

Robust Tunable Trajectory Repairing for Autonomous Vehicles Using Bernstein Basis Polynomials and Path-Speed Decoupling

Kailin Tong¹, Selim Solmaz¹, Martin Horn², Michael Stolz^{1,2} and Daniel Watzenig^{1,2}

Abstract—Adaptation to changing dynamic situations is yet an open problem for automated driving systems that require robust and efficient solutions. Particularly in the context of motion planning algorithms, this problem is typically addressed by re-planning the whole trajectory or repairing the invalid part. The main drawback of all the current approaches is the increased demand for computational resources, a critical safety issue in automated vehicles. Motivated by this, in this paper we propose a novel and efficient method for trajectory repairing utilizing Bernstein basis polynomials and path-speed decoupling. A robustness metric is introduced to tune the driving behavior. Accurate numerical simulations indicate performance figures typically better than 25ms for a feasible solution in representative driving scenarios, which was not achievable in other state-of-the-art approaches.

I. INTRODUCTION

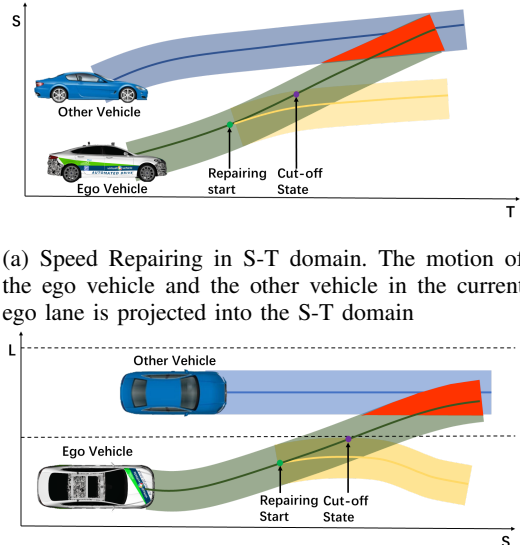
Analysis of recent accident statistics [1] indicates that despite strenuous testing efforts, autonomous vehicles (AVs) still fail to make the right decisions from time to time, potentially leading to property damages or even injuries, particularly in emergency situations. In a dynamic traffic scenario, the behavior of other vehicles might suddenly change and lead to a hazardous situation. From the perspective of an automated driving system, a common way to manage such a situation is to re-plan and update the trajectory from the current state to the target one. This would, however, require searching for alternative trajectories on a continuous basis. A more efficient approach would be first to detect the part of an invalid trajectory that can stay unchanged and then repair only the remaining part of it [2]. The main benefit of this is the elimination of the need to re-plan the whole trajectory continuously, as well as increased robustness against small disturbances.

In this paper, we propose a novel and efficient framework for cut-off state detection and trajectory repairing by exploit-

* This project has received funding from the European GNSS Agency under the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101004181. The publication was written at Virtual Vehicle Research GmbH in Graz and partially funded within the COMET K2 Competence Centers for Excellent Technologies from the Austrian Federal Ministry for Climate Action (BMK), the Austrian Federal Ministry for Digital and Economic Affairs (BMDW), the Province of Styria (Dept. 12) and the Styrian Business Promotion Agency (SFG). The Austrian Research Promotion Agency (FFG) has been authorised for the programme management.

¹ K. Tong, S. Solmaz, M. Stolz and D. Watzenig are with Virtual Vehicle Research GmbH, Inffeldgasse 21a, 8010 Graz, Austria. {kailin.tong, selim.solmaz, michael.stolz, daniel.watzenig}@v2c2.at

² M. Horn, M. Stolz and D. Watzenig are with the Institute of Automation and Control at Graz University of Technology, Inffeldgasse 21b, 8010 Graz, Austria. martin.horn@tugraz.at



(a) Speed Repairing in S-T domain. The motion of the ego vehicle and the other vehicle in the current ego lane is projected into the S-T domain

(b) Path Repairing in L-S domain. The motion of the ego vehicle and the other vehicle is projected into the L-S domain in curvilinear coordinates.

Fig. 1: Trajectory repairing utilizing path-speed decoupling.

ing the property of Bernstein Basis Polynomials and Path-Speed Decoupling. Compared to the existing literature, the contributions that we report in this paper are summarized as follows:

- Formulation of a general convex optimization problem using Bernstein Polynomials for both path and speed repairing, considering kinematic constraints. The algorithm extends speed profile optimization [3] into path optimization using a same problem formulation but with kinematic path constraints. Our implementation is based on Python but is already real-time capable.
- Proposing an efficient and real-time capable scheme to identify the critical time/distance to react and repair the speed profile and the path with safety assurance.
- The first formal definition of a robustness metric (α) for a fail-safe motion planner. It is a parameter that can balance trajectory re-planning and repairing, as well as comfort and robustness, which enables high-level tuning of automated driving behavior.

II. RELATED WORK

A. Computation of Time-To-X

One commonly used safety metric is Time-To-X (TTX), where X refers to the corresponding reaction in the collision

path. For example, Time-to-Collision (TTC) measures the time to collision and guides whether the AD (Automated Driving) system should send a warning to the driver or intervene directly [4]. Other metrics in this family are as follows: TTB (Time-To-Brake) indicates the time to braking with maximum deceleration; TTK (Time-To-Kickdown) indicates the time to reach the maximum velocity with full acceleration; TTS (Time-To-Steer) indicates the time to fully steer to the left or right with the maximum steering angle. Furthermore, the Time-To-React (TTR) has been proposed as a worst-case metric taking account of all Time-To-X metrics mentioned above [5]. The concept of TTX can be extended in the spatial axes, and define metrics such as DTB (Distance-To-Brake), DTS (Distance-To-Steer), DTC (Distance-To-Collision), DTM (Distance-To-Maneuver), and DTR (Distance-To-React) [6].

