

Safer Gap: A Gap-based Local Planner for Safe Navigation with Nonholonomic Mobile Robots

Shiyu Feng^{1,†}, Ahmad Abuaish^{2,†}, Patricio A. Vela²

Abstract—This paper extends the gap-based navigation technique in *Potential Gap* by guaranteeing safety for nonholonomic robots for all tiers of the local planner hierarchy, so called *Safer Gap*. The first tier generates a Bézier-based collision-free path through gaps. A subset of navigable free-space from the robot through a gap, called the keyhole, is defined to be the union of the largest collision-free disc centered on the robot and a trapezoidal region directed through the gap. It is encoded by a shallow neural network zeroing barrier function (ZBF). Nonlinear model predictive control (NMPC), with *Keyhole ZBF* constraints and output tracking of the Bézier path, synthesizes a safe kinematically-feasible trajectory. Low-level use of the *Keyhole ZBF* within a point-wise optimization-based safe control synthesis module serves as a final safety layer. Simulation and experimental validation of *Safer Gap* confirm its collision-free navigation properties.

I. INTRODUCTION

The local planner module in a hierarchical navigation system processes sensory data to perceive the local environment and represent it as needed by planning and control algorithms. Gaps are such representations, which were shown to support collision-free navigation of idealized robots based on artificial potential fields (APF) [1]. The method, called *Potential Gap* generates safe trajectories guiding the robot through its ego-centric free space toward local gap-based goals. Safety guarantees hold for point-mass holonomic robots but not for nonholonomic robots. Additional robustification mechanisms were added to improve forward invariance of the safe space prescribed by the APF. Formal safety guarantees were lost and passage through gaps was compromised. This paper extends gap-based local planning safety guarantees to nonholonomic models.

Safety relies on adequately modeling the collision-free space of the robot. Fig. 2 depicts such a region specialized to the gap between two obstacles. It is represented by the union of the largest robot-centered disc and the region between the gap lines. Also depicts is the inflated free area, in dark gray, due to the robot’s physical footprint. To use zeroing barrier function (ZBF) safe control synthesis, the free space must be represented by a continuously differentiable implicit function positive inside the safe region and negative outside (the zero

level-set defines the boundary). As safe region resembles the shape of a “keyhole”, we will call the ZBF a *Keyhole ZBF*.

The hierarchical navigation system, depicted in Fig. 1, relies on a global planner to generate a candidate path to the goal in the world map. The *Safer Gap* local planner reacts to local unknown environments and achieves obstacle avoidance. First, a smooth collision-free path based on Bézier curves is generated from detected gaps. If the Bézier curve control points are contained in the *Keyhole ZBF* boundary, then the curve is also contained in the same region [2]; thus, the path is guaranteed to be collision-free. Nonlinear model predictive control (NMPC) is applied to track that path under kinematic feasibility constraints. To guarantee safe path tracking, the *Keyhole ZBF* is transcribed as a hard constraint in NMPC. Finally, as a last measure of safety to cope with the rapidly changing environment and the slower rate of NMPC, the *Keyhole ZBF* is enforced in a control BF (CBF) point-wise optimization safe control synthesis.

The contributions and organization of *Safer Gap* as annotated in Fig. 1 (dashed) are: **(i)** Joined Bézier path generation (§III-A) guarantees safety and kinematic passibility through gaps. **(ii)** The real-time synthesized *Keyhole ZBF* (§III-C) models the safe keyhole region. NMPC trajectory tracking (§III-B) with *Keyhole ZBF* constraint and nonholonomic dynamics synthesizes safe trajectories. **(iii)** The *Keyhole ZBF* acts as a CBF (§III-D) to ensure safety at the lowest level. Simulation benchmarking and real experiments in §IV confirm *Safer Gap*’s collision-free properties.

II. RELATED WORK

A. Vision-based Planning

Vision-based navigation in unknown environments has become popular nowadays. Perception acts as the first module in navigation frameworks. It is important to efficiently and accurately describe environments for safe path planning and control. Perception can be generally categorized into allocentric and ego-centric approaches [3]. Within the human’s neural hierarchy, ego-centric processing usually happens before allocentric estimation. It needs fewer memory resources and is computationally efficient in depicting the local region.

Gap is one of the ego-centric methods that can model locally free space and leverage line-of-sight visibility. Although there is no formal definition of the “gap”, it is usually a segment of 1D laserscan measurement, and consists of starting and ending points to represent the collision-free region [1], [4]–[9]. Path planning with gaps should be able to improve safety and passibility. Especially, *Potential*

*This work was supported in part by NSF Award #1849333, by DARPA PAI, and by KACST Fellowship.

† Equal contribution

¹S. Feng is with the School of Mechanical Engineering and the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30308, USA. shiyufeng@gatech.edu

²A. Abuaish, and P.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. {aabuaish, pvela}@gatech.edu

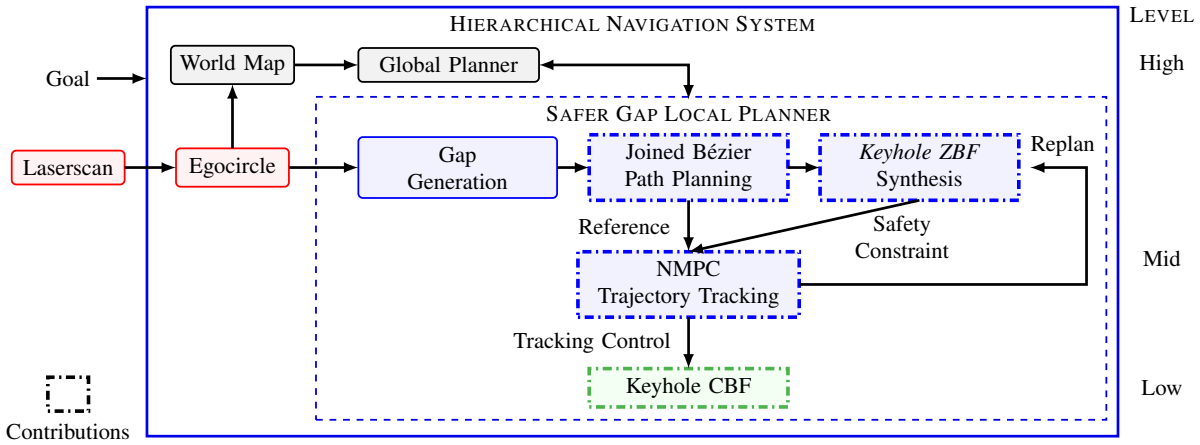


Fig. 1: Hierarchical navigation system with Safer Gap local planner. Red blocks are perception module to generate egocircle. Blue blocks are planning modules. Green block is the control module.

