A Bi-Level Real-Time Microsimulation Framework for Modeling Two-Dimensional Vehicular Maneuvers at Intersections

Rahmani, Saeed; Neumann, Jan; Suryana, Lucas Elbert; Theunisse, Christiaan; Calvert, Simeon C.; Van Arem, Bart

DOI
10.1109/ITSC57777.2023.10422492

Publication date
2023

Document Version
Final published version

Published in
2023 IEEE 26th International Conference on Intelligent Transportation Systems, ITSC 2023

2023 IEEE 26th International Conference on
Intelligent Transportation Systems (ITSC)
24-28 September 2023. Bilbao, Bizkaia, Spain

A Bi-level Real-time Microsimulation Framework for Modeling Two-dimensional Vehicular Maneuvers at Intersections

Saeed Rahmani[1], Jan Neumann[2], Lucas Elbert Suryana[1], Christiaan Theunisse[3], Simeon C. Calvert[1], Bart van Arem[1]

2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC) | 979-8-3503-9946-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ITSC57777.2023.10422492

***Abstract*— Intersections are critical bottlenecks within urban transportation networks. Current models for simulating two-dimensional (2D) vehicular movements at intersections are met with limitations in accurately representing complex interactions and capturing vehicle dynamics. Accordingly, this paper proposes a novel microsimulation framework for trajectory planning and vehicular control at intersections. The model considers vehicle dynamics and control variables, such as acceleration and steering angle, and releases the popular assumption that there is full knowledge sharing or cooperation among vehicles at intersections. These features make the proposed framework more realistic compared to previous microsimulation attempts and applicable to traffic flow and environmental impact assessment studies. In addition, it efficiently operates in real-time for multiple vehicles, overcoming the limitations of offline methods. Moreover, the model is capable of accounting for driver/vehicle detection range, reaction time, and perception and prediction inaccuracies, which enhances its suitability for safety assessments. The evaluation in several scenarios indicates the ability of the proposed framework in real-time planning and following realistic and consistent 2D paths while avoiding collisions with other vehicles.**

I. INTRODUCTION

Intersections are critical components of urban transportation networks and a significant portion of urban congestion and accidents are happening at or near intersections. Accordingly, ensuring their safe and efficient design and operation is paramount to the overall performance of the entire traffic network. Achieving a realistic model that sufficiently represents the interactions and movements of vehicles within intersections can be highly beneficial to the design, control, and evaluation of intersections. However, due to the complex interactions and weak lane-following behaviors, developing microsimulation models for describing the decision-making and movements of vehicles within intersections is challenging and still an open research question [1].

Current endeavors in modeling 2D vehicular movements can be broadly divided into four main categories: Lane-based traffic flow models, cellular automata (CA) models, social force theory, and optimal control techniques [2]. Traffic flow models, which were initially developed to understand and predict the dynamics of vehicular traffic on roadways, have been calibrated [3] or modified [4] for studying the traffic flows at intersections. However, the interaction mechanism described in these studies is following, bi-vehicle interaction, which is not realistic at intersections where multiple vehicles interact with each other from different angles.

CA models have also been utilized for modeling movements of vehicles at intersections by focusing on the interactions among vehicles [5], [6] or including the driver's decision-making process [7]. These models have also been extended to optimize the intersection flow [8], [9]. However, the fixed-size, grid-based lattices in CA models make them inaccurate in representing the kinematics of the vehicle and the spatial variations of vehicular movement.

Social force theory has been mainly applied to modeling the interactions among pedestrians or cyclists [10], but it has been also used for modeling vehicular movements at intersections. Ma et al. [11] developed a model for vehicles at intersections, but their assumptions were valid in lane-based environments. Yang et al. [12] modeled the movement of vehicles around a work zone within an intersection, but the interaction with crossing vehicles was not considered. While social force models offer a more comprehensive consideration of the interactions, they have shortcomings in modeling vehicular traffic. Firstly, the resulting trajectory is determined based on the combination of several types of "forces" rather than explicitly modeling the driver's decision-making or vehicle dynamics [2]. This limits their interpretability and applicability to safety and efficiency impact assessment and their capability in considering the non-holonomic constraints of vehicles. Secondly, due to the high complexity of these models, their scalability and application in dense situations are challenging [11], and the relatively high number of parameters makes their calibration process demanding.

