Safer Gap: Safe Navigation of Planar Nonholonomic Robots With a Gap-Based Local Planner

Shiyu Feng, Ahmad Abuaish, and Patricio A. Vela

***Abstract*—This paper extends the gap-based navigation technique *Potential Gap* with safety guarantees at the local planning level for a kinematic planar nonholonomic robot model, leading to *Safer Gap*. It relies on a subset of navigable free space from the robot to a gap, denoted the keyhole region. The region is defined by the union of the largest collision-free disc centered on the robot and a collision-free trapezoidal region directed through the gap. *Safer Gap* first generates Bézier-based collision-free paths within the keyhole regions. The keyhole region of the top scoring path is encoded by a shallow neural network-based zeroing barrier function (ZBF) synthesized in real-time. Nonlinear Model Predictive Control (NMPC) with *Keyhole ZBF* constraints and output tracking of the Bézier path, synthesizes a safe kinematically feasible trajectory. The *Potential Gap* projection operator serves as a last action to enforce safety if the NMPC optimization fails to converge to a solution within the prescribed time. Simulation and experimental validation of *Safer Gap* confirm its collision-free navigation properties.**

***Index Terms*—Vision-based navigation, collision avoidance, reactive and sensor-based planning.**

I. Introduction

COLLISION-free performance of a path planner based on erroneous or uncertain maps is difficult to guarantee, as map corrections are only available when the robot is in sensing proximity to incorrectly mapped regions. Hierarchical planners incorporate an additional, smaller scale, local planning module whose objective is to correct colliding segments of the global path and follow safe segments [1], see Fig. 1. Periodic replanning at the global level ensures consistency between the collision-free local path and the global go-to-goal path. Collision-free proofs for hierarchical planners reduce to proving local planner collision avoidance [2], [3]. An approach based on discovering *gaps*, which are convex, line-of-sight free space regions containing the robot and passing between obstacles, has formally derived a local planning scheme guaranteeing avoidance [2]. The guarantees hold for holonomic robots moving at low-speed (where momentum effects do not dominate) but not for nonholonomic robots with control constraints [2], [4]. This paper extends gap-based, local planning collision-free guarantees to planar nonholonomic models under control constraints. Key changes are to define an enlarged, star-convex collision-free region local to the robot, and to derive a motion control scheme with collision-free forward invariance.

To that end, the manuscript contributions are: (i) the definition of a *keyhole* region of navigable, collision-avoiding free-space and a data-driven approach to modeling keyhole region constraints; (ii) a combined path generation and trajectory execution approach for safe, kinematically feasible differential drive navigation within the keyhole region; and (iii) Monte Carlo simulations and experimental testing scenarios verifying real-time, collision-free performance. The local planner, called *Safer Gap*, is integrated into a hierarchical navigation system as depicted in Fig. 1. *Safer Gap* addresses collision, dynamics, and control constraints to ensure collision-free, real-time navigation while maximizing local goal attainment.

II. Related Work

In hierarchical navigation systems, the global planner may be any acceptable planning method. Options include graph search planners over costmaps like Dijkstra's algorithm, A*, D*, D* Lite, and AD* [5], [6], [7], [8], [9] or sample-based planners such as PRM, RRT, and RRT-X [10], [11], [12], [13]. Global planning method selection depends on the chosen world representation and robot motion model.

A. Sensor Driven Local Planning

Local navigation from a global plan through uncertain environments requires perception modules that efficiently and accurately describe the sensed scene for safe path planning and control. Perception models generally break down into allocentric and ego-centric implementations [1]. The typical local planning approach is allocentric as it crops the global map to a local map equivalent with faster update rates; examples being EB, DWA, and TEB [14], [15], [16]. However, within the human's neural hierarchy, ego-centric processing usually happens before allocentric estimation. Ego-centric processing requires fewer memory resources and lower compute costs to represent and update the local map. Ego-centric equivalents to DWA and TEB match or best their performance while being more efficient [1], [17].

The *gap* approach noted earlier is an ego-centric method to represent local free space based on line-of-sight visibility. Though published definitions of a "gap" vary [18], [19], [20],

Received 13 September 2024; accepted 3 October 2024. Date of publication 24 October 2024; date of current version 31 October 2024. This article was recommended for publication by Associate Editor S. Zhao and Editor P. Vasseur upon evaluation of the reviewers' comments. This work was supported in part by NSF under Grant 1849333, Grant 2235944, and Grant 2345057, and in part by KACST Fellowship. *(Shiyu Feng and Ahmad Abuaish contributed equally to this work.) (Corresponding author: Shiyu Feng.)*

Shiyu Feng is with the School of Mechanical Engineering and the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30308 USA (e-mail: shiyufeng@gatech.edu).

Ahmad Abuaish and Patricio A. Vela are with the School of Electrical and Computer Engineering and the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30308 USA (e-mail: aabuaish@gatech.edu; pvela@gatech.edu).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3486231, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3486231

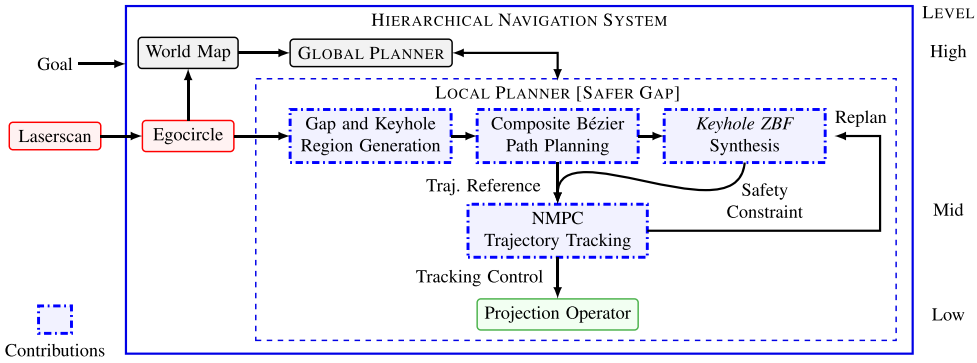

Fig. 1. Hierarchical navigation system with *Safer Gap* local planner. Red blocks are perception modules to generate the local obstacle and free-space model called the egocircle. Blue blocks are local planning modules for *Safer Gap*. The green block is the control module with a safety filter.

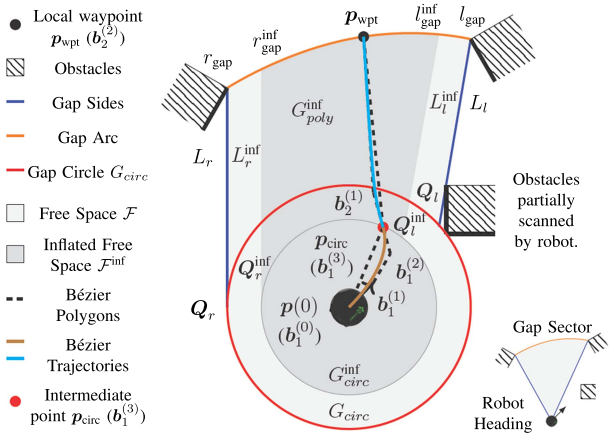

Fig. 2. Composite Bézier path synthesis. $\boldsymbol{p}(0)$ is the robot's origin. The red circle is the largest circular free space in the egocircle. l_{gap} and r_{gap} are defined counter clockwise in the robot's local frame. l_{gap} has a smaller polar angle than r_{gap}. Blue lines L_l and L_r are left and right gap sides. $\boldsymbol{Q}_l$ and $\boldsymbol{Q}_r$ are the left and right intersection points of G_{circ} and gap sides. $(\cdot)^{\text{inf}}$ are inflated symbols. $\boldsymbol{b}_1^{(1)}$ and $\boldsymbol{b}_1^{(2)}$ are the second and third control points for the first cubic Bézier curve. Local waypoint $\boldsymbol{p}_{\text{wpt}}$ is inside the inflated safe region $\mathcal{F}^{\text{inf}}$ to guarantee safety. $\boldsymbol{p}_{\text{circ}}$ is the goal-biased point on G_{circ}^{inf}. Dashed lines show the Bézier polygons. The combination of brown and cyan paths is the final synthesized path. A gap sector region [2] corresponding to the same scenario is depicted in the bottom right corner.

