Mixed Integer-Based Speed Optimization for Autonomous Driving under Situational Variation

Takeru Goto[1,2], Rintaro Imamura[1] and Shumon Koga[1]

***Abstract*— This paper proposes a speed optimization scheme by introducing auxiliary variables to enhance the expressiveness of the formulation for autonomous driving. To achieve safe autonomous driving on general roads, it is essential to model various situations as optimization problems with multiple constraints. Our approach addresses several challenges commonly faced in time-based model predictive control, such as handling distance-based constraints (e.g., speed limit changes), adjusting constraint ranges based on urgency levels, and determining the relative position of vehicles in the target lane during merging. To tackle these issues, we introduce three types of auxiliary variables and formulate the problem as a mixed-integer linear programming model. Additionally, we provide a detailed formulation procedure for handling situational variations. Finally, we validate our approach through experiments, demonstrating that it can generate appropriate solutions for complex scenarios.**

I. Introduction

Autonomous driving is an emerging area as a novel transportation technology, ranging from consumer vehicles to taxi service. A fundamental architecture in autonomous driving systems is composed of multiple modules including perception, prediction, planning, control, etc [1]. For trajectory planning, separating the problem into path planning and speed planning is advantageous in feasibility and computational efficiency relative to solving for the trajectory as a time-sequence of vehicle's coordinate at once [2].

The approaches in speed planning can be generally categorized into optimization-based, sampling-based, and machine learning-based (ML) methods. ML approaches [3], [4], [5], [6] offer the potential to handle complex situations and avoid deadlocks by training from diverse scenario data, while concerns regarding explainability and safety in unforeseen situations remain significant. Sampling-based methods [7], [8], [9] own advantages in computational efficiency and generalizability in cost function to evaluate, while the solution can be obtained only in discrete space fixed a priori. Formal optimization approaches [10], [11] have advantages in their interpretability and theoretical guarantees within a given problem setup.

Quadratic programming (QP) methods [12], [13], [14], commonly used in model predictive control (MPC), have been intensively studied to maintain both optimality and computationally efficiency to optimize a quadratic cost under linear constraints. In [14], dynamic object constraints appearing in complex situations such as merging into a jammed lane

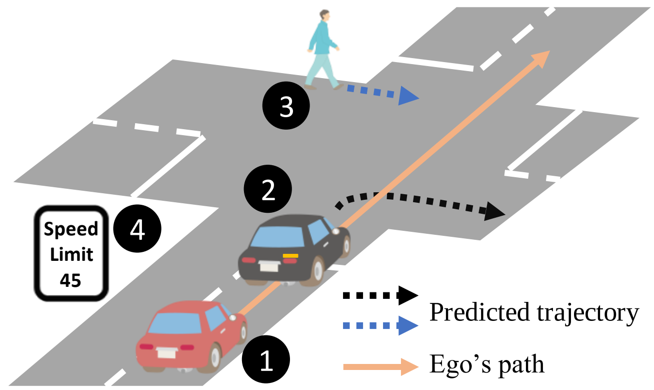

Fig. 1. The scenario for our base formulation. (1) is the ego vehicle, which is subject to acceleration and jerk limitations. (2) is a following target vehicle, requiring the ego vehicle to adjust the distance between them. (3) is a pedestrian to be avoided, and (4) represents the speed limit on this road.

of T-intersection, are converted into a space-time (ST) graph. The authors have also integrated squared norms of acceleration and jerk into the cost function to enhance the driver's ride comfort. However, it cannot handle nonlinear constraints such as speed limit changes over road sections, due to the limitation of formulation in QP. Moreover, adjusting weights of norms of acceleration and jerk requires trial and errors to satisfy desired behavior, which is challenging due to the absence of explicit target values for acceleration and deceleration. Dynamic programming approaches [15], [16], on the other hand, can effectively handle nonlinear problems and various constraints, and encourage states close to target values by increasing their evaluation scores. However, they typically suffer from the curse of dimensionality which is critical especially for speed planning to be solved over finite-time horizon.

