The following publication C. Huang, H. Huang, J. Zhang, P. Hang, Z. Hu and C. Lv, "Human-Machine Cooperative Trajectory Planning and Tracking for Safe Automated Driving," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 8, pp. 12050-12063, Aug. 2022 is available at https://dx.doi.org/10.1109/TITS.2021.3109596.

Human-Machine Cooperative Trajectory Planning and Tracking for Safe Automated Driving

Chao Huang, *Member, IEEE* , Hailong Huang, *Member, IEEE* , Junzhi Zhang, Peng Hang, *Member, IEEE*, Zhongxu Hu *Member, IEEE* , Chen Lv, *Senior Member, IEEE*

***Abstract*—This paper investigates a human-machine cooperative trajectory planning and tracking control approach for automated vehicles. The proposed method is developed based on a novel algorithm of cooperative human-machine rapidly-exploring random (HM-RRT) for path planning, together with the risk assessment of driver behavior. First, the driver's behaviour is assessed according to the information of the predicted vehicle trajectory, the identified safe driving area and the driving risks evaluated in both lateral and longitudinal directions. Based on the driver's expected driving task, when driving risks are identified by real-time assessment, then the human-machine cooperation is activated during trajectory planning. By HM-RRT, the newly developed safety assurance mechanism for path planning, the cooperative trajectory is then generated, which incorporates the driver's desire and actions and automation's corrective actions, to ensure the safety, stability and smoothness of the human-vehicle system. The simulation and experimental results show that the proposed HM-RRT algorithm can effectively improve the convergence rate and reduce the computation load, comparing to the conventional method. Beyond this, the proposed human-machine cooperation approach is able to simultaneously ensure the safety, stability and smoothness of the vehicle and largely reduce human-machine conflicts in real-time applications, demonstrating its feasibility and effectiveness.**

***Index Terms*—Human-machine cooperation, trajectory planning, HM-RRT, tracking control, automated driving**

I. Introduction

In recent years, automated driving has been gaining increasing attention from both academia and industry, as it has a great potential to reduce traffic accidents, optimize transportation efficiency, and improve our mobility experience [1, 2]. Although fully autonomous driving is seen as the ultimate goal, highly automated driving with human-in-the-loop is considered to arrive sooner. Thus, in the foreseeable future, the cooperation between human driver and automation functionalities will play an important role in the development of automated driving technologies [3, 4].

Advanced driver assistance system (ADAS) is a promising means to improve the system performance and reduce the workload of human drivers [5, 6]. It can assist the driver with specific driving tasks. For example, an adaptive cruise control system can automatically apply braking to keep a reference speed and safe distance away from the vehicle ahead [7]. A forward collision avoidance (FCA) system will provide automatic control of the vehicle in case of an imminent forward collision [8]. In these systems, the driver still retains in the control loop and interacts with the automation at the control level.

Although ADAS have many advantages, how to ensure vehicle safety via allocating the control authority to human driver and automation system remains a challenge. An effective approach for achieving such driver-automation collaboration is *shared control* [9]. In the shared control system, the automation continuously supports the driver on control action to follow a reference path, achieving a predefined driving task. Based on the modality of the human-machine interface, the shared control can be divided into visual, auditory and haptic forms. The performance improvement of visual and auditory supports in shared control is limited, as they are usually provided as a warning to the human driver when the risk level exceeds a threshold [10]. By using the haptic interface, such as a haptic force or torque applied on a joystick or a steering wheel, the automation can continuously assist the human driver by providing a corrective control input [11, 12].

Despite the above promising aspects, conflicts would also inevitably occur between the human driver and automation when there is a deviation in their control inputs during shared control [13]. For example, during shared control when the driver steers the vehicle from the lane centre to avoid crashing to an undetected obstacle, yet the automation system keeps the vehicle moving along with the lane centre line. In this case, the human driver has to exert extra torque to override the automation control and execute his or her intended trajectory, resulting in the increased driver control effort and driving risk [14]. Another classic example is that, although the directions of the driver and automation's control inputs are the same during shared control, the amounts of their executed control inputs are different, which also leads to conflicts between the human and machine. Moreover, the frequent conflict between human and machine would deteriorate the subjective driving experience and vehicle control performance [15]. Thus, vehicle safety and smoothness are expected to be enhanced via shared control, however, the conflicts during human-machine collaboration are

C. Huang, P. Hang, Z. Hu, and C. Lv are with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, 639798, Singapore. (E-mails: {chao.huang, peng.hang, zhongxu.hu, lyuchen}@ntu.edu.sg)

H. Huang is with the Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University, Hong Kong. (E-mail: hailong.huang@polyu.edu.hk).

J. Zhang is with the School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China. (E-mail: jzhzhang@mail.tsinghua.edu.cn)

Corresponding author: C. Lv

unexpected and should be addressed [16].

There are some ways to solve the aforementioned human-machine conflict problem. One simple and practical way is to directly shut off automation's support or reduce the level of support when conflicts occur [15]. Besides, Billings applies the principle of human-centred automation to minimize the human-machine conflicts [17]. This approach allows the human driver to hold the highest authority, and the automation's action is generated based on the human driver's intention. [18] investigates the concept of arbitration in order to make an equitable decision for human-machine cooperation in emergencies. The outcome is able to mitigate the conflict by offering the human driver as much the degree of freedom in control as possible. However, the human-machine conflict is oriented from the expectation mismatch in the higher-level task of trajectory planning, rather than the deviation of control inputs between the human driver and the automation. [19] proves that the collaboration between human driver and automation at the planning level can effectively reduce human-machine conflicts. Nevertheless, so far very few studies investigate the human-machine collaboration in trajectory planning [20].

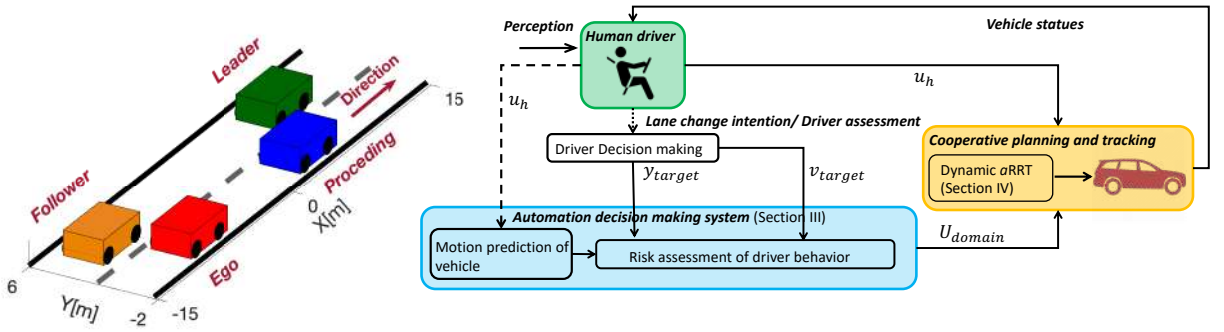
Currently, most of the publications in the area of trajectory planning mainly focus on collision-free trajectory generations by considering vehicle dynamics and surrounding traffic information. The geometry-based approach is well known for its strong ability in reasoning and calculation efficiency. There mainly four types of the geometry-based approach, i.e. the Dubin curve [21], Bezier curve [22, 23], spline [24] and polynomial curves [25]. Besides, the numerical optimization-based methods find feasible trajectories based on the differential cost function and constraints [26]. By solving the formulated optimization problem, an optimal trajectory can be obtained. However, the complex driving conditions may result in a non-convex optimization problem which is computationally difficult to be solved [27]. The graph-search approaches can construct a graphical discretization of the vehicle's state space and generate a cluster of trajectory candidates based on the predefined patterns. Then, the optimal trajectory can be selected based on a certain cost function. The popular used graph-search methods include the Hybrid $A^\star$ [28], Dijkstra algorithm [29], $D^\star$-lite [30] and Anytime Weighted $A^\star$ [31]. Rapidly-exploring Random Trees (RRT) is a powerful tool to find feasible trajectories in complex and dynamic environments [32]. However, the algorithm is very time-consuming due to a large number of iterations needed, and meanwhile, the path generation does not consider the traffic rules [33, 34]. In general, research in human-machine cooperation in trajectory planning of automated driving has rarely been seen.

To further advance the shared control technology and mitigate human-machine conflicts, in this paper, we propose a novel methodology of human-automation cooperation by uniquely considering the driver behaviors in a newly developed cooperative trajectory planning algorithm. The proposed approach allows for a continuous adaption of the vehicle trajectory planning to the varying performance of the *human driver*. As a result, the *human driver* still retains the main authority of trajectory planning, while the *automation* accommodates the *human driver*'s desire but compensates human driver's undesired behaviour to ensure vehicle safety when needed. The contributions of this paper are three folds: 1) A novel cooperative trajectory planning and tracking method is proposed to address the issue of human-machine conflict; 2) A novel HM-RRT algorithm is proposed to significantly decrease the number of iterations and improve the computation efficiency of the traditional RRT algorithm, leading to an accelerated convergence rate in trajectory planning; 3) The proposed approach can effectively integrate the trajectory planning module and tracking control module, which further improves the computational efficiency of the planning and control algorithms for autonomous driving.

The remainder of the paper is structured as follows. The high-level system methodology and algorithm architecture are illustrated in Section II. The decision-making module of human-machine cooperation, which consists of the safe area identification, risk assessment of driver behavior, and the trigger mechanism of human-machine cooperation, is described in Section III. The novel HM-RRT-based cooperative trajectory planning and tracking algorithm is designed in Section IV. The testing, validation and results are presented in Section V. Section VI concludes this paper.

