POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Automotive Engineering

Tesi di Laurea Magistrale

Motion Planning and Control under Cut-in Scenario Considering Dynamic Games Theory in Autonomous Vehicle

Supervisor:

Prof. Massimiliana Carello

Candidate:

Qiyu Zhang S275814

December 2021

Abstract

The cut-in scenario is one of the challenging traffic situations for autonomous vehicles since it requires the ego car to not only avoid the static obstacles on the road but also to react quickly to the sudden behavior changes of the surrounding traffic participants. Traditional approaches commonly adopt a predict-then-plan architecture which decouples the Prediction and Planning modules thus lacking of interactions between them. A planning algorithm considering dynamic games theory is proposed in this paper to get multiple decisions when encountered with a cut-in scenario. Profit functions considering the state deviations and the lane change cost of the side car are designed. Then, the output of the planning is fed into motion control module. A CarSim-Simulink model is constructed and classical LQR and PID controllers are applied to it to motion control module to more intuitively see the planning output and the control effect.

Keywords: cut-in, motion planning and control, dynamic game

Table of content

Chapter 1 Introduction and Research Review

1.1 Research Review

Cut-in is the most common scenarios in the daily travel where the nearby vehicle performs a lane-change maneuver from the origin lane to the target lane where the ego car drives, usually includes a lane change and merging of the traffic participating cars.

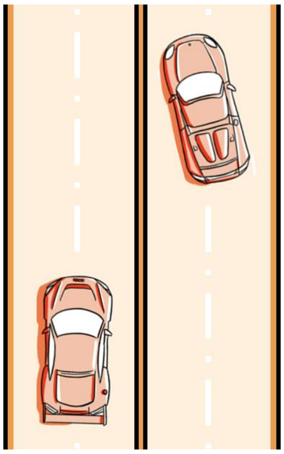


Figure 1.1 Cut-in lane change

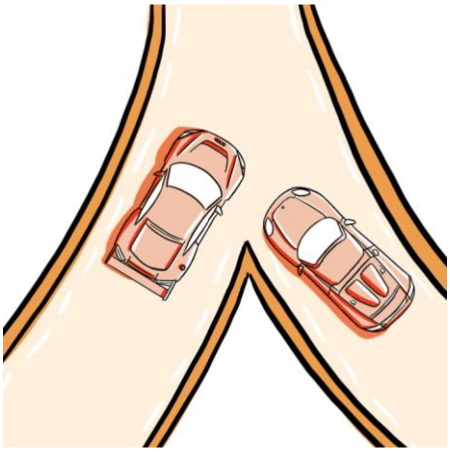


Figure 1.2 Merging on ramp

The algorithms of the autonomous driving vehicle mainly include Perception, Planning and Control three modules. The cut-in scenario is one of the challenging traffic situations for autonomous vehicles.[9] For Planning and Control modules especially for the interactive trajectory planning, it requires high performance to not only to avoid the static obstacles in the road but also to react quickly to the prediction output of the surrounding traffic participants.

Intensive research efforts have been devoted to the driver’s behavior analysis and trajectory tracking control. Methods of driver’s behavior prediction including hidden Markov Model[10], Gaussian Mixture Model[11], and Neural Network[12], are used to forecast drivers’ future behaviors.

Researches on trajectory tracking control methods to follow the reference paths are also multiple. The Back-stepping Algorithm is applied to the trajectory tracking control[13], the applications are covered in the field of mobile robots and autonomous vehicles. Considering the robustness, improved control methods are employed to trajectory tracking considering the model uncertainties and the external disturbances[14]. The MPC method that handles the input/output constraints also involved for vehicle trajectory tracking control[15]. Improved MPC is also proposed considering the working condition of the high-speed driving [16]. From the control perspective of view, the basic requirements for autonomous driving vehicles are to simultaneously track the reference trajectory and cooperate with the cut-in vehicle. We can see that MPC is wildly used in the field of academic researches for its repeated optimizations and prediction views at each time step, however, those advantages means that MPC requires powerful and fast processors with a large memory to solve online. This is why MPC is still not wildly spread in the field test. Therefore, we adopt traditional PID and LQR controller to complete the motion control. it is easy to debug and sufficient control accuracy is guaranteed.

From the preceding works' review, we can see that traditional approaches commonly adopt a predict-then-plan architecture. Firstly, predictions of other surrounding participants' trajectories are computed, then they are fed into a planner which considers them as immutable obstacles. This method usually decouples the prediction and planning modules for the ignorance of the interaction between these two modules, which may cause frequently sharp deceleration that will lead to decrement of the traffic efficiency and occupants' comfort. Actually, the behavior of ego car will affect the behavior of surrounding traffic participants, and vice versa. Obviously, preceding kind of algorithms also lack of attention to the impact of the ego car's behavior on the surrounding cars. Like Yimin Chen, Chuan Hu, and Junmin Wang[8] use a recurrent neural network (RNN) with long short-term memory (LSTM) cells to predict the driver behaviors of the cut-in vehicle, more reliable prediction outputs are proposed but still not considering the interaction with planning module. Decouple of the prediction and planning can also lead to the "frozen robot" problem that arises when the planner finds that all paths to the goal are unsafe[1]. Therefore, a richer interactive behavior is needed. We construct a cost function considering both the ego car and the side car, then, to find an optimal solution which can largely reduce the impact of behaviors on both others.

Dynamic game theory is introduced in this paper to deal with this situation. An open issue focused on influences of the ego car on safety and feasibility for autonomous driving technology is well-described. LUCID Games[1] developed by Cleac'h, Schwager and Manchester from Stanford University introduced an inverse optimal control algorithm that is able to estimate the other players' cost functions in real-time, and feed those estimates online into a receding-horizon game-theoretic planner. The algorithm supposes no definite communication or coordination between the ego player and the other players in the environment. An MPC also implemented to demonstrate real-time performance on complex autonomous driving scenarios. They also developed ALGAMES[3] (Augmented Lagrangian GAME-theoretic Solver), a solver that handles trajectory optimization problems with multiple actors and general nonlinear state and input constraints. To solve these trajectory optimization problems with constraints, Taylor A. Howell, Brian E. Jackson, and Zachary Manchester presents ALTRO[4] (Augmented Lagrangian Trajectory Optimizer), an algorithm that is able to handle general nonlinear state and input constraints. Methods include Differential Dynamic Programming (DDP)[5] and Iterative LQR (ILQR)[6] , as well as various shooting methods[7]. However, the application of these solutions requires particularly high real-time performance of the controlled system, especially in high-speed scenarios. Therefore, the cut-in game model based on optimal control cannot be applied to real vehicles at the primary engineering stage.

This paper introduces a new approach to autonomous driving planning algorithm based on game theory. The algorithm achieves real-time performance by decomposing potential dynamic hierarchical games into short-term strategic games with simplified dynamic models and complete information structure. Then, motion control module is completed and furthermore the simulation results. Traditional LQR and PID controllers are involved in the construction of the control Simulink model.

1.2 Cut-in Scenario and Dynamic Game

Cut in scenario generally affects the original driving states of the two cars which requires tolerance and cooperation between different vehicles. We define the vehicle that does not need to change lanes as the “Ego Car”, and the remaining one as the “Side Car”.

For human driver, they often need to prejudge the behavior of the side car and then decide to overtake or to yield. During this period, the two human drivers are involved into a dynamic game. And we can know the best situation is that both two of them can reach their target lane with minimum costs, which means the optimal solution of this dynamic game is that both cars’ behavior is not affected by the other.

There are two ideal cut-in scenarios, one is that the ego vehicle is far away from the side vehicle, and the collision time is larger than the threshold set; the other is the velocity of the side car is greater than the ego car. In these two cut-in situations, the collision between the ego car and the side car won’t happen and they both do not need to change the original state of motions, therefore, it is no need to consider about dynamic games under those two conditions.

In this paper, we are considering the following two cases:

1. The side car with low speed is in front of the ego car.

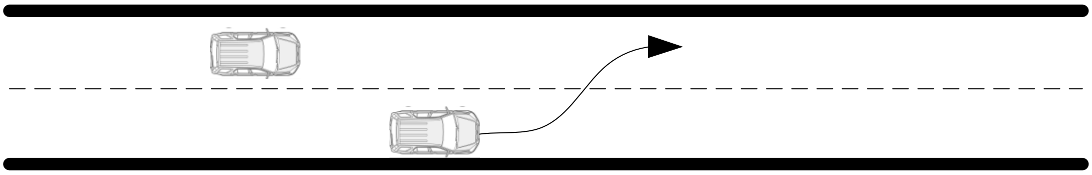


Fig 1.3 side car with low speed in front of the ego car

2. The side car with high speed is behind the ego car

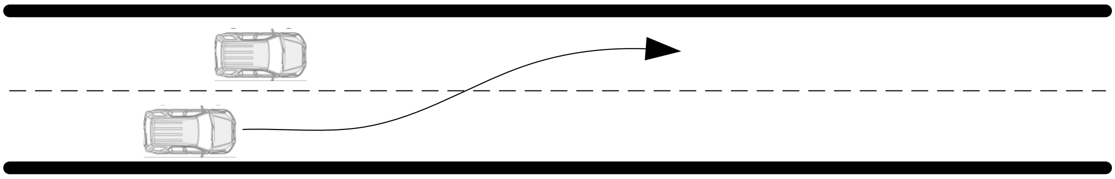


Fig 1.4 side car with high speed behind the ego car

Chapter 2 Dynamic Games Theory

2.1 Vehicle Dynamic Games Model under cut-in scenario

2.1.1 Description about the Game Theory

The application of game theory to human driving behavior has limitations. The main reason is that human drivers are not completely rational, or called bounded rationality. In this paper, cut-in scenario involves two cars, the ego car is autonomous driving vehicle and we assume another one is driven by a fully rational human driver thus game theory can be applied more reliable.

Generally, games can be divided into cooperative games and non-cooperative games. The difference between a cooperative game and a non-cooperative game is whether there is a binding agreement between their interactive behaviors. If there is, it is a cooperative game; if not, it is a non-cooperative game.

From the time sequence of behavior, game theory is further divided into static games and dynamic games: static games are in which players choose at the same time or not at the same time, but latter player does not know what the first player has taken; Dynamic game means that in the game, the actions of the participants have a sequence, and the latter player can observe the actions selected by the first. The interaction between the ego car and the side car in the cut-in scenario is regarded as a non-cooperative dynamic game problem.

$$G = < N, A_i, R_i > \qquad (2.1)$$

Where

N represents the number of the players, in this paper we have two players. One is the ego car, another is the side car;

A_i is the set formed by the strategy of player i;

R_i is the profit function of player i.

For the vehicle decision-making in the cut-in scenario, the ego vehicle has two strategies, they can be written as:

$$A_e = \{deceleration, acceleration\}$$

The strategy set of the side car:

$$A_s = \{cut - in, yield\}$$

all possible results of this two-vehicles game are shown in Table 1.

Ego car Side car	Deceleration	Acceleration
Cut in	*Ego car decelerate, side car cut in (1,1)*	Ego car accelerate, side car cut in (1,2)
Yield	Ego car decelerate, side car yield (2,1)	*Ego car accelerate, side car yield (2,2)*

Table 2.1 Strategy of each player in cut-in scenario

2.1.2 Nash equilibrium solution

If all game participants face a situation that when others do not change their strategy, his strategy at this time is optimal, the strategy combination here can be called Nash Equilibrium. The solution of the game problem is called the Nash equilibrium solution. The Nash equilibrium solution is not only the best strategy for ego player, it is also the best strategy for the other players under the current situation. That is, the players are unwilling to adjust their own strategy under the strategy given by the other. At the Nash equilibrium point, every rational participant will not have the urge to change strategy individually.

Through Table 2.1, it is not difficult to find that for the dynamic game in the cut-in scenario, if the ego car and the side car both want to complete their actions quickly (i.e. solution (1,2)), they are both encountered with risk of collision; if the ego car and the side car both choose to yield (i.e. solution (2,1)) to the each other since both of them prejudge the collision risk and choose humble strategy, a frozen scenario occurs which will decrease the efficiency of traffic flow;

If the ego car chooses to decelerate to yield and the side car cut in the ego lane (i.e. solution (1,1)), Similarly, the ego car chooses to accelerate to drive out of the cut-in scenario. These two solutions (1,1) and (2,2) are Nash equilibrium solutions.

The Nash equilibrium solution of non-cooperative dynamic game satisfies the condition:

$$R_i(a^*_{-i}, a^*_i) \geq R_i(a^*_{-i}, a_i) \qquad (2.2)$$

Where a_i is the ego car's optimal solution,

In the game, all participants try to maximize their Profit Functions R_e and R_s by choosing appropriate strategies, where R_e represents the profit function of the ego car and R_s is the profit function of the side car.

To construct the Profit Function, there are various kind of reference indicators, such as "safety indicators", which give the no-collisions highest priority; "Fast indicators", which aim to reach the target as quickly as possible; "Comfort indicators", which require the speed planning module to be as comfortable as possible, like no sudden accelerations and decelerations. The Profit Function is used as a constraint condition to couple the two game players. Simon Le et al. [2] chose the distance between vehicles as the penalty function, and the closer the ego vehicle is to side vehicle, the penalty will be imposed; Mac Schwager et al. [3] chose the offset between the current position and the reference trajectory as the profit function; David Fridovich-Keil et al. [17] use the distance between the actual position of the vehicle and the center of the road as a cost function.

Chapter 3 Motion Planning

3.1 Problem Statement

3.1.1 Define functions describing the vehicles' motion

When an autonomous vehicle encounters a cut-in scenario, the most ideal situation is that the ego car will not be affected by the sidecar's cut-in action, and can still drive on the original road in the state of the previous time step, and similarly the side car will not be affected by the ego car. Based on the above criteria, the number of changes during the movement of the ego car and the side car are taken as the objective function of the game path planning we want to optimize.

Suggested on a straight road, the motions of the ego car are generally deceleration and acceleration. In order to simplify the mathematical model of the problem, function $v_1(t)$ is used to describe the velocity state of the ego car. Behaviors of the side car are more complicated which involves the lateral and longitudinal position changes and speed changes. The speed function $v_2(t)$ is used to describe the speed change of the side car during the cut-in, and the cut-in angle θ is used to characterize the sidecar's posture.

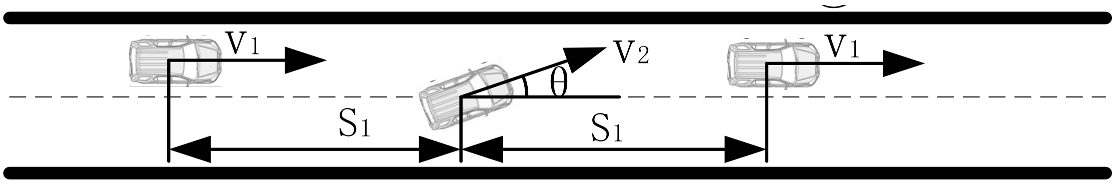


Figure 3.1

3.1.2 Cut-in behavior description

The cut-in behavior of the side car is a process of lane change, sigmoid function can be used to express the spatial path when the side car cuts in, as shown in Figure 2.2. The cut-in angle θ of the sidecar is the derivative of the sigmoid function at x=0.

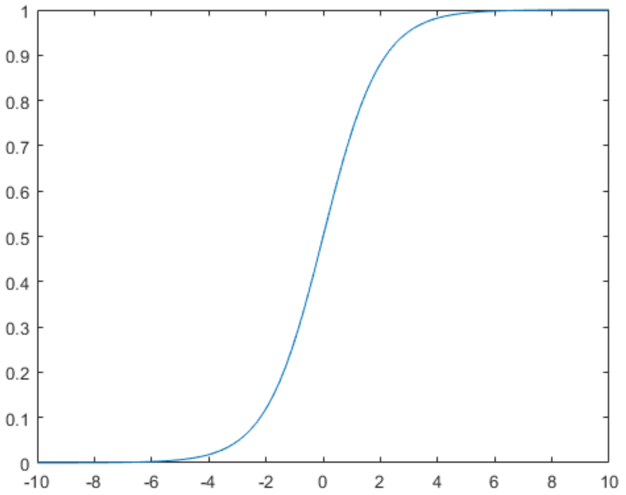


Figure 3.2 Sigmoid Function

Operating at the standard Sigmoid Function to better describe the cut in behavior of the side car and we get:

$$y = \frac{\omega}{1 + e^{-kx}} \qquad (2.3)$$

Where ω is the width of the lane, k is the Sigmoid function's derivative rate coefficient in the origin and $\theta = arctan\frac{k}{4}$ represents the cut-in angle of the side car.