Gap local planner [1] proves safety guarantee for point-mass holonomic robot models. It includes radial extension and projection operator to enhance safety for nonholonomic mobile robots. However, the closed-form proof does not hold.

After modeling environments, the next step is planning a collision-free path within the perception space. Dijkstra’s algorithm, A*, D*, D* Lite, and AD* [10]–[14] are graph search planning methods in costmaps. PRM, RRT, and RRT-X [15]–[18] are sample-based planners that search for collision-free paths in the workspace. These are generally used as global planners. Reactive policies such as EB, DWA, TEB and egoTEB [9], [19]–[21] generate local plans to avoid obstacles and approach to goals.

In addition, artificial potential field (APF) offers fast computation for obstacle avoidance [22]–[26]. *Potential Gap* local planner [1] uses APF to synthesize safety guaranteed paths for point-mass holonomic models. Also, Bézier curve is a well-known technique to generate smooth trajectories for different robot models [27]–[32]. Our recent work, Bézier Gap [33], leverages the property of Bézier curve to synthesize safe trajectories for holonomic quadrupedal robots. However, there is no safety guarantee for nonholonomic models in planning. Consequently, it motivates the research to leverage the advantages of gaps and Bézier curves for nonholonomic path planning.

B. Nonlinear Model Predictive Control

Model predictive control (MPC) is a receding-horizon controller that can be used for both generating or tracking trajectories [34]. For the latter, trajectories are typically generated by a different module that uses a simplified model of the system to allow for fast trajectory generation. In particular, [35], [36] demonstrates the efficacy of NMPC for path and trajectory tracking for nonholonomic robots, supporting the utility of NMPC for local planning. Further, enforcing safety constraints in a MPC fashion leads to unconservative safe trajectories.

C. Safety and Control

In recent years, safety in control systems has been incorporated via invariant set theory analysis. To ensure safety

for a particular set, e.g., free space set \mathcal{F} in navigation applications, one needs to prove that \mathcal{F} is an invariant set. The safe set is typically represented by the zero sublevel set of a continuously differentiable implicit function, termed barrier function, $h(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$.

$$\mathcal{F} = \{ \mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) \geq 0 \} \quad (1)$$

$$\partial\mathcal{F} = \{ \mathbf{x} \in \mathbb{R}^n \mid h(\mathbf{x}) = 0 \} \quad (2)$$

where, $\partial\mathcal{F}$ denotes the boundary of the set.

Further, for controlled systems, the control barrier function (CBF) is introduced, for which the control action of the system must render the safe set invariant. The traditional implementation of CBF-based control takes place at the last tier of the control system hierarchy in the form of a point-wise optimization problem that seeks to synthesize safe control actions by satisfying the CBF constraint while minimally deviating from the base controller output [37]. If the system dynamics are control affine, the aforementioned optimization problem will take the form of a quadratic program (QP).

$$\begin{aligned} \min_{\mathbf{u}} \quad & \| \mathbf{u} - \mathbf{u}_r \|^2 \\ \text{s.t.} \quad & \nabla h^T(\mathbf{x})f(\mathbf{x}) + \nabla h^T(\mathbf{x})g(\mathbf{x})\mathbf{u} \geq -\gamma h(\mathbf{x}) \\ & \mathbf{u}_{min} \leq \mathbf{u}_i \leq \mathbf{u}_{max}, \forall i = 1, \dots, m \end{aligned} \quad (3)$$

where, $\mathbf{u}_r \in \mathbb{R}^m$ is the output of the base controller, $f(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $g(\mathbf{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ are the system dynamics matrices, γ tunable positive parameter, and \mathbf{u}_{min} and \mathbf{u}_{max} are actuation limits.

However, CBF-based full safe trajectory synthesis has been investigated in [38] and [39]. In the former, the CBF constraint is elevated into a state differential equation facilitating the utilization of traditional control techniques, e.g., LQR and pole placement. In the latter, a discrete form of the CBF constraint in (3) is added to a discrete NMPC formulation. But the resulting safe trajectories are similar to those obtained from discrete NMPC with BF constraint with inflation of the boundaries. As a result, in this work, the *Keyhole ZBF* will be added as a discrete position constraint

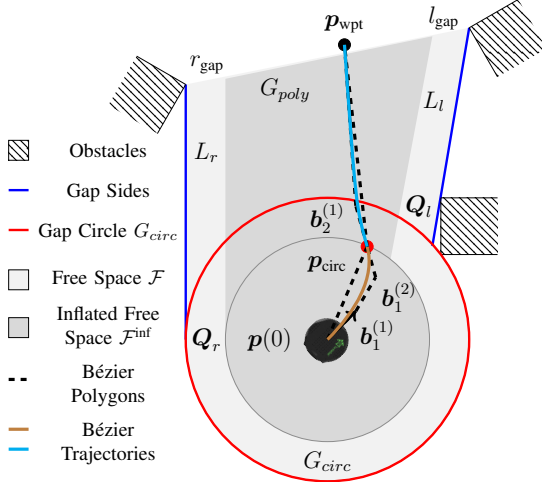


Fig. 2: Bézier trajectory synthesis. $p(0)$ is the robot origin. $b_1^{(1)}$ and $b_1^{(2)}$ are the second and third control points for the first cubic Bézier curve. Blue lines L_l and L_r are left and right gap sides. Red circle is the largest circular free space in egocircle. Q_l and Q_r are the left and right intersection points of G_{circ} and gap sides. Local waypoint p_{wpt} is inside the inflated safe region \mathcal{F}^{inf} to guarantee safety. p_{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.

in NMPC formulation for trajectory tracking, as will be discussed in the next section.

To guarantee the existence of solutions to CBF-QP in (3), $h(\mathbf{x})$ must be a valid CBF, which is usually difficult to certify and can only be done offline. Sum-of-squares is one technique to solve for the polynomial coefficients of $h(\mathbf{x})$ [40]. Similarly, Hamilton Jacobi (HJ) based reachability analysis can be used to generate a safe backward-reachable-set, which can be used as a CBF [41], [42]. Since both of those methods require offline computation, it is difficult to deploy them in changing environments, although some techniques have been proposed to synthesize safe trajectories from a library of offline-computed reachable sets [43], [44].

III. Safer Gap LOCAL PLANNER

This section introduces a gap-based local planning policy to guarantee safe navigation for nonholonomic mobile robots, so called *Safer Gap*. It incorporates line-of-sight visibility from gap detection to construct collision free space. Then safety and passibility are maintained during the design of path planning and motion control.

