Model predictive approach to integrated path planning and tracking for autonomous vehicles

Chao Huang\(^1\), Boyuan Li\(^2\) and Masako Kishida\(^1\)

Abstract—In the field of path planning for autonomous vehicle, the existing studies separately consider the path planning and path tracking problem. To fill in this research gap, this study proposes an integrated path planning and path tracking control method. In addition, this paper studies the collision avoidance problem of autonomous vehicles by considering static and dynamic obstacles. Simulation results show that the proposed method can generate a collision-free path and control the autonomous vehicle to avoid the obstacles simultaneously.

I. Introduction

Autonomous vehicles have a promising future of enabling drivers to partake in other activities and therefore change everyday life across the world. They have the potential to dramatically reduce crashes which are caused by driver’s faults including aggressive driving, slow reaction time and improper driving skill. In addition, autonomous vehicles can increase road capacity, save fuel consumption and decrease harmful emissions [1].

Experimental tests have been conducted on autonomous vehicles since 1920s. Many large global companies such as Google, Tesla, BMW and Lexus are investing fund and resources in the research of autonomous vehicles. Google’s co-founder Sergey Brin at Recode Code Conference in Palos Verdes, California described the concept of autonomous vehicles [2]. The prototype of autonomous vehicles relies on sensor technologies and software systems to safely operate the vehicle’s motion without manual controls.

One of several key problems to be solved before autonomous vehicle becoming a reality is to design a collision-free trajectory of autonomous vehicles based on the information of sensing and mapping module. In general, the autonomous vehicle control is divided into path planning and trajectory tracking. The path planning problem is to find a desired path for the vehicle to reach the goal region while satisfying given global and local constraints [3]. Meanwhile, the path tracking problem is to design control laws for the autonomous vehicle to follow the selected path.

The real-time path planning, however, particularly in the presence of obstacles, remains very challenging [4]. There exist many different approaches for solving the path-planning problem. The graph-search approaches such as Dijkstra algorithm [5], [6], A* algorithm [7], and State Lattice algorithm [8], construct graphical discretization of the vehicle’s state space to search for the shortest path. The main principle of incremental tree-based approaches is to construct a tree of reachable states and select the optimal branch to contribute to a best path. The sampling based approaches have been proven to be an effective framework that is suitable for a large class of problems. Existing methods in robotics such as Probabilistic Roadmap Method (PRM) [9] are extensively tested for automated vehicles. However, these methods are computationally expensive [11], [12].

Regarding the path tracking controller, many valuable outcomes are presented e.g. [13], [14]. However, many studies assume that the desired path for vehicle to follow is given or obtained by an off-line path planner. Then, the main purpose of the tracking controller is to control the vehicle to follow the given path, which is assumed to be achieved by the autonomous vehicle. However, this assumption ignores the vehicle’s dynamics and driving feasibility. On the other hand, many tracking controllers design are based on linearized bicycle models. The oversimplified model cannot present the vehicle’s nonlinear and complex dynamics, and therefore the corresponding tracking controller may make the vehicle operate around the limits of its handling capability [3].

Most existing studies separately consider the path planning and tracking control of autonomous vehicles and only a few of them have integrated the path planning and tracking control together [4]. The reference [15] shows that the integrated control design can simplify the control structure and further improve the computational efficiency. We design a control method that combines the path planning and tracking by using a nonlinear vehicle Model Predictive (MP) algorithm in this paper. This integrated MP algorithm simultaneously optimises the front wheel angle and brake torque directly and predicts the local path in real time. The additional tracking controller is no longer required, and the algorithm can be easily implemented in real-time path planning and tracking. Moreover, instead of assuming the longitudinal velocity of autonomous vehicle is constant, we utilise a 2-degree-of-freedom (2-DOF) nonlinear model to describe the vehicle’s dynamic. First, we calculate a Dubins path by considering the acceleration limits of the vehicle. Compared with road centre line, the Dubins path is proven to be shortest and flexible. Then, an optimisation cost function of the proposed MP algorithm is designed to minimise the distance from the Dubin path, maximize the distance from the road boundary, and improve the smoothness and comfort of the path by