To address these challenges, we propose a speed optimization formulation as a mixed-integer linear programming (MILP) using auxiliary binary variables, similar to those used in scheduling problems[17], [18]. MILP models have been used in trajectory planning to handle the order relationship with other objects as constraints [19], [20] and task assignments [21]. We focus on the expressiveness of this approach and present a formulation scheme that is capable of handling situations where simple time-based optimization schemes cannot manage. Our approach avoids using mixed-integer quadratic programming (MIQP) [22], which offers greater expressiveness than MILP, in order to prevent from increasing computational complexity.

[1]Innovative Research Excellence, Honda R&D Co., Ltd., Tokyo 107-6238, Japan {takeru_goto, rintaro_imamura, shumon_koga}@jp.honda

[2]Graduate School of Information Sciences, Tohoku University, Miyagi 980-8564, Japan

II. Method

Figure 1 illustrates the base scenario, including both following and avoidance targets, which is formulated as follows:

$$\underset{a \in \mathbb{R}^n}{\text{maximize}} \quad \sum_{t=1}^{n} w_t s_t \tag{1a}$$

subject to

$$v_t + a_t \Delta t/2 \le v_{\max}, \tag{1b}$$

$$v_t + a_t \Delta t/2 \ge 0, \tag{1c}$$

$$a_t \le a_{\max}, \tag{1d}$$

$$a_t \ge a_{\min}, \tag{1e}$$

$$\frac{a_t - a_{t-1}}{\Delta t} \le j_{\max}, \tag{1f}$$

$$\frac{a_t - a_{t-1}}{\Delta t} \ge j_{\min}, \tag{1g}$$

$$v_t + a_t \Delta t/2 \le v_{\text{fol}_t} \left(1 + \frac{d_{\text{fol}_t} - d_{\text{idl}_t}}{d_{\text{idl}_t}}\right), \tag{1h}$$

$$s_t \le s_{\text{avo}_t}. \tag{1i}$$

In this formulation, s_t, v_t, and a_t represent the ego vehicle's position, speed, and acceleration at time t, respectively. The variables s_t and v_t depend on the control variable a_t. Their relationships are derived from the general motion equation (see Appendix). w_t in Eq. (1a) represents the weight for s_t at each time; however, in this paper, we set $w_t = 1$. Δt denotes the discrete time step. Time index t ranges from 1 to n, with specific intervals for the following and avoidance targets: $t_{\text{fs}}, \dots, t_{\text{fe}}$ for Eq. (1h) and $t_{\text{as}}, \dots, t_{\text{ae}}$ for Eq. (1i).

This formulation results in a linear programming model, as the objective function evaluates only driving distance, unlike quadratic MPC formulations that include target values. The parameters $v_{\max}$, $a_{\max}$, and $a_{\min}$ denote the maximum speed, maximum acceleration, and minimum acceleration. Similarly, $j_{\max}$ and $j_{\min}$ represent the limits of jerk. The speed of the following target is v_{fol_t}, and the actual following distance is d_{fol_t}, which is linearly dependent on s_t. The variable d_{idl_t} represents the ideal distance, based on the predicted speed of the following target, while s_{avo_t} is the predicted position of the avoidance target. Although safety margins should be considered for other objectives, they are omitted here for simplicity but can be easily added.

This formulation cannot handle the scenarios in Fig. 2. In Fig. 2(a), there is a speed limit change. To solve this, auxiliary variable o_t is introduced. Here, o_t indicates whether s_t has reached the threshold s_{th}, and is formulated as:

$$\underset{\substack{a \in \mathbb{R}^n \\ o \in \{0,1\}^n}}{\text{maximize}} \quad \sum_{t=1}^{n} w_t s_t + C_o \sum_{t=1}^{n} o_t \tag{2a}$$

subject to

$$s_t \ge o_t s_{\text{th}}, \tag{2b}$$

$$v_t \le v_{\max_1} + o_t \left(v_{\max_2} - v_{\max_1}\right). \tag{2c}$$

To focus on the progress points from the base formulation Eq. 1, common constraints are omitted. Due to Eq. (2b), o_t becomes 1 when s_t exceeds s_{th}, triggering the change in speed limit from $v_{\max_1}$ to $v_{\max_2}$ in Eq. (2c). If $v_{\max_2}$ is greater than $v_{\max_1}$, o_t naturally becomes 1, as the higher speed increases the evaluation value. In the opposite case, a second term of Eq. (2a) compensates for its reduction, ensuring that o_t changes to 1.