A. Joined Bézier Path Planning

In our previous work [1], [33], gap-based perception space and Bézier-based trajectory synthesis are demonstrated to have good navigation performance. However, safety is only guaranteed for the holonomic robot model. We propose to synthesize smooth and safe paths from gaps based on joined Bézier curves. Gap depicts the open region between two obstacles by considering robot's line-of-sight visibility, as shown in Fig. 2. It is generated from egocircle and follows the same procedure in [1]. The egocircle [3] is an ego-centric 1D array that contains spatial and temporal information of the environment.

1) *Collision-free space generation*: Similar to [33], a collision-free space \mathcal{F} is geometrically constructed for each gap. However, triangle regions constructed in [33] are too compact when one side of the robot is close to an obstacle; thus, a richer polygonal space should be created. The largest circular collision-free space G_{circ} within egocircle \mathcal{L} is found, e.g., red circle in Fig. 2. Two gap points, l_{gap} and r_{gap} , are initially connected to the tangent points of G_{circ} . The raw gap sides are formulated. To be noticed, the tangent point corresponding to l_{gap} always has smaller polar angle than r_{gap} in the robot local frame. If raw gap sides are obstructed by other obstacles, inward rotations are applied about l_{gap} (clockwise) and r_{gap} (counter clockwise) until there is no obstruction. The maximum rotation can push the tangent point to the center of G_{circ} , which constructs a minimum collision free space \mathcal{F} , which is the same as [33]. After rotation, the gap sides, L_l and L_r are finalized. We define the intersections between L and G_{circ} are Q_l and Q_r . Two gap points and two intersected points formulate a collision free polygon G_{poly} . The full collision free space \mathcal{F} is

$$\mathcal{F} = G_{poly} \cup G_{circ} \quad (4)$$

Considering robot geometry, \mathcal{F} is inflated as \mathcal{F}^{inf} in Fig. 2. The inflation size is a function of robot radius. Intersected points after inflation are denoted as Q_l^{inf} and Q_r^{inf} . Inflated gap circle is G_{circ}^{inf} . Any path \mathcal{P} within the inflated collision free space $\mathcal{P} \in \mathcal{F}^{inf}$ can guarantee safety for the full robot geometry.

2) *Joined Bézier curves*: The first segment is a cubic Bézier curve parameterized by u inside G_{circ}^{inf} ,

$$\mathcal{B}_1(u) = \sum_{i=0}^{n=3} \binom{n}{i} (1-u)^{n-i} u^i \mathbf{b}_1^{(i)} \quad (5)$$

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}, \quad 0 \leq u \leq 1$$

where $\mathbf{b}_1^{(i)}$ is the i th control point of \mathcal{B}_1 .

Since the gap is detected in robot local frame, robot center is used as the first control point $\mathbf{b}_1^{(0)} = p(0)$. An intermediate point p_{circ} is defined on the arc between Q_l^{inf} and Q_r^{inf} , and served as the last control point $\mathbf{b}_1^{(3)}$. The other two control points are designed from initial orientation $\theta(0)$, linear velocity $v(0)$, and acceleration $\vec{a}(0)$ of the nonholonomic robot. $\mathbf{b}_1^{(1)} - \mathbf{b}_1^{(0)}$ is co-linear with the unit orientation vector $\vec{o}(0) = [\cos(\theta(0)), \sin(\theta(0))]$. Curve velocities and accelerations are obtained from the first and second derivatives of cubic Bézier curve,

$$\dot{\mathcal{B}}_1(u) = 3 \sum_{i=0}^2 \binom{2}{i} (1-u)^{2-i} u^i (\mathbf{b}_1^{(i+1)} - \mathbf{b}_1^{(i)}) \quad (6)$$

$$\dot{\mathcal{B}}_1(0) = 3(\mathbf{b}_1^{(1)} - \mathbf{b}_1^{(0)}) \quad (7)$$

$$\ddot{\mathcal{B}}_1(u) = 6 \sum_{i=0}^1 \binom{1}{i} (1-u)^{1-i} u^i (\mathbf{b}_1^{(i+2)} - 2\mathbf{b}_1^{(i+1)} + \mathbf{b}_1^{(i)}) \quad (8)$$

$$\ddot{\mathcal{B}}_1(0) = 6(\mathbf{b}_1^{(2)} - 2\mathbf{b}_1^{(1)} + \mathbf{b}_1^{(0)}). \quad (9)$$

The curve parameter $u \in [0, 1]$ should be scaled to time $t \in [0, T_1]$ and $t = T_1 u$. The final time T_1 is estimated by $\|\mathbf{p}_{\text{circ}} - \mathbf{p}(0)\|/\nu_d$, where ν_d is the robot desired linear velocity. Then the scaled Bézier path $\mathcal{B}_1^s(t) = \mathcal{B}_1(t/T_1)$, and

$$\dot{\mathcal{B}}_1^s(t) = \frac{1}{T_1} \dot{\mathcal{B}}_1\left(\frac{t}{T_1}\right) \quad (10)$$

$$\ddot{\mathcal{B}}_1^s(t) = \frac{1}{T_1^2} \ddot{\mathcal{B}}_1\left(\frac{t}{T_1}\right). \quad (11)$$

Set $\|\dot{\mathcal{B}}_1^s(0)\| = \nu(0)$, which needs $\|\mathbf{b}_1^{(1)} - \mathbf{b}_1^{(0)}\| = T_1 \nu(0)/3$. Then set $\ddot{\mathcal{B}}_1^s(0) = \ddot{\mathbf{a}}(0)$, all control points for the first Bézier path segment $\mathcal{B}_1(u)$, $u \in [0, 1]$ are uniquely defined

$$\begin{aligned} \mathbf{b}_1^{(0)} &= \mathbf{p}(0) \\ \mathbf{b}_1^{(1)} &= \mathbf{p}(0) + \frac{T_1 \nu(0)}{3} \ddot{\mathbf{a}}(0) \\ \mathbf{b}_1^{(2)} &= \frac{T_1^2}{6} \ddot{\mathbf{a}}(0) - \mathbf{b}_1^{(0)} + 2\mathbf{b}_1^{(1)} \\ \mathbf{b}_1^{(3)} &= \mathbf{p}_{\text{circ}} \end{aligned} \quad (12)$$

The second path segment is generated from a quadratic Bézier curve

$$\mathcal{B}_2(u) = (1-u)^2 \mathbf{b}_2^{(0)} + 2(1-u)u \mathbf{b}_2^{(1)} + u^2 \mathbf{b}_2^{(2)}. \quad (13)$$

where $\mathbf{b}_2^{(0)} = \mathbf{p}_{\text{circ}}$.