Generally, there are two ways to calculate TTX online: using empirical estimates or by forward simulation. Schratter et al. [7] use an empirical formula based on the current ego state and surrounding states to estimate TTB and TTS, and, finally, estimate the collision risk for decision-making of an emergency maneuver. Their proposed collision avoidance system can handle the pedestrian-crossing scenario in an occluded area. However, extending this approach to more general critical scenarios is not straightforward.

In the scope of forward simulation for TTX calculation, reachable set analysis has been applied in the literature for searching for TTR [8]. However, this provides an over-approximation of the true TTR. In [2], [9], TTX is obtained by a modified binary search using realistic emergency maneuver models. Therefore, it can compute TTX values with defined accuracy and handle scenarios with multiple static and dynamic obstacles. However, they give the same importance to longitudinal emergency maneuvers (TTB, TTK) and lateral ones (TTS), which is counterintuitive.

As a commonsense behavior in a very typical driving situation, e.g., when facing an emergency on the ego lane, the driver preferably attempts to adapt the speed, if not feasible, then tries to steer the car to avoid an accident. In this work, we use a hierarchical search scheme mimicking this behavior for calculating TTR (or DTR) to improve search efficiency.

B. Planning schemes

Unlike other literature, which classifies planning algorithms by their problem formulations [10], we classify the planning scheme into two groups: re-planning and repairing.

1) *Re-planning*: When an agent navigates in a physical world, the agent’s action shall depend on information gathered during execution, termed as Feedback Motion Planning in [11], or more simply re-planning. The re-planning scheme has been commonly adopted in Autonomous Driving Software Stacks, such as Baidu Apollo [12]. After a re-planning, graph search-based planners and sampling-based planners have the chance to obtain a global optimal result (assuming no time limit); however, the newly planned trajectory might diverge from the previous quite a lot. As a result, the trajectory tracking might be unstable. On the contrary, numerical

optimization approaches rely on the previous planning result, and the newly planned trajectory still follows the original one, but the result is only local optimum. To relieve the effort of re-planning, Apollo EM (Expectation and Maximization) planner proposes a path-speed decoupled iterative optimization scheme [12]. Our work is also motivated by the EM-type iterative algorithm.

2) *Repairing*: Unlike re-planning, repairing means that only the necessary part of a reference trajectory is changed due to environmental disturbances. This concept has been widely used in the robotics domain as well as for UAVs (Unmanned Aerial Vehicles), and UGVs (Unmanned Ground Vehicles), in the form of local re-planning [13] and trajectory deformation [14]. However, they are not explicitly intended for AVs and do not necessarily provide safety assurance. The point of “repairing” depends strictly on the optimization setup. Most recently, Lin et al. proposed a sampling-based trajectory repairing algorithm using closed-loop rapidly-exploring random trees (CL-RRT) and developed a safety assurance scheme for the repaired evasive maneuver. However, the sampling-based approach relies on randomness, which is computationally expensive in some scenarios as it is not easy to sample nodes in a “narrow passage”, which is a typical problem of sampling-based planners.

III. PRELIMINARIES

A. Vehicle Model and Configuration Space

We use a kinematic bicycle model [11] in this work as shown in Fig. 2. A two-wheel bicycle represents a four-wheeled vehicle, with the front wheel in the center of the front axle and the rear wheel in the center of the rear axle. Due to the steering angle δ , the vehicle cannot drive sideways and drives on a circle with a radius $R = L/\tan(\delta)$, where L denotes the distance between the front and rear axles. For the path planning problem for AVs, we define a configuration-space or C-space as $\chi \subset \mathbb{R}^n$. The road curvature in the vehicle C-space is defined as $\kappa = 1/R$.

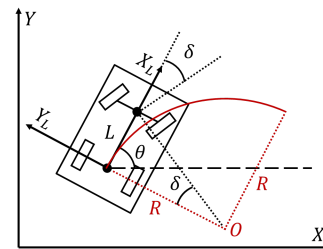


Fig. 2: Illustration of bicycle model.

We use the Frenét frame representation for 2D space because it is suitable for structured environments and traffic semantics modeling [15]. Usually, the driving reference line is extracted from an HD (High-Definition) map. In a Frenét frame, the space is decoupled into two orthogonal axes s and l (see Fig. 3). The vehicle states in a Cartesian frame are decoupled in the lateral and longitudinal directions. The states of the tracked objects are also projected into the Frenét frame.

A point in the C-space represents the ego vehicle. Road boundary considering ego vehicle width is used as the lower and upper bounds in the Frenét frame. Other traffic participants also need to be represented in the C-space. We adopt the safety ellipse to inflate the occupancy of other vehicles. As shown in Fig. 3, the semi-major axis of the safety ellipse are denoted as S_{offset} and L_{offset} respectively.

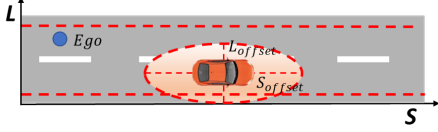


Fig. 3: Illustration of configuration space

B. Bézier Curve and Bézier Trajectory

The Bernstein basis is defined as $b_n^i(x) = \binom{n}{i} \cdot x^i \cdot (1-x)^{n-i}$, $x \in [0, 1]$. The polynomial function represented by linear combinations of the Bernstein basis is called a Bézier curve. A Bézier curve of degree n is expressed as follows:

$$B(x) = c^0 b_n^0(x) + c^1 b_n^1(x) + \dots + c^n b_n^n(x) = \sum_{i=0}^n c^i b_n^i(x) \quad (1)$$

where the polynomial coefficients $[c^0, c^1, \dots, c^n]$ symbolized as \mathbf{c} are the vector of control points for the Bézier curve. Compared to a monomial basis polynomial, the Bernstein basis polynomial has the following properties [16]:

- 1) Fixed interval. The Bézier curve with respect to variable x is defined on $x \in [0, 1]$.
- 2) End point interpolation. The Bézier curve always begins with the first control point, and terminates at the last control point, but does not pass other control points.
- 3) Convex hull. The Bézier curve $B(x)$ comprises a set of control points \mathbf{c} that lie entirely within the convex hull defined by all these control points. If the control points of the Bézier curve satisfy $\underline{p} \leq c^i \leq \bar{p}$, $\forall i \in \{0, 1, \dots, n\}$, it follows that $\underline{p} \leq B(x) \leq \bar{p}$, $\forall x \in [0, 1]$.
- 4) Hodograph. A hodograph is denoted as the derivative curve $B^{(1)}(t)$ of the Bézier curve $B(x)$ and is always a Bézier curve with control points satisfying $c^{i,1} = n \cdot (c^{i+1,0} - c^{i,0})$, where n is the polynomial degree.

