Potential Gap: A Gap-Informed Reactive Policy for Safe Hierarchical Navigation

Ruoyang Xu[1,†], Shiyu Feng[2,†], and Patricio A. Vela[1]

Abstract—**This paper considers the integration of gap-based local navigation methods with artificial potential field (APF) methods to derive a local planning module, called *potential gap*, for hierarchical navigation systems. Central to the construction of the local planner is the use of sensory-derived local free-space models that detect gaps and use them for the synthesis of the APF. Trajectories derived from the APF are provably collision-free for idealized robot models. The provable property is lost when applied to more realistic models. A set of algorithm modifications correct for these errors and enhance robustness to non-ideal models, in particular a nonholonomic robot model. Integration of the *potential gap* local planner into a hierarchical navigation system provides the local goals and trajectories needed for collision-free navigation through unknown environments. Monte Carlo experiments in benchmark worlds confirm the asserted safety and robustness properties.**

Index Terms—**Collision Avoidance, Vision-Based Navigation, Reactive and Sensor-Based Planning**

I. Introduction

POTENTIAL methods generate guaranteed safe reactive policies for fully controlled point-mass agents going from one point in space to another, under the assumption of known collision space regions [1]. A non-trivial performance gap emerges between idealized and actual models, the latter of which occupy space, have dynamics, can be nonholonomic, and have incomplete collision space measurements (from a limited field of view) [2], [3] . A variety of other planning schemes have been derived and refined to address the performance gap [4], [5]. In the context of long-distance navigation, no single strategy provides the best outcome. Hierarchical strategies permit multiple approaches that operate at different temporal and spatial scales [4], [6], [7]. More importantly, they consider the role of visual sensors in providing up-to-date information regarding navigable space that complements existing and potentially stale free space maps. Unfortunately, more research effort goes into the design of general-purpose, single strategy planning schemes, rather than into closed-loop, sensing-informed hierarchical navigation methods.

The research deficit implies that there may be value in further study of hierarchical methods and how classical navigation approaches may by integrated into their architectures. Doing so may also suggest how to best leverage general-purpose planners within an equally general and modular hierarchical navigation framework. It also provides an excellent opportunity to reconsider how idealized approaches, or those with theoretical support but limited applicability, might provide safe local planners that operate on short temporal and small spatial scales. *For these smaller domains, the fragile theoretical guarantees could translate more robustly to non-ideal settings through targeted modifications.*

The contributions, ordered based on local planner processing per Figure 1, are: **(1)** A novel gap detection method that prioritizes line-of-sight visibility properties for establishing candidate local navigation routes (§II-A); **(2)** The explicit construction of potential field inspired flow fields for local trajectory synthesis, with provable collision avoidance in the ideal case (§II-B); and **(3)** Modular extensions that correct for non-ideal robot models, where each module addresses the loss of theoretical support and reduces sensitivity to this loss (§II-D). The local planner, called *potential gap*, inherits the navigation guarantees of potential field methods for idealized systems and demonstrates robustness to non-ideal robot models when the modular extensions are activated. Potential gap provides a framework for improved success and safety within a real-time, hierarchical navigation system.

A. Related Work and Research Context

Visual navigation has an extensive history bridging several core areas of robotics from perception to planning to action. This section narrowly reviews topics most related to the paper, from hierarchical navigation to closed-loop control.

1) Hierarchical Navigation Systems: Navigation systems for mobile vehicles must identify a solution to the high-level task of goal attainment with realizable low-level system control laws. In mobile robot navigation contexts, bipartite hierarchical systems consist of a global planner that generates paths with prior environment knowledge, and a local planner that reacts to the local information obtained from the sensors [4]. They leverage the advantages of different planners while offsetting their limitations, and have long been used in robotics as the hierarchical structure leads to higher fault tolerance and increased robustness [8]–[11]. The global planner runs at a lower rate and considers all known map information to provide a global path from a given start pose to a goal pose, can be any applicable planner [12]. In contrast, the

Manuscript received: February 24, 2021; Revised June 13, 2021; Accepted July 14, 2021.

This paper was recommended for publication by Editor Stephen J. Guy upon evaluation of the Associate Editor and Reviewers' comments.

[1] R. Xu was, and P.A. Vela is with the School of Electrical and Computer Engineering and the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30308, USA. {rxu74, pvela}@gatech.edu

[2] S. 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

† Equal contribution

*This work supported in part by NSF Award #1849333.

Digital Object Identifier (DOI): see top of this page.

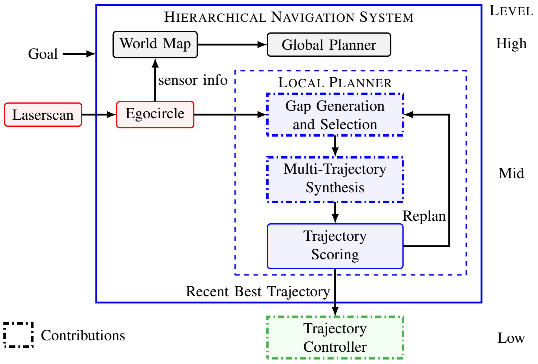


Fig. 1. Hierarchical Navigation System information flow with global planner modules (gray boxes) and local planner modules (blue boxes).

local planner executes at a higher rate to produce kinodynamically controllable commands for driving robots and avoiding obstacles. There is flexibility to select the local planner so long as it is structured to receive and assimilate sensory input into its local collision-avoiding navigation decision structure [13]–[16]. Its limited scale supports real-time, sensory-driven, collision-avoiding navigation.