G1 continuity maintains a smooth connection between Bézier curves. Therefore, the direction vector $\vec{\mathbf{v}}$ should satisfy the equality:

$$\vec{\mathbf{v}} = \frac{\mathbf{b}_2^{(1)} - \mathbf{b}_2^{(0)}}{\|\mathbf{b}_2^{(1)} - \mathbf{b}_2^{(0)}\|} = \frac{\mathbf{b}_1^{(3)} - \mathbf{b}_1^{(2)}}{\|\mathbf{b}_1^{(3)} - \mathbf{b}_1^{(2)}\|} \quad (14)$$

The magnitude of $\mathbf{b}_2^{(1)} - \mathbf{b}_2^{(0)}$ is calculated by desired linear velocity ν_d . With quadratic Bézier curve and similar scale mechanism,

$$\dot{\mathcal{B}}_2(u) = 2(1-u)(\mathbf{b}_2^{(1)} - \mathbf{b}_2^{(0)}) + 2u(\mathbf{b}_2^{(2)} - \mathbf{b}_2^{(1)}) \quad (15)$$

$$\dot{\mathcal{B}}_2(0) = 2(\mathbf{b}_2^{(1)} - \mathbf{b}_2^{(0)}) \quad (16)$$

$$\dot{\mathcal{B}}_2^s(t) = \frac{1}{T_2} \dot{\mathcal{B}}_2\left(\frac{t}{T_2}\right) \quad (17)$$

where $T_2 = \|\mathbf{p}_{\text{wpt}} - \mathbf{p}_{\text{circ}}\|/\nu_d$.

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

$$\begin{aligned} \mathbf{b}_2^{(0)} &= \mathbf{p}_{\text{circ}} \\ \mathbf{b}_2^{(1)} &= \mathbf{p}_{\text{circ}} + \lambda \frac{T_2 \nu_d}{2} \vec{\mathbf{v}} \\ \mathbf{b}_2^{(2)} &= \mathbf{p}_{\text{wpt}} \end{aligned} \quad (18)$$

Local waypoint \mathbf{p}_{wpt} candidates are initially found based on global plans and then bounded by \mathcal{F}^{inf} . The intermediate point \mathbf{p}_{circ} starts with the middle point of the arc, then is biased by the relative position between $\mathbf{p}(0)$ and \mathbf{p}_{wpt} to

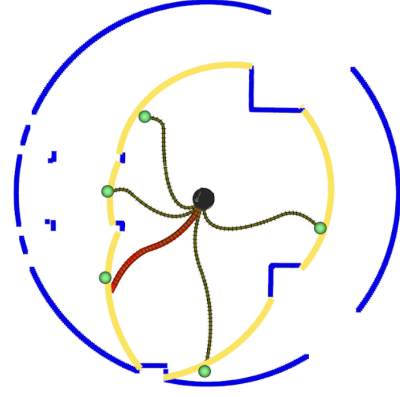


Fig. 3: Joined Bézier paths for all gaps. Blue is egocircle \mathcal{L} . Yellow are 5 detected gaps. Green points are local waypoints \mathbf{p}_{wpt} . Black paths are the synthesized Bézier paths \mathcal{P} . Red path is the selected \mathcal{P}^* based on the scoring equation.

synthesize smoother paths. If \mathbf{p}_{wpt} is within $G_{\text{circ}}^{\text{inf}}$, only first Bézier segment is computed. The final Bézier-based path is

$$\mathcal{P}(u) = \begin{cases} \mathcal{B}_1(u), & \mathbf{p}_{\text{wpt}} \in G_{\text{circ}}^{\text{inf}} \\ \mathcal{B}_1(u) \cup \mathcal{B}_2(u), & \text{otherwise} \end{cases} \quad (19)$$

From the above design, the first Bézier polygon for $\mathcal{B}_1(u)$ is always within $G_{\text{circ}}^{\text{inf}}$. The second Bézier polygon is within the convex region $G_{\text{poly}}^{\text{inf}}$. Therefore, the joined Bézier path is inside the inflated collision free space, $\mathcal{P}(u) \subseteq \mathcal{F}^{\text{inf}}$. Safety and passibility are achieved for nonholonomic robots. It only takes $\leq 2ms$ to generate path for each gap (on Intel i7-8700). The full path planning time depends on the number of detected gaps. A set of new paths are synthesized in every planning loop.

3) *Path scoring*: A scoring function is computed for each joined Bézier path to choose the best executed one \mathcal{P}^* . This function is an improved version from [1] by adding an orientation cost. The path has lower deviation from robot's orientation is preferable, since nonholonomic robots cannot suddenly turn backwards. It is also helpful to pick the correct path when the final goal point is on the other side of walls.

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

$$\text{where } C(d) = \begin{cases} c_{\text{obs}} e^{-w_2(d-r_{\text{ins}})}, & d > r_{\text{ins}} \\ 0, & d > r_{\text{max}} \\ \infty, & \text{otherwise} \end{cases}$$

$d(\mathbf{x}, \mathcal{L})$ is the distance from path pose $\mathbf{x} = [x_1, x_2, \theta]^T$ to the nearest point on egocircle \mathcal{L} . $\|\mathbf{x}_{\text{end}} - \mathbf{p}^*\|$ measures the distance between the end pose of \mathcal{P} and the local goal \mathbf{p}^* from a global plan. $|\theta_{\text{end}} - \theta(0)|$ is the angle difference between end pose and initial pose. r_{ins} and r_{max} are proportional to the robot radius to control the safe distance. w_1 , w_2 and c_{obs} are tunable weights. Each time, every best path \mathcal{P}_i^* compares with the previous executed path \mathcal{P}_{i-1}^* to decide whether switching to the new path. One example is shown in Fig. 3. The best path (red) is selected from a set of Bézier path candidates.