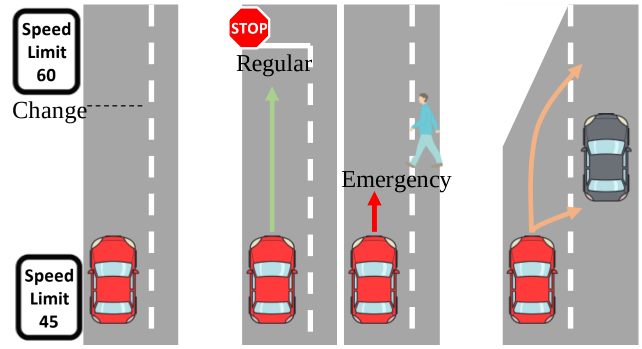

Fig. 2. (a) illustrates the situation to face the change of the speed limit. (b) contrasts situations requiring and not requiring sharp brake. (c) depicts the dilemma of merging in front of or behind adjacent vehicles.

Fig. 2(b) contrasts the regular and emergency situations. Eq. (1) defines only single limitation values for both acceleration and jerk. If these values are set for emergency situations, the driving would be excessively aggressive, and if set for regular conditions, it cannot avoid accidents in emergency situations. To balance these scenarios, an auxiliary variable e_t is introduced, and the formulation is extended as follows:

$$\underset{\substack{a \in \mathbb{R}^n \\ e \in [0,1]^n}}{\text{maximize}} \quad \sum_{t=1}^{n} w_t s_t - C_e \sum_{t=1}^{n} e_t \tag{3a}$$

subject to

$$a_t \le a_{\text{ptyp}} + e_t \left(a_{\max} - a_{\text{ptyp}}\right), \tag{3b}$$

$$a_t \ge a_{\text{ntyp}} + e_t \left(a_{\min} - a_{\text{ntyp}}\right), \tag{3c}$$

$$\frac{a_t - a_{t-1}}{\Delta t} \le j_{\text{ptyp}} + e_t \left(j_{\max} - j_{\text{ptyp}}\right), \tag{3d}$$

$$\frac{a_t - a_{t-1}}{\Delta t} \ge j_{\text{ntyp}} + e_t \left(j_{\min} - j_{\text{ntyp}}\right), \tag{3e}$$

$$v_t \le v_{\text{fol}_t} \left(1 + r_{\text{fol}} \frac{d_{\text{fol}_t} - d_{\text{idl}_t}}{d_{\text{idl}_t}}\right) + e_t c, \tag{3f}$$

$$s_t \le s_{\text{fol}_t}. \tag{3g}$$

The constraints are adjusted between typical and extreme values based on e_t. Moreover, it disables the following speed constraints by the enough huge constant c. The summation of e_t is included in the objective function to prevent unnecessary relaxation of the constraints.

Finally, we address situations where other objects appear during the time horizon, as shown in Fig. 2(c). To tackle these cases, binary auxiliary variables f_{fol} and f_{avo} are introduced for both the following and avoidance targets. These variables indicate whether the ego vehicle can pass in front of or behind these objects, and the formulation is given as follows:

$$\underset{\substack{a \in \mathbb{R}^n \\ f_{\text{fol}}, f_{\text{avo}} \in \{0,1\}}}{\text{maximize}} \quad \sum_{t=1}^{n} w_t s_t \tag{4a}$$

subject to

$$v_t + a_t \Delta t/2 \le v_{\text{fol}_t}\left(1 + r_{\text{fol}} \frac{d_{\text{fol}_t} - d_{\text{idl}_t}}{d_{\text{idl}_t}}\right) + q_{\text{fol}} c, \tag{4b}$$

