Long Horizon Risk-Averse Motion Planning: A Model-Predictive Approach

Chris van der Ploeg1,2, Robin Smit1, Arjan Teerhuis3, Emilia Silvas1,3

Abstract—Developing safe automated vehicles that can be proactive, safe, and comfortable in mixed traffic requires improved planning methods that are risk-averse and that account for predictions of the motion of other road users. To consider these criteria, in this article, we propose a non-linear model-predictive trajectory generator scheme, which couples the longitudinal and lateral motion of the vehicle to steer the vehicle with minimal risk, while progressing towards the goal state. The proposed method takes into account the infrastructure, surrounding objects, and predictions of the objects’ state through artificial potential-based risk fields included in the cost function of the model-predictive control (MPC) problem. This trajectory generator enables anticipatory maneuvers, i.e., mitigating risk far before any safety-critical intervention would be necessary. The method is proven in several case studies representing both highways- and urban situations. The results show the safe and efficient implementation of the MPC trajectory generator while proving its real-time applicability.

I. INTRODUCTION

Increasing levels of vehicle automation are projected to reduce significantly the number of fatalities caused by collisions and maximize driver comfort and efficiency [1]. To achieve this, it is of great importance for automated vehicles to negotiate and navigate in all traffic situations, dangerous or not, with minimal risk. While minimal risk forms an essential argument in terms of safety, progression towards the desired destination forms a large additional factor for human acceptance of automated vehicles. For example, an excessively slow-driving vehicle, which is arguably less susceptible to risks, may severely reduce acceptance [2]. While planning the motion of a vehicle, these factors of risk and progression need to be taken into account.

Although various automated driving architectures exist, there is yet no expert consent on an optimal one. Nevertheless, in general, motion planning for an automated vehicle can be divided into global (e.g., route planning) and local planning (e.g., trajectory generation) [3]. The task of global planning, i.e., finding a route from some initial location to a destination, occurs on a large road segment level and is generally done through path search algorithms such as A* [4]. It is thereafter the task of a trajectory generator to process these global points and create drivable, safe, and comfortable trajectories to traverse the path towards the destination while satisfying the requirements of minimal risk and progression.

Optimization-based methods, more specifically MPC algorithms, are often used to solve the trajectory generation problem. These methods explicitly take into account the vehicle model, as they can optimally balance multiple objectives and various constraints. Several works deploy an MPC-based trajectory generator, where the road boundaries and dynamical object avoidance are either formulated through hard constraints [5] or through the use of artificial potential fields (APFs) [6]. Note, that the notion of risk fits well with the control-theoretic notion of artificial potential fields, first introduced in [7]. These APFs allow the analytical formulation of repulsive fields, i.e., regions to stay away from, and attractive fields, i.e., regions to be attracted towards. Several works in literature advocate the use of these APFs as a replacement for hard anti-collision constraints, due to the potential non-convexity of the constraints and, hence, a higher computational burden [8]. Moreover, artificial potential fields circumvent the issue of the potential infeasibility of the optimization problem caused by unavoidable collisions, since they act in the problem cost function and, hence, a collision would not lead to the infeasibility of the problem. In this case, a minimal-risk form of damage mitigation would result in a solution, instead of no solution at all.

Model-predictive APF-based trajectory generators have been proposed in several different contexts in literature, ranging from emergency collision avoidance [6] up to longer-term planning, e.g., highway maneuvering [9]–[11]. Many of these solutions, however, only consider relatively short horizons (e.g., 1.6s [8]) due to, e.g., the high fidelity of the used model. To make informed decisions and actions, a trajectory generator is expected to look ahead for a sufficient distance or time. For reference, a human driver would, depending on the context, look ahead and anticipate on the environment 3–4 seconds up to tens of seconds [12]. Another shortcoming in the literature is related to decision-making modules or pre-planned trajectories, which accompany the model-predictive trajectory generator, and are deployed to decide and give directions on, e.g., changing a lane [8], [11]. Involving separate components to pre-determine actions or directions of the trajectory generator may result in a loss of optimality (i.e., increased risk or reduced progression).