B. NMPC Trajectory Tracking

Safe joined Bézier path \mathcal{P}^* is generated in §III-A. In order to safely track the path for nonholonomic model, NMPC is applied. Assume the unicycle nonholonomic model with state $\mathbf{x} = [x_1, x_2, \theta]^T$ and control $\mathbf{u} = [\nu, \omega]^T$.

$$\begin{aligned}\dot{x}_1 &= \nu \cos(\theta) \\ \dot{x}_2 &= \nu \sin(\theta) \\ \dot{\theta} &= \omega\end{aligned}\quad (20)$$

In order to assign time stamps to path \mathcal{P}^* based on non-holonomic dynamics, near-identity trajectory $\mathbf{x}_{\text{ref}}(t)$ [45] is synthesized given the path and desired linear velocity ν_d . Time stamps and the velocity profile \mathbf{u}_{ref} are assigned to the dynamically feasible trajectory reference. However, it is possible to slightly deviate from the original Bézier path. NMPC with the safety *Keyhole* ZBF constraint can guarantee safety during tracking. The scheme is formulated with initial state $\mathbf{x}(t)$ and control $\mathbf{u}(t)$ at current time t :

$$\begin{aligned}\min_{\mathbf{u}(t+k)} \quad & J(t) = \sum_{k=0}^{N-1} \|\mathbf{x}(t+k) - \mathbf{x}_{\text{ref}}(t+k)\|_Q \\ & + \|\mathbf{u}(t+k) - \mathbf{u}_{\text{ref}}(t+k)\|_R \\ \text{s.t.} \quad & \mathbf{x}(t+k+1) = f(\mathbf{x}(t+k), \mathbf{u}(t+k)) \\ & \mathbf{u}_{lb} \leq \mathbf{u}(t+k) \leq \mathbf{u}_{ub} \\ & \mathbf{a}_{lb} \leq |\mathbf{u}(t+k+1) - \mathbf{u}(t+k)| \leq \mathbf{a}_{ub} \\ & h(\mathbf{x}(t+k)) \geq 0\end{aligned}\quad (21)$$

where $\|z\|_Q = z^T Q z$, and \mathbf{u}_{lb} , \mathbf{a}_{lb} , \mathbf{u}_{ub} , and \mathbf{a}_{ub} are the lower and upper bounds of velocities and accelerations to maintain smooth motions. N is the number of time step 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}^{inf} .

C. Keyhole ZBF Synthesis

The inflated collision-free space \mathcal{F}^{inf} is captured by the zero level-set of the *Keyhole ZBF*. We will use a shallow, two-layer neural network with rectified linear units (ReLU) to model the barrier function. In order to keep the network shallow and minimal, we need to leverage the geometry of keyhole shape, i.e., the straight lines and the circle. The complete expression of the *Keyhole ZBF* is

$$\begin{aligned}h(x) &= \alpha_1 R_1 + \alpha_2 R_2 + \alpha_3 R_3 + \alpha_4 R_c + \alpha_5 R_1 R_2 \\ &+ \alpha_6 R_c R_1 + \alpha_7 R_c R_2 + \alpha_8 R_c R_3 \\ &+ \alpha_9 R_1 R_2 R_3 + \alpha_{10} R_1 R_4 R_5 + \alpha_{11} R_2 R_4 R_5 \\ &+ \alpha_{12} R_c R_1 R_4 + \alpha_{13} R_c R_2 R_4 + \alpha_{14} R_c R_1 R_2 \\ &+ \alpha_{15} R_c R_1 R_2 R_3 + b\end{aligned}\quad (22)$$

Effectively, 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 point that maps onto a negative level-set is set to zero by the ReLU. As shown in (22) by the subscripts of R , three additional straight lines

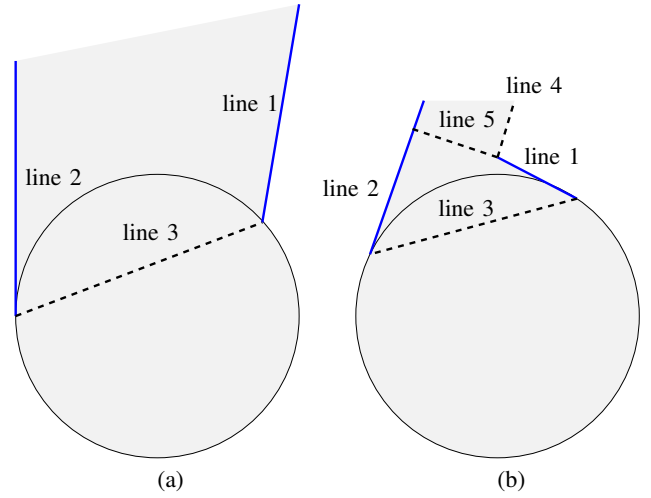


Fig. 4: Keyhole diagram with additional virtual lines.

are added, line 3, 4, and 5. Fig. 4-a shows an illustrative example of the keyhole shape with line 3, which connects points Q_l^{inf} and Q_r^{inf} . Lines 4 and 5 were added to cope with a special keyhole configuration shown in Fig. 4-b, where, $x = [x_1, x_2]^T$, $R_i = \text{ReLU}(c_i^T x + d_i)$, $R_c = \text{ReLU}(r^2 - (x - x_c)^T (x - x_c))$, $\text{ReLU}(z) = \max(0, z)$, c_i and d_i are the coefficients for the straight lines, and x_c and r are the center and radius of $G_{\text{circ}}^{\text{inf}}$, respectively.

The synthesis process for the ZBF (i.e., training of the neural network) follows the technique presented in [46], which is a linear program (LP). The LP needs sampled sets, \mathcal{X}^u and \mathcal{X}^s , from the unsafe and safe regions, respectively. The unsafe points are sampled along the gap lines and circle edge, excluding the arc between the gap lines. The safe points are generated from the unsafe point by pushing them along the gradient inwards an ϵ distance. ϵ then should be set to a small value, e.g., 3% of the circle radius. The linear program for learning α_i and b coefficients is

$$\begin{aligned}\min_{\mathbf{u}} \quad & \vec{1}^T \alpha \\ \text{s.t.} \quad & h(x_i) \leq -1, \forall i \in \{l : x_l \in \mathcal{X}^u\} \\ & h(x_j) \geq +1, \forall j \in \{l : x_l \in \mathcal{X}^s\} \\ & b \leq 0, \alpha_k \geq 0, \forall k = 1, \dots, 15\end{aligned}\quad (23)$$

where $\vec{1} = [1, \dots, 1]^T$ and $\alpha = [\alpha_1, \dots, \alpha_{15}]$. The coefficients α_i have a positivity constraint while the b has a negativity constraint. Those constraints are needed so that the cost function acts as L_1 regulation, which promotes sparsity in the solution. Also, the value constant ± 1 affects the scaling of the ZBF, much like for support vector machines. The synthesized *Keyhole ZBF* for the examples given in Fig. 4 are shown in Fig. 5.

The linear program was solved using Google OR-Tools [47] in C++. For 2000 runs on the development machine (Ubuntu 20.04, Intel i7-8750H CPU), the maximum, minimum, and average execution times were 1.6 ms, 0.71 ms, and 0.75 ms, respectively.

