Research on lateral control of autonomous vehicle based on driver steering model

Runqi Qiu\textsuperscript{1a*}, Shangfeng Xin\textsuperscript{1b*}

\textsuperscript{1}School of International Education, Wuhan University of Technology, Wuahn 430070, China
\textsuperscript{a}qiurunqi@whut.edu.cn, \textsuperscript{b}282436@whut.edu.cn

Abstract—Path tracking is an important issue in autonomous driving research. In this paper, based on the vehicle two-degree-of-freedom linear dynamics model for path tracking, the study of vehicle path tracking is carried out by applying model predictive control. A linear state space model of the vehicle is established based on the linear two-degree-of-freedom dynamics model, which provides the basis for the path tracking research. Based on the established state space model and the model predictive control method, the path tracking control algorithm is established using the preview point of driver steering model instead of the prediction part in the traditional model predictive control algorithm, which reduces the complexity of the algorithm and greatly improves the real-time performance. Finally, the effectiveness of the algorithm is verified by the joint software simulation of CarSim and Simulink.

1. Introduction

Automobiles have always been required to be maneuvered by humans. With the improvement of computer processing power, deep learning, an image recognition method with high recognition rate, began to be downlinked to the vehicle computers. At the same time, the development of LIDAR technology has allowed cars to perceive their surroundings more accurately and rapidly. Along with the vigorous development of the above technologies, autonomous driving has become a popular topic. However, the road conditions faced by passenger cars are very sophisticated and put forward high requirements for the reliability of the automatic driving system, so it is very meaningful to study the automatic driving of passenger cars. In this, the lateral control of the vehicle, i.e., the trajectory tracking strategy, is the key to whether the vehicle can follow the planned path form.

Model Predictive Control (MPC) is a control method that combines the mathematical model of the object to be controlled, so that the weights of each control quantity can be controlled more easily and with better control results. Model Predictive Control is based on a dynamic model, characterized by optimizing each time for the current time zone, and then optimizing the next time for the time zone, so that future events can be predicted and processed.

With the enhancement of vehicle computer capabilities and the development of control theory, MPC algorithms have begun to attract attention in vehicle dynamics control. The core idea of MPC is to establish a vehicle dynamics model, use this model to simulate the vehicle, and then can invert the solution to the vehicle control strategy. In order to adapt to the low performance of vehicle computers, the dynamics model used to control the vehicle generally uses a two-degree-of-freedom (2DOF) dynamics model, that is, the left and right wheels of one axis are equivalent to one wheel, which can significantly simplify the calculation and make real-time vehicle model predictive control possible [1].
Based on the 2DOF dynamic model of the vehicle, this paper proposes a path tracking control algorithm by combining MPC with the preview point method in the driver steering model. The working performance of the system is verified by the joint simulation of CarSim and Simulink.

2. Mathematic Model
The model predictive control algorithm needs to build a vehicle dynamics model and predict the future driving state of the vehicle based on the model, and back-calculate the predicted amount by the least squares method to obtain the control rate.

2.1. 2DOF vehicle dynamics model
According to the extant literature [2], on the basis of the two-degree-of-freedom vehicle dynamics model given in Figure 1, the motion of the vehicle in the absolute coordinate system can be described by the following differential equation:

\[
\begin{align*}
\ddot{m} \dot{y} &= -m \ddot{x} \phi + 2F_{yf} + F_{yr} \\
I_{x} \ddot{\phi} &= 2aF_{yf} - 2bF_{yr}
\end{align*}
\]

Fig.1 2DOF vehicle dynamics model

For \(a_y \leq 0.4 g\), the lateral force of the tire satisfies:

\[
\begin{align*}
F_{yf} &= C_f \alpha_f \\
F_{yr} &= C_r \alpha_r
\end{align*}
\]

Under the minor rotation angle assumption \((\sin \delta \approx 0, \cos \delta \approx 1)\), we get:

\[
\begin{align*}
\alpha_r &= \frac{v_y - \dot{\phi} b}{v_x} - \delta_r \\
\alpha_f &= \frac{v_y + \dot{\phi} a}{v_x} - \delta_f
\end{align*}
\]