The Bézier curve is defined on a fixed interval $[0, 1]$. To get an interval of arbitrary length for each trajectory segment, we need a scale factor h to scale any x assigned to that segment. Thus, the basic Bernstein piecewise trajectory with m segments can be written as follows [16]:

$$f(x) = \begin{cases} h_0 B_0\left(\frac{x-X_0}{h_0}\right), x \in [X_0, X_1] \\ h_1 B_1\left(\frac{x-X_1}{h_1}\right), x \in [X_1, X_2] \\ \dots \\ h_{m-1} B_{m-1}\left(\frac{x-X_{m-1}}{h_{m-1}}\right), x \in [X_{m-1}, X_m] \end{cases} \quad (2)$$

where $B_j(t)$ is the j -th Bézier polynomial. c_j^i is the i -th control point of the j -th segment of the whole trajectory.

X_1, X_2, \dots, X_m are the interval end of each segment. The total interval length is $X = X_m - X_0$. h_0, h_1, \dots, h_{m-1} are the scale factors for each piece of the trajectory, such that the interval of a Bézier polynomial is scaled from $[0, 1]$ to the interval $[X_{j-1}, X_j]$ allocated in one segment.

To support the further formulation of the optimization problem, we give some important definitions and theorems. The arbitrary j -th piece of a Bézier trajectory $f(x)$ is denoted by $f_j(x)$.

Definition 1 (Collision-free Space Ω): Assuming that the occupancy of all obstacles at time t in the configuration space χ is known and denoted as $Occ(t)$. The set $\Omega(t) \subset \chi$ is the set of collision-free states at time t without collision with $Occ(t)$, i.e. $\Omega(t) = \chi \setminus Occ(t)$.

Definition 2 (Convex Corridor S^{cor}): A convex set in Ω is called a convex corridor, denoted as S^{cor} . If $f_j(x)$ resides in S^{cor} for convex hull property, $f_j(x)$ is collision-free.

Theorem 1 [16]: Assume that an arbitrary control point of $f_j(x)$ satisfies $c_j^i \in \{c_j^i | \underline{p}_j^0 \leq h_j c_j^i \leq \bar{p}_j^0\}$, where \underline{p}_j^0 and \bar{p}_j^0 denote the lower bound and upper bound bias, respectively. Then the convex corridor $S^{cor} = \{(x, y) | \underline{p}_j^0 \leq y \leq \bar{p}_j^0, x \in [X_j, X_{j+1}]\}$ is also a rectangular corridor, denoted as S^{rec} , where $f_j(x)$ is a collision-free trajectory residing in S^{rec} .

Theorem 1 is an extension of convex hull property. The optimization of Bézier trajectories with S^{rec} has been applied in UAVs [16], and AVs [17]. Furthermore, by combining the convex hull property and hodograph property, we can use control points to constrain the Bézier trajectory's hodograph, such as the trajectory's velocity, acceleration, and jerk.

Lemma 1 [3]: Let $M \in \mathbb{R}^{(n+1) \times (n+1)}$ denote a change of basis matrix from Monomial basis $(1, x, \dots, x^n)$ to Bernstein basis $(b^0(x), b^1(x), \dots, b^n(x))$. We have $M_{i,0} = 1$, $0 \leq M_{i,j} \leq 1$, $i \in \{0, 1, \dots, n\}$, $j \in \{0, 1, \dots, n\}$.

Theorem 2: Assume that an arbitrary control point of $f_j(x)$ satisfies $c_j^i \in \{c_j^i | \underline{p}_j^0 + h_j \underline{p}_j^1 M_{i,1} \leq h_j c_j^i \leq \bar{p}_j^0 + h_j \bar{p}_j^1 M_{i,1}\}$, where $\underline{p}_j^0, \underline{p}_j^1$ are the lower bound bias and skew, and \bar{p}_j^0, \bar{p}_j^1 are the upper bound bias and skew. Then the convex corridor $S^{cor} = \{(x, y) | \underline{p}_j^0 + h_j \underline{p}_j^1 \frac{x-X_j}{h_j} \leq y \leq \bar{p}_j^0 + h_j \bar{p}_j^1 \frac{x-X_j}{h_j}, x \in [X_j, X_{j+1}]\}$ is also a trapezoidal corridor, denoted as S^{tra} , where $f_j(x)$ is a collision-free trajectory residing in S^{tra} .

The proof of Theorem 2 in [3] does not consider the scale factor. We corrected the proof in our work. S^{tra} can more accurately approximate Ω than S^{rec} , which has been applied in speed profile optimization in [3]. Our work implements S^{tra} for both speed and path optimization for its simplicity and sufficient accuracy. It should also be noted that the most recent work [18] provides a sufficient condition for convex hull property for more general convex corridors.

C. Robust Trajectory Repairing

Fig. 4 illustrates the relationships of the necessary definitions for the robust trajectory repairing method, which are described next.

Definition 3 (XTR): XTR is the maximum metric that the ego vehicle can follow the reference trajectory $u([x_0, x_h])$ with

respect to variable x for which a collision-free trajectory is guaranteed. x_0 is the initial state, x_h is the horizon. x can be time t or distance s , correspondingly we have TTR or DTR.