Our contributions: In summary, there exist various solutions for model-predictive minimal risk trajectory planning. However, there does not yet exist an approach that finds minimal risk decisions and actions in an optimal manner (i.e., without the use of additional finite state machines or
Risk-averse trajectory planning without discrete decisions or pre-defined trajectories: we propose a monolithic trajectory planner which can undertake anticipatory complex risk-averse maneuvers towards a destination without the need for discrete decisions or pre-defined trajectories to follow.

(ii) Multiple operational domains: the planner is simulated both in relevant highway scenarios, as well as an urban VRU crossing scenario, incorporating both timely and late object detection (e.g., bad weather effects on sensors). This shows its applicability to a variety of operational domains and conditions.

(iii) Real-time implementation: the algorithm is implemented in the CasADi [13] optimization framework using an interior point solver (IPOPT [14]) and is deployed in real-time, showing the feasibility of solving such a problem in experimental settings.

The outline of our work is as follows. First, in Section II, a mathematical description of the problem statement is introduced. In Section III, the optimization problem is formulated through the notions of risk and progression. In Section IV, a simulation study is presented in both highway and urban scenarios, showing the expected minimal-risk results of our scenarios and their robustness in the case of late object detections. Finally, Section V concludes the work.

II. PRELIMINARIES AND PROBLEM STATEMENT

In various on-road situations (e.g., the lane change motion tasks shown in Fig. 1a) the automated vehicle needs to plan a safe, comfortable and drivable motion. To do so, the automated vehicle is expected to obey the structured environment, (i.e., the road infrastructure, traffic rules, etc.), interact with mixed-traffic (manually driven vehicles, cyclists, pedestrians), and account for their future uncertain movements in various situations (e.g. bad light, poor road markings, bad weather, dense traffic).

The vehicle, with coordinate system \((x, y)\), is expected to drive from an initial state given by \((x_0, y_0, \theta_0, v_0)\), located on top of the coordinate system (i.e., \(x_0, y_0, \theta_0 = 0\)) towards the reference state \(r_N\) which is positioned relative to the initial condition of the vehicle and contains a planar position and longitudinal velocity \(v\) as a reference, i.e., \((x_N, y_N, \theta_N, v_N)\). Here, \(N\) denotes the horizon of the trajectory generator to be planned towards the goal state, i.e., the look-ahead time. Throughout this work, we assume that the reference state, \(r_N\), is located in the center of the rightmost lane on the road at a distance of \(N \cdot t_s \cdot \bar{v}\), where \(t_s\) represents the sampling time between each trajectory element and \(\bar{v}\) represents the maximally allowed velocity, provided by the infrastructure. Furthermore, it is assumed that this reference state is known a-priori, and could be provided by a global planner (e.g., an A* algorithm [4]). The road infrastructure is assumed in this work to be described by an integer set of polynomials, representing the boundaries of the lanes, i.e.:

\[
\tilde{y}_i = c_0^{(i)} + c_1^{(i)} \tilde{x}_i + c_2^{(i)} \tilde{x}_i^2 + c_3^{(i)} \tilde{x}_i^3
\]  

where \(i \in \mathbb{Z}\), the superscript \((i)\) selects the polynomial constants of the \(i\)-th line measurement, i.e., \(c_j^{(i)}, j \in \{0, 1, 2, 3\}\), and \((\tilde{x}, \tilde{y})\) denote the planar coordinates of the \(i\)-th line relative to the vehicle initial condition. Note, that polynomial functions of arbitrary order can be used following our methodology. Finally, the vehicle is expected to safely drive along with the infrastructure in the presence of objects (e.g., vulnerable road users (VRUs) or other vehicles). The state of the objects \(o_k \forall k \in [0 \ldots N]\) is denoted as \((x_k^o, y_k^o, \theta_k^o)\), which is measured and predicted with respect to the initial condition of the vehicle. To include the predicted motion of the object in motion planning, the state of the object is assumed to be exactly measurable/predictable up to time \(N\).