2) Reactive and Potential Field Methods: The artificial potential field method and its variants are a family of planners whose instances offer simple and fast computation for mobile robot obstacle avoidance [1], [17]–[20]. While the potential field is particularly attractive due to its elegance and simplicity, there are substantial shortcomings inherent to this method such as a lack of consideration for robot kinematics, dynamics, as well as local minimum problems with regard to world geometry [2], [5]. Significant efforts were made to alleviate those problems [21], [22]. When implemented as reactive planners, the family of APFs nevertheless directly map robot state and sensor observation to available actions, offering better computational performance than deliberative planners. They share valuable traits with perception space methods, such as minimal sensor processing and planning complexity in the egocentric robot frame. *Integrating reactive methods and perception space methods leverages their fast compute properties, and the limited deliberation associated with local planning modules informed by a global planner.*

3) Perception-Space and Gap Navigation Methods: Recent work has explored the use of perception-space representations, inspired from Marr's 2.5D space, and argued in favor of mixed representation hierarchical navigation strategies [4], [16]. In particular, a local planner (limited to a short time and small spatial scale) gains computational advantages by minimally processing the sensor data and recasting local navigation as an ego-centric decision process. Related work has established similar benefits for stereo MAVs [23].

For ground vehicle navigation, the topology of the space and how to interpret it from sensor data has emerged as an important local navigation decision [16]. Gap-based processing aimed at detecting passable free-space is compatible with and improves the synthesis of local optimal paths. While there is no widely agreed-upon representation–or processing approach–for gaps, gap-based methods typically reduce 1D laser scan measurements to a set of "gaps" comprised of beginning and terminating points that represent collision-free regions in the observable space [24]–[28]. A gap-based method would then generate reactive motion commands towards a selected gap, which gives favorable performance in reasonably sparse and structured environments. However existing gap representations either rely on the presence of free space to infinity [24] or dense and accurate sensor measurements for detecting discontinuities [25], leaving the representation prone to potentially false collision free spaces and errors. Gap-based methods operate semi-deliberatively; the construction and selection of free space are intentional while execution is reactive. *There is an opportunity to connect perception space methods with potential field methods through the affordance of gaps to deliver safe planning.*

The value of gap detection in terms of establishing navigation affordances for local planning was established in [16]. The intent of this study is to explore more deeply the gap representations associated with gap-based path planning and their connection to safe navigation through local space. *Based on their use, gap-based methods will naturally connect the local planning module to the potential field navigation strategies reviewed above, in turn overcoming the latter's fundamental limitations for complex worlds.*

4) Safety in Control: Control barrier functions (CBFs) are a class of functions whose role is to serve as instantaneous, point-wise constraints on the applied control to achieve forward invariance with respect to a safe set, in our context a *collision-free set* (or to achieve forward reachability then invariance with respect to a *target set*) [29]. By converting safety (reachability) constraints into linear-in-the-control constraint equations, CBFs simplify the certification of safety (reachability) with a trade-off in short-horizon optimality. Looking back at the history of CBFs, which derive from barrier functions, there is a connection to soft-constraint optimal control methods. Barriers were used to generate soft-constraint modifications to the optimization landscape, and could be used to generate a family of optimal control solutions from a parametrically varied barrier specification, which often had properties similar to or were potential fields. By sequentially solving and warm-starting a series of soft-constraint problems with increasing penalties, the intent was to converge to the equivalent hard-constraint solution. *In connecting potential methods to hierarchical planning, we have the opportunity to translate the collision-critical potentials to CBF-based command reshaping operators for ensuring the safe execution of local paths.*

For an idealized robot model (full control, 1st order, point-mass, etc.), the safe forward-invariant set from a CBF specification aligns with the safe reachable set generated using HJB methods [30]. This set is described by a barrier function level-set. For non-ideal robots a gap emerges between the idealized level-set defined by the barrier function and the safe set defined by HJB reachability/CBF feasibility analysis. Both approaches require offline analysis to characterize the gap to arrive at the correct set. *This paper takes a step towards the efficient, online modification of idealized, sensor-derived safe sets and applied controls for non-ideal robots.*

B. Hierarchical Planning for Navigation

As briefly noted in §I-A1 and depicted in Figure 1, a hierarchical navigation system decomposes its task into two parts with distinct temporal and spatial scales: a global planner and a local planner. Given a goal point and an estimated map, the global planner synthesizes a candidate path from the current robot pose to the goal. The global path is passed to the local planner, whose objective is to generate local goals and local paths that lead the robot to the final goal. The reduced temporal and spatial scale of the local planner permits real-time operation in the face of novel information generated from sensor measurements. These measurements are also passed to the global planner to enrich the global map for generating valid global paths. The objective of the local planner, as informed by the global path, is to sequentially synthesize collision-free trajectory segments whose concatenation terminates at the final goal. The *potential gap* method described next will define the operation of the local planner.

II. The Potential Gap Local Navigation Module

This section describes the approach taken to identify *gaps* for navigation purposes and how they are used to specify potential-based local navigation vector fields. By design, the vector fields have guaranteed collision-free *passage through the gap* for idealized settings (first-order, point-mass, holonomic travel, full vision). For non-idealized robot models, the potential fields require modification to their construction and use barrier functions for a run-time hedge against collisions.

A. Gap Analysis and Gap Detection

Existing gap detection methods consider limited set of navigation and sensing scenarios, leading to correct but not fully generalizable gap detections. Though Closet Gap seeks to address these circumstances, the set of scenarios covered is still limited [25]. Our conception of a *gap* is informed by earlier work using gaps to define local navigation goals [16]. In considering the intent behind gap-based methods and their utility for navigation, we propose a different conception of gaps that leads to a new method for detection and analysis.