Definition 4 (Cut-off State): In the real world, every dynamic system has actuation delays and errors. ΔX denotes here the compensation time or distance for the actuation delay. Subtracting ΔX from XTR, we get the cut-off state, which is the maximal X where the AD system must execute an evasive maneuver.

Definition 5 (α -Robustness): Due to changes in driving conditions, a collision is likely to happen at XTC (X-To-Collision), and a critical XTR is respectively identified. α in $[0, 1]$ denotes the robustness metric for a choice of a feasible state X_f (t or s) to react. We then define α -Robustness as $\alpha X_f = \alpha \cdot (XTR - \Delta X)$, which indicates the state (t or s) at which the repairing will start.

Tuning α is straightforward. With a smaller α , a larger segment of the reference trajectory must be repaired; the AD system is more sensitive to driving condition changes but provides a more comfortable reaction. On the contrary, with a larger α , a smaller segment of the reference trajectory must be repaired, and the AD system is more robust against driving condition changes; however, the maneuver is more aggressive due to approaching the critical point. If α is 0, the planning scheme is the same as replacing the trajectory (re-planning). If α is 1, the planning scheme is the same as repairing.

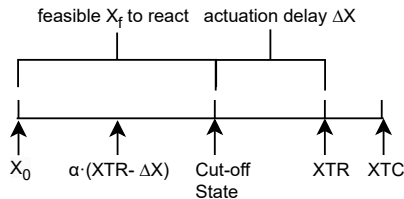


Fig. 4: Illustration of α -robustness

IV. OVERALL APPROACH

Fig. 5 gives an overview of the proposed trajectory repairing scheme. We first detect that the initial reference trajectory violates the traffic rule or potentially causes a collision. The first option is to adapt the ego vehicle velocity, and we search for TTR in the S-T domain. If adjusting the speed to avoid the traffic rule violation is possible and proper (e.g., not leading to a full stop), the speed repairing is activated, and the updated speed profile is further given to the control and actuation layer. However, suppose abating the speed is impossible or improper. In this case, we compute the DTR in the L-S domain and check if it is possible and proper to avoid traffic rule violations. If it is, the path repairing is activated, and speed repairing is again executed to update the speed profile for the new path. If it is not, AEB (Automatic Emergency Braking) system is triggered and performs an emergency brake.

A. Cut-off State Detection

To provide sufficient space for possible driving maneuvers from the cut-off state, we need to under-approximate the XTR considering evasive maneuvers related to speed (i.e., brake and kick-down) and evasive maneuvers related to the path (i.e., steering left or right). In previous work [2], both M_{speed} (speed-related maneuvers) and M_{path} (path-related maneuvers) are computed simultaneously for under-approximating TTR. However, this is not efficient and is counter-intuitive. We propose a hierarchical search scheme, in which we firstly under-approximate TTR considering M_{speed} ; if it is not proper, we search for DTR considering M_{path} . As shown in Scenario (1) of Table II, we reduce the computation time by avoiding the unnecessary search for DTR.

Fig. 6a shows the generated M_{speed} in the $S-T$ domain. An obstacle suddenly cuts in at $1.9s$, leading to a potential collision. Hence the reference speed profile must be adjusted. In the example, the time resolution is $0.1s$, TTK is $0.3s$, and TTB is $0.7s$. Therefore, TTR is $0.7s$.

Fig. 6b shows the generated M_{path} in the $X-Y$ domain. We follow the design of evasive steering maneuvers in [19]. The lateral target of the evasive path has a lateral offset L_{offset} to the obstacle and is parallel to the reference path. Different from [19] using a polynomial model for the evasive path, we adopt Dubins Path [11] for the evasive path, as it takes account of the minimum turning radius of a bicycle model. In Fig. 6b, the traffic rule disallows the ego vehicle to steer to the right. The DTR is hence the DTS to the left.

B. Trajectory Repairing Using Bézier Curve Optimization

We utilize the same Bézier trajectory optimization formulation for path repairing and speed repairing with slightly different formulations of constraints. The repairing starts

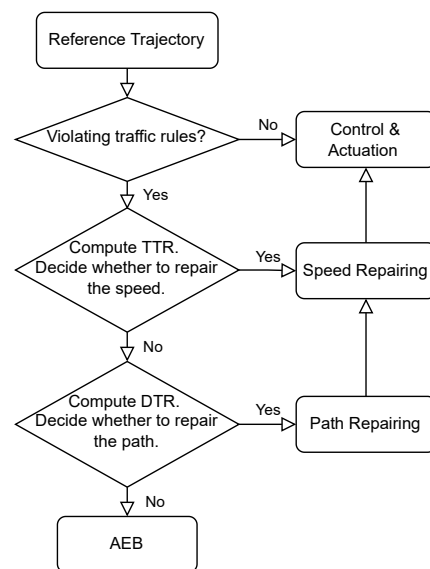


Fig. 5: Flowchart of the proposed trajectory repairing framework

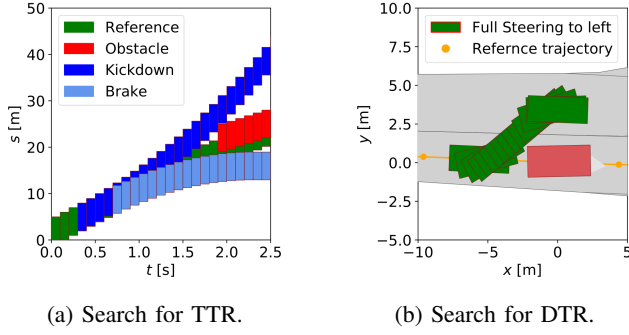


Fig. 6: Exemplary results of binary search in S-T domain and X-Y domain.

from the desired point with α -robustness. The objective function for the Bézier trajectory is designed as follows:

$$J = w_1 \int_0^X (f(x) - r(x))^2 dx + w_2 \int_0^X (f'(x) - r')^2 dx + w_3 \int_0^X f''(x)^2 dx + w_4 \int_0^X f'''(x)^2 dx + w_5 (f(X) - r(X))^2 \quad (3)$$

where x can represent t or s , and $f(x)$ can represent $s(t)$ or $l(s)$. $r(x)$ is the reference speed profile or path. r' is the constant reference speed v_r or the constant reference lateral change rate l'_r . X is the horizon T_m or S_m . $w_1 \dots w_5$ are weights for each term. The first term is to minimize the difference between the Bézier trajectory and the reference trajectory. The second term minimizes the deviations between the actual $f'(x)$ and the reference r' . The third and fourth objectives smooth the Bézier trajectory by penalizing acceleration and jerk, respectively. In addition, we expect the end station to reach a certain value $r(X)$ by the last term.