![Fig. 1: Two illustrations introduce the problem statement and show the typical trade-off between driving towards a goal versus minimal-risk driving.](image-url)
avoiding a dangerous situation with the object, meaning it will inherently not reach its reference state \( r_N \).

Next to the planar movement of the vehicle (i.e., \( x, y, \theta \)), a notion of time needs to be included to be able to incorporate the risk of moving objects. As a result, the planar movement is augmented with the longitudinal velocity of the vehicle, denoted by \( v \). The main challenge of a trajectory generator is, hence, to find a reference trajectory, \( \mathcal{T} \), for the vehicle to follow, which takes into account all the above aspects,

\[
\mathcal{T} = (x_k; y_k; \theta_k, v_k) \forall k \in \{0 \ldots N\}
\]

where the trajectory length is \( N \) and satisfies kinematic vehicle behavior and comfort-driven constraints while minimizing the risk to the passengers and surrounding road users.

### III. Model-Predictive Planning for Risk-Averse Driving

In this work, we approach the trajectory planning problem from a model-predictive control (MPC) perspective. The characteristics of an MPC are in line with our requirements, namely, minimizing risk (i.e., formulating the cost as a function of risk), while satisfying kinematic vehicle dynamical behavior and constraints (e.g., limited steering input or longitudinal acceleration).

#### A. Formulation of the vehicle model

The model, used for MPC synthesis, considered in this work is the non-linear kinematic bicycle model. This model is particularly suitable for larger sampling times, i.e., larger time horizons while maintaining numerical stability [9]. The model is depicted in Fig. 2 and is described in discrete time, using forward Euler discretization with sampling time \( t_s \).

We denote the state of the model as \( x = [x, y, \theta, v] \) and its inputs as \( u = [\delta, a] \), and the state-update rules are denoted as follows

\[
\begin{align*}
x_{k+1} &= x_k + t_s v_k \cos \theta_k , \\
y_{k+1} &= y_k + t_s v_k \sin \theta_k , \\
\theta_{k+1} &= \theta_k + t_s \frac{v_k}{L} \tan \delta_k , \\
v_{k+1} &= v_k + t_s a_k .
\end{align*}
\]

where the states \( x, y, \theta \) denote the planar movement relative to the initial condition, as denoted in Fig. 1a, \( v \) represents the velocity in the direction of the \( x \)-axis, \( \delta \) denotes the steering angle of the front axle, \( a \) denotes the acceleration in the direction of the \( x \)-axis and finally \( L \) denotes the wheelbase of the vehicle. The state vector \( x \) and the input vector \( u \) belong to the sets

\[
\begin{align*}
X &= \{ x \in \mathbb{R}^4 | x \leq \bar{x} \leq x \}, \\
U &= \{ u \in \mathbb{R}^2 | u \leq \bar{u} \leq u \},
\end{align*}
\]

respectively. The bounds of the sets are governed by infrastructural, dynamic, and comfort limitations.

#### B. Formulation of the risk

Similar to [15], we consider two sources of risk, namely, infrastructural risk and object-imposed risk. The infrastructural risk is described as the safety risk taken to travel across certain road boundaries [15]. We define the infrastructural risk, at time instance \( k \), by \( \mathcal{U}_k \). Furthermore, we define a risk-free situation, i.e., \( \mathcal{U}_k \approx 0 \) if and only if the vehicle is driving in the center of a lane and assuming that all lanes are driveable. The risk-function of the infrastructure, perceived by the vehicle at time instance \( k \), can be described as a set of structured Gaussian functions, using the polynomial road description (1), as follows

\[
\mathcal{U}_k = \sum_{i=1}^{N_f} A_i e^{-\frac{(v_k - \tilde{v}_i(t_i))^2}{2\sigma^2}}, \forall i \in \mathbb{Z}^+
\]

where \( N_f \) represents the number of road lines, \( A_i \) represents the amplitude of the risk and \( \sigma \) represents the standard deviation of the risk. The amplitude and standard deviation of the infrastructural risk are assumed constant, though may depend on the type of lanes that are divided (e.g., a lane with oncoming traffic may be riskier to drive on, hence the amplitude may be higher). This risk-field description is a slight adaption of the description in [16], where only a two-lane highway is considered. An illustrational example of an infrastructural potential field is provided in Fig. 3a.

