus rendering different tracking behaviors, as seen in Figure 2.2. It can be summarized as follows: maintaining the path and regaining the path. In general, a large look-ahead distance can maintain the vehicle's tracking state, thus rendering larger curvatures near the corner. A sufficient look-ahead distance may result in cutting corners, dramatically reducing performance. On the contrary, a short look-ahead distance can guide the vehicle to travel faster toward the reference path so that it can quickly regain the path between the waypoints. As depicted in Figure 2.2, the vehicle overshoots the intended path and tends to oscillate along the path.

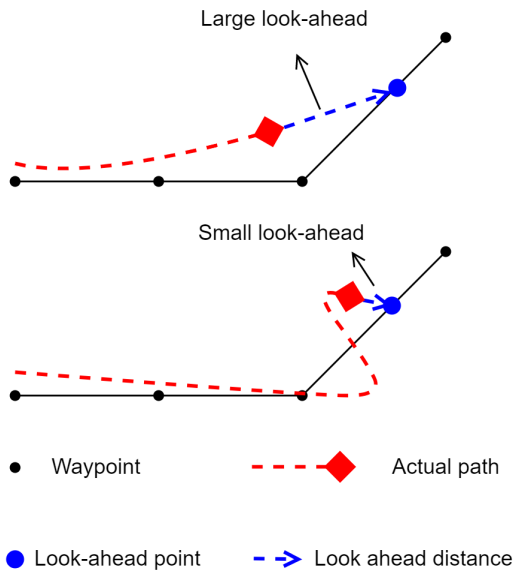


Figure 2.2: The property of look-ahead distance.

2.3 Simulation Results and Discussions

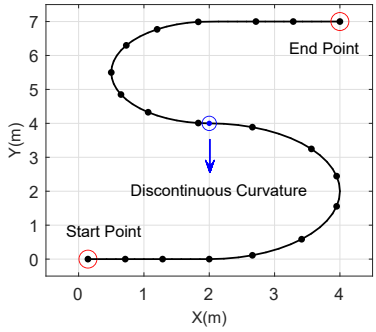
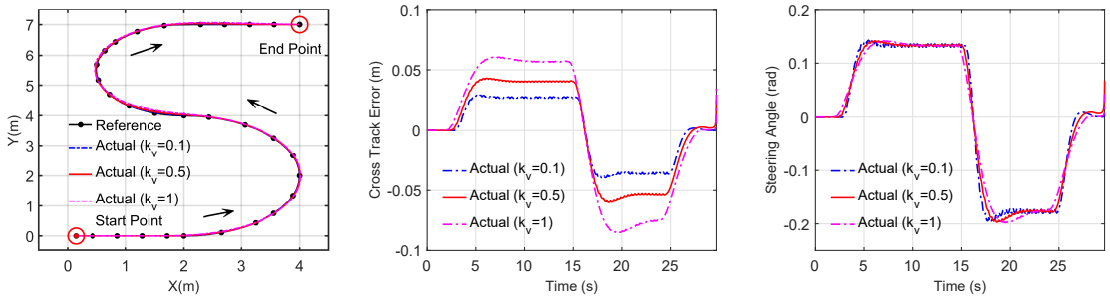


Figure 2.3: The discontinuous reference path for simulation test.

To further analyze the characteristic of the PP method, a series of simulations have been carried out in MATLAB/Simulink. An S-shape reference path featured with discontinuities at the point $P_r{}' = [2, 4]^T$ was devised by fitting the well-spaced waypoints, as shown in Figure 2.3. In which, straight lines and circular arcs are connected to construct driving straight maneuvers and turning circle maneuvers. The initial position of the vehicle was set as the same as the start point of the reference path, as $P_c = [0.143, 0, 0]^T$, and the

wheelbase was chosen as 0.256 m. The controlled car-like vehicle was tested at different constant velocities, namely 0.5m/s, 1m/s, and 1.5m/s. Moreover, we use different control gains in each case for the purpose of investigating how look-ahead distance influences tracking performance. Finally, from (2–7), it is known that a large k_v indeed corresponds to a longer look-ahead distance and vice-versa. The value of L_{dl} was determined as 0.35 m to ensure a minimum threshold to prevent failed path tracking.



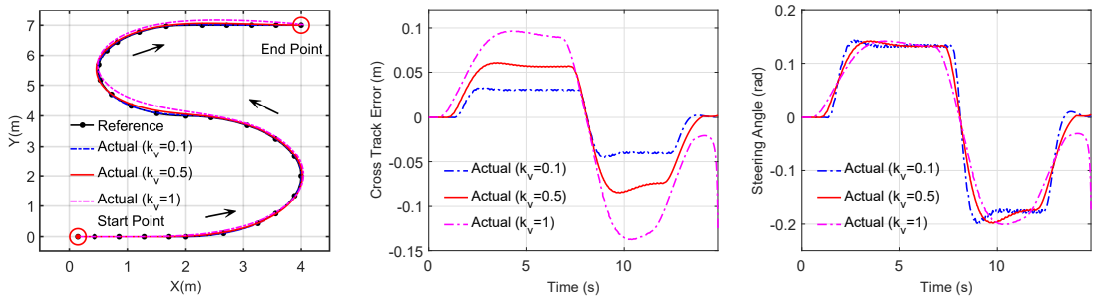
(a) (b) (c)

Figure 2.4: Simulation results (under low speed).

In Figure 2.4, it is clear that the car-like vehicle by PP method works surprisingly well at low speed, resulting in marginal cross tracking error within 0.1 m and similar tracking results. Specifically, the vehicle using the smallest control gain can realize path tracking with the least offsets during the entire duration, as evident in Figure 2.4(b). However, it seems that the steering commands have some tiny fluctuations, which should be concerned with real-time control. Additionally, the vehicle cuts corners from the discontinuous point, which may also be caused by unreasonable look-ahead distances.

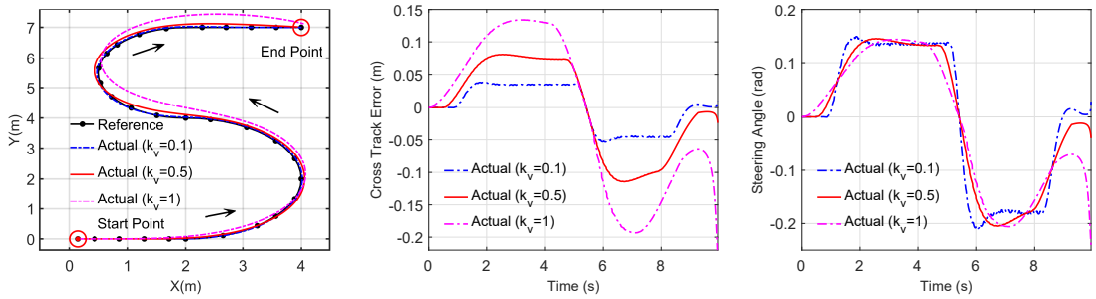
The simulation tracking results based on moderate speed (1 m/s) are summarized in Figure 2.5. Again, the car-like vehicle follows the reference path with the lowest value of gain, obtaining highest accuracy, as shown in Figure 2.5(b). While employing the largest look-ahead distance ensures the most stable and smooth steering, the vehicle apparently cuts the corner in the latter part of the tracking, as seen in Figure 2.5(a).

Given in Figure 2.6 are the simulation results of the car-like vehicle at high speed.



(a) (b) (c)

Figure 2.5: Simulation results (under moderate speed).



(a) (b) (c)

Figure 2.6: Simulation results (under high speed).

Notably, cutting corner problems reoccur and cause the turning of the vehicle not wide enough, particularly at the second corner, when the large k_v is selected. Contrasted with previous simulation results, the impact of look-ahead distance becomes greater on tracking performance as the speed and control gains are increased. The property of look-ahead distance is pointed out in Figure 2.6(b) and 2.6(c), from which we can observe that small k_v lead to smaller cross tracking error, but may make the vehicle suffer from unstable steering. Intuition would leave one to believe that enough small look-ahead distance should somehow cause the vehicle to steer quickly to regain its path between waypoints and render a more accurate following, which will be pointed out and verified in further experimental tests. By contrast, the vehicle can maintain its path in a stable way by utilizing a large look-ahead distance in spite of bringing about larger cross tracking errors. From these results, the vehicle successfully transits from the first circle to the second one regardless of the

speed and look-ahead distance, which indicates the strong robustness of the PP approach confronting the discontinuous path.

2.4 Experimental Results and Analysis

Comprehensive experiments, which include low, moderate, and high speed maneuvers with different look-ahead distances, have been applied on the Quanser self-driving car (QCar) platform to investigate the tracking performance of the PP controller, as well as the effect of the look-ahead distance in tracking control. Opposite to simulation tests, we directly specify the look-ahead distance at this stage without using the threshold in (2–7). The reference path used in the following experimental tests is not very common for car-like vehicles since it was designed to include the point $P_r{}' = [1.5, 2.5]^T$ with discontinuous curvatures, as displayed in Figure 2.7. This reference path can help us to have an insight into the robustness of the PP tracker.

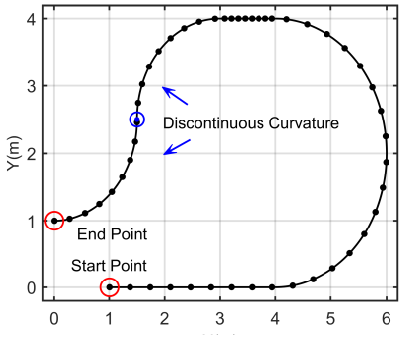


Figure 2.7: The discontinuous reference path for experimental test.

2.4.1 Experimental Test Bench

The test bench is based on the Quanser self-driving car research studio. As the core of this platform, the QCar is a 1/10 scaled model vehicle powered by an NVIDIA Jetson TX2 supercomputer and equipped with a wide range of sensors, such as 360 degree vision,

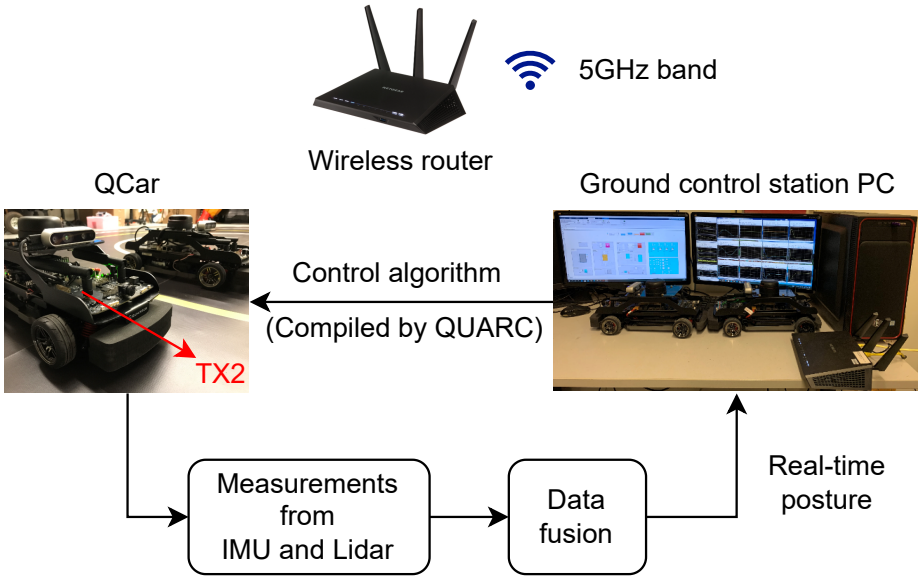
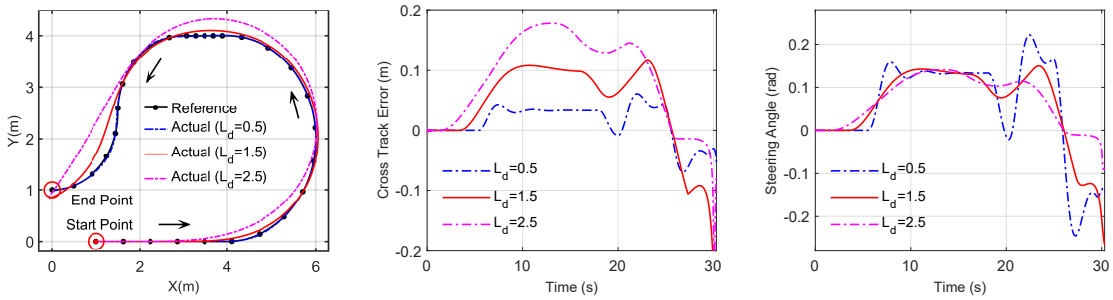


Figure 2.8: The schematic of the QCar experimental platform.

depth sensor, encoders, etc. The velocity measurement is available by an E8T optical shaft encoder. Also, the 9-axis inertial measurement unit (IMU) and RPLIDAR A2M8 are supplied to localize the QCar. Concerning QCar's motion, it is controlled by the drive motor and steering servo. The Quanser real-time control software (QUARC) applied on the Windows target generates C-code directly from the Simulink environment in which a tracking system model is established. In an embedded Linux system, the algorithm could be compiled and deployed into TX2. The connectivity and data transmission between the ground control station PC and the QCar is via a WiFi network pre-configured by the Netgear R7000 wireless router, which means that the PC receives the QCar's current posture in real-time. With respect to the forward (longitudinal) velocity and steering angle of the QCar, they are confined within [-1.5,1.5] m/s and [-0.524,0.524] radians, respectively. The wheelbase of QCar is 0.256 m. The whole structure of this test bench is shown in Figure 2.8.

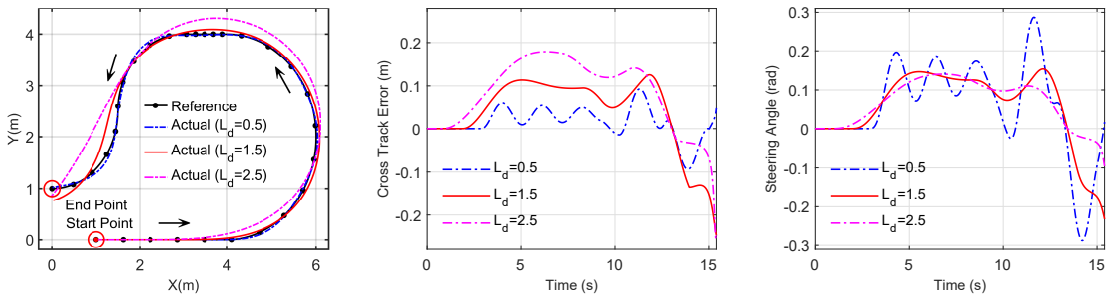


(a) (b) (c)

Figure 2.9: Experimental results (under low speed).

2.4.2 QCar Tests Results

Figure 2.9 illustrates the experimental tracking results of the QCar performing at a low speed. It is evident that the QCar can constantly pursue the moving point and accomplish path tracking precisely by the smallest look-ahead distance, as evident in Figure 2.9(a). The QCar with other look-ahead distances, however, cannot guarantee enough tracking accuracy, especially during 25-30 s. Also, the turn of the QCar at the first corner is not wide enough, that is, the QCar moves not far enough forward before starting to turn the wheel, which is also called cutting corner behavior.

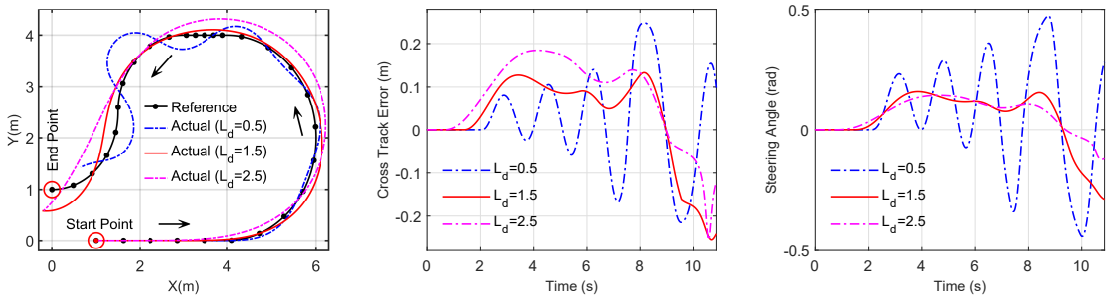


(a) (b) (c)

Figure 2.10: Experimental results (under moderate speed).

Presented in Figure 2.10 is the actual trajectory of the QCar at 1 m/s. Again, the smallest look-ahead distance ensures minimum offsets, while the largest one contributes the

smoothest steering commands. From Figure 2.10(b) and 2.10(c), the 1.5 m look-ahead distance can enable the QCar to stably track the path without compromising too much tracking precision. Despite the fact that the cross tracking error is limited in the range [-0.1,0.1] by selecting the shortest look-ahead distance, the steering commands start to be oscillatory from $t = 2.9$ s.



(a) (b) (c)

Figure 2.11: Experimental results (under high speed).

The QCar works at high-speed conditions, leading to apparently dissimilar experimental results, as shown in Figure 2.11. Opposite to previous experimental tests, the smallest look-ahead distance no longer guarantees the minor cross tracking error for the reason that the tracking speed is too fast and the L_d is too short for the QCar. As a result, the QCar overshoots the path and oscillates along the intended path, as seen in Figure 2.11(a). Simultaneously, wild swings occur in the steering wheels, and this effect is amplified as the curvature of the radius switches, as evident in Figure 2.11(c). Besides, the QCar with distinct look-ahead distances stops at different positions, which are not close to the endpoint of the desired path. This is because the PP controller cannot stabilize the QCar at an exact point. A distance threshold for the desired position can be applied to stop the QCar near the goal location, as described in (2–7).

Noted that the measurements of cross tracking error cannot explicitly describe the exact deviations between the QCar and reference path, given that the cross tracking error is simply computed through the definition of (2–5). For example, in Figure 2.11(a), some offsets

are obvious, but the cross tracking errors are only capped within [-0.3,0.3]. In this case, it will be challenging to judge the real performance of the QCar. As an alternative, the errors defined in [7] can clearly portray the differences between the desired path and actual trajectory. Based on tracking results in Figure 2.9–2.11, we notice that the PP controller has a high level of robustness to the quick transient section, since (2–4) can directly derive the steering commands via only one variable α. This implies that the QCar will immediately execute turning whenever there is an angle error between the look-ahead vector and the heading vector of the QCar. Further, in Figure 2.9(a), 2.10(a) and 2.11(a), the QCar by PP controller can still follow and stay on the path as it moves to the last corner, which demonstrates the strong robustness against discontinuity. It turns out that the QCar with suitable look-ahead distances performs fairly well on constant curvature paths, whereas this performance cannot be sustained with a different curvature or speed. This phenomenon is attributable to the design of the PP algorithm, which ignores the curvature of the intended path and calculates an arc founded on the geometric relationships in Figure 2.1. Under such a circumstance, the actual trajectory of the QCar may deviate from the intended path due to the incorrect model-predicted curvature. This problem will be more pronounced for high curvature and high tracking speed.

Chapter 3

Tracking Control based on Stanley Method

3.1 Introduction

As another sort of geometric tracking approach, the Stanley tracker has been a standard benchmark to verify a new tracking control scheme proposed by active researchers because of its simplicity and desirable tracking performance. Stanford University deployed the Stanley algorithm, initially proposed by Hoffmann et al. [5] into AVs, and finally won the Second DARPA Grand challenge in 2005 from the 195 participating teams. Even though the stellar tracking performance of the Stanley vehicle in this race may be attributed to the perception and planning module, the Stanley controller has been proved to be a handy tool in AVs and robotics.