1) *Keyhole ZBF suitability as a CBF*: The *Keyhole ZBF* meets the requirements to be used as a CBF. It is monotonic

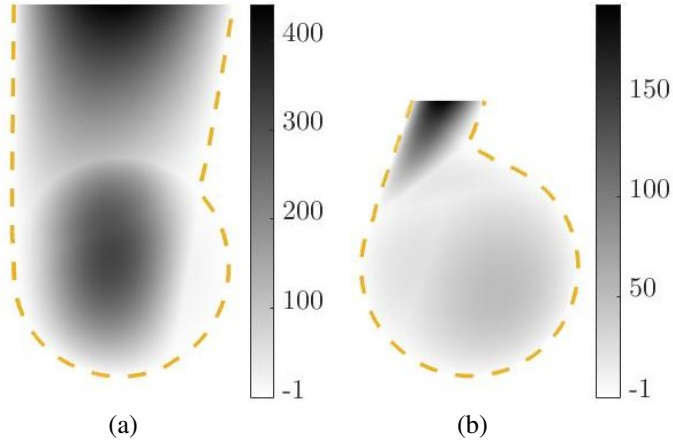


Fig. 5: *Keyhole ZBF* for configurations in Fig. 4. The value of the ZBF is represented by the color map, and the zero level-set of the ZBF is depicted by the yellow dashed line. The unsafe region outside the ZBF boundary has a negative value.

across the boundary and differentiable everywhere in the positive region. The neural network in (22) has no dead gradient, given that the terms are multiplicative combinations of the line and circle equations. Although due to the ReLU, the gradient may be non-smooth, it is not a problem for optimization, as subgradients can be used.

D. Keyhole Control Barrier Function

Since the domain of the *Keyhole ZBF* will be the position of the robot (excluding orientation), a single integrator model is assumed for the robot in CBF-QP. The reference control command is the instantaneous translational velocity of the robot, i.e., $u_r = [\nu_r \cos \theta, \nu_r \sin \theta]^T$. Again, the subscript r denotes the outputs from the reference controller, which NMPC in this case. The calculated safe velocity commands by CBF-QP (3), $u = [\dot{x}_s, \dot{y}_s]^T$, are mapped to the robot commands using (24) and (25). $\Delta\theta$ is the angle difference between the vectors u_r and u and is added to the current rotation rate to correct the angle difference. k_ω is a positive tunable parameter. The translational velocity is damped down proportional to the ratio of $\Delta\theta$ to a maximum angle θ_{max} . If $|\Delta\theta| \geq \theta_{max}$, the robot will only rotate.

$$\omega = \omega_r + k_\omega \Delta\theta \quad (24)$$

$$\nu = \max\left(0, 1 - \frac{|\Delta\theta|}{\theta_{max}}\right) \|u\| \quad (25)$$

Overall, the *Safer Gap* local planner is designed to maintain safety and passibility for nonholonomic mobile robots. From joined Bézier path planning, NMPC trajectory tracking with *Keyhole ZBF*, and control barrier function, safety guarantee is proved.

IV. EXPERIMENTS

To test navigation performance with *Safer Gap*, simulation benchmark and real robot experiments are conducted.

A. Simulation Benchmark

1) *Simulation Configuration*: We benchmarked *Safer Gap* local planner in ROS with move_base hierarchical navigation

system. The benchmark is performed in both STDR and Gazebo simulators. STDR uses 1st order circular nonholonomic model with 360° Field-of-View (FoV) laser scanner. In Gazebo, Turtlebot is used as the 2nd order nonholonomic mobile robot with limited 60° FoV. We setup four different scenarios [3] in Fig. 6 for benchmark: sector, dense, campus, and office. They simulate multiple navigation environments, e.g. hallway, open area with obstacles, campus roads and etc. The obstacles are randomly spawned in the scenarios with a minimum distance to each other as 1m. Robot start and end poses are also randomly chosen in the designated areas. Ground truth robot locations are used for navigation. We have 50 Monte Carlo runs for each scenario to quantitatively compare *Safer Gap* (SG) navigation performance with *Potential Gap* (PG) which serves as the baseline method. Radial extension and projection operator are enabled in PG for nonholonomic robot.

2) *Evaluation Metric*: Success, abort and collision rates are collected for each planner. Success means that the robot can successfully reach the goal. Abort represents that the robot cannot have any possible plan to the goal after recovery behavior. Collision is counted whenever a colliding happens.

3) *Simulation Results*: The overall simulation results of STDR and Gazebo are in Table I and II. The comparison of success rates is in Fig. 7. Both PG and SG have 100% success rates in STDR with full sensing of the environments. However, when simulating 2nd order nonholonomic model with limited FoV, PG's success rate drops 9% including 8% aborts and 1% collisions.

From the full results of 4 Gazebo scenarios in Table III, PG does not perform well in the hallway and campus roads. The robot navigates back and forth, and is stuck to find successful paths to the goal. The projection operator in PG conservatively keep safety by sacrificing passibility. Due to the limited FoV, the PG still has collisions when all nonholonomic extensions are enabled. Our proposed local planner SG not only maintains safety, but also leverages the line-of-sight visible gaps for passibility. This comparison demonstrates safety guarantee for nonholonomic mobile robots in the design of *Safer Gap*.

4) *Computational Efficiency*: The benchmark is performed on a workstation with Intel i7-8700. CasADi optimization framework is used for solving NMPC. We set the number of horizon $N = 6$. The total time of each control loop is averaged as $\sim 75ms$, including Bézier path synthesis, keyhole generation, NMPC optimization, and CBF-QP. Real time application is achievable.

B. Real Experiments

The simulation benchmark quantitatively compares *Safer Gap* navigation performance with *Potential Gap*, and presents the outperforming results of our proposed work. In this section, the planner is applied on the real platform, LoCoBot, to navigate through unknown environments. It has a Kobuki base with nonholonomic dynamics. Robot's odometry provides the pose information. RealSense D435i

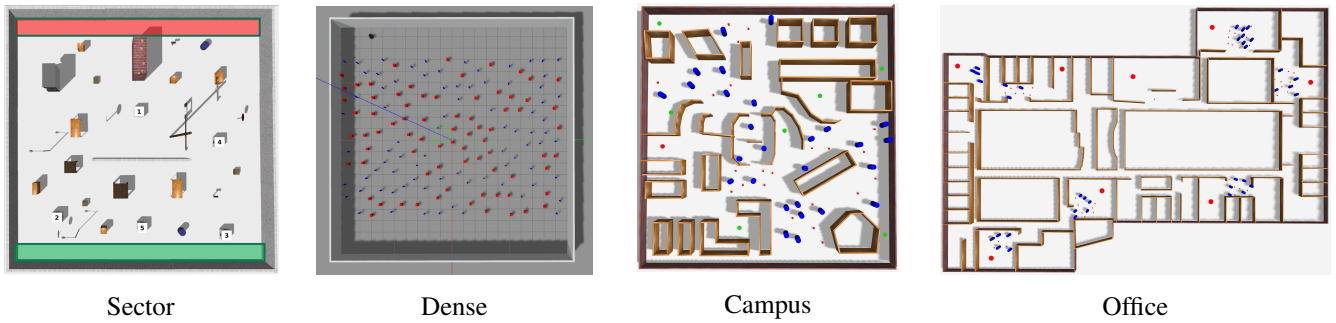


Fig. 6: Four simulation scenarios. In sector and campus worlds, start region/poses (red) and end region/poses (green) are labeled. In the dense world, robot navigates from top to bottom. In the office world, the start and goal poses are randomly chosen from the red points.