[21], [22], [17], [23], the one defined in [2] prioritizes a specification with collision-free properties. There, a "gap" is defined to be a collision-free, robot-centered circular sector, see Fig. 2. *Path planning with gaps emphasizes candidate directions to attract towards, rather than directions to avoid (e.g. obstacles).* Gaps simplify the local planner of hierarchical navigation systems, naturally provide multiple distinct path options, and support safety and traversability. The *Potential Gap* local planner [2] has proven safety guarantees for point-mass holonomic robot models, but not for nonholonomic mobile robots. Additional algorithmic components were used to improve navigation safety due to the lost guarantees, i.e., radial extension and projection operator, but without theoretical support.

The artificial potential field (APF) is an efficient goal attraction and obstacle avoidance strategy [24], [25], [26], [27], [28]. The *Potential Gap* local planner [2] uses APF ideas to synthesize safety guaranteed paths for point-mass holonomic models. The Bézier curve is also a well-known technique for generating smooth trajectories for different robot models [29], [30], [31], [32], [33], [34]. Recent work called Bézier Gap [4], leverages Bézier curve properties to synthesize safe trajectories for box-shaped holonomic robots. However, safety guarantees for nonholonomic models are still missing. This paper takes advantage of gaps and Bézier curves for nonholonomic path planning to address the weak safety guarantee issue.

B. Nonlinear Model Predictive Control

Model predictive control (MPC) is a receding-horizon controller for generating or tracking trajectories [35]. For tracking, trajectories are typically generated by a different module that uses a simplified system model for fast trajectory generation. From this trajectory, efficient nonlinear MPC (NMPC) for path and trajectory tracking by nonholonomic robots [36], [37] enable real-time, local planning. Constraint specification in NMPC uses either soft [37], [38] or hard constraints [39]. Soft costs act like potential functions and penalize movement outside of the safe regions [38]. The MPC-based TEB planner [37] uses soft constraints to avoid obstacles. However, safety is not guaranteed with soft constraints. Therefore, applying a solvable hard constraint to prevent collisions is necessary for the safe navigation of nonholonomic mobile robots.

C. Safety and Control

In safe control synthesis, a zeroing barrier function (ZBF) is a differentiable implicit function whose zero super level-set defines safe (*i.e.*, collision-free) space. A ZBF with the additional property of a non-empty admissible control set for all points in the safe space is termed control barrier function (CBF). Safe, real-time control synthesis for control affine nonlinear systems, may be achieved through an online point-wise quadratic program (QP) optimization, referred to as CBF-QP. Employing a ZBF in the CBF-QP formulation without certification is undesirable. Certifying a ZBF as a valid CBF is computationally intensive and often performed offline. Popular methods used for constructing and validating CBFs are sum-of-square (SoS) [40], [41] and Hamilton-Jacobi (HJ) reachability [42]. Recently, machine learning techniques have been studied [43], [44]. In particular, there are techniques for online, real-time ZBF synthesis based on shallow neural networks with data from online environment measurements [45], [46]. In [45], the outer weights of the neural network are solved using a linear program (LP). These ideas are leveraged to build keyhole region ZBFs.

III. *Safer Gap* Local Planner

This section introduces a gap-based local planner, called *Safer Gap*, designed to sense environmental elements, accomplish obstacle avoidance, and guarantee safe navigation for planar nonholonomic mobile robots. A gap (sector) is an open region between two obstacles based on line-of-sight visibility, as shown in Fig. 2. For ground robots, it is generated from the egocircle [2], which is an ego-centric circular field with spatial information [1]. To address nonholonomic constraints, a larger collision-free *keyhole* region is introduced and defined here. Safe motion planning synthesizes smooth paths within collision-free *keyhole* regions using composite Bézier curves. Nonlinear Model Predictive Control (NMPC) addresses kinematic, *Keyhole ZBF* safety, and control constraints feasibility during curve tracking.

A. Collision-Free Space Generation: Keyhole Regions

Each gap requires defining a collision-free space $\mathcal{F}$ local to the robot. In [2], [4], the points $l_{\text{gap}}/r_{\text{gap}}$ define line segments from the robot position whose collision-free space lies to the left/right of obstacles identified from the egocircle. This gap sector region is limiting when the robot has one side close to an obstacle [4], or the robot is nonholonomic [2]. A more expansive definition of the collision-free space is defined here, formed by the union of the largest robot-centered disc and the region between the gap lines. Given the safe region resemblance to a keyhole, it is called a *keyhole* region.

Start with the largest circular collision-free space G_{circ} within the egocircle $\mathcal{L}$, e.g., the red circle in Fig. 2. Now connect the two gap points, $l_{\text{gap}}/r_{\text{gap}}$, to the tangent points of G_{circ}, leading to raw gap sides. The tangent point corresponding to l_{gap} always has a smaller polar angle than that of r_{gap} in the robot's local frame. If obstacles obstruct raw gap sides, inward rotations are applied around l_{gap} (clockwise) and r_{gap} (counter clockwise) until there is no obstruction. See L_l in Fig. 2 for an example. By definition of a gap, the maximum rotation cannot pass the center of G_{circ} (i.e., the robot position). The left and right gap sides (L_l/L_r) are finalized after rotations, shown as blue line segments in Fig. 2. They represent safe region-maximizing linear connections between the gap points and G_{circ}. These intersection points are denoted by $\boldsymbol{Q}_l/\boldsymbol{Q}_r$. The four points $(l_{\text{gap}}, r_{\text{gap}}, \boldsymbol{Q}_l, \boldsymbol{Q}_r)$ define a collision-free polygon G_{poly}. The full, star-convex, obstacle-free space $\mathcal{F}$ is

$$\mathcal{F} = G_{poly} \cup G_{circ}. \tag{1}$$

To consider robot geometry, obstacle space is inflated so that $\mathcal{F}^{\text{inf}} \subset \mathcal{F}$ based on a robot size parameter s_{inf}, as depicted by dark gray in Fig. 2. s_{inf} can be radii for circular robots and equivalent passing lengths for non-circular robots [4]. The robot simplifies to a point when planning in $\mathcal{F}^{\text{inf}}$. In Fig. 2, superscripts $(\cdot)^{\text{inf}}$ are added to symbolically represent the gap circle, gap points, and gap sides under inflation. Intersected points after inflation are denoted as $\boldsymbol{Q}_l^{\text{inf}}$ and $\boldsymbol{Q}_r^{\text{inf}}$. Any path $\mathcal{P}$ within the inflated collision-free space $\mathcal{P} \in \mathcal{F}^{\text{inf}}$ guarantees safety for the robot geometry.

B. Composite Bézier Path Planning

Given keyhole region(s) local to the robot, the next step is to identify a candidate path traversing each keyhole region and crossing its gap arc, after which one is chosen to follow.