$$s_t \ge q_{\text{fol}} s_{\text{fol}_t}, \tag{4c}$$

$$s_t \le s_{\text{fol}_t} + q_{\text{fol}} c, \tag{4d}$$

$$s_t \ge q_{\text{avo}} s_{o_t}, \tag{4e}$$

$$s_t \le s_{\text{avo}_t} + q_{\text{avo}} c. \tag{4f}$$

According to Eq. (4c) and Eq. (4e), q_{fol} and q_{avo} become 1 if the ego vehicle can pass in front of the objects. In the opposite case, they are set to 0, and Eq. (4d) and Eq. (4f) are applied. No additional terms are needed in the objective function, as it already aims to maximize the driving distance.

However, the combined scenario in Fig. 2 should be considered in practical applications. Therefore, we provide the combination algorithm shown in Fig. 3. This algorithm can handle complex situations involving multiple factors, such as speed limit changes, following and avoidance targets, and balancing regular and emergency situations.

III. Experiments

We conducted fundamental experiments for Eqs. (2) to (4), as well as a complex scenario test for the combined algorithm depicted in Fig. 3. Fig. 4 presents the test scenarios. The results in Fig. 5 primarily display the vertical motion states s_t, v_t, and a_t, along with the transitions of the auxiliary variables. Fig. 5(a) illustrates the effect of the auxiliary variable o_t during the speed limit change. It transitions from 0 to 1 at the change point, ensuring that v_t satisfies both the initial and subsequent speed limits. Fig. 5(a) also shows the effect of the auxiliary variable e_t in emergency situations. The ego vehicle performs a hard brake to avoid a pedestrian suddenly jumping into the lane by relaxing the acceleration and jerk constraints. The ego vehicle is also able to enter the space between two cars, as shown in Fig. 5(b), due to the auxiliary variables $q_{\text{fol}}^{(0)}$ and $q_{\text{fol}}^{(1)}$, which are not displayed in the figure since they are not time-based variables.

The complex scenario depicted in Fig. 4(d) includes two avoidance targets and one following target, which emerge during the time horizon along with a speed limit change. The results in Fig. 5(d) demonstrate that our algorithm can handle all of these challenges. The auxiliary variable e_t successfully handles the first jumping pedestrian. Based on $q_{\text{fol}}^{(i)}$, the algorithm determines whether to pass in front of or behind each pedestrian or vehicle, while also satisfying both speed limits and maintaining the appropriate following distance. Overall, these results show that our scheme is highly adaptable to situational variations.

IV. Conclusion and Discussion

We proposed a highly adaptive speed optimization formulation as a mixed-integer linear programming model. The formulation introduces auxiliary variables to handle various situations, enhancing the expressiveness for scenarios such as speed limit changes based on position, balancing emergency and regular acceleration, and managing the relative position between the ego vehicle and other objects that emerge