II. The High-Level System Architecture

The proposed human-machine cooperative trajectory planning and tracking algorithm is depicted in Fig. 1. As shown in Fig. 1a, the Red vehicle is the controlled ego vehicle (denoted in short as *ego*). Three obstacle vehicles, including the *preceding*, *leader* and *follower*, are considered in this study. The high-level structure of the proposed approach is represented in Fig. 1b. Based on the sensing information of the traffic environment and the predicted intentions of surrounding vehicles, the decision-making module of the automated driving system makes an appropriate decision on the driving mode (e.g. lane-keeping or lane-changing), which further determines the target lane y_{target} and the desired velocity v_{target}. The design of the decision-making module is not involved in the present research, and we mainly focus on the downstream trajectory planning and tracking via human-machine cooperation. At each step of the trajectory planning, the human driver's input, $u_h = [a_h, \delta_h]^T$, is implemented to the *constant turn rate and acceleration* (CTRA) model for predicting the vehicle's motion. δ_h and a_h are the steering wheel angle and vehicle acceleration provided by the *human driver*, respectively. If the predicted vehicle motion generated by the driver action is judged as safe and reasonable, then no support from the *automation* will be provided in trajectory planning and tracking control, and the *Human driver* owns the full authority on vehicle control. However, when the driver's control input is assessed as risky, then the automation system will be activated, and the newly developed HM-RRT algorithm will be utilized to generate an additional corrective control



(a) Defined driving scenario (b) Overview of the proposed trajectory planning and tracking-based shared control system

Fig. 1: The architecture of the proposed human-machine cooperative trajectory planning and tracking approach.

signal u_a, together with u_h, to control the *ego* moving safely within a bounded space. More specifically, the constrained Delaunay Triangulations (CDT) technique is adopted to identify the bounded safe space for the *ego* to travel, and the artificial potential field (APF) and the dynamic potential field (DPF) are applied to evaluate the driving behavior of *human driver* in the *lateral* and *longitudinal* directions, separately. In this way, the human driver's intention and preferences are embedded into the trajectory planning level, and the *automation* collaborates with the *human driver* in a more harmonic manner, and the human-machine conflicts can be effectively reduced. More details of the key functionality modules and algorithms are discussed subsequently.

III. The Trigger Mechanism of the Human-Machine Cooperation

In this section, the prediction method of the vehicle motion based on the driver's inputs is firstly introduced. Then, the identification of the safe space for driving is developed, and this is the foundation for risk assessment of *human driver*'s behavior. Finally, the trigger mechanism of the human-machine cooperation is designed, and the corresponding control input domain U_{domain}, which is used for the HM-RRT-based trajectory planning and tracking algorithm, is also determined.

A. Prediction of vehicle states using the CTRA model

In this subsection, vehicle's future trajectory is predicted under the human driver's control input $u_h = [a_h \ \delta_h]^T$ by using the bicycle model and the CTRA model [35]. In the CTRA model, the following assumptions are made: the vehicle operates in a circular path between two consecutive time steps, and the yaw rate and acceleration of the vehicle within the two time steps are kept constant. For the control vector u_h, the corresponding yaw rate r, yaw angle ψ and lateral velocity v_x of the vehicle can be calculated based on the bicycle model, which will be described in Section IV-A. Based on r, ψ and v_x, and considering the heading angle of the road ψ_{road}, the CTRA model is utilized to predict vehicle's trajectory over a time horizon. The following equation describes the kinematics of the CTRA vehicle model:

$$\xi(k+\tau_p) = \begin{cases} x^{\oplus}(k) + \Delta x^{\oplus}(\tau_p) \\ y^{\oplus}(k) + \Delta y^{\oplus}(\tau_p) \\ \psi(k) - \psi_{road} + \tau_p r \\ v_x(k) + a_h \tau_p \\ a_h \\ r(k) \end{cases} \tag{1}$$

with

$$\begin{aligned} \Delta x^{\oplus}(\tau_p) = \frac{1}{r^2}[&(v_x r_f + a_h r \tau_p)\sin(\psi^{\star}(k) + \tau_p r) \\ &+ a_h \cos(\psi^{\star}(k) + \tau_p r) \\ &- v_x r \sin(\psi^{\star}(k)) - a_h \cos(\psi^{\star}(k))] \end{aligned}$$

and

$$\begin{aligned} \Delta y^{\oplus}(\tau_p) = \frac{1}{r^2}[&(-v_x r_d - a_h r \tau_p)\cos(\psi^{\star}(k) + \tau_p r) \\ &+ a_h \sin(\psi^{\star}(k) + \tau_p r) \\ &+ v_x r \cos(\psi^{\star}(k)) - a_h \sin(\psi^{\star}(k))]. \end{aligned}$$

where $\psi^{\star}(k) = \psi(k) - \psi_{road}$, $\xi = [x^{\oplus} \ y^{\oplus} \ \psi \ v_x \ a_h \ r]^T$ is the state vector. $x^{\oplus}$ and $y^{\oplus}$ represent the vehicle's position. τ_p is the prediction horizon. It should be noted that a small prediction horizon may lead to a result that the predicted trajectory is close to the current trajectory. However, a large prediction horizon may result in over-shoots of the predicted trajectory. In this work, we set $\tau_p = 0.5s$. The predicted position $x^{h,p}_{pred} = (x^{\oplus}(\tau_p), y^{\oplus}(\tau_p))$ and predicted longitudinal velocity $x^{h,v_x}_{pred} = v_x(\tau_p)$ are used to generate a trajectory, where subscript $pred$ denotes prediction, and superscript h represents *human driver*, and p and v_x denote the position and longitudinal velocities, respectively.

B. Identification of the safe driving space

In this subsection, the CDT technique is adopted to identify a safe driving space for the vehicle. The safe space is used for assessing the risk of the driver's control behavior (refer to

Section III-C for details), and it is leveraged as the configuration space in the proposed HM-RRT to explore (refer to Section IV for details). One advantage of the CDT method is that it requires much less data storage of environment information than other cellular methods. Given a driving task, i.e. lane changing or lane keeping, as illustrated in Fig. 2, the obstacle vehicles and road boundaries are described by line segments. Then the feasible driving region can be partitioned into multiple triangles. The safe space can be considered as a polygon, which consists of a sequence of triangles connecting the starting point to the end region. The starting point is set as the *ego*'s current position. The end region is defined as the position, which is a pre-defined distance d_o away from the *target* vehicle (i.e. the *preceding* during lane-keeping and from the *leader* in the lane change task). It should be noted that the safe area in the lateral direction is also bounded by the traffic rules. In the case of lane-keeping, the 'width' of the safe space cannot exceed the width of the current lane (the distance between the white dash line and the lower black solid line). And in the case of lane-changing, it can not exceed the width of two lanes.

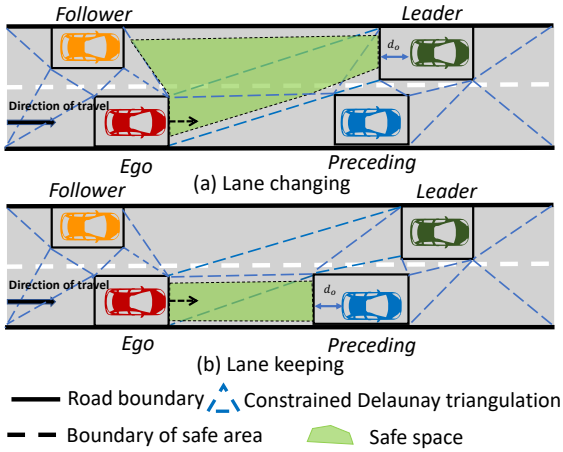


Fig. 2: Safe space identification based on the CDT technique in different driving tasks.

The steps of identifying the safe space C can be summarized in Algorithm 1. The boundaries of the safe space C are considered as non-overcome boundaries. The vehicle should not drive closely to the *upper* boundary and *lower* boundary. More details can be found in [36].

C. Risk assessment of the driver's control behavior

In this subsection, the risk assessment of driver's control behavior in both lateral and longitudinal directions is illustrated. First, we utilize the APF method to measure how close the vehicle's predicted position is to the boundaries of C in the lateral direction. The following equation describes the calculation of the generated repulsive potential of the boundaries:

Algorithm 1: GenerateSpace()

For a given driving task, identify S_p and F_p, and discrete the driving region into N constrained Delaunay triangles $\mathcal{T} = \{T_1, T_2, \cdots, T_N\}$;
Identify the triangle $S_p \in T_s, 1 \leq s \leq N$, and $F_p \in T_f, 1 \leq f \leq N, f \neq s$;
Construct an adjacent matrix $\mathbf{D}^{N\times N}$ to represent the relationship between any pair of triangles. $\mathbf{D}(i,j) = 1$ represents that triangle T_i and T_j are adjacent. Otherwise, $\mathbf{D}(i,j) = 0$
Start with T_s, based on the matrix $\mathbf{D}$, search for the triangle which is adjacent to T_s, repeat this process until we find T_f.
Based on the driving mode, calculate the intersection between a set of triangles and road boundaries.

$$U^l(x^{h,p}_{pred}) = \begin{cases} 0, & r_b > d_c + \frac{v_w}{2} \\ \alpha_f \exp\left(-\frac{r_b^2}{\sigma_y^2}\right), & Otherwise \end{cases} \tag{2}$$

where r_b is the minimum distance from the vehicle's predicted position to the *lower* bound or the *upper* bound of the safe space. v_w is the vehicle's width, and d_c is the safety margin. α_f is the coefficient for adjusting the magnitude of the potential field. σ_y stands for the convergence coefficient along the direction of Y. Eq. (2) indicates that when the vehicle's predicted position keeps a distance away from the boundaries, the repulsive potential will be around zero. And the value of the repulsive potential is negatively correlated to the distance. Here, we set a threshold, denoted by $U^{\bar{l}}$, for the lateral repulsive potential. If $U^l(x^{h,p}_{pred})$ is greater than $U^{\bar{l}}$, the vehicle's motion under the driver's steering input is considered risky, and a modification of $x^{h,p}_{pred}$ is needed. The repulsive force enforces the *point* $x^{h,p}_{pred}$ to move away from the boundary. The corresponding repulsive force can be given by the negative gradient of Eq. (2):

$$\vec{F} = -\bigtriangledown U^l = \begin{cases} 0, & r_b > d_c + \frac{v_w}{2} \\ \frac{2\alpha_f r_b}{\sigma_y^2} \exp\left(-\frac{r_b^2}{\sigma_y^2}\right), & Otherwise \end{cases} \tag{3}$$

Algorithm 2 describes the gradient descent procedure used in the generation of $x^{a,p}_{goal}$ for the HM-RRT algorithm. $x^{a,p}_{goal}$ is the goal position of *automation* in each replanning cycle and λ is a small incremental distance. To avoid frequent oscillations in the trajectory and to ensure the smoothness of the planned trajectory, the curvature of $x^{a,p}_{goal}$ is constrained as **Judge**$(\kappa^g) = |\kappa^g(t)| < \kappa^g_{max}$ (Line 2). The calculation of κ^g will be introduced in Section IV-A.