1) Composite Bézier Curves: Dual segment composite Bézier curves serve as the chosen path parametrization. The first segment is a cubic Bézier curve parameterized by u:

$$\mathcal{B}_1(u) = \sum_{i=0}^{n=3} \binom{n}{i} (1-u)^{n-i} u^i \boldsymbol{b}_1^{(i)}$$

$$\binom{n}{i} = \frac{n!}{i!(n-i)!},\ 0 \le u \le 1 \tag{2}$$

where $\boldsymbol{b}_1^{(i)}$ is the ith control point of $\mathcal{B}_1$. In *Safer Gap*, selecting control points inside G_{circ}^{inf} defines the first curve within G_{circ}^{inf}; see the brown, solid curve in Fig. 2.

Since the gap is detected in the robot's local frame, the robot center is the first control point $\boldsymbol{b}_1^{(0)} = \boldsymbol{p}(0)$. An intermediate point $\boldsymbol{p}_{\text{circ}}$ is defined on the arc between $\boldsymbol{Q}_l^{\text{inf}}$ and $\boldsymbol{Q}_r^{\text{inf}}$, serving as the last control point $\boldsymbol{b}_1^{(3)}$, defined later. The other two control points are established from the initial orientation $\theta(0)$, linear velocity $\nu(0)$, and acceleration $\vec{\boldsymbol{a}}(0)$ of the nonholonomic robot. The vector $\boldsymbol{b}_1^{(1)} - \boldsymbol{b}_1^{(0)}$ is co-linear with the unit orientation vector $\vec{\boldsymbol{o}}(0) = [\cos(\theta(0)), \sin(\theta(0))]$. Curve velocities and accelerations are obtained from the first and second derivatives of the cubic Bézier curve,

$$\dot{\mathcal{B}}_1(u) = 3\sum_{i=0}^{2} \binom{2}{i} (1-u)^{2-i} u^i (\boldsymbol{b}_1^{(i+1)} - \boldsymbol{b}_1^{(i)}) \tag{3}$$

$$\dot{\mathcal{B}}_1(0) = 3(\boldsymbol{b}_1^{(1)} - \boldsymbol{b}_1^{(0)}) \tag{4}$$

$$\ddot{\mathcal{B}}_1(u) = 6\sum_{i=0}^{1} \binom{1}{i} (1-u)^{1-i} u^i (\boldsymbol{b}_1^{(i+2)} - 2\boldsymbol{b}_1^{(i+1)} + \boldsymbol{b}_1^{(i)}) \tag{5}$$

$$\ddot{\mathcal{B}}_1(0) = 6(\boldsymbol{b}_1^{(2)} - 2\boldsymbol{b}_1^{(1)} + \boldsymbol{b}_1^{(0)}). \tag{6}$$

The curve parameter $u \in [0,1]$ should be scaled to time $t \in [0, T_1]$ through $t = T_1 u$. The final time T_1 is estimated by $||\boldsymbol{p}_{\text{circ}} - \boldsymbol{p}(0)||/\nu_d$, where ν_d is the robot's desired linear velocity. The scaled Bézier path is $\mathcal{B}_1^s(t) = \mathcal{B}_1(t/T_1)$, with

$$\dot{\mathcal{B}}_1^s(t) = \frac{1}{T_1}\dot{\mathcal{B}}_1\left(\frac{t}{T_1}\right) \text{ and } \ddot{\mathcal{B}}_1^s(t) = \frac{1}{T_1^2}\ddot{\mathcal{B}}_1\left(\frac{t}{T_1}\right). \tag{7}$$

Assign $||\dot{\mathcal{B}}_1^s(0)|| = \nu(0)$, which needs $||\boldsymbol{b}_1^{(1)} - \boldsymbol{b}_1^{(0)}|| = T_1\nu(0)/3$ from (4) and (7). Setting $\ddot{\mathcal{B}}_1^s(0) = \vec{\boldsymbol{a}}(0)$, $\boldsymbol{b}_1^{(2)}$ is calculated by (6) and (7). With these assignments, all control points for the segment $\mathcal{B}_1(u), u \in [0,1]$ are uniquely defined

$$\begin{aligned}
\boldsymbol{b}_1^{(0)} &= \boldsymbol{p}(0) \\
\boldsymbol{b}_1^{(1)} &= \boldsymbol{p}(0) + \frac{T_1\nu(0)}{3}\vec{\boldsymbol{o}}(0) \\
\boldsymbol{b}_1^{(2)} &= \frac{T_1^2}{6}\vec{\boldsymbol{a}}(0) - \boldsymbol{b}_1^{(0)} + 2\boldsymbol{b}_1^{(1)} \\
\boldsymbol{b}_1^{(3)} &= \boldsymbol{p}_{\text{circ}}
\end{aligned} \tag{8}$$

The second path segment (cyan path in Fig. 2) is generated from a quadratic Bézier curve

$$\mathcal{B}_2(u) = (1-u)^2\boldsymbol{b}_2^{(0)} + 2(1-u)u\boldsymbol{b}_2^{(1)} + u^2\boldsymbol{b}_2^{(2)}, \tag{9}$$

where $\boldsymbol{p}_{\text{circ}}$ is chosen as $\boldsymbol{b}_2^{(0)}$, the start point of the curve.

G1 or tangency continuity (i.e., the tangent vectors are the same at the joining point of two connected curves) maintains a

geometrically smooth connection between Bézier curves. Therefore, the direction vector $\vec{\mathbf{v}}$ should satisfy the equality:

$$\vec{\mathbf{v}} = \frac{\boldsymbol{b}_2^{(1)} - \boldsymbol{b}_2^{(0)}}{||\boldsymbol{b}_2^{(1)} - \boldsymbol{b}_2^{(0)}||} = \frac{\boldsymbol{b}_1^{(3)} - \boldsymbol{b}_1^{(2)}}{||\boldsymbol{b}_1^{(3)} - \boldsymbol{b}_1^{(2)}||}, \tag{10}$$

where $\boldsymbol{b}_1^{(2)}$ and $\boldsymbol{b}_1^{(3)}$ are known by (8). The derivative and time-scaled version of quadratic Bézier curve are

$$\dot{\mathcal{B}}_2(u) = 2(1-u)(\boldsymbol{b}_2^{(1)} - \boldsymbol{b}_2^{(0)}) + 2u(\boldsymbol{b}_2^{(2)} - \boldsymbol{b}_2^{(1)}) \tag{11}$$

$$\dot{\mathcal{B}}_2(0) = 2(\boldsymbol{b}_2^{(1)} - \boldsymbol{b}_2^{(0)}) \tag{12}$$

$$\dot{\mathcal{B}}_2^s(t) = \frac{1}{T_2}\dot{\mathcal{B}}_2\left(\frac{t}{T_2}\right) \tag{13}$$

where $T_2 = ||\boldsymbol{p}_{\text{wpt}} - \boldsymbol{p}_{\text{circ}}||/\nu_d$ given a customized linear velocity ν_d. $\boldsymbol{p}_{\text{wpt}}$ is the local terminal waypoint attached to the gap for the robot, so that $\boldsymbol{b}_2^{(2)}$ should be $\boldsymbol{p}_{\text{wpt}}$.