3.1.3 Profit function design

(1) Considering the variance of the speed function of the two cars is the smallest

$$J_D = D_1 + D_2 = \frac{1}{t_e - t_0}\int_{t_0}^{t_e} [v_1(t) - v_1]^2 dt + [v_2(t) - v_2]^2 dt \qquad (2.4)$$

Where t_0 is the time of start of planning. t_e is when the side car complete lane change considering the sign of the yaw angle of the side car becomes to 0.

(2) Considering the difference between the actual cut-in angle of the sidecar and the ideal cut-in angle at the current speed is the smallest

$$J_\theta = |(\theta_e(v_2(t_m)) - \theta_r(k)| \qquad (2.5)$$

Where t_m is when the side car crosses the lane line. k is the coefficient in Sigmoid function we stated before. θ_e is the ideal cut-in angle. θ_r is the real cut-in angle. θ is a function with respect of time.

$$\theta_e = \begin{cases} \frac{\pi}{4}, & v_2 < 2.4m/s \\ \arccos\left(1 - \frac{1.8}{v_2^2}\right), & otherwise \end{cases} \tag{2.6}$$

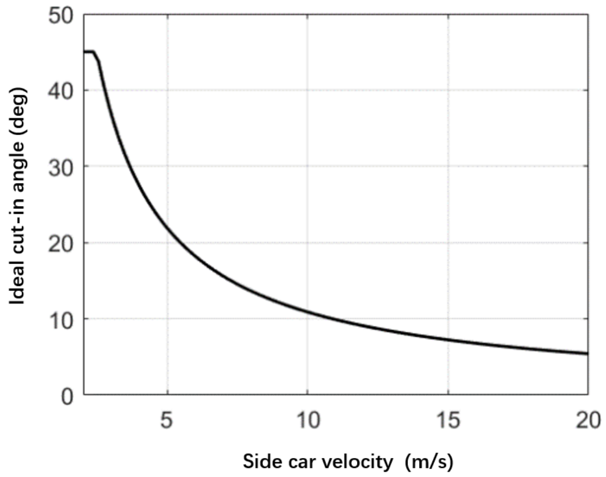


Figure 3.3 Relationship between the ideal cut-in angle and side car velocity

3.1.4 Constraints type

We choose safety constraints, give the non-collision the highest priority. There are two situations for the Nash equilibrium solution of the final game result of the two cars. As shown in Figure 3.4, for situation ①, the ego car chooses to yield the side car, and the side car enters ego lane; for situation ②, the ego car chooses to overtake and the side car is also able to complete lane change behind the ego car.

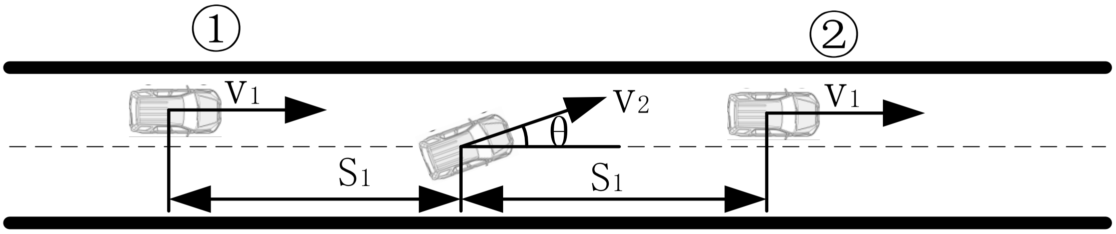


Figure 3.4

3.2 Planning algorithm

3.1.1 Constraints set up

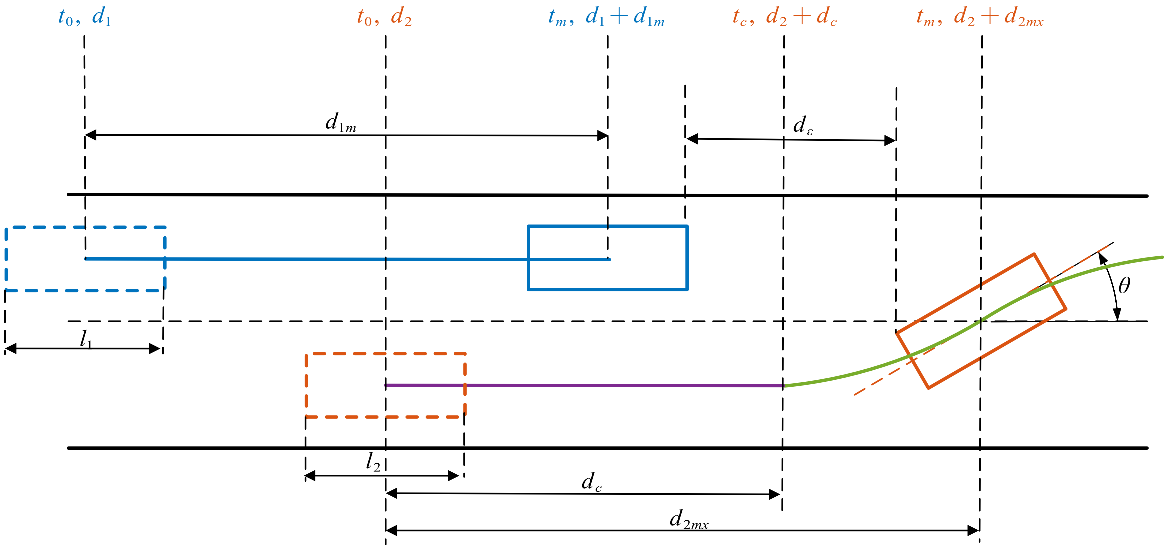


Figure 3.5

(1) The speed relationship between the ego car and the side car at the midpoint of the lane change t_m

$$\begin{cases} v_{1m} > v_{2m}, & ego\ car\ goes\ first \\ v_{1m} < v_{2m}, & side\ car\ goes\ first \end{cases} \tag{3.1}$$

Where $v_{1m} = v_1 + a_1(t_m - t_0)$, $v_{2m} = v_2 + a_2(t_c - t_0)$, t_c is the acceleration time of the side car and $t_c < t_m$

(2) The position relationship between the ego car and the side car at the midpoint of the lane change t_m

$$\begin{cases} \left(d_1 + d_{1m} - \frac{l_1}{2}\right) - \left(d_2 + d_{2m} - \frac{l_2}{2}cos\theta\right) > d_\varepsilon(v_{1m}), & when\ ego\ car\ goes\ first \\ \left(d_2 + d_{2m} - \frac{l_2}{2}cos\theta\right) - \left(d_1 + d_{1m} - \frac{l_1}{2}\right) > d_\varepsilon(v_{2m}), & when\ side\ car\ goes\ first \end{cases} \tag{3.2}$$

Where, d_ε is the safety distance under different velocity,

we assume:

$$d_{\varepsilon}(v_{1m}) = \frac{1}{2} v_{1m}$$

$$d_{1m} = v_1 t_0 + \frac{1}{2} a_1 (t_m - t_0)^2$$

$$d_{2m} = d_c + \frac{1}{2} S_m$$

S_{ε} is the half length of the Sigmoid function，computing it by curvilinear integral

$$S(x) = \frac{1}{1 + \mathrm{e}^{-kx}} \qquad (3.3)$$

$$S'(x) = \frac{k\,\mathrm{e}^{-kx}}{(1 + \mathrm{e}^{-kx})^2} \qquad (3.4)$$

We can calculate s_{12}, the length of $S(x)$ between $[x_1, x_1]$:

$$s_{12} = \int_{x_1}^{x_2} \sqrt{1 + \frac{k\,\mathrm{e}^{-2kx}}{(1 + \mathrm{e}^{-kx})^4}}\,\mathrm{d}x \qquad (3.5)$$

But s_{12} does not exist analytical solution, therefore, the exponential function is used to perform a piecewise fitting of $S'(x)$. The image of the $S'(x)$ is as follow.

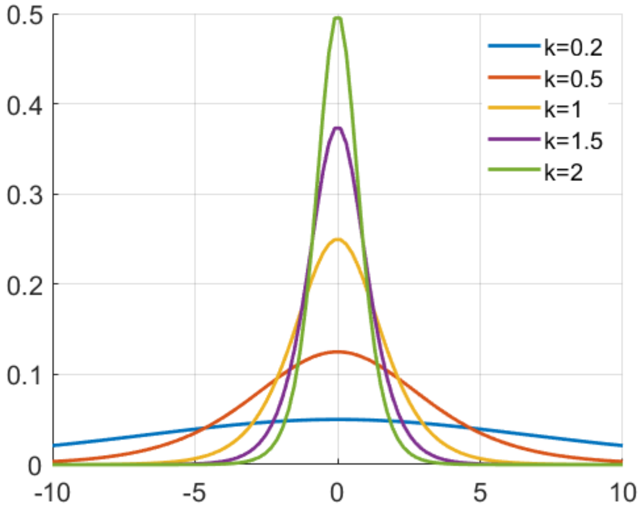


Figure 3.6 image of S'(x)

Divide the function into two sections with $x = 0$ as the limit, fitted with an exponential function $y = pe^{qx}$. In the left part and the right part, the parameter p is equal while the parameter q is opposite. p and q are both the function of k. Then the equation (3.5) can be rewritten as:

$$s_{12} = 2\int_{x_1}^{x_2} \sqrt{1+p^2\mathrm{e}^{2qx}}\,\mathrm{d}x \tag{3.6}$$

Let $p\,\mathrm{e}^{qx} = \tan u$, and making integration

$$x = \frac{1}{q}(\mathrm{lntan}\,u - \mathrm{ln}p) \tag{3.7}$$

$$\mathrm{d}x = \frac{\mathrm{d}u}{q\sin u\cos u} \tag{3.8}$$

Substituting equation (3.7) and equation (3.8) into equation (3.6), we have

$$s_{12} = 2\int_{x_1}^{x_2} \sqrt{1+p^2\mathrm{e}^{2qx}}\,\mathrm{d}x = 2\left(\frac{1}{b}\sec u + \mathrm{lnsin}\,\frac{u}{2} - \mathrm{lncos}\,\frac{u}{2}\right) \tag{3.9}$$

Applying the universal formula of trigonometric function on equation (3.9)

$$s_{12} = \frac{2}{b}\sqrt{1+\tan^2 u} + 2\ln\frac{\sqrt{1+\tan^2 u}-1}{\tan u} \tag{3.10}$$

Let $p\,\mathrm{e}^{qx} = \tan u$， $u = \arctan(p\,\mathrm{e}^{qx})$， Substitution into equation (3.10)

$$s_{12} = \frac{2}{b}\sqrt{1+p^2\mathrm{e}^{2qx}} + 2\ln\left(\sqrt{1+p^2\mathrm{e}^{2qx}}-1\right) - 2\mathrm{ln}p - 2qx \tag{3.11}$$

Therefore,

$$d_{2m} = d_c + \frac{1}{2}s_{12} \tag{3.12}$$

Combining (3.12) with (3.2)，we can get the second constraint which is the relationship between fitting parameters p， q and the parameter k in Sigmoid function

$$\begin{cases} p = 0.3015k \\ q = 0.566k \end{cases}, \quad k \in (0, 5] \tag{3.13}$$

(3) Dynamic constraints

The lateral acceleration must be limited to exclude the extreme working conditions

$$a_y = \sqrt{\frac{v^2}{R}} = g \cdot \min\,(\mu_{yp}, \frac{t}{2h_G})$$

Where h_G is the height of the center of the gravity of the vehicle.

We determine

$$a_y < 0.4g$$

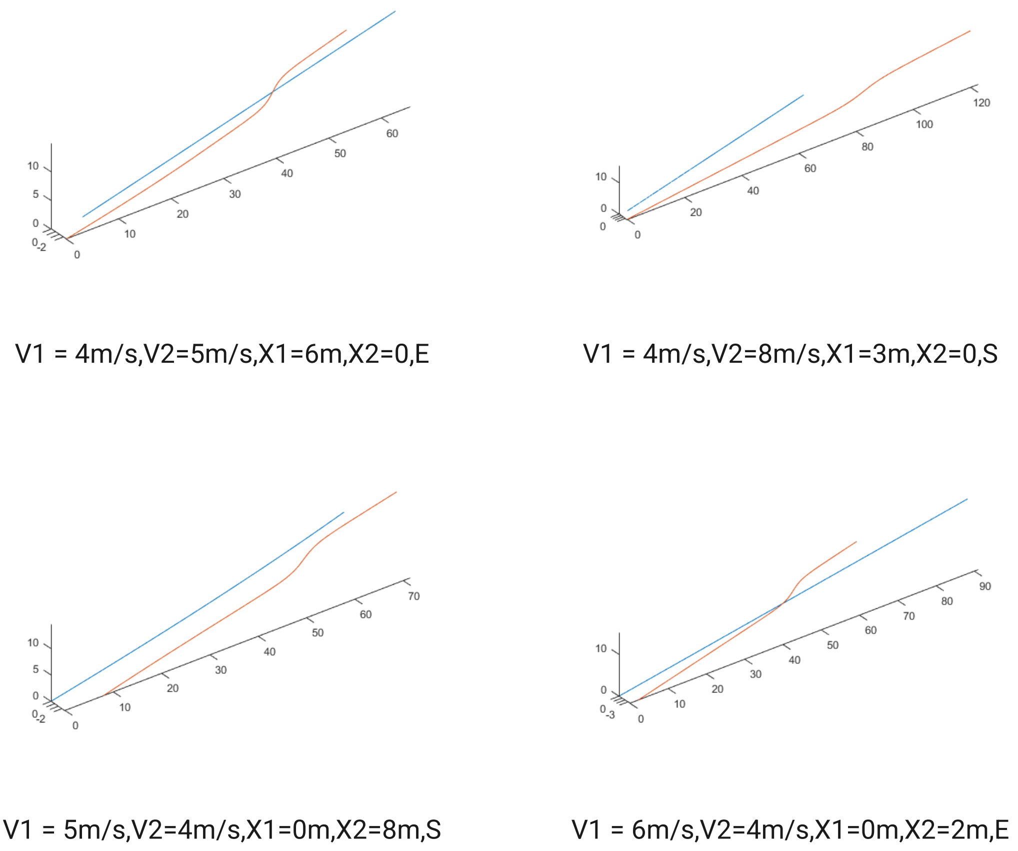
3.1.2 Algorithm backbone

```
Algorithm 1: Planning considering Games
Initial Data: $X_1(x_1, y_1, v_1), X_2(x_2, y_2, v_2), T_0$
Result: $A_1(a_1), A_2(a_2, \theta)$
For $k \leftarrow T_0$ to $T_e$
    For $X_1(x_1, y_1, v_1), X_2(x_2, y_2, v_2) \epsilon\ S$
        For $A_1$， $A_2\ \epsilon$ A
            $X_1^k \leftarrow (X_1^{k-1}, A_1^k)$
            $X_2^k \leftarrow (X_2^{k-1}, A_2^k)$
            $Q_D \leftarrow \sum_{T_0} J_D(X_1^k, X_1^{k-1}, X_2^k, X_2^{k-1})$
            $Q_\theta \leftarrow \sum_{T_0} J_\theta(X_2^k, X_2^{k-1},\ \theta_r)$
        $a_1, a_2, \theta \leftarrow \min Q_A = Q_D + Q_\theta$
    $X_1^k = \left(x_1 + \frac{1}{2}a_1 t^2, y_1, v_1 + a_1 t\right)$
    $X_2^k = \left(x_2 + \frac{1}{2}a_2 {t_c}^2 + s_{12}(\theta, t_s)\cos\theta, s_{12}(\theta, t_s)\sin\theta, v_2 + a_2 t_c\right)$
End
```

Complete codes in Matlab can be found in Appendix Ⅰ.

3.1.3 Planning output generated by Algorithm 1

Changing the initial conditions of the two vehicles, the initial speed of the ego car, the initial speed of the side car, and the distance between the two, through the Algorithm 1, four different interactive situations can be obtained, as shown in Figure 3.7



V1 = 4m/s,V2=5m/s,X1=6m,X2=0,E

V1 = 4m/s,V2=8m/s,X1=3m,X2=0,S

V1 = 5m/s,V2=4m/s,X1=0m,X2=8m,S

V1 = 6m/s,V2=4m/s,X1=0m,X2=2m,E

Figure 3.7 images of 4 Case

Chapter 4 Motion Control

4.1Three Coordinate System

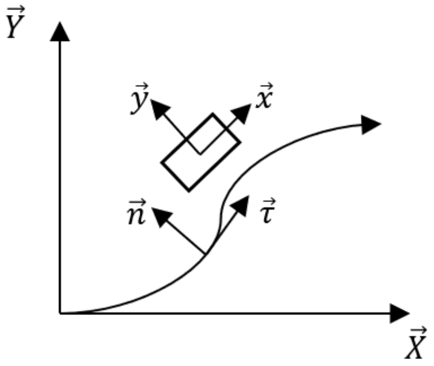


Figure 4.1 three coordinates

- (X,Y) World Coordinate System
- (x,y) Vehicle Coordinate System
- (τ,n) Frenet Coordinate System

4.1.1 World Coordinate System

The World coordinate system is referred to the right-handed Cartesian world coordinate system defined in ISO 8855, where all vehicles and their related coordinate systems are placed. A world coordinate system is important in global path planning, localization, mapping, and driving scenario simulation.