In the longitudinal direction, the Dynamic potential field (DPF) method is used to evaluate the *ego*'s motion risk with consideration of the relative position and velocity between the *ego* vehicle and obstacle vehicles.

Algorithm 2: GenerateGoal()

$x_{goal}^{a,p} \leftarrow x_{pred}^{h,p}$
while $U^l(x_{goal}^{a,p}) > U^{\bar{l}}$& **Judge**$(\kappa^g) \geq \kappa_{max}^g$ **do**
  $\vec{F} \leftarrow$ **Potential_Gradient** $(x_{goal}^{a,p})$;
  $x_{goal}^{a,p} \leftarrow x_{goal}^{a,p} + \lambda(\frac{\vec{F}}{|\vec{F}|})$;
end

$$
\begin{aligned}
U^d(\Delta v_{eo}, \Delta x_{oe}) &= U_{eo} \cdot U_{rd} \\
U_{eo} &= \alpha \frac{\exp[\eta(\Delta v_{eo})\cos(\theta) - I]}{2\pi I_0[\eta(\Delta v_{eo})]} \\
U_{rd} &= A_f \cdot \exp\left[-\frac{(\Delta x_{oe}^{\oplus})^{2b}}{2\sigma_x^{2b}}\right] \\
\eta(\Delta v_{eo}) &= sign(\Delta v_{eo})|\Delta v_{eo}|
\end{aligned} \tag{4}
$$

where U_{eo} stands for the potential field related to the relative velocity $\Delta v_{eo} = x_{e,pred}^{h,v_x} - x_{o,pred}^{h,v_x}, o \in \{p, l, f\}$. p, l and f denote the *proceeding*, *leader* and *follower*, respectively. $x_{i,pred}^{h,v_x}, i \in \{e, p, l, f\}$ denotes the corresponding predicted velocity. U_{rd} represents the potential field associated with the longitudinal relative distance $\Delta x_{oe}^{\oplus} = x_o^{\oplus}(\tau_p) - x_e^{\oplus}(\tau_p), o \in \{p, l, f\}$. $x_i^{\oplus}(\tau_p), i \in \{e, p, l, f\}$ denotes the predicted longitudinal position. The calculation of $x_{i,pred}^{h,v_x}$ and $x_i^{\oplus}(\tau_p), i \in \{e, f, l, p\}$ have been described in Eq. (1). θ is the relative heading angle between two vehicles, and $\theta = \pi \times (1 + \min(-sign(x_e - x_o), 0))$. α is the maximum value of the potential field of U_{eo}. A_f is the maximum value of the potential field of U_{rd}. σ_x stands for the convergence coefficient along the direction of X. b is the coefficient for changing the shape. $I_0(\cdot)$ is the modified Bessel function of order 0. By introducing U_{eo}, the potential field value can be drifted to the direction Δv_{eo}.

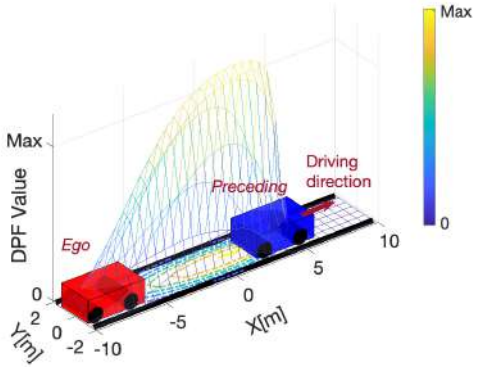


Fig. 3: Schematic diagram of the dynamic potential field designed based on the relative velocity and distance between vehicles.

Fig. 3 gives an example of the potential energy distribution based on the relative distance and velocity difference in a lane keeping task. When the *ego* moves faster than the *preceding*, the potential field value is significantly increased with the decrease of the relative distance. To quantitatively evaluate the risk of the *ego*'s motion, we define the *maximum* DPF with respect to the potential field in the condition of a minimum safe distance, as shown below:

$$
\begin{aligned}
\bar{U}_{eo}^d &= \begin{cases} I, & U_{eo}^d \leq 0 \\ U^d(\Delta v_{eo}, sign(x_{oe}^{\oplus}) d_{eo}^s), & Otherwise \end{cases} \\
d_{eo}^s &= \frac{|(x_{e,pred}^{h,v_x})^2 - (x_{o,pred}^{h,v_x})^2|}{2\bar{a}} \\
&\quad + \max(x_{o,pred}^{h,v_x}, x_{e,pred}^{h,v_x})(t_r + \frac{t_i}{2}) + d_o \\
&\quad o \in \{f, l, p\}.
\end{aligned} \tag{5}
$$

where d_{eo}^s represents the minimum safe distance. $\bar{a}$ is the maximum deceleration for all vehicles. t_r is the reaction time of a driver, and t_i is the build-up time of vehicle deceleration. d_o is the minimum clearance distance. If the longitudinal motion is evaluated as risky and unacceptable, then a support from the *automation* in longitudinal direction will be needed. It should be noted that the calculation of DPF depends on the different driving tasks, and this will be further explained in the next subsection.

D. The trigger mechanism of the automation

The *automation*'s control input is firstly defined as $u_a = [a_a, \delta_a] \in U_{domain} = A_{domain} \times \Delta_{domain}$. A_{domain} is the set of the assisted lateral acceleration, while Δ_{domain} is the set of assisted steering wheel angle from automation. Let n_a and n_δ be given positive integers, and they specify the numbers of feasible acceleration and steering angle per interval, respectively. We consider the following class of control inputs:

IF driving task is $'lane\ keeping'$, Then,

$$
A_{domain} = \begin{cases} \{a_a | \frac{i_a \bar{a}}{n_a}, i_a = 1, \cdots, n_a\}, & \text{if } U_{ep}^d \geq \bar{U}_{ep}^d \\ \{0\}, & Otherwise. \end{cases}
$$

IF driving task is $'lane\ changing'$, Then,

$$
A_{domain} = \begin{cases} \{a_a | \frac{i_a \times \bar{a}}{n_a}, i_a = 1, \cdots, n_a\}, \text{if } U_{ef}^d \geq \bar{U}_{ef}^d \& U_{el}^d < \bar{U}_{el}^d \\ \{a_a | \frac{i_a \times \underline{a}}{n_a} + \underline{a}, , i_a = 1, \cdots, n_a\}, \text{if } U_{ef}^d < \bar{U}_{ef}^d \& U_{el}^d \geq \bar{U}_{el}^d \\ \{0\}, Otherwise. \end{cases} \tag{6}
$$

and

$$
\Delta_{domain} = \begin{cases} \{\delta_a | \frac{i_\delta \times (\bar{\delta} - \underline{\delta})}{M}, i_\delta = 1, \cdots, n_\delta\}, & \text{if } x_{goal}^{a,p} \neq x_{pred}^{h,p} \\ \{0\}, & Otherwise. \end{cases} \tag{7}
$$

where a_a and δ_a are the acceleration and steering angle provided by *automation*, respectively. $\bar{a}$ is the minimum value of a. In Eq. (6), the calculation of the DPF is based on the driving task. More specifically, the obstacle vehicle is the *proceeding* in the

lane keeping maneuver, while the *follower* and the *leader* in the adjacent lane are considered as the obstacle vehicles in the lane changing maneuver. In Eq. (7), $\bar{\delta}$ and $\underline{\delta}$ denote the saturation levels of the front wheel angle δ_a. It should be noticed that if $U_{domain} = (0, 0)$, the *human driver*'s driving behavior is acceptable with low risk, and no support from the *automation* in trajectory planning and tracking is needed. Then the vehicle can be fully controlled by *human driver* in this replanning cycle, i.e.,

$$\text{Occurrence\_Auto} = \begin{cases} 0, & \text{if } U_{domain} = (0, 0) \\ 1, & Otherwise. \end{cases} \tag{8}$$

where Occurrence_Auto indicates the *occurrence* of the support given by *automation*. In the case of $\text{Occurrence\_Auto} = 1$, *automation* is required to be triggered and to collaborate with *human driver* in trajectory planning and tracking. The HM-RRT algorithm is then activated and used to generate the modified trajectory cooperatively planned by the driver and the automation.

IV. HM-RRT-BASED COOPERATIVE TRAJECTORY PLANNING AND TRACKING

In this section, the newly developed HM-RRT-based cooperative trajectory planning and tracking algorithm will be introduced in detail. First, we extend the HM-RRT algorithm by considering the kinematic and dynamic behaviors of the vehicle. In this way, there is no need to design a tracking controller as the planning and execution are interleaved. Then, the *tree trimming* procedure and its usage in trajectory replanning under a dynamic environment will be discussed. The tree trimming-and-regrowing procedure is expected to highly improve the computation efficiency.

Notation: The index h and a represent the *human driver* and *automation*, respectively. The superscripts $(\cdot)^p$ and $(\cdot)^s$ denote the position and vehicle's status, respectively. The symbols $\bar{x}$ and $\underline{x}$ represent the *upper* bound and *lower* bound of the variable x. The variables which are used in the HM-RRT algorithm are summarized in TABLE I.

TABLE I: Variables used in HM-RRT algorithm

Variable	Parameter
x_{rand}	the random node in the configuration space
x_{near}	the nearest node of the tree to the x_{rand}
$\mathbf{x}_{cand}$	a set of new node candidates
$\mathbf{x}_{Rcand} \subset x_{cand}$	the remaining node candidates after selection
x_{new}	the new node added to the tree

A. HM-RRT algorithm

In this work, we develop the novel HM-RRT, a real-time algorithm which incorporates the constrained Delaunay triangulation and artifical potential field into the existing RRT. Algorithm 3 illustrates the implementation of the HM-RRT algorithm, and the main procedures of the algorithm are discussed as follows.