Similarly, set $||\dot{\mathcal{B}}_2^s(0)|| = \nu_d$, which requires $||\boldsymbol{b}_2^{(1)} - \boldsymbol{b}_2^{(0)}|| = T_2\nu_d/2$ from (12) and (13). When $\boldsymbol{p}_{\text{circ}}$ is close to L_l^{inf} or L_r^{inf}, $\boldsymbol{b}_2^{(1)}$ is possible to be outside of the inflated gap sides after scaling. A length scale number $\lambda \in (0, 1]$ is calculated to bound $\boldsymbol{b}_2^{(1)}$ inside $\mathcal{F}^{\text{inf}}$. Finally, $\boldsymbol{b}_2^{(1)}$ is computed by (10). All control points for the second Bézier path segment $\mathcal{B}_2(u)$ are constrained

$$\begin{aligned} \boldsymbol{b}_2^{(0)} &= \boldsymbol{p}_{\text{circ}} \\ \boldsymbol{b}_2^{(1)} &= \boldsymbol{p}_{\text{circ}} + \lambda\frac{T_2\nu_d}{2}\vec{\mathbf{v}} \\ \boldsymbol{b}_2^{(2)} &= \boldsymbol{p}_{\text{wpt}} \end{aligned} \tag{14}$$

The local waypoint $\boldsymbol{p}_{\text{wpt}}$ for each gap is initially found based on the global plan [2] and then bounded by $\mathcal{F}^{\text{inf}}$. All gaps share a local goal $\boldsymbol{p}^*$ defined from the global path. If $\boldsymbol{p}^* \in \mathcal{F}^{\text{inf}}$, then $\boldsymbol{p}_{\text{wpt}} = \boldsymbol{p}^*$. Otherwise, it is shifted to the inflated gap point, $l_{\text{gap}}^{\text{inf}}$ or $r_{\text{gap}}^{\text{inf}}$, nearest to the local goal $\boldsymbol{p}^*$ with a further offset added to ensure the robot can safely move past the gap. The intermediate point $\boldsymbol{p}_{\text{circ}}$ starts at the midpoint of the arc $\overset{\frown}{\boldsymbol{Q}_l^{\text{inf}}\boldsymbol{Q}_r^{\text{inf}}}$. Then a perpendicular line is calculated from the point $\boldsymbol{p}_{\text{wpt}}$ to the line segment connecting the midpoint and either $\boldsymbol{Q}_l^{\text{inf}}$ or $\boldsymbol{Q}_r^{\text{inf}}$, depending on which is closer to $\boldsymbol{p}_{\text{wpt}}$. If the perpendicular line intersects the arc $\overset{\frown}{\boldsymbol{Q}_l^{\text{inf}}\boldsymbol{Q}_r^{\text{inf}}}$, $\boldsymbol{p}_{\text{circ}}$ is moved to that intersection point. If $\boldsymbol{p}_{\text{wpt}}$ is within G_{circ}^{inf}, then $\boldsymbol{p}_{\text{circ}} = \boldsymbol{p}_{\text{wpt}}$, and only the first Bézier segment is computed. The final Bézier-based path is

$$\mathcal{P}(u) = \begin{cases} \mathcal{B}_1(u), & \boldsymbol{p}_{\text{wpt}} \in G_{circ} \\ \mathcal{B}_1(u) \cup \mathcal{B}_2(u), & \text{otherwise} \end{cases} \tag{15}$$

From the above design, the first Bézier polygon for $\mathcal{B}_1(u)$ is always within G_{circ}^{inf}. The second Bézier polygon is within the convex region G_{poly}^{inf}. The composite Bézier path is guaranteed to be within the inflated collision-free space [47], $\mathcal{P}(u) \subseteq \mathcal{F}^{\text{inf}}$. Safety and traversability of the keyhole region are achieved for nonholonomic motion, in the absence of control constraints. Each gap generates one composite Bézier path, which requires ≤ 2 ms (on an Intel i7-8700). Total path synthesis time is linear in the number of gaps.

2) Path Scoring: A scoring function is computed for each composite Bézier path to choose a single local path $\mathcal{P}^*$ to follow. It improves on the scoring in [2] by adding an orientation cost. Paths with lower deviation from the robot's orientation are preferable since nonholonomic robots cannot rotate instantaneously. It also helps to pick the correct path when a final goal point is on the other side of a wall.

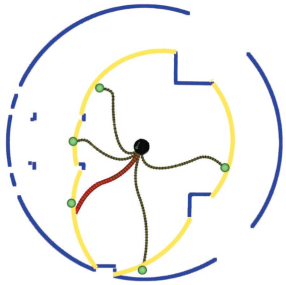

Fig. 3. Composite Bézier paths for all gaps. Blue is egocircle $\mathcal{L}$. Yellow are five detected gaps. Green points are local waypoints $\boldsymbol{p}_{\text{wpt}}$. The black paths are the synthesized Bézier paths $\mathcal{P}$. The red path is the selected $\mathcal{P}^*$ based on the scoring equation.

$$J(\mathcal{P}) = \sum_{\mathbf{x}\in\mathcal{P}} C(\mathrm{d}(\mathbf{x}, \mathcal{L})) + w_1||\mathbf{x}_{\text{end}} - \boldsymbol{p}^*|| + w_2|\theta_{\text{end}} - \theta(0)|$$

$$\text{where} \quad C(d) = \begin{cases} c_{\text{obs}}e^{-w_3(d-r_{\text{ins}})}, & d > r_{\text{ins}} \\ 0, & d > r_{\text{max}} \\ \infty, \text{otherwise} \end{cases}$$

$\mathrm{d}(\mathbf{x}, \mathcal{L})$ is the distance from path pose $\mathbf{x} = [\mathrm{x}_1, \mathrm{x}_2, \theta]^T$ to the nearest point on $\mathcal{L}$. $||\mathbf{x}_{\text{end}} - \boldsymbol{p}^*||$ measures the distance between the end pose of $\mathcal{P}$ and the local goal $\boldsymbol{p}^*$ from a global plan. Only one local goal is created in every planning step [2]. $|\theta_{\text{end}} - \theta(0)|$ is the angle difference between the end pose and the initial pose. r_{ins} and r_{max} are proportional to the robot radius to control the safe distance. w_1, w_2, w_3 and c_{obs} are tunable weights. Each time, the best path $\mathcal{P}_i^*$ compares with the previous path $\mathcal{P}_{i-1}^*$ to decide whether to switch to the new one. An example is in Fig. 3. The best path (red) is selected from a set of Bézier path candidates.

C. Keyhole ZBF *Synthesis*

The inflated collision-free space $\mathcal{F}^{\text{inf}}$ is captured by the zero super level-set of the *Keyhole ZBF*. A shallow, two hidden-layer neural network with rectified linear units (ReLU) describes the barrier function. To keep the network shallow and minimal involves leveraging the geometry of the keyhole region, i.e., the straight lines and the circle. The complete expression of the *Keyhole ZBF* is

$$\begin{aligned} h(\mathbf{x}) = {} & \alpha_1 R_1 + \alpha_2 R_2 + \alpha_3 R_3 + \alpha_4 R_c + \alpha_5 R_1 R_2 \\ & + \alpha_6 R_c R_1 + \alpha_7 R_c R_2 + \alpha_8 R_c R_3 \\ & + \alpha_9 R_1 R_2 R_3 + \alpha_{10} R_1 R_4 R_5 + \alpha_{11} R_2 R_4 R_5 \\ & + \alpha_{12} R_c R_1 R_4 + \alpha_{13} R_c R_2 R_4 + \alpha_{14} R_c R_1 R_2 \\ & + \alpha_{15} R_c R_1 R_2 R_3 + b \end{aligned} \tag{16}$$

where, $\mathbf{x} = [\mathrm{x}_1, \mathrm{x}_2]^T$, $R_i = ReLU(c_i^T\mathbf{x} + d_i)$, $R_c = ReLU(r^2 - (\mathbf{x} - \mathbf{x}_c)^T(\mathbf{x} - \mathbf{x}_c))$, $ReLU(z) = max(0, z)$. c_i and d_i are the coefficients for the straight lines, and $\mathbf{x}_c$ and r are the center and radius of G_{circ}^{inf}, respectively.