Viewing gaps from a perception space approach, it is more natural to represent them in robot-centered polar coordinates. This perspective leads to the categorization of gaps into swept and radial gaps based on their dominant direction. Conceptually, swept gaps involve curves with larger angular sweeps, while radial gaps have larger distance variation and little angular difference. The former "face" the robot and have good line-of-sight visibility properties. Swept gaps provide information both about passage *to* the gap, as well as *through* the gap. Radial gaps are oriented "sideways" to the robot and have bad line-of-sight visibility properties. They exhibit poor properties regarding passage to the gap and/or through it, see Figure 2(b) and Figure 3(a) mark A. *The objective of a gap detection strategy should be to provide the maximal quantity of swept gaps to leverage their favorable properties.*

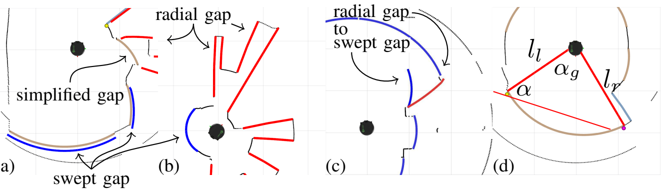


Fig. 2. (a,b) Visual of swept and radial gaps, (c) the radial gap conversion process, and (d) gap angular geometry. Black dot denotes the robot.

1) Swept and Radial Gap Detection: We define a new gap detection policy that simplifies gaps when they arise from complex world structure. The policy avoids potentially false information arising from measurement sparsity. Gap modeling hinges on classifying them as swept or radial.

Suppose that a full (360°) laser scan-like measurement $\mathcal{L}$ is available to the robot. In the instance of a limited field of view, measurement propagation using the *egocircle* [16] would keep track of measured parts of the surrounding collision-space. Let the range measurements $\mathcal{L}$ provide n readings with a maximum range value $d_{\max}$. Perform a first pass through $\mathcal{L}$ looking for the following two outcomes:

1) Large interval: $I = [i, i+k] \ni \mathcal{L}(j) = d_{\max}, \forall j \in I$, and $\mathrm{d}(\zeta_i, \zeta_{i+k}) > 2r_{\text{ins}}$ for $\zeta_j = (\mathcal{L}(j), \theta(j))^T$.
2) Instantaneous change in range: $|\mathcal{L}(i+1) - \mathcal{L}(i)| > 2r_{\text{ins}}$,

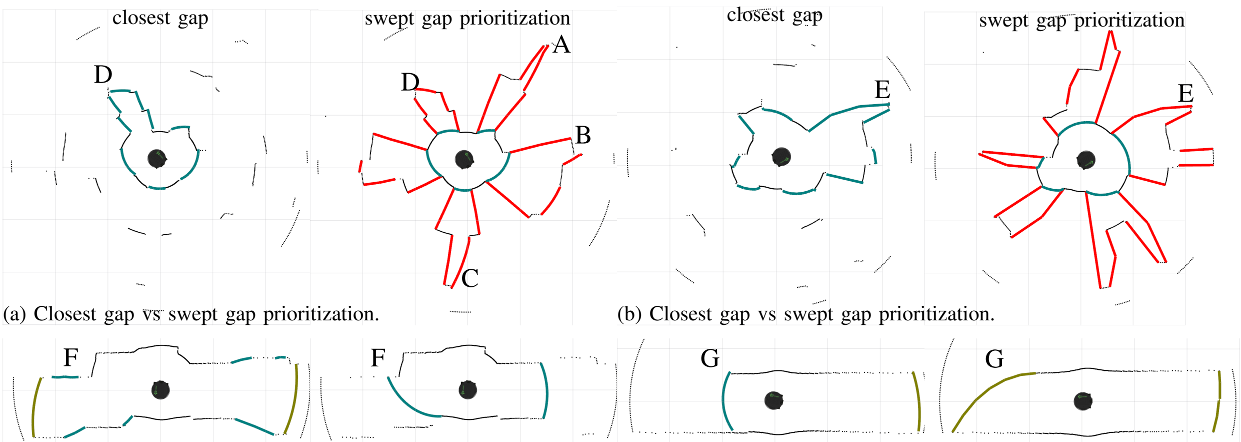
where $\theta(j)$ is the scan angle associated to index j, $\mathrm{d}(\cdot,\cdot)$ is the Euclidean distance, and r_{ins} is the inscribed radius of robot. Segments triggering the above are gaps that populate the gap set $\mathcal{G}$. Each gap in $\mathcal{G}$ is then classified as swept or radial, see Figure 2. Gaps picked based on the second test are automatically radial gaps. The others are classified based on the dominant direction. This direction is measured via the angle α on the short side of the triangle formed by the robot and the two sides of the gap in the laser scan, Figure 2(d). Let $l_l = \mathcal{L}(i)$, $l_r = \mathcal{L}(i+k)$, and $l_{\min} = \min[l_l, l_r]$. Then,

$$\alpha = \pi - 2\pi\frac{k}{n} - \sin^{-1}\frac{l_{\min}}{\sqrt{l_l^2 + l_r^2 - 2l_l l_r \cos(2\pi\frac{k}{n})}\sin(2\pi\frac{k}{n})}. \quad (1)$$

The threshold $\tau_\alpha = 3\pi/4$ differentiates radial from swept gaps. Assign the gap type as *left* when $l_r > l_l$ and *right* otherwise. For each detected gap $G \in \mathcal{G}$, record the angular index and distance of its sides.

2) Gap Simplification: A second pass through $\mathcal{G}$ merges radial gaps and replaces them with swept gaps. This process is done by moving gaps into a stack and recording the last continuously mergable radial gaps before reaching a termination condition. The termination condition is defined by threshold c_a on the angular difference between the pending gap and candidate terminating gap. A gap is mergable with a candidate gap if there are no closer obstacle points in between them. Mergability is determined by a threshold c_d on the acceptable range difference between the two gaps. Traversing every gap once when added to the final set and once more when determining the need for deletion, the time complexity of gap simplification is linear, e.g., $\mathcal{O}(n)$.

3) Radial Gap Conversion: For holonomic point-mass robots with full (360°) scanning the SGP strategy is sufficient for collision-free path planning. However, for robots



(a) Closest gap vs swept gap prioritization.

(b) Closest gap vs swept gap prioritization.

(c) Closest gap vs swept gap prioritization.

(d) Closest gap vs swept gap prioritization.

