GPF-BG: A Hierarchical Vision-Based Planning Framework for Safe Quadrupedal Navigation

Shiyu Feng[1], Ziyi Zhou[1], Justin S. Smith[2], Max Asselmeier[1], Ye Zhao[1] and Patricio A. Vela[2]

***Abstract*— Safe quadrupedal navigation through unknown environments is a challenging problem. This paper proposes a hierarchical vision-based planning framework (GPF-BG) integrating our previous Global Path Follower (GPF) navigation system and a gap-based local planner using Bézier curves, so called *Bézier gap* (BG). This BG-based trajectory synthesis can generate smooth trajectories and guarantee safety for point-mass robots. With an empirical robot geometry extension and stabilized perception space, safety and robustness are significantly improved for quadrupedal navigation. Both simulation and real experiments are conducted to evaluate safe navigation performance of the proposed framework and multiple benchmarking frameworks. The proposed GPF-BG demonstrates the best safety results among all experiments.**

I. Introduction

Quadrupedal robots have demonstrated superior terrain traversability compared to traditional wheeled robots [1]. Significant progress has been made during the past few years to improve the robustness and agility of legged locomotion control [2]–[8], which enables the incorporation of exteroceptive sensors for autonomous navigation (e.g. Fig. 1). Taking into account legged robot morphology, prior navigation works have been mostly focusing on rough terrain traversability [9]–[14], multi-modal planning [14]–[18], and multi-robot exploration [19]. However, formal guarantees such as locomotion safety and obstacle avoidance are not formally analyzed or guaranteed, and the tested scenarios are limited to less dense environments. Safe navigation through unknown or partially unknown environments for quadrupeds yet remains an open question and critical to deploy these legged robots into the wild.

Vision-based navigation using depth cameras provides richer perception information and relatively accurate distance estimation to objects. It improves the capability of quadrupedal robots to move on different terrains and avoid obstacles. Moreover, the hierarchical structure is commonly used in safe navigation [20]. The entire planning problem is hierarchically decomposed into multiple layers. Global path follower (GPF-X) navigation system [21] in our previous work generates safe paths and additionally checks collisions through Planning in Perception Space (PiPS) [22] to guarantee safety. A potential gap [20] local planner is guaranteed to be safe for holonomic point-mass robots. Integrating this local planner with GPF-X can theoretically prove safety. However, this framework is only designed for grounded mobile robots. Generalizing it to quadrupeds needs further investigation.

Fig. 1: Unitree A1 quadruped navigates in an unknown and constrained environment.

In this paper, we propose a novel hierarchical vision-based system, GPF-BG (depicted in Fig. 2), for safe quadrupedal navigation. The term BG stands for *Bézier gap* local planner, which is an improved version of potential gap. It incorporates robot orientation and geometry during gap detection and trajectory synthesis. In addition, due to high oscillation of the robot body during locomotion, a perception space stabilization mechanism is integrated for correcting perception space.

The contributions of this paper are listed below and match various modules in Fig. 2: **1)** A hierarchical navigation system is proposed to solve safe navigation for quadrupedal locomotion (Section III). **2)** The proposed *Bézier gap* local planner (Section IV-A) offers a new Bézier-based trajectory synthesis method from gaps. Safety is guaranteed for point-mass robot. With the empirical robot geometry extension, safety is improved for quadrupedal robots. **3)** Stabilized perception space provides safer collision checking with the locomotion dynamics of quadrupedal robots (Section IV-B). **4)** Simulation benchmark and real experiments demonstrate the hierarchical vision-based planning framework for safe quadrupedal navigation (Section V).

II. Related Work

Quadrupedal Navigation: Previous works have achieved impressive results integrating navigation and legged locomotion control [9]–[19]. Specifically, medium and large-size quadrupeds in [9], [11], [12] used laser scanners to detect obstacles for navigation and environment exploration [19]. Stereo cameras were used in [11], [12] but only

[1]S. Feng, Z. Zhou, M. Asselmeier, and Y. Zhao are 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

[2]J. Smith 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.