Next, we introduce the typical constraints for the optimization problem for both $S-T$ domain and $L-S$ domain, including boundary constraints, continuity constraints, security constraints, and physical constraints. However, the kinematic constraints differ in the $S-T$ domain and the $L-S$ domain. In the following formulation, $c_j^{i,l}$ is the i -th control point of the j -th segment of the Bézier trajectory with the l -th order derivative. h_j is the scale factor for the j -th segment of the Bézier trajectory. We provide more general constraints compared to previous work [16], [17], [3].

1) *Boundary Constraints*: The piecewise Bézier trajectory starts at a fixed value of the zero-order, first-order, and second-order derivative, and it is defined as

$$(h_0)^{1-l} c_0^{0,l} = \frac{d^l f(x)}{dx^l} \Big|_{x=0}, l = 0, 1, 2 \quad (4)$$

2) *Continuity Constraints*: The piecewise Bézier trajectory is continuous at the connecting points in terms of the zero-order, first-order, and second-order derivatives. It follows that

$$(h_j)^{1-l} c_j^{n,l} = (h_{j+1})^{1-l} c_{j+1}^{0,l}, l = 0, 1, 2, j = 0, 1, \dots, m-1. \quad (5)$$

3) *Safety Constraints*: With trapezoidal corridors S^{tra} , we come to the safety constraints:

$$\underline{p}_j^0 + h_j \underline{p}_j^1 M_{i,1} \leq h_j c_j^{i,0} \leq \bar{p}_j^0 + h_j \bar{p}_j^1 M_{i,1} \quad (6)$$

where $i = 0, 1, \dots, n$, $j = 0, 1, \dots, m-1$. As stated in Section III-A, the ego vehicle is represented as a point, and other obstacles are correspondingly inflated according to the safe ellipse. To have linear constraints, we take the S_{offset} and L_{offset} as an additional safety margin in the safety constraint formulation. We implemented a corridor generation algorithm that generates s^{tra} concerning a minimum resolution and merges similar pieces of the Bézier trajectory.

4) *Physical Constraints*: We take into account the real physical limits of the vehicle and constrain the velocity, acceleration, and jerk. We use the Hodograph property (see Section III-B) to obtain the Bézier polynomials of velocity, acceleration, and jerk. The constraints follow that

$$\underline{\beta}_j^l \leq (h_j)^{1-l} c_j^{i,l} \leq \bar{\beta}_j^l \quad (7)$$

where $i = 0, 1, \dots, n$, $l = 1, 2, 3$, $j = 0, 1, \dots, m-1$. $\underline{\beta}_j^l$ and $\bar{\beta}_j^l$ are upper bound and lower bound for l -th derivative of the j -th segment respectively. The bounds for acceleration and jerk remain constant for different pieces of the Bézier trajectory.

5) *Kinematic Speed Constraints*: The generated speed profile must be kinematically feasible. Let a_{lat}^{des} be the desired lateral acceleration in the vehicle frame and $|k|_{r,max}$ be the maximum absolute curvature of the reference path for $t \in [T_j, T_{j+1}]$. Similar to [20], the lateral acceleration is constrained as

$$c_j^{i,1} \leq \min\{\bar{\beta}_j^1, \sqrt{a_{lat}^{des} / |k|_{r,max}}\} \quad (8)$$

6) *Kinematic Path Constraints*: The generated path must be kinematically feasible, and its turning radius must be greater than the minimum turning radius of the ego vehicle. Following the formulation in [21], we assume that 1) the ego vehicle's orientation is approximately parallel to the reference line of the Frenét frame; 2) The lateral "acceleration" l'' is numerically small and is assumed to be 0. The second assumption can easily be fulfilled as $l''(s)$ or $f''(x)$ is minimized during optimization. However, for an evasive maneuver, assumption (1) is not true, so we reshape the formulation of the constraints.

In [21], the curvature of the path is bounded by:

$$\tan(\delta_{max}) * \kappa_r * l - \tan(\delta_{max}) + |\kappa_r| * l_{wb} \leq 0 \quad (9)$$

where δ_{max} is the maximal steering angle, κ_r is the reference curvature, and l_{wb} is the wheelbase length. The inequality (9) shows that bounding κ is equivalent to bounding l . We can express the corresponding inequality for the j -th segment as follows

$$h_j c_j^{i,0} \leq \max\left\{\frac{1}{\kappa_r} - \frac{l_{wb}}{\tan(\delta_{max})}, l_{fu}\right\} \text{ if } \kappa_r > 0 \quad (10)$$

$$h_j c_j^{i,0} \geq \min\left\{\frac{1}{\kappa_r} + \frac{l_{wb}}{\tan(\delta_{max})}, l_{fl}\right\} \text{ if } \kappa_r < 0 \quad (11)$$

where l_{fu} and l_{fl} are a feasible upper bound and a feasible lower bound, respectively. In practice, we use the curvature with maximal absolute value as κ_r in the path kinematic constraint for each path segment, which is already conservative. Therefore we limit the boundary provided by the

curvature constraint because an ultra-conservative boundary makes the optimization problem infeasible in the context of a lane change.