4.1.2 Vehicle Coordinate System

The vehicle coordinate system is anchored to the ego vehicle

The X_V axis points forward from the vehicle.

The Y_V axis points to the left, as viewed when facing forward.

The Z_V axis points up from the ground to maintain the right-handed coordinate system.

The vehicle coordinate system follows the ISO 8855 convention for rotation. Each axis is positive in the clockwise direction, when looking in the positive direction of that axis.

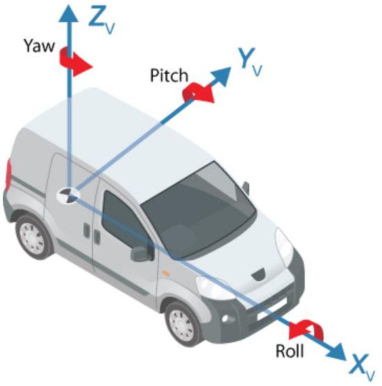


Figure 4.2 vehicle coordinate system

4.1.3 Frenet Coordinate System

The Frenet coordinate system is a way to express the road position in a more intuitive way than traditional x, y Cartesian coordinates. Using Frenet coordinates, we use variables *s* and *d* to describe the position of the vehicle on the road. The s coordinate represents the distance along the road (also called longitudinal displacement) and the d coordinate represents the left and right position on the road (also called lateral displacement).

Using the Frenet coordinate system in control design can decouple the longitudinal and lateral control.

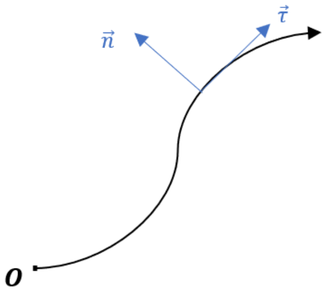


Figure 4.3

$$\vec{v} = \frac{ds}{dt} \qquad a_\tau = \frac{d^2s}{dt^2} \qquad a_n = \frac{v^2}{R} = \frac{v^2 tan\delta}{L}$$

Where s is directly related to a_τ. a_n is related to v and δ. When the longitudinal control is in steady state and the fluctuation of v is not much and the lateral control can be regarded to be related to δ only.

4.2 Vehicle dynamic model

4.2.1 Tire model and its linearization

Since the tire structure is complex and the dynamic performance is nonlinear, choosing a tire model that is practical and easy to use is the key to establishing a vehicle dynamics model. At present, the main tire models can be divided into theoretical tire models, empirical tire models and physical tire models.

The most common is the semi-empirical tire model based on magic formula proposed by Pacejka. When the tire slip angle is small, the tire lateral force can be approximately expressed as a linear function of the tire slip angle. The tire linearization model has a high fitting accuracy for conventional tires under the scenarios of lateral acceleration and tire slip angle.

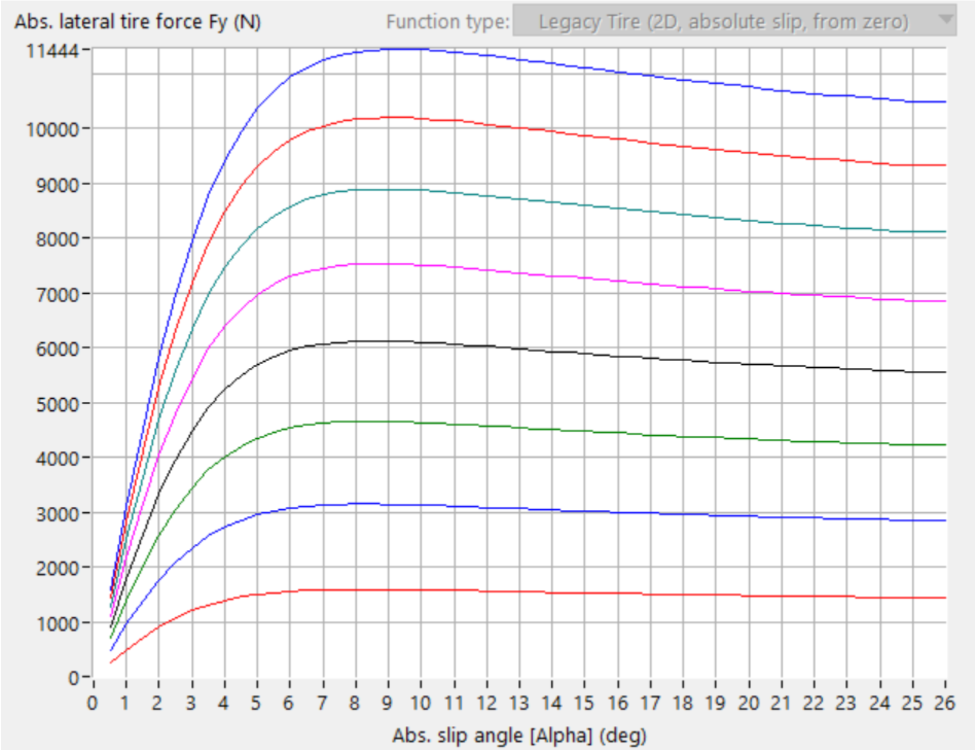


Figure 4.4

$$F_y = C_\alpha \alpha \qquad (4.1)$$

Therefore, when applying the tire linearization model, the lateral acceleration and tire sideslip angle must be limited.

4.2.2 Vehicle dynamic model

The more the number of degrees of freedom in Vehicle dynamic model used to complete motion control cannot always leads to better control result. When a complex model being adopted, we have to take into account all effects when vehicle is moving and the contribution is too much dedicated. Also, the risk of neglecting some terms is increasing. In primary design phase, in order to give qualitative interpretation of what are the behavior of vehicle, the simplified 2 degrees of freedom model can give an explanation of what happen with the complete model. It is able to capture enough of the nonholonomic constraints of the actual vehicle dynamics

There are assumptions for Bicycle Model:

- The vehicle is symmetric with respect to xz plane;
- The track of the vehicle can be ignored, which means $\frac{t}{2} \ll R_1$;
- No aerodynamic forces and no self-aligning moment of the tire.

The motion of the vehicle can be described only in the XY plane. And the vehicle is an hyperstatic structure, the vertical tire-ground forces are undefined if the structure compliance is not accounted for. This is bypassed if the so called ‘bicycle model’ is adopted. The layout of the vehicle and reference system where

- β is the side slip angle of the vehicle body (assumed to be small)
- ψ is the heading angle
- δ is the steering angle of the front wheel. The steering angle of the rear wheel is 0
- α_f, α_r are the side slip angle of the front and rear wheel respectively.

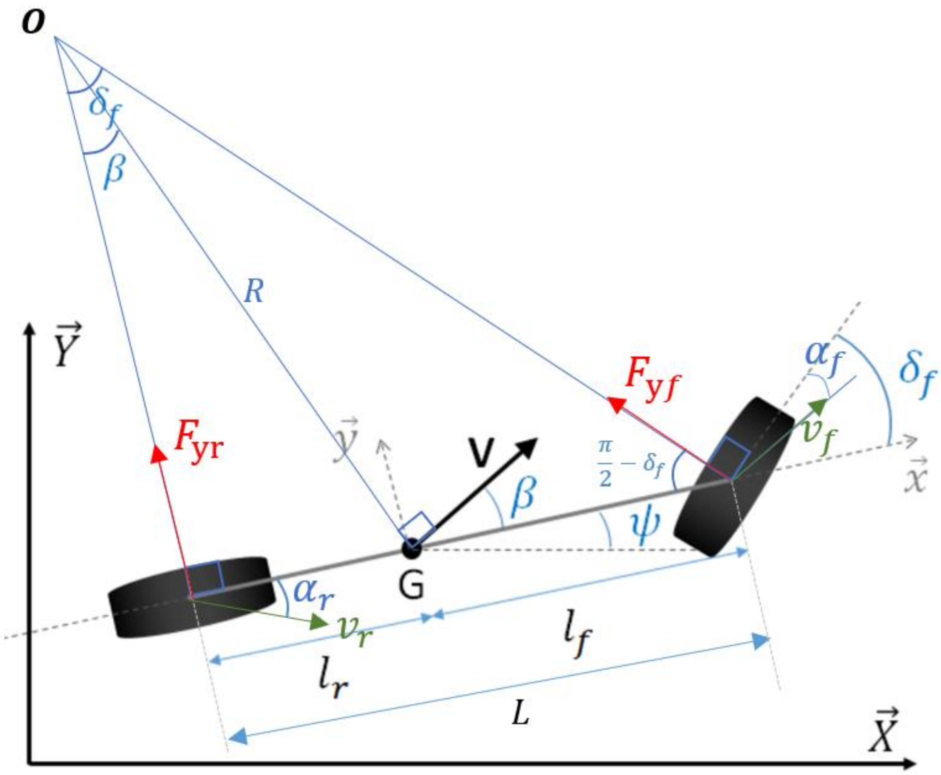


Figure 4.5 Layout of the vehicle and reference system

The reference frame is represented in fig. 4.5, the motion of the vehicle is described by the X, Y coordinates of the center of gravity G and by the yaw angle ψ. With reference to the world coordinate system, the equations of motion can be written as

$$\begin{cases} m\ddot{Y} = F_Y \\ J\ddot{\psi} = M_Z \end{cases} \qquad (4.2)$$

It is more convenient to write the equations of motion with reference to the frame Gxyz.

$$\begin{cases} m\ddot{y} = \sum F_y \\ J\ddot{\psi} = \sum M_z \end{cases} \qquad (4.3)$$

$$\begin{cases} ma_y = F_{yf}cos\delta + F_{yr} \\ J\ddot{\psi} = (F_{yf}cos\delta)\cdot l_f - F_{yr}\cdot l_r \end{cases}$$

Hypothesis must be presented here that δ is small, which means $cos\delta = 1$. Combined with equ.(4.1), we can simplify the upper equations.

$$\begin{cases} ma_y = C_{\alpha f}\alpha_f + C_{\alpha r}\alpha_r \\ J\ddot{\psi} = C_{\alpha f}\alpha_f \cdot l_f - C_{\alpha r}\alpha_r \cdot l_r \end{cases} \qquad (4.4)$$

4.3 Controller design

4.3.1 Control system description

Our aim is to control the lateral distance y and the yaw angle ψ through the steering angle δ. Then what we need to find out is the relationship between δ, y and ψ. The $v_x\dot{\psi}$ term represents the effect of inertial force.

$$v_y = \dot{y}$$

$$a_y = \ddot{y} + v_x\dot{\psi} \qquad (4.5)$$

Referring to fig.4.5, move v_r to the center of the gravity, assume α_r and α_f to be small and geometric relationship can be seen

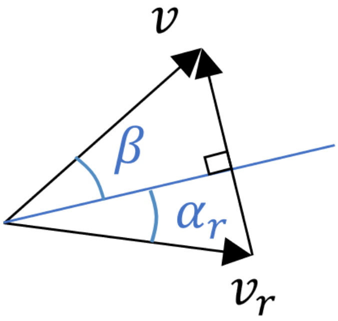


Figure 4.6 move v_r to the center of the gravity

$$tan\alpha_r = \frac{\dot{\psi}l_r - v_y}{v_x} \approx \alpha_r$$

$$\Rightarrow \alpha_r = \frac{v_y - \dot{\psi}l_r}{v_x} \qquad (4.6)$$

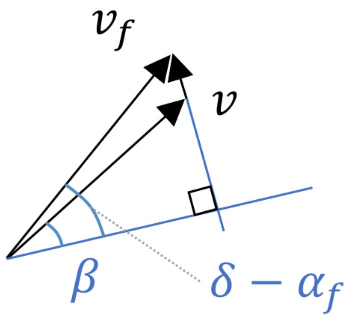


Figure 4.7 move v_f to the center of the gravity

$$tan(\delta-\alpha_f)=\frac{\dot{\psi}l_f+v_y}{v_x}\approx\delta-\alpha_f$$

$$\Rightarrow\alpha_f=\frac{v_y+\dot{\psi}l_f}{v_x}-\delta \qquad (4.7)$$

Considering the equation $(4.4)-(4.7)$, we can calculate

$$\begin{cases} ma_y=m(v_y+v_x\dot{\psi})=C_{\alpha f}\left(\frac{v_y+\dot{\psi}l_f}{v_x}-\delta\right)+C_{\alpha r}\left(\frac{v_y-\dot{\psi}l_r}{v_x}\right) \\ J\ddot{\psi}=C_{\alpha f}\alpha_f\cdot l_f-C_{\alpha r}\alpha_r\cdot l_r=l_fC_{\alpha f}\left(\frac{v_y+\dot{\psi}l_f}{v_x}-\delta\right)-l_rC_{\alpha r}(\frac{v_y-\dot{\psi}l_r}{v_x}) \end{cases} \qquad (4.8)$$

Equation (4.8) can be rewritten in matrix.

$$\begin{pmatrix}\ddot{y}\\ \ddot{\psi}\end{pmatrix}=\begin{pmatrix}\frac{C_{\alpha f}+C_{\alpha r}}{mv_x} & \frac{l_fC_{\alpha f}-l_rC_{\alpha r}}{mv_x}-v_x \\ \frac{l_fC_{\alpha f}-l_rC_{\alpha r}}{Jv_x} & \frac{{l_f}^2C_{\alpha f}+{l_r}^2C_{\alpha r}}{Jv_x}\end{pmatrix}\begin{pmatrix}\dot{y}\\ \dot{\psi}\end{pmatrix}+\begin{pmatrix}-\frac{C_{\alpha f}}{m}\\ -\frac{l_fC_{\alpha f}}{J}\end{pmatrix}\delta \qquad (4.9)$$

Choose state $x=\begin{pmatrix}\dot{y}\\ \dot{\psi}\end{pmatrix}$, the input $u=\delta$

$$\dot{x}=Ax+Bu$$

Assuming the reference trajectory is $\vec{x}_r$,

$$\vec{x}-\vec{x}_r=\vec{e}_{rr}$$

$$\dot{e}_{rr}=\bar{A}e_{rr}+\bar{B}u \qquad (4.10)$$

Our aim is to find an optimal u to minimize the e_{rr}.

4.3.2 Controller Design

Considering the cut-in scenario, the control of the ego car focus on longitudinal control and the side car focus on the longitudinal and lateral control. Considering the pros and cons of the commonly used controller in motion control of autonomous driving, we can know,

Proportional Integral Derivative (PID)control is probably the most popular control approach in industrial applications. The main reason for such a popularity is its simplicity and it can be suitably tuned. The existence of the Proportional term accounts for the present. It can make the system converge to the steady state rapidly by providing a command finalized at reducing the current tracking error. While the Proportional terms cannot handle the case with steady state error. And the Integral action is introduced, which allows precise tracking error by getting information from the past via integration. Integration terms may lead oscillation, this is why derivative action is involved. The derivative action can get a futural perspective by derivative the signal to know the trend of variation. It improves the dynamic performance and robustness.

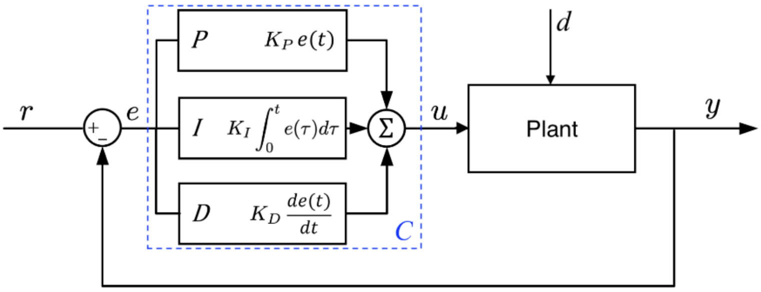


Figure 4.8 interpretation about PID controller

Liner Quadratic Regulator(LQR) is an optimal control based on state space. It is similar to pole placement method which can put the closed-loop poles on the desired positions. LQR find an optimal K by choosing characteristics. LQR can also set weights for each variable which can lead us to a balance between the performance and energy consumed.