Optimal control theory, which is a popular tool in modeling vehicle movements at intersections [13], [14], has been often applied based on the assumption of cooperation, connectivity, or controllability among vehicles [15]. Recently, Zhao et al. [1] proposed a model for a single vehicle moving within an intersection and further extended their model by modifying its cost function to incorporate the interactions among vehicles [2]. Although this model proved to successfully plan two-dimensional trajectories for vehicles at intersections, it still lacks some important features.

Firstly, the proposed model in [2] assumes that all vehicles are aware of the utility function of other vehicles and that

[1]Saeed Rahmani, Lucas Elbert Suryana, Simeon C. Calvert, and Bart van Arem are with the Department of Transport and Planning, Delft University of Technology, 2628 CN Delft, The Netherlands. s.rahmani@tudelft.nl; l.e.Suryana@tudelft.nl; s.c.calvert@tudelft.nl; b.vanarem@tudelft.nl.

[2]Jan Neumann is with the Department of Computer Science, Czech Technical University in Prague, Prague, Czech Republic. neumaja5@fel.cvut.cz

[3]Christiaan Theunisse is with the Department of Cognitive Robotics, Delft University of Technology, 2628 CN Delft, The Netherlands. c.theunisse@student.tudelft.nl

there is full cooperation among them to avoid collisions. Moreover, the kinematics of the vehicles is not modeled, which overlooks the state dependencies and non-holonomic constraints of vehicles. In addition, their model is estimated offline, which limits its applicability for real-time simulation and visualization. Lastly, their framework lacks some features crucial for safety impact assessment. For instance, the sight distance of the driver/vehicle and the perception and anticipation errors cannot be added to the model plainly.

To address some of these gaps, a microsimulation framework is proposed for modeling 2D vehicular maneuvers at intersections. The proposed framework releases the assumptions that there is full knowledge sharing or cooperation among vehicles since each vehicle is controlled decentrally using its own model and without accessing others' utility functions. This is achieved by utilizing a collision avoidance module. It is also designed efficiently for real-time simulation. This is important since many of the previous optimization-based methods are solved offline and cannot be used for real-time simulation and visualized investigations. Furthermore, the proposed framework utilizes a vehicle dynamics model to account for its kinematics. The controller within the framework uses acceleration and steering angle as the input variables, which is intuitive and makes the model interpretable and applicable to traffic flow, environmental, and motion comfort studies. Finally, the detection range of the driver/vehicle, their reaction time, and their perception inaccuracy can be seamlessly incorporated into the proposed framework, which makes it suitable for safety impact assessment. Evaluation in several scenarios indicates the validity of the presented framework in planning and following two-dimensional trajectories for vehicles in real-time.

The remainder of the paper is organized as follows. In the next section, the proposed model is described in detail. In section III, the designed scenarios for evaluating the framework are presented. section IV presents the results of the designed scenarios. Finally, future research directions are discussed in section V.

II. METHODOLOGY

The proposed framework enjoys a bi-level design. At the higher level, an efficient global planner is designed to find a kinematically feasible and optimal path for the vehicle considering the design of the intersection and traffic rules (which is called *reference trajectory* hereinafter). Then, a controller is utilized to find the optimal values for acceleration and steering angle to follow the reference trajectory while avoiding collisions with other vehicles. It is worth noting that there is a feedback loop, meaning that if the variation from the reference trajectory is large, the global planner is again activated to find a new reference path. This design enables achieving a real-time simulation model (as will be discussed in Sections II-B and II-C), and provides flexibility for employing different collision avoidance modules and incorporating the detection range and reaction time of the driver. In the following subsections, different components of the proposed frameworks are described in more detail.