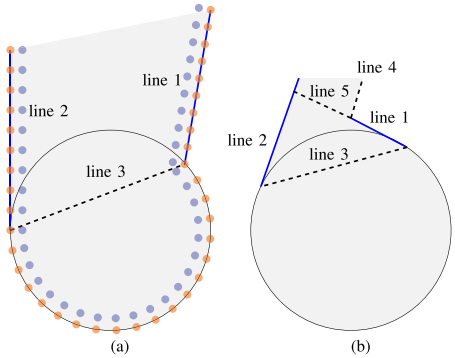

Fig. 4. Keyhole diagram with virtual lines. Blue and orange dots are safe and unsafe samples. The distance between unsafe and safe samples is small and is exaggerated for visual purposes. .

All points in the domain are mapped onto the level sets of the line and circle equations (layer 1) and their polynomial combinations (layer 2). Any input that maps to negative (level-set) values is set to zero by the ReLU function. As shown in (16) by the subscripts of R, three additional straight lines 3, 4, and 5 are added. Fig. 4(a) shows an illustrative example of the keyhole shape with line 3, which connects points $\boldsymbol{Q}_l^{\text{inf}}$ and $\boldsymbol{Q}_r^{\text{inf}}$. Lines 4 and 5 were added to address a special keyhole configuration shown in Fig. 4(b).

The synthesis process for the ZBF (i.e., training of the neural network) employs a linear program (LP) construction technique [45]. The LP-based synthesis formulation requires two sets of unsafe and safe sample points, $\mathcal{X}^u$ and $\mathcal{X}^s$, respectively. As shown in Fig. 4(a), the unsafe points (orange) are sampled along the inflated gap sides and egocircle edge, excluding the arc between the gap sides. The safe points (blue) are generated from the unsafe points by pushing them along the gradient inwards a small ϵ distance, e.g., 3% of the circle radius. The linear program for learning outer weights α_i and bias b of the neural network is

$$\begin{aligned} \min_{\alpha,b} \quad & \vec{1}^T\alpha - b \\ \text{s.t.} \quad & h(\mathbf{x}_i) \leq 0,\ \forall x_i \in \mathcal{X}^u \\ & h(\mathbf{x}_j) \geq 1,\ \forall x_j \in \mathcal{X}^s \\ & b \leq 0,\ \alpha_k \geq 0,\ \forall k = 1,\ldots,15 \end{aligned} \tag{17}$$

where $\vec{1} = [1,\ldots,1]^T$ and $\alpha = [\alpha_1,\ldots,\alpha_{15}]^T$. The constant value 1 in the safe point constraints is arbitrary and affects the scaling of the weights, similar to support vector machines. The choice of the cost function is not unique, as the goal of the LP is to determine α_i and b that satisfy the constraints. Though a cost-free LP will generate a feasible solution, most LP software packages take longer to converge for cost-free programs. Consequently, α_i and b were added to the cost function, which forms a L_1 regulation penalty due to the positivity and negativity constraints on α_i and b. As a result, not only does the LP compute time improve, but the solutions are sparse, which allows for pruning the neural network to improve compute time.

D. NMPC Trajectory Tracking

To address control constraints, an NMPC problem is defined to compute a safe tracking control for the planar nonholonomic robot based on the composite Bézier path $\mathcal{P}^*$ from §III-B. Assume a unicycle nonholonomic model with state $\mathbf{x} = [\mathrm{x}_1, \mathrm{x}_2, \theta]^T$ and control $\mathbf{u} = [\nu, \omega]^T$,

$$\begin{bmatrix} \dot{\mathrm{x}}_1 \\ \dot{\mathrm{x}}_2 \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \nu \\ \omega \end{bmatrix}. \tag{18}$$

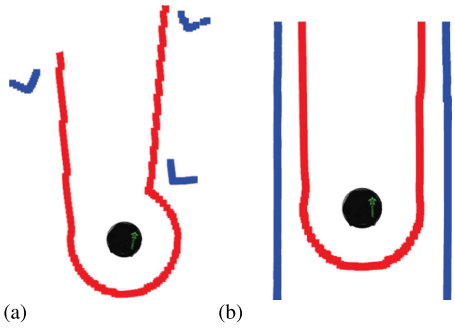

Fig. 5. *Keyhole ZBF* examples. Blue shapes are obstacles. Red boundaries are *Keyhole ZBF* modeling $\mathcal{F}^{\text{inf}}$. (a) *Keyhole ZBF* synthesis with scattered obstacles; (b) *Keyhole ZBF* synthesis in a hallway.

$\mathcal{P}^*$ is potentially–but not confirmed to be–dynamically feasible under control constraints. The near-identity trajectory [48] is utilized to create a nonholonomic time-varying trajectory $\mathbf{x}_{\text{ref}}(t)$ given the path $\mathcal{P}^*$ and a desired linear velocity ν_d. The time stamps and the velocity profile $\mathbf{u}_{\text{ref}}$ are assigned to the dynamically feasible trajectory reference. However, $\mathbf{x}_{\text{ref}}(t)$ may deviate from the original Bézier path $\mathcal{P}^*$ and impact safety. Safety constraints are necessary. Naively using distance as soft constraints with obstacles/raw points leads to slow computation and large memory consumption when the environment has a large quantity of obstacles/raw points [16], [37]. The same holds for hard constraints. The *Keyhole ZBF* is a single function to model collision-free regions, see Fig. 5, and serve as a safety constraint in NMPC. The scheme is formulated at current time t with the initial state $\mathbf{x}(t)$ and control $\mathbf{u}(t)$:

$$\begin{aligned} \min_{\mathbf{u}(t),\ldots,\mathbf{u}(t+N-1)} \quad & J(t) = \sum_{k=0}^{N} ||\mathbf{x}(t+k) - \mathbf{x}_{\text{ref}}(t+k)||_Q \\ & \quad + \sum_{k=0}^{N-1} ||\mathbf{u}(t+k) - \mathbf{u}_{\text{ref}}(t+k)||_R \\ \text{s.t.} \quad & \mathbf{x}(t+k+1) = f(\mathbf{x}(t+k), \mathbf{u}(t+k)) \\ & \mathbf{u}_{lb} \leq \mathbf{u}(t+k) \leq \mathbf{u}_{ub} \\ & \boldsymbol{a}_{lb} \leq |\mathbf{u}(t+k+1) - \mathbf{u}(t+k)|/\Delta t \leq \boldsymbol{a}_{ub} \\ & h(\mathbf{x}(t+k)) \geq 0 \end{aligned} \tag{19}$$

where $||\mathbf{m}||_A = \mathbf{m}^T A \mathbf{m}$. $\mathbf{u}_{lb}$, $\boldsymbol{a}_{lb}$, $\mathbf{u}_{ub}$, and $\boldsymbol{a}_{ub}$ are the lower and upper bounds of velocities and accelerations to maintain smooth motions. N is the number of time steps in the prediction horizon. Q and R are the state and control weights. $h(\mathbf{x})$ is the *Keyhole ZBF*, which represents the inflated collision-free space $\mathcal{F}^{\text{inf}}$. For some environmental configurations, $\mathcal{F}^{\text{inf}}$ is a non-convex region. Directly using G_{circ}^{inf} and G_{poly}^{inf} is impossible to model the entire collision-free space. The *Keyhole ZBF* provides a consistent function to describe both convex and non-convex scenarios. It reflects safety and traversability for gap-based nonholonomic navigation. Feasible solutions are highly likely to be collision-free local paths, subject to time-discretization artifacts.