*This work supported in part by NSF Award #1849333, #2144309 and Georgia Tech IRIM/IPaT Aware Home Seed Grant.

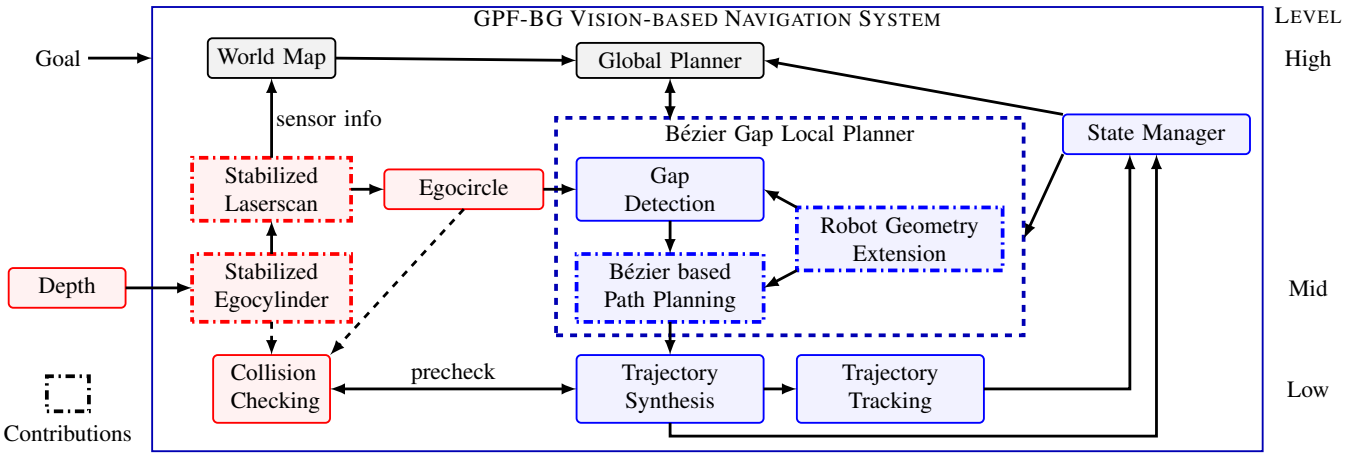


Fig. 2: GPF-BG hierarchical navigation system. The red blocks include perception space. Blue blocks present following and tracking modules. The dashed arrows indicate the option to choose either egocylinder or egocircle for collision checking.

for terrain mapping. In [13], [14], [17], [18], stereo and tracking cameras were used on small-scale quadrupeds for terrain mapping, navigation, and localization. Off-the-shelf or customized path planners were used to allow collision avoidance or multi-modal selection. However, none of them formally studied safety in vision-based navigation.

Safety in Vision-based Navigation: The hierarchical system has a long history to apply on multiple robotic platforms to increase the robustness and fault tolerance [23]–[26]. Safety in multiple hierarchical levels is explored.

At the planning level, both global and local planners can find collision free paths. Sampling-based methods such as Dijkstra's algorithm, A*, D*, Probabilistic Road Map (PRM), and Rapidly-exploring random trees (RRT) are global planning in discrete spaces [22]. Local reactive planners, e.g. DWA [27] and TEB [28], generate safe velocity commands based on costmap or optimal trajectories. Some gap-based local methods [20], [29]–[34] use the gap as navigation affordance and incorporate sensory data into planning. Especially, potential gap [20] proves safety guarantee for ideal robots. It provides a basic for safe quadrupedal navigation.

Safety in control avoids obstacles in the low level of hierarchical structure. A real time vision-based navigation method based on Hamilton-Jacobi reachability can guarantee strong safety in unknown environments [35]. Control barrier functions (CBF) also serve as constraints applied on the control to reach collision-free sets [36].

Moreover, fast collision checking assures safety. PiPS [22] collision checking with egocylinder is fast, and propagates temporal sensing information to improve safety with limited sensor field of view. Stixel egocentric navigation [21] is an alternative sparse method using stereo camera, which has better scalability among different computational platforms.