\(1\)

\(2\)

\(3\)

\(4\)

\(5\)

\(6\)

\(7\)

\(8\)

\(9\)

\(10\)

\(11\)

\(12\)

\(13\)

\(14\)

\(15\)
minimizing the yaw acceleration of the autonomous vehicle throughout the whole manoeuvre. In addition, both static and dynamic obstacles on the road are considered in the lane change manoeuvre. The simulation results show the proposed MP algorithm can successfully generate a collision-free path and control the vehicle to avoid obstacles simultaneously.

The remainder of the paper is structured as follows. The dynamics model of the autonomous vehicle is provided in Section II. The driving manoeuvre and Dubins path are designed in Section II-A and the MP-based integrated path planning and tracking algorithm is presented in Section III-A. Two sets of simulation results are demonstrated in Section IV. Finally, Section V concludes this paper.

II. VEHICLE DYNAMICS MODEL

The use of autonomous vehicles brings with a number of advantages compared to human operated vehicles [16], and autonomous vehicles have already been developed and produced for production, construction, and urban environments.

The dynamics of the vehicle is modeled by a bicycle model as shown in Fig. 1 [17]:

\[
\begin{align*}
\dot{x} &= v_x = \frac{2}{m} \left( F_{cf} \sin \delta_f - T_r / R_w \right) \\
\dot{y} &= -r v_y + \frac{2}{m} \left( F_{cf} \cos \delta_f + F_{cr} \right) \\
\dot{r} &= \frac{2}{I_z} \left( l_f F_{cf} - l_r F_{cr} \right) \\
\dot{\psi} &= v_x \cos \psi - v_y \sin \psi \\
\dot{\psi} &= v_x \sin \psi + v_y \cos \psi
\end{align*}
\]

where \( x \) and \( y \) are the coordinates of the centre of mass (CM) in an inertial frame. \( v_x = \dot{x} \) and \( v_y = \dot{y} \) are the longitudinal and lateral speeds of CM. \( X \) and \( Y \) denote the longitudinal and lateral speeds of CM in the body frame \((X, Y)\), respectively. \( \Psi \) is the inertial heading angle and \( r = \dot{\psi} \) is the yaw rate. \( m \) denotes the vehicle’s mass and \( I_z \) is the yaw inertia. \( l_f \) and \( l_r \) represent the distance from CM of the vehicle to the front and rear axles, respectively. \( F_{cf} \) and \( F_{cr} \) denote the lateral tyre forces at the front and rear wheels, respectively. It is assumed that these lateral tyre forces are expressed by

\[
F_{ci} = -C_{ai} \alpha_i, \quad i \in \{f, r\}
\]

where \( C_{ai}, (i = f, r) \) are the front and rear tyre cornering stiffness, and \( \alpha_i, (i = f, r) \) are the front and rear slip angles which can be given by

\[
\begin{align*}
\alpha_f &= \frac{v_y + l_f r}{v_x} - \delta_f \\
\alpha_r &= \frac{v_y - l_r r}{v_x}
\end{align*}
\]

with a small angle assumption.

1) Control constraints: In this paper, we consider the constraints on the front wheel steering angle and brake torque. In general, the constraints on the traditional front wheel angle is between \(+35^\circ\) and \(-35^\circ\) [18] and between \(-45^\circ\) and \(45^\circ\) in Steer-by-Wire (SBW) systems. Moreover, the front wheel angle can achieve a rotation of \(\pm 90^\circ\) in four-wheel-independent-steering system. In this paper, we assume that the front wheel angle can lie only within \(-45^\circ\) and \(45^\circ\), In addition, the total braking torque \(T_{\text{brake}}\) is \(160N\)m and the maximum driving torque of the real wheel \(T_{\text{max}}\) is \(200N\)m. It is noted that any signal outside of these ranges is invalid.