Algorithm 3: GrowHMRRT()

Inputs : $x_{init}, \Delta t, U_{domain}, C, x^{h,p}_{pred}, \mathcal{T} = (V, E)$

$x^{a,p}_{goad} \leftarrow$ **GenerateGoal**$(C, x^{h,p}_{pred})$
while $\left| x^{a,p}_{new} - x^{a,p}_{goal} \right| > d$ **do**
  $x^{a,p}_{rand} \leftarrow$ **RandomState**$(C, x^{a,p}_{goal})$
  $x^{a}_{near} \leftarrow$ **NearestNeighbor**$(x^{a,p}_{rand}, \mathcal{T})$
  $x^{a}_{Cnew} \leftarrow$ **GenerateNew**$(x^{a}_{near}, U_{domain}, \Delta t)$
  $x^{a}_{CCnew} \leftarrow$ **JudgeNew**$(x^{a}_{Cnew}, C, \kappa)$
  $x^{a}_{new} \leftarrow$ **OptimalNew**$(x^{a}_{CCnew}, x^{a,p}_{goal})$
  $u^{a}_{new} \leftarrow$ **IdentifyControl**$(x^{a}_{new}, x^{a}_{CCnew})$
  $\mathcal{T} \leftarrow add\_vertex(x^{a}_{new}, u^{a}_{new}, \mathcal{T})$
  $\mathcal{T} \leftarrow add\_edge(x^{a}_{near}, x^{a}_{new}, u^{a}_{new}, \mathcal{T})$
end
return $\mathcal{T}$.

1) x_{init}: The state space is represented by the set $X \in \mathbb{R}^n$, $n \in \mathbb{N}$. $x \in X$ is a particular configuration of the *ego* and $x = [x^{a,p}\ x^{a,s}]$. Superscript a is the abbreviation of *automation* and s represents vehicle states, including the heading angle ψ, yaw rate r, longitudinal velocity v_x and lateral velocity v_y. A tree $\mathcal{T} = (V, E)$ constitutes of a set of vertices $V \subset X$ sampled from configuration space and edges E that connect these vertices together.

*2) **Sampling**:* The function **RandomState()** returns independently and identically distribute samples from the triangulation T_i ($\in C = \{T_s, \cdots, T_f\}$) to which $x^{a,p}_{goal}$ belongs. In this way, the searching speed and convergence rate are effectively improved.

*3) **Nearest Neighbor**:* The function **NearestNeighbor()** returns the node $v \in V$ in $\mathcal{T}$, and that is the nearest to the configuration $x^{a,p}_{rand} \in X$ in terms of the Euclidean distance. It should be noticed that $x^{a,s}_{rand} := \emptyset$.

*4) **Generate new node candidates**:* The conventional RRT is to choose the control that pulls the new node toward the random node. However, in our proposed new method, considering the vehicle dynamics, **GenerateNew()** returns a sequence of new node candidates ($x^{a}_{cand,i},\ i = 1, \cdots, n_e, n_e = n_a \times n_\delta$), which are generated by trying all possible control inputs $(a_a, \delta_a) \in U_{domain}$. The state variables of the vehicle model are $[x^{\oplus}, y^{\oplus}, \psi, r, v_x, v_y]$. The dynamics of the vehicle can be modeled as:

$$\begin{aligned} \dot{v}_x &= r v_y + a_a + a_h, \\ \dot{v}_y &= -r v_x + \frac{2}{m}(F_{cf}\cos(\delta_a + \delta_h) + F_{cr}), \\ \dot{r} &= \frac{2}{I_z}(l_f F_{cf} - l_r F_{cr}), \\ \dot{x}^{\oplus} &= v_x \cos\psi - v_y \sin\psi, \\ \dot{y}^{\oplus} &= v_x \sin\psi + v_y \cos\psi. \end{aligned} \tag{9}$$

where m and I_z denote the vehicle's mass and yaw inertia, respectively. l_f and l_r represent the distance from the CM of

the vehicle to the front and rear axles, respectively. R_w is the radius of the wheel. $F_{cm} = -C_{\alpha m}\alpha_m, (m \in \{f, r\})$ denote the lateral tyre forces at the front and rear wheels, respectively. $C_{\alpha m}(m = f, r)$ denote the stiffness coefficients pf the front and rear tires. $\alpha_f \approx \frac{v_y + l_f r}{v_x} - \delta_f$ and $\alpha_r \approx \frac{v_y - l_r r}{v_x}$ denote the slip angle of the front and rear tires, respectively. For simplification, the vehicle dynamic model can be rewritten as a compact form: $x(k+1) = \textbf{VehicleControl}(x(k), u_a(k), u_h(k))$.

*5) **Feasibility judgement**:* The state distributions of the new node candidates must be dynamically feasible. Here, we consider the kinematic and dynamic behaviors of the vehicle. It can limit the search space via constraints but still maintain the richness of the feasible solutions. Therefore, we design the function **JudgeNew**() to remove the undesired nodes from the set of new nodes. To ensure the stability of the vehicle, we take the yaw rate into account [5]:

$$|x^r_{cand,i}| \leq \frac{\mu g}{v_x}, i = 1, \cdots, n_e \tag{10}$$

where, μ is the constant coefficient of the road adhesion, and g is the gravitational constant. In addition, the new node should be within the safe space identified, i.e.,

$$x^{a,p}_{cand,i} \subset C, i = 1, \cdots, n_e \tag{11}$$

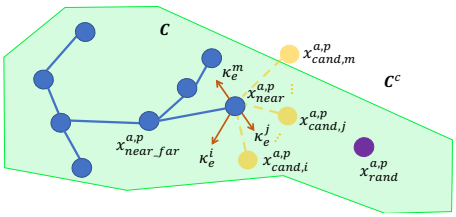


Fig. 4: The procedure of removing bad nodes of x_{cand}

Besides, we use curvature to remove undesired nodes to further smooth the path. As shown in Fig. 4, there are 3 new candidate nodes: $x^{a,p}_{cand,i}$, $x^{a,p}_{cand,j}$ and $x^{a,p}_{cand,m}$ connecting $x^{a,p}_{near}$. $x^{a,p}_{near\_far}$ is the parent node of $x^{a,p}_{near}$. As $x^{a,p}_{cand,m}$ is out of the safe space C, it should be discarded according to Eq. (11). To ensure the smoothness of the steering maneuver, we calculate the curvature of the node (κ_e) of $(x^{a,p}_{near\_far}, x^{a,p}_{near}, x^{a,p}_{cand,n_e})$. If κ^i_e is larger than the threshold, the generated new node will be discarded (e.g. $x^{a,p}_{cand,i}$). More details will be described as follows.

The calculation of $\kappa^i_e, i = 1, \cdots, n_e$ is given by:

$$\kappa^i_e := \frac{x''y' - x'y''}{((x')^2 + (y')^2)^{3/2}}, \ i = 1, \cdots, n_e \tag{12}$$

where $(x, y) = (x^{a,p}_{near\_far}, x^{a,p}_{near}, x^{a,p}_{cand,i})$. x_{near_far} is the parent of x_{near} in the tree. The details on the calculation of curvature can be found in [37]. The curvature κ_e should satisfy the following conditions:

$$|\kappa^i_e| \leq \min\{\kappa^{turn}_{max}, \kappa^{accel}_{max}\}, \text{ for } i = 1, \cdots, n_e. \tag{13}$$

where κ^{turn}_{max} and κ^{accel}_{max} can be calculated by

$$\begin{aligned} \kappa^{turn}_{max} &= \frac{1}{\sqrt{l^2_r + l^2 \cot^2 \delta}}; \\ \kappa^{accel}_{max} &= \frac{a^{lat}_{max}}{v^2_x} \end{aligned} \tag{14}$$

where, l is the vehicle wheelbase, and a^{lat}_{max} is the maximum lateral acceleration. After all the undesired nodes are discarded, the remaining new nodes constitute $x_{Rcand} = \{x_{Rcand,1}, \cdots, x_{Rcand,n_c}\}, \ 1 \leq n_c \leq n_e$.

*6) **New node selection**:* To select the optimal x_{new}, we design a cost function as follows:

$$c(j) = w_1 \times ||x^{a,p}_{Rcand,j} - x^{a,p}_{rand}|| + w_2(\theta_{near,j} + \theta_{Rcand,j}) \quad j = 1, \cdots, n_c \tag{15}$$

where, $||\cdot||$ is the Euclidean norm. The first term represents the Euclidean distance between $x_{near,j}$ and $x_{Rcand,j}, j = 1, \cdots, n_c$. The second term is related to the orientation change between two consecutive nodes. w_1 and w_2 are the weights on the Euclidean distance and smoothness of the path. $\theta_{near,j}, \theta_{Rcand,j} \in [0, \pi]$ should satisfy the following criteria:

$$\begin{aligned} \theta_{near} &= \frac{x^{a,\tilde{\psi}}_{near} \cdot \overrightarrow{x^{a,p}_{near} x^{a,p}_{Rcand,j}}}{||\overrightarrow{x^{a,p}_{near} x^{a,p}_{Rcand,j}}||}, \\ x^{a,\tilde{\psi}}_{near} &= \left[\cos x^{a,\tilde{\psi}}_{near} \quad \sin x^{a,\tilde{\psi}}_{near}\right], \\ \theta_{Rcand,j} &= \frac{x^{a,\tilde{\psi}}_{Rcand,j} \cdot \overrightarrow{x^{a,p}_{near} x^{a,p}_{Rcand,j}}}{||\overrightarrow{x^{a,p}_{near} x^{a,p}_{Rcand,j}}||}, \\ x^{a,\tilde{\psi}}_{Rcand,j} &= \left[\cos x^{a,\tilde{\psi}}_{Rcand,j} \quad \sin x^{a,\tilde{\psi}}_{Rcand,j}\right] \end{aligned} \tag{16}$$

Based on the above cost funtion, the optimal x_{new} is selected by calculating the cost from each $x_{Rcand,j}, j = 1, \cdots, n_c$ to the random node and the nearest node, which can be described by function **OptimalNew**(). The corresponding control inputs $[a_a, \delta_a] \neq \emptyset$ are obtained by **IdentifyControl**() as shown in the Line 8 of the Algorithm 3. x^a_{new} and u^a_{new} are then added to the tree $\mathcal{T}$.

Remark 4.1: A good property of the proposed HM-RRT is that the tree growth can be strongly biased towards the goal position by considering the kinematic and dynamic behaviors of the vehicle. It should be noted that planning and execution are interleaved. At each replanning cycle, the execution phase consists of the generation of a smooth trajectory and tracking of the generated trajectory.