The relative risk of dynamical objects is described using a repulsive potential field formulated as a multivariate, potentially rotated, Gaussian distribution as follows:

\[
\mathcal{U}_k^{O} = \sum_{n=1}^{N_o} A_O e^{-\frac{r^2}{2\sigma^2}},
\]

\[
f_r^{(n)}(x_k) = \left[ x_k - \hat{x}_k^{(n)} \ y_k - \hat{y}_k^{(n)} \right] \Sigma^{-1} R^T \left[ x_k - \hat{x}_k^{(n)} \\
y_k - \hat{y}_k^{(n)} \right],
\]

\[
\Sigma = \begin{bmatrix} \sigma_x^2 & 0 \\ 0 & \sigma_y^2 \end{bmatrix}, \quad R = \begin{bmatrix} \cos \hat{\theta}_k^{(n)} & -\sin \hat{\theta}_k^{(n)} \\ \sin \hat{\theta}_k^{(n)} & \cos \hat{\theta}_k^{(n)} \end{bmatrix}
\]

where \( \hat{x}_k^{(n)}, \hat{y}_k^{(n)}, \hat{\theta}_k^{(n)} \) represents the state of the \( n \)-th object, \( N_o \) represents the number of objects, \( A_O \in \mathbb{R}^N \) represents for each object the maximal risk, \( \sigma_x \) represents the longitudinal standard deviation of the perceived risk of the object and \( \sigma_y \) represents the lateral standard deviation of the perceived risk of the object. Note, again, that for simplicity...
the amplitude and standard deviations are assumed constant, though they can be varied based on the type of object (e.g., pedestrian versus a car) or for example the velocity at which the ego vehicle is driving. An illustrational example of such a risk field is provided in Fig. 3b.

C. Formulation of the risk-averse trajectory generator

Using the definition of the vehicle dynamic model (3), its limitations (4) and the definition of the risk field for the infrastructure (5) and objects (6), the optimization problem for a risk-averse trajectory generator can be composed. The cost and the constraints of a risk-averse trajectory generator are formulated by the following model-predictive program:

\[
J_{x_0}^{*,N}(x_s) = \min_{u,x} \sum_{k=1}^{N-1} \ell_x(x_k - r_k) + \sum_{k=0}^{N-1} \ell_u(u_k)
\]

\[
+ m(x_N - r_N) + U_k^2 + U_k^1, \quad \text{s.t.} \quad x_0 = x_s
\]

\[
x_{k+1} = f(x_k, u_k), \quad \forall k \geq 0
\]

\[
g(x_k, u_k) \leq 0, \quad \forall k \geq 0
\]

where, in (7a), the functions \(\ell_x(\cdot)\), \(\ell_u(\cdot)\) quadratically penalizes the control error \(x_k - r_k\), and the input \(u_k\), respectively. Furthermore, the term \(m(\cdot)\) quadratically penalizes the terminal state at time \(N\). The state \(x_s\) in (7b) represents the state at present time. The function \(f\) in (7c) represents the nonlinear dynamics as depicted in (3). Finally, the constraints in (7d) enforce the boundedness of the state \(x\) and the input \(u\) through enforcement of the sets (4). Note, that the stage cost \(\ell_x\) is not used in this work (i.e., equal to zero), due to the fact that there is only the requirement to drive to a goal at time instance \(N\), hence, only the terminal cost \(m(\cdot)\) is used to penalize the state.