Finally, speed repairing and path repairing can be formulated as a quadratic programming (QP) problem :

$$\begin{aligned} \min_{\mathbf{c}} \quad & \mathbf{c}^T Q_c \mathbf{c} + \mathbf{p}_c^T \mathbf{c} + const \\ \text{s.t.} \quad & A_{eq} \mathbf{c} = \mathbf{b}_{eq} \\ & A_{ie} \mathbf{c} \leq \mathbf{b}_{ie} \end{aligned} \quad (12)$$

where \mathbf{c} is a combined vector of $(\mathbf{c}_0, \mathbf{c}_1, \dots, \mathbf{c}_{m-1})$. Due to the limit of pages, we refer the interested reader to the appendix of [3] for the formulation of speed profile optimization.

V. EVALUATION

We evaluate our approach using traffic scenarios from the open-source CommonRoad platform [22]. The implementation of our approach is in Python and runs on a computer with an Intel(R) Xeon(R) W-2123 CPU @ 3.60GHz. The search-based planner shown in [20] is able to provide a reference trajectory. The vehicle parameters for the ego vehicle is for Ford Escort [22]. Furthermore, we limit the maximal execution time for all algorithms to 1.0s.

A. Baseline: CL-RRT

CL-RRT was originally proposed for the 2007 DARPA Urban Challenge [23], and it has been recently improved and applied in trajectory repairing problems [2]. Following [2], [23], we implemented a variant of CL-RRT, which samples in the state space of (s, l, Φ, v_{cmd}) . Φ denotes the sampled orientation and v_{cmd} is the desired command velocity. The sampling strategy in [2] uses a Gaussian mixture model to increase the sampling efficiency. However, its performance relies on the quality of reference trajectory and requires well-designed heuristics. For more general cases, we adopt the random sampling strategy in the original work [23].

B. Scenario 1: Urban T-Intersection

We select a challenging urban T-intersection scenario¹ to validate our approach. The scenario animation can be found in the Scenario Selection Tool of CommonRoad. According to the flowchart in Fig. 5, we first compute the TTR. Each car in the scenario is assumed to have a rectangular shape. We project the predicted trajectories of the extreme vertices of obstacles that intervene in the ego-driving lane into the S-T domain. Two cars crossing the driving lane are presented in the middle of Fig. 7. After computing TTR, the algorithm checks if it is feasible to adapt the speed (Its average computation time is 9.8ms in Table II). In this scenario, TTR is only 0.9s, which is challenging for sampling-based approaches. The deceleration is the proper evasive maneuver.

Next, we start speed repairing. The actuation time delay ΔT is assumed to be 0.3s. The constant reference speed v_r is 12.5m/s. The safety margin $S_{\text{offset}} = 4m$. The weights in the objective function are: $w_1, w_2 = 2, w_3, w_4, w_5 = 1$. Fig. 7 shows the optimized speed profile and generated trapezoidal

corridors with respect to different α . Fig. 8 demonstrates the optimized speed, acceleration, and jerk with respect to different α . The tradeoff effect of α can be summarized as follows: a larger α gives the system more “waiting time” to react and improves the robustness. However, the maneuver becomes more aggressive and produces higher jerks.

One exemplary “unlucky” sampling of CL-RRT projected into the S-T domain is shown in Fig. 9. We attempted different α values for initiating repairing for CL-RRT. One notable problem of it is that CL-RRT might sample a large number of invalid samples resulting in collisions, traffic rule violations, and physical limit excesses (an example is visible in Fig. 9). Therefore, its performance can vary in each simulation due to its stochastic nature. The average computation time of CL-RRT is higher than our time limit of 1.0s. By comparison, as shown in Table II, our speed repairing algorithm resulted in average computation times of 9.2 – 13.8ms for different α levels.

C. Scenario 2: Blocked T-Intersection

The second scenario is a blocked T-intersection, modified from another T-intersection scenario². We insert a static obstacle in front of the intersection so that the ego vehicle must execute a lane change to continue the route. We first searched for TTR. However, the ego vehicle must stop fully and cannot continue driving. Hence we again search for DTR, which is 27m in this scenario, and steering to the left is a proper evasive maneuver. The average computation time of the search for TTR and DTR is 11.4ms and 1.8ms, respectively.

The path repairing is afterward triggered. As we use path repairing that does not comprise temporal information, we project the occupancy of each traffic agent in a prediction horizon of 10s into the L-S and X-Y domain to ensure safety. The predicted trajectory of the car behind the ego vehicle is ignored, as it has to comply with the ego vehicle. The actuation distance delay compensation is $\Delta s = v_r \cdot \Delta T = 3.75m$. The longitudinal safety margin S_{offset} is 4m, and the lateral safety margin L_{offset} is 2.3m. All weights of the objective function are 1. Fig. 10a and Fig. 10b show the path repairing in L-S and X-Y domains with different α values, respectively. The path is repaired due to an emerging static obstacle and returns to the reference path before the T-intersection.

After path repairing, the speed must be re-planned as well. To avoid aggressive behavior, kick-down is forbidden. The speed re-planning is the same as speed repairing with $\alpha = 0$, shown in the first scenario. In the interests of brevity, the optimization result is not reported here, but its computation time is indicated in Table II.

One exemplary result of CL-RRT is shown in Fig. 10c. Due to invalid samples, CL-RRT’s average computation time exceeds the time limit (1.0s), which is remarkably slower than our proposed approach (total average computation time is approximately 25ms) (See Table II).