Trajectory Synthesis from Potential Field: In potential gap local planner [20], trajectories are synthesized through constructed potential field by integrating along the vector field. The robot orientation and dynamics are not considered, which could generate infeasible trajectories for tracking and cause collisions. Bézier curve is well known to synthesize smooth trajectories in the local planning with various robots [37]–[42]. Improving standard potential gap by synthesizing trajectories based on Bézier curve is able to increase safety for quadrupedal robots.

III. Hierarchical Navigation System

For quadrupedal navigation, the GPF-X hierarchical navigation system [21] is applied. The system flowchart is described in Fig. 2. GPF-X maintains safety through the hierarchical structure. In the global following mode, the system tracks the trajectory synthesized from global plan that is collision free in the local region. If the global planning is invalid, the state manager switches to trajectory based local planning mode. A gap-based local planner, called *Bézier gap* (BG), is proposed in §IV-A to guarantee safety for point mass robot. It is an improved version of potential gap (PG) [20] by redesigning path synthesis. An empirical extension for the quadrupedal robot geometry is added to balance traversability and safety. The entire system will be shortened as GPF-BG. In addition, PiPS [22] collision checking is performed on both global and local trajectories to prevent collision. Due to the locomotion of quadrupedal robots, a stabilized perception space is designed for successful planning within egocircle and collision checking with egocylinder. Therefore, the safe quadrupedal navigation is achieved through the hierarchical system.

IV. Methodology

This section describes the design of a *Bézier gap* local planner including trajectory synthesis, safety guarantee, and geometry extension. Then stabilized perception space is an improved version from standard egocylinder and egocircle generation.

A. Bézier Gap Local Planner

Potential Gap [20] local planner utilizes gaps as a navigation affordance. It offers fast free space modeling from egocentric perception space by line-of-sight visibility. Safety is guaranteed for holonomic point-mass robot. However, due to the dynamic constraints and geometry of quadrupedal

robots, the potential gap method cannot maintain safe navigation. Modifications should be made to improve safety and robustness. Gap set $\mathcal{G}$ is detected from egocircle $\mathcal{L}$ through the same procedures in potential gap.

1) Bézier-based Trajectory Synthesis: In potential gap, the path is integrated based on the vector field constructed from gaps without any consideration of robot orientation and dynamics. When switching to a new local path, it may lead to a sharp turning that is dangerous to quadrupedal robots. Therefore, smooth trajectory generation is required. Quadratic Bézier curve is able to create smooth paths incorporating robot initial orientation and reaching the local way point. The closed form of quadratic Bézier curve is

$$\boldsymbol{B}(u) = (1-u)^2\boldsymbol{Q}_0 + 2(1-u)u\boldsymbol{Q}_1 + u^2\boldsymbol{Q}_2, 0 \leq u \leq 1 \quad (1)$$

where $\boldsymbol{Q}_i$ is the control point. For each gap $G \in \mathcal{G}$, the robot initial position $\boldsymbol{p}(0)$ is used as the first control point $\boldsymbol{Q}_0 = \boldsymbol{p}(0)$. The local way point position $\boldsymbol{p}_{\text{wpt}}$ is set as the last control point $\boldsymbol{Q}_2 = \boldsymbol{p}_{\text{wpt}}$. The second control point is constructed based on robot initial orientation $\theta(0)$ and forward linear velocity $\nu(0)$. The unit orientation vector is $\vec{\boldsymbol{o}}(0) = [cos(\theta(0)), sin(\theta(0))]$. The vector $\boldsymbol{Q}_1 - \boldsymbol{Q}_0$ should be co-linear with the orientation vector $\vec{\boldsymbol{o}}$. Then the derivative of quadratic Bézier curve is

$$\boldsymbol{B}'(u) = 2(1-u)(\boldsymbol{Q}_1 - \boldsymbol{Q}_0) + 2u(\boldsymbol{Q}_2 - \boldsymbol{Q}_1) \quad (2)$$

where $\boldsymbol{B}'(0) = 2(\boldsymbol{Q}_1 - \boldsymbol{Q}_0)$. So the length is obtained from $||\boldsymbol{Q}_1 - \boldsymbol{Q}_0|| = \nu(0)/2$. All control points are finalized

$$\boldsymbol{Q}_0 = \boldsymbol{p}(0), \quad \boldsymbol{Q}_1 = \boldsymbol{p}(0) + \frac{\nu(0)}{2}\vec{\boldsymbol{o}}(0), \quad \boldsymbol{Q}_2 = \boldsymbol{p}_{\text{wpt}} \quad (3)$$