Algorithm 1 Formulation of speed optimization problem

Set objective function f_o as Eq. (3a)
Set constraints as Eqs. (1c) and (3b) to (3e)
for $t = 1, \ldots, n$ **do**
  $f_{v\max_t} \leftarrow v_{\max_1}$ ▷ For speed limits
  for $i = 1, \ldots, n_{\text{sp}} - 1$ **do**
    Introduce auxiliary variable $o_t^{(i)}$
    $f_{v\max_t} \leftarrow f_{v\max_t} + o_t^{(i)}(v_{\max_{i+1}} - v_{\max_i})$
    Add constraint $s_t \ge o_t^{(i)} s_{\text{th}}^{(i)}$
  end for
  Add constraint $v_t \le f_{v\max_t}$
end for
$f_o \leftarrow f_o + C_o \sum_{i=1}^{n_{\text{sp}}} \sum_{t=1}^{n} o_t^{(i)}$
for $i = 1, \ldots, n_{\text{fol}}$ **do** ▷ For following targets
  if $t_{\text{fs}}^{(i)} \ne 1$ **then** ▷ May pass in front of the target
    Introduce auxiliary variable $q_{\text{fol}}^{(i)}$
  end if
  for $t = t_{\text{fs}}^{(i)}, \cdots, t_{\text{fe}}^{(i)}$ **do**
    if $t_{\text{fs}}^{(i)} = 1$ **then**
      Add constraint
      $v_t + a_t \Delta t/2 \le v_{\text{fol}_t}\left(1 + \frac{d_{\text{fol}_t} - d_{\text{idl}_t}}{d_{\text{idl}_t}}\right) + e_t c$
    else
      Add constraint
      $v_t + a_t \Delta t/2 \le v_{\text{fol}_t}\left(1 + \frac{d_{\text{fol}_t} - d_{\text{idl}_t}}{d_{\text{idl}_t}}\right) + (q_{\text{fol}}^{(i)} + e_t) c$
      Add constraint $s_t \ge q_{\text{fol}}^{(i)} s_{\text{fol}_t}^{(i)}$
      Add constraint $s_t \le s_{\text{fol}_t}^{(i)} + q_{\text{fol}}^{(i)} c$
    end if
  end for
end for
for $i = 1, \ldots, n_{\text{avo}}$ **do** ▷ For avoidance targets
  if $t_{\text{as}}^{(i)} \ne 1$ **then** ▷ May pass in front of the target
    Introduce auxiliary variable $q_{\text{avo}}^{(i)}$
  end if
  for $t = t_{\text{as}}^{(i)}, \cdots, t_{\text{ae}}^{(i)}$ **do**
    if $t_{\text{as}}^{(i)} = 1$ **then**
      Add constraint $s_t \le s_{\text{avo}_t}^{(i)}$
    else
      Add constraint $s_t \ge q_{\text{avo}}^{(i)} s_{\text{avo}_t}^{(i)}$
      Add constraint $s_t \le s_{\text{avo}_t}^{(i)} + q_{\text{avo}}^{(i)} c$
    end if
  end for
end for
Solve this problem

Fig. 3. The combination algorithm. This scheme can handle multiple speed limits (n_{sp}), following targets (n_{fol}), and avoidance targets (n_{avo}) with consideration for emergency situations.

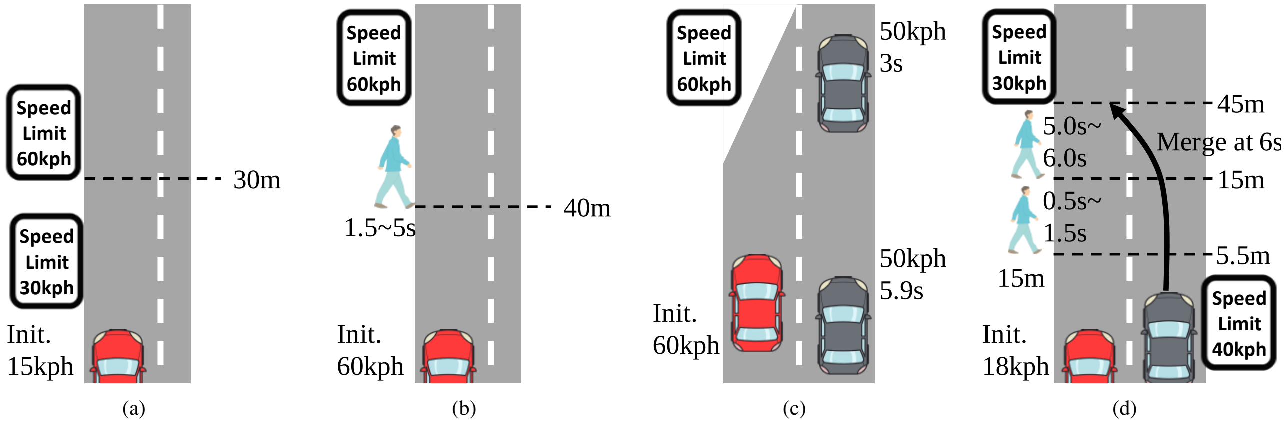

Fig. 4. Test scenarios for our experiments. (a) to (c) correspond to the scenarios in Fig. 2, while (d) represents the combined scenario of these situations.