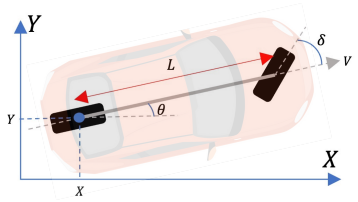

Fig. 1. Kinematic bicycle model representation of a vehicle

A. Vehicle Dynamics and Motion

The controller used in the proposed framework is a model predictive controller (MPC), which requires a system model (in this case, a vehicle model) for predicting and simulating the vehicle states through time and finding the optimal control values. In this study, the kinematic bicycle model is used to represent the vehicle dynamics (Figure 1). Using a bicycle model allows for considering the kinematics of the vehicle, including the interrelations between the position, heading angle, steering angle, speed, acceleration, and length of the vehicle, which is essential for 2D movement modeling. Polak et al. [16] showed that the kinematic bicycle model can produce consistent trajectory when compared to the 9th-order realistic vehicle dynamics model, which is a significant improvement over the simplified assumption of point mass that is often used in traffic flow microsimulation models.

The model used in this paper is represented as a differential equation $\dot{x} = f(x, u)$, where $f(x, u)$ consists of four differential equations $\dot{X}, \dot{Y}, \dot{V}$, and $\dot{\theta}$. Those differential equations can be arranged as a non-linear state space:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{V} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} V cos(\theta) \\ V sin(\theta) \\ a \\ \frac{V}{L} tan(\delta) \end{bmatrix} \tag{1}$$

where the state of the vehicle, symbolized by x, includes the longitudinal position, latitudinal position, heading, and speed of the vehicle that are represented by X, Y, θ, and V, respectively. The acceleration and steering angle are the control inputs (u) and are depicted by a and δ, respectively. The distance between the front and rear wheels of the bicycle model is denoted by L.

1) Linearization: Although using the bicycle model allows for considering the dynamics of the vehicle, its nonlinear form might be computationally inefficient. Therefore, a linearized version of the bicycle model is used in this study to increase the efficiency of the framework for the sake of real-time simulation. Estimating the linear bicycle model is achieved by using the Taylor series expansion (Equation 2). This linear form can be sufficiently accurate, especially at low speeds [17]. As vehicles do not adopt very high speeds at intersections, this representation can produce reliable results. This will be evaluated in section IV.

$$\dot{f}(x, u) \approx f(\bar{x}, \bar{u}) + \frac{\partial f(x, u)}{\partial x}|_{\bar{x},\bar{u}}(x - \bar{x}) + \frac{\partial f(x, u)}{\partial u}|_{\bar{x},\bar{u}}(u - \bar{u}) \tag{2}$$

where $\bar{x}$ and $\bar{u}$ represent the state and input elements of the operating point. Accordingly, the linearized kinematic bicycle state-space equation is obtained as follows:

$$\dot{x} = Ax + Bu + d \tag{3}$$

where x, u, A, B, and d are as follows:

$$x = \begin{bmatrix} X \\ Y \\ V \\ \theta \end{bmatrix}, A = \begin{bmatrix} 0 & 0 & \cos(\phi) & -v\sin(\phi) \\ 0 & 0 & \sin(\phi) & v\cos(\phi) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{tan(\sigma)}{L} & 0 \end{bmatrix},$$
$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & \frac{v}{L\cos^2(\sigma)} \end{bmatrix}, C = \begin{bmatrix} a \\ \delta \end{bmatrix}, d = \begin{bmatrix} v\sin(\phi)\phi \\ -v\cos(\phi)\phi \\ 0 \\ \frac{-v\sigma}{L\cos^2(\phi)} \end{bmatrix} \tag{4}$$

The operating point of θ, V, and δ are ϕ, v, and σ, respectively. The linearization is carried out for every time step of the simulation and then the linearized model is used in the optimization problem.

B. Global Planner

Most current models for describing 2D vehicular maneuvers at intersections use a pre-defined path for the vehicle [1], which limits their realism in capturing the dynamics of the vehicles within the intersection. Contrarily, in this study, a high-level planner is designed to construct a two-dimensional trajectory for the vehicle. The resulting trajectory (including the reference speed) is then given as a reference path to the controller. In this regard, starting from the vehicle's current position, a directed graph of motion primitives is constructed towards the goal area and an optimal graph search algorithm is designed to find the optimum path from the current position to the goal area on the graph. The graph expands to (and the search algorithm only visits) those branches that do not collide with any obstacle, do not enter the areas prohibited by traffic rules, and do not deviate much from the location and the orientation of the goal area. Putting these conditions significantly decreases the graph size resulting in a much faster implementation of the global planner. This reduced the planner's run-time from several seconds to a hundredth of a second, which is a significant improvement toward achieving real-time planning and simulation.