The trajectory is synthesized as

$$\begin{aligned}\mathcal{T}(u) =& (1-u)^2\boldsymbol{p}(0) + 2(1-u)u[\boldsymbol{p}(0) + \frac{\nu(0)}{2}\vec{\boldsymbol{o}}(0)] \\ &+ u^2\boldsymbol{p}_{\text{wpt}}, 0 \leq u \leq 1\end{aligned} \quad (4)$$

2) Trajectory Reparameterization: The Bézier based trajectory from (4) is parameterized by $u \in [0, 1]$ instead of time t. A second pass reparameterizes the raw trajectory based on desired linear velocity ν_d. The desired unit length is $l_d = \nu_d(t_{i+1} - t_i)$. The new trajectory is represented as a set of poses $\mathcal{T}(t) = \{\boldsymbol{p}(t_i) \mid 0 \leq t_i \leq t_{\text{end}}\}$. The length between two consecutive poses is l_d.

3) Safety Guarantee for Point-mass Robot: The safety can be guaranteed through the design of Bézier trajectory synthesis. A gap example is shown in Fig. 3. Two gap lines, L_l and L_r, are represented as blue lines starting from the robot origin. The gap lines construct a gap triangle G_{tri}. The red circle G_{circ} is the largest circular free space in egocircle. The entire free space $\mathcal{F}$ is the union of gap triangle and circle $\mathcal{F} = G_{tri} \cup G_{circ}$. The region outside the gap arc is also included in this free space. Moreover, the convex hull of Bézier polygon $\mathcal{B}$ contains Bézier curve. If $\mathcal{B} \subseteq \mathcal{F}$, the safety is guaranteed for point-mass robot when following the Bézier trajectory.

The safety guarantee problem becomes the design of three control points. The first and second control points are found

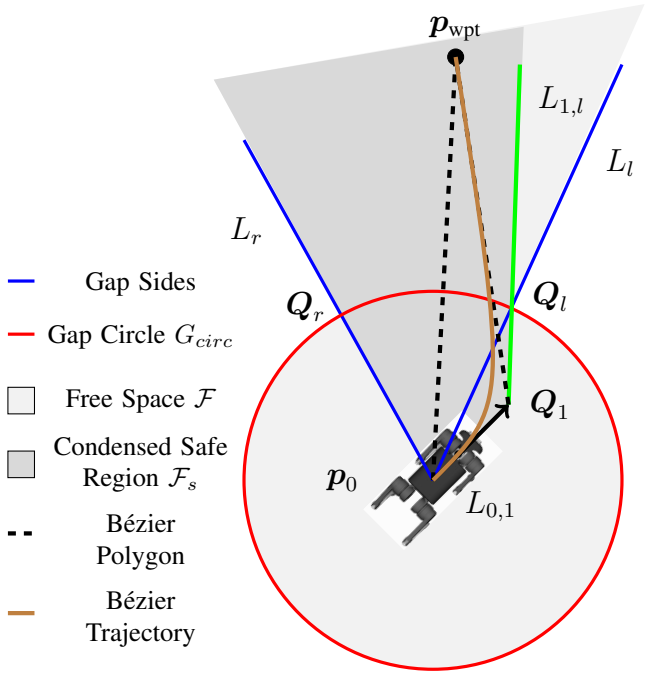


Fig. 3: Bézier trajectory synthesis. $\boldsymbol{p}_0$ is the robot origin. $\boldsymbol{Q}_1$ is the second control point. Blue lines L_l and L_r are left and right gap lines. Red circle is the largest circular free space in egocircle. $\boldsymbol{Q}_l$ and $\boldsymbol{Q}_r$ are the left and right intersection points of G_{circ} and G_{tri}. $L_{0,1}$ is the line segment constructed by $\boldsymbol{p}_0$ and $\boldsymbol{Q}_1$, and also represents the robot orientation. $L_{1,l}$ is the line through $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_l$. Local way point $\boldsymbol{p}_{\text{wpt}}$ is inside the condensed safe region $\mathcal{F}_s$ to guarantee safety. Dashed lines show the Bézier polygon. Brown is the synthesized trajectory.