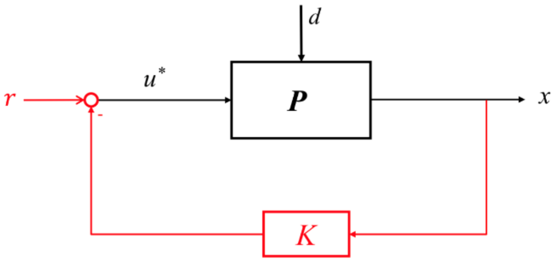


Figure 4.9 LQR controller structure

To design the LQR, a linear model must be developed. Assuming a liner system $\dot{x}(t) = Ax(t) + Bu(t)$, For any initial condition, the optimal LQR controller is given by

$$u^*(t) = -Kx(t) \qquad (4.11)$$

Where

$$K = R^{-1}B^T P$$

And P is obtained solving the following Algebratic Riccati Equation:

$$A^T P + PA + Q - PBR^{-1}B^T P = 0 \qquad (4.12)$$

The optimal controller K can also be easily design in Matlab by the command ‘K=lqr(A,B,Q,R)’, where the Q and R are the weight matrices.

Another controller mentioned frequently in research field is the Model Predictive Control (MPC). MPC controller need repeated prediction and optimization to find the real-time solution. This characteristic means that MPC need powerful, fast processor with a large memory to solve online optimization problem at each time step. And this is a large limitation for MPC to apply to a real vehicle.

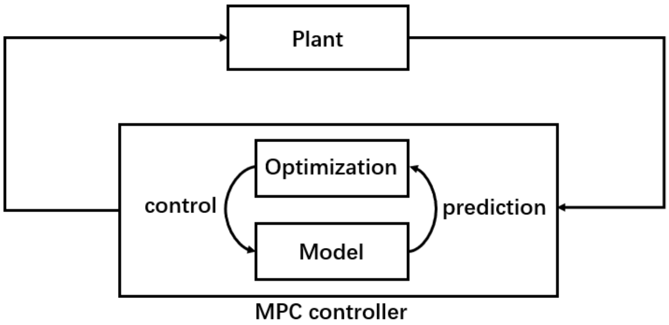


Figure 4.10 MPC controller structure

4.3.2.1 The ego car control

The control of the ego car mainly focuses on longitudinal control. We choose a double PID controller, one for position and another is for velocity. We are controlling the vehicle by controlling the acceleration pedal to change the power of the motor and therefore the torque to change the velocity and acceleration of the vehicle. Meanwhile, by controlling the brake pedal to change the brake pressure to decelerate the vehicle.

(1) Vehicle and electric motor parameters

Regarding C class Vehicle in CarSim library, the parameters are as follows:

parameters	units	values
l_f	1.015	m
L	2.91	m
C_f	-148970	N/deg
C_r	-82204	N/deg
m	1412	kg
J	1536.7	$kg \cdot m^2$

Table 4.1 vehicle parameters

The electric vehicle motor characteristic curves are generally like below:

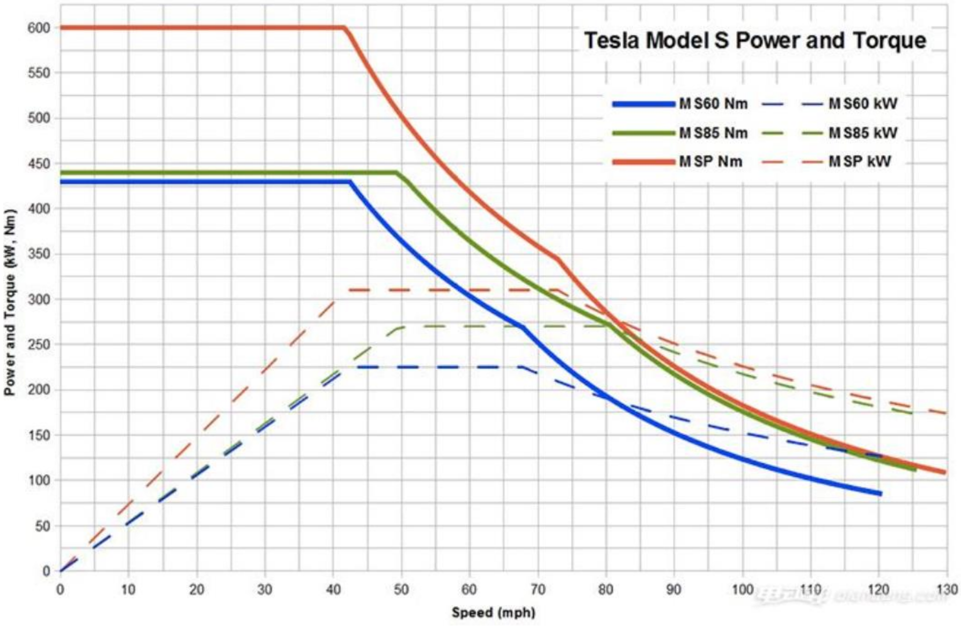


Figure 4.11

Our motor parameters are chosen:

Parameter	Value
Maximum power P	180kW
Maximum torque M	380Nm

Table 4.2 Motor parameters

$$\omega = \frac{P}{M} = \frac{180000}{380} = 4523\,rpm$$

And the relationship between the torque and motor speed is:

$$M = \begin{cases} 380 \cdot thr & 0 < \omega < 4523rpm \\ \dfrac{380 \cdot 4523 \cdot thr}{\omega} & \omega > 4523rpm \end{cases} \qquad (4.13)$$

Where thr represents the position of throttle.

There is no electric motor in CarSim, we can cancel the effect of gearbox in Carsim and create a motor model in Matlab. Codes can be seen in Appendix.

(2) Calibrations about throttle and brake

The aim of this calibration is to find a 3D curve relationship between the velocity and acceleration regarding different positions of throttle.

Using the following Simulink model and Matlab codes in Appendix to get the calibration data.

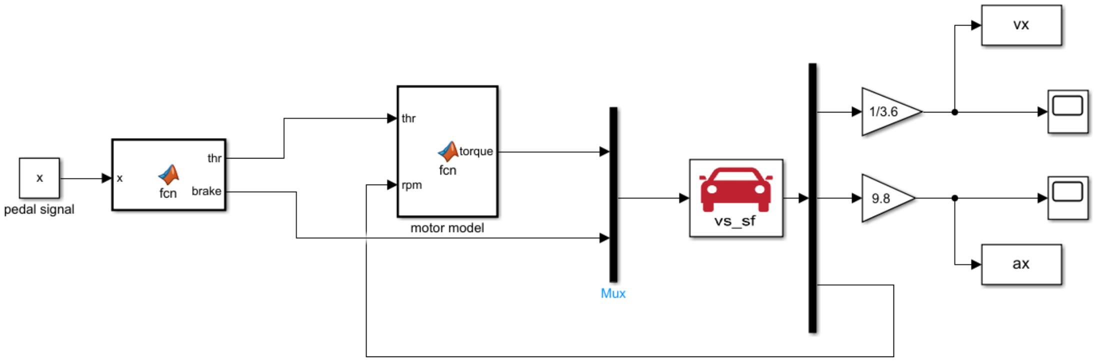


Figure 4.12 Simulink model used to calibrate

After getting the calibration table, we can add the lookup table dynamic block in Simulink model as an input signal of the control model.

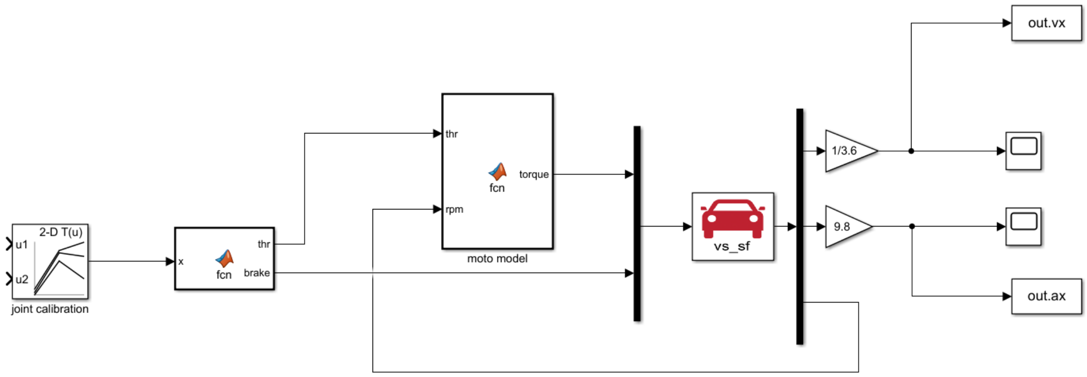


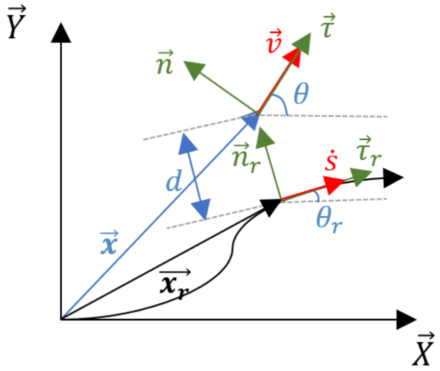
Figure 4.13 longitudinal control using joint calibration table

4.3.2.2 The side car controller

The side car needs to complete a lane change in this cut-in scenario. Therefore, we need to consider not only the longitudinal control but also the lateral control. We still choose PID controller to complete the longitudinal control. LQR controller are applied to the lateral control.

According to equation (4.10), using LQR to solve the problem. The cost function is defined:

$$J = e_{rr}^T Q e_{rr} + u^T R u \qquad (4.14)$$



$\vec{x_r}$: *the reference position vector*

$\vec{x}$: $\vec{x_r}$ *projected onto the reference trajectory*

θ_r: *the reference yaw angle*

θ: *the real yaw angle*

$\vec{v}$: *the velocity vector*

$\dot{s}$: $\vec{v}$ *projected onto the reference trajectory*

Figure 4.14 Reference and real trajectories

(1) Calculation about $\dot{d}$

Considering in Frenet coordinates, we can define the errors:

- d is the lateral error

- $\theta - \theta_r$ is the heading error. θ is the yaw angle which is equal to $\beta + \psi$.

According to the geometric relationship, we can calculate the lateral error

$$\vec{x}_r + d\vec{n}_r = \vec{x} \qquad (4.15)$$

$$\Rightarrow d = (\vec{x} - \vec{x}_r)\vec{n}_r \qquad (4.16)$$

Making the derivative

$$\dot{d} = \left(\dot{\vec{x}} - \overrightarrow{\dot{x}_r}\right) \cdot \vec{n}_r + (\vec{x} - \vec{x}_r) \qquad (4.17)$$

The vector of position can be written as

$$\dot{\vec{x}} = |\vec{v}|\vec{\tau} \qquad (4.18)$$

$$\overrightarrow{\dot{x}_r} = \dot{s}\overrightarrow{\tau_r} \qquad (4.19)$$

Combining with the equation (4.17)

$$\dot{d} = (|\vec{v}|\dot{\tau} - \dot{s}\vec{\tau}_r) \cdot \vec{n}_r + (\vec{x} - \vec{x}_r) \cdot \frac{d\vec{n}_r}{dt}$$

where

$$\frac{d\vec{n}_r}{dt} = \frac{d\vec{n}_r}{ds} \cdot \frac{ds}{dt} = \dot{s}(-k\vec{\tau})$$

Considering in Frenet frame

$$\frac{d\vec{\tau}}{ds} = k\vec{n} \qquad \frac{d\vec{n}}{ds} = -k\vec{\tau}$$

Where k represents the road curvature.

Then the equation (4.16) can be rewritten as

$$\dot{d} = (|\vec{v}|\dot{\tau} - \vec{n}_r) + d\vec{n}_r \cdot (-k\dot{s}\overrightarrow{\tau_r})$$

$$= |\vec{v}||\vec{\tau}||\vec{n}_r|cos < \vec{\tau}, \vec{n}_r >$$

$$= |\vec{v}| \cos\left(\frac{\pi}{2} - (\theta - \theta_r)\right)$$

$$\dot{d} = |\vec{v}|\sin(\theta - \theta_r) \qquad (4.20)$$

(2) Calculation about $\dot{s}$

Then we need to calculate the velocity vector's projection vector $\dot{s}$, considering

$$\vec{x}_r + d\vec{n}_r = \vec{x}$$

Then making derivation of each side

$$\overrightarrow{\dot{x}_r} + \dot{d}\vec{n}_r + d\overrightarrow{\dot{n}_r} = \dot{\vec{x}}$$

Combining with the equation (4.18) and (4.19)

$$\dot{s}\vec{\tau}_r + |\vec{v}|\sin(\theta - \theta_r)\,\vec{n}_r + d(-k\dot{s}\vec{\tau}_r) = |\vec{v}|\vec{\tau}$$

Multiply both sides of the equation by a $\vec{\tau}_r$

$$\dot{s} + (-kd\dot{s}) = |\vec{v}|\vec{\tau}\cdot\vec{\tau}_r = |\vec{v}|\cos(\theta - \theta_r)$$

So $\dot{s}$ can be calculated:

$$\dot{s} = \frac{|\vec{v}|\cos(\theta - \theta_r)}{1 - kd} \qquad (4.21)$$

In conclusion, from (4.20) and (4.21) we can know:

$$\begin{cases} \dot{d} = |\vec{v}|\sin(\theta - \theta_r) \\ \dot{s} = \dfrac{|\vec{v}|\cos(\theta - \theta_r)}{1 - kd} \end{cases}$$

Because that the yaw angle $\theta = \beta + \psi$

$$\dot{d} = |\vec{v}|\sin(\theta - \theta_r)$$

$$= |\vec{v}|sin\beta\cos(\psi - \theta_r) + |\vec{v}|cos\beta\sin(\psi - \theta_r)$$

$$= v_y\cos(\psi - \theta_r) + v_x\sin(\psi - \theta_r)$$

Assuming $\psi - \theta_r$ is a small angle, then $\dot{d}$ can be simplified as

$$\dot{d} \approx v_y + v_x(\psi - \theta_r) \qquad (4.22)$$

(3) Error matrix

Let

$$e_d = d \quad e_\psi = \psi - \theta_r$$

Therefore, according to the (4.21) and (4.22)

$$\dot{e_d} = v_y + v_x(\psi - \theta_r)$$

$$v_y = \dot{e_d} - v_x e_\psi \qquad\qquad \dot{v_y} = \ddot{e_d} - v_x \dot{e_\psi}$$

$$\dot{e_\psi} = \dot{\psi} - \dot{\theta}_r \qquad\qquad \ddot{e_\psi} = \ddot{\psi} - \ddot{\theta}_r \approx \ddot{\psi}$$

$\ddot{\theta}_r$ being neglected because the road curvature is commonly not too large.

In conclusion,

$$\begin{cases} v_y = \dot{e_d} - v_x e_\psi \\ \dot{v_y} = \ddot{e_d} - v_x \dot{e_\psi} \\ \dot{\psi} = \dot{e_\psi} + \dot{\theta}_r \\ \ddot{\psi} = \ddot{e_\psi} \end{cases} \qquad (4.23)$$

Combining with the vehicle dynamic model in (4.9)

$$\begin{pmatrix} \dot{e}_d \\ \ddot{e}_d \\ \dot{e}_\psi \\ \ddot{e}_\psi \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & \dfrac{C_{\alpha f} + C_{\alpha r}}{m v_x} & -\dfrac{C_{\alpha f} + C_{\alpha r}}{m} & \dfrac{l_f C_{\alpha f} - l_r C_{\alpha r}}{m v_x} \\ 0 & 0 & 0 & 2 \\ 0 & \dfrac{l_f C_{\alpha f} - l_r C_{\alpha r}}{J v_x} & -\dfrac{l_f C_{\alpha f} - l_r C_{\alpha r}}{J} & \dfrac{{l_f}^2 C_{\alpha f} + {l_r}^2 C_{\alpha r}}{J v_x} \end{pmatrix} \begin{pmatrix} e_d \\ \dot{e}_d \\ e_\psi \\ \dot{e}_\psi \end{pmatrix} + \begin{pmatrix} 0 \\ -\dfrac{C_{\alpha f}}{m} \\ 0 \\ -\dfrac{l_f C_{\alpha f}}{J} \end{pmatrix} \delta$$

$$+ \begin{pmatrix} 0 \\ \dfrac{l_f C_{\alpha f} - l_r C_{\alpha r}}{m v_x} - v_x \\ 0 \\ \dfrac{{l_f}^2 C_{\alpha f} + {l_r}^2 C_{\alpha r}}{J v_x} \end{pmatrix} \dot{\theta}_r \qquad (4.24)$$

In short, it is

$$\dot{e}_{rr} = A e_{rr} + B u + C \dot{\theta}_r \qquad (4.25)$$

Where

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & \dfrac{C_{\alpha f} + C_{\alpha r}}{m v_x} & -\dfrac{C_{\alpha f} + C_{\alpha r}}{m} & \dfrac{l_f C_{\alpha f} - l_r C_{\alpha r}}{m v_x} \\ 0 & 0 & 0 & 2 \\ 0 & \dfrac{l_f C_{\alpha f} - l_r C_{\alpha r}}{J v_x} & -\dfrac{l_f C_{\alpha f} - l_r C_{\alpha r}}{J} & \dfrac{{l_f}^2 C_{\alpha f} + {l_r}^2 C_{\alpha r}}{J v_x} \end{pmatrix} \quad B = \begin{pmatrix} 0 \\ -\dfrac{C_{\alpha f}}{m} \\ 0 \\ -\dfrac{l_f C_{\alpha f}}{J} \end{pmatrix}$$

$$C = \begin{pmatrix} 0 \\ \dfrac{l_f C_{\alpha f} - l_r C_{\alpha r}}{m v_x} - v_x \\ 0 \\ \dfrac{{l_f}^2 C_{\alpha f} + {l_r}^2 C_{\alpha r}}{J v_x} \end{pmatrix}$$

(4) Calculating the feedback gain K using LQR

Considering $\dot{e}_{rr} = A e_{rr} + Bu$ firstly, to find

$$u_k = -K e_{rr}(k) \qquad (4.26)$$

At present, most control system use digital computers to implement the controllers. So, the control system must be discretized.