The Stanley method is essentially a nonlinear feedback function whose exponential convergence characteristic was proved. Also, the Stanley controller can make the decay of tracking errors to be independent of the speed. Three considerations form the basis of the final steering control law. More concretely, the reference point of vehicles is switched to the center of the front axle rather than the c.g. or the rear axle. Next, the heading errors

are added with the aim of aligning the vehicle's heading with the reference path, and the cross tracking error is measured from the center of the front axle to the nearest point on the reference path without using a look-ahead distance. Finally, the solutions of the control law can be capped to fall within the maximum steering angle bounds.

3.2 Control Algorithm

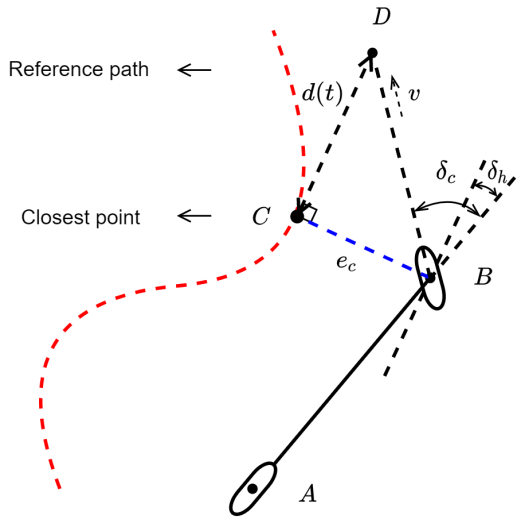


Figure 3.1: Stanley strategy.

Now, we explain more details about the Stanley control algorithm. Figure 3.1 demonstrates the geometric relationship between the pre-determined path and the kinematic bicycle model. The steering angle of the controlled vehicle consists of two parts, which are caused by the heading error and lateral error, respectively. Without considering the cross tracking error, the direction of the front wheel should be consistent with that of the path tangent so as to correct the misalignment between the vehicle and the reference path. Therefore, the heading error can be set equal to the steering angle δ_h, such that:

$$\delta_h = \theta_r - \theta_c = \theta_e \tag{3–1}$$

Similarly, an appropriate steering command needs to be determined when the heading error is assumed to be zero. In order to eliminate the cross tracking error, we suppose that the intended trajectory of the vehicle intersects the path tangent of the closest point C at point D, as displayed in Figure 3.1. Apart from this, a proportional control whose gain is set inversely to the forward speed is introduced to estimate the distance of CD, which can be expressed as:

$$d(t) = v/k \tag{3–2}$$

According to the simple geometric relationship, we can derive:

$$\delta_y = \arctan \frac{e_c(t)}{d(t)} = \arctan \frac{ke_c(t)}{v(t)} \tag{3–3}$$

Consequently, the total steering angle can be obtained as an intuitive way as follows:

$$\delta_c = \delta_h + \delta_y = \theta_e + \arctan \frac{ke_c(t)}{v(t)} \tag{3–4}$$

From (3–4), it is evident that the steering control objective can be realized. That is to say; the steering angle tends to be small to keep driving safety as the speed of the vehicle is high. If the vehicle is operating at a low speed, this control law will generate large steering commands to quickly adjust the heading of the vehicle. Also, a larger steering command can be attained by Stanley control law to correct the vehicle's posture to guide it to move towards the path as long as the cross tracking error increases.

According to the boundary between steering regions, the resulting control law can be written as:

$$\delta_c = \begin{cases} \delta_{max} & \delta_c > \delta_{max} \\ \delta_c & |\delta_c| \leq \delta_{max} \\ -\delta_{max} & \delta_c < -\delta_{max} \end{cases} \tag{3–5}$$

Furthermore, it is easy to get the rate of change of the cross tracking error with the geometric relationships displayed in Figure 3.1, which can be described as follows:

$$\dot{e_c} = -v \sin \delta_y \tag{3–6}$$

Using a trigonometric identity for seeking the explicit expression of $\sin \delta_y$, such that:

$$\sin \delta_y = \frac{ke_c}{\sqrt{v^2 + (ke_c{}^2)}} \tag{3–7}$$

Then, by substituting (3–7) into (3–6), the variable $\dot{e_c}$ may be rewritten as:

$$\dot{e_c} = \frac{-ke_c}{\sqrt{1 + (ke_c/v)^2}} \tag{3–8}$$

With respect to small cross tracking error, equation (3–8) can be simplified by regarding the quadratic term is negligible, such that:

$$\dot{e_c}(t) \approx -ke_c(t) \tag{3–9}$$

As a consequence, it is easy to find that the solution for this first-order differential equation is exponential. Since the value of control gain is positive, the cross tracking error will finally converge to zero exponentially as time goes to infinity. An interesting observation is that (3–9) does not associate with the speed term, which implies that vehicles having the same initial posture will finally converge to the desired path with the same amount of travel time regardless of the speed.

When the vehicle is operated in a low-speed range, the Stanley controller confronted with noisy velocity estimates may behave aggressively. The cross tracking errors tend to be amplified in the steering commands due to the fact that the speed term is in the denominator in (3–4). In order to avoid unreasonable steering and enhance the tracking stability, a positive softening constant term should be added as:

$$\delta_c = \delta_h + \delta_y = \theta_e + \arctan \frac{k e_c(t)}{k_s + v(t)} \tag{3–10}$$

Accordingly, the denominator has a minimum threshold to guarantee no wild swings in the steering wheel. During the high speed operation, another problem comes up that the vehicle's steering should be changed slowly to ensure tracking stability. Thereby, we can add a damping term on the heading rate to act essentially as a proportional-derivative controller to reduce the overly aggressive response of Stanley tracker.

3.3 Simulation Results and Analysis

In this section, the tracking performance of the car-like vehicle by the Stanley approach has been fully verified and evaluated via MATLAB/Simulink. Three simulation cases are based on different constant driving speed and control gains, aiming to comprehend how these variables affect the tracking results. The initial position for the car-like vehicle and U-shaped reference path are $P_c{}' = [0, 0]^T$ and $P_r{}' = [1, 0]^T$. The wheelbase was set as 0.256 m, which is the same as the parameter of QCar. In addition, side slip is not taken into account during these simulation tests due to the relatively slow motion of the car-like vehicle.

In the first instance, the car-like vehicle follows the U-shaped path at a low speed (0.5 m/s) from the start point. Obviously, the tracking results based on the smallest gain are

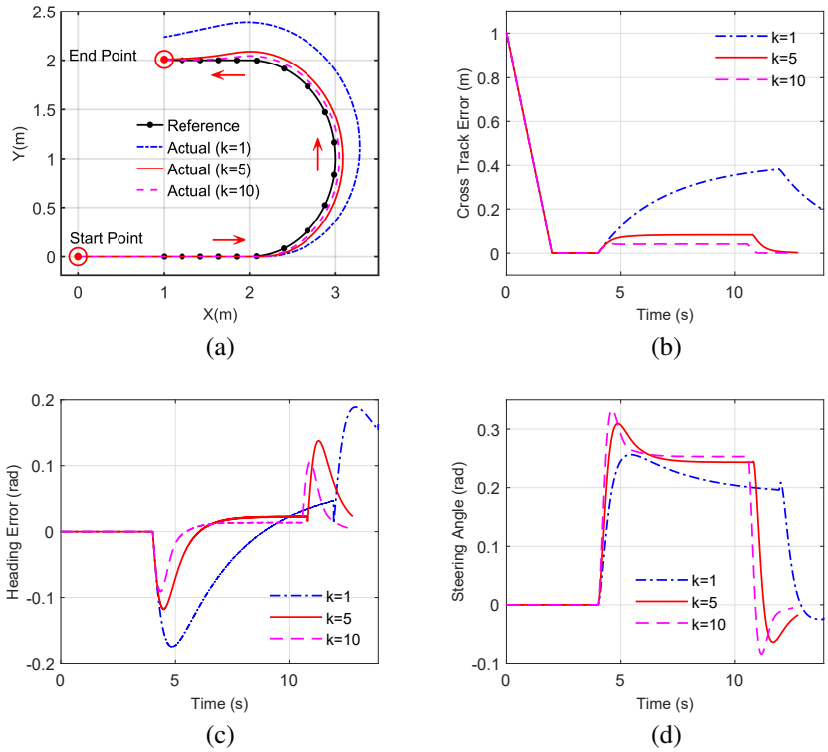


Figure 3.2: Simulation results (under low speed).

unsatisfactory because of the overlarge deviation and delayed convergence of the heading error. In contrast, the car-like vehicle with other control gains contributes to negligible cross tracking error, as well as fast adjustment of the vehicle's heading, as evident in Figure 3.2(a)–3.2(c). The evolution of the steering angle with time is illustrated in Figure 3.2(d), in which no unstable steering occurs during the whole tracking period.

Afterwards, the longitudinal velocity was tuned to 1 m/s as a moderate velocity for the car-like vehicle. The corresponding simulation results are shown in Figure 3.3. Again, despite resulting in the most stable steering, the vehicle with the smallest gain steers too late and even does not converge to the U-shaped path at the ending time. The tracking results by other control gains are similar, and they both enable the vehicle to converge to the path without too much steady-state error. For this phase, it seems that increasing the value of gain can reduce the cross tracking error and enhance the convergence rate of heading error, as shown in Figure 3.3(b) and 3.3(c).

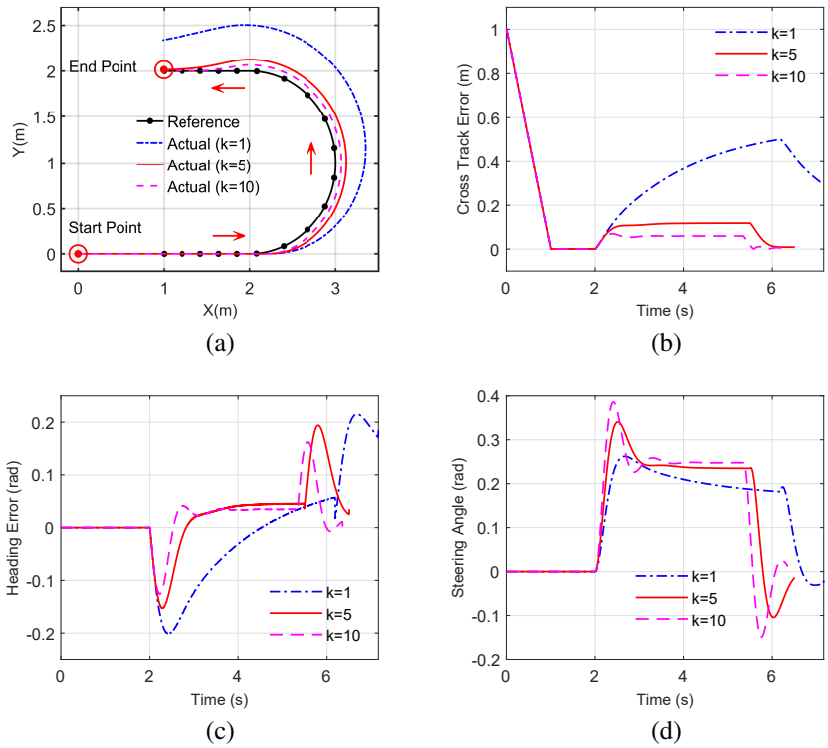


Figure 3.3: Simulation results (under moderate speed).

Figure 3.4 demonstrates the high speed (1.5 m/s) simulation results of the car-like vehicle. Compared with the simulation results in Figure 3.2 and 3.3, the tracking performance deteriorates as the driving speed boosts. That is to say, the cross tracking error and the heading error become larger than before, especially when the vehicle performs turning maneuvers. Also, with $k = 10$, it should be noted that the steering signals tend to be oscillatory during around 1.5-3 s, which does not occur in other simulation results. In other words, even though the larger control gain ensures higher tracking precision and fast convergence rate, the over-tuned gain may lead to unstable steering and oscillations in the tracking system. On the other hand, under-tuned gains directly give rise to large errors in the vehicle's position and heading, as evident in Figure 3.2–3.4. Consequently, well-tuned gains play a crucial role in Stanley control. An interesting point is that the Stanley tracker is better suited to high-speed tracking for car-like vehicles compared with the PP method on the condition that the control gain is chosen properly.

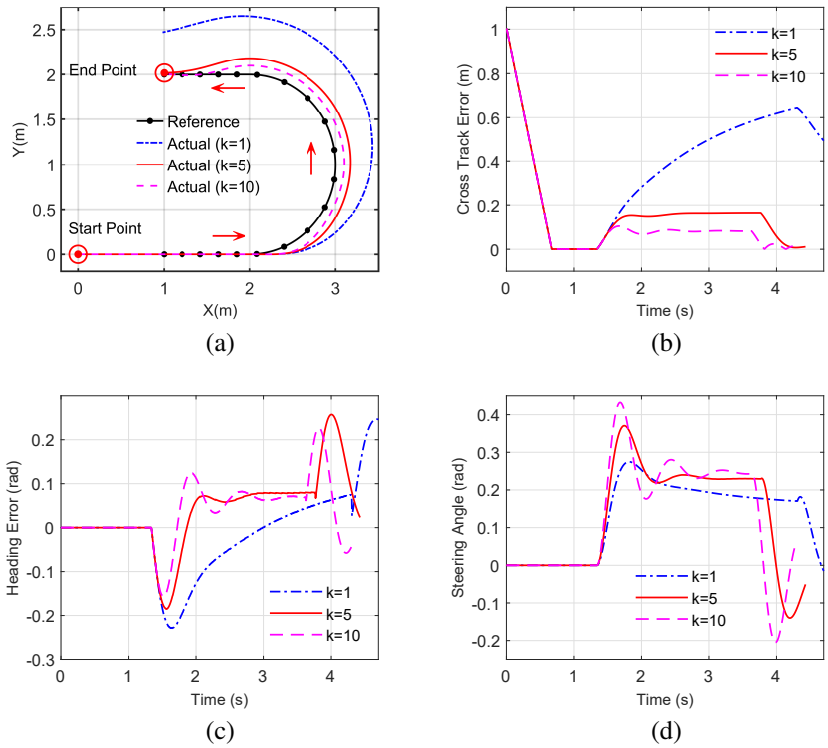


Figure 3.4: Simulation results (under high speed).

3.4 Experimental Results and Analysis

To further investigate and verify the real-world performance of the Stanley method, two extreme driving scenarios involving large initial cross tracking error and large initial heading error are conducted based on the QCar platform. In this section, the measurements of cross tracking error are replaced with that of absolute position tracking error due to the fact that the pre-defined path is designed as a function of time. In this case, the latter is more likely to clearly describe the exact extent of deviation between the desired position and the current position.

3.4.1 Large Initial Cross Tracking Error

In the first case, the value of the large initial cross tracking error and the maximum allowable steering angle were set to roughly 1.42 m and 0.52 radian, respectively. The

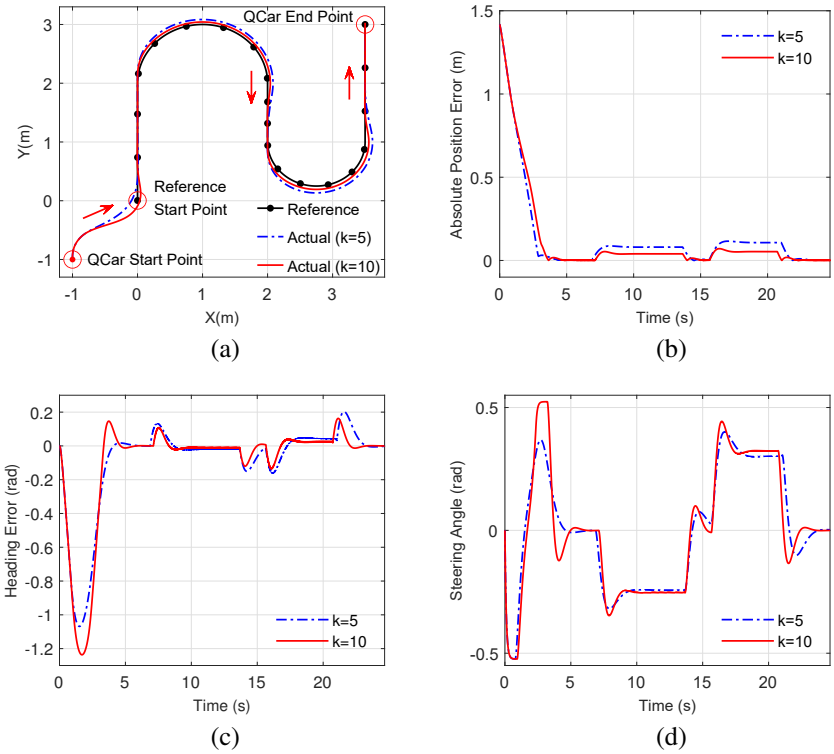


Figure 3.5: Experimental results (large initial cross tracking error & low speed).

initial posture of the QCar is $P_c = [-1, -1, \pi/2]^T$. Figure 3.5 illustrates the relatively good experimental tracking results at 0.5 m/s. Despite the fact that the controller gain was tuned from 5 to 10, the QCar works quite well and results in similar tracking performance. Therefore, the tuning of gains in this region has a slight effect on tracking under low-speed conditions. The experimental results under moderate speed (1 m/s) are similar to that of low speed, as seen in Figure 3.5. During this state, the tracking performances by different gains are still satisfied. Nonetheless, the gaps between the blue dashed line and the red solid line are widened, which indicates that the effects of control gains become manifest as the speed boosts. The high speed (1.5 m/s) tracking results of the QCar is provided in Figure 3.7, where obvious discrepancies can be detected by comparison to previous low and moderate counterparts. While the QCar with the over-tuned gain ($k = 10$) tracks the reference path without obvious deviations, the steering signals varying too fast may result in excessive lateral velocity, acceleration, and instability. In this way, the tracking performance may be

degraded with more oscillations and instability, and the operational life of actuators will be considerably reduced, as evident in Figure 3.7(c) and 3.7(d).

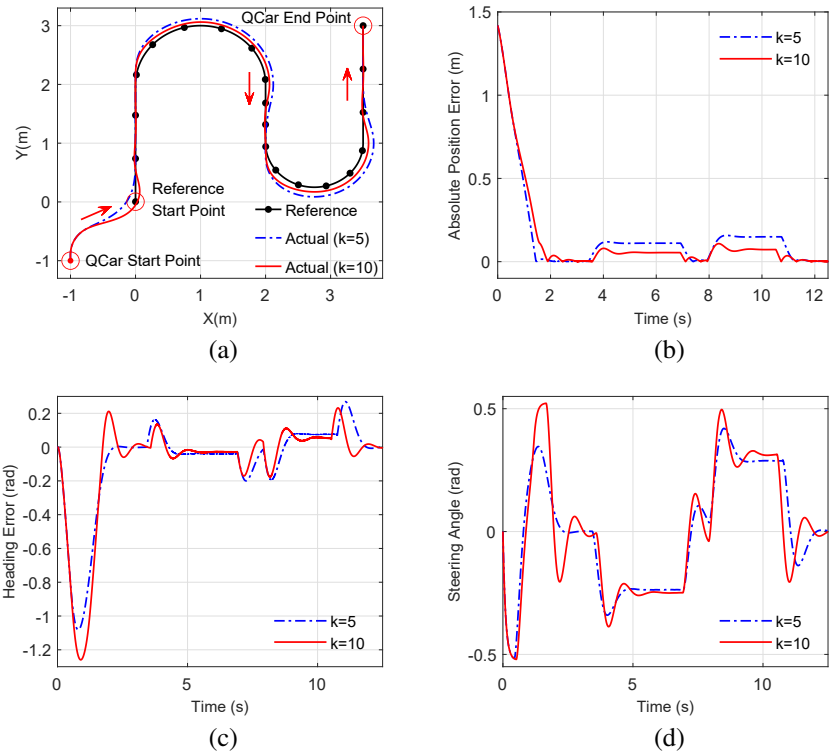


Figure 3.6: Experimental results (large initial cross tracking error & moderate speed).