We design a single lane change manoeuvre to test the vehicle handling ability. Lane change is one of the key challenging driving tasks and poor lane change maneuver may cause severe traffic accident on the motorway. The road information including road width, road boundary and the obstacles information including surrounding vehicles’ velocity are assumed to be known from the ego vehicle’s sensors.

A. Dubins path

One of the most common path planning methods found in robotics is Dubins path provides a minimum-distance path between source and destination[19]. This section describes the construction of a Dubins path that is suitable for the ego vehicle to perform collision-free manoeuvre [20]. Given a maximum acceleration of vehicle \(\mu g \ [m/s^2]\) [21], then, a circular path with radius \(R\) is obtained as

\[
R = \frac{X^2}{(\mu g)}
\]

where \( X \) is the longitudinal speed of the vehicle, \( \mu \) is the friction coefficient. \( g \) is the gravity constant.

III. DRIVING MANOEUVRE

Fig. 2 shows the construction of a Dubins path which involves straight lines and circular arcs in a single lane change manoeuvre. The road width is \(3.5m\) and its boundaries are represented by black solid lines. There are three obstacles
\[
J = \min_{\delta f(k) \cdots \delta f(k+N_p-1), T_r(k) \cdots T_r(k+N_p-1)} \sum_{i=1}^{N_p} \left\{ a_1 d(\hat{x}_a, X_d, \hat{y}_a, Y_d) + b_1 \left[ \frac{1}{d(\hat{x}_a, X_d, \hat{y}_a, Y_d)} \right]^2 + b_2 \left[ \frac{1}{d(\hat{x}_a, X_i^2, \hat{y}_a, Y_i)} \right]^2 + b_3 \left[ \frac{d\hat{r}(k+i)}{dt} \right]_{t=\varepsilon}^2 \right\}
\]

s.t.

\[-\delta_{max} \leq \delta_f(k+i) \leq \delta_{max} \quad \text{for all } i = 1 \cdots N_p.
\]

the premise of the information of obstacles, the path can feasibly avoid static and dynamic obstacles in the lane change manoeuvre. More details can be found in [21].

A. MP optimisation

However, the transition between arc and line segments entails discontinuous changes in lateral acceleration, making it impossible for real autonomous vehicle to track exactly. In this paper, MP algorithm is used to obtain control inputs \(\delta_f\) and \(T_r\) to minimize the error.

In this paper, the cost function of MP optimisation algorithm at time \(k\) can be presented in (5). In (5), \(d(x, y, Y, Y_g) = (x(k+1) - X(k+i))^2 + (y(k+i) - Y(k+i))^2\), \(N_p\) is the prediction horizon, \(X_d\) and \(Y_d\) are the longitudinal and lateral positions of the Dubins path which are generated in Section II-A. \(X_i\) \((i = u, l)\) and \(Y_i\) \((i = u, l)\) are the longitudinal and lateral positions of road upper boundary and lower boundary, respectively. \(\hat{x}_a\) and \(\hat{y}_a\) are the predicted vehicle position (longitudinal and lateral) optimised by the integrated MP algorithm. \(a_1\), \(b_1\), \(b_2\) and \(b_3\) are the weighting factors and are given according to the priority of difference scenarios. Specifically, \(a_1\) is the weighting factor of minimizing the distance between the ego vehicle position and Dubins path. \(b_1\) and \(b_2\) are the weighting factors of maximizing the distance between the ego vehicle and road boundaries. \(b_3\) is related to the term of minimizing yaw acceleration throughout the manoeuvre. The optimisation variables are front wheel steering angle \(\delta_f(k) \cdots \delta_f(k+N_p-1)\) and brake torque \(T_r(k) \cdots T_r(k+N_p-1)\).