Considering

$$\dot{x} = Ax + Bu$$

Integrating the both sides

$$\int_t^{t+dt} \dot{x} = \int_t^{t+dt} (Ax + Bu) dt$$

According to the Integral Mid-value Theorem

$$\Rightarrow x(t + dt) - x(t) = Ax(\xi)dt + Bu(\xi)dt \qquad (4.27)$$

To determine the value of ξ, we have three methods

- Forward Euler Method

  Assume $x(\xi) = x(t)$

- Backward Euler method

  Assume $x(\xi) = x(t + dt)$

- Midpoint Euler method

Assume $x(\xi) = \frac{x(t)+x(t+dt)}{2}$

We consider the $x(\xi)$ in Midpoint Euler method and the $u(\xi)$ the Backward Euler Method since the value of $u(t+dt)$ is unable to know. Therefore, the equation (4.27) can be rewritten as

$$x(t+dt) = Ax(\xi)dt + x(t) + Bu(\xi)dt$$

$$= Adt\left(\frac{x(t)+x(t+dt)}{2}\right) + x(t) + Bdt \cdot u(t)$$

$$\left(I - \frac{Adt}{2}\right)x(t+dt) = \left(I + \frac{Adt}{2}\right)x(t) + Bdt \cdot u(t)$$

$$x(t+dt) = \left(I - \frac{Adt}{2}\right)^{-1}\left(I + \frac{Adt}{2}\right)x(t) + \left(I - \frac{Adt}{2}\right)^{-1} Bdt \cdot u(t)$$

$$\approx \left(I - \frac{Adt}{2}\right)^{-1}\left(I + \frac{Adt}{2}\right)x(t) + Bdt \cdot u(t)$$

Assume $dt = 0.01$,

$$x(k+1) = \bar{A}x(k) + \bar{B}u(k)$$

Where

$$\bar{A} = \left(I - \frac{Adt}{2}\right)^{-1}\left(I + \frac{Adt}{2}\right) \qquad \bar{B} = Bdt$$

And the feedback control gain K can be calculated as

$$K = -(R + \bar{B}^T P \bar{B}^{-1})\bar{B}^T P \bar{A} \qquad (4.28)$$

During the Simulink model construction process, we use the Matlab command lqr(A,B,Q,R), since the LQR computation depends on system matrix A and B which is only related to the vehicle parameter and the longitudinal velocity v_x, assume in normal working conditions, the vehicle parameter slightly change, we can get to the conclusion that for every longitudinal velocity v_x, we get a control gain solution. In order to solve the LQR more quickly, we can create offline a table containing all control gain corresponding to different v_x, and every time we need to compute a new gain K, we look up the value in the table. The complete codes can be looked up in Appendix.

(5) Feedforward control

Considering equation (4.25) and (4.26)

$$\dot{e}_{rr} = Ae_{rr} + B(ue_{rr} + \delta_f) + C\dot{\theta}_r$$

Here we need to find an optimal δ_f to make the $e_{rr} = -(A - BK)^{-1} \cdot (B\delta_f + C\dot{\theta}_r)$ infinitely close to 0. Assume the feedback control gain $K = (K_1\ K_2\ K_3\ K_4\ )$

Then e_{rr} can be calculated:

$$e_{rr} = \begin{pmatrix} \frac{1}{K_1}\left\{\delta_f - \frac{\dot{\theta}_r}{v_x}\left[l_f + l_r - l_r K_3 - \frac{m{v_x}^2}{l_f + l_r}\left(\frac{l_r}{C_f} + \frac{l_f}{C_r}K_3 - \frac{l_f}{C_r}\right)\right]\right\} \\ 0 \\ -\frac{\dot{\theta}_r}{v_x}(l_r + \frac{l_f}{l_f + l_r} \cdot \frac{m{v_x}^2}{C_r}) \\ 0 \end{pmatrix} = \begin{pmatrix} e_d \\ \dot{e_d} \\ e_\psi \\ \dot{e_\psi} \end{pmatrix} \tag{4.29}$$

Our aim is to make e_{rr} infinitely close to 0.

Considering $e_d = 0$,

$$e_d = \frac{1}{K_1}\left\{\delta_f - \frac{\dot{\theta}_r}{v_x}\left[l_f + l_r - l_r K_3 - \frac{m{v_x}^2}{l_f + l_r}\left(\frac{l_r}{C_f} + \frac{l_f}{C_r}K_3 - \frac{l_f}{C_r}\right)\right]\right\} = 0$$

$$\delta_f = \frac{\dot{\theta}_r}{v_x}\left[l_f + l_r - l_r K_3 - \frac{m{v_x}^2}{l_f + l_r}\left(\frac{l_r}{C_f} + \frac{l_f}{C_r}K_3 - \frac{l_f}{C_r}\right)\right] \tag{4.30}$$

Considering e_ψ

$$e_\psi = -\frac{\dot{\theta}_r}{v_x}\left(l_r + \frac{l_f}{l_f + l_r} \cdot \frac{m{v_x}^2}{C_r}\right) \tag{4.31}$$

e_ψ is not affected by δ_f and K

Operating on e_ψ to get simplification. Considering equation (4.21)

$$\dot{s} = \frac{|\vec{v}|\cos(\theta - \theta_r)}{1 - k \cdot e_d} = \frac{|\vec{v}|\cos(\beta + \psi - \theta_r)}{1 - k \cdot e_d}$$

$$= \frac{|\vec{v}|\cos\beta\cos\left(\psi-\theta_r\right) - |\vec{v}|sin\beta\sin\left(\psi-\theta_r\right)}{1-k\cdot e_d}$$

$$\dot{s} = \frac{v_x \cos e_\psi - v_y \sin e_\psi}{1-k\cdot e_d} \qquad (4.32)$$

Considering the definition of curvature

$$k = \frac{d\theta}{ds} = \frac{\dot{\theta}}{\dot{s}} \qquad \Rightarrow \dot{\theta}_r = k\dot{s} \qquad (4.33)$$

Assuming the

$$|k| \ll 1 \qquad |e_\psi| \ll 1 \qquad |v_y| \ll 1$$

Therefore

$$\frac{1}{1-k\cdot e_d} \approx 1 \qquad v_x \cos e_\psi \approx\ v_x \qquad v_y \sin e_\psi \approx 0$$

Then we can say

$$\dot{s} \approx v_x$$

Then equation (4.33) can be rewritten as

$$\dot{\theta}_r = k v_x$$

Returning to equation (4.31),

$$e_\psi = -k\left(l_r + \frac{l_f}{l_f+l_r}\cdot\frac{m{v_x}^2}{C_r}\right)$$

$$= -\left(\frac{l_r}{R} + \frac{l_f}{l_f+l_r}\cdot\frac{m{v_x}^2}{R}\frac{1}{C_r}\right) \qquad (4.34)$$

From equation (4.5) we can know $a_y = \dot{v_y} + v_x\dot{\psi}$

Where

$$\dot{\psi} = \frac{\vec{v}}{R} = \frac{\overrightarrow{v_x}+\overrightarrow{v_y}}{R} \approx \frac{\overrightarrow{v_x}}{R}$$

Then a_y is approximately equal to $a_y \approx \frac{v_x^2}{R}$

e_ψ can be further abbreviated as

$$e_\psi = -(\frac{l_r}{R} + \frac{l_f}{l_f + l_r} m a_y \cdot \frac{1}{C_r})$$

And we can know from Chassis Design that

$$\frac{l_f}{l_f + l_r} m = m_r$$

$$e_\psi = -\left(\frac{l_r}{R} + \frac{m_r a_y}{C_r}\right) = -\left(\frac{l_r}{R} + \alpha_r\right) \qquad (4.35)$$

Where α_r is the side slip angel of the rear wheel.

Furthermore, we can consider geometrical relationship in Bicycle model

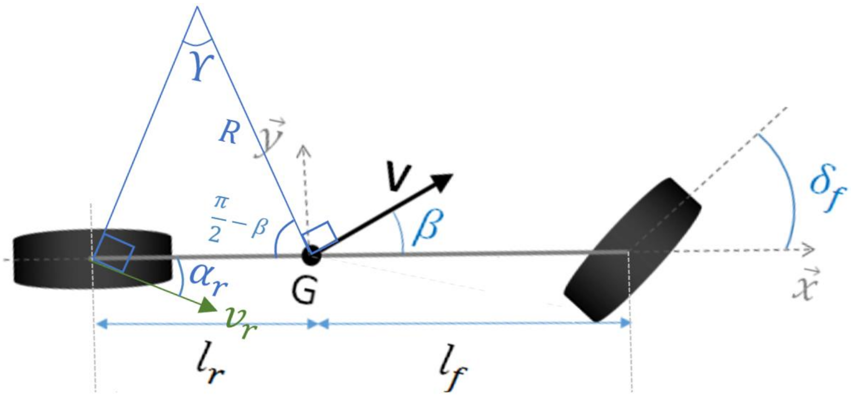


Figure 4.15 geometrical relationship in bicycle model

$$\gamma + \frac{\pi}{2} - \beta + \frac{\pi}{2} - (-\alpha_r) = \pi$$

$$\Rightarrow -\beta = -(\gamma + \alpha_r) = -(\frac{l_r}{R} + \alpha_r)$$

Combined with equation (4.35) we can know that

$$e_\psi = -\beta \qquad (4.36)$$

That means the steady state error of e_ψ is equal to $-\beta$. Since we define $e_\psi = \psi - \theta_r$, this means the error of yaw angle $\theta - \theta_r$ will be infinitely close to 0 when e_ψ reaches $-\beta$.

In conclusion, the optimal answer of the whole system can be written as

$$u = -Ke_{rr} + \delta_f$$

And from equation (4.30),

$$\delta_f = \frac{\dot{\theta}_r}{v_x}\left[l_f + l_r - l_r K_3 - \frac{m{v_x}^2}{l_f + l_r}\left(\frac{l_r}{C_f} + \frac{l_f}{C_r}K_3 - \frac{l_f}{C_r}\right)\right]$$

$$= k\left[l_f + l_r - l_r K_3 - - \frac{m{v_x}^2}{l_f + l_r}\frac{l_r}{}(\frac{l_r}{C_f} + \frac{l_f}{C_r}K_3 - \frac{l_f}{C_r})\right]$$

Where k is the road curvature and K_3 is the third parameter in controller $K = [K_1\ \ K_2\ \ K_3\ \ K_4\ ]$

The scheme of the system control is shown as follow

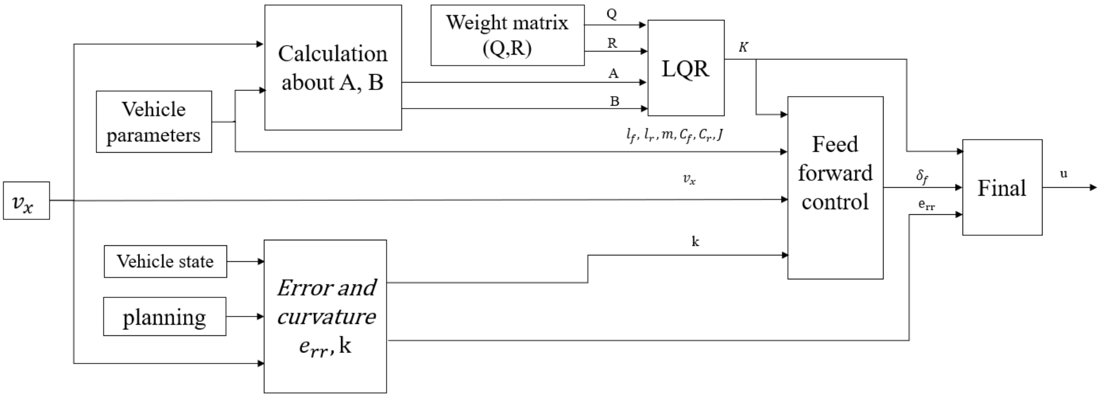


Figure 4.16 work scheme

4.4 CarSim-Simulink Model

4.4.1 Scenario construction in Carsim

A two lanes scenario was set up in CarSim

And the parameters about the road are as follows.

Friction coefficient: 0.85

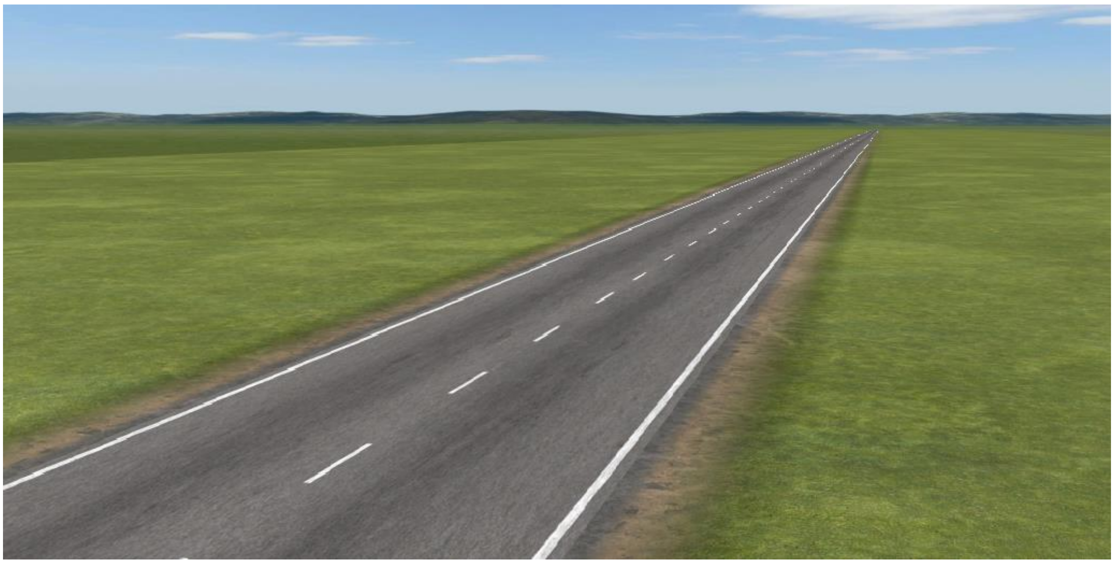


Figure 4.17 scenario set up

Adding two C-class vehicles on the road.