To show the generalizability and validation of the proposed framework, the motion primitives for the global planner are created by simulating a nonlinear vehicle model consistent with the kinematics described in Section II-A. Using a *nonlinear* vehicle model for the planner allows testing the accuracy of the controller utilizing a *linearized* bicycle model in following the generated trajectory. Also, using a vehicle model in the global planner forces the planner to search through and generate only those trajectories that are consistent with the kinematics of the vehicle, which makes the optimization problem even more efficient. However, any set of motion primitives could be generated and used according to the problem-specific needs.

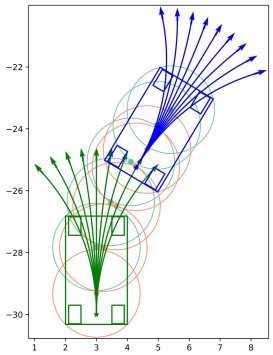

Fig. 2. Motion primitives for two consecutive vehicle positions.

Figure 2 shows the motion primitives for two adjacent nodes (one vehicle at two different time steps). The circles drawn around the vehicle are for collision checking. The vehicle is represented as a set of two circles of the same radius to the front and back. This representation streamlines collision checking by keeping a balance between computational efficiency and accuracy by reducing the problem into an efficient matrix multiplication problem based on the radius of the circles and the location of the obstacles.

1) Graph Search Algorithm: The graph search algorithm in this framework is an A* algorithm. Multiple functions are defined to find the optimal path on the graph: 1) a "neighbor" function, which for a given node, returns a set of adjacent nodes and the cost of the respective edges. The cost of an edge is the length of the motion primitives, which is defined by the user. 2) A "goal check" function, determining if a node is within the goal area. The goal function accepts any node as the final goal if its *location* is within a predetermined goal area, and the *orientation* of the motion primitive in that node is within the allowed range θ_{margin}. 3) Lastly, an admissible heuristic function assigns a value to each node to guide the A* algorithm toward the goal area. The heuristic is shown in Equation 5, where Δx and Δy represent the shortest distance to the goal in x and y directions, respectively, and $\Delta\theta$ and θ_{margin} are the difference between the current and goal orientations and the relaxation on the goal orientation, respectively. α is an empirically chosen parameter that weighs the two parts of the equation.

$$H = \sqrt{\Delta x^2 + \Delta y^2} + \alpha * \max(0, |\Delta\theta| - \theta_{\text{margin}}) \tag{5}$$

Figure 3 shows the result of the search algorithm. All graph nodes visited by the search algorithm are also depicted using colored dots. As is shown, the graph is only expanded to a limited area, which enables real-time re-planning when required due to the interactions with other vehicles.

C. Controller and Vehicle's Motion

After finding a reference trajectory, a controller is needed to specify the control inputs and describe the dynamics of the vehicle when following the trajectory. This is achieved by specifying the amount of acceleration and steering angle

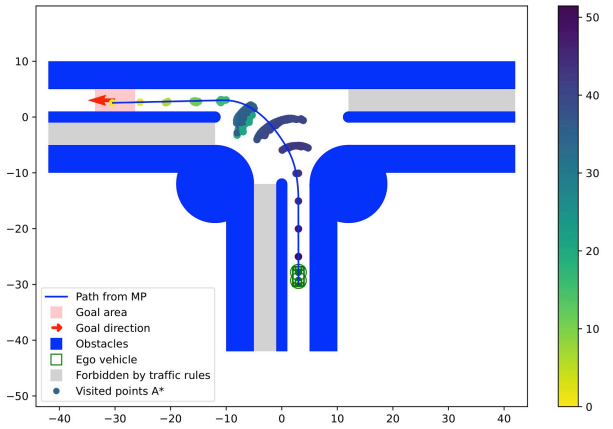

Fig. 3. Trajectory found by the global planner for a sample T-intersection. graph nodes visited by the search algorithm are visualized. The color of the points represents the heuristic function value.

by solving an optimal control problem in the context of an MPC. The selection of acceleration and steering angle stems from their correlation with the controls typically used in real life (that are the acceleration and braking pedals, and the steering wheel). Furthermore, this configuration enables the imposition of penalties on variations of acceleration or steering angle, which can lead to non-smooth and uncomfortable driving. Thus, the framework can be readily utilized for evaluating the efficiency and smoothness of different driving policies.