Fig. 3. Comparison between closest gap (CG) [25] and swept gap prioritization (SGP). Dark dots represents laserscan results, red lines denote raw gaps for SGP, and the teal lines denote returned gaps (CG and SGP). **(a)** Performing SGP collapses dangerous regions as A, B, and C. **(a,b)** Since SGP explicitly reasons about swept gaps, it simplifies regions D and E while CG does not due to boundary geometry readings. **(c)** SGP is less sensitive to sensor noise and provides a smaller, safer set of gaps as in region F, since regions behind swept gaps are collision free. **(d)** CG and SGP are comparable for G.

with partial scanning, non-circular shapes, or nonholonomic constraints, radial gaps cause problems based on the limited visual information available about what lies beyond the gap. To remedy the problem, a *radial gap conversion* process may be added to convert all remaining radial gaps into swept gaps by rotating their gap representation about the nearest gap point. The amount of rotation is controlled by the parameters $\epsilon = (\epsilon^1, \epsilon^2)$. Radial gaps are rotated by $\eta = \text{atan}(\epsilon^2/\epsilon^1)$ so that the local goal, a temporary point just ϵ^2 away on the other side of the gap is line-of-sight visible. Figure 2(c) shows a preserved radial gap converted to a swept gap.

4) Swept Gap Prioritization versus Closest Gap: We call the above process *Swept Gap Prioritization* (SGP). Figure 3 visualizes some of the benefits of SGP over closest gap (CG) [25], which attempts similar processing. The figures show examples of gap detection where raw gaps are marked bright red, and simplified gaps are marked teal. SGP collapses compound situations (A, B, and C regions), reducing gaps to the ones most relevant to the robot. The simplification reduces uncertainty behind radial gaps and avoids dangerous compound scenarios such as at the tip of region A and B. While both approaches generate similar results in most scenarios, SGP is computationally simpler. It leverages gap positioning in polar space during the reduction of gap sets to give $O(n)$ runtime while CG is $O(n^2)$ due to checking nested gaps. SGP is also more robust to situations where sensors readings are uncertain. Figure 3(c) illustrates where sensor uncertainty due to environment geometry causes discontinuities in range readings. SGP simplifies the gaps and returns ones better suited to motion planning.

B. Construction of Gradient Field with Circulation

The output of the SGP algorithm leads to a modified (and usually reduced) gap set $\mathcal{G}_{\text{SGP}}$. For each $G \in \mathcal{G}_{\text{SGP}}$ we establish a gap local goal that is at least ϵ^2 away from the gap, and ϵ^1 from the closest side of gap if the local goal lies outside of the (visible) angular space of the gap. The generation of a local navigation trajectory for the gap involves the creation of attractive potentials whose gradients are augmented with circulation terms [31]. Under normal circumstances the addition of circulation would have sign ambiguity. Here, the desired flow direction to the gap is known, and hence the circulation sign, so that the constructed solution has guaranteed passage through the gap.

1) Closing the Gap for Convexity: If the gap angular extent, α_g in Figure 2(d), is beyond τ_{GA} (usually 90 or 180°), then the gap angle is shrunk to give a new $G' \subset G$ whose angular extent is τ_{GA}, thereby ensuring that the gap region is a polar triangle convex in Euclidean space. Euclidean convexity ensures that the gradient vectors from the potential and circulation elements will have the necessary properties to guarantee gap passage. Gap closure is biased to ensure that the local gap goal lies within the outer gap region. The set $\mathcal{G}_{\text{SGP}}$ is remapped to consist of convex gaps only.

2) Potential and Circulation Fields: After convexification, and possibly modification (see §II-D3), to an angular extent equal to or less than τ_{GA}, the local goal on the other side of the gap is line-of-sight visible from any point within the gap region. Let the local goal point be x^*_{LG} as determined from the chosen gap $G^* \in \mathcal{G}_{\text{SGP}}$. The attractive potential is

$$\Phi(x) = \text{d}(x, x^*_{LG}) + \text{d}(x, G^*), \tag{2}$$

where the first distance is to the local goal point and the second is to the gap curve (then vanishes on the other side of the gap). These potentials attract the robot to the gap curve then through to the local goal. Rather than use the gradient for the flow, we will follow the normalized gradient.

The premise behind gaps is that there is an obstacle in the world that must be avoided by staying *within* the gap region, which is known to be collision-free. Rather than impose an obstacle avoiding potential, which could create a fixed point in the resulting vector field, a purely rotational vector field is created

$$\Theta(x) = \mathbb{J} e^{-\,\text{d}_\theta(x,p_l)/\sigma} \frac{p_l - x}{||p_l - x||} - \mathbb{J} e^{-\,\text{d}_\theta(x,p_r)/\sigma} \frac{p_r - x}{||p_r - x||}, \tag{3}$$

where $\mathbb{J} = R(-\pi/2)$ is skew-symmetric, and $\text{d}_\theta(\cdot,\cdot)$ is the angular distance. p_l and p_r are points of left and right side of gaps. The vector fields are two rotational fields anchored at the left and right gap points. Figure 4 shows an example circulation vector field.

3) Guaranteed Passage: Proving that passage through the gap for the robot must happen involves showing that the boundary of the gap navigation region points inwards along the robot-to-gap-endpoint edges (or simply *gap sides*) and that there is no fixed point interior to the region. The only reasonable flow for any point in the region is to exit via the gap curve, e.g., *gap passage*. Along the gap edges, the gradients of (2) either point inwards or parallel to it, never out by virtue of gap convexity and by definition of the gap region. Thus, what must be shown is that the circulation components also point inwards. On the gap edge, one circulation term has d_θ vanishing; the circulation is purely perpendicular and inward pointing. Let this vector be $e_\perp$. Let the other circulation term contribute the vector f_ϕ. It satisfies

$$(e_\perp + f_\phi) \cdot e_\perp = (1 - \cos(\pi - \phi)) \geq 0, \text{ for some } \phi > 0, \quad (4)$$

which means that $\Theta(\cdot)$ restricted to gap edges is inward pointing. The only outward flow can be on the gap curve.