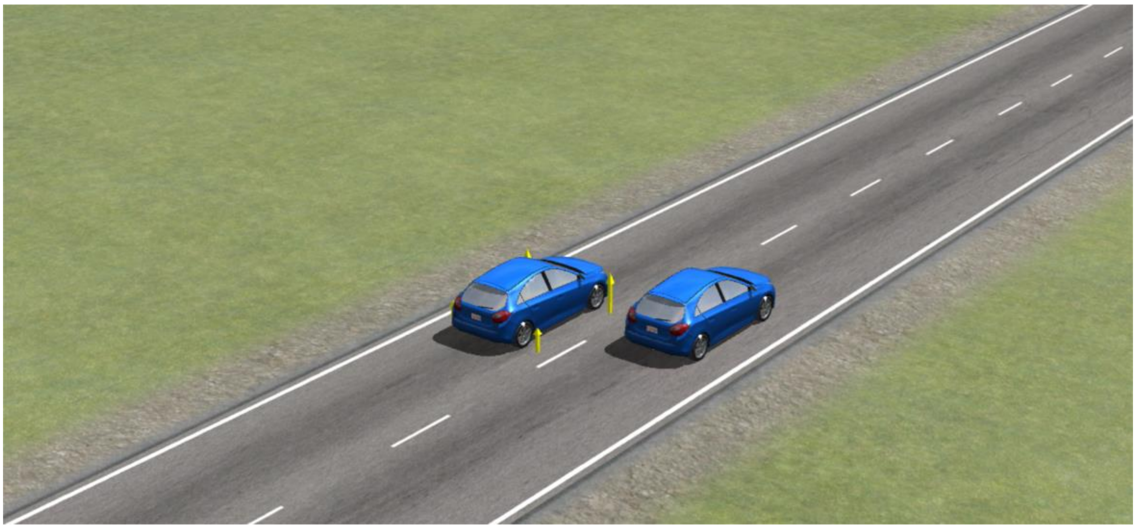
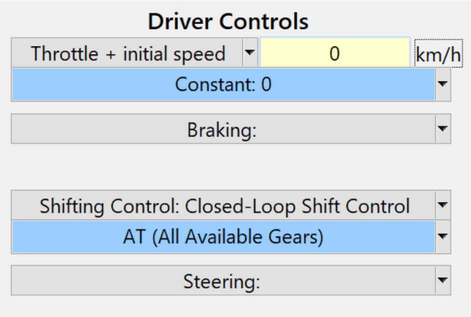


Figure 4.18 C class vehicle on road

Cancel the effect of the braking and steering because we need to complete the brake and steer in Matlab.



Driver Controls

Throttle + initial speed ▾ | 0 | km/h

Constant: 0 ▾

Braking: ▾

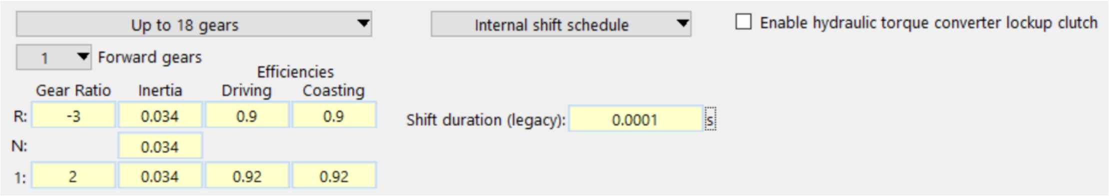
Shifting Control: Closed-Loop Shift Control ▾

AT (All Available Gears) ▾

Steering: ▾

Figure 4.19 Carsim Model settings

Modifying the powertrain settings to simulate the electric motor:



Up to 18 gears ▾

1 ▾ Forward gears

Internal shift schedule ▾

□ Enable hydraulic torque converter lockup clutch

Shift duration (legacy): 0.0001 s

	Gear Ratio	Inertia	Efficiencies Driving	 Coasting
R:	-3	0.034	0.9	0.9
N:		0.034		
1:	2	0.034	0.92	0.92

Figure 4.20 Carsim powertrain settings

After completing the CarSim settings, a vehicle model can be sent to Simulink.

4.4.2 The ego car Simulink Model

We choose double PID to control the system. One is for position PID control, another is for velocity PID control, the scheme is as follow:

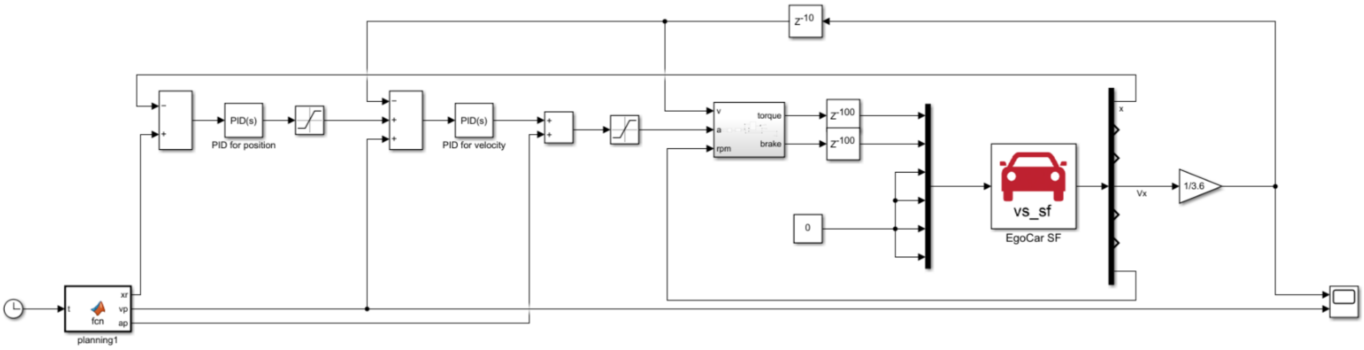


Figure 4.21 Double PID for ego car's longitudinal control

4.4.3 The side car Simulink Model

Form the instruction shown in fig.4.17, a complete Simulink model used to accomplish the longitudinal and lateral control of the side car's lane change behavior was set up.

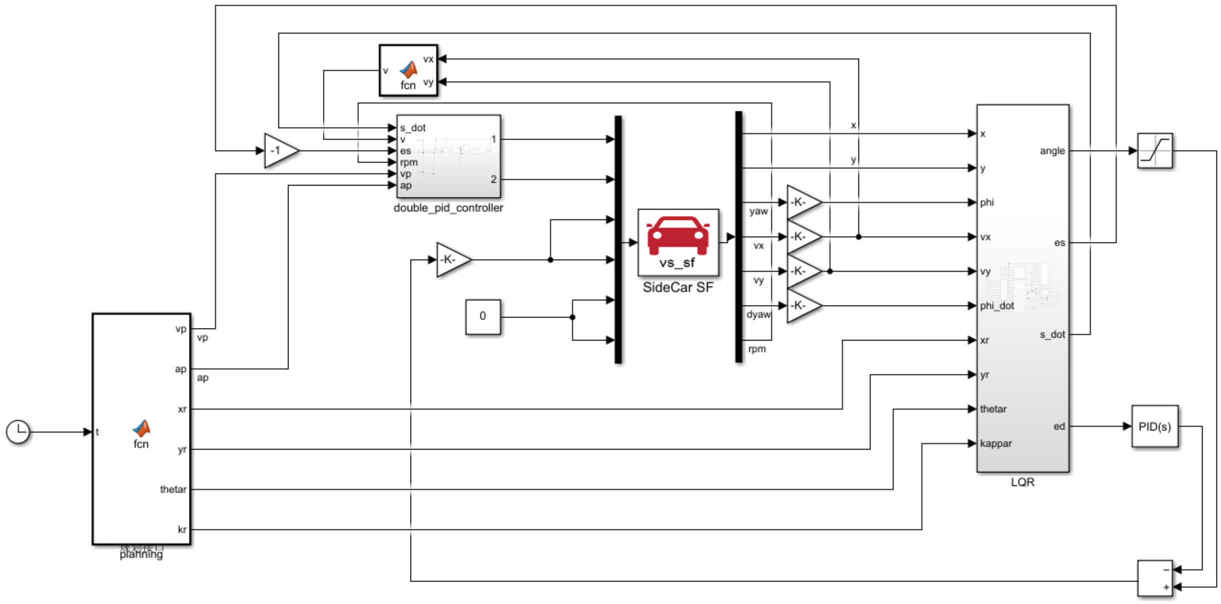


Figure 4.22 Simulink model of side car control

In order to eliminate the control inaccuracy caused by understeer characteristics of the vehicle, a PID module(only Proportional part valid) was added to compensate the steering angle error.

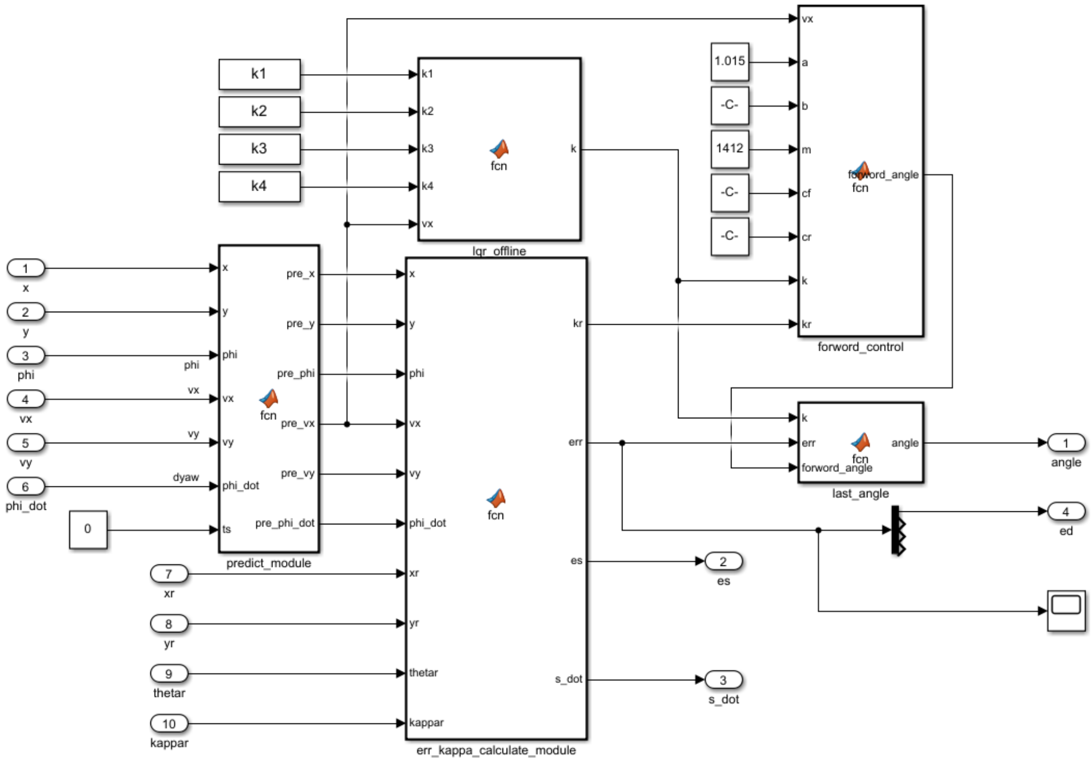


Figure 4.23 LQR module

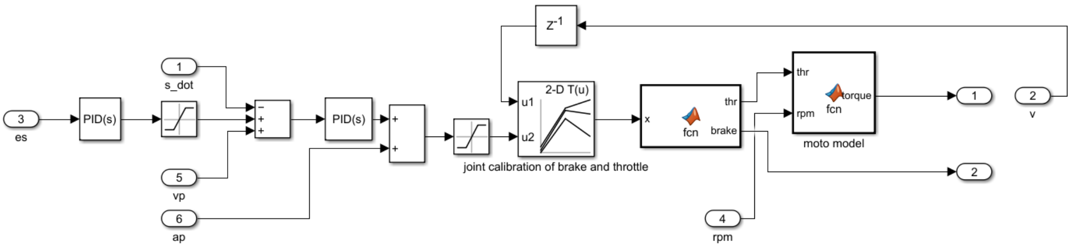


Figure 4.24 double PID module

Chapter 5 Results and Analyses

5.1 Trajectory generated under different conditions

Choosing two typical trajectories generated by Algorithm 1 in Chapter 3, We consider two cases to furtherly verify our motion control module.

Case 1 corresponds to the solution (1,1) in Chapter 2 in table 2.1, where the side car executes lane change before the ego car.

Initial conditions:

$v_{ego} = 6m/s, X_0 = 0m, X_0 = 1.875m$

$v_{side} = 5m/s, X_0 = 8m, Y_0 = -1.875m$

Case 2 corresponds to the solution (2,2) in Chapter 2 in table 2.1, where the side car completes the lane change maneuver behind the ego car.

Initial conditions:

$v_{ego} = 7m/s, X_0 = 0m, X_0 = 1.875m$

$v_{side} = 5m/s, X_0 = 2m, Y_0 = -1.875m$

5.1.1 Case 1

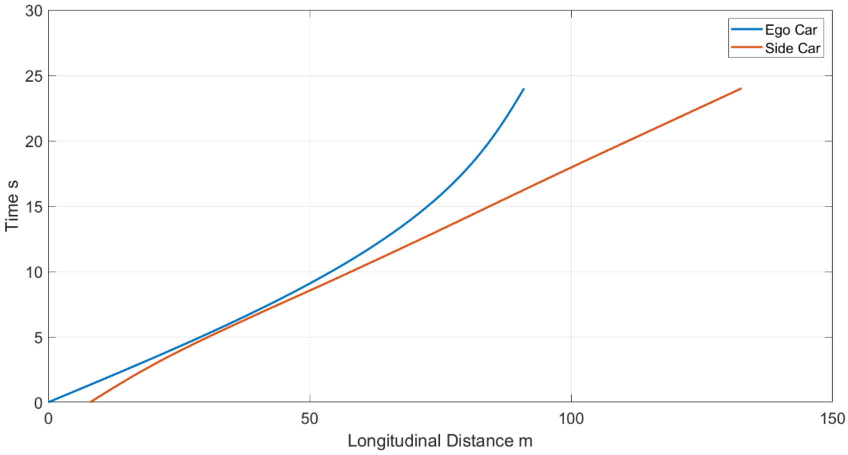


Figure 5.1 longitudinal distance comparison

Ego car:

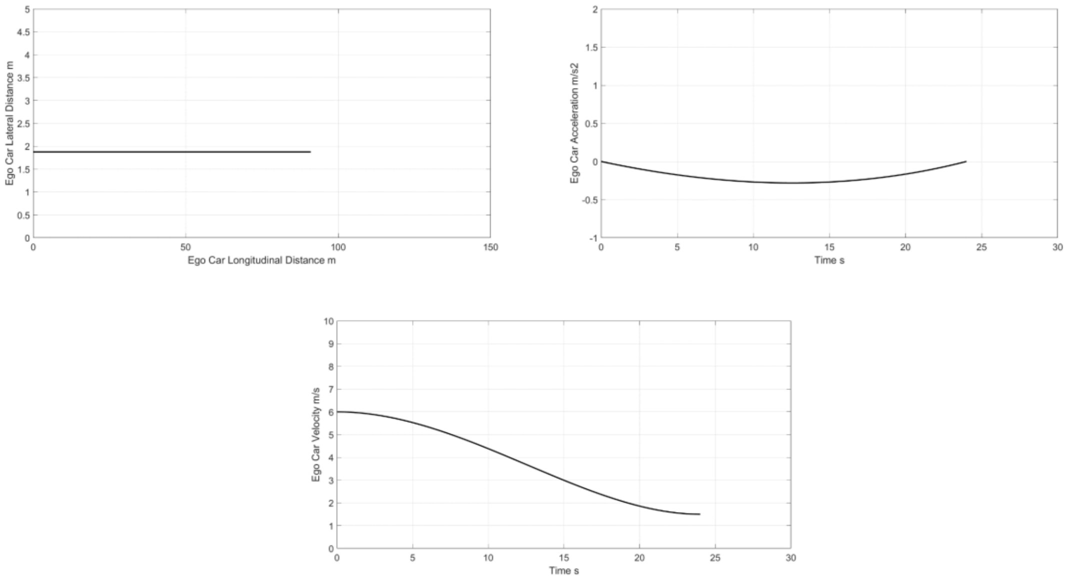
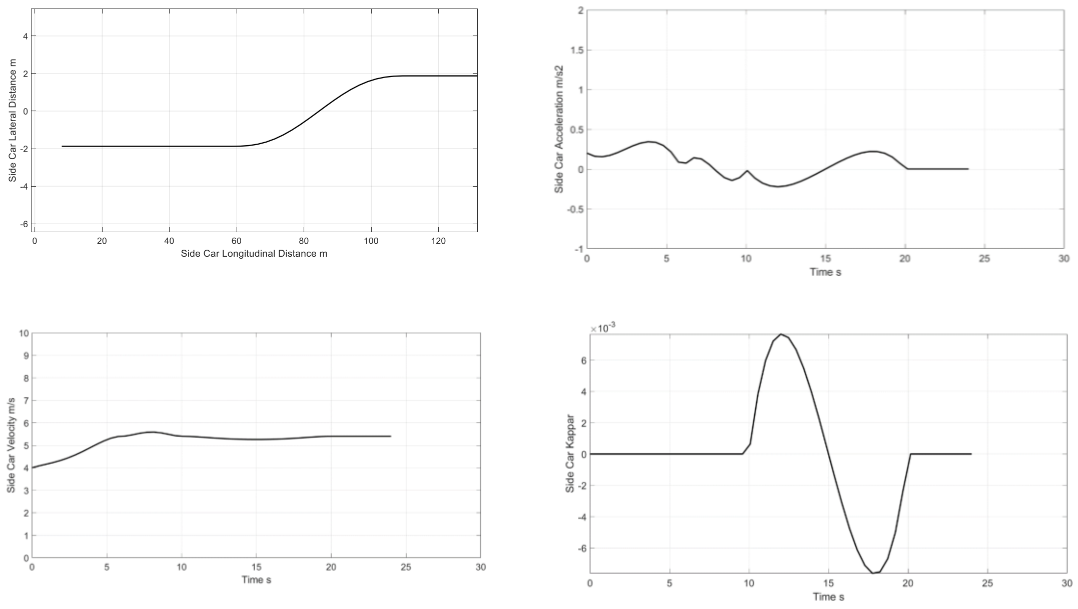


Figure 5.2 ego car planning output

Side car:



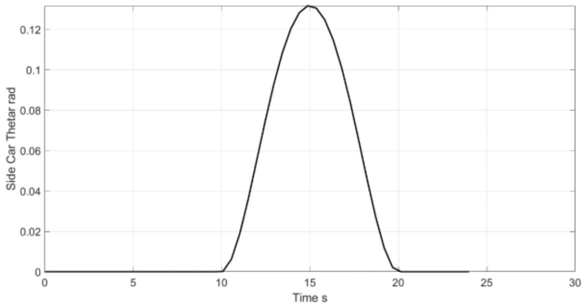


Figure 5.3 side car planning output

In figure 5.3

0-6s: side car accelerates;
6-10s: side car is moving at constant speed;
10-20s: side car completes the lane change maneuver.