From an overall perspective, these experimental results point out how the Stanley controller eliminates a large cross tracking error and drives the QCar to the intended path. At the early stage of tracking, the large cross tracking error results in large steering signals that adjust the direction of QCar to that of the reference path, and then there is an equilibrium achieved between θ_e and e_c terms. Before the QCar travels into the first corner, the cross tracking error decays to zero in an exponential way, as shown in Figure 3.5(b), 3.6(b) and 3.7(b). Simultaneously, the heading error has been reduced to zero so that the QCar continues in a straight line towards the reference path, as shown in Figure 3.5(c), 3.6(c) and 3.7(c). When the vehicle turns into the first corner, the QCar tracks the intended path as the principle that we explained before. That is, the QCar using the Stanley controller can correct the misalignment in heading and errors in position to realize path tracking.

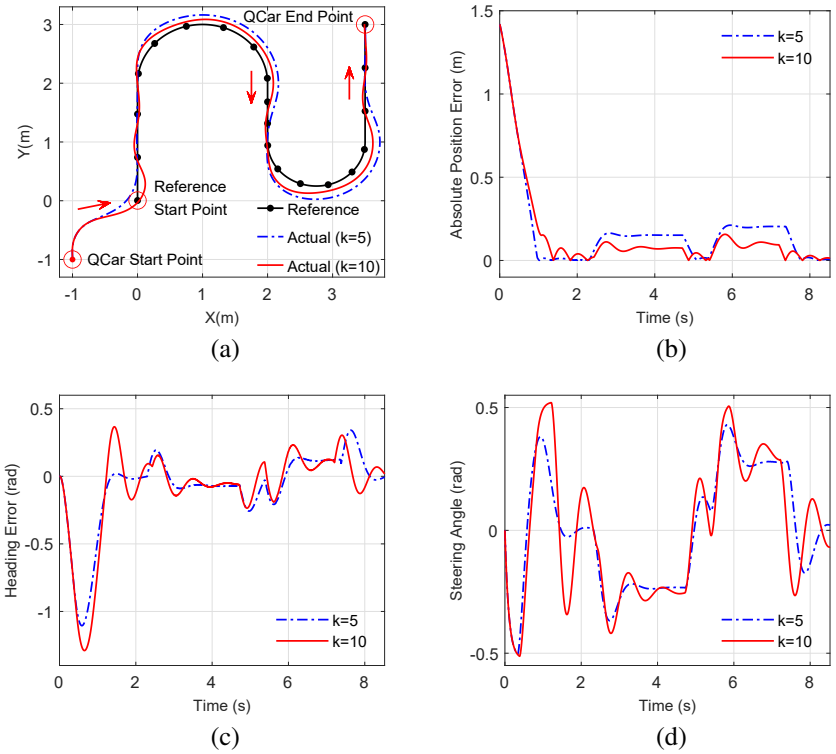


Figure 3.7: Experimental results (large initial cross tracking error & high speed).

3.4.2 Large Initial Heading Error

For the second case, this scenario involving a large initial heading error was regenerated based on case one. The parameters and the settings of velocity were chosen as the same in the first case. The initial posture error $P_e = [0, 0, \pi/2]^T$, however, allows for the QCar to point quite a wrong direction at the start point. The low-speed experimental tracking results are given in Figure 3.8. Although there exist some minor cross tracking errors during the turning stage, the QCar utilizing Stanley control performs well in heading tracking and maintains the heading error around zero from around 5 s until the end, as shown in Figure 3.8(c). In Figure 3.8(b), as the value of control gain is increased from 5 to 10, the tracking precision is enhanced due to the smaller absolute position tracking errors. The experimental tracking results based on moderate speed (1 m/s) are displayed in Figure 3.9, where apparent differences in contrast with the low-speed experimental results is that the

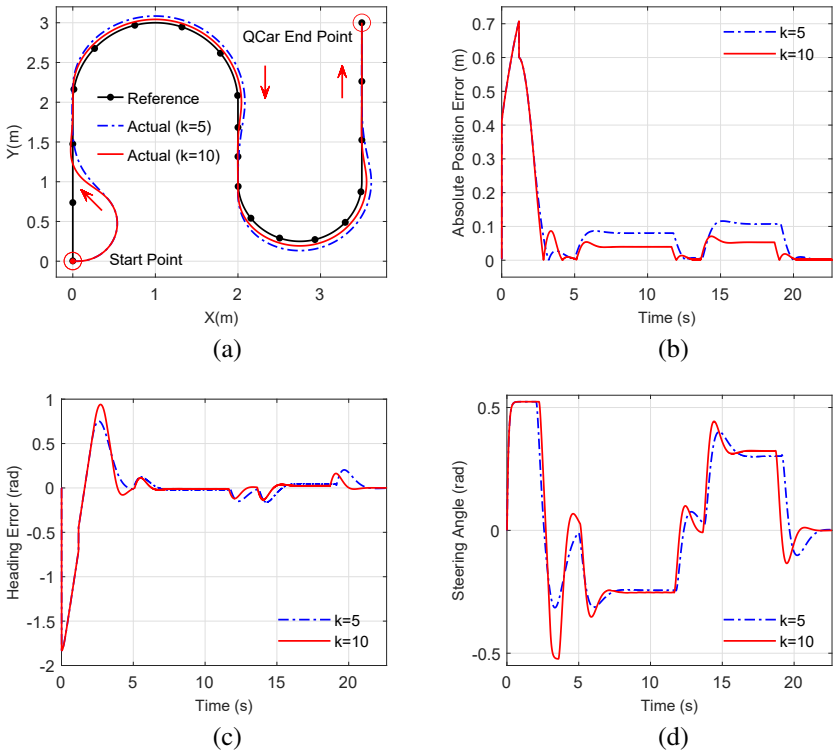


Figure 3.8: Experimental results (large initial heading error & low speed).

effect of control gain becomes more visible. It seems that there exist some control gains between 5 and 10, which could balance the problem between smooth steering and poor tracking precision. The experimental results of the high speed tracking of the QCar, given in Figure 3.10, are the worst compared with other results. While these tracking results are not desirable, it is worth mentioning that path tracking is still realized in some respects. Despite having oversteering behaviors at around point $P_r{}' = [0, 1.5]^T$, the QCar utilizing large gain can generate large steering commands to rapidly regain its path and mitigate the tracking accuracy deterioration arising from high speed. From Figure 3.8(a), 3.9(a) and 3.10(a), there is an interesting point that the QCar with higher speed drives farther before completely reaching the reference path. In spite of this, the QCar costs the same amount of time to make the error convergence to zero in each test. This high speed maneuver is simply treated as an analysis tool for tuning gains, and the tracker does not need to complete it.

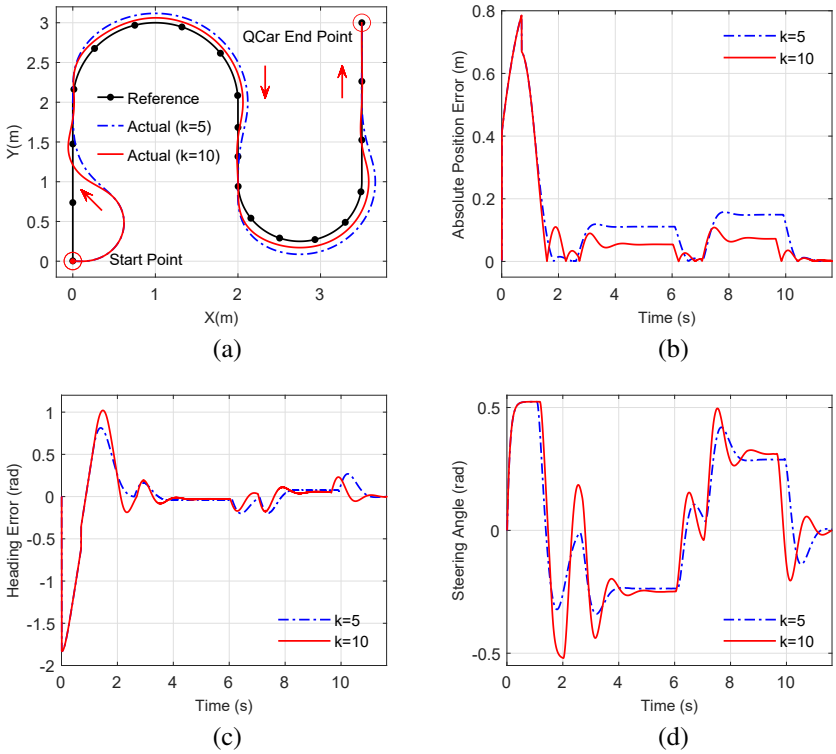


Figure 3.9: Experimental results (large initial heading error & moderate speed).

Overall, these experimental results show how the Stanley control eliminates the large initial heading error. Firstly, the value of steering angle surges the maximum limits within 1.5 s as the heading error increases. At the same time, the value of absolute position tracking error and cross tracking error reaches a peak, followed by a dramatic drop to zero. Then, the steering commands continuously correct the heading of the QCar so that it can remove misalignment with the path. As a result, the QCar exponentially converges to the reference path, which reflects on the experimental results that the QCar travels on the line segments before turning into the first corner. From Figure 3.8–3.10, it turns out that the effect of control gain becomes more and more influential as the speed grows. This phenomenon can also be observed in PP control. So it is crucial to select control parameters to avoid tracking instability and guarantee tracking precision based on some trial and error testing before we do real-time control.

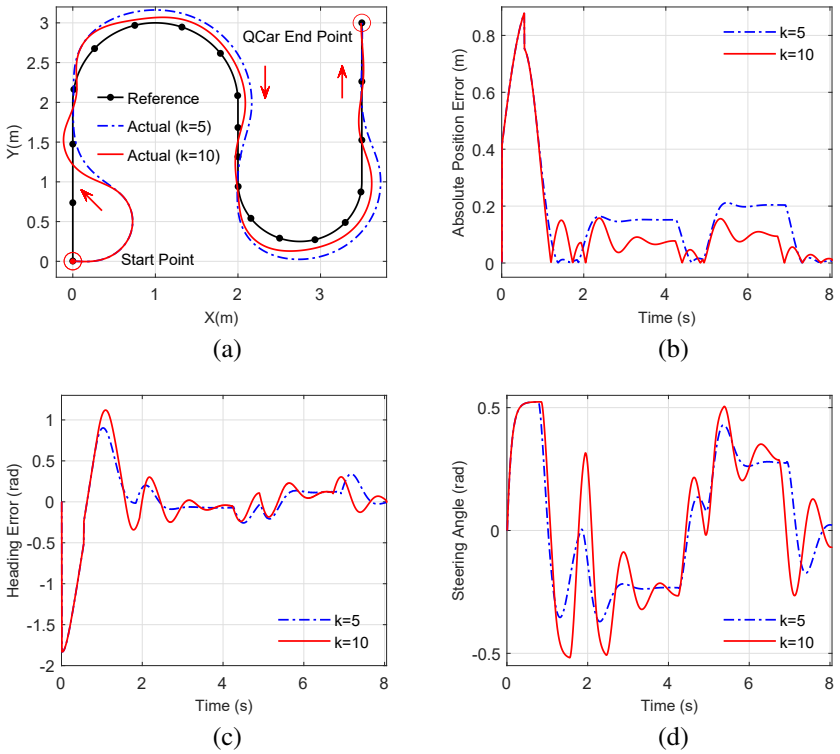


Figure 3.10: Experimental results (large initial heading error & high speed).

Chapter 4

Observer-based Adaptive Trajectory Tracking Control of Car-Like Vehicles Using Particle Swarm Optimization

4.1 Problem Statements

4.1.1 Vehicle Kinematic Model and Reference Trajectory

To investigate the trajectory tracking control, a kinematic model has been formulated to describe the vehicle's motion. The model configuration with reference trajectory is shown in Figure 4.1. In this research, four assumptions have been imposed on the development of this car-like vehicle's model:

(1)The vehicle's body and suspension systems are rigid, ignoring the front and rear axle load transfer.

(2)The kinematics is deduced purely from geometric relationships without relating the force that affects the motion.

(3)No slipping occurs between wheels and ground surface for low-speed vehicles, and z-axis motion is not considered.

(4)Speed and steering angle of the left and right wheels are assumed to be identical for this front-wheel-only steering model. The rear wheels are fixed and parallel to the body.

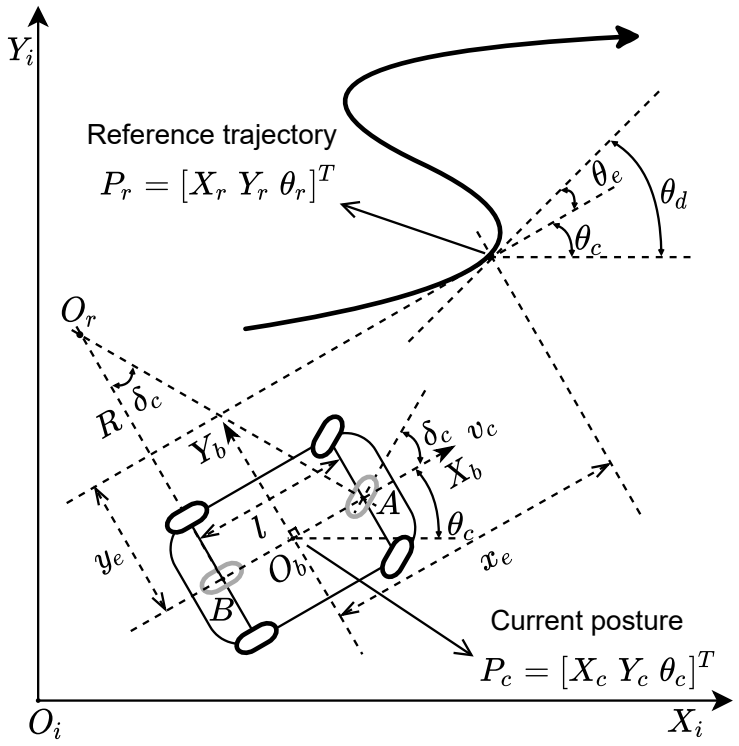


Figure 4.1: Kinematic model, reference trajectory, and posture errors.

In order to analyze the 2D planar motion of this model, the global Cartesian coordinate frame $F_i(O_i\ X_i\ Y_i)$ (a right-handed one) is used as a reference for the current vehicle's posture $P_c = [X_c\ Y_c\ \theta_c]^T$, where the variable $P_c{}' = [X_c\ Y_c]^T$ denotes the vehicle's location of the c.g., and θ_c is called the yaw angle of the vehicle whose negative direction is taken clockwise from the orientation of vehicle's body to X_i-axis. Also, body frame $F_b(O_b\ X_b\ Y_b)$ attached to the vehicle is established so as to facilitate kinematic analysis. In other words, this frame translates and rotates with the vehicle. The axis of X_b and Y_b indicate forward

speed and lateral error direction, respectively. The central control point O_b is at the c.g. because of the short wheelbase $l = 0.256$ m.

Due to Assumption 4, this model could be treated as a bicycle model. The front and rear turning radius AO_r and BO_r can be acquired respectively by separately building vertical lines in the direction of the wheels. The bicycle model turns about the instantaneous rotation point O_r with a rear turning radius R (BO_r). Referring to the Ackerman model and low-speed hypothesis [5], it is reasonable to reckon that the radius of a vehicle's trajectory varies slowly. That is, the vehicle's angular velocity ω_c and yaw (orientation) change rate $\dot{\theta}_c$ are equivalent. As a result, the radius R of curvature and angular velocity ω_c of the kinematic model can be described as:

$$R = l/\tan\delta_c \quad \omega_c = \dot{\theta}_c = v_c \times \tan\delta_c/l \tag{4–1}$$

in which $q_c = [v_c \ \delta_c]^T$ is actual control commands for the tracking system. The forward speed command $v_c \geq 0$ governs the vehicle's translational motion, while rotational motion is controlled by the steering command δ_c. After rigorously analyzing the motion features of the vehicle model based on multiple assumptions, the entire kinematics may be deduced by geometric relationships, as shown in (4–2).

$$\dot{P}_c = \begin{bmatrix} \dot{X}_c \\ \dot{Y}_c \\ \dot{\theta}_c \end{bmatrix} = \begin{bmatrix} \cos\theta_c & 0 \\ \sin\theta_c & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_c \\ \omega_c \end{bmatrix} \tag{4–2}$$

The goal of trajectory tracking control is for the car-like vehicle to follow the reference trajectory. The desired reference information can be derived according to the predetermined reference position and corresponding derivatives as following:

$$\begin{cases} \theta_r = \operatorname{atan2}\left(\dot{Y}_r, \dot{X}_r\right) \\ v_r = \sqrt{\dot{X_r}^2 + \dot{Y_r}^2} \\ \omega_r = \dot{\theta}_r = (\dot{X}_r\ddot{Y}_r - \dot{Y}_r\ddot{X}_r)/v_r^2 \end{cases} \tag{4–3}$$

where $v_r > 0$, ω_r, θ_r are the linear velocity, angular velocity and yaw angle of the desired trajectory concerning the inertia frame, respectively. The desired steering angle is expressed as δ_r. The actual output of trajectory planning is $q_r = [v_r\ \delta_r]^T$ rather than velocity information $q_r{}' = [v_r\ \omega_r]^T$. The reference posture $P_r = [X_r\ Y_r\ \theta_r]^T$ represents the posture of the set-point. The reference signals and their relevant derivatives should be bounded, continuous, and persistently excited. Once the above conditions hold, the trajectory planning problem can be transferred to design time functions $X_r(t)$ and $Y_r(t)$. Thus, together with proper control law, the car-like vehicle is supposed to achieve the desired pose introduced by (4–3).

4.1.2 Posture Error

After defining the car-like vehicle kinematics, reference trajectory, fixed and moving coordinate frame with detailed descriptions, we introduce a posture error concept intending to figure out the tracking distance. Tracking errors are transformed from the fixed coordinate frame to the moving counterpart, which is depicted in Figure 4.1 and defined as:

$$\begin{cases} x_e = (X_r - X_c)\cos\theta_c + (Y_r - Y_c)\sin\theta_c \\ y_e = (X_c - X_r)\sin\theta_c + (Y_r - Y_c)\cos\theta_c \\ \theta_e = \theta_r - \theta_c \end{cases} \tag{4–4}$$

where $P_e = [x_e\ y_e\ \theta_e]^T$ is a state-tracking error implying the difference between P_r and P_c at a certain moment. Specifically, x_e and y_e are the longitudinal and lateral error, and θ_e represents the yaw error inside the interval $(-\pi, \pi)$.

Assumptions 2 and 3 suggest nonholonomic constraints subject to the car-like vehicle, which leads to:

$$0 = \dot{X}_c \sin\theta - \dot{Y}_c \cos\theta \tag{4–5}$$

Substituting (4–1), (4–2) and (4–5) into the time derivative of (4–4), so the tracking error dynamics can be attained:

$$\begin{cases} \dot{x_e} = y_e\omega_c - v_c + v_r\cos\theta_e \\ \dot{y_e} = -x_e\omega_c + v_r\sin\theta_e \\ \dot{\theta}_e = \omega_r - \omega_c = \dot{\theta}_r - \dot{\theta}_c \end{cases} \tag{4–6}$$

4.1.3 Control Gains

For the sake of explaining simply and explicitly, the classical control law proposed in [7] is a typical example of a tracking method that can be used for giving details on the effect of control gain. The suggested control inputs are expressed as:

$$\begin{cases} v_c = v_r\cos\theta_e + K_1 x_e \\ \omega_c = \omega_r + v_r(K_2 y_e + K_3\sin\theta_e) \end{cases} \tag{4–7}$$

where $K = [K_1\ K_2\ K_3]$ is a set of positive control gains.