1) Controller: The MPC controller is formulated in a linear and discrete form. The state-space model of the vehicle is described in Equation 6:

$$x(k+1) = A_d x(k) + B_d u(k) + d_d \tag{6}$$

where A_d B_d, and d_d are the discrete versions of matrix A, B, and d from Equation 4. The vehicle's states and inputs for each time instants $t = kT_s$, $k = 0, 1, 2, ...$, are represented by $x(k)$ and $u(k)$, respectively, where T_s represents the time sampling interval for discretizing the system from Equation 3. The system here is assumed to be full-state feedback so that all states can be measured.

The general aim of the controller is to minimize the cost $\mathbf{J}$ of a quadratic function, which is formulated as follows:

$$\begin{aligned}\mathbf{J} = &\sum_{\substack{k=0\dots N \\ x^{\text{ref}}(k) \text{ not end}}} ||\delta^k_{XY}||^2_{Q_{XY\perp}} + ||\delta^k_{XY}||^2_{Q_{XY\parallel}} + ||\delta^k_{V\theta}||^2_{Q_{V\theta}} \\ &+ \sum_{k=0}^{N-1} ||u(k)||^2_R + ||u(k+1) - u(k)||^2_{R_d} \\ &+ \sum_{\substack{k=0\dots N \\ x^{\text{ref}}(k) \text{ is end}}} ||x(k)||^2_{Q_f}\end{aligned} \tag{7}$$

where $\delta^k_{XY} = x_{XY}(k) - x^{\text{ref}}_{XY}(k)$ and $\delta^k_{V\theta} = x_{V\theta}(k) - x^{\text{ref}}_{V\theta}(k)$, and the term ref refers to the reference trajectory. N, $x_{XY}(k)$, $x_{V\theta}(k)$ are, respectively, the time horizon, coordinate states, and velocity and yaw states. The parallel and perpendicular weight matrices, $Q_{XY\parallel}$ and $Q_{XY\perp}$, are used to penalize the *parallel* and *perpendicular* deviations from the reference path, differently. The weight $Q_{V\theta}$ aims at penalizing the deviation of speed and heading from the reference point on the trajectory. R and R_d are the weight matrices to penalize the intensity of input ($u(k)$) and the variations of input, respectively. These weights try to minimize the effort needed to reach the goal and to assure a smooth movement for the vehicle. Lastly, Q_f is the weight matrix for the final state to ensure stability and reaching the target state. The optimization problem is formulated as follows:

$$\begin{aligned}\min \quad & \mathbf{J} \\ \text{s.t.} \quad & x(0) = x_{init} \\ & x(k+1) = A_d x(k) + B_d u(k) + d_d \\ & |\dot{\delta}(k)| \leq \max\ \dot{\delta} \\ & |\delta(k)| \leq \max\ \delta \\ & \min\ V \leq V(k) \leq \max\ V \\ & \min\ a \leq a(k) \leq \max\ a\end{aligned} \tag{8}$$

where $x(0) = x_{init}$ indicates the initial state, $max\ \dot{\delta}$ and $max\ \delta$ specify the maximum steering angle and maximum changing rate of the steering angle, min and $max\ V$ are the minimum and maximum allowed speeds, min and $max\ a$ are the minimum and maximum allowed accelerations.

2) Collision Avoidance: In the proposed framework, it is assumed that the information about the intentions of vehicles is not known by the ego vehicle. Also, unlike previous studies [2], there is no assumption about full cooperation among vehicles. Therefore, a collision avoidance module is needed so that the ego vehicle can avoid colliding with others. This is done by predicting the future states of the ego vehicle and the surrounding vehicles in a finite space and time horizon.