E. Projection Operator

In some situations, the NMPC solver with *Keyhole ZBF* constraint may not be feasible *in continuous time* or fail to converge to a solution within the prescribed time (empirically, ∼4% of optimizations). For the latter, the control from the previous optimized NMPC horizon is applied. In both cases, the safety guarantee is void. The projection operator from [2] serves as a safety filter to prevent collisions; it provides a similar correction as a CBF [49]. Define variables for nonholonomic case, $\hat{\mathbf{x}} = [\mathbf{x}^{(1)}, \mathbf{x}^{(2)}]^T = [\mathrm{x}_1, \mathrm{x}_2]^T$ and $\hat{\mathbf{u}} = [\nu\cos(\theta), \nu\sin(\theta)]^T$. The reshaped control $\mathbf{u}'$ is.

$$\mathbf{u}' = \begin{cases} \mathbf{u} & \text{if } \psi(\hat{\mathbf{x}}) < 0 \\ \mathbf{u} & \text{if } \psi(\hat{\mathbf{x}}) \geq 0 \ \wedge \ g(\hat{\mathbf{x}}, \hat{\mathbf{u}}) > 0 \\ \mathbf{u} - \boldsymbol{\gamma}\psi(\hat{\mathbf{x}}) g(\hat{\mathbf{x}}, \hat{\mathbf{u}}) f(\hat{\mathbf{x}}) & \text{if } \psi(\hat{\mathbf{x}}) \geq 0 \ \wedge \ g(\hat{\mathbf{x}}, \hat{\mathbf{u}}) \leq 0 \end{cases}$$

$$\text{where} \quad f(\hat{\mathbf{x}}) = \frac{\nabla\psi(\hat{\mathbf{x}})}{\|\nabla\psi(\hat{\mathbf{x}})\|}, \quad g(\hat{\mathbf{x}}, \hat{\mathbf{u}}) = \langle f(\hat{\mathbf{x}}), \hat{\mathbf{u}} \rangle \tag{20}$$

The potential function $\psi(\hat{\mathbf{x}})$ [2] is the distance to the nearest obstacle point within the egocircle region. A higher value represents proximity to an obstacle, while a negative potential indicates farness from obstacles. It constructs a smooth safety boundary. The controls are reshaped when the robot is close enough to the boundary and moving towards it. $\boldsymbol{\gamma}$ represents a tunable diagonal gain matrix. When active, the robot will slow down and steer away from obstacles. The projection operator is also active when G^{inf}_{circ} is too small, causing the robots to move from dead zones and improve planning traversability.

IV. Experiments

A. Ablation Study in Simulation

1) Configurations: Benchmarking the *Safer Gap* local planner in ROS utilizes the move_base hierarchical navigation system. An ablation study is performed on STDR and Gazebo simulators. STDR uses a 1st order circular nonholonomic model with a 360° Field-of-View (FoV) laser scanner. Gazebo employs a Turtlebot as the 2nd order nonholonomic mobile robot with a limited 60° FoV Kinect depth camera. There are four benchmarking scenarios [1]: sector, dense, campus, and office. They simulate multiple navigation environments, e.g., hallways, open areas with obstacles, campus roads, etc. Unknown obstacles are randomly spawned with a 1m minimum distance between each other. The robot's start and end poses are also randomly chosen in designated areas. Ground truth robot poses are used for navigation. 500 Monte Carlo runs were initially run for each planner to quantitatively compare navigation performance.

Table I summarizes the configuration of the seven tested planners. PG is the potential gap with radial extension and tuned feedback gains [2], while CB is the composite Bézier path planner with the same feedback following controller. To adapt to different kinematics, a trajectory synthesis module is added to generate time-varying trajectories. PG+M and CB+M use NMPC with *Keyhole ZBF* for trajectory tracking. PO adds the projection operator as a final safety module. A custom recovery behavior is triggered to safely move the robot away from the nearest obstacles when the planner cannot generate a solution. SG is the proposed work *Safer Gap* and includes all modules. PG+PO does not have the custom recovery behavior to preserve the original implementation [2]. Details of the recovery behavior and parameters used can be found in the public repository [50].

TABLE I
Configurations of all Planners in the Ablation Study

Planners	Path Planning	Trajectory Synthesis†	Tracking Method	Projection Operator*	Move Recovery‡
PG	P	N	PF	N	Y
CB	B	N	PF	N	Y
PG+M	P	Y	MT	N	Y
CB+M	B	Y	MT	N	Y
PG+M+PO	P	Y	MT	Y	Y
SG	B	Y	MT	Y	Y
PG+PO	P	N	PF	Y	N

P: Potential Field B: Composite Bézier Path Y: Yes N: No
PF: Pose-based Path Following
MT: NMPC Trajectory Tracking with *Keyhole ZBF*
† Generate time-varying trajectory with desired velocities.
* PG+PO uses projection operator (PO) in every planning loop; PG+M+PO and SG only need PO if NMPC cannot converge within prescribed time.
‡ Recovery behavior moves robots away from nearest obstacles.

2) Evaluation Metric: Success, abort, and collision rates are collected to represent navigation performance. Success: the robot can reach the goal; Abort: the robot cannot generate any motion plans to reach the local goals after invoking a recovery behavior. Collision is counted whenever it occurs.

3) Simulation Results: Fig. 6 presents the results of the ablation study. Per Table I, PG and CB in the first block employ pose feedback controllers with tuned path following gains. Among these, CB achieves significantly lower collision rates (0.4% and 0.2%) compared to PG (2.4% in STDR and 2.6% in Gazebo). This indicates that the composite Bézier path planning stage of Safer Gap better accommodates nonholonomic dynamics and enhances safety over the potential vector field approach used in Potential Gap. However, feedback controllers have limitations in explicitly assigning desired robot velocities. Adjusting gains may result in robot velocities that meet desired values but compromise navigation performance.

Integrating time-varying trajectory synthesis, as discussed in Section III-D, allows for the assignment of desired robot velocities and smooth motions. The inclusion of NMPC with a *Keyhole ZBF* constraint, denoted as PG+M and CB+M in the second block, ensures safe tracking of time-varying trajectories while meeting velocity requirements. This integration optimizes planning and control strategies to improve safety and traversability. It is important to note that PG+M and CB+M cannot be directly compared with methods in the first block, as they represent two distinct navigation schemes (i.e., trajectory tracking and path following). According to the results, CB+M exhibits lower collision and abort rates than PG+M in both simulators. Failures typically occur when NMPC or Keyhole synthesis cannot find solutions within the prescribed time (around 4% of invocations), leading to collisions in the absence of a safety guard. Abort scenarios also arise when NMPC fails to generate any motion after all recovery behaviors are exhausted. The conflicting demands between trajectory tracking and obstacle avoidance in PG+M result in higher failure rates for NMPC and Keyhole synthesis, which in turn increases collision and abort rates. Therefore, the consistent design of Keyhole safe regions in both the path planning and trajectory tracking stages of *Safer Gap* outperforms the simpler combination of PG and NMPC.