IV. SIMULATION STUDY

The non-linear program is implemented in a real-time operating optimization framework CasADi [13] using the IPOPT Interior-point embedded solver [14]. The trajectory generator is deployed in four different studies, where urban- and highway driving in a risk-defined environment forms the basis. Throughout the first three simulations, the same vehicle parameter set is used to show the versatility of the algorithm without having to tune it for every separate use case. For the urban case, slight adjustments are made. The parameter-set is provided in Table I, where it has to be noted, that any parameters which are not denoted are either 0 in the case of stage weights or not set in the case of state bounds. In each of the test cases, the computational time is denoted\(^1\). The optimization problem is initiated at each instance \(t_s\), hence, by real-time we mean that the MPC trajectory generator can generate a new solution within a period of \(t_s\) seconds. Finally, in the simulation, we assume perfect tracking of the trajectory by the vehicle model.

A. Case I: single overtake

In the first case, we consider a scenario where the vehicle drives in the rightmost lane and approaches several slow-driving vehicles. The objects are driving at a velocity of 5\(\text{ms}^{-1}\) (bottom lane) and 2\(\text{ms}^{-1}\) (middle lane), respectively as typical in highway traffic jam situations. The results are shown in Fig. 4 at several interesting points in time. First, the vehicle approaches the slow-driving vehicle in the bottom lane and overtakes the vehicle. Then, the vehicle has three choices, it can either pass the object on the top or bottom lane, or it can slow down and drive behind the vehicle. In

---

\(^1\)The results are generated on an Intel Core i7-10850 2.7GHz platform.
this case, it is of minimal cost (i.e., a combination of risk and benefit) to return to the original lane, such that the vehicle can, once again, drive towards the goal point. The average computational time of the planner in this scenario is 30.4ms with a standard deviation of 5.1ms.

![Fig. 4: Simulation results of case I with two vehicles driving slowly on the two rightmost lanes, motivating a single lane change.](image)

**B. Case II: double overtake**

In the second case, we consider the scenario where the vehicle starts on the rightmost lane, overtakes a slow vehicle, and then decides whether to slow down, or to once again overtake due to a static vehicle in the middle lane. The objects are driving at a velocity of $5\text{ms}^{-1}$ (bottom lane) and $0\text{ms}^{-1}$ (middle lane), respectively. The results are shown in Fig. 5. The first overtake maneuver is identical to case I. Once the vehicle approaches the static vehicle, it can be observed that a lane change to the right would compromise the taken risk due to the proximity of the right-lane object prediction at horizon $N$ ahead. As a result, the MPC takes the anticipatory decision to take another lane change to the left, at the cost of a higher lateral distance error to the desired goal point. The average computational time of the planner in this scenario is 30.2ms with a standard deviation of 4.1ms.

![Fig. 5: Simulation results of case II with two vehicles driving slowly on the two rightmost lanes, motivating a double lane change.](image)

**C. Case III: blocked lanes, slowly driving participants**

In the third case, we consider the scenario where the three-lane road is blocked by slowly driving objects. The objects are all driving at a velocity of $5\text{ms}^{-1}$ and their longitudinal position initial condition deviates to challenge the algorithm. The results are shown in Fig. 6. It can be observed that the MPC notices that there is only one real sensible solution, which is to stay inside the lane and brake to the same velocity as the succeeding vehicles. Note, that the steady-state euclidean distance between the vehicles can be tuned using the longitudinal parameter $\sigma_x$. The average computational time of the planner in this scenario is 28.2ms with a standard deviation of 1.6ms.