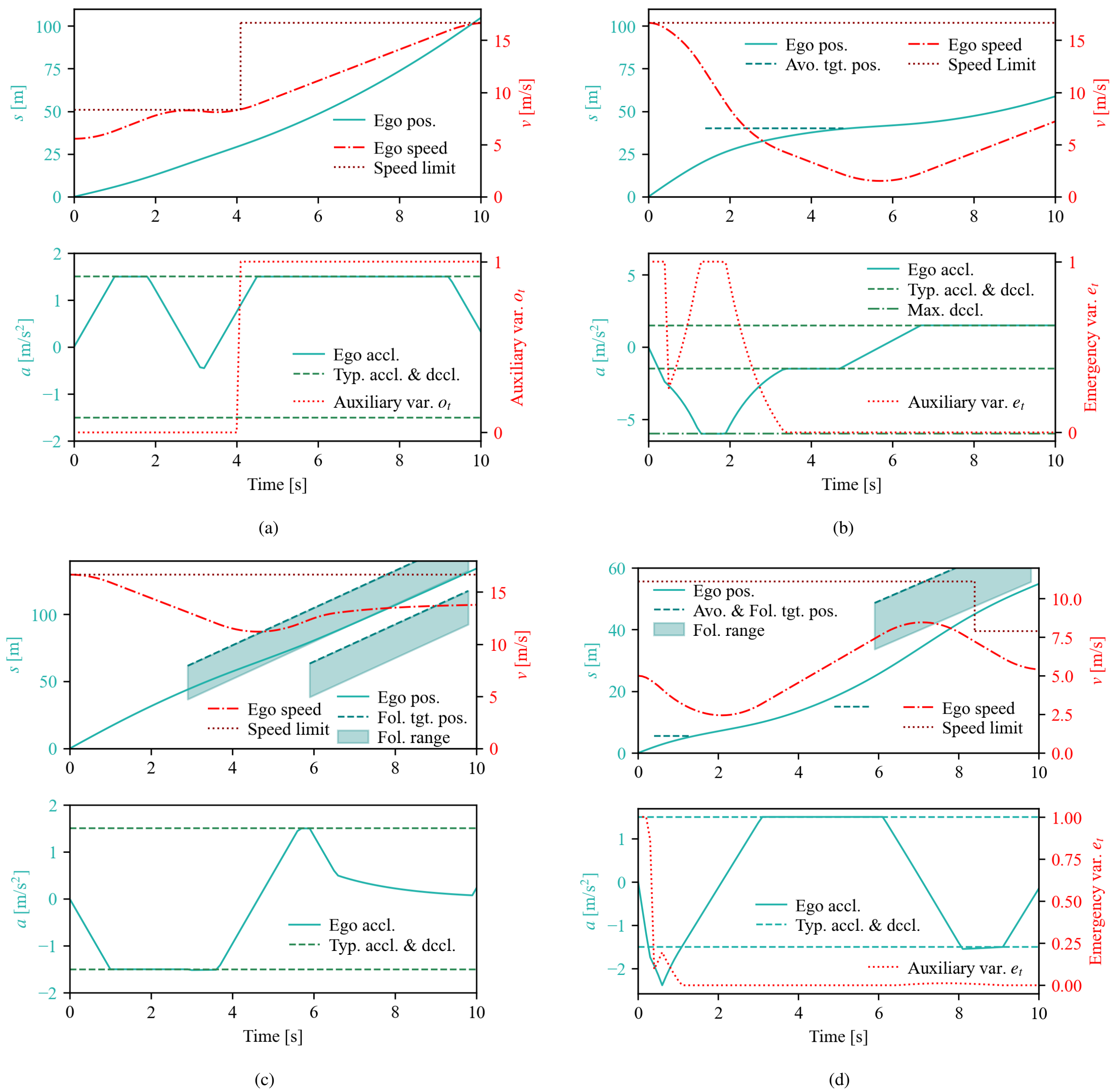

Fig. 5. The experiment results correspond to the test scenarios in Fig. 4. In (c), $q^{(0)}\text{fol} = 0$ and $q^{(1)}\text{fol} = 1$ are the result values of the auxiliary variables. The test conditions for all experiments are: $\Delta t = 0.1$, $n = 100$, $C_o = 2000$, $C_e = 10000$, and $c = 1000$.