5.1.2 Case 2

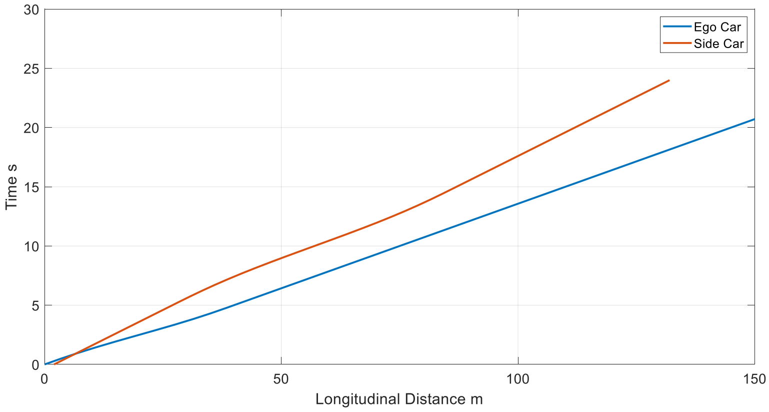


Figure 5.4 longitudinal distance comparison

Ego car:

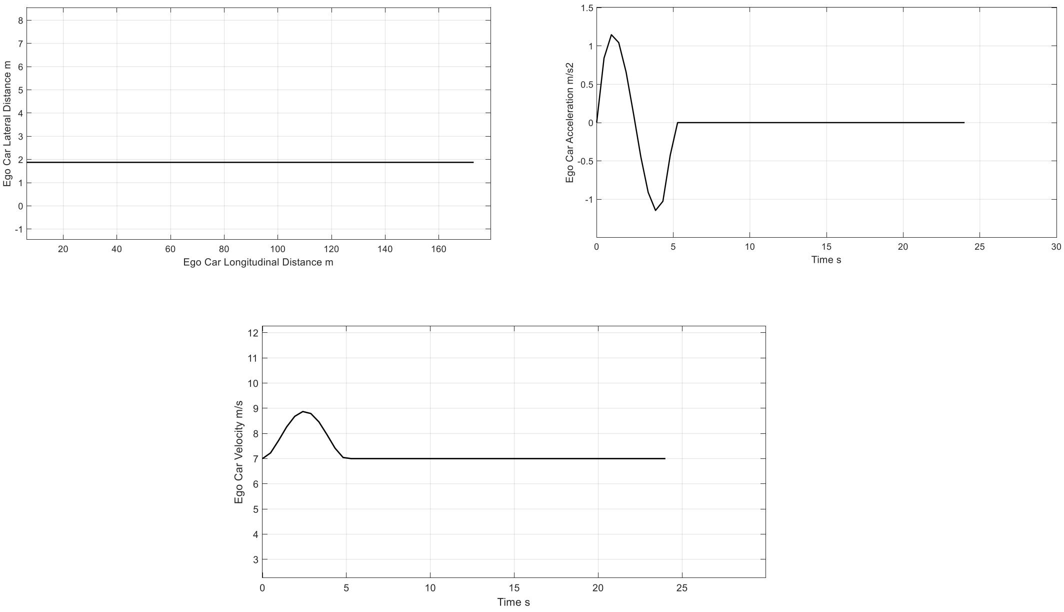
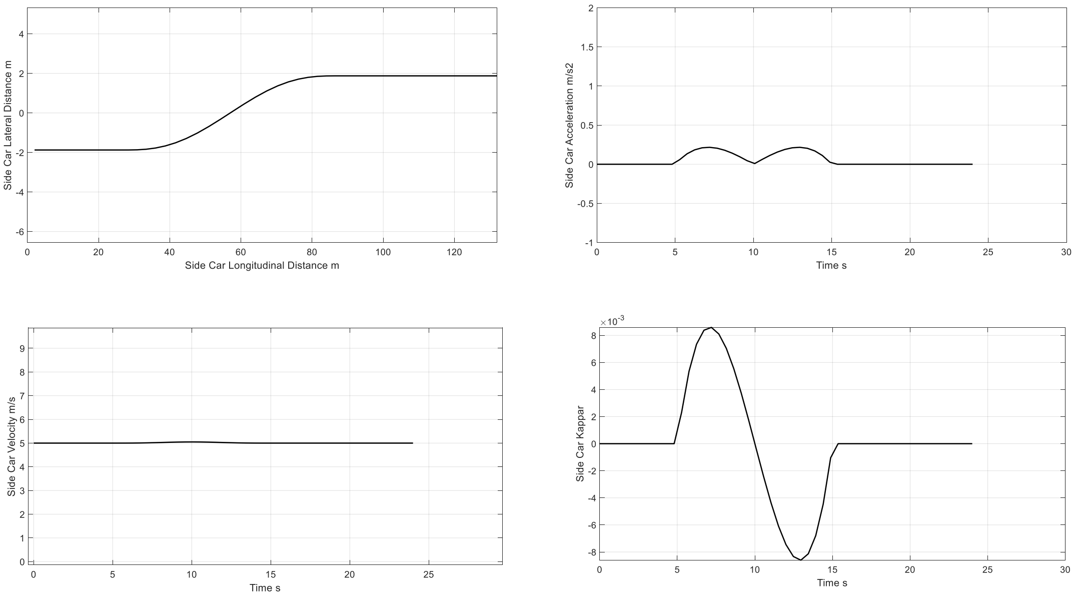


Figure 5.5 ego car planning output

Side Car:



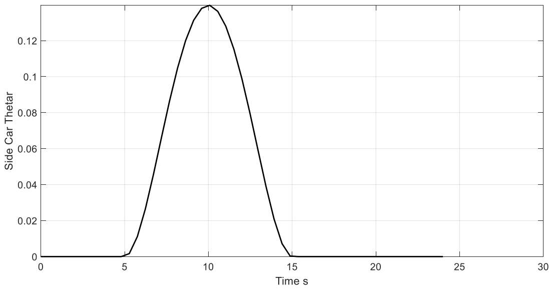


Figure 5.6 side car planning output

5.2 Control effect

After tuning the parameters in LQR and PID controller, the accuracy of control is at the centimeter level.

5.2.1 Case 1

Ego car:

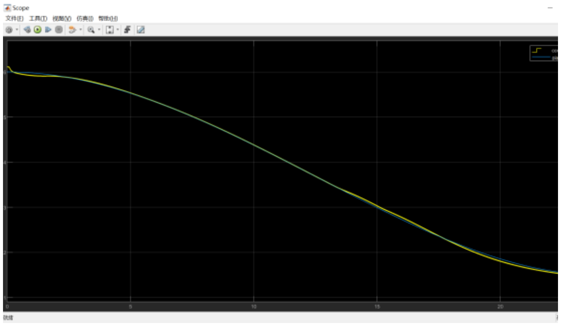


Figure 5.7 ego car velocity track

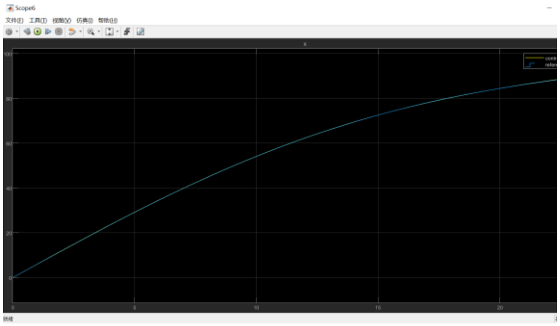


Figure 5.8 ego car trajectory track

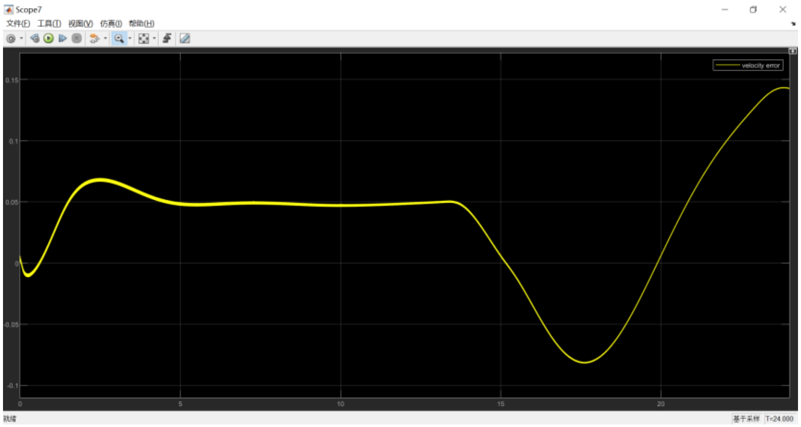


Figure 5.9 ego car longitudinal error

The maximum longitudinal error is 0.15m.

Side car:

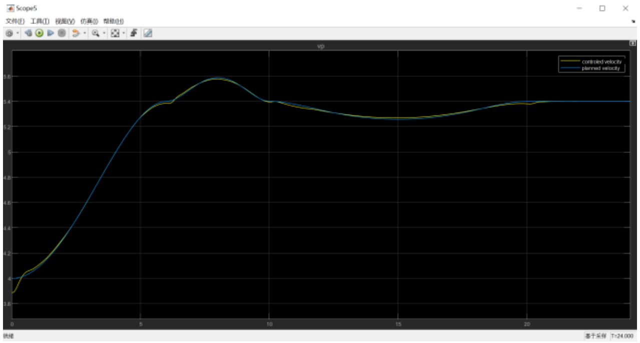


Figure 5.10 side car velocity track

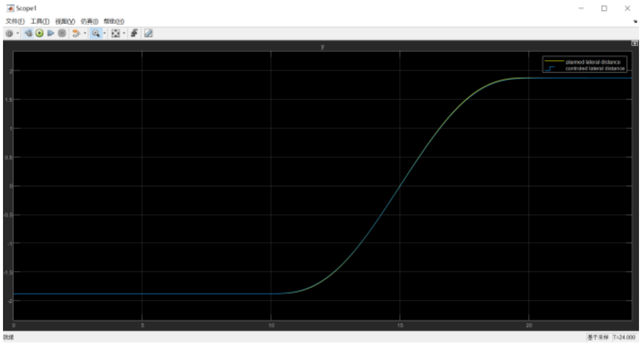


Figure 5.11 side car lateral distance track

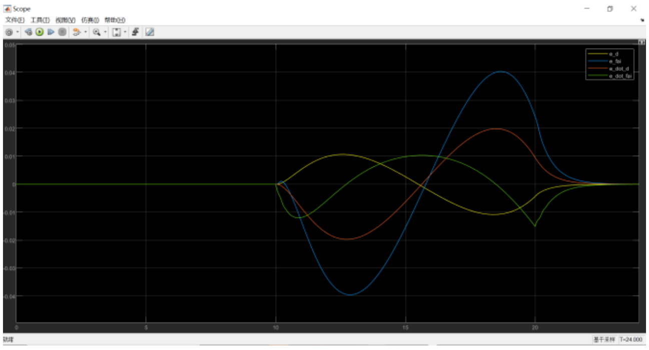


Figure 5.12 side car lateral control error

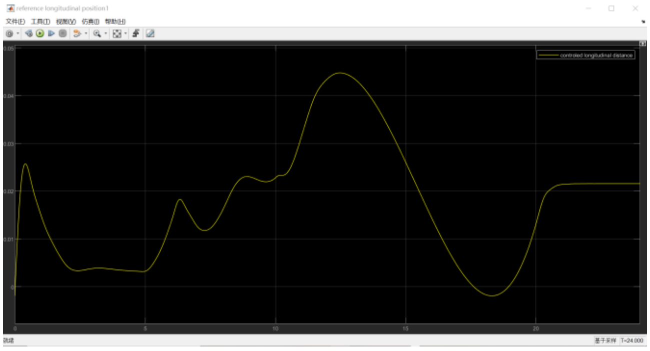


Figure 5.13 side car longitudinal error

The physical meaning of the four elements of the errors:

$$e_{rr} = \begin{pmatrix} e_d \\ \dot{e}_d \\ e_\psi \\ \dot{e}_\psi \end{pmatrix}$$

The blue line represents the error of e_ψ which is equal to $-\beta$ from equation (4.36) in Chapter 4. And we can see from the fig that all errors are in centimeter level.(longitudinal error within 4.5cm and lateral error within 1cm)

5.2.2 Case 2

Ego car:

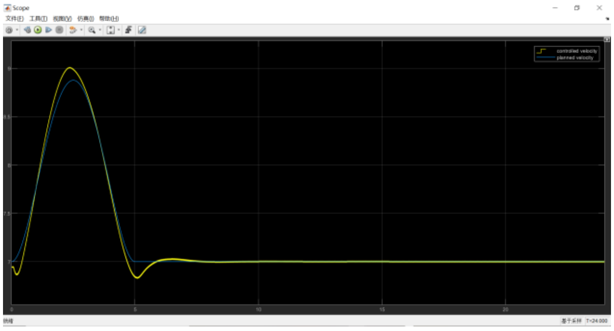


Figure 5.14 ego car velocity track

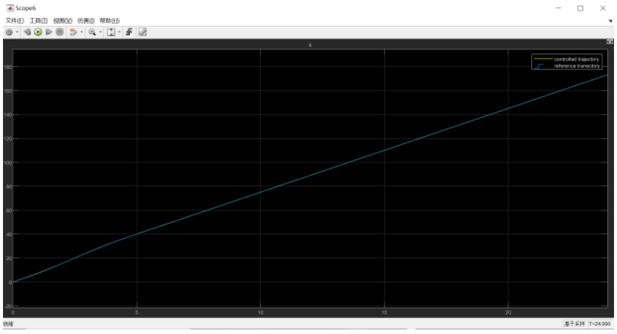


Figure 5.15 ego car trajectory track

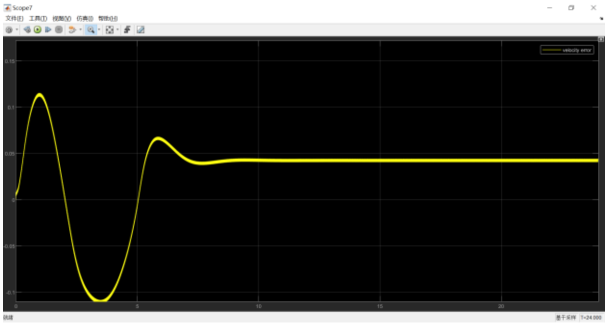


Figure 5.16 ego car longitudinal error

The maximum longitudinal tracking error is 10cm, which is acceptable.