Generally, gain determines the convergence characteristics of the error. Even though a well-designed controller plays a pivotal role in the tracking system, different combinations of gains contribute directly to the value of control input $q_c{}' = [v_c\ \omega_c]^T$. In other words, a valid set of gains are conducive to stable, smooth and effective tracking results with moderate convergence rates. In contrast, irrational gain selection may lead to oscillations or instabilities emerging from the control system. Sometimes even bring about a vehicle's stuck state or failed trajectory tracking. Consequently, how to choose the gain is of the

essence of the control law.

4.2 Control Strategies

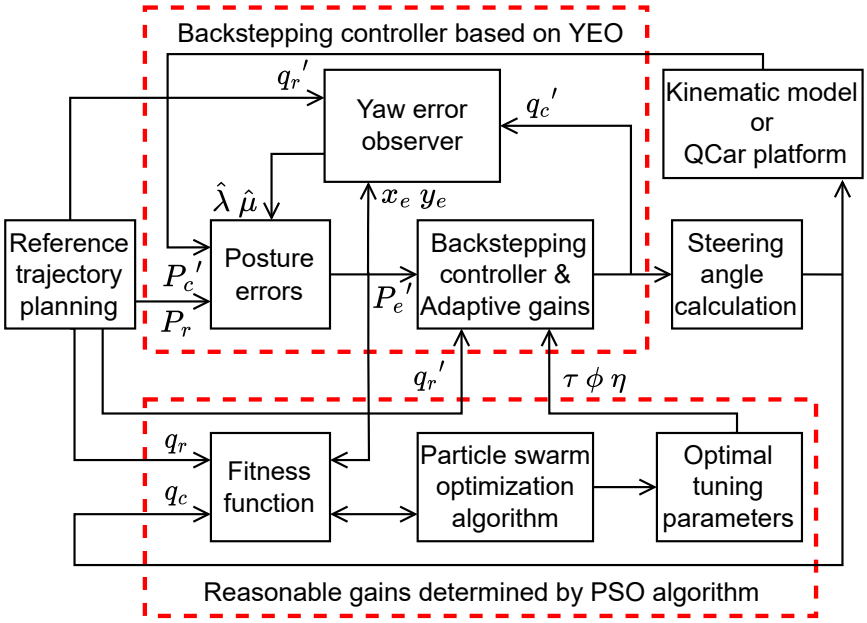


Figure 4.2: The schematic of the proposed kinematic-based tracking strategy.

The complete closed-loop structure of the tracking control system is depicted in Figure 4.2. In the first step, we design a backstepping controller considering input boundaries based on Lyapunov functions, which guarantees posture errors converge to zero states in a global asymptotically stable way. A robust yaw error observer (YEO) is proposed to estimate yaw error-related variables accurately, namely $\sin\theta_e$ and $\cos\theta_e$, solving measurement difficulty and integration drift problem of sensor techniques. Finally, optimal tuning parameters are obtained employing a novel method: particle swarm optimization (PSO) algorithm combined with the proposed controller and YEO to minimize posture errors by using the fitness function. With the tuning parameters, the design of self-updating gains are realized. The reader is referred to the following sections for more details.

4.2.1 Backstepping Controller and Adaptive Gains

The objective of the controller is to design a control law subjected to input saturation, which forces the tracking errors to converge to zeroes asymptotically as $t \to \infty$. Consequently, the car-like vehicle realizes trajectory tracking stably. Observe equations of error dynamic (4–6), it is obviously not able to control y_e directly. Nevertheless, this obstacle can be addressed by making use of the backstepping control strategy. We decompose the nonlinear system into cascaded subsystems. The following related Lyapunov-like equations are set up on the Lyapunov direct method, meaning that functions are positive definite (PD) and their derivatives are negative definite (ND). The first candidate Lyapunov function is chosen based on the definitions as:

$$L_1 = \frac{1}{2}{x_e}^2 + \frac{1}{2}{y_e}^2 \tag{4–8}$$

Taking the time derivative of (4–8) along (4–6), such that:

$$\dot{L_1} = x_e\dot{x_e} + y_e\dot{y_e} = x_e(-v_c + v_r\cos\theta_e) + y_e v_r \sin\theta_e \tag{4–9}$$

Since we wish $x_e \to 0$ and $y_e \to 0$ as $t \to \infty$, an auxiliary error function $\sin\theta_{ev} = -K_2 y_e / v_r$ is designed, where $\sin\theta_{ev}$ is taken as a virtual control input and we assume $\sin\theta_e$ is identical to $\sin\theta_{ev}$ at this stage. For the foward speed, it is chosen as $v_r\cos\theta_e + K_1 x_e$ to keep $\dot{L_1}$ ND. Hence, we obtain:

$$\dot{L_1} = -K_1{x_e}^2 - K_2{y_e}^2 \tag{4–10}$$

From (4–8) and (4–10), we know that $\dot{L_1}$ and its derivative satisfy in PD and ND status by choosing appropriate gains, respectively. This reveals x_e and y_e can approach the equilibrium point 0 asymptotically as time goes to infinity.

As we mentioned in Section 4.1.3, the combination of controller gains determines the

overall properties of convergence, thereby affecting system stability and comprehensive tracking performance of the car-like vehicle. Compared to K_1, overlarge K_2 makes the tracking system unstable and slow longitudinal tracking down, while a low value of K_2 causes conspicuous lateral deviation on tracking results. For these reasons, K_2 is designed as a function of K_1 with the aim of using one gain term to control x_e and y_e convergence rate. On top of that, v_r is considered to be another argument in the function so that K_2 compensates for the effects from lateral error convergence rate and forward speed. The gain K_1 is replaced by τ to avoid confusion, and thus v_c is rewritten as:

$$v_c = v_r \cos\theta_e + \tau x_e \tag{4–11}$$

K_2 can be formulated as:

$$K_2 = K_1 \phi v_r = \tau \phi v_r \tag{4–12}$$

then the virtual input becomes:

$$\sin\theta_{ev} = -\tau \phi y_e \tag{4–13}$$

where τ and ϕ represent postive tuning parameters, and they are all bounded constants. Because v_r is a function of time, K_2 is converted to an adaptive form. Not only can K_2 enhance robustness of controller by introducing speed term but also make connections between x_e and y_e convergence rate. Substituting (4–11) and (4–12) into (4–9) yeilds:

$$\dot{L_1} = -\tau {x_e}^2 - \tau \phi v_r {y_e}^2 \tag{4–14}$$

It should be noted that above analysis and equation are valid on the basis of one prerequisite, which depends on $\sin\theta_e = \sin\theta_{ev}$. Now, the control objective is to ensure that $\sin\theta_e$ tracks desired virtual input precisely, such that $sin\theta_e \rightarrow \sin\theta_{ev}$. To achieve this goal, an

error variable $\widetilde{\sin}\,\theta_e$ describing the error between $\sin\theta_e$ and $\sin\theta_{ev}$ is defined as:

$$\sin\tilde{\theta}_e = \sin\theta_e - \sin\theta_{ev} \tag{4–15}$$

Taking the time derivative of (4–15) along (4–6) and (4–13), which can be calculated as:

$$\begin{aligned}\dot{\sin\tilde{\theta}_e} &= \cos\theta_e\dot{\theta}_e + \tau\phi\dot{y_e} \\ &= \cos\theta_e(\omega_r - \omega_c) + \tau\phi(v_r\sin\theta_e - x_e\omega_c)\end{aligned} \tag{4–16}$$

Only when $\sin\tilde{\theta}_e$ and y_e converge to zero concurrently as $t \to \infty$ can guarantee $\sin\theta_e \to 0$, which implies $\theta_e \to 0$. Next, define the Lyapunov-like function as follows:

$$L_2 = \frac{1}{2}{x_e}^2 + \frac{1}{2}{y_e}^2 + \frac{1}{2}\sin(\tilde{\theta}_e)^2 \tag{4–17}$$

The time derivative of (4–17) along (4–9), (4–13) and (4–15) is:

$$\dot{L_2} = -\tau{x_e}^2 - \tau\phi v_r{y_e}^2 + \sin\tilde{\theta}_e(y_e v_r + \dot{\sin\tilde{\theta}_e}) \tag{4–18}$$

Equation (4–18) is supposed to follow a ND format, such that:

$$-K_3\sin\tilde{\theta}_e = (y_e v_r + \dot{\sin\tilde{\theta}_e}) \tag{4–19}$$

Imagine that lateral error is able to converge to zero rapidly. This situation is apparently caused by a larger steering command, leading to a quick change of vehicle's orientation. For these reasons, we set K_3 proportional to K_2 in order that vehicle adjusts to desired orientation with a reasonable yaw error convergence rate based on K_2, that is:

$$K_3 = \eta K_2 = \tau\phi\eta v_r \tag{4–20}$$

The parameter η represents a postive constant similar to τ and ϕ. After substituting (4–16) and (4–20) into (4–19), ω_c is derived as:

$$\omega_c = \frac{v_r y_e + \omega_r \cos\theta_e + \tau\phi v_r \sin\theta_e + \tau\phi\eta v_r(\sin\theta_e + \tau\phi y_e)}{\cos\theta_e + \tau\phi x_e} \tag{4–21}$$

and the steering input is calculated as:

$$\delta_c = \arctan\left(\omega_c \times l/v_c\right) \tag{4–22}$$

From (4–18), $\dot{L}_2$ may be rewritten as:

$$\dot{L}_2 = -\tau {x_e}^2 - \tau\phi v_r {y_e}^2 - \tau\phi\eta(\sin\theta_e + \tau\phi y_e)^2 \tag{4–23}$$

The stability of the closed-loop system is proven by LaSalle's Invariance Principle. Equation (4–17) shows L_2 is PD and bounded while (4–23) reveals that $\dot{L}_2$ is negative semi-definite when $-\tau {x_e}^2 - \tau\phi\eta\sin\theta_e$ may be zero. Considering error dynamics, $\dot{L}_2 = 0$ only if $x_e = 0$, $y_e = 0$ and $\sin\tilde{\theta}_e = 0$. Futher, yaw error will converge to zero due to the fact that $y_e \to 0$ and $\sin\tilde{\theta}_e \to 0$ simultaneously. Therefore, the control system is globally asymptotically stable at equilibrium point 0 by implementing control law (4–11) and (4–21), which also indicates the control goal $P_e \to 0$ as $t \to \infty$ is achieved.

Given that the implementation capacity of the drive motor and steering servo, control constraints are imposed in order to limit overlarge velocity commands. To this end, the final velocity and steering control law within saturations are:

$$v_c = \begin{cases} V_{max} & v_c > V_{max} \\ v_c & v_c \leq V_{max} \end{cases} \tag{4–24}$$

$$\delta_c = \begin{cases} \delta_{max} & \delta_c > \delta_{max} \\ \delta_c & |\delta_c| \leq \delta_{max} \\ -\delta_{max} & \delta_c < -\delta_{max} \end{cases} \tag{4–25}$$

where $V_{max} = 1$ m/s and $\delta_{max} = 0.5$ rad are maximum values of control commands.

4.2.2 Yaw Error Observer Design

This subsection solves the obstacle of measuring yaw angle by replacing the real sensor with an observer-controller combination method. Compared with the vehicle's yaw angle measurement, the current position measurements are relatively accurate and more accessible. In real applications, yaw angle receiving from a positioning system is not infallible and occasionally unavailable owing to sensor noises, external disturbances, uncertainties and especially integration drift from IMU. Besides, in (4–6),(4–11) and (4–21), it is clear that yaw error only appears in the form of $\sin\theta_e$ and $\cos\theta_e$ rather than θ_e. Hence we design an observer, which serves as a sensor to provide angle variables to the backstepping controller.

Now, set $\lambda = \sin\theta_e$ and its estimation is $\hat{\lambda} = \hat{\sin}\,\theta_e$. Similarly, $\mu = \cos\theta_e$ and $\hat{\mu} = \hat{\cos}\,\theta_e$. In this case, the explicit expression of x_e and y_e can be written as:

$$\begin{aligned} x_e =& (X_r - X_c)(\cos\theta_r\hat{\mu} + \sin\theta_r\hat{\lambda}) + (Y_r - Y_c)(\sin\theta_r\hat{\mu} - \cos\theta_r\hat{\lambda}) \\ y_e =& (X_c - X_r)(\sin\theta_r\hat{\mu} - \cos\theta_r\hat{\lambda}) + (Y_r - Y_c)(\cos\theta_r\hat{\mu} + \sin\theta_r\hat{\lambda}) \end{aligned} \tag{4–26}$$

We define a second-order observer state variable as:

$$\begin{aligned} m_1 &= \sin\theta_e - K_4 y_e \\ m_2 &= \cos\theta_e - K_5 x_e \end{aligned} \tag{4–27}$$

where $K_o = [K_4, K_5]$ is postive observer gain. The derivative of (4–27) along (4–6) is given by:

$$\begin{aligned} \dot{m_1} &= \cos\theta_e(\omega_r - \omega_c) - K_4(-x_e\omega_c + v_r \sin\theta_e) \\ \dot{m_2} &= \sin\theta_e(\omega_c - \omega_r) - K_5(y_e\omega_c - v_c + v_r\cos\theta_e) \end{aligned} \tag{4–28}$$

then the yaw error obersver can be designed as:

$$\begin{cases} \dot{\hat{m}_1} = \hat{\mu}(\omega_r - \omega_c) - K_4 v_r \hat{\lambda} + K_4 x_e \omega_c \\ \dot{\hat{m}_2} = \hat{\lambda}(\omega_c - \omega_r) - K_5(-v_c + v_r\hat{\mu} + y_e\omega_c) \end{cases} \tag{4–29}$$

where

$$\begin{cases} \hat{\lambda} = \hat{m_1} + K_4 y_e \\ \hat{\mu} = \hat{m_2} + K_5 x_e \end{cases} \tag{4–30}$$

Next, we verify the stability of the YEO using the Lyapunov direct method. With observer errors $\tilde{\lambda} = \lambda - \hat{\lambda}$ and $\tilde{\mu} = \mu - \hat{\mu}$, a candidate Lyapunov is chosen as:

$$L_3 = \frac{1}{2}\tilde{\lambda}^2 + \frac{1}{2}\tilde{\mu}^2 \tag{4–31}$$

The derivatives of observer errors along (4–29) and (4–30) are:

$$\begin{aligned} \dot{\tilde{\lambda}} &= (\omega_r - \omega_c)\tilde{\mu} - K_4 v_r \tilde{\lambda} \\ \dot{\tilde{\mu}} &= (\omega_c - \omega_r)\tilde{\lambda} - K_5 v_r \tilde{\mu} \end{aligned} \tag{4–32}$$

Differentiating L_3 with respect to time along the (4–32) yields:

$$\dot{L_3} = \tilde{\lambda}\dot{\tilde{\lambda}} + \tilde{\mu}\dot{\tilde{\mu}} = -K_4 v_r \tilde{\lambda}^2 - K_5 v_r \tilde{\mu}^2 \tag{4–33}$$

We find L_3 is PD and $\dot{L_3}$ is ND because K_o and the value of v_r are both non-negative.

This implies that observer errors will converge to zero asymptotically as $t \to \infty$. Therefore, the problem of unmeasurable yaw information is settled by using known reference variables and available position measurements. We shall transfer the estimated variables from YEO into the entire closed-loop controller so that actual control commands in (4–24) and (4–25) may be rewritten as follows:

$$\begin{cases} v_c = v_r\hat{\mu} + \tau x_e \\ \delta_c = \arctan \dfrac{[v_r y_e + \omega_r \hat{\mu} + \tau\phi v_r \hat{\lambda} + \tau\phi\eta v_r(\hat{\lambda} + \tau\phi y_e)]l}{(\hat{\mu} + \tau\phi x_e)v_c} \end{cases} \tag{4–34}$$

Remark 1. *Note that x_e and y_e in (4–34) are computed instantly from (4–26) instead of errors defined in (4–4), when $\hat{\lambda}$ and $\hat{\mu}$ are estimated by means of YEO. The posture error is redefined as $P_e{}' = [x_e\ y_e\ \hat{\lambda}\ \hat{\mu}]^T$.*

Remark 2. *This design of observer-controller combination, differing vastly from conventional observability problems, means that we construct observer and controller separately.*

4.2.3 Optimal Tracking System based on PSO

In nature, for a swarm of birds, the most effective and simplest foraging strategy is to explore the surrounding space where a leader is closest to the food. PSO algorithm is inspired by the simulation of the abovementioned intelligent behavior characteristics of birds flocking, and it is used to solve optimization problems. Each bird in the swarm is regarded as a particle, representing a candidate solution and maps a fitness value computed from the fitness function. Regarding the moving direction and distance of a particle, it is decided by its velocity that is dynamically adjusted with the moving experience of itself and its neighbor so as to seek the optimal solution for an individual in the search space. Hence, the PSO algorithm is introduced into the tracking system for the purpose of accomplishing optimal tracking, as shown in Figure 4.3. In other words, the best solution for control gain (particle) can be determined by deploying PSO.

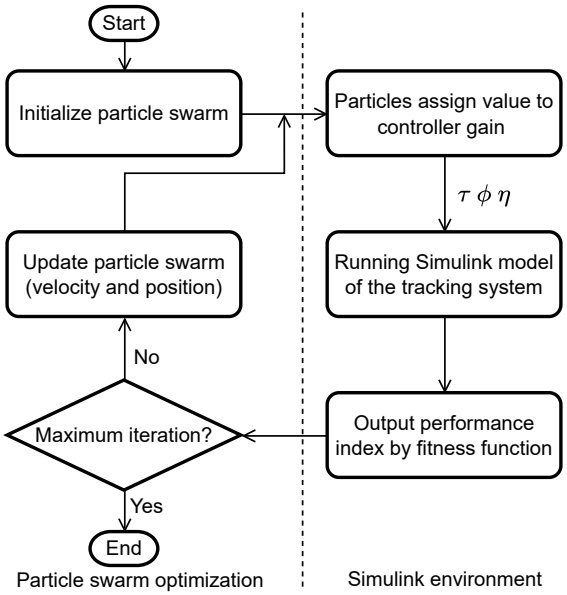


Figure 4.3: The flow chart of the proposed PSO.

Futhermore, each particle i possesses a current velocity vector $V_i = [V_{i1}, V_{i2}, V_{i3}]$ and a current position vector $X_i = [X_{i1}, X_{i2}, X_{i3}]$. The update of particle's velocity and position in each iteration are according to best position of particle P_b and that of whole swarm G_b, which are given as follows:

$$V_{i+1} = WV_i + C_1R_1(P_b - X_i) + C_2R_2(G_b - X_i) \tag{4–35}$$

$$X_{i+1} = X_i + V_{i+1} \tag{4–36}$$

A linear decreasing inertia weight method is introduced in PSO, where it maintains the extensive global exploration at the early stage and significantly strengthens the local search ability during the latter part. The mathematical formula of this method is given in the following form:

$$W = W_{max} - (W_{max} - W_{min})Iter/Iter_{max} \tag{4–37}$$

where C_1 and C_2 are the acceleration coefficients, and R_1 and R_2 denote random numbers in the range [0,1]. The term W within [W_{min},W_{max}] interval is inertia weight factor, and $Iter$ and $Iter_{max}$ are the current and maximum iteration number.