**D. Case IV: VRU crossing with potential late object detection**

Whereas case I to III presented highway maneuvers, the fourth and final case will cover an urban driving setting, showing the applicability of the MPC trajectory generator for design domains other than highway driving. In this case, we consider the vehicle driving along a two-lane road. On this two-lane road, a VRU is about to cross the road as depicted in Fig. 7. This case is deemed to be highly relevant and largely responsible for VRU incidents (up to 51% [17]). The proposed simulations are twofold. First, we study how the MPC trajectory generator acts when it sees the VRU and can anticipate its actions based on these observations. Subsequently, we study how the trajectory generator acts when the VRU observation is obstructed and the detection of the VRU is late (advocated in [17] as a relevant root cause in this particular scenario). Transitioning from a highway scenario towards an urban scenario requires an adaptation of MPC parameters. The revised parameters, different from Table I, are proposed in Table II. In an urban scenario, the maneuvers likely require a higher tracking fidelity, for this reason, the sampling time is reduced and, as a result, the horizon is increased to achieve a similar look-ahead time concerning the previous cases. Moreover, the reduced longitudinal velocity results in a change of bound and a change...
in the longitudinal standard deviation of the object risk-field. Finally, an urban scenario with two-lane roads usually allows bi-directional traffic. As a result, the infrastructural potential field amplitude is increased (as shown in Table I and also in Fig. 7 by the infrastructural risk field color). The VRU crosses the road at a velocity of 1ms$^{-1}$ and the ego vehicle starts at an initial velocity of 8.33ms$^{-1}$. First, we consider the results without late VRU detection, as depicted in Fig. 8. It is clear that, from 23s onward the vehicle starts to anticipate and brake, down to a velocity of 3ms$^{-1}$, to wait for the VRU to pass. Once the VRU has passed, the vehicle accelerates back to its original velocity, which is a natural and intuitive expected behavior of the vehicle. Now, consider a late detection of the VRU (coming, e.g., from bad weather conditions) of which the results are depicted in Fig. 9. The detection occurs at around 30s (i.e., the top graph), a time at which there is only 13m of distance left concerning the object. At this time, the trajectory generator path indicates that braking is no longer an option and an evasive maneuver towards the left lane is necessary. Once the evasive maneuver is finished, the vehicle returns to its lane while a safe distance has been kept with the VRU. The average computational time of the planner for the early VRU detection is 43.5ms with a standard deviation of 4.2ms. The average computational time of the planner in the late detection case is 48.2ms with a standard deviation of 8.1ms.

| Parameter       | Description                        | Value  | Unit  |
|-----------------|------------------------------------|--------|-------|
| $t_s$           | Sampling time                       | 0.38 s |       |
| $N$             | MPC horizon                         | 20     | -     |
| $A_I$           | Infrastructure risk amplitude       | 200    | -     |
| $\sigma_x$      | Longitudinal risk deviation         | 10 m   |       |
| $\gamma, \eta$  | Lateral displacement bounds         | [1,6]  | m     |
| $v, \nu$        | Vehicle velocity bounds             | [0, 8.33] | ms$^{-1}$|

TABLE II: Changed parameters w.r.t. Table I for case IV.

The increased mean computational time can be explained largely by the fact that, due to the higher required fidelity of the path, the horizon is doubled.
V. CONCLUSION

In this work, we have proposed a monolithic model-predictive trajectory generator that, as opposed to many works in literature, does not use a set of pre-defined decisions or trajectories to decide on actions to progress towards a goal point while minimizing risk. This offers a simpler solution to trajectory generation, which can be used for longer horizons while being computationally feasible. The proposed method explicitly considers predicted trajectories of other road users and it’s shown effective in various use cases. The method is shown to be effective by demonstrating it in four relevant scenarios in which the approach shows the ability to make logical and safe decisions in both highway and urban scenarios. Furthermore, the planner is challenged in a more difficult urban scenario, where the late detection of a pedestrian (for example happening due to bad weather) is compared with the timely detection. The results show that this approach can function safely under both circumstances. Future work includes the adaptation of the MPC problem formulation and the potential fields to incorporate even more complex situations and be robust for measurement uncertainty, induced by predictions or faults (for example, the lack of fault isolability in object fault detection [18]). Moreover, adaptive potential fields will be studied to make the trajectory generator suitable for miscellaneous environments.

VI. ACKNOWLEDGMENTS

This work is supported by the EU Horizon 2020 R&D program under grant agreement No. 861570, project SAFE-UP (proactive SAFety systems and tools for a constantly UPgrading road environment).