Substituting (5) and (6), as well as (3) and (4) into the linear two-degree-of-freedom vehicle dynamics model and simplifying it, and converting the model into a state-space expression, we have:

\[
\begin{bmatrix}
\dot{v}_y(t) \\
\dot{\omega}(t)
\end{bmatrix} =
\begin{bmatrix}
-\frac{2C_f + 2C_r}{v_x m} & -\frac{2aC_f - 2bC_r}{v_x m} & -v_x \\
\frac{2aC_f - 2bC_r}{v_x I_x} & \frac{2a^2C_f + 2b^2C_r}{v_x I_x}
\end{bmatrix}
\begin{bmatrix}
v_y(t) \\
\omega(t)
\end{bmatrix} +
\begin{bmatrix}
\frac{2C_f}{m} & \frac{2C_r}{m} \\
\frac{2aC_f}{I} & \frac{2bC_r}{I}
\end{bmatrix}
\begin{bmatrix}
\delta_f(t) \\
\delta_r(t)
\end{bmatrix}
\]

As shown in Figure 2, so that the absolute coordinate system coincides with the vehicle coordinate system, it can be assumed that the vehicle has a small angular displacement \(\phi\) in the future period, at which time there is:

\[
\begin{align*}
\dot{x}(t) &= U - v(t)\phi(t) \\
\dot{y}(t) &= U\phi(t) + v(t)
\end{align*}
\]
By writing (8) and (9) in the form of state space equations and connecting them with (7), the following expression can be obtained:

\[
\begin{bmatrix}
v_y(t) \\
v_\omega(t) \\
y(t) \\
\phi(t)
\end{bmatrix} \quad \begin{bmatrix}
-\frac{2C_f + 2C_r}{v_\omega} - \frac{2aC_f - 2bc_r}{v_xm} - v_x & 0 & 0 \\
-\frac{2aC_f - 2bc_r}{v_xm} & \frac{2aC_f + 2b^2C_r}{v_\omega} & 0 & 0 \\
\frac{v_\omega I_z}{1} & \frac{v_x I_z}{1} & 0 & U \\
0 & 0 & 0 & 0
\end{bmatrix} \begin{bmatrix}
v_y(t) \\
v_\omega(t) \\
y(t) \\
\phi(t)
\end{bmatrix} + \begin{bmatrix}
\frac{2C_f}{m} & \frac{2C_r}{m} & 0 & 0 \\
\frac{2aC_f}{m} & \frac{2bC_r}{m} & 0 & 0 \\
0 & 0 & \frac{1}{I} & 0 \\
0 & 0 & 0 & \frac{1}{I}
\end{bmatrix} \begin{bmatrix}
\delta_f(t) \\
\delta_r(t)
\end{bmatrix}
\]

Taking \([v_y(t) \quad \omega(t) \quad y(t) \quad \phi(t)]^T\) as the state quantity \(x(t)\), the front wheel rotation angle \(\delta(t)\) as the control quantity \(u(t)\), and choosing the angular displacement \(\phi(t)\) and the longitudinal displacement \(y(t)\) as the output quantity \(z(t)\), a simplified state space equation for the car can be written:

\[
x(t) = Ax(t) + Bu(t)
\]

\[
z(t) = Cx(t)
\]

Among them,

\[
A = \begin{bmatrix}
-\frac{2C_f + 2C_r}{v_\omega} - \frac{2aC_f - 2bc_r}{v_xm} - v_x & 0 & 0 \\
-\frac{2aC_f - 2bc_r}{v_xm} & \frac{2aC_f + 2b^2C_r}{v_\omega} & 0 & 0 \\
\frac{v_\omega I_z}{1} & \frac{v_x I_z}{1} & 0 & U \\
0 & 0 & 0 & 0
\end{bmatrix}
\]

\[
B = \begin{bmatrix}
\frac{2C_f}{m} & \frac{2C_r}{m} \\
\frac{2aC_f}{m} & \frac{2bC_r}{m} \\
0 & 0 \\
0 & 0
\end{bmatrix}
\]

\[
C = \begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

### 2.2. Model Predictive Control

To facilitate computer processing, the state space model needs to be discretized:

\[
x(k + 1) = Ax(k) + Bu(k)
\]

\[
z(k) = Cx(k)
\]

According to literature [3], the vehicle status in the prediction domain \(N_p\) can be calculated:
\[
\begin{bmatrix}
x(k + 1) \\
x(k + 2) \\
\vdots \\
x(k + N_p)
\end{bmatrix}
= \begin{bmatrix} A & A^2 & \cdots & A^{N_u} \\
A^2 & \cdots & \cdots & \cdots \\
\vdots & \cdots & \cdots & \cdots \\
A^{N_u} & \cdots & \cdots & \cdots \\
\end{bmatrix}
\begin{bmatrix} x(k) \\
x(k+1) \\
\vdots \\
x(k+N_p)
\end{bmatrix}
+ \begin{bmatrix} B & 0 & \cdots & 0 \\
AB & B & 0 & \cdots & 0 \\
\vdots & \cdots & \cdots & \cdots \\
A^{N_u-1}B & \cdots & AB & B \\
\end{bmatrix}
\begin{bmatrix} u(k) \\
u(k+1) \\
\vdots \\
u(k+N_p-1)
\end{bmatrix}
\tag{18}
\]