After generating an initial population, the tuning parameter of $\psi = [\tau\ \phi\ \eta]$ should be obtained by a zoom factor $f = [f_1, f_2, f_3]$ so that all particles fly in a limited searching space, which can be formulated as follows:

$$\psi = X_i \times f \tag{4–38}$$

The process of deploying PSO can be described as:

(1)Generate the initial population, each particle in the swarm has a random velocity and position, and particles assign the value to the controller gains in order.

(2)Running the Simulink model. Meanwhile, the fitness of particles is evaluated by the fitness function designed in advance. Thus, P_b and G_b are obtained.

(3)Compare the fitness value of each particle with that of P_b. If a smaller value of fitness exists, the best position is updated by this particle. A similar comparison of each particle and the whole swarm is carried out. The best global position remains the same, on condition that it corresponds to the minimum fitness value.

(4)Update the velocity and position for the particle through (4–35) and (4–36).

(5)Judge the termination condition. The procedures mentioned above are repeated until the maximum iteration is reached.

In order to evaluate the tracking performance, we design a fitness function as a criterion as following:

$$f(z) = \sum_{i=0}^{t} z_i = \sum k_a[k_b(|x_e| + |y_e| + \left|\hat{\lambda}\right| + |\hat{\mu}|) + k_c\,|v_r - v_c| + k_d\,|\delta_r - \delta_c|] \tag{4–39}$$

where k_a represents the scaling factor. The constants k_b, k_c, k_d stand for the posture, velocity, and steering error weight respectively. The parameter t is the simulation or experiment time duration of the tracking. The fitness function computes the sum of the fitness value at each sampling time. Afterwards, the minimum of $f(z)$ and the corresponding best combination of controller gains are obtained by implementing the PSO algorithm. Accordingly, the optimal global tracking performance is ultimately fulfilled. Equation (4–39) implies that not only do we judge the tracking performance based on the posture error, but also velocity and steering error are taken into account, serving as a performance criterion in an effort to guarantee the least tracking error and minimum control inputs simultaneously.

Remark 3. *The particle (controller gain) and output performance index can provide a bridge to make connections between the PSO algorithm and the Simulink model.*

Remark 4. *The tracking performance evaluation criterion can be redefined by simply tuning the error weights in (4–39).*

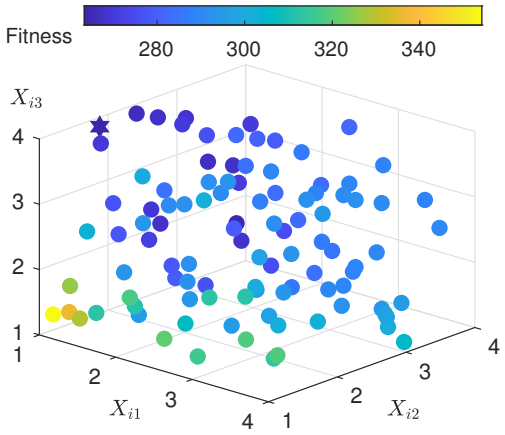
4.3 Simulation Results

In this section, simulations were carried out through MATLAB/Simulink to evaluate the tracking performance. To illustrate the advantages of the suggested approach, comparative simulations were conducted with three different controllers. The well-regarded controller mentioned in (4–7) was marked as controller A, while a robust nonlinear backstepping controller introduced in [38] and a recently developed adaptive controller proposed in [37] were labeled as B and C, respectively. A smooth circular reference trajectory is applied as:

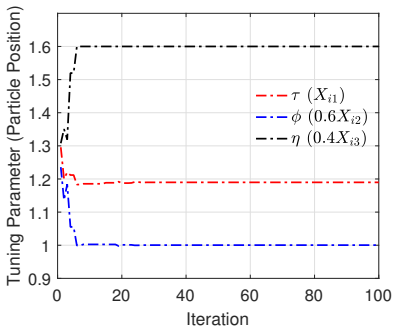
$$\begin{aligned} X_r =& 1.8\sin(0.35t + 0.2) + 0.8 \\ Y_r =& -1.8\cos(0.35t + 0.2) \end{aligned} \tag{4–40}$$

where the sampling time and t are 0.002 s and 16 s. The initial conditions of the vehicle and reference trajectory are $P_c = [0, 0, 0]^T$ and $P_r = [1.16, -1.76, 0.2]^T$. The observer gain was set as $K_o = [7, 7]$. The parameter of fitness function was chosen as $k = [k_a, k_b, k_c, k_d] = [0.1, 0.74, 0.13, 0.13]$. In addition, parameters of PSO are listed as follows: $W_{min} = 0.82$, $W_{max} = 1.18$, $C_1 = C_2 = 1.5$, $Iter_{max} = 10$, $V_i \in [-0.15, 0.15]$, $X_i \in [1, 4]$, and $f = [1, 0.6, 0.4]$. Following the steps of Section 4.2.3, we obtain the optimal gain as $K = [1.19, 0.75, 1.2]$. For the sake of fair comparisons, the control gain for other controllers was chosen as 1.5 to get acceptable tracking results. Also, threshold distance d_x and tuning parameter α in Controller C are as the same as in [37].

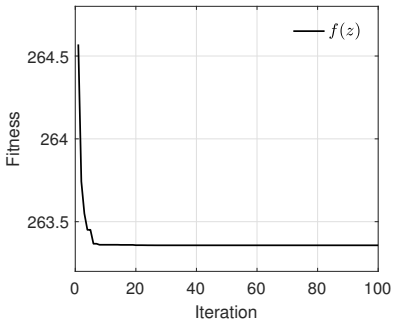
The simulation results of PSO are shown in Figure 4.4. The initial population (circle) with fitness distributed randomly over the searching space, as seen in Figure 4.4(a), where the best position (hexagram) having the lowest fitness is found effortlessly by the PSO, at nearly $G_b = [1.19, 1.67, 4]$. It can be observed that fitness depends to a great extent on the value of K_2 and K_3, which signifies that they play a dominant role in tracking. Noted that we still present the optimization results from the 0th to 100th iterations so that we can explicitly describe how does the proposed PSO model find the optimal solution in a short time. Provided in Figure 4.4(b) is information about optimizing tuning parameters, revealing that the PSO is constantly seeking an optimal position. We detect the values of τ, ϕ and η all changed marginally after the sixth iteration, standing at $\psi = [1.19, 1, 1.6]$. This can also be proved in Figure 4.4(c), where the value of $f(z)$ is on a downward trend and reaches a trough of only 263.36 fitness (hexagram) at the 6th iteration, at which point it leveled off until the last iteration. The reasons for this phenomenon can be attributed to: we specify small-scale boundaries of X_i and design a fitness function to output performance



(a)

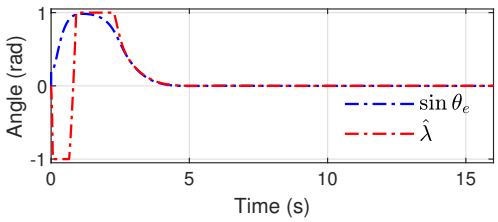


(b)

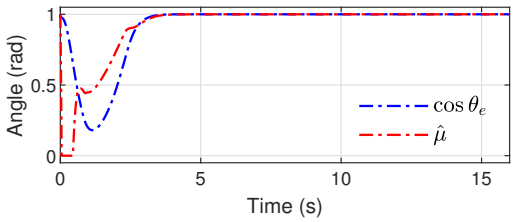


(c)

Figure 4.4: Results of particle swarm optimization.



(a)



(b)

Figure 4.5: The outputs of the yaw error observer.

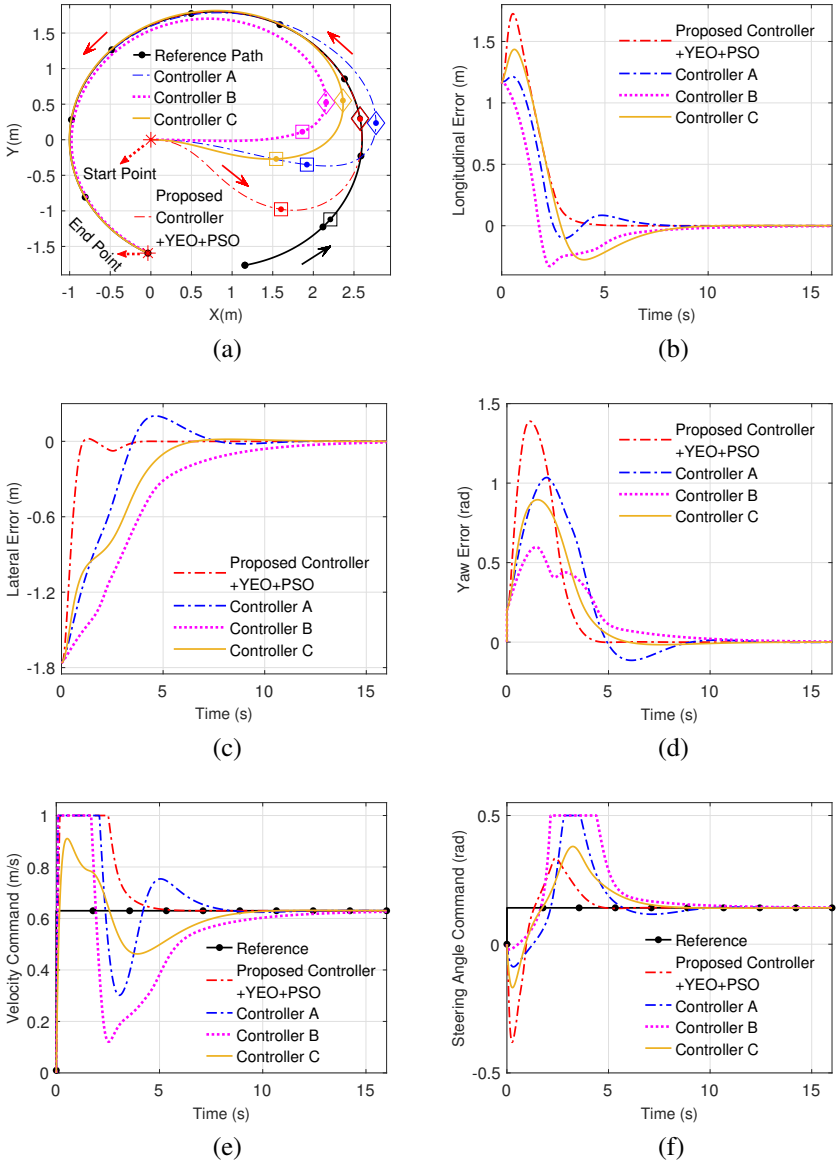


Figure 4.6: Results for tracking a circular trajectory.

index at each sampling time so as to considerably differ fitness values at each particle's position, which hastens the progress of the PSO and therefore facilitates exploration of superior space in the early phase. From Figure 4.5, the estimated value $[\hat{\lambda}, \hat{\mu}]$ converges quickly to the actual value $[\sin\theta_e, \cos\theta_e]$, which implies that YEO meets the requirement of asymptotical stability.

The simulation results are summarized in Figure 4.6. From an overall perspective, all controllers enable the car-like vehicle to realize trajectory tracking ($P_e \to 0$) but with diverse performance. Figure 4.6(a) displays the actual and reference trajectory. The proposed controller drives the vehicle near the target point at $t = 2$ s (square), while the results by other controllers are not ideal because of too late steering. At $t = 4$ s (diamond), Controller B and C lead the vehicle to track inside the reference trajectory, and Controller A makes it outside. However, given in Figure 4.6(a)–4.6(d), the deviations and posture errors neither exist nor appear in the tracking period (during 4-16 s) by utilizing the proposed controller. Figure 4.6(b)–4.6(d) present the posture error versus time, indicating that the proposed controller offers a much faster response and convergence rate than other controllers. A striking improvement is that lateral error converges swiftly to zero at $t = 3.5$ s and keeps steady until the end. By contrast, Controller C requires 6.7 s and Controller A about 13 s to achieve this stable status. Despite having a minor magnitude of yaw error (during 0-2.7 s), Controller B results in the worst tracking on account of the slowest convergence rate. For Controller C, it adjusts the vehicle to track at a moderate convergence rate due to the design of adaptive gains. Nonetheless, selecting of the value of d_x and α is a challenge, which directly influences the longitudinal error and velocity tracking.

The commanded velocity and steering angle within bounds, as well as desired commands, are presented in Figure 4.6(e) and 4.6(f). At time $t \in [0, 4]$, the proposed controller outputs a larger velocity command so that the vehicle approaches the desired point at a fast pace. Simultaneously, the steering angle command has a dramatic change, especially in

the first 0.27 s, which quickly regulates vehicle's orientation for lateral error to converge speedy to zero, as evident in Figure 4.6(a) and 4.6(c). However, it takes more than 8 s for other controllers to control the vehicle to follow the circular trajectory with the desired velocity and steering angle commands.

Remark 5. *In spite of the fact that the gains in YEO are chosen at random, we can obtain optimized observer gains by expanding the dimension of V_i and X_i.*

Remark 6. *Compared with conventional PSO, fewer iterations have remarkably reduced the high computational cost.*

4.4 Experimental Results

A series of experimental tests were performed on the QCar platform so as to examine and evaluate the tracking performance of the four proposed controllers. To further illustrate the superiority of the proposed control strategy, we contrasted and analyzed experimental results. The test video is available in https://www.youtube.com/watch?v=6VmLnf0X288. After operating the Simulink model, the optimized tuning parameters were attained as $\psi = [1.5225, 1.6551, 1.2]$ for Scenario I and $\psi = [2.0306, 2.4, 1.6]$ for Scenario II.

4.4.1 Scenario I

A circular-like trajectory providing slightly varied velocity and steering angle was adopted, which was designed as:

$$\begin{aligned} X_r =& 1.1\sin(0.5t - 0.2) \\ Y_r =& 1.165\cos(0.5t - 0.2) \end{aligned} \tag{4–41}$$

The initial states of the desired and current posture are $P_r = [-0.2185, 1.1418, 0.211]^T$ and $P_c = [-1.165, 0, 0]^T$. The observer gain was chosen as $K_o = [6.669, 7.051]$. The

fitness parameters were designed as $k = [0.1, 0.5, 0.25, 0.25]$, and $f = [1, 0.5, 0.3]$. The suitable gain for the compared controllers was set to 1.5525, and d_x and α were selected as 1.5 and 1. Finally, t was tuned to 11.6 s. The remaining parameters are the same as in Section 4.3.

Figure 4.7 illustrates the tracking results of a circular-like trajectory. In general, the QCar by the proposed controller has the slightest deviation, particularly at the early stage, exemplified by two tracking moments at $t = 2$ s (diamond) and $t = 4$ s (square). The Controller A is less desirable due to more oscillations and the significant deviation during 2-8 s, guiding the QCar tracking outside the reference target. A distinct phenomenon is that all controllers, except for the proposed one, give rise to a late steering and slower convergence rate in the first six seconds, as evident in Figure 4.7(a)–4.7(d); where the proposed controller makes the stable times of the condition ($P_e \rightarrow 0$) are reduced considerably to 2.9, 3.8 and 5 s compared to relatively good Controller C corresponding to 6, 8 and 6 s.

As shown in Figure 4.7(e) and 4.7(f), the proposed control scheme is able to implement velocity and steering tracking expeditiously since the performance evaluation criterion is revised by turning up the error weights in (4–39). Figure 4.7(g) presents the results of YEO, which demonstrates that accurate estimations are rapidly acquired. Despite existing large initial differences between estimated values and sensor-based measurements, the former fastly approaches the latter. So, exact estimations are obtained from the time $t = 2$ s. The evolutions of adaptive gains are depicted in Figure 4.7(h), where the value of gains is tuned automatically to boost tracking performance. For example, there is an increase of K_2 at time $t \in [0, 2]$, which is beneficial to accelerate the convergence of the lateral error. As expected, y_e slumps quickly, and this is fulfilled because of a larger steering angle, as seen in Figure 4.7(f). Meanwhile, a similar trend of K_3 can be observed, which results in a faster convergence rate of yaw error. For these two reasons, the value of θ_e declines dramatically to -0.96 at 1.13 s, with a sharp surge from this moment to $t = 2$ s. That

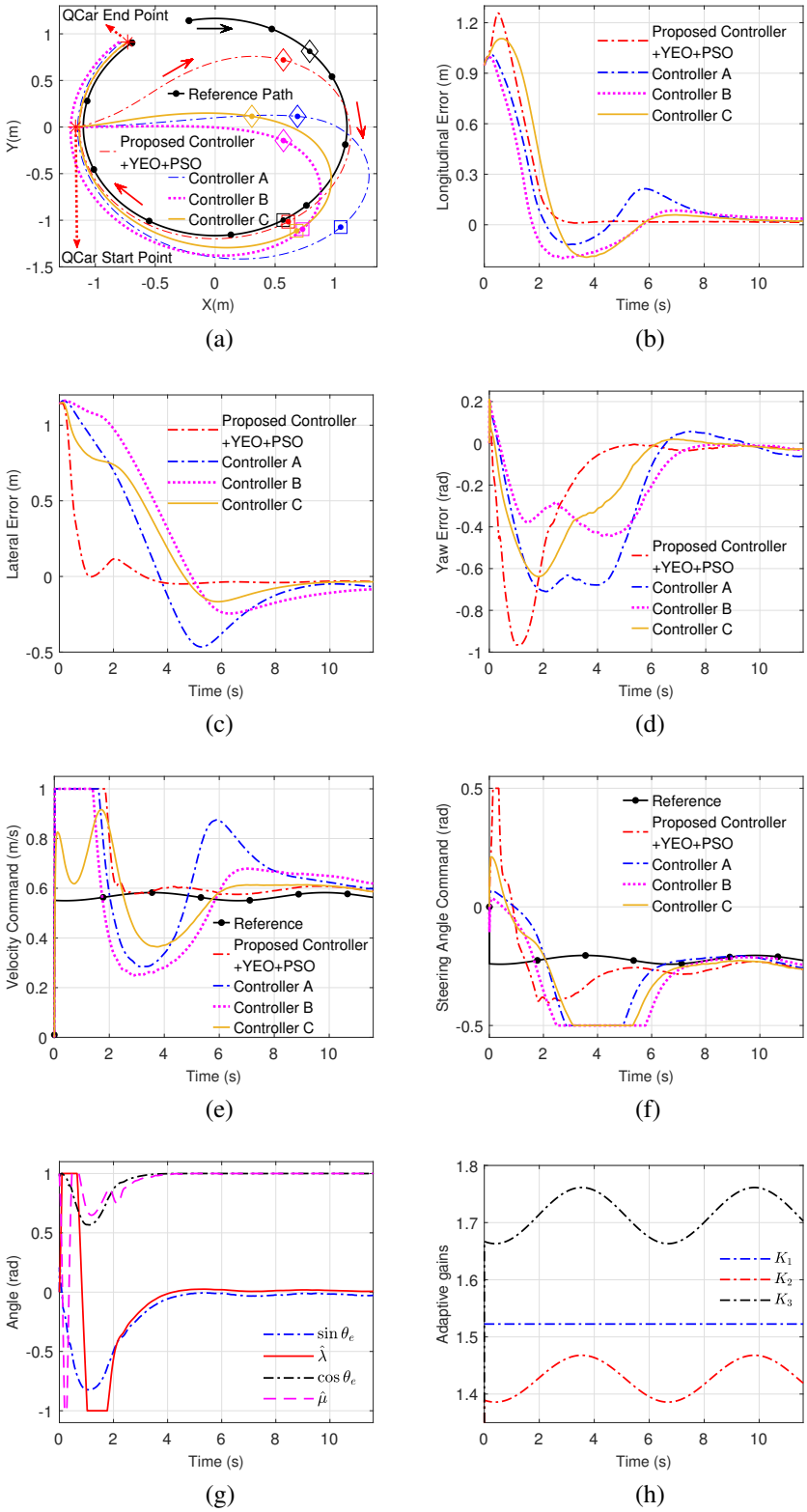


Figure 4.7: Tracking results of Scenario I (under large initial position errors).

is to say, the QCar rapidly narrows position errors through a large steering and regulates its orientation, such that $\theta_e \to 0$. Consequently, the K_2 and K_3 can compensate for each other so that the proposed controller offers the fastest overall convergence rate and the most precise tracking.