B. Dynamic replanning based on HM-RRT

As described above, during each replanning cycle, if the automation assistance is needed, the *automation* creates a tree, executing the returned path for one step. Then a new tree is repeatedly generated at the next replanning cycle until the driving task is finished. To quickly find the feasible and smooth

path, we let the new tree be an extension based on the previous tree. The new tree can reuse the information from the previous planning episode. By doing so, the computational burden can be significantly reduced.

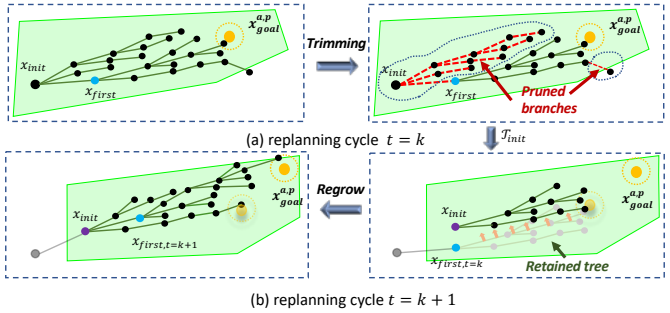


Fig. 5: The tree trimming procedure.

1) Tree Trimming: For a certain replanning cycle, a path is generated by the HM-RRT (shown as the black solid line in Fig. 5a), and the waypoints of the path are represented as $\sigma = \{x_{init}, x_{first}, x_{second}, \cdots, x_{goal}\}$. The corresponding control inputs are $\nu = \{u_{init}, u_{first}, u_{second}, \cdots, u_{goal}\}$. By implementing control input u_{first} of the *automation* and considering human driver's input u_h, the vehicle moves towards $x^{a,p}_{first}$ (shown as the blue point in Fig. 5a). The replanning occurs again, and x^a_{first} is considered as x^a_{init} (Fig. 5b). Typically, the current tree is abandoned, and a new tree is grown from the updated x^a_{init}. This would be very time-consuming and may lead to discotinuity of the trajectory. In this work, we partially keep the current tree for trajectory planning without replanning from scratch. More specifically, we set the childnodes of the previous x^a_{init} (the black point in Fig. 5a) as invalid exception σ and its childnodes. These invalid nodes need to be trimmed (Fig. 5a). x^a_{first} and its childnodes remaining in the tree are set as valid. The corresponding control actions are also stored in the remaining tree. At the next replanning cycle, as $x^a_{init} \neq x^a_{first}$ due to the measurement error and disturbance, the tree at x^a_{first} is copied and pasted to x^a_{init} by the function **RepmatTree**(). The tree can be easily regenerated at x^a_{init} by implementing the stored control actions. Then, the tree at x^a_{init} will grow until the new x^a_{goal} is reached again (Fig. 5d). In this way, a new trajectory can be quickly found with less computing, instead of completely abandoning the old tree and growing a new one. As a result, the proposed approach is much more efficient than replanning from scratch in a dynamic driving environment. The pseudocode for trimming the tree is presented in Algorithm 4.

2) Dynamic replanning: In summary, the pseudocode for human-machine cooperative trajectory planning and tracking is summarized in Algorithm 5. **Determine**() is used to identify whether the driving task is finished or not. In practice, human-machine cooperation can be terminated by manually shutting down the automation assistance system. Alternatively, the cooperation mode can be disengaged once the vehicle's states satisfy some pre-defined conditions. **FeasiblePath**() returns a path which consists of waypoints σ and corresponding control inputs ν. The *ego* can move multiple steps ahead by implementing the control sequence. It should be noted that the number of steps is correlated to the sampling time, the vehicle's initial conditions and the value range of control inputs. In this work, we set the step number as 1. Function **VehicleControl**() describes the execution phase of vehicle motion, which has been introduced in Section IV-A4. **CTRA**() returns the predicted vehicle states based on the driver's behavior, which has been detailed in Section III-A.

Algorithm 4: TrimHMRRT()

Inputs : $\mathcal{T} = (V, E), \sigma$
for *each* $x_i \in V \& x_i \in \{$***Child***$(x_{init}) - x_{first}\}$ **do**
  Mark x_i as INVALID
  $x_j =$ **Child**(x_i)
  while $x_j \neq \emptyset$ **do**
    Mark x_j as INVALID;
    $x_j \leftarrow$ **Child**(x_j);
  end
end
Delete all the INVALID nodes and edges from the tree
return $\mathcal{T}$.

Algorithm 5: Main()/ Cooperative trajectory planning and tracking for autonomous driving

Inputs : Driving_task, $x_{init}, k, \Delta t, u_h$
Initialize: $\mathcal{T}.V \leftarrow \{x_{init}, u_{init}\}$, $\mathcal{T}.E \leftarrow \emptyset$,;
while ***Determine***(Driving_mode)=*true* **do**
  if Occurence_Auto $== 1$ **then**
    $\mathcal{T} \leftarrow$**RepmatTree**($\mathcal{T}$, x_{init})
    $\mathcal{T} \leftarrow$**GrowHMRRT**($\mathcal{T}$, $C, x^{h,p}_{pred}, x_{init}, \Delta t$)
    $[\sigma\ \upsilon] \leftarrow$**FeasiblePath**($\mathcal{T}$)
    $\mathcal{T} \leftarrow$ **TrimHMRRT**$(\sigma, \mathcal{T})$
    $x_{init} \leftarrow$ **VehicleControl**$(x_{init}, v_{first}, u_h)$
  else
    $x_{init} \leftarrow$ **VehicleControl**$(x_{init}, \emptyset, u_h)$
  end
  $U_{domain} \leftarrow$ **ControlDomain**(Driving_mode, x_{init}, x^h_{pred})
  $C \leftarrow$**GenerateSpace**(Driving_mode, x_{init})
  $x^{h,p}_{pred} \leftarrow$ **CTRA**(x_{init}, u_h)
end

V. Testing, Validation and Results

In this section, the superiority of the proposed HM-RRT over the conventional RRT algorithm is first demonstrated through testing in *Case 1*. In *Case 1*, we compare the number of the generated nodes in each replanning cycle by using

different methods. In *Case 2*, the feasibility and effectiveness of the proposed human-machine cooperative trajectory planning approach are further demonstrated under different driving scenarios. Finally, the experimental tests are conducted on a driving simulator to demonstrate the feasibility and real-time implementation capability of the proposed algorithm.

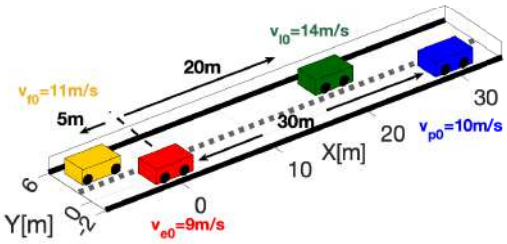


Fig. 6: The driving scenario of Case 1

A. Testing Case 1

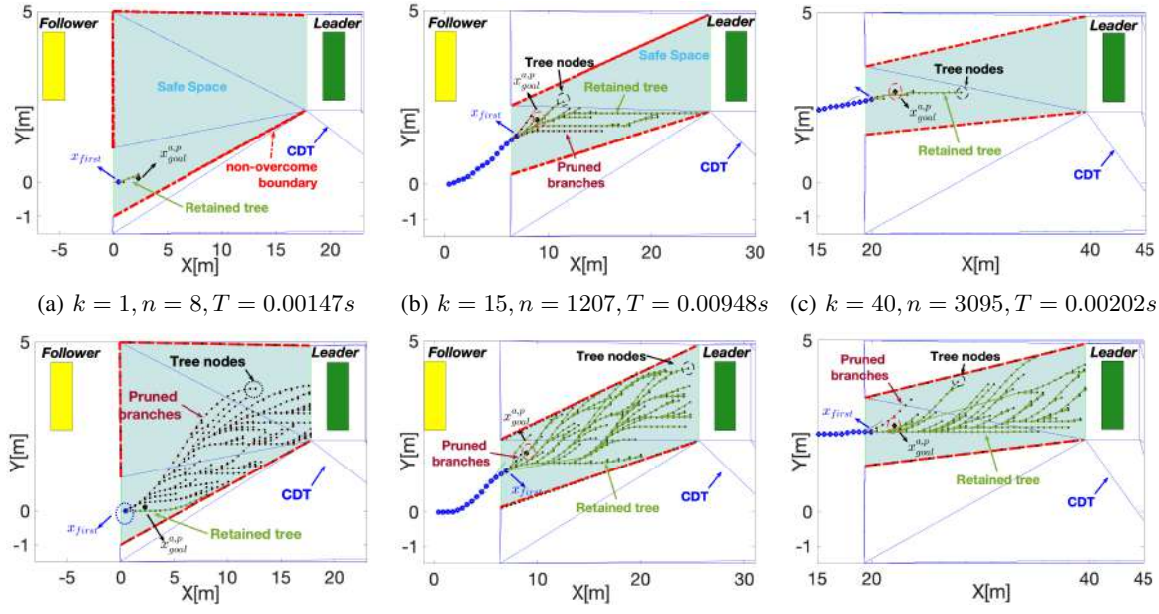
1) Testing scenario: The driving scenario is a road with 2 lanes, as shown in Fig. 6. The *ego* (red vehicle) aims to change the lane and follows the *leader* (green vehicle) in the adjacent lane. The sampling time is $0.01s$ and the initial position of the *ego* is $(0, 0)$. The parameters of the APF model and the DPF model are provided in TABLE II. To better demonstrate the advantage of the *trimming-and-regrow* procedure, we set occurrence_Auto as 1. The driver's behavior and performance are assessed based on a driver-vehicle model, which will be described in the *Case* 2. The constrains of the vehicle plant are: $-20° \times \frac{\pi}{180} \leq \delta_a \leq 20° \times \frac{\pi}{180}$ and $-2\ [m/s^2] \leq a_a \leq 2\ [m/s^2]$.

TABLE II: Potential field parameters

Parameter	Value	Parameter	Value
α_f	30	σ_y	1.1
σ_x	10	b	2
A_f	2	d_0	0.8[m]

2) Testing results: Firstly, we demonstrate the benefits of the proposed method by using **RandomState()** in sampling procedure (as listed in Line 3 of Algorithm 3). Let n be the number of steps executed by the HM-RRT and the traditional RRT. For a fair comparison, the parameters of both algorithms are set as the same. The only difference is that in the conventional RRT, the random sample is selected from the whole safe space (C). The comparison results are shown in Fig. 7. It indicates that by using **RandomState()**, there is a lesser dispersion of the samples in the identified safe space at the same time step. The average computation time of the proposed algorithm is around $0.0043s$, which is less than that of the traditional RRT algorithm ($0.0145s$). Therefore, the proposed HM-RRT is proved to be able to provide a solution with a faster convergence rate and less computational load, comparing to the conventional RRT.