Side car:

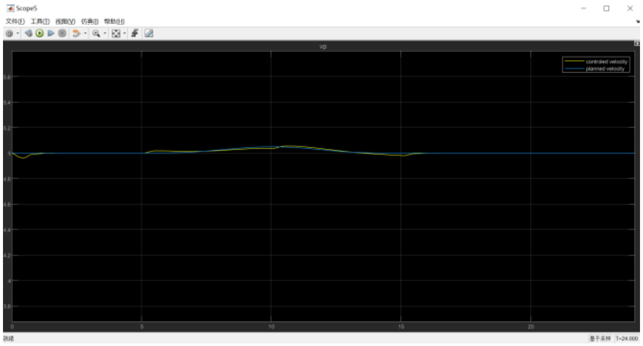


Figure 5.17 side car velocity track

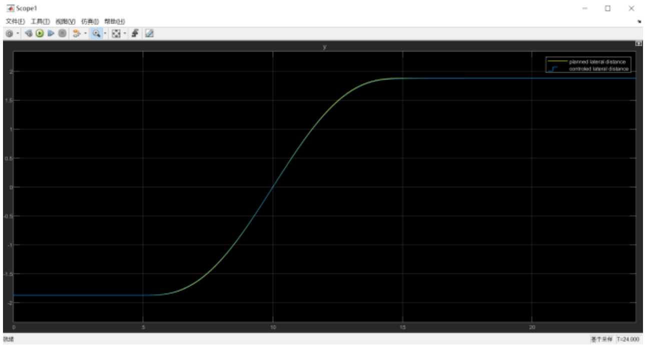


Figure 5.18 side car lateral distance track

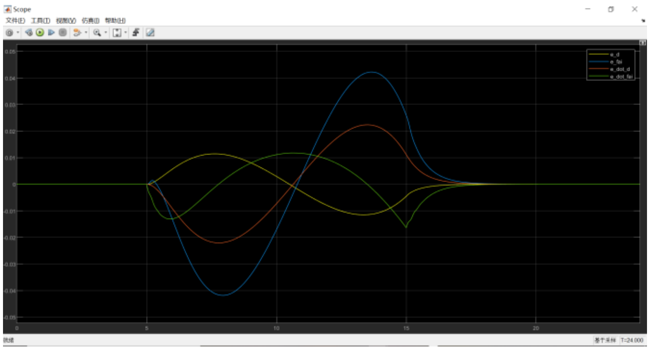


Figure 5.19 side car lateral control error

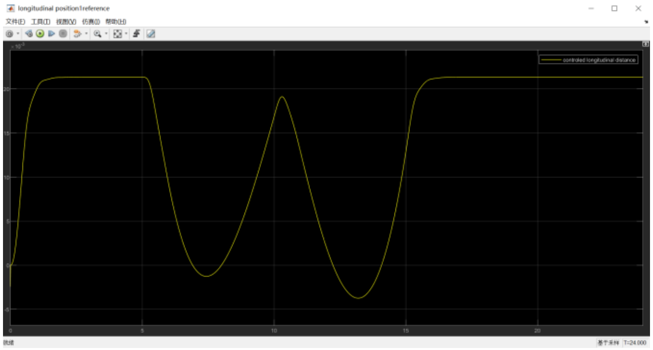


Figure 5.20 side car longitudinal control error

We can see that all errors except the e_ψ represented by the blue line are smaller than 0.02. We know from Chapter4 that e_ψ is more over equal to $-\beta$ so the value of e_ψ is also acceptable. And the longitudinal control error is limited within 2.2cm.

Appendix Ⅰ : Planning codes

Cut-in main codes:

```
%% Cut-in main function
global v1 v2 d1 d2 l1 l2 w % Initial value: 1-Initial speed of the ego vehicle; 2-Initial speed of the side vehicle; 3-Initial position of the ego vehicle; 4-Initial position of the side vehicle; 5-Length of the ego vehicle; 6-Length of the side vehicle; 7-Lane width (side distance of changing lanes)l1=5;
l2=5;
w=3.75;

% Example: the ego car is behind the side car, the ego car goes first
% v1=6;
% v2=4;
% d1=0;
% d2=2;

% Example: the ego car is behind the side car, the ego car yields
% v1=5;
% v2=4;
% d1=0;
% d2=8;

% Example: The side car is behind the ego car, the ego car yields
% v1=4;
% v2=8;
% d1=3;
% d2=0;

% Example: The side car is behind the ego car, the ego car goes first
v1=4;
v2=5;
d1=6;
d2=0;

tic;
```

```
% Design variables [1-ego car acceleration 2- side car acceleration 3-Sigmoid's k value 4-by-side vehicle shift time]
x1=[-4 4]; % Acceleration value range of ego car
x2=[-4 4]; % Acceleration value range of side car
x3=[0.05 4]; % characteristic value of sigmoid function k’s value range
x4=[0.5 6]; % Range of time of the lane change of sidecar
options=gaoptimset('Generations',1000);%'CrossoverFcn',@Heuristic); %Set the algebra solving of “genetic algorithm” to 1000
options=gaoptimset(options,'CrossoverFcn',@crossoverheuristic); % Set the cross function of genetic algorithm to Heuristic
options=gaoptimset(options,'InitialPopulation',[-2 2 2 8]);  %Set the optimized initial value
[X,Target]=ga(@Cut_in_objective,4,[],[],[],[],[x1(1) x2(1) x3(1) x4(1)],[x1(2) x2(2) x3(2) x4(2)],@Cut_in_constraints1,options);

v2m=v2+X(2)*X(4);
scm=Integrated_sigmoid([10/X(3) 0 X(3) w]);
tcm=scm/v2m;
tm=tcm+X(4);
tra_tf=linspace(0,X(4));
tra_tr=linspace(X(4),tm+tcm);
tra_t=[tra_tf tra_tr];
tra_x1=d1+v1*tra_t+0.5*X(1)*tra_t.^2;
tra_y1=zeros(1,200);
tra_x2_f=d2+v2*tra_tf+0.5*X(2)*tra_tf.^2;
tra_y2_f=-w*ones(1,100);
tra_x2_r=zeros(1,100);
tra_y2_r=zeros(1,100);
for i=1:100
    ss=(tra_tr(i)-X(4))*v2m;
    C=Length_for_coordinates([X(3) ss]);
    tra_x2_r(i)=C(1)+max(tra_x2_f)+10/X(3);
    tra_y2_r(i)=C(2)-w;
end
tra_x2=[tra_x2_f tra_x2_r];
tra_y2=[tra_y2_f tra_y2_r];
plot3(tra_x1,tra_y1,tra_t,tra_x2,tra_y2,tra_t);
axis equal;
toc;
```

Subfunction 1: objective function design

```
function y=Cut_in_objective(X) %% Design variables [1-ego car acceleration
2-side car acceleration 3-Sigmoid's k value 4-side car lane change time]
global v1 v2 d1 d2 w  % Initial value (global variable): 1-Initial speed of
the ego car 2-Initial speed of the side car 3-Initial position of the ego
car 4-Initial position of the side car 5-Lane width
J1=abs(X(1))+abs(X(2)); % The first objective function is to minimize the
sum of the accelerations of the two cars
v2m=v2+X(2)*X(4);
expe_cutangle=Cut_in_angle(v2m);
real_cutangle=atan(w*X(3)/4);
J2=abs(expe_cutangle-real_cutangle); % The second goal is to minimize the
difference between the actual and ideal entry angles
y=J1+20*J2;
```

Subfunction 2: constraints

```
%% Cut-in planning algorithm constraints (considering multiple situations)
function [c,ceq]=Cut_in_constraints1(X) % Design variables [1-ego car
acceleration 2-side car acceleration 3-Sigmoid's k value 4-side car lane
change time]
global v1 v2 d1 d2 l1 l2 w  %Initial value (global variable): 1-Initial
speed of ego car 2-Initial speed of side car 3-Initial position of ego car
4-Initial position of side car 5-Length of ego car 6-Length of side car 7-
lane width (lane change side Distance)
ttc=(d2-d1)/(v2-v1);
v2m=v2+X(2)*X(4);
sx=10/X(3); % Sigmoid curve X-direction range, ±sx
S=Integrated_sigmoid([-sx 0 X(3) w]);  % Length of half a sigmoid
tcm=S/v2m; % Half the time of the sigmoid path
tm=X(4)+tcm; % side car lane change time
v1m=v1+X(1)*tm; % the speed of the ego car when the side car changes lanes
s1m=d1+v1*tm+0.5*X(1)*tm^2;  % When the side car changes lanes, the position
of the center of the ego car
s2m=d2+v2*X(4)+0.5*X(2)*X(4)^2+sx; % When the side car changes lanes, the
position of the center of the side car (x direction)
real_cutangle=atan(w*X(3)/4); %Yaw angle at the time of lane change (cut-in
angle)
```

```
de=v2m/2; %Safety distance at current speed

if ttc<-5 || ttc>0 % When the absolute value of the time difference between the two cars is greater than 5 seconds, the order of the initial position is the order of passage
    c(1)=sign(d1-d2)*(v2m-v1m); % The first constraint: the speed of the ego car is less than the speed of the side car at the time of lane change
    c(2)=de+0.5*(l1+l2*cos(real_cutangle))-sign(d1-d2)*(s1m-s2m); % The second constraint: keep a safe distance between the front (tail) of the ego car and the rear (head) of the side car at the time of lane change
else % When the absolute value of the time difference between the two vehicles is less than 5 seconds, the reverse order of the initial position is the passing order
    c(1)=sign(d1-d2)*(v1m-v2m); % The first constraint: the speed of the ego car is less than the speed of the side car at the time of lane change
    c(2)=de+0.5*(l1+l2*cos(real_cutangle))-sign(d1-d2)*(s2m-s1m); % The second constraint: keep a safe distance between the front (tail) of the ego car and the rear (head) of the side car at the time of lane change
end

v1e=v1+X(1)*(tm+tcm); % Speed of the ego car at the end of the cut
c(3)=0.5-v1e; %The third constraint: the speed at the end is greater than 0.5m/s
ceq=[]; % No equality constraints
```

Subfunction 3: ideal cut-in angle computation

```
%% Calculate the ideal cut-in angle function
function thetta=Cut_in_angle(X)
ww=1.8; % Half of lane width
a0=1; %Acceleration corresponding to ideal cut-in angle during cut-in process
thetta=acos(1-ww*a0./(X)^2);
% thetta=thetta*180/pi;
if thetta>pi/4
    thetta=pi/4;
end
```

Appendix II : Calibration codes

Brake calibration (initial velocity need to be 180kph)

```
x=0;
for i=1:81
    sim('calibration');
    v_temp1(:,i)=vx.data;
    a_temp1(:,i)=ax.data;
    brake_temp1(:,i)=ones(length(vx.data),1)*x;
    %to eliminate the singularity, no matter brake=1 or 2, will lead to the
v,a=0,caused multi-value
    for j=1:length(v_temp1(:,i))
        if v_temp1(j,i)<0.01
            brake_temp1(j,i)=0;
        end
    end
    x=x-0.1;
end
a_temp1(1,:)=a_temp1(2,:);

vbr=v_temp1(:,1)';
abr=a_temp1(:,1)';
br=brake_temp1(:,1)';
for i=2:length(v_temp1(1,:))
    vbr=[vbr,v_temp1(:,i)'];
    abr=[abr,a_temp1(:,i)'];
    br=[br,brake_temp1(:,i)'];
end
```

Throttle calibration: (initial velocity need to be 0kph)

```
x=0;
for i=1:21
    sim('calibration');
    v_temp(:,i)=vx.data;
    a_temp(:,i)=ax.data;
    thr_temp(:,i)=ones(length(vx.data),1)*x;
    x=x+0.05;
end

v=v_temp(:,1)';
 a=a_temp(:,1)';
 tr=thr_temp(:,1)';
for i=2:length(v_temp(1,:))
    v=[v,v_temp(:,i)'];
    a=[a,a_temp(:,i)'];
    tr=[tr,thr_temp(:,i)'];
end
```

Joint calibration of brake and throttle:

```
v2=[v,vbr];
a2=[a,abr];
br2=[tr,br];

F=scatteredInterpolant(v2',a2',br2');
vubr=0:0.05:50;
aubr=-8:0.05:5;
tablebr=zeros(length(vubr),length(aubr));
for i=1:length(vubr)
    for j=1:length(aubr)
        tablebr(i,j)=F(vubr(i),aubr(j));
    end
end
```

Appendix Ⅲ: Offline LQR codes

```
cf=-110000;
cr=cf;
m=1410;
Iz=1536.7;
a=1.015;
b=2.91-1.015;
k=zeros(5000,4);
for i=1:5000
    vx=0.01*i;
    A=[0,1,0,0;
        0,(cf+cr)/(m*vx),-(cf+cr)/m,(a*cf-b*cr)/(m*vx);
        0,0,0,1;
        0,(a*cf-b*cr)/(Iz*vx),-(a*cf-
b*cr)/Iz,(a*a*cf+b*b*cr)/(Iz*vx)];
    B=[0;
        -cf/m;
        0;
        -a*cf/Iz];
    Q=15*eye(4);
    R=10;
    k(i,:)=lqr(A,B,Q,R);
end
k1=k(:,1)';
k2=k(:,2)';
k3=k(:,3)';
k4=k(:,4)';
```

Bibliography

[1] P. Trautman and A. Krause, “Unfreezing the robot: Navigation in dense, interacting crowds,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (Taipei), pp. 797–803, IEEE, Oct. 2010.

[2] Simon Le Cleac’h, Mac Schwager and Zachary Manchester “LUCIDGames: onLine UnsCented Inverse Dynamic Games for Adaptive Trajectory Prediction and Planning” ,2020

[3] Simon Le Cleac’h, Mac Schwager and Zachary Manchester” ALGAMES: A Fast Solver for Constrained Dynamic Games”, *arXiv e-prints, Feb.2020*

[4] Taylor A. Howell, Brian E. Jackson, and Zachary Manchester “ALTRO: A Fast Solver for Constrained Trajectory Optimization”

[5] D. Q. Mayne, “A second-order gradient method of optimizing non- linear discrete time systems,” *Int J Control, vol. 3, p. 8595, 1966.*

[6] W. Li and E. Todorov, “Iterative Linear Quadratic Regulator Design for Nonlinear Biological Movement Systems,” in Proceedings of the *1st International Conference on Informatics in Control, Automation and Robotics, Setubal, Portugal, 2004.*

[7] H. B. Keller, “Numerical methods for two-point boundary-value problems”. *Courier Dover Publications, 2018.*

[8] Yimin Chen, Chuan Hu, and Junmin Wang, “Human-Centered Trajectory Tracking Control for Autonomous Vehicles With Driver Cut-In Behavior Prediction” in *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 68, NO. 9, Sep. 2019*

[9] S. Kim et al., “Analysis of human driver behavior in highway cut-in scenarios,” *SAE Tech. Paper, 2017-01-1402, 2017.*

[10] T. Taniguchi, S. Nagasaka, K. Hitomi, N. P. Chandrasiri, T. Bando, and K. Takenaka, “Sequence prediction of driving behavior using double articulation analyzer,” *IEEE Trans. Syst., Man, Cybern., Syst., vol. 46, no. 9, pp. 1300–1313, Sep. 2016.*

[11] J. Wiest, M. Karg, F. Kunz, S. Reuter, U. Kreßel, and K. Dietmayer, “A probabilistic maneuver prediction framework for self-learning vehicles with application to intersections,” *in Proc. IEEE Intell. Veh. Symp. (IV), pp. 349–355, 2015*

[12] J. Morton, T. A. Wheeler, and M. J. Kochenderfer, "Analysis of recurrent neural networks for probabilistic modeling of driver behavior," *IEEE Trans. Intell. Transp. Syst., vol. 18, no. 5, pp. 1289–1298, May 2017.*

[13] A. P. Aguiar and J. P. Hespanha, "Trajectory-tracking and path-following of underactuated autonomous vehicles with parametric modeling uncertainty," *IEEE Trans. Autom. Control, vol. 52, no. 8, pp. 1362–1379, Aug. 2007.*

[14] C. Hu, H. Jing, R. Wang, F. Yan, and M. Chadli, "Robust H outputfeedback control for path following of autonomous ground vehicles," *Mech. Syst. Signal Process., vol. 70, pp. 414–427, 2016.*

[15] P. Falcone, F. Borrelli, J. Asgari, H. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Trans. Control Syst. Technol., vol. 15, no. 3, pp. 566–580, May 2007.*

[16] F. Altché, P. Polack, and A. de La Fortelle, "High-speed trajectory planning for autonomous vehicles using a simple dynamic model," *in Proc. IEEE 20th Int. Conf. Intell. Transp. Syst., pp. 1–7, 2017.*

[17] Fridovich-Keil D, Ratner E, Peters L, et al. "Efficient Iterative Linear-Quadratic Approximations for Nonlinear Multi-Player General-Sum Differential Games" *International Conference on Robotics and Automation (ICRA). IEEE, 2020.*