4.4.2 Scenario II

Another S-shape trajectory with higher tracking requirements was considered as a reference trajectory, formulated as:

$$\begin{aligned} X_r =& 2.2 - 2.5\cos(0.2t + 0.15) \\ Y_r =& 0.7 + \sin(0.4t + 0.15) \end{aligned} \tag{4–42}$$

where the changed parameter settings are listed as follows: $t = 19$, $d_x = 0.75$, $\alpha = 1.5$, $k = [0.1, 0.35, 0.3, 0.3]$, and $K_o = [6.65, 6.01]$. The initial condition $P_e = [0.73, 0.85, 1.39]^T$ provides a large yaw error. The rest of the parameters were set the same as those in Section 4.3. Note that the error weights are similar, implying that the velocity and steering tracking are deemed as equally important as posture tracking.

The results of tracking an S-shape trajectory are shown in Figure 4.8, where the QCar with the proposed controller realizes tracking with the fastest convergence rate, highest accuracy and minimum cutting corner. Controller A shows the worst tracking performance, while Controller B and C have similar properties. Figure 4.8(a) presents the actual trajectory of the QCar. When the QCar utilizing other methods drives from its start point to the position at $t = 3$ s (marked as diamond), cutting-corner errors are noticeably widened. This may have a detrimental impact on the control stability because the reference posture moves 'behind' the real posture. From an algorithmic view, it only executes $P_e \to 0$ and is unable to contemplate the vehicle's current state, which sometimes makes unreasonable vehicle control, such as overlarge steering, non-stop steering, etc. Apart from the proposed

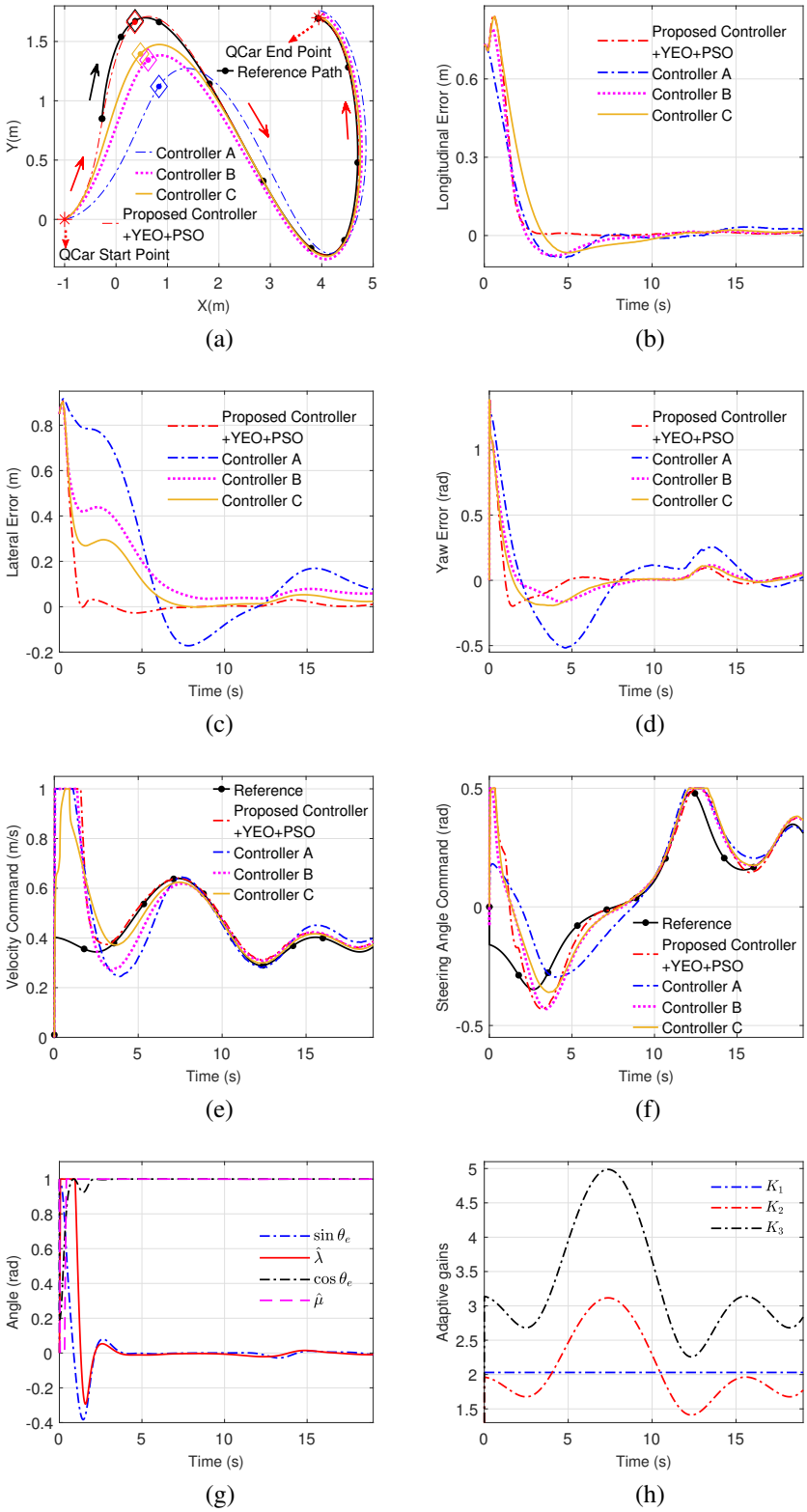


Figure 4.8: Tracking results of Scenario II (under large initial yaw errors).

method, all controllers lead to insufficient steering, which means the QCar cuts corners or drives not far enough. By contrast, almost no deviation emerging after the QCar with the proposed controller moves to the reference start point. Besides, a conspicuous feature is that y_e is eliminated quite rapidly to avoid cutting-corner errors. These tracking characteristics can be attributed to the proposed controller offering rapid convergence rates, that is, the posture errors converge to zero at approximately 4, 1.3, and 4.6 s, whereas it spends more than 7.2, 7.3, and 8 s for other controllers, as evident in Figure 4.8(b)–4.8(d).

The commands of velocity and steering angle are depicted in Figure 4.8(e) and 4.8(f). Using the proposed controller, the QCar makes full use of the driving and steering capability during the early stage to drive to desired positions and regulates orientation simultaneously in a short time. See Figure 4.8(g) for the estimations and sensor-based measurements. Despite an initial error, the YEO quickly estimated the relevant inputs required for the proposed controller. The adaptive control gains can further minimize the cutting-corner error due to the fact they are tuned to match current velocity and reduce inappropriate steering responses at corners, as evident in Figure 4.8(e) and 4.8(h).

Chapter 5

Trajectory Tracking for Autonomous Vehicles based on Frenet Frame

5.1 Problem Formulation

5.1.1 Dynamic Bicycle Model

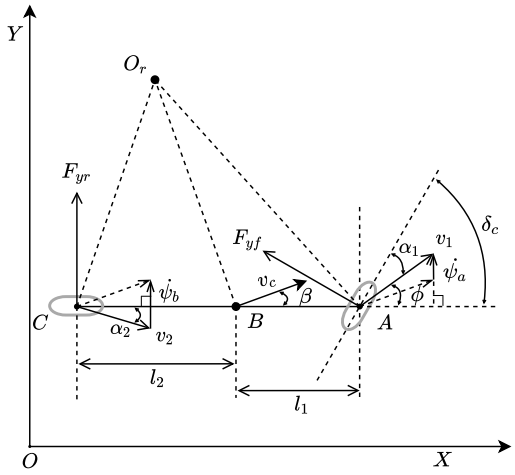


Figure 5.1: Dynamic bicycle model.

While most geometric and kinematic controllers neglect dynamic effects, the internal

forces, accelerations, and tire deformation problem should be taken into account at high speed. The control objective is to develop a tracking system for AVs in a wide-speed range, so a bicycle model with a simplified configuration for providing dynamic responses and vehicle states is utilized, as shown in Figure 5.1. In this study, the steering angle δ_c governed by the front wheel is assumed to be small, while the rear wheel is set to be fixed. Consequently, it is reasonable to deem that $\cos\delta_c \approx 1$, $\tan\phi \approx \phi$, and $\tan\alpha_2 \approx \alpha_2$. The equation of lateral translational motion and yaw dynamics are obtained as:

$$\begin{aligned} ma_y &= F_{yf}\cos\delta_c + F_{yr} = C_1\alpha_1 + C_2\alpha_2 \\ I\ddot{\psi} &= l_1F_{yf}\cos\delta_c - l_2F_{yr} = l_1C_1\alpha_1 - l_2C_2\alpha_2 \end{aligned} \tag{5–1}$$

After analyzing the geometric relationship of velocity vectors and using small-angle approximations, the slip angle of the front and rear wheels are expressed by:

$$\begin{aligned} \alpha_1 &= \phi - \delta_c = (v_y + \dot{\psi}_a)/v_x - \delta_c \\ \alpha_2 &= (v_y - \dot{\psi}_b)/v_x \end{aligned} \tag{5–2}$$

where

$$\begin{aligned} \dot{\psi}_a &= l_1\dot{\psi} \\ \dot{\psi}_b &= l_2\dot{\psi} \\ v_x &= v_c\cos\beta \\ v_y &= v_c\sin\beta \end{aligned} \tag{5–3}$$

Substituting from (5–2) and (5–3) into (5–1), with the Frenet formula $a_y = \ddot{y} + v_x\dot{\psi}$, the state space model is given by:

$$\begin{bmatrix} \ddot{y} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} \dfrac{C_1+C_2}{mv_x} & \dfrac{l_1C_1 - l_2C_2}{mv_x} - v_x \\ \dfrac{l_1C_1 - l_2C_2}{Iv_x} & \dfrac{l_1^2C_1 + l_2^2C_2}{Iv_x} \end{bmatrix} \begin{bmatrix} \dot{y} \\ \dot{\psi} \end{bmatrix} - \begin{bmatrix} \dfrac{C_1}{m} \\ \dfrac{l_1C_1}{I} \end{bmatrix} \delta_c \tag{5–4}$$

where a_y represents the inertia acceleration of the vehicle at c.g. along the lateral axis of the vehicle body frame, β the sideslip angle, y the lateral position, ψ_c the yaw angle, v_x the longitudinal velocity, v_y the lateral velocity (same as $\dot{y}$), and v_c the velocity at the c.g. of the vehicle. The terms v_1, F_{yf}, and C_1 stand for the velocity, lateral force, and cornering stiffness of the front tire, while v_2, F_{yr}, and C_2 are those of the rear tire. The vehicle mass and yaw moment of inertia is denoted by m and I respectively. The distances from the c.g. to the front and rear tires are l_1 and l_2. The intersection O_r is the instantaneous rolling center for the vehicle, and O_rB describes the turning radius.

5.1.2 Coordinate Transformation based on Frenet Frame

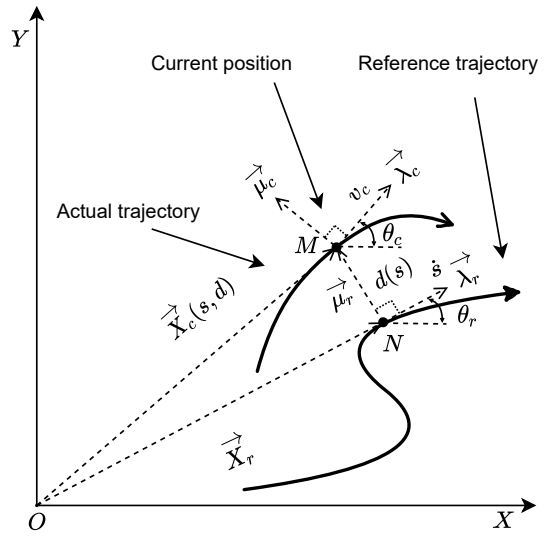


Figure 5.2: The motions of the vehicle under the Frenet frame.

The longitudinal and lateral control are connected together and influence each other on a kinematic level. Nonetheless, the motions of the vehicles are able to be decoupled by integrating the Frenet frame with the utilized dynamic model. Consider the vehicle's motion under the global and Frenet frame, as depicted in Figure 5.2. From which, the reference trajectory as the road centerline determines the s-axis, describing the driving distance s from the initial reference point, and d-axis maps the lateral offset d. Point $M(x_c, y_c)$ represents

the vehicle's current position, and $N(x_r, y_r)$ is the projected position on the reference trajectory. The unit vectors of the current vehicle position, namely $\vec{\lambda_c}$ and $\vec{\mu_c}$, are constructed along the Frenet frame to analyze the vehicle motion. Note that the projection point is assumed to coincide with the reference point at this stage. The direction of $\vec{\lambda_c}$ and v_c are the same, but $\vec{\mu_c}$ is perpendicular to $\vec{\lambda_c}$. Similarly, the vector $\vec{\lambda_r}$ and $\vec{\mu_r}$ are established for the projected point, of which exact coordinates can be described by the definition of the orthonormal basis as:

$$\begin{aligned} \vec{\mu_r} &= (-\sin\theta_r, \cos\theta_r) \\ \vec{\lambda_r} &= (\cos\theta_r, \sin\theta_r) \end{aligned} \tag{5–5}$$

and the coordinate of the vector $\vec{q}$ can be written as:

$$\vec{q} = \vec{NM} = (x_c - x_r, y_c - y_r) \tag{5–6}$$

The variable $\dot{s}$, having the same direction as $\vec{\lambda_r}$, describes the velocity of the projection point in the Frenet frame. The term θ_c is called the course angle of the vehicle, whereas θ_r denotes the angle between the X-axis and tangent of the projection point. By means of vector methods, the projection distance from the point N to M may be defined as:

$$d = (\vec{X_c} - \vec{X_r}) \cdot \vec{\mu_r} \tag{5–7}$$

then take the derivative of (5–7) yields:

$$\dot{d} = (\vec{X_c} - \vec{X_r}) \cdot \dot{\vec{\mu_r}} + (\dot{\vec{X_c}} - \dot{\vec{X_r}}) \cdot \vec{\mu_r} \tag{5–8}$$

where the derivatives of the real vehicle's position vector $\dot{\vec{X}}_c(s, d)$ and projective counterpart can be expressed as:

$$\begin{aligned}\dot{\vec{X}}_c &= |v_c|\,\vec{\lambda}_c \\ \dot{\vec{X}}_r &= \dot{s}\vec{\lambda}_r\end{aligned} \tag{5–9}$$

Next, we introduce the 2-dimensional Frenet equations so as to derive the differential equation of the lateral deviation, which is given by the following equation:

$$d\vec{\mu_r}/ds = -k_r\vec{\lambda_r} \tag{5–10}$$

which upon closer inspection is seen to be:

$$d\vec{\mu_r}/dt = d\vec{\mu_r}/ds \times ds/dt = -k_r\dot{s}\vec{\lambda_r} \tag{5–11}$$

where the curvature for the reference trajectory is k_r. After applying the vector operations, the explicit expression for $\dot{d}$ and $\dot{s}$ can be written as:

$$\begin{aligned}\dot{d} &= v_x \sin(\psi - \theta_r) + v_y \cos(\psi - \theta_r) \\ \dot{s} &= [v_x \cos(\psi - \theta_r) - v_y \sin(\psi - \theta_r)]/(1 - k_r y_e)\end{aligned} \tag{5–12}$$

5.1.3 Tracking Error Dynamics

Defining the difference between the actual and the pre-determined trajectory is essential for evaluating the proposed controller's tracking ability. In Figure 5.3, according to the definition of the Frenet frame, the longitudinal and lateral tracking errors shall be defined as:

$$\begin{aligned}x_e &= \vec{q} \cdot \vec{\lambda_r}^T \\ y_e &= d = \vec{q} \cdot \vec{\mu_r}^T\end{aligned} \tag{5–13}$$

More specifically, the longitudinal error is the arc length from the point N to P, regarded as approximately equal to the projection of $\vec{q}$ along the direction of $\vec{\lambda_r}$. With respect

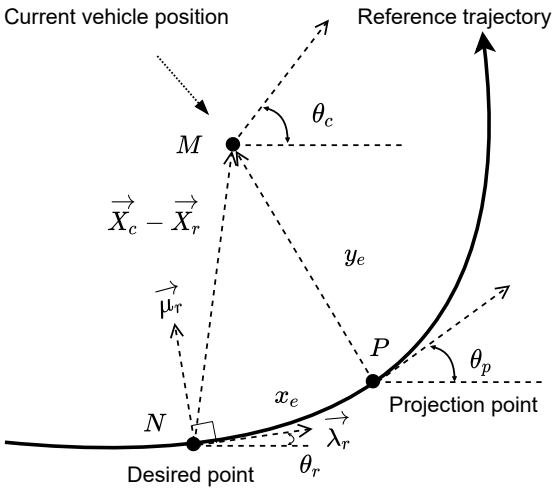


Figure 5.3: Tracking errors in the Frenet frame.

to the lateral error, it is roughly equivalent to the projection of $\vec{q}$ in the direction of $\vec{\mu_r}$. Commonly the value of $\psi - \theta_r$ is considered to be small. Besides, the second derivative of θ_r is negligible because the curvature of roads changes gradually in the real world. Upon the foregoing premises, the lateral error, course error, and their derivatives can be formulated as:

$$\begin{aligned} \dot{y_e} &= \dot{y} + \psi_e v_x \\ \ddot{y_e} &= \ddot{y} + \dot{\psi_e} v_x \\ \dot{\psi_e} &= \dot{\psi} - \dot{\theta_r} \\ \ddot{\psi_e} &= \ddot{\psi} \end{aligned} \tag{5–14}$$

where

$$\psi_e = \psi - \theta_r \tag{5–15}$$

Note that ψ_e is the virtual course error rather than the real counterpart. Substituting (5–14) and (5–15) into (5–4), the state space model for lateral dynamics is rewritten in a compact form as:

$$\dot{L_e} = AL_e + Bu + C\dot{\theta_r} \tag{5–16}$$

where

$$\begin{aligned} L_e &= \begin{bmatrix} y_e \\ \dot{y_e} \\ \psi_e \\ \dot{\psi_e} \end{bmatrix} B = \begin{bmatrix} 0 \\ -\dfrac{C_1}{m} \\ 0 \\ -\dfrac{l_1 C_1}{I} \end{bmatrix} C = \begin{bmatrix} 0 \\ \dfrac{l_1 C_1 - l_2 C_2}{m v_x} - v_x \\ 0 \\ \dfrac{l_1^2 C_1 - l_2^2 C_2}{I v_x} \end{bmatrix} \\ A &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \dfrac{C_1 + C_2}{m v_x} & -\dfrac{C_1 + C_2}{m} & \dfrac{l_1 C_1 - l_2 C_2}{m v_x} \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{l_1 C_1 - l_2 C_2}{I v_x} & \dfrac{l_2 C_2 - l_1 C_1}{I} & \dfrac{l_1^2 C_1 - l_2^2 C_2}{I v_x} \end{bmatrix} \end{aligned} \tag{5–17}$$

5.1.4 Problem Statement

The tracking control problem has now transferred to design an appropriate control law to guide the vehicle to precisely track the pre-defined trajectory, that is, the proposed trajectory tracking scheme needs to guarantee that the tracking errors converge to zero stably as time goes to infinity.