mid-way through the time horizon. We also presented a formulation algorithm capable of addressing these complex scenarios. The experiments demonstrated that our scheme can generate appropriate speed plans for such complicated situations.

However, integer variables increase computational complexity. Therefore, we plan to investigate computation time in detail and explore hardware solutions such as graphics processing units (GPUs)[23] and field-programmable gate arrays (FPGAs)[24], or consider machine learning-assisted schemes[25] to achieve real-time calculation.

Additionally, our scheme, which only generates time-based positions along a path, relies on both a prediction module and a path generation module to function. Future work will focus on developing accurate prediction algorithms[26] and exploring multiple path generation methods[27] to build a high-performance trajectory planner. Then, we will verify the advantages of our scheme by comparing with other methods.

APPENDIX

We formulate the state equation for the vertical motion of the ego vehicle as:

$$\begin{pmatrix} s_{t+1} \\ v_{t+1} \\ a_{t+1} \end{pmatrix} = \begin{pmatrix} 1 & \Delta t & \Delta t^2/3 \\ 0 & 1 & \Delta t/2 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} s_t \\ v_t \\ a_t \end{pmatrix} + \begin{pmatrix} \Delta t^2/6 \\ \Delta t/2 \\ 1 \end{pmatrix} a_{t+1}. \tag{5}$$

We choose acceleration a_t as the control variable, but we assume constant jerk motion. Therefore, a_t is also included in the state vector. The jerk from t to $t+1$ is $(a_{t+1} - a_t)/\Delta t$, and Eq. (5) is derived from the general equation of motion. Defining the state vector as x_t, and denoting the matrix and vector as A and B, respectively, the state equation Eq. (5) becomes:

$$x_{t+1} = Ax_t + Ba_{t+1}. \tag{6}$$

Given the initial state x_0 and the control inputs $[a_1, \cdots, a_n]$, the responses are computed similarly to typical MPC as:

$$\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} A \\ A^2 \\ \vdots \\ A^n \end{pmatrix} x_0 + \begin{pmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ A^{n-1}B & A^{n-2}B & \cdots & B \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}. \tag{7}$$

REFERENCES

[1] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[2] S. Glaser, B. Vanholme, S. Mammar, D. Gruyer, and L. Nouveliere, "Maneuver-based trajectory planning for highly autonomous vehicles on real road with traffic and driver interaction," *IEEE Transactions on intelligent transportation systems*, vol. 11, no. 3, pp. 589–606, 2010.

[3] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2019.

[4] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 740–759, 2020.

[5] Y. Lu, J. Fu, G. Tucker, X. Pan, E. Bronstein, R. Roelofs, B. Sapp, B. White, A. Faust, S. Whiteson, *et al.*, "Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 7553–7560.

[6] T. Goto, Y. Kizumi, and S. Iwasaki, "Design of reward function on reinforcement learning for automated driving," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 7948–7953, 2023.

[7] M. McNaughton, C. Urmson, J. M. Dolan, and J.-W. Lee, "Motion planning for autonomous driving with a conformal spatiotemporal lattice," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 4889–4895.

[8] W. Xu, J. Wei, J. M. Dolan, H. Zhao, and H. Zha, "A real-time motion planner with trajectory optimization for autonomous vehicles," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2061–2067.

[9] C. E. Tuncali and G. Fainekos, "Rapidly-exploring random trees for testing automated vehicles," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 661–666.

[10] Z. Shiller, Y.-R. Gwo, *et al.*, "Dynamic motion planning of autonomous vehicles," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 2, pp. 241–249, 1991.