To simplify the calculation, it can be considered that the control quantity \( \delta(k+N_u-1) \) does not change after the control domain \( N_u \). Substituting the expression for the state quantity \( x(k) \) in (18), we get:

\[
\begin{bmatrix}
x(k + 1) \\
x(k + 2) \\
\vdots \\
x(k + N_p)
\end{bmatrix}
= \begin{bmatrix} C & 0 & \cdots & 0 \\
0 & C & \cdots & \cdots \\
\vdots & \cdots & \cdots & \cdots \\
\vdots & \cdots & \cdots & \cdots \\
\vdots & \cdots & \cdots & \cdots \\
\end{bmatrix}
\begin{bmatrix} x(k) \\
x(k+1) \\
\vdots \\
x(k+N_p)
\end{bmatrix}
+ \begin{bmatrix} u(k) \\
u(k+1) \\
\vdots \\
u(k+N_p-1)
\end{bmatrix}
\tag{19}
\]

The output quantity \( z(k) \) at this point can be abbreviated in the form of (22):

\[
z(k) = \Psi x(k) + \Theta u(k)
\tag{22}
\]

In order to make the output as close as possible to the input, so the loss function \( v(k) \) is introduced:

\[
v(k) = \| z(k) - r(k) \|^2_Q + \| u(k) \|^2_R
\tag{23}
\]

Where \( r(k) \) is the reference value of the output quantity, i.e., the steering wheel angle corresponding to the path to be tracked by the unmanned vehicle, \( Q \) is the weight of the output quantity, satisfying \( S_Q^T S_Q = Q \), \( R \) is the weight of the control quantity, satisfying \( S_R^T S_R = R \).

The error between the true value and the reference value is:

\[
e(k) = r(k) - \Psi x(k)
\tag{24}
\]

To minimize the error in the control objective, it is necessary to make \( v(k) = 0 \). Also, to simplify the computation, the error is considered to be linear with respect to the computed input, from which the control quantity \( u_{\text{optimal}} \) can be solved for all moments in the control domain:

\[
u_{\text{optimal}}(k) = K_{\text{full}} e(k)
\tag{25}
\]

In turn, we can obtain:

\[
K_{\text{full}} = \begin{bmatrix} S_Q \Theta & S_Q \\
S_R & 0 \end{bmatrix}
\tag{26}
\]

Thus, by taking the first row of the \( K_{\text{full}} \) matrix \( K_\omega = K_{\text{full}} (1,:) \), we can obtain the front wheel rotation angle that minimizes the trajectory tracking error at the current moment:

\[
u(k) = \delta(k) = [-K_\omega \Psi K_\omega^{-1}] [x(k)] T(k)
\tag{27}
\]

Among them,

\[
\begin{bmatrix} x(k) \\
T(k) \end{bmatrix} = [v_y \ \omega \ \gamma \ \varphi \ \gamma_1 \ \varphi_1 \ \cdots \ \gamma_{N_p} \ \varphi_{N_p}]^T
\tag{28}
\]
In the driver steering model as shown in Figure 3, \( N_p \) preview points can be set at an interval \( d \) meters perpendicular to the front axis, and the distance between the preview points and the actual track point \( [y_1 \ y_2 \ \cdots \ y_{N_p}]^T \) can be calculated [4].

However, the driver steering model cannot output the difference between the heading angle at the preview point and the tangent slope at the actual trajectory point \( [\varphi_1 \ \varphi_2 \ \cdots \ \varphi_{N_p}] \), so it needs to be estimated based on known information:

\[
\varphi_i \approx \frac{y_{i+1} - y_i}{d} \tag{29}
\]

The converted matrix is:

\[
\begin{bmatrix}
\varphi_1 \\
\varphi_2 \\
\vdots \\
\varphi_{N_p}
\end{bmatrix} = \begin{bmatrix}
y_1 \\
\frac{y_1}{d} \\
y_2 \\
\frac{y_2}{d} \\
\vdots \\
\frac{y_{N_p}}{d}
\end{bmatrix} \begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_{N_p}
\end{bmatrix} \tag{30}
\]