The \(\hat{x}_a\) and \(\hat{y}_a\) are the future vehicle status which can be predicted in the previous time step based on the following discrete dynamics model:

\[
\begin{align*}
\hat{x}_a(t) &= x_g(t-1) + \hat{\delta}_\alpha(t) \Delta t, \\
\hat{y}_a(t) &= y_g(u-1) + \hat{\delta}_\gamma(t) \Delta t
\end{align*}
\]

with \(t = k + 1 \cdots k + N_p\).

where \(x_g(u-1)\) is the vehicle longitudinal position in the previous step in the global coordinate system and \(y_g(t-1)\) is the vehicle lateral position in the previous time step in the global coordinate system. Here, \(\Delta t = t(t) - t(t-1)\) where \(t(t-1)\) and \(t(t)\) are the time of the current step and next step, respectively. \(\hat{\delta}_\alpha(t)\) and \(\hat{\delta}_\gamma(t)\) are the predicted vehicle
\[ \begin{align*}
\dot{v}_x(t) &= v_x(t-1) + \left( v_y(t-1) r(t+i-1) - 2 \hat{F}_{c_f}(t-1) \sin \delta_f(t-1) - \frac{T_r(t-1)}{m} \right) \Delta t, \\
\dot{v}_y(t) &= v_y(t-1) + \left( -v_x(t-1) r(t-1) + 2 \hat{F}_{c_r}(t-1) \cos \delta_f(t-1) + \hat{T}_{c_r}(t-1) \right) \Delta t, \\
\dot{\psi}(t) &= \psi(t-1) + r(t-1) \Delta t, \quad \text{with } t = k + 1, \ldots, k + N_p
\end{align*} \]

longitudinal and lateral velocities in the current time step. The \( \hat{v}_{xy}(t) \) and \( \hat{v}_{yy}(t) \) can be presented as follows:

\[
\begin{bmatrix}
\hat{v}_{xy}(t) \\
\hat{v}_{yy}(t)
\end{bmatrix} =
\begin{bmatrix}
\cos(\hat{\psi}(t)) & -\sin(\hat{\psi}(t)) \\
\sin(\hat{\psi}(t)) & \cos(\hat{\psi}(t))
\end{bmatrix}
\begin{bmatrix}
\hat{v}_x(t) \\
\hat{v}_y(t)
\end{bmatrix}
\] 

(7)

where \( \hat{\psi}(t) \) is the predicted vehicle yaw angle in the body-fixed coordinate system and \( \hat{v}_x(t) \) and \( \hat{v}_y(t) \) are predicted longitudinal and lateral velocities. According to (1), \( \hat{v}_x(t) \), \( \hat{v}_y(t) \) and \( \hat{r}(t) \) can be calculated in (8).

In (8), \( v_x(t-1) \) is the vehicle longitudinal velocity in the previous time step. \( v_y(t-1) \) is the lateral velocity in the previous time step and \( r(t-1) \) is the yaw rate in the previous time step. \( \hat{F}_{c_f}(t-1) \) is the predicted tyre front wheel side force in the previous time step and \( \hat{F}_{c_r}(t-1) \) is the predicted tyre rear wheel side force in the previous time step. Based on (2), \( \hat{F}_{c_f}(t-1) \) and \( \hat{F}_{c_r}(t-1) \) can be calculated as follows:

\[
\begin{align*}
\hat{F}_{c_f}(t-1) &= -C_{af} \left( \frac{v_x(t-1) + r(t-1) v_y(t-1)}{v_x(t-1)^2} - \delta_f(t-1) \right), \\
\hat{F}_{c_r}(t-1) &= -C_{ar} \left( \frac{v_y(t-1) - r(t-1) v_x(t-1)}{v_x(t-1)} \right),
\end{align*}
\]

with \( t = k + 1, \ldots, k + N_p \).

Overall, the real-time optimal path can be optimised by solving (5) through (6) - (9). The corresponding control actions i.e., front wheel angle and brake torque can be calculated and implemented into the autonomous vehicle. This method is summarized in Algorithm 1.