To this end, firstly, a detection range is defined for the ego vehicle, within which it can *observe* the other vehicles' states (including their current location, speed, and heading). This can represent the sight distance of the driver or the detection range of the vehicle's sensors. Secondly, a prediction horizon is defined, for which the vehicle should predict the future states of itself and others. This is important because the control horizon in MPC is chosen relatively short for the sake of model efficiency and real-time simulation, but this short horizon might not be sufficient to timely react to potential collisions. Therefore, it is necessary to make predictions for a longer time horizon even for the ego vehicle. In this study, the prediction horizon is specified so that the ego vehicle can stop with its maximum allowed deceleration from the maximum allowed speed. For the sake of simplicity, and to show the capability of the proposed framework in avoiding collisions, the trajectory of other vehicles in this study is predicted based on the simple assumption that they keep their current speed and steering angle. As the simulation is updated with a frequency of 5-10 Hz, this simple assumption, given a sufficiently large prediction horizon, can successfully avoid collisions (as it is shown in section IV). For predicting the ego vehicle's trajectory, it is assumed that it will

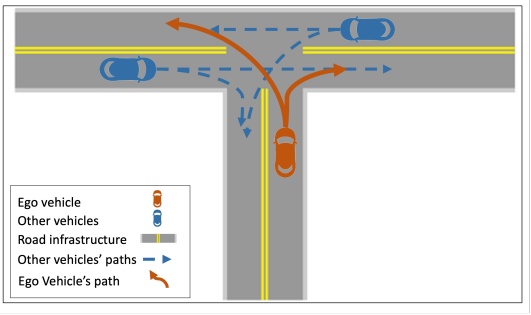

Fig. 4. Possible combination of maneuvers in Scenario Set 3

accelerate with its desired (comfortable) acceleration toward the desired speed along the planned path. This is because we have basic information about the intentions of the ego vehicle, so we can have a more realistic prediction for the ego vehicle compared to the others.

If a conflict is detected based on the predicted trajectories of the ego vehicle and all other vehicles, the controller is informed about the location of the conflict, and the reference trajectory is updated according to the conflict point. This modified reference trajectory is then used as the new reference trajectory within the controller. This collision avoidance is run at each time step, and the reference trajectory is updated, if necessary. This is possible thanks to the efficient global planner and local controller.

III. EXPERIMENT SETUP

In order to evaluate the performance of the developed modeling framework, several scenarios were designed and tested in the context of a T-intersection. These scenarios include the followings:

- *Scenario 1* – The ego vehicle moves within the intersection without interacting with other vehicles. This scenario is intended to test the validity of the global planner, the graph search algorithm, and the ability of the controller in following the planned trajectory while considering the kinematics of the vehicle.
- *Scenario 2* – The ego vehicle *turns right* and interacts with another vehicle crossing the intersection. This scenario aims to evaluate both the performance of the global planner and the controller coupled with the collision avoidance module.
- *Scenario Set 3* – The ego vehicle *turns left* and interacts with two other vehicles coming from different directions and making turning or straight-line maneuvers. Also, different goal areas are defined for the ego vehicle to test the right-turning and left-turning capability of the controller. Scenario Set 3 aims to evaluate the ultimate performance of the established framework in different and challenging conditions.

Figure 4 shows a combination of these scenarios.

IV. RESULTS

Figure 5 shows the result of testing *Scenario 1*. As can be seen in Figure 5.a, the global planner has been successful in finding a smooth path toward the goal, and the controller has followed the reference path accurately (blue is the planned path and red is the followed path achieved by using the controller). The speed profile for the whole simulation period (Figure 5.b) indicates that the vehicle smoothly increased its speed and then decreased it toward zero when getting closer to the goal area. The amount of deviation from the planned path depicted in Figure 5.c indicates that the controller using a linearized bicycle model has successfully tracked the global path and its deviation has been in the order of centimeters. As mentioned in section II-B, the global path is achieved based on the motion primitives of a nonlinear vehicle model, while the controller uses a linearized bicycle model. It is worth noting that part of this deviation could be related to the limitations and constraints set within the optimal control problem, and therefore, by fine-tuning the optimization problem, even lower deviations may be achievable.

The results of running *Scenario 2* are depicted in Figure 6. The red dot in Figure 6.a (which is a snapshot of the scenario during the simulation) shows that the collision avoidance module has successfully detected a potential conflict and avoided it by setting a stopping point before the conflict point. Moreover, Figure 6.b shows that the vehicle decreased its speed at around t=3.5s when it detected the conflict, and at time t=7.5s, when no potential collision is detected any further, it starts accelerating to the desired speed again. Finally, when the vehicle approaches the goal area, it decelerates again to stop at the final point.