¹CommonRoad ID: DEU_Flensburg-6.1.T-1

²CommonRoad ID: ZAM_Tjunction-1.196.T-1

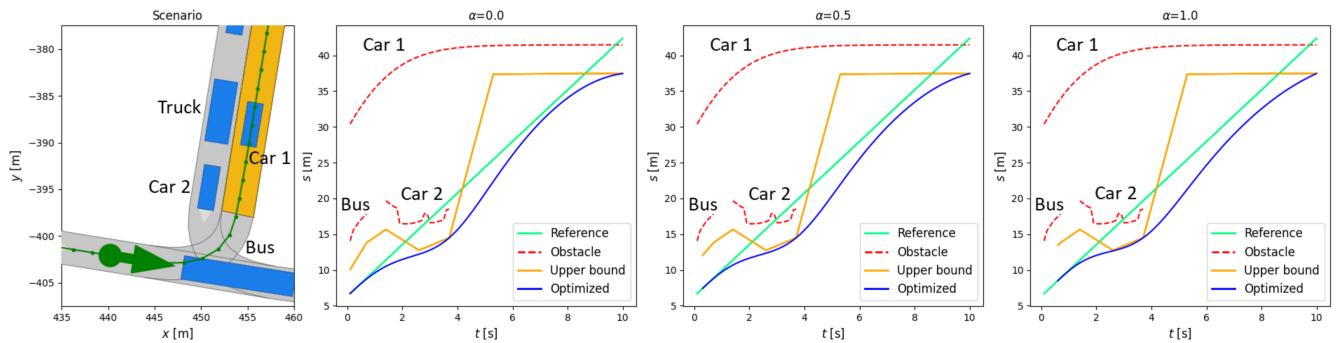


Fig. 7: Optimization and trapezoidal corridors with different α . The left picture shows the urban T-intersection scenario, where planning starts from the arrow following a green line and attempts to reach the yellow goal region. The three right pictures include the other vehicles’ extreme points projected into the S-T domain (in the ego lane) and optimization results.

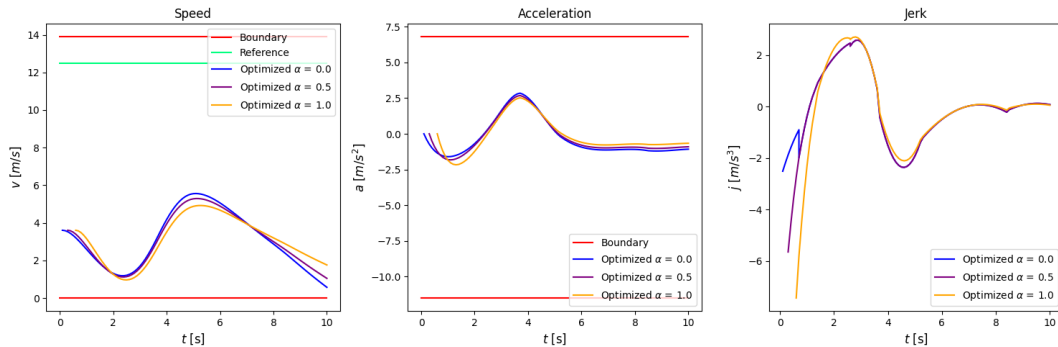


Fig. 8: Optimized speed, acceleration, and jerk with different α .

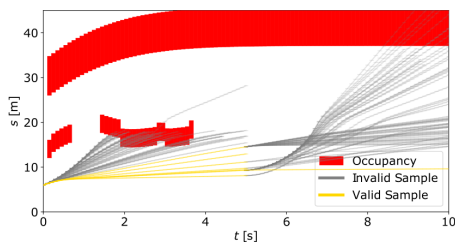


Fig. 9: “Unlucky” sampling of CL-RRT projected into S-T domain. After 500 samples (17.7 s), CL-RRT did not find a solution to the goal due to too many invalid samples.

We compared the maximal and average curvature of the generated path for our approach for different α . The kinematic path constraints in Section IV-B.6 are activated and ensure that the curvature of the generated path is less than the maximal curvature (0.54) of the single-track model. With a smaller α , a smoother path can be generated. However, due to the path-speed decoupling, the optimization result is sub-optimal and has larger average curvature than the trajectory generated by CL-RRT.

VI. CONCLUSION

We developed a robust tunable trajectory repairing framework for AVs based on Bernstein basis polynomials and path-

TABLE I: Comparison of curvature.

	$\alpha=0$	$\alpha=0.5$	$\alpha=1$	CL-RRT	Maximal
Maximal Curvature	0.46	0.46	0.45	0.10	0.54
Average Curvature	0.06	0.08	0.13	0.03	0.54

speed decoupling. We improved the search efficiency for the critical measures by decoupling the search scheme into search in the S-T domain and X-Y domain. In addition, we proposed the concept α – Robustness. It is a generalization of re-planning and repairing and can be used to balance the driving comfort against robustness to external disturbances. The trajectory repairing approach based on random sampling in the C-space might cause numerous invalid samples and non-deterministic performance. Heuristics can improve the sampling efficiency (i.e., [2]) but requires a subtle design. By contrast, we formulated a unified QP problem with different kinematic constraints for both speed and path repairing. The QP formulation ensures that the optimization can be solved with a limited time cost meanwhile achieving kinematic feasibility, safety and comfort. Our experiments indicated typical feasible solutions within 25ms, which is sufficient for real-time safety-critical applications.

REFERENCES

- [1] R. L. McCarthy, “Autonomous vehicle accident data analysis: California ol 316 reports: 2015–2020,” *ASCE-ASME J Risk and Uncert in*