Besides, the results shown in Fig. 8 illustrate the trajectory planning in replanning cycle from $k = 19$ to $k = 22$, which reveals the benefits offered by the *trimming-and-regrow*

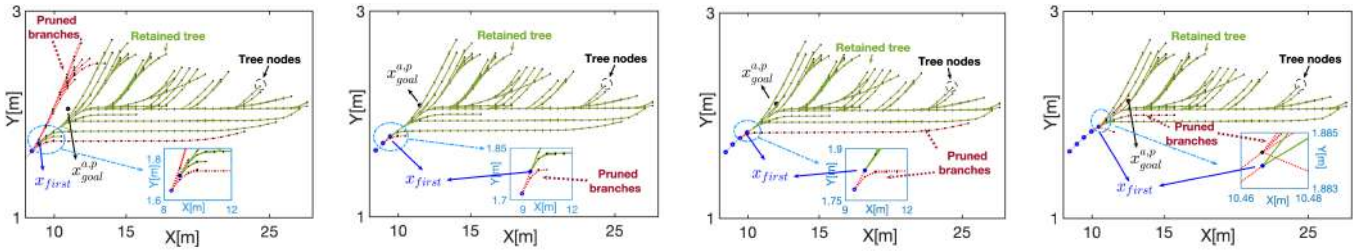


(a) $k = 1, n = 8, T = 0.00147s$ (b) $k = 15, n = 1207, T = 0.00948s$ (c) $k = 40, n = 3095, T = 0.00202s$

(d) $k = 1, n = 500, T = 0.01702s$ (e) $k = 15, n = 1892, T = 0.01412s$ (f) $k = 40, n = 3476, T = 0.01244s$

Fig. 7: The performance of the HM-RRT (a-c) and traditional RRT (d-f) on trajectory planning. (k: the time step, n: the number of execution steps

).



(a) $k = 19, n = 1785, w = 499$ (b) $k = 20, n = 1785, w = 408$ (c) $k = 21, n = 1785, w = 403$ (d) $k = 22, n = 1785, , w = 367$

Fig. 8: The example trimming and regrowing procedures of the HM-RRT (k: the time step, n: the number of execution steps, w: the number of the nodes in the tree)

.

procedure of the Algorithm 4. w denotes the number of nodes in the tree. It shows that n remains the same in these replanning cycles, which means that no new node is generated in these successive steps. In each step, a new tree is obtained only by trimming the unnecessary branches from the tree generated in the previous replanning cycle. The computation load is therefore significantly reduced.

B. Testing Case 2

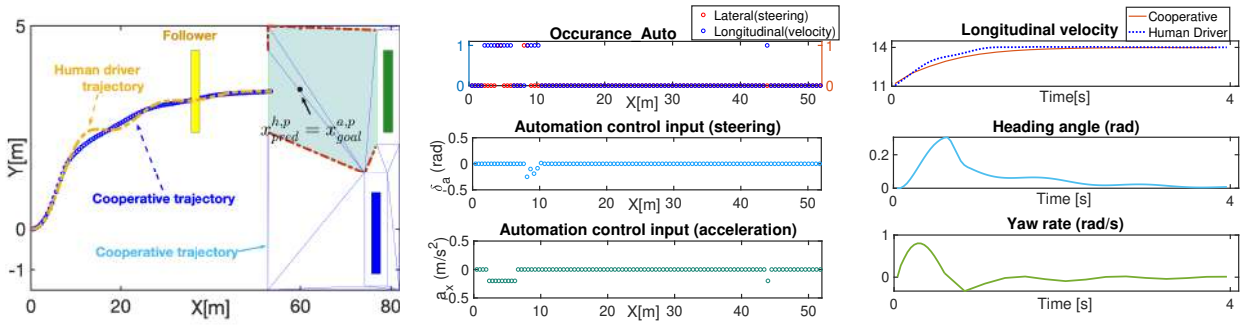
In Case 2, the feasibility and effectiveness of the proposed cooperative trajectory planning algorithm are extremely tested in the different driving scenarios. We compare the proposed human-machine cooperation method with the manual control approach, in which only human drivers perform the driving task.

1) Driving scenario 1: The driving course is designed as a straight road in an urban scenario with two lanes. The human driver can be modelled as an optimization controller and the lateral & longitudinal control behaviors can be described as:

$$\begin{aligned}\dot{\delta}_d =& -\frac{1}{T_h}\delta_d + \frac{R_g G_h}{T_h}(y^{\oplus}_{target} - y^{\oplus}) \\ &+ \frac{R_g G_h \tau_h}{T_h}(y^{\oplus}_{target} - v_x\psi - v_y) \\ a_a =& K_p(v_x - v_{target}) + K_d(\dot{v}_x - \dot{v}_{target}).\end{aligned} \tag{17}$$

where $y^{\oplus}_{target}$ is the target lateral position of the vehicle, and $y^{\oplus}$ is the current vehicle's lateral position. R_g is the gear ratio of the steering system. G_h is the steering proportional gain, T_h is time delay of the driver's response, while τ_h is a derivative time constant of the driver's steering. v_{target} is the target velocity of the vehicle, and it is set equal to the *leader*'s velocity. K_p and K_d are non-negative and denote the coefficients of the proportional and derivative terms, respectively. The identification of the human driver parameters K_p, K_d, T_h, G_h and τ_h can be obtained by experiments and the details are given in Appendix A.

Fig. 9 shows an example result of the comparison between the two control approaches with a human participant. As observed in Fig. 9b, under the proposed approach, the *human driver* holds the entire control authority and performs his expected trajectory most of the time. The *automation* is only triggered only when human driving is assessed as risky, and it collaborates with

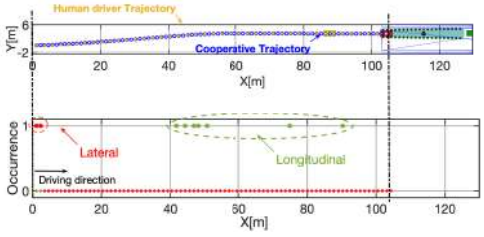


(a) The cooperative trajectory planning (https://youtu.be/kPxHdDdDNpA) (b) Trigger of the *automation* and its control actions (c) Vehicle states

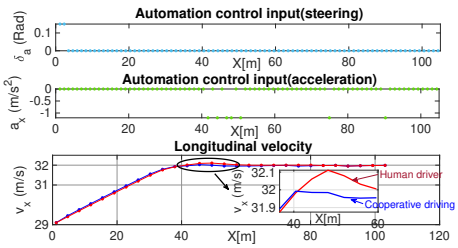
Fig. 9: Testing results in the driving scenario 1

human driver on trajectory planning and tracking control by executing a_a and δ_a. In this way, safety and ride comfort are guaranteed, while the human-machine conflicts are effectively reduced. As a result, the cooperatively planned trajectory is much smoother than that of manual driving, which can be reflected from the variation of the vehicle's velocity (Fig. 9a). In addition, the stability and handling performance of the vehicle is also guaranteed, as shown in Fig. 9c.

2) Driving scenario 2: The driving course is designed as a highway, where all the vehicles are with high velocities. Fig. 10a shows the cooperatively planned trajectory (marked as a blue circle) and the manual trajectory (marked as a yellow circle). It can be found that the automation assists the human driver in the lateral direction at the very early stage and in the longitudinal direction during $X = 40 - 50[m]$, $X = 78[m]$ and $X = 90[m]$. The occurrence of lateral assistance is because the human driver sharply steers the hand wheel at the beginning of the lane-change process. The lateral driving risk is then increased, triggering the intervention of the automation. The longitudinal assistance is to prevent the vehicle's velocity from exceeding the reference value v_{target}.



(a) The cooperative trajectory planning



(b) Control actions of the automation

Fig. 10: Testing results of the driving scenario 2 (*https://www.youtube.com/watch?v=QuFoY3YSvIs*).

Fig. 10b shows that the longitudinal velocity exceeds the reference velocity ($v_{target} = 32\ m/s$) after $X = 40[m]$, and the automation starts providing control actions (a_x) to the vehicle. Thus, the velocity is decreased afterwards. In contrast, the velocity of manual driving is continuously increased after $X = 40[m]$. It shows that the automation can successfully ensure the vehicle's safety during the execution of the driving task. In addition, the proposed algorithm shows strong robustness to the highly dynamic driving situation.

3) Driving scenario 3: In this driving scenario, we consider a corner case, that the *follower* is aggressive and accelerates to prevent the *ego* from changing the lane. In this case, the *ego* will stop changing the lane and execute lane-keeping, and it will restart lane-change after the feasibility of a lane-change is confirmed again. It should be noted that the *follower* vehicle will be the new *leader* after restarting the lane-change maneuver.

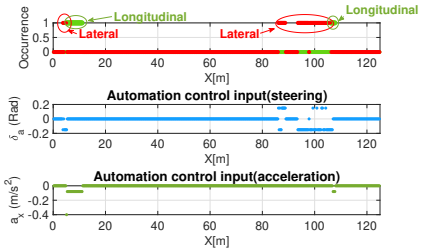


Fig. 12: The control actions of the automation in the driving scenario 3.