**Algorithm 1** The proposed algorithm

1. Set prediction horizon \( N_p \)
2. for \( t = l_{initial} \) to \( l_{final} \) do
3. Generate Dubins path based on Section II-A;
4. Solve optimisation problem (5);
5. Go back to 3 with \( t = k + 1 \).
6. end for

**IV. Simulation results**

In this section, two sets of simulations are presented to demonstrate the effectiveness of the integrated path planning and tracking controller. The simulation environment is set up as follows:

1) The simulations are carried by MATLAB;
2) The standard Matlab function \textit{fmincon} with receding control strategy is implemented to solve (5) with the sampling time 0.1s;
3) The prediction horizon is \( N_p = 3 \), \( a_1 = 1000 \), \( b_1 = 10 \), \( b_2 = 10 \) and \( b_3 = 0.1 \);
4) We assume that the ego vehicle can estimate the velocities of surrounding vehicles and then predict the future movements of the surrounding vehicles in the lane change manoeuvre;

5) The vehicle parameters are listed in TABLE I;

| Parameter  | Value   |
|------------|---------|
| $I_g$ (m)  | 1.2     |
| $I_z$ (m)  | 1.05    |
| $m(kg)$    | 2000    |
| $I_e(kg \cdot m^2)$ | 1300 |
| $\mu$      | 0.5     |
| $C_{of} (N/rad)$ | 12000 |
| $C_{ar} (N/rad)$ | 12000 |

Three algorithms for fmincon: interior point, sequential quadratic programming (SQP) and active set algorithms are compared with the traditional two-level method in Fig. 3, Fig. 4 and Fig. 5. It is assumed that the initial velocity of the autonomous vehicle is 10 m/s and all the obstacles are static. Compared with the traditional two-level method, the MP trajectories can provide a smooth and flexible path. We select the active set algorithm (red dash line) in the following simulations.

Fig. 6 and Fig. 7 show the first set of simulation where the obstacles are static. The initial velocity of the autonomous vehicle is set as $v_{x0} = 10 m/s$. Fig. 6 shows the Dubins path (black solid), actual path which is generated and controlled by the integrated MP algorithm (red dash) and the path which is based on the traditional two-level method. In the traditional two-level method, the desired path is assumed to be given (Dubins path) and the two-level path tracking controller is designed to follow the path. The simulation results show that both two methods can accurately follow the Dubins path, and the proposed MP method can provide a smoother lane change path.

Fig. 7 shows the vehicle yaw rate response of these methods in which the yaw rate of traditional two-level method changes drastically and rapidly. In contrast, the proposed MP methods obtain more stable yaw rate response and therefore improve the stability of vehicle’s lateral motion during the lane change.

Fig. 8, Fig. 9 and Fig. 10 show the results of the second set of simulations in which the obstacles are moving. The initial velocity of all these three vehicles are 10 m/s. The obstacle 1 will accelerate and drive at 10.5 m/s, obstacle 2 will accelerate and drive at 12 m/s, and obstacle 3 will accelerate and drive at 11.5 m/s. All the parameters of the ego vehicle are the same as that in the first set of simulation. Fig. 8 shows the actual vehicle path of the proposed integrated MP method and traditional two-level methods. The control inputs including front wheel angles and brake torque are shown in Fig. 9. Fig. 10 shows the yaw rate response in the second set of simulation with dynamic obstacles. It is indicated that the proposed MP algorithm can successfully generate a collision-free and smooth lane change path.

V. Conclusion

This paper proposes a real-time control method that integrates path planning and tracking based on the MP algorithm for autonomous vehicles. The major contributions can be summarised as:

1) The path planning and tracking control can be successfully integrated in the MP optimisation algorithm without the application of the path tracking controller,
2) Compared with the two-level method, the proposed MP algorithm can significantly improve the system stability and tracking performance during the lane change.

3) The proposed MP algorithm can successfully avoid the static and dynamic obstacles and generate a smooth obstacle avoidance path.

In the future, we are interested in extending the approach for multiple autonomous vehicles.

REFERENCES

[1] D. J. Fagnant and K. Kockelman, “Preparing a nation for autonomous vehicles: 1 opportunities, barriers and policy recommendations for 2 capitalizing on self-driven vehicles 3,” Transportation Research, vol. 20, 2014.

[2] P. Griffiths and R. B. Gillespie, “Shared control between human and machine: Haptic display of automation during manual control of vehicle heading,” in 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS’04). IEEE, 2004, pp. 358–366.