A similar argument as above applies to show that the interior gap region points have a non-trivial outward pointing flow. Define $e_\rho(x)$ to be the radially directed outward vector for a point $x \in G$. The following properties hold:

$$e_\rho(x) \cdot \Theta(x) \geq 0, \quad \text{and} \quad e_\rho(x) \cdot \hat{\nabla}\Phi(x) > 0 \qquad (5)$$

by virtue of α_g being 90° (see §II-D3 for the $\tau_{\text{GA}} = 180°$ case and how it gets further reduced to 90°). The definition of $\hat{\nabla}$ is that it computes the gradient in (2) then makes it unit length when non-zero. The vector field interior to the gap region always has positive outward pointing contributions, so there can be no fixed points. Positivity implies that any initial point in G will flow out through the gap curve, see vector field in Figure 4. All planar trajectories starting in the polar triangle defined by G and the robot (which includes the robot location) and following the constructed vector field are guaranteed to exit the gap region through the *gap curve*, to be attracted to the local goal, and to be non-colliding. A formal theorem statement of this result can be found at [32].

C. Trajectory Synthesis and Scoring

With the generated set of gaps from Section §II-A, perform forward integration on every potential field from the current robot position to create a set of trajectories $\mathcal{X}$. Each trajectory is scored through cumulative egocentric pose cost and the terminating pose distance to the local navigation goal [4]. Let $\mathcal{T}$ denote a single trajectory discretized into poses p, p^* denote the local navigation goal, p_{end} denote the last pose on the trajectory, and $\mathrm{d}(p, \mathcal{L})$ return the distance from p to the nearest point in the egocircle $\mathcal{L}$. Score tuning parameters include: $w_1, w_2, c_{\text{obs}}, r_{\max}$. The trajectory score is:

$$J(\mathcal{T}) = \sum_{p \in \mathcal{T}} C(\mathrm{d}(p, \mathcal{L})) + w_1 \| p_{\text{end}} - p^* \| \ , \quad \text{where}$$

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

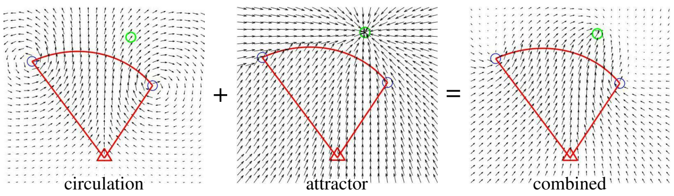


Fig. 4. Gap gradient field construction. Red triangle is robot location, blue circles are gap curve endpoints, and green circle is goal point.

At each planning iteration, the best trajectory from the set $\mathcal{X}$ is selected and compared against the currently executing trajectory, with an oscillation cost to limit path switching.

D. Safe Extensions for Non-Ideal Robots

The mathematics and algorithms employed to now presumed the use of an idealized robot model. Deviations from this model require modifications to the base method. This section describes the modifications required for (a) robots with area/volume, (b) non-holonomic robots, and (c) other deviations from the ideal.

1) Perception-Space Collision Checking: This module is an automatic part of our perception-space navigation stack [4], [16], [33]. It takes candidate local trajectories and synthesizes simulated *egocircle* measurements at their future poses along the trajectory that measure the far side of the robot. If any range measurements in the simulated *egocircle* lie beyond the current actual *egocircle* estimate, then the pose is in collision and the trajectory is rejected as infeasible. The process repeats until a collision-free path is found through a gap or all are rejected. The former gets passed on for trajectory tracking while the latter triggers a global replan.

2) Nonholonomic Vehicles: The control $u \in \mathfrak{se}(2)$ is not fully realizable due to vehicle motion constraints. However, most nonholonomic vehicles (and certainly all standard mobile robots) are linearly controllable along feasible trajectories. Let the vehicle be controlled through forward speed and turn rate. The control is converted to feasible movement via:

$$\xi_u = [\nu, 0, \omega]^T, \ \nu = u^1, \ \omega = \lambda_y u^2 + u^3. \qquad (6)$$

where $\xi, u \in \mathfrak{se}(2)$. The control is both realizable and guides the robot towards the local gap goal.

3) Radial Extension of Gaps: Since nonholonomy impacts the linear controllability of robot models, the paths generated from the vector field in §II-B cannot be precisely followed, but are asymptotically tracked by (6). The robot may actually leave the gap region when attempting to do so (typically when oriented away or orthogonal to the local goal). We address this problem by expanding the gap region in the vicinity of the robot while closing it further from the robot. This process is called *radial extension* since the vertex at the robot is shifted away from the gap curve along a radial line. Shifting the origin permits the construction of a new gap region that puts the robot inside of it. The robot may move for a non-zero distance and remain inside.

The process first finds the largest circular region centered on the robot known to be collision-free. Such a circle must exist

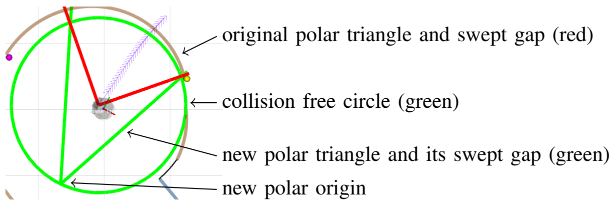

original polar triangle and swept gap (red)
collision free circle (green)
new polar triangle and its swept gap (green)
new polar origin

Fig. 5. Radial extension. The original swept gap polar triangle is converted to a new triangle with a shifted origin (backwards along the swept gap arc bisector). The shift distance is the radius of the largest free-space disc.

if the robot is non-colliding. The new origin is mapped to the intersection of the circle with the gap boundary bisector that passes through the robot frame, see Figure 5. A new gap region is established from this new origin and the radial vectors that pass through the two intersecting points of the safe circle and the gap region boundary lines. Simple geometric arguments show that the angular extent of this new gap is half that of (or smaller than) the original gap. Gaps extending out to $\tau_{\text{GA}} = 180$ are transformed to gaps with a gap angle less than or equal to 90°, as preferred in §II-B.