Finally, a sample of the results for Scenario Set 3 is depicted in Figure 7. The red dot in Figure 7.a indicates that the vehicle has successfully identified the potential collision and modified its reference trajectory, accordingly. Also, based on Figure 7.b, the vehicle has detected the potential collision with the first vehicle at around t=4s and starts decelerating. At approximately t=7s, it decides to accelerate because the first vehicle is passing the intersection, but suddenly the second vehicle starts a left turn maneuver, and therefore, the ego vehicle again decelerates to nearly a full stop. After the second vehicle also passes the intersection at around time t=9s, the ego vehicle again accelerates to the desired speed. Similarly to previous scenarios, when approaching the goal area, the vehicle again decelerates to a full stop. The results related to the deviation from the planned trajectory (depicted in Figure 7.c) also indicate that despite several changes in

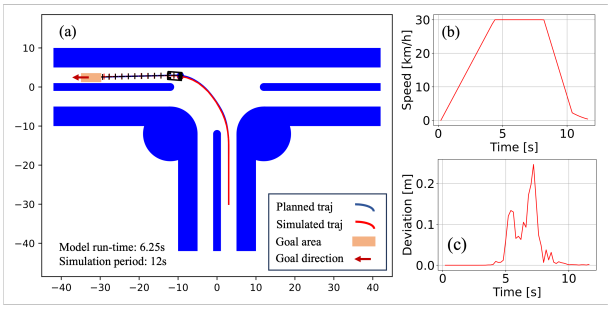

Fig. 5. Results of running the model in Scenario 1

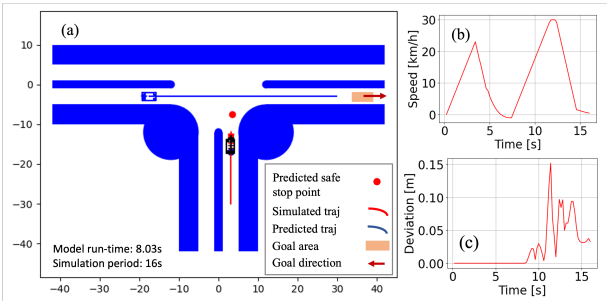

Fig. 6. Results of running the model in Scenario 2

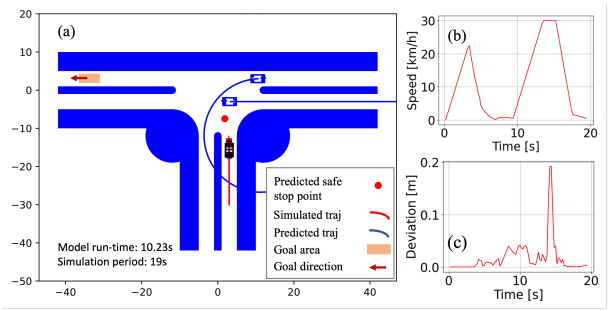

Fig. 7. Results of running the model in a sample scenario from the Scenario Set 3

trajectory planning, the controller has been successful in following the planned trajectories.

In all scenarios, the model was successful in *real-time planning and trajectory following*. Also, the speed profiles indicate that the movement of the vehicle was fairly smooth in the evaluated scenarios. All in all, the proposed framework showed it can successfully plan and follow two-dimensional trajectories for describing vehicular movements at intersections while avoiding collisions. Moreover, the values of run-time and the simulation period indicated in Figure 5.a, Figure 6.a, and Figure 7.a show that the framework has successfully found a reference trajectory and followed it in real-time, which is a significant improvement compared to the model proposed in [2] with a reported run-time of around 150 seconds for a two-vehicle scenario.

Thanks to its efficient implementation and realistic results, as well as its capability to take into account the kinematics of the vehicles and the detection range of the drivers or sensors, this framework is suitable for simulating vehicular movements at intersections.