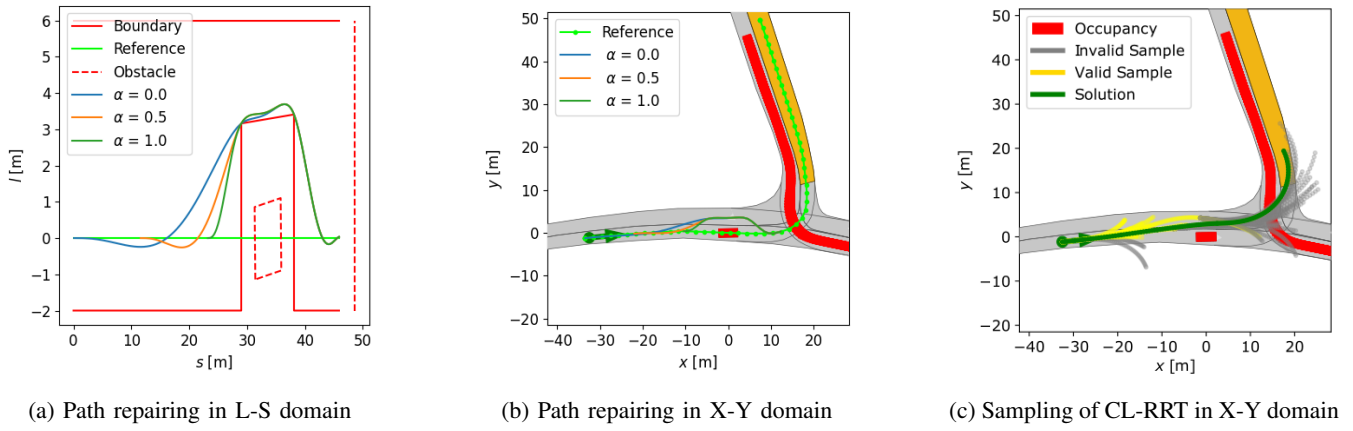


Fig. 10: Benchmark of path repairing

TABLE II: Comparison of computation time. We run 100 iterations for each algorithm. Speed and path refer to speed repairing and path repairing, including the computation time for generating trapezoidal corridors and establishing and solving the optimization problem. The number before and after \pm are the average and standard deviation, respectively.

Scenario	Cut-off State		$\alpha = 0$		$\alpha = 0.5$		$\alpha = 1$		CL-RRT
	TTR	DTR	Speed	Path	Speed	Path	Speed	Path	
(1)	9.8 \pm 0.4ms	-	9.2 \pm 2.7ms	-	9.8 \pm 1.0ms	-	13.8 \pm 1.2ms	-	TIMEOUT
(2)	11.4 \pm 1.8ms	1.8 \pm 0.0ms	18.7 \pm 1.8ms	6.1 \pm 0.2ms	18.8 \pm 1.8ms	6.0 \pm 0.3ms	18.8 \pm 1.4ms	6.2 \pm 0.7ms	TIMEOUT

- Engrg Sys Part B Mech Engrg*, vol. 8, no. 3, 2022.
- [2] Y. Lin, S. Maierhofer, and M. Althoff, "Sampling-based trajectory repairing for autonomous vehicles," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 572–579.
 - [3] J. Li, X. Xie, H. Ma, X. Liu, and J. He, "Speed planning using bezier polynomials with trapezoidal corridors," *arXiv preprint arXiv:2104.11655*, 2021.
 - [4] J. Guo, U. Kurup, and M. Shah, "Is it safe to drive? an overview of factors, metrics, and datasets for driveability assessment in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3135–3151, 2020.
 - [5] J. Hillenbrand, A. M. Spieker, and K. Kroschel, "A multilevel collision mitigation approach—its situation assessment, decision making, and performance tradeoffs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 4, pp. 528–540, 2006.
 - [6] S. Kim, J. Wang, G. J. Heydinger, and D. A. Guenther, "The criticality index development for steering evasive maneuver based on mixed h2/hcontrol with parameter uncertainties," in *2019 American Control Conference (ACC)*, 2019, pp. 3963–3968.
 - [7] M. Schratte, M. Hartmann, and D. Watenig, "Pedestrian collision avoidance system for autonomous vehicles," *SAE International Journal of Connected and Automated Vehicles*, vol. 2, no. 4, 2019.
 - [8] S. Sontges, M. Koschi, and M. Althoff, "Worst-case analysis of the time-to-react using reachable sets," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1891–1897.
 - [9] A. Tamke, T. Dang, and G. Breuel, "A flexible method for criticality assessment in driver assistance systems," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 062011, pp. 697–702.
 - [10] D. Gonzalez, J. Perez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1135–1145, 2016.
 - [11] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.
 - [12] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, "Baidu apollo em motion planner," *arXiv preprint arXiv:1807.08048*, 2018.
 - [13] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.
 - [14] Q.-C. Pham and Y. Nakamura, "A new trajectory deformation algorithm based on affine transformations," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 1054–1063, 2015.
 - [15] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenét frame," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 03.05.2010 - 07.05.2010, pp. 987–993.
 - [16] F. Gao, W. Wu, Y. Lin, and S. Shen, "Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 344–351.
 - [17] W. Ding, L. Zhang, J. Chen, and S. Shen, "Safe trajectory generation for complex urban environments using spatio-temporal semantic corridor," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2997–3004, 2019.
 - [18] W. Zhang, P. Yadmellat, and Z. Gao, "A sufficient condition for convex hull property in general convex spatio-temporal corridors," in *2022 IEEE Intelligent Vehicles Symposium (IV)*, 2022, pp. 1033–1039.
 - [19] C. G. Keller, T. Dang, H. Fritz, A. Joos, C. Rabe, and D. M. Gavrilu, "Active pedestrian safety by automatic braking and evasive steering," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1292–1304, 2011.
 - [20] K. Tong, S. Solmaz, and M. Horn, "A search-based motion planner utilizing a monitoring functionality for initiating minimal risk maneuvers," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 8/10/2022 - 12/10/2022.
 - [21] Y. Zhang, H. Sun, J. Zhou, J. Pan, J. Hu, and J. Miao, "Optimal vehicle path planning using quadratic optimization for baidu apollo open platform," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2020, pp. 978–984.
 - [22] M. Althoff, M. Koschi, and S. Manzingler, "Commonroad: Composable benchmarks for motion planning on roads," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 719–726.
 - [23] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli, and J. P. How, "Real-time motion planning with applications to autonomous urban driving," *IEEE Transactions on control systems technology*, vol. 17, no. 5, pp. 1105–1118, 2009.