5.2 Trajectory Planning

The fifth-order polynomial featured with the constraints of the curvature, position, velocity, and acceleration is adopted for planning a feasible reference trajectory as:

$$\begin{aligned} x_r(t) &= a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \\ y_r(x_r) &= b_0 + b_1 x_r + b_2 x_r^2 + b_3 x_r^3 + b_4 x_r^4 + b_5 x_r^5 \end{aligned} \tag{5–18}$$

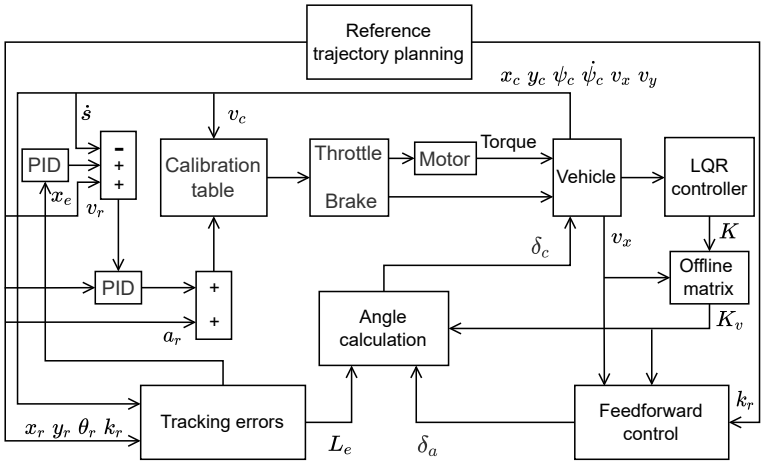


Figure 5.4: The architecture of the proposed dynamic-based tracking scheme.

where a_i and b_i are unknown coefficients of the polynomial. The desired trajectory requires boundary conditions, which can be formulated as follows:

$$\begin{aligned} &x_r(0)\ \dot{x_r}(0)\ \ddot{x_r}(0) \\ &x_r(T)\ \dot{x_r}(T)\ \ddot{x_r}(T) \\ &y_r(0)\ \dot{y_r}(0)\ \ddot{y_r}(0) \\ &y_r(x_T)\ \dot{y}(x_T)\ \ddot{y}(x_T) \end{aligned} \tag{5–19}$$

where the ending time is represented by T and (x_r,y_r) is the ending coordinate. Once the boundary conditions are set, the unknown coefficients can be derived. Based on these, the time function of $y_r(t)$ and its derivatives can be written as:

$$\begin{aligned} y_r(t) &= y[x_r(t)] \\ \dot{y_r}(t) &= y_r'[x_r(t)]\dot{x_r}(t) \\ \ddot{y_r}(t) &= y_r''[x_r(t)]\dot{x_r}(t)^2 + y'[x_r(t)]\ddot{x_r}(t) \end{aligned} \tag{5–20}$$

Differentiating (5–18) and its derivatives, the desired course angle and the road curvature versus time can be expressed as:

$$\begin{aligned} \theta_r(t) &= \arctan\left(y_r'[x_r(t)]\right) \\ k_r(t) &= y_r'' x_r(t)/(1 + y_r'[x_r(t)]^2)^{1.5} \end{aligned} \tag{5–21}$$

In regard to the desired velocity, acceleration, and the velocity tracking error, they can be described as:

$$\begin{aligned} v_r &= \sqrt{\dot{x_r}(t)^2 + \dot{y_r}(t)^2} \\ a_r &= \sqrt{\ddot{x_r}(t)^2 + \ddot{y_r}(t)^2} \\ v_e &= v_r - \dot{s} \end{aligned} \tag{5–22}$$

Based on the geometric relations in Figure 5.3, the estimated course angle function of projected points is designed to mitigate the chattering phenomena and achieve smooth steering, which is:

$$\theta_{re} = \theta_r + k_r x_e \tag{5–23}$$

5.3 Tracking Control Strategy

The schematic of the closed-loop tracking system is displayed in Figure 5.4. Due to the decomposition of vehicle motions, it is clear that we can take advantage of different sorts of controllers to complete the tracking mission. In this study, the conventional PID and LQR controls are chosen to systematically govern the vehicle's longitudinal and lateral motions.

5.3.1 LQR Control

The LQR control, characterized by stable control performance, can find the optimal solution for the control law. The state feedback law $u = -Kx$ minimizes the quadratic cost function, which is defined as:

$$J(u) = \frac{1}{2}\int_0^\infty [L_e^T(t)QL_e(t) + u^T(t)Ru(t)]dt \tag{5–24}$$

subject to the tracking system dynamics:

$$\dot{L}_e = AL_e + Bu \tag{5–25}$$

where Q and R stand for positive definite matrices and off-diagonal elements, t the time, and $K = [k_1, k_2, k_3, k_4]$ the control gain of the LQR controller. From (5–16) and (5–17), it is obvious that the term v_x is the only variable in the lateral error dynamics. Therefore, by utilizing the simulation software, all the solutions of LQR under various velocity can be computed in an offline manner and stored in a matrix K_v. That is, each value of the velocity corresponds to a specific set of K. As a result, with the associated Riccati equations, the LQR gives the optimal steering command with the minimal cost in real-time by indexing the matrix elements. The computational time and tuning efforts are considerably reduced due to the fact that the control gains can tune automatically. However, there is a challenge that stems from selecting the weights of Q and R: a larger Q corresponds to a superior performance while sacrificing stability. Compared to a small Q, the high value of R ensures driving safety, ride comfort, and smooth steering, but the cost is the deterioration of the tracking effect.

5.3.2 Feedforward Control

From (5–16), the term $C\dot{\theta}_r$ will bring the steady-state error, thus rendering the instability of the control system apparently. The lateral tracking errors and their derivatives fail to be zero simultaneously irrespective of the value of K. However, the lateral position error y_e can be ensured to be zero at a steady state by reasonably choosing the feedforward input δ_a. The feedback control law becomes:

$$u = \delta_c = -KL_e + \delta_a \tag{5–26}$$

Substitute (5–26) into (5–16) and simplify the tracking errors, such that:

$$L_e = \begin{bmatrix} \frac{\delta_a}{k_1} - \frac{\dot{\theta}_r}{k_1 v_x}[l - k_3 l_2 - \frac{m v_x^2}{l}(\frac{l_2}{c_1} + \frac{l_1 k_3 - l_1}{c_2})] \\ 0 \\ -\frac{\dot{\theta}_r}{v_x}(l_2 + \frac{l_1 m v_x^2}{l c_2}) \\ 0 \end{bmatrix} \tag{5–27}$$

where

$$\dot{\theta}_r = k_r \dot{s} \approx k_r v_x \tag{5–28}$$

If the lateral steady-state error can be made zero, the feedforward steering angle is chosen as:

$$\delta_a = k_r[l - k_3 l_2 - \frac{m v_x^2}{l}(\frac{l_2}{c_1} + \frac{l_1 k_3 - l_1}{c_2})] \tag{5–29}$$

After employing the mass equivalent method and considering the geometric relationship of the used model, it can be deduced that ψ_e is approximately equal to $-\beta$. By (5–15), the real course error can be calculated as:

$$\psi_{er} = \beta + \psi - \theta_r = \theta_c - \theta_r = 0 \tag{5–30}$$

Consequently, the real tracking error $L_{er} = [y_e, \dot{y_e}, \psi_{er}, \dot{\psi_e}]^T$ eventually converges to 0 without steady-state error.

5.3.3 Longitudinal Control

As seen in Figure 5.4, the longitudinal control is implemented through a double PID controller. A brief statement of this approach is presented because the PID control has been mature and it is readily to be deployed into the tracking system. The calibration table of the throttle and brake is established in advance, where a set of velocity and acceleration signals corresponds to an exact value of the throttle opening or the brake pressure t_r. According to the model of the electric motor, the total drive torque at the wheel is:

$$T = \begin{cases} T_{max} t_r & w_{max} > w > 0 \\ T_{max} t_r w_c / w & w > w_{max} \end{cases} \tag{5–31}$$

where

$$w_c = P_{max} / T_{max} \tag{5–32}$$

The terms w, P, and T denote the speed, power, and torque of the motor. Similarly, the maximum values of the aforementioned variables are represented by w_{max}, P_{max}, and T_{max}. The motor will run the peak torque before reaching the maximum power. If the motor's speed reaches the maximum power point, it will be operated at constant power.

As indicated in Figure 5.4, the PID position controller regulates the vehicle's longitudinal motion to make $x_e \rightarrow 0$, whereas the PID velocity controller enables the velocity tracking that $\dot{s} \rightarrow v_r$. The position tracking is further enhanced in light of the fact that

the velocity controller takes the signals tuned from the position controller as the input. As a result, the double PID controller indirectly determines the torque and brake pressure to fulfill the longitudinal position and velocity tracking. Also, it is suggested that longitudinal tracking could be improved by tuning the PID gains.

5.4 Simulation Results

To further verify the effectiveness of the proposed tracking control scheme, the simulations under the different scenarios have been carried out in MATLAB/Simulink and CarSim. An E-Class vehicle (Sedan) was chosen as the mathematical model to mimic tracking behavior. The key parameters of the mathematical model and electric motor are listed as follows: $l_1 = 1.4$ m, $l_2 = 1.65$ m, $C_1 = C_2 = -118857$ N/rad, $m = 1830$ kg, $I = 3234$ kg·m^2, $P_{max} = 180$ kw and $T_{max} = 380$ N·m. The gains of the double PID controller were all selected as $k_P = 3$ and $k_I = k_D = 0.1$. Regarding the parameters of the planning module, it can be summarized as $x_r(T) = 100$ m and $y_r(x_T) = 30$ m for Scenario I, while $x_r(T) = 200$ m and $y_r(x_T) = 80$ m for Scenario II. In addition, Q was set as a 4×4 identity matrix, whereas R was chosen as 15 for Scenario I and 100 for Scenario II.

5.4.1 Scenario I

The tracking results of Scenario I is summarized in Figure 5.5. From an overall perspective, the vehicle automatically tracks an S-shape trajectory with satisfactory tracking performance. Provided in Figure 5.5(a) are the reference and actual trajectories. It can be seen that almost no deviation exists during the whole tracking duration, revealing the accurate and stable tracking ability. The desired and projected velocity $\dot{s}$ are shown in Figure 5.5(b), which reflects that the double PID method successfully controls the throttle

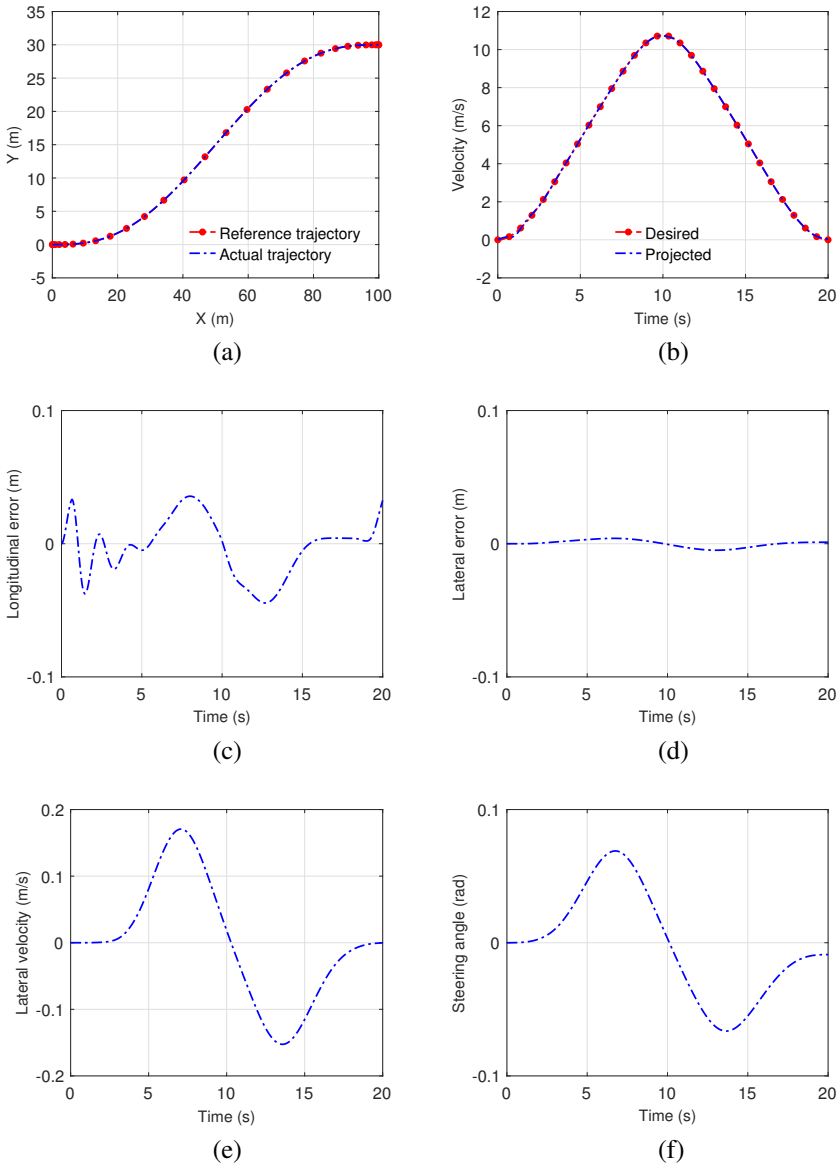


Figure 5.5: Tracking results of Scenario I.

or brake to accomplish velocity tracking. As seen in Figure 5.5(c) and 5.5(d), the longitudinal error is confined within $[-0.04, 0.04]$ despite slight fluctuations, while the lateral error rapidly converges to and stabilizes around 0. This phenomenon can be attributed to the LQR integrated with the feedforward control guaranteeing $L_{er} \rightarrow 0$ while offering adaptive gains against speed variation. Apart from this, the PID position controller has the potential to further strengthen the longitudinal tracking of the vehicle by adding the velocity controller. The lateral velocity in Figure 5.5(e) and the steering angle input in Figure 5.5(f) indicate that oscillations or improper driving behaviors barely appear when the vehicle operates automatically at a low speed. This phenomenon can be attributed to the superiority of the proposed tracking control scheme.

5.4.2 Scenario II

Since the proposed algorithm is designed for the wide-speed range control objective, another testing scenario with more stringent tracking requirements is applied to evaluate the control performance; the tracking results under high speed are displayed in Figure 5.6. Overall, the proposed steering control law is highly effective and stable so that the lateral error levels off at nearly zero without any oscillations during the whole tracking period. While the maximum forward speed is extended to 23.5 m/s at time $t = 10$ s, the AVs could still track the pre-defined trajectory at a high accuracy level, as evident in Figure 5.6(a). From Figure 5.6(b) and 5.6(c), the proposed double PID control shows strong robustness regardless of the change of current velocity since the longitudinal error only changes in the range of $[-0.1, 0.05]$. The proposed steering control law greatly benefits driving safety and ride comfort due to the smaller value and slow change of the lateral velocity, as evident in Figure 5.6(e). Even though raising the value of R from 15 to 100 may sacrifice the performance of the algorithm, the instability of lateral control can be eliminated significantly by the smaller steering angle, as evident in Figure 5.6(d) and 5.6(f).

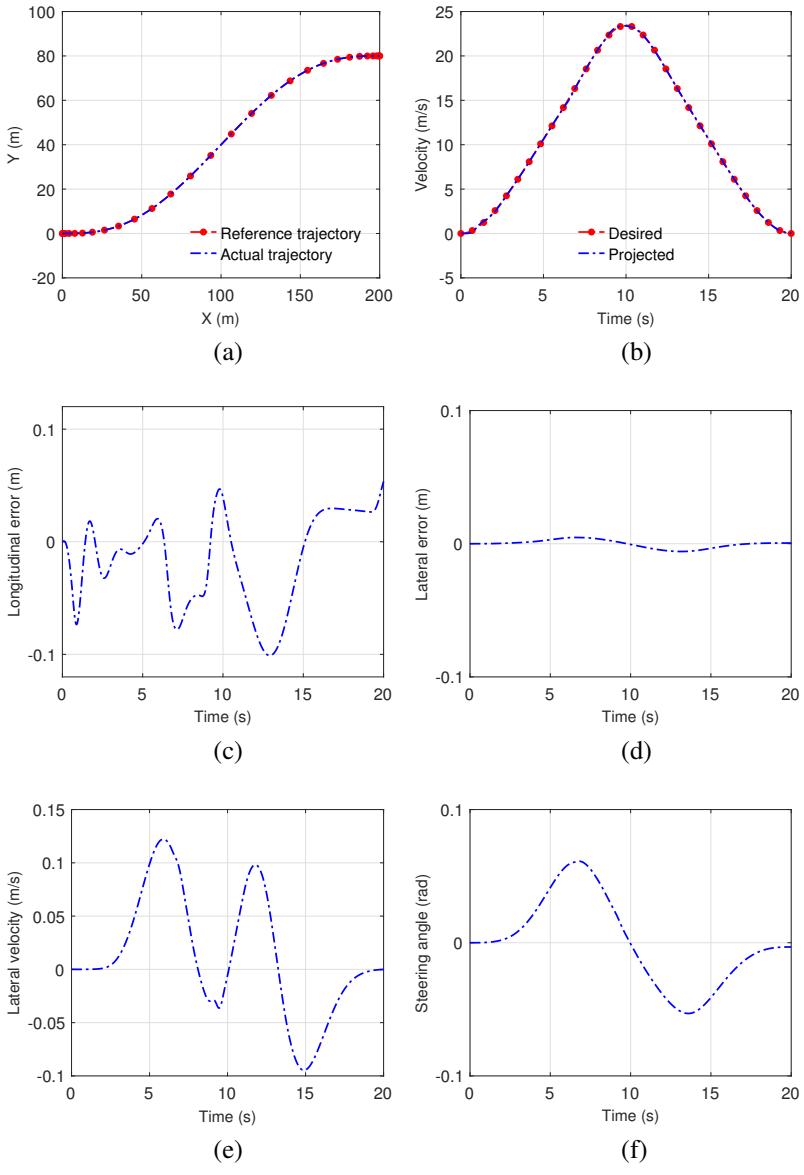


Figure 5.6: Tracking results of Scenario II.

Chapter 6

Contributions, Conclusions and Suggestions for Further Studies

6.1 Major Contributions and Highlights

The first contribution of this thesis research is the design, analysis, and verification of a novel backstepping control strategy for low-speed AVs to realize more speedy, accurate, and smooth trajectory tracking. More specifically:

(1)A control law, characterized by online self-tuning gains, was derived via the backstepping technique, which results in superior tracking performances, including quicker convergence rate, response, higher tracking precision, etc.

(2)The yaw error observer is designed as a parallel system to provide data, aiming at replacing the sensor when the vehicle's orientation is unmeasurable.

(3)Obtain the approximately optimal tuning parameters in fewer iterations by merging the PSO algorithm with the proposed backstepping control law, reducing tuning effort, time, and human intervention.

(4)A fitness function containing full vehicle variables is created to evaluate the tracking performance. The velocity and steering tracking can be swiftly achieved on the posture tracking foundation by simply adjusting the error weights.