Make:

\[
M = \begin{bmatrix}
1 & 0 & 0 & 0 \\
\frac{1}{d} & 1 & 0 & 0 \\
0 & \frac{1}{d} & 1 & 0 \\
\vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \frac{1}{d}
\end{bmatrix}, \quad Y_{\text{preview}} = \begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_{N_p}
\end{bmatrix} \tag{31}
\]

Then we have:

\[
u(k) = \delta(k) = -[K_\omega \ \Psi \ K_\omega] \begin{bmatrix}
x(k) \\
M Y_{\text{preview}}(k)
\end{bmatrix} \tag{32}
\]

Since the performance parameters of the car are constants, the matrix \( [-K_\omega \ \Psi \ K_\omega] \) is constant as long as the model remains unchanged, at which point the steering wheel angle input of the vehicle is simplified to a linear expression.

3. Simulation

As shown in Figure 4, the preview point can be set in CarSim, and the length of the vertical line from the preview point to the target trajectory can be output. Considering the assumption that the absolute
coordinate system overlaps with the vehicle coordinate system has been made previously, the longitudinal displacement \([y_1, y_2, \cdots, y_{N_p}]^T\) of the reference path can be directly replaced by \(L_{DRV_i}\).

Since a certain longitudinal speed is assumed in the two-degree-of-freedom dynamics model of the vehicle, the brakes are set in CarSim to the open-loop (i.e., independent of external influences) and always brakeless case, and the gear control is in closed-loop mode, which is handled automatically by the software.

According to the model established in Section 2, the only control quantity of the vehicle during trajectory tracking, i.e., the only input quantity, is the steering wheel angle (which can also be the front wheel angle), and the output quantity corresponds to the information contained in the matrix \([x(k), MY_{preview}(k)]^T\). The input interface of CarSim is IMP_STEER_SW, and the output interfaces are \(V_y, AV_z, Y_0, Y_\text{yaw}, L_{DRV_1}, L_{DRV_2}, L_{DRV_3}, L_{DRV_4}, L_{DRV_5}\). Considering that CarSim uses the angle system, the angle quantities need to be converted to the radian system for calculation.

Referring to literature [5], Simulink and CarSim are used for joint simulation to establish the simulation model as shown in Figure 5:

---

**Fig.5 CarSim preview point setting**

**Fig.5 Joint Simulation Model**
The simulation time is set to 45 seconds, and the path tracking results of the vehicle at 90 km/h high speed are shown in Figure 6. It can be seen that the proposed control strategy can obtain a better path tracking effect.

![Fig.6 Path Tracking Results](image)

From Figure 7, the curves of vehicle transverse velocity, lateral velocity and transverse velocity as a function of time can be seen that the vehicle is always in a safer driving condition, and the great and small values occurring in the simulation cycle are within the acceptable range.

![Fig.7 Distribution of measuring points of specimens](image)

4. Conclusion

Based on the vehicle two-degree-of-freedom linear dynamics model and driver steering model, a model predictive control algorithm is designed, setting the absolute coordinate system to coincide with the vehicle coordinate system and using the preview points provided by CarSim instead of the predicted values calculated by the dynamics model, while ignoring the steering wheel turning angle constraint when solving the steering wheel turning angle input using the least squares method. By simplifying the control model through the above steps, the real-time performance of vehicle path tracking is greatly improved, and the proposed method obtains better path tracking results.
The shortcomings of the algorithm are that it only takes into account the current tire characteristics, and when the vehicle load changes and the number of occupants changes, the control rate needs to be recalculated to obtain better tracking, and the above theory lacks hardware-in-the-loop simulation and validation by real vehicle testing. In the next step of the study, the use of nonlinear tires will be considered and an attempt will be made to improve the applicability of the algorithm.

References
[1] Xing, S.Q., Feng, K., Xue, L.L., (2021) Research on Lateral Control Algorithm of Autonomous Driving Based on Model Prediction, Automotive Practical Technology, 46(15): 45-47.
[2] Yu, Z.S. (2009) Automobile Theory. China Machine Press, Beijing.
[3] Cole, D.J., Pick, A.J., (2006) Predictive and linear quadratic methods for potential application to modelling driver steering control, Vehicle System Dynamics, 44(3): 259-284.
[4] Shen, H., Tan, Y.S., Jin, Z.L., Li, S.Y., (2013) Research on driver steering model considering rear wheel active steering, Chinese Journal of Scientific Instrument, 34(12): 2736-2742.
[5] Math Works. (2017) Autonomous Vehicle Steering Using Model Predictive Control. https://ww2.mathworks.cn/help/releases/R2017b/mpc/examples/autonomous-vehicle-steering-using-model-predictive-control.html