through (3). Assuming the second control point is within gap circle $\boldsymbol{Q}_1 \in G_{circ}$, the local way point is a free point to keep safety. Define the intersection points of gap lines and circle as $\boldsymbol{Q}_l$ and $\boldsymbol{Q}_r$. The line constructed by second control point $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_l$ is denoted as $L_{1,l}$. The line created by first and second control points is $L_{0,1}$. A condensed safe region $\mathcal{F}_s \subset \mathcal{F}$ is enclosed by L_l, $L_{0,1}$, $L_{1,l}$ and all regions outside the gap arc. The safety is guaranteed by locating local way point $\boldsymbol{p}_{\text{wpt}}$ in the condensed free region.

$$\boldsymbol{p}_{\text{wpt}} \in \mathcal{F}_s \Rightarrow \mathcal{B} \subseteq \mathcal{F} \Rightarrow \text{Safety Guaranteed} \quad (5)$$

To be aware, if the second control point is outside the gap circle, this gap will be ignored for generating trajectories. When considering the robot geometry, two boundary lines L_r and $L_{1,l}$ of condensed safe region $\mathcal{F}_s$ will be rotated inwards to prevent collision from $\boldsymbol{Q}_l$ and $\boldsymbol{Q}_r$. The safe buffer is a ratio of robot width.

4) Quadrupedal Robot Geometry Extension: Since safety guarantee is only proved for point-mass robot, quadrupedal robot geometry should be considered in gap detection and path planning. The potential gap local planner normally assumes circular mobile robots so that a constant robot radius is enough as the safe buffer size. However, when applying to a quadrupedal robot, this assumption does not hold any more, due to the fact that using the robot's diagonal length as a constant parameter is overly conservative. The safe buffer size should vary based on the quadrupedal robot orientation. Therefore, two equivalent lengths are used to compromise the problem.

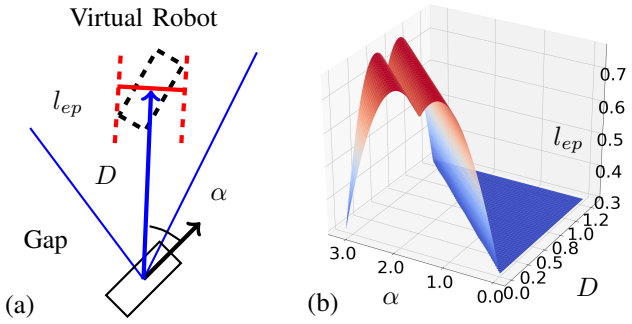


Fig. 4: Equivalent passing length. (a) l_{ep} finding. Black arrow is robot orientation. Blue arrow is gap direction. α is the angle difference between robot orientation and gap direction. D is the distance to the virtual pose. Red line l_{ep} is normal to gap direction. (b) An example of $l_{ep}(\alpha, D)$. The robot geometry is 0.7m x 0.3m. $\nu_d = 0.2$m/s, $\omega_d = 0.5$rad/s.

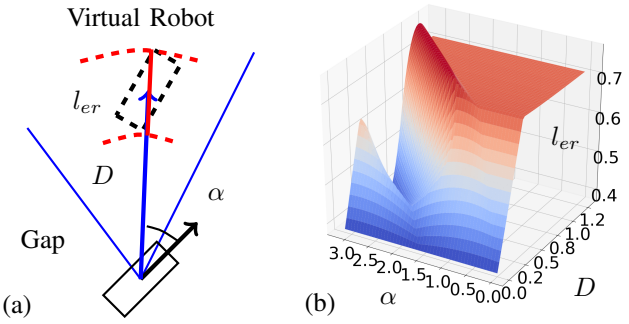


Fig. 5: Equivalent radial length. (a) The red solid line is l_{er}. Two red dashed curves are the largest and smallest circle arcs that can enclose virtual robot boundary. (b) An example of $l_{er}(\alpha, D)$.