REFERENCES

[1] A. Papadoulis, M. Quddus, and M. Imprialou, “Evaluating the safety impact of connected and autonomous vehicles on motorways,” Accident Analysis and Prevention, vol. 124, pp. 12–22, 2019.
[2] J. De Freitas, A. Censi, B. Smith, L. Lillo, S. Anthony, and E. Frazzoli, “From driverless dilemmas to more practical commonsense tests for automated vehicles,” Proceedings of the National Academy of Sciences of the United States of America, vol. 118, no. 11, 2021.
[3] 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, 2016.
[4] M. Likhachev and D. Ferguson, “Planning long dynamically feasible maneuvers for autonomous vehicles,” International Journal of Robotics Research, vol. 28, no. 8, pp. 933–945, 2009.
[5] B. Gutjahr, L. Gröll, and M. Werling, “Lateral vehicle trajectory optimization using constrained linear time-varying mpc,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 6, pp. 1586–1595, 2017.
[6] J. Ji, A. Khajepour, S. Melek W.W., and Y. Huang, “Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints,” IEEE Transactions on Vehicular Technology, vol. 66, no. 2, pp. 952–964, 2017.
[7] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” IEEE International Conference on Robotics and Automation, pp. 500–505, 1985.
[8] S. Dixit, U. Montanaro, S. Fallah, M. Dianati, D. Oxtoby, T. Mizutani, and A. Mouzakitis, “Trajectory planning for autonomous high-speed overtaking using mpc with terminal set constraints,” IEEE Conference on Intelligent Transportation Systems, pp. 1061–1068, 2018.
[9] V. A. Laurense and J. C. Gerdes, “Long-horizon vehicle motion planning and control through serially cascaded model complexity,” IEEE Transactions on Control Systems Technology, vol. 30, no. 1, pp. 166–179, 2022.
[10] Z. Huang, Q. Wu, J. Ma, and S. Fan, “An apf and mpc combined collaborative driving controller using vehicular communication technologies,” Chaos, Solitons and Fractals, vol. 89, pp. 232–242, 2016.
[11] P. Hang, C. Lv, C. Huang, J. Cai, Z. Hu, and Y. Xing, “An integrated framework of decision making and motion planning for autonomous vehicles considering social behaviors,” IEEE Transactions on Vehicular Technology, vol. 69, no. 12, pp. 14458–14469, 2020.

[12] S. Calvert, W. Schakel, and J. van Lint, “A generic multi-scale framework for microscopic traffic simulation part ii – anticipation reliance as compensation mechanism for potential task overload,” Transportation Research Part B: Methodological, vol. 140, pp. 42–63, 2020.

[13] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi – A software framework for nonlinear optimization and optimal control,” Mathematical Programming Computation, 2018.

[14] A. Wächter and L. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” Mathematical Programming, vol. 106, no. 1, pp. 25–57, 2006.

[15] L. Li, J. Gan, X. Ji, X. Qu, and B. Ran, “Dynamic driving risk potential field model under the connected and automated vehicles environment and its application in car-following modeling,” IEEE Transactions Intelligent Transportation Systems, vol. 23, no. 1, pp. 122–141, 2022.

[16] F. Siebenrock, M. Gunther, and S. Hohmann, “Ltvmpc based trajectory planning considering uncertain object prediction through adaptive potential fields,” IEEE Conference on Control Technology and Applications, pp. 666–672, 2020.

[17] L. Yue, M. Abdel-Aty, Y. Wu, O. Zheng, and J. Yuan, “In-depth approach for identifying crash causation patterns and its implications for pedestrian crash prevention,” Journal of Safety Research, vol. 73, pp. 119–132, 2020.

[18] C. van der Ploeg, R. Smit, A. Siagkris-Lekkos, F. Benders, and E. Silvas, “Anomaly detection from cyber threats via infrastructure to automated vehicle,” in European Control Conference, 2021, pp. 1788–1794.