4) The Safety Projection Operator: One final addition aims to prevent collisions, when nonholonomic motion or the discrete time implementation of continuous control laws lead to collision. The addition applies the *projection operator* from adaptive control to prevent entrance to a region (as opposed to preventing exit). It has long been used to prevent parameter drift and is an early example of a barrier-like function that performs command reshaping with provable forward invariance (for a convex set) [34], much like a CBF [29]. Let $\mathcal{C}$ be the set of collision curves to the left and to the right of the gap (in polar space) and $\mathrm{d}(\cdot, \mathcal{C})$ the Euclidean distance function to $\mathcal{C}$. Define the potential function:

$$\psi(x) = \left(\frac{r_{\text{min}}}{\mathrm{d}(x,\mathcal{C})} - \frac{r_{\text{min}}}{r_{\text{nom}}}\right) \Big/ \left(1 - \frac{r_{\text{min}}}{r_{\text{nom}}}\right), \tag{7}$$

where $r_{\text{min}} < r_{\text{nom}}$ and which is depicted in Figure 6. Far enough away, the potential is negative. At r_{nom}, ψ passes through zero and at r_{min} it passes through unity. These properties are essential to the projection operator, which is defined in (8). In adaptive control, the projection operator keeps adaptive parameters from going unbounded by restricting the parameters to a compact set. Here, by definition of ψ, the opposite holds with the intent being to drive the state away from the open, bounded set associated to $\Omega_O = \{x | \psi(x) > 1\}$. When ψ evaluates to less than zero, no control modification occurs. When greater than zero, but with the control vectored away from the boundary, no control modification occurs. As $\psi \to 1$ under a violating law u, the control modification increases until canceling out the component leading to the safety violation. Under full control, the projection operator for ψ prevents control dynamics that would enter the unsafe set Ω_O established by the obstacles, and satisfies the same properties as a control barrier function.

III. Experiments, Outcomes, and Discussion

This section covers the experiments and outcomes conducted for potential gap and for the state-of-the-art method TEB [15]. Monte Carlo rollouts are performed for navigation tests in benchmark worlds, which are populated with randomized unknown obstacles and start/goal locations.

A. Benchmark Environments and Performance Criteria

The TEB and *potential gap* (PG) navigation planner are run in four benchmarking worlds. Dense, Campus, and Office are from [16]. The fourth, Sector world, is a single large room intended to model indoor spaces with locally untraversable regions such as chair legs and obstacles such as trash cans and cabinets. The simulation environment is a slightly modified version of the Simple Two Dimensional Robot Simulator (STDR) [35]. A very short range 360° range finder acts as a virtual bumper for collision detection. The platforms are configurable with different angular fields of view (FoV) of the laser sensor, starting from 60° to 360°, for partial to full observability. Additionally, experiments are conducted in Gazebo with Turtlebots (60° FoV) for navigation performance with nonholonomic, 2nd order robot. Parameter and implementation settings are found at [32]. Performance evaluation criteria are the success and collision rates, where higher and lower are better, respectively.

B. Experiments and Outcomes

The experiments cover a range of test cases, starting with the idealized robot model under variable field of view (FoV) settings. The idealized robot model has 1st order, fully actuated control equations (i.e., holonomic) with a circular body that operates like a point mass. The second experiment changes the control equations to being nonholonomic. The third experiment moves from STDR to Gazebo, where the motion model is nonholonomic, 2nd order. Each setting is run with 100 different seeds in all four worlds. STDR experiments are run 25 at a time and Gazebo experiments 3 at a time on an AMD Ryzen 7 3800X processor (single-thread passmark of 2744; multi-thread score of 43,447).

1) Idealized Robot Model: Figure 7 gives the failure mode vs FoV for PG and holonomic TEB implementations. Firstly, for the full FoV case, PG achieves no collisions and a nearly perfect success rate. A small quantity (1%) of runs resulted in an abort, which means that the local and global planner could not coordinate to find a feasible path to the goal. TEB has collisions for all FoV cases. PG continues to match or outperform TEB in collisions, aborts, and time outs in lower FoV cases. PG also maintains a high success rate, much like TEB preserves its moderate success rate. However, for 60° and 90° FoV, PG has more collisions, and matches TEB's success rate at the lowest FoV. Enabling radial extension (PG$^+$) gives a bit more navigation free-space around the robot to compensate for the limited FoV. PG$^+$ matches or outperforms TEB for all cases except for 60°, where the collision count remains higher (but success rate is higher).

The results show that in the idealized setting, the PG navigation system matches the theoretical expectations (high success, no collision). TEB has moderate success and few collisions. The lower FoV results involve collisions that arose from a lack of FoV but can be corrected with the RE module.

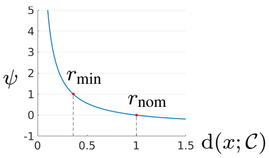


Fig. 6. Obstacle Potential

$$u' = \mathrm{Proj}(u;x) = \begin{cases} u & \text{if } \psi(x) < 0 \\ u & \text{if } \psi(x) \geq 0 \ \wedge \ \left\langle \frac{\nabla\psi(x)}{\|\nabla\psi(x)\|}, u \right\rangle > 0 \\ u - \psi(x)\left\langle \frac{\nabla\psi(x)}{\|\nabla\psi(x)\|}, u \right\rangle \frac{\nabla\psi(x)}{\|\nabla\psi(x)\|} & \text{if } \psi(x) \geq 0 \ \wedge \ \left\langle \frac{\nabla\psi(x)}{\|\nabla\psi(x)\|}, u \right\rangle \leq 0 \end{cases} \tag{8}$$