Equivalent Passing Length l_{ep}. As shown in Fig. 4(a), the equivalent passing length l_{ep} is denoted by the distance between two lines that are parallel to gap direction and maximally enclose the rectangle. It assesses the passibility of gaps. If the gap length $l_{gap} > l_{ep}$, the gap is labelled as passible. This equivalent value is a function of the angle α between gap direction and robot orientation. However, in the planning process, forwardly passing through the gap is always preferred, i.e. eventually aligning robot orientation with gap direction. But due to the robot speed $\boldsymbol{v}_d = [\nu_d, \omega_d]$, the alignment is gradually achieved. The actual l_{ep} should be calculated based on the future robot pose passing through the gap. A virtual robot is placed to represent this future pose. Its orientation can be estimated from the distance D to the virtual pose and $\boldsymbol{v}_d$. Assume α is positive.

$$\alpha(D) = \begin{cases} \alpha(0) - \omega_d \frac{D}{\nu_d}, & \alpha(D) \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

l_{ep} becomes a function of α and D. An example of $l_{ep}(\alpha, D)$ is shown in Fig. 4(b). If D is large enough, the alignment is always achievable. l_{ep} becomes the robot width.

Equivalent Radial Length l_{er}. Similar to l_{ep}, an example demonstrating the equivalent radial length l_{er} is shown in Fig. 5(a). A virtual robot will be placed at the future pose. This equivalent value measures the radial difference between the largest and smallest circles that have intersections with the virtual robot boundary. It is used in radial gap manipulation and as the buffer size when radially shrinking the egocircle. Same as l_{ep}, this equivalent length is the function $l_{er}(\alpha, D)$, as shown in Fig. 5(b). When D is large enough, l_{er} becomes a value that is slightly larger than robot length.

5) Trajectory Scoring: The Bézier-based trajectory is synthesized for each gap. A set of raw trajectories will be scored to pick the best for following. The scoring function is similar to potential gap [20] with necessary modifications for rectangular robot geometry.

$$J(\mathcal{T}) = \sum_{\boldsymbol{p} \in \mathcal{T}} C(d_{\text{bound}}(\boldsymbol{p}, \mathcal{L})) + w_1 ||\boldsymbol{p}_{\text{end}} - \boldsymbol{p}_{\text{goal}}||, \text{where}$$

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

$d_{\text{bound}}(\boldsymbol{p}, \mathcal{L})$ is the nearest distance from robot boundary to egocircle at a trajectory pose $\boldsymbol{p}$. w_1, w_2, c_{obs} and r_{max} are tuning parameters and described in [20]. $\boldsymbol{p}_{\text{goal}}$ is the local goal found from global plans.

B. Stabilized Perception Space

1) Egocylinder: The egocylinder is an egocentric means of representing the local environment. It consists of a virtual cylinder whose surface is a range image I_c. As a 2D, perception space representation, it benefits from low latency and computational cost.

A point is added to the egocylinder by projecting it onto the virtual cylinder and storing the range of the point in the corresponding pixel of the range image. As the robot moves, the values in I_c are projected back into Cartesian coordinates, transformed by the robot's relative motion, and projected back onto I_c.

By placing the egocylinder's origin at the same position as the sensor (eg. depth camera), the projective nature of the egocylinder ensures that each pixel of I_c contains the range of the nearest (known) object in that direction.

However, rigidly fixing the egocylinder's frame to that of the sensor can have undesirable consequences if the sensor is subjected to significant oscillation. Since the egocylinder can only store one point in each direction, updates frequently result in some points being lost as multiple points map to the same pixel. As a result, the loss of points is exacerbated by frequent oscillations around the pitch and roll axes.

In this situation, it is beneficial to define a virtual, *stabilized* frame to compensate for some of the motion and reduce the point loss. While the position of this frame is still fixed to the sensor's, its orientation is not. Rather, the orientation is defined such that the frame remains level relative to the global frame and only rotates around its yaw axis.