[11] J. Chen, W. Zhan, and M. Tomizuka, "Constrained iterative lqr for on-road autonomous driving motion planning," in *2017 IEEE 20th International conference on intelligent transportation systems (ITSC)*. IEEE, 2017, pp. 1–7.

[12] Z. Wang, G. Li, H. Jiang, Q. Chen, and H. Zhang, "Collision-free navigation of autonomous vehicles using convex quadratic programming-based model predictive control," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1103–1113, 2018.

[13] S. Dixit, U. Montanaro, M. Dianati, D. Oxtoby, T. Mizutani, A. Mouzakitis, and S. Fallah, "Trajectory planning for autonomous high-speed overtaking in structured environments using robust mpc," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 6, pp. 2310–2323, 2020.

[14] A. M. Anon, S. Bae, M. Saroya, and D. Isele, "Multi-profile quadratic programming (mpqp) for optimal gap selection and speed planning of autonomous driving," *arXiv preprint arXiv:2401.06305*, 2024.

[15] Y. Wei, C. Avcı, J. Liu, B. Belezamo, N. Aydın, P. T. Li, and X. Zhou, "Dynamic programming-based multi-vehicle longitudinal trajectory optimization with simplified car following models," *Transportation research part B: methodological*, vol. 106, pp. 102–129, 2017.

[16] Y. Guo, J. Ma, C. Xiong, X. Li, F. Zhou, and W. Hao, "Joint optimization of vehicle trajectories and intersection controllers with connected automated vehicles: Combined dynamic programming and shooting heuristic approach," *Transportation research part C: emerging technologies*, vol. 98, pp. 54–72, 2019.

[17] C. A. Floudas and X. Lin, "Mixed integer linear programming in process scheduling: Modeling, algorithms, and applications," *Annals of Operations Research*, vol. 139, pp. 131–162, 2005.

[18] R. Manzini, R. Accorsi, T. Cennerazzo, E. Ferrari, and F. Maranesi, "The scheduling of maintenance. a resource-constraints mixed integer linear programming model," *Computers & Industrial Engineering*, vol. 87, pp. 561–568, 2015.

[19] S. A. Fayazi and A. Vahidi, "Mixed-integer linear programming for optimal scheduling of autonomous vehicle intersection crossing," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 3, pp. 287–299, 2018.

[20] Y. Cao, Z. Zhang, F. Cheng, and S. Su, "Trajectory optimization for high-speed trains via a mixed integer linear programming approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 17 666–17 676, 2022.

[21] D. Ioan, I. Prodan, S. Olaru, F. Stoican, and S.-I. Niculescu, "Mixed-integer programming in motion planning," *Annual Reviews in Control*, vol. 51, pp. 65–87, 2021.

[22] C. Bliek1ú, P. Bonami, and A. Lodi, “Solving mixed-integer quadratic programming problems with ibm-cplex: a progress report,” in *Proceedings of the twenty-sixth RAMP symposium*, 2014, pp. 16–17.

[23] V. Boyer, D. El Baz, and M. A. Salazar-Aguilar, “Gpu computing applied to linear and mixed-integer programming,” *Advances in GPU Research and Practice*, pp. 247–271, 2017.

[24] Y. Shimai, J. Tani, H. Noguchi, H. Kawaguchi, and M. Yoshimoto, “Fpga implementation of mixed integer quadratic programming solver for mobile robot control,” in *2009 International Conference on Field-Programmable Technology*. IEEE, 2009, pp. 447–450.

[25] E. Khalil, P. Le Bodic, L. Song, G. Nemhauser, and B. Dilkina, “Learning to branch in mixed integer programming,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[26] Y. Huang, J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen, “A survey on trajectory-prediction methods for autonomous driving,” *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 652–674, 2022.

[27] D. González, J. Pérez, V. Milanés, and F. Nashashibi, “A review of motion planning techniques for automated vehicles,” *IEEE Transactions on intelligent transportation systems*, vol. 17, no. 4, pp. 1135–1145, 2015.