Fig. 11 shows the entire cooperative path, which includes lane-change, lane-keeping, and re-lane changing. In the beginning, the *ego* tends to change the lane by operating the steering wheel. As the *follower* accelerates, and the driving risk is continuously increased (between $X = 3[m] \sim 12[m]$ shown in Fig. 12a). The *automation* starts assisting the human driver in both the lateral and longitudinal directions until the human driver quits the lane-change task. The human driver starts driving the vehicle in the current lane and following the *preceding* vehicle. It should be noted that in the lane-keeping maneuver, the y_{target} is the centre line of the current lane, and v_{target} is the velocity of the *preceding*. After $X = 85[m]$ when reaching a safe distance between the *ego* and *follower*, the human driver starts changing the lane. At the same time, the automation is responsible for providing the support if necessary (between $X = 85 \sim 105[m]$ in Fig. 12b). It should be noted that there is a collision between the *ego* and road boundary around $X = 105[m]$. However, with the help of the *automation*,

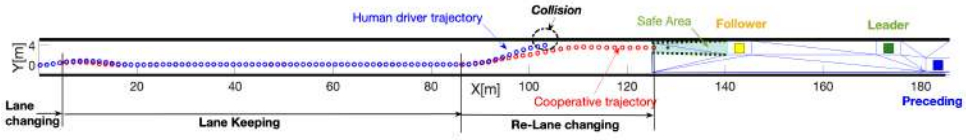
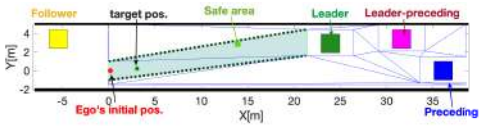


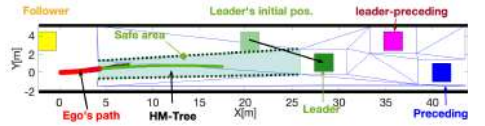
Fig. 11: Testing result in driving scenario 3 (https://www.youtube.com/watch?v=mGNyWPs9dG0)

the *ego* can successfully change the lane without any collision happening in the cooperative driving. The testing results show that the proposed algorithm can also provide a safe path under the emergency driving situation.

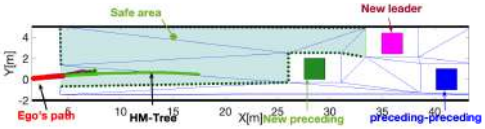
4) Driving scenario 4: As illustrated in the previous driving situation, all the obstacle vehicles are assumed to stay in their current lanes. And a driving scenario, in which the *leader* changes its lane when the *ego* changes its lane, is designed to test the effectiveness of the proposed algorithm in the complex driving scenario.



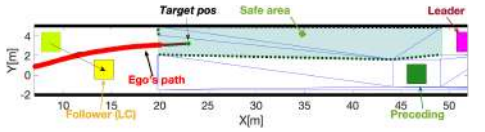
(a) Lane-changing at $k = 1$



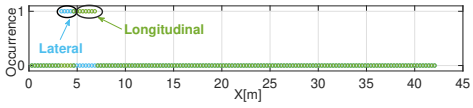
(b) Lane-changing at $k = 14$



(c) Lane-changing at $k = 15$



(d) Lane-changing at $k = 60$



(e) Occurrence of the automation

Fig. 13: Testing results of driving scenario 4 (*https://www.youtube.com/watch?v=DjdIJ4zD1uk*).

Fig. 13 shows the cooperative path generated in the driving scenario 4. It shows that at the beginning, the y_{target} is the longitudinal position of the *leader*, and v_{target} is the *leader*'s velocity. As the values of y_{target} and v_{target} are decreased, the *automation* starts providing extra control effort in order to obtain the desirable driving performance. After the *leader* crosses the lane line, the automation re-defines the *target* position and velocity. The *target* vehicle then becomes from the *leader* to the *leader-preceding*. The *automation* stops providing extra effort afterwards. The testing results indicate that the proposed algorithm can identify the changes in the driving scenario in real time. It is able to recalculate the safe area, continuously estimate the driving risk, and provide support if necessary in the complex driving situation.

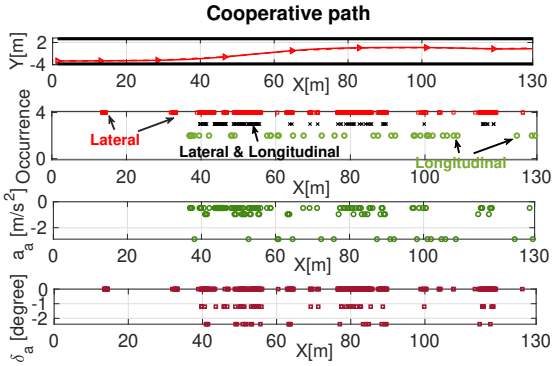


Fig. 15: An example of the cooperative path generated in experimental test (https://www.youtube.com/watch?v=46E5hZP2ETw).

C. Human-in-the-loop Experimental Validation

In this subsection, the effectiveness of the proposed approach is further validated through human-in-the-loop experiments on a driving simulator.

1) Experimental design: The experiments are conducted on a human-in-the-loop driving simulator as shown in Fig. 14a. The platform consists of a computer equipped with an NVIDIA GTX 2080 Super GPU, three joint head-up monitors, a Logitech G29 steering wheel suit, and a driver seat.

In the experiment, the driving course was set to be a straight urban road with two lanes, as shown in Fig. 14c. The participant

(a) (b) (c)

Fig. 14: The experimental platform: (a) driving simulator, (b) experimental participant, (c) driving course

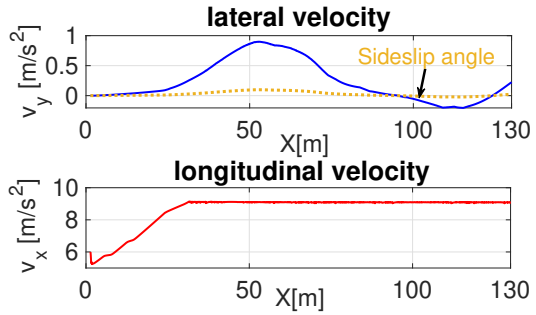


Fig. 16: Lateral velocity & longitudinal velocity

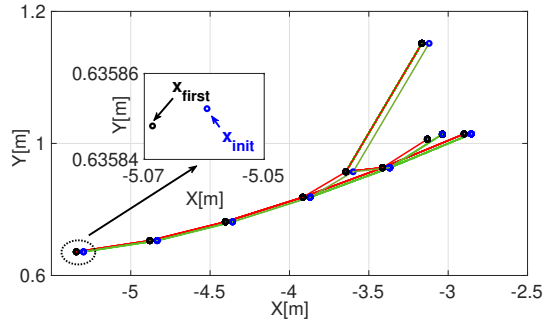


Fig. 17: The tree creation by **RepmatTree**() at $t = 1.13$

was asked to execute a lane-change task when he or she was ready. The sampling time is $0.01s$ and the prediction horizon is $0.5s$. The driver's inputs were measured through the steering wheel angle, brake and acceleration pedal positions. The vehicle's states, including longitudinal and lateral acceleration, velocity, and position, were recorded. The study protocol and consent form were approved by the Nanyang Technological University Institutional Review Board (protocol number IRB-2018-11-025).

2) Experimental results: Fig. 15a shows an example of the generated cooperative path, and it lasts for $21s$. Fig. 15b indicates the occurrence of the *automation* in the driving. The red square indicates the occurrence of the automation's support in the lateral direction, and the black cross indicates the occurrence of the support in lateral & longitudinal directions. The green circle represents the occurrence of the support in the longitudinal direction. The Fig. 15c and Fig. 15d show the corresponding control actions of the *automation*. With the support of the *automation*, the human driver successfully changed the lane. It should be noted that the *automation* supports the human driver for around $6s$ in the driving. It can significantly reduce human-machine conflicts and ensure driving safety and stability simultaneously.

In addition, Fig.16 indicates that with the support of the *automation*, the controlled vehicle's longitudinal velocity can successfully track the reference velocity (v_l) and maintain the lateral velocity in a stable range.

Fig. 17 shows the tree creation by using **RepmatTree**() at $t = 1.13$. It shows that the x_{first} is very close to x_{init}, and the relative distance is $0.0129[m]$. Therefore, the tree trimming-and-regrow procedure can be successfully applied in cooperative driving. In addition, the average computation time is $0.00745s$ for each planning cycle, which means that the proposed algorithm has a good real-time implementation ability, assisting human drivers timely.

VI. Conclusion

This paper investigates a human-machine cooperative trajectory planning and tracking control approach for automated vehicles. First, a novel HM-RRT algorithm is designed for improving the efficiency of path planning. Next, based on the identification of the safe driving area and assessment of the human driving risks, the trigger mechanism of the human-machine cooperation is developed. By using the HM-RRT, cooperative trajectory planning is realized via human-machine collaboration, which incorporates the driver's desire and actions and automation's corrective actions. Experimental validation is conducted on a real-time driving simulator. Results show that the proposed HM-RRT algorithm can effectively improve the convergence rate and reduce the computation load during path planning. Beyond this, the proposed human-machine cooperative trajectory planning approach is able to simultaneously ensure the safety, stability and smoothness of the vehicle and largely reduce human-machine conflicts. Future work will be focusing on the improvement of the proposed algorithms and their implementation on a real vehicle platform.

Acknowledgment

This work was supported in part by in part by A*STAR National Robotics Programme under Grant SERC 1922500046; the Alibaba Group, through the Alibaba Innovative Research Program and the Alibaba–Nanyang Technological Univer-sity Joint Research Institute (grant AN-GC-2020-012); the SUG-NAP under Grant M4082268.050, Nanyang Technological University, Singapore; and the State Key Laboratory of Automotive Safety and Energy under Project (no. KF2021).

References

[1] F. Tang, F. Gao, and Z. Wang, "Driving capability-based transition strategy for cooperative driving: From manual to automatic," *IEEE Access*, 2020.

[2] C. Huang, P. Hang, Z. Hu, and C. Lv, "Collision-probability-aware human-machine cooperative planning for safe automated driving," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2021.

[3] L. Nie, Z. Yin, and H. Huang, "Decision making and trajectory planning of intelligent vehicle's lane-changing behavior on highways under multi-objective constrains," SAE Technical Paper, Tech. Rep., 2020.

[4] J. Zhou, H. Zheng, J. Wang, Y. Wang, B. Zhang, and Q. Shao, "Multi-objective optimization of lane-changing strategy for intelligent vehicles in complex driving environments," *IEEE Transactions on Vehicular Technology*, 2019.