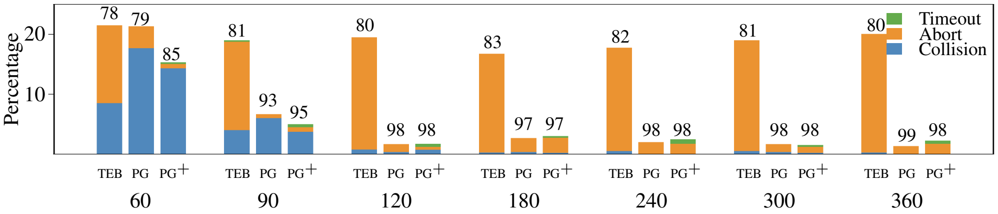


Fig. 7. TEB (left) and PG (middle) performance vs FoV for the idealized robot model (1st order holonomic). Numbers over columns are success rate (%). PG+RE (right, as PG$^+$) adds radial extension for FoV issues.

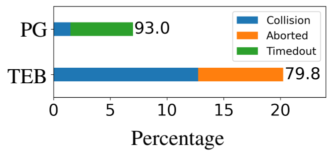


Fig. 9. PG and TEB performance for fixed FoV in Gazebo.

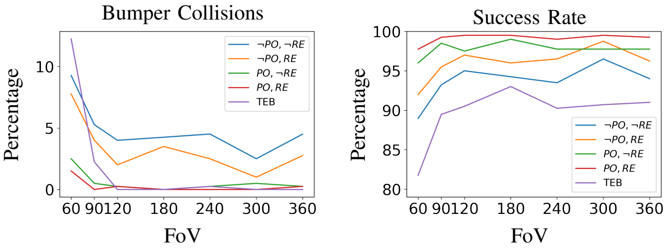


Fig. 8. PG and TEB bumper collisions (left) and success rate (right) vs FOV for the 1st order nonholonomic case.

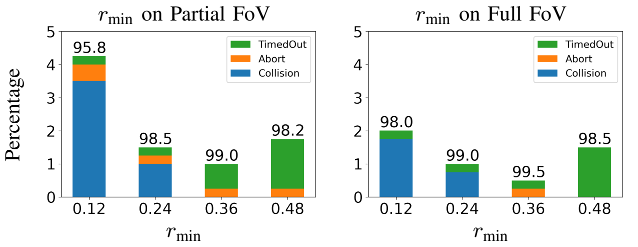


Fig. 10. Safety vs Passage test for 120° and 360° FoV in nonholonomic 1st order case. Performance vs $r_{\min}$.

PG employs a greedy gradient descent approach to path generation. In contrast, TEB employs a soft-constraint optimal control formulation to establish the local navigation paths and has higher computational cost. The high success rate of PG and PG$^+$ confirms that the notion of using gaps to serve as navigation affordances is critical to achieving safe, goal attaining navigation through an unknown environment.

2) Nonholonomic, 1st order: Figure 8 provides the outcomes for bumper collisions (left) and success rate (right). Each plot contains implementations of nonholonomic TEB, baseline PG without projection operator (¬PO) and radial extend (¬RE), the full nonholonomic (PO,RE), and mixed activations. The baseline PG, which uses the idealized robot solutions, has the worst performance in lower FoVs yet higher success rate than TEB. TEB maintains a low collision count. Of the two partially corrected PG implementations, the projection operator performs best (PO,¬RE). Adding radial extension improves it somewhat (for lower and higher FoV), showing that both contribute to safe navigation. The low collision performance of TEB is undermined by its moderate success rate induced by aborts. PG+(PO,RE) method has more consistent success rate as a function of FoV. Overall PG+(PO,RE) has the best performance in terms of high success rates and low collision rates across FoV.

Similar collision outcomes between PG+(PO,RE) and TEB show that the modified method functions as intended and that the greedily derived paths are similar to optimally derived paths from a safety perspective. They are better than the sampled and optimally derived paths of TEB from a task completion perspective. Compared to the holonomic case in Figure 7, both planners displayed an outcome less sensitive to FoV. This is caused by the motion model limiting motion to angles local to the camera optical axis.

3) Nonholonomic, 2nd order: Figure 9 displays the failure rates for PG+(PO,RE) and nonholonomic TEB in the Gazebo simulated Turtlebot (FoV = 60°). The TEB collision rate matches the 1st order case. The 2nd order nonholonomic PG collision rate is similar to the 1st order case and significantly better than TEB (∼6x). The success rate for PG drops from 96% to 93%, but is well above those of the 1st and 2nd order TEB cases.

PG is relatively consistent when applied to the dynamic case. Again, this is an important outcome based on the fact that PG is based on APF methods that were designed for idealized robot models. As a greedy path generation approach, it outperforms the soft-constraint optimal control TEB solution. Though TEB could explore a far richer trajectory-space for a given local goal through probabilistic roadmaps (PRM), the approximate PRM generated from sensor data produces paths less representative of the local navigable space compared to gap methods due. Consequently, the PRM does not exploit navigable space as well as the potential gap method does. The experiments indicate that PG typically explores 5x more trajectory options than TEB. Furthermore, the soft constraint approach cannot guarantee feasibility of the trajectories, which may result in poor sequential decision making if some of the solutions are immediately feasible but eventually infeasible. By construction PG avoids these issues and creates protective mechanisms to prevent collision while aiming to complete the task.

4) Nonholonomic, Safety vs Passage: In §III-B2, the modified PG method with full FoV did not achieve perfect navigation as for the PG holonomic case in §III-B1. This is a function of the lack of full control, the tuning of the projection operator, and some non-determinism in the outcomes. There is a trade-off between tuning for task completion versus safety. Figure 10 shows the effect of varying the parameter $r_{\min}$ in (7). For lower values there are less aborts or time outs, but at the cost of increased collisions. For higher values there are more time outs and some aborts, but no collisions. The results hold for the two FoV cases tested, which represent the lowest reasonably safe and available FoV for popular range-based sensors, and the maximal FoV settings. Future work aims to address the trade-off by exploring how to dynamically adjust the parameter.