To improve collision avoidance, a PO safety filter is incorporated in case optimization fails, completing the design of *Safer*

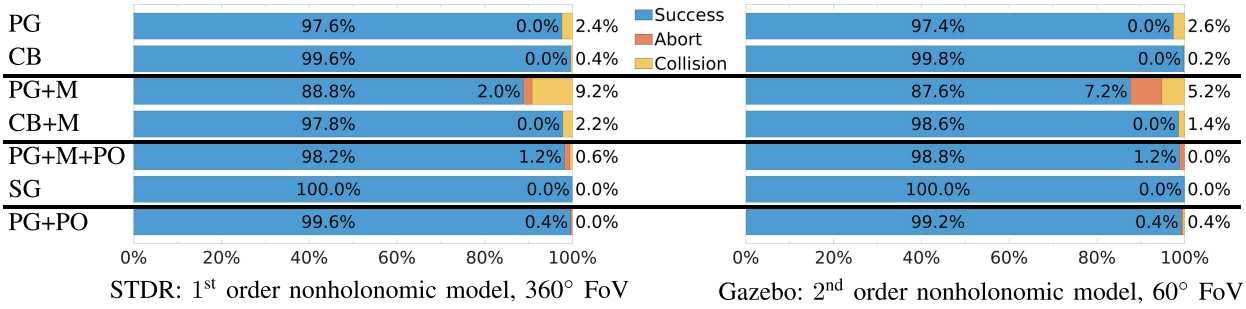

Fig. 6. Ablation study results in STDR and Gazebo simulators.

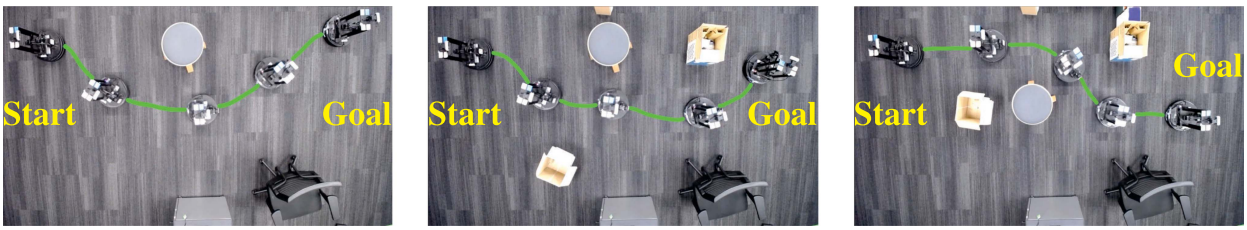

Fig. 7. Real experiment top view. Three environment densities are shown. The green paths are real robot traces. It starts from the left to the right.

Gap. It "pushes" away from obstacles and helps the robot avoid problematic dead zones, thereby reducing abort rates. Compared to PG+M+PO, SG achieves 0% collision and abort rates in both simulators, demonstrating enhanced safety and traversability. Due to the low failure rates, STDR and Gazebo benchmark runs were extended to 4500 and 3000, respectively. SG maintained 0% collision rates, 99.9% and 100% success rates, and 0.09% abort rates in STDR, while PG+PO had collision rates of 0.11% and 0.17%, and abort rates of 0.18% and 0.07% in STDR and Gazebo. Fisher's exact test [51] reveals p-values of 0.0311 for the collision rates between the two methods, indicating a statistically significant difference. SG outperforms PG+PO.

4) Computational Efficiency: Google OR-Tools [52] solves the *Keyhole ZBF* LP of (17). A timing study is performed on an Intel E5-2680 workstation. CasADi optimization framework with IPOPT is used to solve NMPC. The horizon number is set to $N = 6$. The total time of each control loop is averaged as 69 ms with a standard deviation (std) of 30, including composite Bézier path synthesis (mean: 13 ms, std: 5), keyhole generation (mean: 1.6 ms, std: 1.4), NMPC optimization (mean: 53 ms, std: 29), and projection operator (mean: 6 μs, std: 0.03). Real-time application is achievable. Tests of the state-of-the-art solver ACADOS [53] on the proposed NMPC problem had a processing time average of 75 ms (std: 25). It also supports real-time implementation, here defined to be at least 5 Hz.

B. Real Experiments

The simulation ablation study quantitatively compares the navigation performance of *Safer Gap* with *Potential Gap*, and shows that the former achieves the desired, collision-free outcomes. This section reports the results of the *Safer Gap* implementation on a real platform, a LoCoBot, for navigation through unknown environments. The robot base is differential drive. The robot's odometry provides pose information. The depth images from a RealSense D435i depth camera are converted to laserscan measurements [54].

Testing involves five scenarios with varying obstacle densities, ranging from low to high (see Fig. 7). Top-view figures depict the traces of robot navigation. Two runs are repeated for each configuration to demonstrate consistent results. A 100% success rate across the 10 trials provides evidence that the *Safer Gap* local planner supports navigation safety and traversability for (nonholonomic) differential drive robots.

V. Conclusion

The *Safer Gap* local planner is a constructive safe navigation policy for differential drive mobile robots. It generates guaranteed safe, kinematically valid, composite Bézier paths within collision-free, keyhole regions defined by gaps. NMPC tracks the reference paths under constrained nonholonomic (differential drive) control with the synthesized *Keyhole ZBF* as a safety constraint. A projection operator enhances safety and traversability at the lowest navigation level when the optimizations fail to converge in time. The full local planner construction ensures forward invariance within a collision-free region. Ablation studies and real experiments confirm safe navigation without compromising gap traversability. Future work includes testing different robot dynamics and environment complexities for robustness. The current design ensures safety for static obstacles only, leaving open the extension to moving obstacles.

References

[1] J. S. Smith, S. Feng, F. Lyu, and P. A. Vela, *Real-Time Egocentric Navigation Using 3D Sensing*. Cham, Switzerland: Springer, 2020, pp. 431–484.
[2] R. Xu, S. Feng, and P. A. Vela, "Potential gap: A gap-informed reactive policy for safe hierarchical navigation," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 8325–8332, Oct. 2021.
[3] S. Kousik, S. Vaskov, F. Bu, M. Johnson-Roberson, and R. Vasudevan, "Bridging the gap between safety and real-time performance in receding-horizon trajectory design for mobile robots," *Int. J. Robot. Res.*, vol. 39, no. 12, pp. 1419–1469, 2020.

[4] S. Feng, Z. Zhou, J. S. Smith, M. Asselmeier, Y. Zhao, and P. A. Vela, "GPF-BG: A hierarchical vision-based planning framework for safe quadrupedal navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 1968–1975.

[5] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.

[6] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[7] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1994, pp. 3310–3317.

[8] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 354–363, Jun. 2005.

[9] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, "Anytime search in dynamic graphs," *Artif. Intell.*, vol. 172, no. 14, pp. 1613–1643, 2008.

[10] L. Kavraki, M. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Trans. Robot.*, vol. 14, no. 1, pp. 166–171, Feb. 1998.

[11] R. Bohlin and L. Kavraki, "Path planning using lazy PRM," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, vol. 1, pp. 521–528.

[12] E. Frazzoli, M. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," in *Proc. Amer. Control Conf.*, 2001, vol. 1, pp. 43–49.

[13] M. W. Otte and E. Frazzoli, "RRTX: Real-time motion planning/replanning for environments with unpredictable obstacles," in *Proc. Workshop Algorithmic Found. Robot.*, 2014, pp. 461–478.

[14] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1993, pp. 802–807.

[15] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.

[16] C. Rösmann, F. Hoffmann, and T. Bertram, "Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control," in *Proc. Eur. Conf. Comput.*, Jul. 2015, pp. 3352–3357.

[17] J. S. Smith, R. Xu, and P. Vela, "egoTEB: Egocentric, perception space navigation using timed-elastic-bands," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 2703–2709.