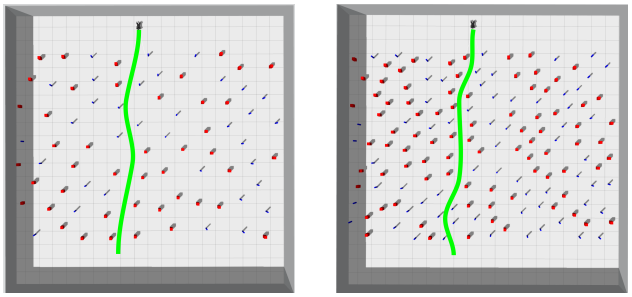
2) Egocylinder to laser scan: Another benefit of the stabilized egocylinder is that it simplifies the process of generating a 1D synthetic laser scan representation of the local environment. Upper and lower planes define the volume relative to the robot within which a point should be considered an obstacle. Projecting a ray through each row of I_c and calculating its intersection with either plane yields the range threshold below which a point represents an obstacle. Converting I_c to a laser scan is simply a matter of generating a mask of all pixels with values less than their rows' thresholds and applying a column-wise *min* reduction to the masked I_c.

V. Experiments

A. Simulation Configurations

In order to evaluate the performance of GPF-BG, Monte Carlo ROS/Gazebo simulation benchmark is applied. The platform is an Intel Xeon E5-2680 workstation (single-thread Passmark score: 1547).

1) Scenario setup: The same testing scenarios [20] are used: Sector, Dense, Campus and Office worlds. Obstacles are randomly spawned in the last three environments. Minimum distance between obstacles can control the environment complexity. Smaller means denser. 1.5m and 1m are tested to assess the safe navigation capability of different navigation frameworks. Fig. 6 shows the different complexity for dense world.



(a) 1.5m (b) 1m

Fig. 6: Dense worlds with different complexity. The robot is at the top. Robot actual paths are depicted in the environments. In the more complex environment, GPF-BG can still achieve successful navigation with similar path length.

2) Robot configurations: Three different robot configurations are used. 1) STDR box robot: a 2D planar robot without height geometry. A laser scanner with 360 degrees field of view (FoV) is used as the sensor to generate costmap and egocircle. 2) RidgeDog Gazebo robot: a 3D robot with the appearance of Unitree A1 model. The motion of holonomic mobile robot is simulated without moving legs. It replicates the robot geometry but removes the robot oscillations due to locomotion. A depth camera with 60 degrees FoV is the sensing source. A laser scan is created from the depth camera for costmap and egocircle. Egocylinder is used for collision checking. No stabilized perception space is required for the first two robot configurations. 3) Unitree A1 robot: a full-order model with 3 actuated joints on each leg and 18-DoF in total. The same low-level controller as in [5] is used to drive the robot. Perception spaces, egocylinder and egocircle, are built based on the same depth camera with stabilization described in §IV-B.

3) Navigation systems: To compare with GPF-BG, we use other three navigation systems: move_base with DWA [27], move_base with TEB [28] and GPF-PG (GPF-X with potential gap local planner). The global and local costmaps in the above systems are created from pure laserscan, depth image to laserscan and stabilized laserscan.

Monte Carlo benchmark provides statistics to evaluate performance. We have 80 runs for each navigation system with STDR and RidgeDog, and 40 runs with A1. Success, abort and collision rates are recorded to compare results.

B. Results and Analysis

The results are in Table I and II. STDR and RidgeDog do not need stabilized perception space. A1 needs perception stabilization for the robot oscillations.

1) STDR and RidgeDog: GPF-BG can achieve 100% success rate without any collision. For STDR, DWA and TEB do not have any collision, since 360 FoV is used to sense the laser safe environments. But they have more abort cases than GPF-BG. It indicates that GPF-BG not only can maintain safety, but also consider passibility through the environment.

For RidgeDog, there are a few collision cases in DWA and TEB due to the limited FoV. GPF-BG uses egocylinder and egocircle for collision checking and planning, which store and propagate temporal information to improve safety.