TABLE I: STDR Simulation Benchmark (1st order nonholonomic model)

Total	Success	Abort	Collision
PG	100%	0%	0%
SG	100%	0%	0%

TABLE II: Gazebo Simulation Benchmark (2nd order nonholonomic model)

Total	Success	Abort	Collision
PG	91%	8%	1%
SG	100%	0%	0%

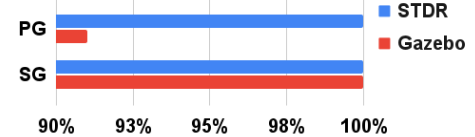


Fig. 7: Success rates comparison between PG and SG.

TABLE III: Simulation results in 4 Gazebo scenarios

	Sector			Dense			Campus			Office		
	Success	Abort	Collision	Success	Abort	Collision	Success	Abort	Collision	Success	Abort	Collision
PG	100%	0%	0%	100%	0%	0%	84%	16%	0%	80%	16%	4%
SG	100%	0%	0%	100%	0%	0%	100%	0%	0%	100%	0%	0%

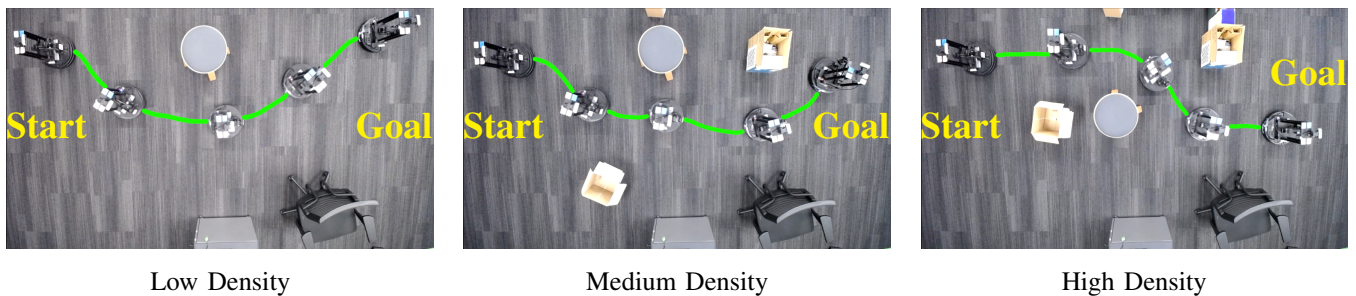


Fig. 8: Real experiment topview. Three environment densities are shown. The green paths are the robot's real traces. Robots start from left to the right.

depth camera initially produces depth images that are converted to laserscan measurements through a ROS package `depthimage_to_laserscan` [48].

We test *Safer Gap* local planner in five scenarios with different obstacle densities from low to high in Fig. 8. The robot navigation traces are depicted on the top-view figures. For each configuration, 2 runs are repeated to show consistent results. We have 100% success rate in totally 10 trials, which is the same as simulation. Therefore, *Safer Gap* local planner is applicable on the real nonholonomic robots to maintain navigation safety and passibility.

V. CONCLUSION

The proposed *Safer Gap* local planner designs a safe navigation policy for nonholonomic mobile robots. It generates smooth joined Bézier paths in the collision free space defined by gaps to guarantee safety. We performs NMPC to track the reference path considering nonholonomic dynamics. A synthesized *Keyhole ZBF* for the free space is integrated as a safety constraint to prevent collision with obstacles. At

the end, keyhole control barrier function provides additional safety at the lowest level of navigation hierarchy. From the simulation benchmark and real experiments, *Safer Gap* is demonstrated to achieve safe navigation for nonholonomic robots without losing gap passibility. In the future work, different robot dynamics and environment complexities should be tested to evaluate the robustness of *Safer Gap*.

REFERENCES

- [1] R. Xu, S. Feng, and P. A. Vela, "Potential gap: A gap-informed reactive policy for safe hierarchical navigation," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8325–8332, 2021.
- [2] H. Prautzsch, W. Boehm, and M. Paluszny, *Bézier and B-Spline Techniques*. Springer Berlin Heidelberg, 2002. [Online]. Available: <https://doi.org/10.1007/978-3-662-04919-8>
- [3] J. S. Smith, S. Feng, F. Lyu, and P. A. Vela, *Real-Time Egocentric Navigation Using 3D Sensing*. Cham: Springer International Publishing, 2020, pp. 431–484.
- [4] V. Sezer and M. Gokasan, "A novel obstacle avoidance algorithm: follow the gap method," in *RAS*, vol. 60, no. 9, July 2012, pp. 1123–1134.
- [5] M. Mujahad, D. Fischer, B. Mertsching, and H. Jaddu, "Closest gap based (cg) reactive obstacle avoidance navigation for highly cluttered environments," in *IROS*, 2010, pp. 1805–1812.