V. CONCLUSION

In this study, a bi-level framework is proposed for modeling two-dimensional vehicular movements at intersections for microsimulation purposes. The proposed framework presents a significant advancement in planning and controlling vehicle maneuvers at intersections by effectively accounting for vehicle dynamics and embracing a decentralized and real-time approach. Its efficiency, real-time applicability, and ability to incorporate key parameters like driver/vehicle detection range, reaction time, and perception inaccuracy, distinguish it from previous methods. These features not only enhance its applicability to traffic flow and environmental studies, but also position it as a valuable tool for safety impact assessment studies. Future research can be directed toward incorporating perception and decision-making errors into the proposed framework. Also, calibrating the model against real-world data instead of simulated trajectories seems a relevant following step. Finally, validating the framework is a wider range of scenarios, including full intersections, is suggested.

REFERENCES

[1] J. Zhao, V. L. Knoop, and M. Wang, "Two-dimensional vehicular movement modelling at intersections based on optimal control," *Transportation Research Part B: Methodological*, vol. 138, pp. 1–22, 2020.

[2] ——, "Microscopic traffic modeling inside intersections: interactions between drivers," *Transportation science*, vol. 57, no. 1, pp. 135–155, 2023.

[3] T. V. Mathew and P. Radhakrishnan, "Calibration of microsimulation models for nonlane-based heterogeneous traffic at signalized intersections," *Journal of Urban Planning and Development*, vol. 136, no. 1, pp. 59–66, 2010.

[4] K. D. Alemdar, A. Tortum, Ö. Kaya, and A. Atalay, "Interdisciplinary evaluation of intersection performances—a microsimulation-based mcda," *Sustainability*, vol. 13, no. 4, p. 1859, 2021.

[5] M. E. Foulaadvand and S. Belbasi, "Vehicular traffic flow at a non-signalized intersection," *Journal of Physics A: Mathematical and Theoretical*, vol. 40, no. 29, p. 8289, 2007.

[6] X.-G. Li, Z.-Y. Gao, B. Jia, and X.-M. Zhao, "Cellular automata model for unsignalized t-shaped intersection," *International Journal of Modern Physics C*, vol. 20, no. 04, pp. 501–512, 2009.

[7] C. Chai and Y. D. Wong, "Fuzzy cellular automata model for signalized intersections," *Computer-Aided Civil and Infrastructure Engineering*, vol. 30, no. 12, pp. 951–964, 2015.

[8] F. Zhu and S. V. Ukkusuri, "Modeling the proactive driving behavior of connected vehicles: A cell-based simulation approach," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 4, pp. 262–281, 2018.

[9] L. Cruz-Piris, M. A. Lopez-Carmona, and I. Marsa-Maestre, "Automated optimization of intersections using a genetic algorithm," *IEEE Access*, vol. 7, pp. 15 452–15 468, 2019.

[10] X. Chen, M. Treiber, V. Kanagaraj, and H. Li, "Social force models for pedestrian traffic–state of the art," *Transport reviews*, vol. 38, no. 5, pp. 625–653, 2018.

[11] Z. Ma, J. Sun, and Y. Wang, "A two-dimensional simulation model for modelling turning vehicles at mixed-flow intersections," *Transportation Research Part C: Emerging Technologies*, vol. 75, pp. 103–119, 2017.

[12] D. Yang, X. Zhou, G. Su, and S. Liu, "Model and simulation of the heterogeneous traffic flow of the urban signalized intersection with an island work zone," *IEEE Transactions on intelligent transportation systems*, vol. 20, no. 5, pp. 1719–1727, 2018.

[13] C. Yu, W. Sun, H. X. Liu, and X. Yang, "Managing connected and automated vehicles at isolated intersections: From reservation-to optimization-based methods," *Transportation research part B: methodological*, vol. 122, pp. 416–435, 2019.

[14] Y. Bichiou and H. A. Rakha, "Developing an optimal intersection control system for automated connected vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1908–1916, 2018.

[15] J. Wu and X. Qu, "Intersection control with connected and automated vehicles: a review," *Journal of Intelligent and Connected Vehicles*, vol. 5, no. 3, pp. 260–269, 2022.

[16] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle, "The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles?" in *2017 IEEE intelligent vehicles symposium (IV)*. IEEE, 2017, pp. 812–818.

[17] Q. Ge, Q. Sun, S. E. Li, S. Zheng, W. Wu, and X. Chen, "Numerically stable dynamic bicycle model for discrete-time control," in *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*. IEEE, 2021, pp. 128–134.