[3] B. Paden, M. Cáp, 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.

[4] C. Shen, Y. Shi, and B. Buckham, “Integrated path planning and tracking control of an AUV: A unified receding horizon optimization approach,” IEEE/ASME Transactions on Mechatronics, vol. 22, no. 3, pp. 1163–1173, 2017.

[5] R. Kala and K. Warwick, “Multi-level planning for semi-autonomous vehicles in traffic scenarios based on separation maximization,” Journal of Intelligent & Robotic Systems, vol. 72, no. 3-4, pp. 559–590, 2013.

[6] J. Bohren, T. Foote, J. Keller, A. Kushleyev, D. Lee, A. Stewart, P. Vernaza, J. Derencik, J. Spletzer, and B. Satterfield, “Little ben: The ben franklin racing team’s entry in the 2007 darpa urban challenge,” in The DARPA Urban Challenge. Springer, 2009, pp. 231–255.

[7] J. Ziegler, M. Werling, and J. Schroder, “Navigating car-like robots in unstructured environments using an obstacle sensitive cost function,” in 2008 IEEE Intelligent Vehicles Symposium. IEEE, pp. 787–791.

[8] M. Rufli and R. Siegwart, “On the application of the d* search algorithm to time-based planning on lattice graphs,” in 4th European Conference on Mobile Robotics (ECMR). Eidgenössische Technische Hochschule Zürich, 2009, pp. 105–110.

[9] L. Kavraki, P. Svestka, and M. H. Overmars, “Probabilistic roadmaps for path planning in high-dimensional configuration spaces,” IEEE Transactions on Robotics and Automation, vol. 1994, p. 566–580.

[10] S. M. LaValle and J. J. Kuffner Jr, “Randomized kinodynamic planning,” The international journal of robotics research, vol. 20, no. 5, pp. 378–400, 2001.

[11] J. Horst and A. Barbera, “Trajectory generation for an on-road autonomous vehicle,” in Unmanned Systems Technology VIII, vol. 6230. International Society for Optics and Photonics, 2006, p. 62302J.

[12] M. Brezak and I. Petrovic, “Real-time approximation of clothoids with bounded error for path planning applications,” IEEE Transactions on Robotics, vol. 30, no. 2, pp. 507–515, 2014.

[13] J. Yang, R. Ma, Y. Zhang, and C. Zhao, “Sliding mode control for trajectory tracking of intelligent vehicle,” Physics Procedia, vol. 33, pp. 1160–1167, 2012.

[14] T. Sangyam, P. Laohapiengsak, W. Chongcharoen, and I. Nilkhamhang, “Path tracking of uav using self-tuning pid controller based on fuzzy logic,” in Proceedings of SICE annual conference 2010. IEEE, 2010, pp. 1265–1269.

[15] B. Li, H. Du, W. Li, and B. Zhang, “Dynamically integrated spatiotemporal-based trajectory planning and control for autonomous vehicles,” IET Intelligent Transport Systems, vol. 12, no. 10, pp. 1271–1282, 2018.

[16] R. Pepy, A. Lambert, and H. Mounier, “Path planning using a dynamic vehicle model,” in Information and Communication Technologies, ICITA’06, 2nd, vol. 1. IEEE, pp. 781–786.

[17] R. Rajamani, Vehicle dynamics and control. Springer Science & Business Media, 2011.

[18] D. Karnopp, Vehicle stability. CRC Press, 2004.

[19] L. E. Dubins, “On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents,” American Journal of mathematics, vol. 79, no. 3, pp. 497–516, 1957.

[20] R. Oliveira, P. F. Lima, M. Cirillo, J. Mårtensson, and B. Wahlberg, “Trajectory generation using sharpness continuous dubins-like paths with applications in control of heavy-duty vehicles,” in 2018 European Control Conference (ECC). IEEE, 2018, pp. 935–940.

[21] G. P. Bevan, “Development of a vehicle dynamics controller for obstacle avoidance,” Ph.D. dissertation, University of Glasgow, 2008.