- [6] M. Mujahed, H. Jaddu, D. Fischer, and B. Mertsching, "Tangential closest gap based (tcg) reactive obstacle avoidance navigation for cluttered environments," in *SSRR*, 2013, pp. 1–6.
- [7] M. Mujahed, D. Fischer, and B. Mertsching, "Safe gap based (sg) reactive navigation for mobile robots," in *ECMR*, 2013, pp. 325–330.
- [8] M. Mujahed and B. Mertsching, "The admissible gap (ag) method for reactive collision avoidance," in *ICRA*, 2017, pp. 1916–1921.
- [9] J. S. Smith, R. Xu, and P. Vela, "egoTEB: Egocentric, Perception Space Navigation Using Timed-Elastic-Bands," in *ICRA*, 2020, pp. 2703–2709.
- [10] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, p. 269–271, dec 1959. [Online]. Available: <https://doi.org/10.1007/BF01386390>
- [11] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
- [12] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, 1994, pp. 3310–3317 vol.4.
- [13] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 354–363, 2005.
- [14] M. Likhachev, D. Ferguson, G. Gordon, A. Stentz, and S. Thrun, "Anytime search in dynamic graphs," *Artificial Intelligence*, vol. 172, no. 14, pp. 1613–1643, 2008. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S000437020800060X>
- [15] L. Kavraki, M. Kolountzakis, and J.-C. Latombe, "Analysis of probabilistic roadmaps for path planning," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 1, pp. 166–171, 1998.
- [16] R. Bohlin and L. Kavraki, "Path planning using lazy prm," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 1, 2000, pp. 521–528 vol.1.
- [17] E. Frazzoli, M. Dahleh, and E. Feron, "Real-time motion planning for agile autonomous vehicles," in *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, vol. 1, 2001, pp. 43–49 vol.1.
- [18] M. W. Otte and E. Frazzoli, "Rrtx: Real-time motion planning/replanning for environments with unpredictable obstacles," in *Workshop on the Algorithmic Foundations of Robotics*, 2014.
- [19] S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," in *[1993] Proceedings IEEE International Conference on Robotics and Automation*, 1993, pp. 802–807 vol.2.
- [20] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *RA-M*, vol. 4, no. 1, pp. 23–33, March 1997.
- [21] C. Rösmann, F. Hoffmann, and T. Bertram, "Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive control," in *ECC*, July 2015, pp. 3352–3357.
- [22] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *ICRA*, 1985, pp. 500–505.
- [23] J. Barraquand, B. Langlois, and J. Latombe, "Numerical potential field techniques for robot path planning," *T-SMC*, vol. 22, no. 2, pp. 224–241, 1992.
- [24] S. S. Ge and Y. J. Cui, "New potential functions for mobile robot path planning," *IEEE Trans. on Rob. and Aut.*, vol. 16, no. 5, pp. 615–620, 2000.
- [25] M. Park, J. Jeon, and M. Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," in *ISIE*, vol. 3, 2001, pp. 1530–1535 vol.3.
- [26] G. Li, A. Yamashita, H. Asama, and Y. Tamura, "An efficient improved artificial potential field based regression search method for robot path planning," in *ICMA*, 2012, pp. 1227–1232.
- [27] J. wung Choi, R. E. Curry, and G. H. Elkaim, "Path planning based on bézier curve for autonomous ground vehicles," *Advances in Electrical and Electronics Engineering - IAENG Special Edition of the World Congress on Engineering and Computer Science 2008*, pp. 158–166, 2008.
- [28] M. Elhoseny, A. Tharwat, and A. E. Hassanien, "Bezier curve based path planning in a dynamic field using modified genetic algorithm," *Journal of Computational Science*, vol. 25, pp. 339–350, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S187750317308906>
- [29] 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 multicopter uavs with any-angle pathfinding algorithms," *Sensors*, vol. 21, no. 7, 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/7/2460>
- [30] L. Zheng, P. Zeng, W. Yang, Y. Li, and Z. Zhan, "Bézier curve-based trajectory planning for autonomous vehicles with collision avoidance," *IET Intelligent Transport Systems*, vol. 14, no. 13, pp. 1882–1891, 2020. [Online]. Available: <https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-its.2020.0355>
- [31] V. Hassani and S. V. Lande, "Path planning for marine vehicles using bézier curves," *IFAC-PapersOnLine*, vol. 51, no. 29, pp. 305–310, 2018, 11th IFAC Conference on Control Applications in Marine Systems, Robotics, and Vehicles CAMS 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2405896318322092>
- [32] B. Ingersoll, J. Ingersoll, P. DeFranco, and A. Ning, "Uav path-planning using bezier curves and a receding horizon approach," in *AIAA Modeling and Simulation Technologies Conference*, 06 2016.
- [33] 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," to appear in 2023 IEEE International Conference on Robotics and Automation (ICRA), 2023.
- [34] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.
- [35] T. P. Nascimento, C. E. T. Dórea, and L. M. G. Gonçalves, "Nonlinear model predictive control for trajectory tracking of nonholonomic mobile robots," *International Journal of Advanced Robotic Systems*, vol. 15, no. 1, p. 172988141876046, Jan. 2018. [Online]. Available: <https://doi.org/10.1177/1729881418760461>
- [36] C. Ršmann, A. Makarow, and T. Bertram, "Online motion planning based on nonlinear model predictive control with non-euclidean rotation groups," in *2021 European Control Conference (ECC)*, 2021, pp. 1583–1590.
- [37] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.
- [38] H. Almubarak, N. Sadegh, and E. A. Theodorou, "Safety embedded control of nonlinear systems via barrier states," *IEEE Control Systems Letters*, vol. 6, pp. 1328–1333, 2022.
- [39] J. Zeng, B. Zhang, and K. Sreenath, "Safety-critical model predictive control with discrete-time control barrier function," in *2021 American Control Conference (ACC)*, 2021, pp. 3882–3889.
- [40] Y. Chen, M. Jankovic, M. Santillo, and A. D. Ames, "Backup control barrier functions: Formulation and comparative study," in *2021 60th IEEE Conference on Decision and Control (CDC)*, 2021, pp. 6835–6841.
- [41] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-jacobi reachability: A brief overview and recent advances," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 2017, pp. 2242–2253.
- [42] M. Chen and C. J. Tomlin, "Hamilton–jacobi reachability: Some recent theoretical advances and applications in unmanned airspace management," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, no. 1, pp. 333–358, May 2018. [Online]. Available: <https://doi.org/10.1146/annurev-control-060117-104941>
- [43] 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," *The International Journal of Robotics Research*, vol. 39, no. 12, pp. 1419–1469, Aug. 2020. [Online]. Available: <https://doi.org/10.1177/0278364920943266>
- [44] M. Chen, S. L. Herbert, H. Hu, Y. Pu, J. F. Fisac, S. Bansal, S. Han, and C. J. Tomlin, "Fastrack:a modular framework for real-time motion planning and guaranteed safe tracking," *IEEE Transactions on Automatic Control*, vol. 66, no. 12, pp. 5861–5876, 2021.
- [45] R. Olfati-Saber, "Near-identity diffeomorphisms and exponential /spl epsi/-tracking and /spl epsi/-stabilization of first-order nonholonomic se(2) vehicles," in *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*, vol. 6, 2002, pp. 4690–4695 vol.6.
- [46] A. Abuaish, M. Srinivasan, and P. A. Vela, "Geometry of radial basis neural networks for safety biased approximation of unsafe regions," 2022. [Online]. Available: <https://arxiv.org/abs/2210.05596>
- [47] L. Perron and V. Furnon, "Or-tools," Google. [Online]. Available: <https://developers.google.com/optimization/>
- [48] (2020) depthimage_to_laserscan - ros wiki. [Online]. Available: http://wiki.ros.org/depthimage_to_laserscan