[18] V. Sezer and M. Gokasan, "A novel obstacle avoidance algorithm:follow the gap method," *Robot. Auton. Syst.*, vol. 60, no. 9, pp. 1123–1134, Jul. 2012.

[19] M. Mujahad, D. Fischer, B. Mertsching, and H. Jaddu, "Closest gap based (CG) reactive obstacle avoidance navigation for highly cluttered environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2010, pp. 1805–1812.

[20] M. Mujahed, H. Jaddu, D. Fischer, and B. Mertsching, "Tangential closest gap based (TCG) reactive obstacle avoidance navigation for cluttered environments," in *Proc. IEEE Int. Symp. Saf., Secur. Rescue Robot.*, 2013, pp. 1–6.

[21] M. Mujahed, D. Fischer, and B. Mertsching, "Safe gap based (SG) reactive navigation for mobile robots," in *Proc. 2013 IEEE Eur. Conf. Mobile Robots*, 2013, pp. 325–330.

[22] M. Mujahed and B. Mertsching, "The admissible gap (AG) method for reactive collision avoidance," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 1916–1921.

[23] H. Chen, S. Feng, Y. Zhao, C. Liu, and P. A. Vela, "Safe hierarchical navigation in crowded dynamic uncertain environments," in *Proc. IEEE 61st Conf. Decis. Control*, 2022, pp. 1174–1181.

[24] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1985, pp. 500–505.

[25] J. Barraquand, B. Langlois, and J.. Latombe, "Numerical potential field techniques for robot path planning," *IEEE Trans. Syst., Man, Cybern.*, vol. 22, no. 2, pp. 224–241, Mar./Apr. 1992.

[26] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Trans. Robot. Autom.*, vol. 16, no. 5, pp. 615–620, Oct. 2000.

[27] M. Park, J. Jeon, and M. Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," in *Proc. 2001 IEEE Int. Symp. Ind. Electron. Proc.*, 2001, vol. 3, pp. 1530–1535.

[28] G. Li, A. Yamashita, H. Asama, and Y. Tamura, "An efficient improved artificial potential field based regression search method for robot path planning," in *Proc. 2012 IEEE Int. Conf. Mechatron. Autom.*, 2012, pp. 1227–1232.

[29] J. wung Choi, R. E. Curry, and G. H. Elkaim, "Path planning based on Bézier curve for autonomous ground vehicles," in *Proc. IAENG World Congr. Eng. Comput. Sci.*, 2008, pp. 158–166.

[30] M. Elhoseny, A. Tharwat, and A. E. Hassanien, "Bézier curve based path planning in a dynamic field using modified genetic algorithm," *J. Comput. Sci.*, vol. 25, pp. 339–350, 2018.

[31] H. A. Satai, M. M. A. Zahra, Z. I. Rasool, R. S. Abd-Ali, and C. I. Pruncu, "Bézier curves-based optimal trajectory design for multirotor uavs with any-angle pathfinding algorithms," *Sensors*, vol. 21, no. 7, 2021, Art. no. 2460.

[32] L. Zheng, P. Zeng, W. Yang, Y. Li, and Z. Zhan, "Bézier curve-based trajectory planning for autonomous vehicles with collision avoidance," *IET Intell. Transp. Syst.*, vol. 14, no. 13, pp. 1882–1891, 2020.

[33] V. Hassani and S. V. Lande, "Path planning for marine vehicles using Bézier curves," *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 305–310, 2018.

[34] B. Ingersoll, J. Ingersoll, P. DeFranco, and A. Ning, "UAV path-planning using Bézier curves and a receding horizon approach," in *Proc. AIAA Model. Simul. Technol. Conf.*, 2016, Art. no. 3675.

[35] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[36] T. P. Nascimento, C. E. T. Dórea, and L. M. G. Gonçalves, "Nonlinear model predictive control for trajectory tracking of nonholonomic mobile robots," *Int. J. Adv. Robot. Syst.*, vol. 15, no. 1, Jan. 2018, Art. no. 172988141876046.

[37] C. Rsmann, A. Makarow, and T. Bertram, "Online motion planning based on nonlinear model predictive control with non-Euclidean rotation groups," in *Proc. IEEE Eur. Conf. Comput.*, 2021, pp. 1583–1590.

[38] Z. Zuo et al., "MPC-based cooperative control strategy of path planning and trajectory tracking for intelligent vehicles," *IEEE Trans. Intell. Veh.*, vol. 6, no. 3, pp. 513–522, Sep. 2021.

[39] T. Ding, Y. Zhang, G. Ma, Z. Cao, X. Zhao, and B. Tao, "Trajectory tracking of redundantly actuated mobile robot by MPC velocity control under steering strategy constraint," *Mechatron.*, vol. 84, 2022, Art. no. 102779.

[40] X. Xu, J. W. Grizzle, P. Tabuada, and A. D. Ames, "Correctness guarantees for the composition of lane keeping and adaptive cruise control," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, no. 3, pp. 1216–1229, Jul. 2018.

[41] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames, "Backup control barrier functions: Formulation and comparative study," in *Proc. IEEE Conf. Decis. Control*, 2021, pp. 6835–6841.

[42] J. J. Choi, D. Lee, K. Sreenath, C. J. Tomlin, and S. L. Herbert, "Robust control barrier–value functions for safety-critical control," in *Proc. IEEE Conf. Decis. Control*, 2021, pp. 6814–6821.

[43] W. Xiao, C. G. Cassandras, and C. A. Belta, "Learning feasibility constraints for control barrier functions," in *Proc. IEEE Eur. Control Conf.*, 2023, pp. 1–6.

[44] B. Dai, P. Krishnamurthy, and F. Khorrami, "Learning a better control barrier function," in *Proc. IEEE Conf. Decis. Control*, 2022, pp. 945–950.

[45] A. Abuaish, M. Srinivasan, and P. A. Vela, "Geometry of radial basis neural networks for safety biased approximation of unsafe regions," in *Proc. IEEE Amer. Control Conf.*, 2023, pp. 1459–1466.

[46] M. Srinivasan, A. Dabholkar, S. Coogan, and P. A. Vela, "Synthesis of control barrier functions using a supervised machine learning approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 7139–7145.

[47] H. Prautzsch, W. Boehm, and M. Paluszny, *Bézier and B-Spline Techniques*. Berlin, Germany: Springer Berlin Heidelberg, 2002.

[48] R. Olfati-Saber, "Near-identity diffeomorphisms and exponential ϵ-tracking and ϵ-stabilization of first-order nonholonomic SE(2) vehicles," in *Proc. Amer. Control Conf.*, 2002, pp. 4690–4695.

[49] A. Singletary, K. Klingebiel, J. Bourne, A. Browning, P. Tokumaru, and A. Ames, "Comparative analysis of control barrier functions and artificial potential fields for obstacle avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 8129–8136.

[50] Safer gap repository, 2024. [Online]. Available: https://github.com/ivaROS/SaferGap

[51] E. H. Giannini, "Chapter 6 - design, measurement, and analysis of clinical investigations," in *Textbook of Pediatric Rheumatology*, 5th ed. Philadelphia: W.B. Saunders, 2005, pp. 142–173.

[52] L. Perron and V. Furnon, "OR-Tools," Google. [Online]. Available: https://developers.google.com/optimization/

[53] R. Verschueren and et al., "Acados–a modular open-source framework for fast embedded optimal control," *Math. Program. Comput.*, vol. 14, no. 1, pp. 147–183, 2022.

[54] depthimage_to_laserscan - ros wiki, 2020. [Online]. Available: http://wiki.ros.org/depthimage_to_laserscan