Another contribution of this thesis research is that a novel trajectory tracking control scheme is proposed for wide-speed range AVs, which can be summarized as follows:

(1)Decouple the vehicle motion into longitudinal and lateral motions by using the Frenet frame and Frenet equations. This means that the vehicle planar motion could be decomposed into two one-dimensional motions, reducing control complexity.

(2)Develop a composed control method for trajectory tracking, that is, the LQR control is integrated with the double PID controller to realize lateral and longitudinal tracking control, respectively.

(3)Introduce feedforward control into the closed-loop tracking system in order that lateral steady-state errors can be effortlessly eliminated and tracking stability is increased.

(4)The optimal steering could be guaranteed during the whole tracking period due to the design of adaptive gains, which implies that the proposed steering controller can utilize the optimal control gains as long as the forward speed of the vehicle changes.

6.2 Major Conclusions and Recommendations for Future Studies

The major conclusions drawn from various aspects of this thesis research and recommendations for future studies are listed below:

•There is a trade-off between tracking performance and stability for the PP controller, which is challenging to resolve and may cause course-dependent problems. This is because the PP approach does not consider the reference path's curvature and ignores the vehicle's lateral dynamics. Further, over tuning the value of control parameters may result in wide discrepancies with expected tracking, which is more and more influential as the curvature and speed increase. While the PP algorithm has these shortcomings, a superior characteristic worth highlighting during experimental tests is that the PP method has strong robustness against discontinuity in the reference path. For future research directions, we should explore a systematic way to tune the control parameters or devise an adaptive control gain that is related to either the path curvature or forward speed to match reasonable look-ahead distance with the aim of optimizing the tracking performance and ensuring the vehicle to converge to the path over time.

•The Stanley method is a valuable asset for tracking control. Based on the simulation and laboratory tests, it can be summarized that the Stanley control is able to correct the vehicle's heading and finally drive the vehicle converging to the path, no matter what the initial condition is. This should be attributed to the global exponentially convergence properties of the Stanley controller. Still, the Stanley controller is only a geometric tracking controller, similar to the PP controller, without considering many aspects of real life self-driving conditions, such as tire force effect, noisy measurements, and actuator dynamics. Moreover, the Stanley tracker may fail to track discontinuous paths or paths with high curvature because it only focuses on the current tracking state rather than making use of look-ahead distance to predict path curvatures, and the reference dynamics are not taken into account in the derivation process. These may cause undesirable rider comfort and tracking characteristics during maneuvers and sometimes even affects driving safely. Regarding the tuning of

control gains, it can be summarized as follows: Increasing the value of gains leads to more extensive and oscillatory steering commands but higher tracking accuracy, whereas decreasing the gains results in smaller and smooth steering commands but poor tracking accuracy. Hence, how to make further enhancements (e.g., gain tuning, adding other tuning terms) are of vital importance in Stanley control.

•In Chapter 4, the car-like vehicle kinematics, error dynamics, and control laws were derived and applied to a cutting-edge experimental testing bench featured with a Quanser QCar. The tracking control problems that appear in Chapter 2 and Chapter 3 have been successfully solved. The proposed YEO as an alternative system estimates yaw error, replacing the real sensors to provide data. For comparison purposes, the other three control strategies also have been investigated. Although all controllers successfully regulate the QCar to accomplish tracking, the proposed nonlinear backstepping control law combined with optimized tuning parameters from PSO offers the best performance and results. Another noticeable improvement is that the proposed controller enhances steering capacity to eliminate cutting corners. Finally, the experiment results reveal that not only can the proposed control strategy significantly boost the error convergence rate but also enhances the tracking precision and robustness, and the adaptive gains have the potential to further strengthen steering capacity and compensate for each other. This research gives insight into future work regarding trajectory tracking of car-like vehicles. Far more efforts, however, are desirable in controlling the transient performance and obtaining optimized tuning parameters in real-time. Therefore, it is recommended that the prescribed performance control could be applied to car-like vehicles. In this manner, the controller is designed to make the tracking error of the closed-loop system converge to a prescribed allowable range and ensures that the convergence rate, overshoot, and undershoot of control signals or control inputs meet the prescribed conditions, that is, the transient and

steady-state performances are satisfied at the same time so as to improve the performance of the tracking. Moreover, the pick of control gains become less significant, given that the tracking control problem of the original constrained system is transformed into the uniformly ultimately bounded problem of the states of the mapped unconstrained system.

•In Chapter 5, a trajectory tracking control scheme based on Frenet frame and Frenet equations was presented for wide-speed range vehicles to autonomously track the reference trajectory generated from the planning module in real-time, leading to marginal tracking errors and higher system stability. Also, with the reduced dimensions of the vehicle's motion, different control strategies, namely the double PID control and LQR control with adaptive gains, were utilized for the longitudinal and lateral tracking. By doing so, the complexity of the tracking control problems is decreased. The feedforward control was introduced into the steering control law, thus rendering the lateral position error zero at a steady state. The simulation results revealed the robustness and superiority of the proposed method under different driving scenarios. This research offers insight into the future work concerning trajectory tracking of AVs. Regarding the selection of control parameters of PID and LQR in this proposed tracking method, it is suggested that these parameters can be optimized and obtained by the PSO mechanism proposed in Chapter 4. Also, the constraints on steering rate and accelerations of AVs should be considered in real-world driving.

Bibliography

[1]C.-L. Hwang and H.-H. Huang, "Experimental validation of a car-like automated guided vehicle with trajectory tracking, obstacle avoidance, and target approach," in *2017 43rd Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pp. 2858–2863, 2017.

[2]A. D. Luca, G. Oriolo, and C. Samson, "Feedback control of a nonholonomic car-like robot," *Robot Motion Planning and Control*, pp. 171–253, 1998.

[3]A. Akhtar, C. Nielsen, and S. L. Waslander, "Path following using dynamic transverse feedback linearization for car-like robots," *IEEE Transactions on Robotics*, vol. 31, no. 2, pp. 269–279, 2015.

[4]N. Naderi Samani, M. Danesh, and J. Ghaisari, "Parallel parking of a car-like mobile robot based on the p-domain path tracking controllers," *IET Control Theory & Applications*, vol. 10, no. 5, pp. 564–572, 2016.

[5]N. H. Amer, H. Zamzuri, K. Hudha, and Z. A. Kadir, "Modelling and control strategies in path tracking control for autonomous ground vehicles: a review of state of the art and challenges," *Journal of Intelligent & Robotic Systems*, vol. 86, no. 2, pp. 225–254, 2017.

[6]T. Zou, J. Angeles, and F. Hassani, "Dynamic modeling and trajectory tracking control of unmanned tracked vehicles," *Robotics and Autonomous Systems*, vol. 110, pp. 102–111, 2018.

[7]Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *IEEE International Conference on Robotics and Automation*, pp. 384–389, 1990.

[8]X. Yang, P. Wei, Y. Zhang, X. Liu, and L. Yang, "Disturbance observer based on biologically inspired integral sliding mode control for trajectory tracking of mobile robots," *IEEE Access*, vol. 7, pp. 48382–48391, 2019.

[9]M. A. M. Obaid, A. R. Husain, and A. A. M. Al-Kubati, "Robust backstepping tracking control of mobile robot based on nonlinear disturbance observer," *International Journal of Electrical and Computer Engineering*, vol. 6, no. 2, p. 901, 2016.

[10]J. Jakubiak, E. Lefeber, K. Tcho ń, and H. Nijmeijer, "Two observer-based tracking algorithms for a unicycle mobile robot," *International Journal of Applied Mathematics and Computer Science*, vol. 12, no. 4, pp. 513–522, 2002.

[11]S. Dixit, S. Fallah, U. Montanaro, M. Dianati, A. Stevens, F. Mccullough, and A. Mouzakitis, "Trajectory planning and tracking for autonomous overtaking: State-of-the-art and future prospects," *Annual Reviews in Control*, vol. 45, pp. 76–86, 2018.

[12]D. Calzolari, B. Sch ürmann, and M. Althoff, "Comparison of trajectory tracking controllers for autonomous vehicles," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–8, 2017.

[13]B. Paden, M. Čáp, 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.

[14]A. Eskandarian, C. Wu, and C. Sun, "Research advances and challenges of autonomous and connected ground vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 2, pp. 683–711, 2019.

[15]H. Ohta, N. Akai, E. Takeuchi, S. Kato, and M. Edahiro, "Pure pursuit revisited: field testing of autonomous vehicles in urban areas," in *2016 IEEE 4th International Conference on Cyber-Physical Systems, Networks, and Applications (CPSNA)*, pp. 7–12, 2016.

[16]M. Cibooglu, U. Karapinar, and M. T. S öylemez, "Hybrid controller approach for an autonomous ground vehicle path tracking problem," in *2017 25th Mediterranean Conference on Control and Automation (MED)*, pp. 583–588, 2017.

[17]Y. Shan, W. Yang, C. Chen, J. Zhou, L. Zheng, and B. Li, "Cf-pursuit: A pursuit method with a clothoid fitting and a fuzzy controller for autonomous vehicles," *International Journal of Advanced Robotic Systems*, vol. 12, no. 9, p. 134, 2015.

[18]M. Elbanhawi, M. Simic, and R. Jazar, "Receding horizon lateral vehicle control for pure pursuit path tracking," *Journal of Vibration and Control*, vol. 24, no. 3, pp. 619–642, 2018.

[19]Y. Shan, B. Zheng, L. Chen, L. Chen, and D. Chen, "A reinforcement learning-based adaptive path tracking approach for autonomous driving," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 10581–10595, 2020.

[20]K. Lee, S. Jeon, H. Kim, and D. Kum, "Optimal path tracking control of autonomous vehicle: Adaptive full-state linear quadratic gaussian (LQG) control," *IEEE Access*, vol. 7, pp. 109120–109133, 2019.

[21]Q. Yao, Y. Tian, Q. Wang, and S. Wang, "Control strategies on path tracking for autonomous vehicle: State of the art and future challenges," *IEEE Access*, vol. 8, pp. 161211–161222, 2020.

[22]E. Alcala, V. Puig, J. Quevedo, T. Escobet, and R. Comasolivas, "Autonomous vehicle control using a kinematic lyapunov-based technique with lqr-lmi tuning," *Control Engineering Practice*, vol. 73, pp. 1–12, 2018.

[23]X. Duan, Q. Wang, D. Tian, J. Zhou, J. Wang, Z. Sheng, D. Zhao, and Y. Sun, "Implementing trajectory tracking control algorithm for autonomous vehicles," in *2021 IEEE International Conference on Unmanned Systems (ICUS)*, pp. 947–953, 2021.

[24]W. Farag, "Complex trajectory tracking using pid control for autonomous driving," *International Journal of Intelligent Transportation Systems Research*, vol. 18, no. 2, pp. 356–366, 2020.

[25]X. Wang, G. Zhang, F. Neri, T. Jiang, J. Zhao, M. Gheorghe, F. Ipate, and R. Lefticaru, "Design and implementation of membrane controllers for trajectory tracking of nonholonomic wheeled mobile robots," *Integrated Computer-Aided Engineering*, vol. 23, no. 1, pp. 15–30, 2016.

[26]P. Dai and J. Katupitiya, "Force control for path following of a 4ws4wd vehicle by the integration of pso and smc," *Vehicle System Dynamics*, vol. 56, no. 11, pp. 1682–1716, 2018.

[27]T. Y. Abdalla and A. Abdulkarem, "Pso-based optimum design of pid controller for mobile robot trajectory tracking," *International Journal of Computer Applications*, vol. 47, no. 23, pp. 30–35, 2012.

[28]P. Dai, J. Taghia, S. Lam, and J. Katupitiya, "Integration of sliding mode based steering control and pso based drive force control for a 4ws4wd vehicle," *Autonomous Robots*, vol. 42, no. 3, pp. 553–568, 2018.

[29]G. A. R. Ibraheem, A. T. Azar, I. K. Ibraheem, and A. J. Humaidi, "A novel design of a neural network-based fractional pid controller for mobile robots using hybridized fruit fly and particle swarm optimization," *Complexity*, vol. 2020, 2020.

[30]M. Gheisarnejad and M.-H. Khooban, "Supervised control strategy in trajectory tracking for a wheeled mobile robot," *IET Collaborative Intelligent Manufacturing*, vol. 1, no. 1, pp. 3–9, 2019.

[31]N. H. Amer, K. Hudha, H. Zamzuri, V. R. Aparow, Z. A. Kadir, and A. F. Z. Abidin, "Hardware-in-the-loop simulation of trajectory following control for a light armoured vehicle optimised with particle swarm optimisation," *International Journal of Heavy Vehicle Systems*, vol. 26, no. 5, pp. 663–691, 2019.

[32]Y. Dai, D. Jia, L. Guo, S. Ye, D. Luo, W. Hu, and W. Peng, "Backstepping controller for trajectory tracking of wheeled mobile robot based on particle swarm optimization," in *2019 Chinese Control Conference (CCC)*, pp. 4259–4264, 2019.

[33]P. Wang, S. Gao, L. Li, S. Cheng, and L. Zhao, "Automatic steering control strategy for unmanned vehicles based on robust backstepping sliding mode control theory," *IEEE Access*, vol. 7, pp. 64984–64992, 2019.

[34]J. Guo, Y. Luo, and K. Li, "Adaptive non-linear trajectory tracking control for lane change of autonomous four-wheel independently drive electric vehicles," *IET Intelligent Transport Systems*, vol. 12, no. 7, pp. 712–720, 2018.

[35]X. Wu, P. Jin, T. Zou, Z. Qi, H. Xiao, and P. Lou, "Backstepping trajectory tracking based on fuzzy sliding mode control for differential mobile robots," *Journal of Intelligent & Robotic Systems*, vol. 96, no. 1, pp. 109–121, 2019.

[36]B. Ibari, L. Benchikh, A. R. H. Elhachimi, and Z. Ahmed-Foitih, "Backstepping approach for autonomous mobile robot trajectory tracking," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 2, no. 3, pp. 478–485, 2016.

[37]J. Hu, Y. Zhang, and S. Rakheja, "Adaptive trajectory tracking for car-like vehicles with input constraints," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 3, pp. 2801–2810, 2021.

[38]H. Fang, R. Fan, B. Thuilot, and P. Martinet, "Trajectory tracking control of farm vehicles in presence of sliding," *Robotics and Autonomous Systems*, vol. 54, no. 10, pp. 828–839, 2006.

[39]M. Abdelwahab, V. Parque, A. M. F. Elbab, A. Abouelsoud, and S. Sugano, "Trajectory tracking of wheeled mobile robots using z-number based fuzzy logic," *IEEE Access*, vol. 8, pp. 18426–18441, 2020.

[40]P. Panahandeh, K. Alipour, B. Tarvirdizadeh, and A. Hadi, "A self-tuning trajectory tracking controller for wheeled mobile robots," *Industrial Robot: the International Journal of Robotics Research and Application*, 2019.

[41]H. Guo, D. Cao, H. Chen, Z. Sun, and Y. Hu, "Model predictive path following control for autonomous cars considering a measurable disturbance: Implementation, testing, and verification," *Mechanical Systems and Signal Processing*, vol. 118, pp. 41–60, 2019.

[42]Z. Li, J. Deng, R. Lu, Y. Xu, J. Bai, and C.-Y. Su, "Trajectory-tracking control of mobile robot systems incorporating neural-dynamic optimized model predictive approach," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 6, pp. 740–749, 2015.

[43]H. Peng, W. Wang, Q. An, C. Xiang, and L. Li, "Path tracking and direct yaw moment coordinated control based on robust mpc with the finite time horizon for autonomous independent-drive vehicles," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 6, pp. 6053–6066, 2020.

[44]T. P. Nascimento, C. E. D órea, and L. M. G. Gonçalves, "Nonholonomic mobile robots' trajectory tracking model predictive control: A survey," *Robotica*, vol. 36, no. 5, pp. 676–696, 2018.

[45]T. Elmokadem, M. Zribi, and K. Youcef-Toumi, "Trajectory tracking sliding mode control of underactuated AUVs," *Nonlinear Dynamics*, vol. 84, no. 2, pp. 1079–1091, 2016.

[46]A. Bessas, A. Benalia, and F. Boudjema, "Integral sliding mode control for trajectory tracking of wheeled mobile robot in presence of uncertainties," *Journal of Control Science and Engineering*, vol. 2016, 2016.

[47]X. Wang and W. Sun, "Trajectory tracking of autonomous vehicle: A differential flatness approach with disturbance-observer-based control," *IEEE Transactions on Intelligent Vehicles*, 2022.

[48]J. Guo, Y. Luo, K. Li, and Y. Dai, "Coordinated path-following and direct yaw-moment control of autonomous electric vehicles with sideslip angle estimation," *Mechanical Systems and Signal Processing*, vol. 105, pp. 183–199, 2018.

[49]M. Cui, W. Liu, H. Liu, H. Jiang, and Z. Wang, "Extended state observer-based adaptive sliding mode control of differential-driving mobile robot with uncertainties," *Nonlinear Dynamics*, vol. 83, no. 1, pp. 667–683, 2016.

[50]D. Huang, J. Zhai, W. Ai, and S. Fei, "Disturbance observer-based robust control for trajectory tracking of wheeled mobile robots," *Neurocomputing*, vol. 198, pp. 74–79, 2016.

[51]Y. Zou, C. Deng, and M. Lu, "Distributed output feedback consensus tracking control of multiple nonholonomic mobile robots with directed communication graphs and sensor faults," *Nonlinear Dynamics*, vol. 105, no. 3, pp. 2327–2339, 2021.

[52]C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz, *et al.*, "Self-driving cars: A survey," *Expert Systems with Applications*, vol. 165, p. 113816, 2021.

[53]F. Gottmann, M. B öhm, and O. Sawodny, "Integrated trajectory and path tracking of under-actuated and over-actuated cars up to the handling limits," in *2017 American Control Conference (ACC)*, pp. 954–960.

[54]A. Houenou, P. Bonnifait, V. Cherfaoui, and W. Yao, "Vehicle trajectory prediction based on motion model and maneuver recognition," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4363–4369.

[55]S. Zhu and B. Aksun-Guvenc, "Trajectory planning of autonomous vehicles based on parameterized control optimization in dynamic on-road environments," *Journal of Intelligent & Robotic Systems*, vol. 100, no. 3, pp. 1055–1067, 2020.

[56]M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenet frame," in *2010 IEEE International Conference on Robotics and Automation*, pp. 987–993.

[57]Y. Dai and S.-G. Lee, "Perception, planning and control for self-driving system based on on-board sensors," *Advances in Mechanical Engineering*, 2020.

[58]Y. Yoshihara, Y. Morales, N. Akai, E. Takeuchi, and Y. Ninomiya, "Autonomous predictive driving for blind intersections," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3452–3459.

[59]H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, "Baidu Apollo EM motion planner," *arXiv:1807.08048*, 2018.

[60]M. Park and Y. Kang, "Experimental verification of a drift controller for autonomous vehicle tracking: a circular trajectory using LQR method," *International Journal of Control, Automation and Systems*, vol. 19, no. 1, pp. 404–416, 2021.

[61]D. Watzenig and M. Horn, *Automated Driving: Safer and More Efficient Future Driving*. Springer, 2016.

[62]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, 2019.

[63]R. Rajamani, *Vehicle Dynamics and Control*. Springer Science & Business Media, 2011.

[64]P. Hang and X. Chen, "Path tracking control of 4-wheel-steering autonomous ground vehicles based on linear parameter-varying system with experimental verification," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 235, no. 3, pp. 411–423, 2021.

[65]T. Weiskircher, Q. Wang, and B. Ayalew, "Predictive guidance and control framework for (semi-)autonomous vehicles in public traffic," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 6, pp. 2034–2046, 2017.