5) Timing: The median per trajectory compute time of potential gap was less than 1ms, while TEB was just under 10ms, leading to an order of magnitude difference. The average processing time per frame for PG was 8ms, half that of TEB (∼17ms). Recall that, on average, PG evaluated 5x more paths than TEB per invocation.

IV. Conclusion

Potential gap is a local planner, designed using ideas from artificial potential field theory to create a spatially localized gradient descent problem with guarantees on safe passage under an idealized robot model with full sensing. Central to the idea is the use of perception space to identify gaps for constructing the local gradient flow. Deviations from this model, such as field of view limitations and nonholonomic motion models, require modifications of the baseline method to recover lost safety and performance properties. Simulated scenarios demonstrate that the method achieves high completion rates and matches the safety of an optimal trajectory synthesis method but with faster runtime and higher completion rate. The code is open-sourced [32].

The study is a first step towards provably safe, hierarchical navigation with realistic robot models and configurations. Additional investigation is needed to strengthen the safety properties for the nonholonomic case. Lastly, more aggressive maneuvering with strong momentum effects were not considered. Further study is needed to explore how the potential gap method may operate under these conditions.

References

[1] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” in *ICRA*, 1985, pp. 500–505.

[2] Y. Koren and J. Borenstein, “Potential field methods and their inherent limitations for mobile robot navigation,” in *ICRA*, 1991, pp. 1398–1404.

[3] S. Patel, Sang-Hack Jung, J. P. Ostrowski, R. Rao, and C. J. Taylor, “Sensor based door navigation for a nonholonomic vehicle,” in *ICRA*, 2002, pp. 3081–3086.

[4] 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.

[5] T. Urakubo, K. Okuma, and Y. Tada, “Feedback control of a two wheeled mobile robot with obstacle avoidance using potential functions,” in *IROS*, 2004, pp. 2428–2433.

[6] D. J. Zhu and J. . Latombe, “New heuristic algorithms for efficient hierarchical path planning,” *IEEE Trans. on Rob. and Aut.*, vol. 7, no. 1, pp. 9–20, 1991.

[7] J. Saunders, R. Beard, and J. Byrne, “Vision-based reactive multiple obstacle avoidance for micro air vehicles,” in *ACC*, 2009, pp. 5253–5258.

[8] V. Dimitrov *et al.*, “Hierarchical navigation architecture and robotic arm controller for a sample return rover,” in *T-SMC*, Oct 2013, pp. 4476–4481.

[9] M. Teiner, I. Rojas, K. Goser, and O. Valenzuela, “A hierarchical fuzzy steering controller for mobile robots,” in *Int. Workshop on Scientific Use of Submarine Cables and Related Technologies*, 2003, pp. 7–12.

[10] J. Guldner, V. I. Utkin, and R. Bauer, “Mobile robots in complex environments: a three-layered hierarchical path control system,” in *IROS*, vol. 3, Sep. 1994, pp. 1891–1898.

[11] T. Y. Teck, M. Chitre, and P. Vadakkepat, “Hierarchical agent-based command and control system for autonomous underwater vehicles,” in *ICoIAS*, June 2010, pp. 1–6.

[12] S. M. LaValle, *Planning algorithms*. Cambridge university press, 2006.

[13] J. Borenstein and Y. Koren, “The vector field histogram-fast obstacle avoidance for mobile robots,” *IEEE Trans. on Rob. and Aut.*, vol. 7, no. 3, pp. 278–288, June 1991.

[14] 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.

[15] 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.

[16] J. S. Smith, R. Xu, and P. Vela, “egoTEB: Egocentric, Perception Space Navigation Using Timed-Elastic-Bands,” in *ICRA*, 2020, pp. 2703–2709.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] D. Koditschek, “Exact robot navigation by means of potential functions: Some topological considerations,” in *ICRA*, 1987, pp. 1–6.

[22] C. I. Connolly, J. B. Burns, and R. Weiss, “Path planning using laplace’s equation,” in *ICRA*, 1990, pp. 2102–2106 vol.3.

[23] L. Matthies, R. Brockers, Y. Kuwata, and S. Weiss, “Stereo vision-based obstacle avoidance for micro air vehicles using disparity space,” *ICRA*, pp. 3242–3249, 2014.

[24] 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.

[25] 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.

[26] 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.

[27] M. Mujahed, D. Fischer, and B. Mertsching, “Safe gap based (sg) reactive navigation for mobile robots,” in *ECMR*, 2013, pp. 325–330.

[28] M. Mujahed and B. Mertsching, “The admissible gap (ag) method for reactive collision avoidance,” in *ICRA*, 2017, pp. 1916–1921.

[29] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in *ECC*, June 2019, pp. 3420–3431.

[30] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, “Hamilton-jacobi reachability: A brief overview and recent advances,” in *CDC*, 2017, pp. 2242–2253.

[31] D. Chang and J. Marsden, “Gyroscopic forces and collision avoidance with convex obstacles,” in *New Trends in Nonlinear Dynamics and Control and their Applications*, W. Kang, C. Borges, and M. Xiao, Eds. Springer Berlin Heidelberg, 2003, pp. 145–159.

[32] R. Xu, S. Feng, and P. Vela, “Potential gap,” https://github.com/ivaROS/PotentialGap.

[33] J. S. Smith and P. A.Vela, “PiPS: Planning in perception space,” in *ICRA*, May 2017, pp. 6204–6209.

[34] E. Lavretsky and K. Wise, *Robust and Adaptive Control*, ser. Advanced Textbooks in Control and Signal Processing. Springer, 2013.

[35] A. T. M. Tsardoulias, C. Zalidis. (2014) stdr_simulator - ros wiki. [Online]. Available: http://wiki.ros.org/stdr_simulator