[5] M. Li, H. Cao, X. Song, Y. Huang, J. Wang, and Z. Huang, "Shared control driver assistance system based on driving intention and situation assessment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4982–4994, 2018.
[6] J. C. de Winter and D. Dodou, "Preparing drivers for dangerous situations: A critical reflection on continuous shared control," in *2011 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2011, pp. 1050–1056.
[7] M. Marcano, S. Díaz, J. Pérez, and E. Irigoyen, "A review of shared control for automated vehicles: Theory and applications," *IEEE Transactions on Human-Machine Systems*, 2020.
[8] M. Lu, K. Wevers, and R. Van Der Heijden, "Technical feasibility of advanced driver assistance systems (adas) for road traffic safety," *Transportation Planning and Technology*, vol. 28, no. 3, pp. 167–187, 2005.
[9] M. Kamezaki, H. Hayashi, U. E. Manawadu, and S. Sugano, "Human-centered intervention based on tactical-level input in unscheduled takeover scenarios for highly-automated vehicles," *International Journal of Intelligent Transportation Systems Research*, pp. 1–10, 2019.
[10] K. Bengler, M. Zimmermann, D. Bortot, M. Kienle, and D. Damböck, "Interaction principles for cooperative human-machine systems," *it-Information Technology Methoden und innovative Anwendungen der Informatik und Informationstechnik*, vol. 54, no. 4, pp. 157–164, 2012.
[11] M. Itoh, F. Flemisch, and D. Abbink, "A hierarchical framework to analyze shared control conflicts between human and machine," *IFAC-PapersOnLine*, vol. 49, no. 19, pp. 96–101, 2016.
[12] M. Mulder, S. Kitazaki, S. Hijikata, M. Mulder, M. van Paassen, and E. R. Boer, "Reaction-time task during car-following with an active gas pedal," in *2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583)*, vol. 3. IEEE, 2004, pp. 2465–2470.
[13] C. Sentouh, A.-T. Nguyen, M. A. Benloucif, and J.-C. Popieul, "Driver-automation cooperation oriented approach for shared control of lane keeping assist systems," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 5, pp. 1962–1978, 2018.
[14] D. A. Abbink, M. Mulder, and E. R. Boer, "Haptic shared control: smoothly shifting control authority?" *Cognition, Technology & Work*, vol. 14, no. 1, pp. 19–28, 2012.
[15] C. Passenberg, A. Glaser, and A. Peer, "Exploring the design space of haptic assistants: the assistance policy module," *IEEE Transactions on Haptics*, vol. 6, no. 4, pp. 440–452, 2013.
[16] K. K. Tsoi, M. Mulder, and D. A. Abbink, "Balancing safety and support: Changing lanes with a haptic lane-keeping support system," in *2010 IEEE international conference on systems, man and cybernetics*. IEEE, 2010, pp. 1236–1243.
[17] C. E. Billings, *Aviation automation: The search for a human-centered approach*. CRC Press, 2018.
[18] F. Flemisch, A. Schieben, J. Kelsch, and C. Löper, "Automation spectrum, inner/outer compatibility and other potentially useful human factors concepts for assistance and automation," *Human Factors for assistance and automation*, 2008.
[19] A. Benloucif, A.-T. Nguyen, C. Sentouh, and J.-C. Popieul, "Cooperative trajectory planning for haptic shared control between driver and automation in highway driving," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9846–9857, 2019.
[20] A. W. de Jonge, J. G. Wildenbeest, H. Boessenkool, and D. A. Abbink, "The effect of trial-by-trial adaptation on conflicts in haptic shared control for free-air teleoperation tasks," *IEEE transactions on haptics*, vol. 9, no. 1, pp. 111–120, 2015.
[21] D. Yang, D. Li, and H. Sun, "2D dubins path in environments with obstacle," *Mathematical Problems in Engineering*, vol. 2013, 2013.
[22] X. Qian, I. Navarro, A. de La Fortelle, and F. Moutarde, "Motion planning for urban autonomous driving using bézier curves and MPC," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. Ieee, 2016, pp. 826–833.
[23] A. Goswami, "Trajectory generation for lane-change maneuver of autonomous vehicles," 2015.
[24] T. Maekawa, T. Noda, S. Tamura, T. Ozaki, and K.-i. Machida, "Curvature continuous path generation for autonomous vehicle using b-spline curves," *Computer-Aided Design*, vol. 42, no. 4, pp. 350–359, 2010.
[25] J.-W. Lee and B. Litkouhi, "A unified framework of the automated lane centering/changing control for motion smoothness adaptation," in *2012 15th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2012, pp. 282–287.
[26] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, "Autonomous parking using optimization-based collision avoidance," in *2018 IEEE Conference on Decision and Control (CDC)*. IEEE, 2018, pp. 4327–4332.
[27] C. Huang, B. Li, and M. Kishida, "Model predictive approach to integrated path planning and tracking for autonomous vehicles," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1448–1453.
[28] D. Fassbender, A. Mueller, and H.-J. Wuensche, "Trajectory planning for car-like robots in unknown, unstructured environments," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3630–3635.
[29] M. Parulekar, V. Padte, T. Shah, K. Shroff, and R. Shetty, "Automatic vehicle navigation using dijkstra's algorithm," in *2013 International Conference on Advances in Technology and Engineering (ICATE)*. IEEE, 2013, pp. 1–5.
[30] J. Wang, M. A. Garratt, and S. G. Anavatti, "Real-time path planning algorithm for autonomous vehicles in unknown environments," *International Journal of Mechatronics and Automation*, vol. 6, no. 1, pp. 1–9, 2017.
[31] T. Hesse, D. Hess, and T. Sattel, "Motion planning for passenger vehicles-force field trajectory optimization for automated driving," in *The 15th IASTED International Conference on Robotics and Applications*, 2010.
[32] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," 1998.
[33] H. Liu, X. Li, M. Fan, G. Wu, W. Pedrycz, and P. N. Suganthan, "An autonomous path planning method for unmanned aerial vehicle based on a tangent intersection and target guidance strategy," *arXiv preprint arXiv:2006.04103*, 2020.
[34] C. Huang, H. Huang, P. Hang, H. Gao, J. Wu, Z. Huang, and C. Lv, "Personalized trajectory planning and control of lane-change maneuvers for autonomous driving," *IEEE Transactions on Vehicular Technology*, 2021.
[35] R. Schubert, C. Adam, M. Obst, N. Mattern, V. Leonhardt, and G. Wanielik, "Empirical evaluation of vehicular models for ego motion estimation," in *2011 IEEE intelligent vehicles symposium (IV)*. IEEE, 2011, pp. 534–539.
[36] C. Huang, C. Lv, F. Naghdy, and H. Du, "A reference-free approach for mitigating human-machine conflicts in sharedcontrol of automated vehicles," *IET Control Theory & Applications*, 2020.
[37] Z. Zhu, E. Schmerling, and M. Pavone, "A convex optimization approach to smooth trajectories for motion planning with car-like robots," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 835–842.

APPENDIX

A. Human vehicle model

To identify the driver parameters, a lane-changing experiment with 5 participants (mean age at 26.5) are conducted in a real-time driving simulator, as shown in Fig. 14a. The driving course is designed as a

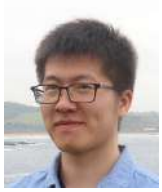
straight road in urban scenario with two lanes. The parameters of the driving scenario are set as the same as those in Fig. 6. The participants are asked to change the lane when they feel necessary. All participants are briefed on the task requirements by an experimental instructor before trials, and they are allowed to practice firstly to get familiarised with the simulator operation. Each participant is required to drive for 30 minutes. The driver's inputs are measured through the steering wheel angle, and brake and accelerator pedal positions. The vehicle's states, including the longitudinal and lateral accelerations, velocity, and position are recorded. The identification results of the driver model with the five participants are given in TABLE III.

TABLE III: Identification results of the driver parameters.

Param.	K_p	K_d	T_h	G_h	τ_h	**Fit** %
Driver 1	-0.72	-0.22	0.26	0.14	1.34	88.94
Driver 2	-0.9	-0.21	0.33	0.14	1.33	80.53
Driver 3	-0.75	-0.22	0.24	0.12	1.14	82.28
Driver 4	-0.79	0.26	0.24	0.10	1.74	74.53
Driver 5	-0.77	0.60	0.50	0.06	1.29	73.76

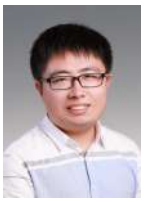
The parameters of **Driver 2** are used in the *Case 1* and *Case 2* to demonstrate the effectiveness of the proposed algorithm.

Chao Huang received the B.Sc. degree in automation, from China University of Petroleum, Beijing, China, in 2012, and received Ph.D degree from the University of Wollongong, Wollongong, NSW, Australia, in 2018. Her current research interests include human-machine shared control and path planning of autonomous vehicles.



Hailong Huang received the B.Sc. degree in automation, from China University of Petroleum, Beijing, China, in 2012, and received Ph.D degree in Systems and Control from the University of New South Wales, Sydney, Australia, in 2018. He was a postdoctoral research fellow at the School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, Australia. He is now an Assistant Professor at the Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University, Hong Kong. His current research interests include guidance, navigation, and control of mobile robots, multi-agent systems, and distributed control.

Junzhi Zhang received the B.E. degree in Transportation Engineering and his M.S. and Ph.D. degree in Vehicle Engineering from the Jiliin University of Technology, Changchun, China in 1992, 1995 and 1997, respectively. From 1998-1999, Dr. Zhang was a Research Associate in Department of Automotive Engineering in Tsinghua University, Beijing, China. In 1999, Dr. Zhang joined Tsinghua University and founded the Hybrid Powertrain Systems Laboratory whose major research interests include modeling, control and diagnosis of hybrid, electric vehicle. Dr. Zhang become a full professor in Department of Automotive Engineering in Tsinghua University in 2008. Dr. Zhang is the author or co-author of more than 50 peer-reviewed publications and 20 patents.



Peng Hang received a Ph.D. degree from the School of Automotive Studies, Tongji University, Shanghai, China, in 2019. He is a research fellow with the School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore. His research interests include vehicle dynamics and control, decision making and motion control of autonomous vehicles.

Zhongxu Hu received a Ph.D. degree in mechatronics in 2018 from Huazhong University of Science & Technology of China. He was a senior engineer at Huawei. He is currently a research fellow within the AUTOMAN group of Nanyang Technological University in Singapore. His current research interests include computer vision and machine learning applied to driver behavior analysis and autonomous vehicles in multiple scenarios.

Chen Lv is currently an assistant professor at Nanyang Technology University, Singapore. He received a Ph.D. degree from the Department of Automotive Engineering, Tsinghua University, China in 2016. His research focuses on advanced vehicle control and intelligence, where he has contributed over 100 papers and obtained 12 granted patents.