When increasing the environment complexity from 1.5m to 1m, the success rates of DWA and TEB drop around 4%. GPF-BG has no reduction to have 100% success rate.

2) A1: GPF-BG has higher success rate and less collision rate than DWA and TEB. The collisions are caused by limited FoV and failed robot locomotions. The tested navigation systems only consider velocity constraints of quadrupedal robot without incorporating full dynamics. GPF-BG has less success rate decreasing (only 2.5%) than DWA (10%) and TEB (7.5%) when testing in the more complex environments. Robot paths in dense world with different complexity are shown in Fig. 6.

3) Comparison between BG and PG: In all scenarios, GPF-PG is also tested. The stabilized perception space is used to obtain fair comparison. The success rate is better than or equal to DWA and TEB, but worse than GPF-BG. Collision rate is higher than GPF-BG, especially A1 robot (5% more).

4) Comparison for stabilized perception space: In simulation, only A1 robot requires stabilized perception space. The results are in Table III. GPF-BG$^-$ denotes the GPF-BG navigation system without stabilized perception space. Collision rate increases 12.5% causing less success.

Overall, from the simulation benchmark, GPF-BG has the best safety performance among tested systems. The *Bézier gap* local planner and stabilized perception space improves the safety and robustness of quadrupedal navigation.

TABLE I: Simulation Benchmark (minimum distance between obstacles = 1.5m)

STDR	Success	Abort	Collision
DWA	100%	0%	0%
TEB	97.5%	2.5%	0%
GPF-PG	100%	0%	0%
GPF-BG	100%	0%	0%

RidgeDog	Success	Abort	Collision
DWA	98.75%	0%	1.25%
TEB	100%	0%	0%
GPF-PG	100%	0%	0%
GPF-BG	100%	0%	0%

A1	Success	Abort	Collision
DWA	85%	0%	15%
TEB	85%	2.5%	12.5%
GPF-PG	85%	0%	15%
GPF-BG	87.5%	0%	12.5%

TABLE II: Simulation Benchmark (minimum distance between obstacle = 1m)

STDR	Success	Abort	Collision
DWA	96.25%	3.75%	0%
TEB	93.75%	6.25%	0%
GPF-PG	96.25%	1.25%	2.5%
GPF-BG	100%	0%	0%

RidgeDog	Success	Abort	Collision
DWA	95%	2.5%	2.5%
TEB	96.25%	2.5%	1.25%
GPF-PG	100%	0%	0%
GPF-BG	100%	0%	0%

A1	Success	Abort	Collision
DWA	75%	10%	15%
TEB	77.5%	0%	22.5%
GPF-PG	80%	0%	20%
GPF-BG	85%	0%	15%

TABLE III: Comparison of Stabilized Perception Space

A1	Success	Abort	Collision
GPF-BG	85%	0%	15%
GPF-BG$^-$	72.5%	0%	27.5%

TABLE IV: Real Experiment Results

Easy	Success	Abort	Collision
DWA	1/5	4/5	0/5
TEB	4/5	1/5	0/5
GPF-BG	5/5	0/5	0/5

Hard	Success	Abort	Collision
DWA	0/5	5/5	0/5
TEB	2/5	1/5	2/5
GPF-BG	4/5	0/5	1/5

C. Quadrupedal Navigation Evaluation

1) Experiment setup: The Unitree A1 is equipped with an Intel D435i depth camera and a 2D laser scanner. The laser scanner is only used for a built-in SLAM package generating a static global map without the obstacles discussed below, as well as providing the odometry data. The stabilized perception space is disabled in hardware tests, since A1 moves more stably than simulation. An indoor scenario is set up as shown in Fig. 1 and Fig. 7. The courses are characterized into easy and hard ones depending on the minimum distance between the obstacles. The easy course consists of three garbage cans (two smaller blue cans and one larger green can) with a 2m minimum obstacle distance. The hard course keeps the same setup but with an additional chair that further shrinks the minimum distance to 1.2m. The start and end poses are same among all experiments. 5 repeated runs are tested for each navigation system.

2) Experiment results and analysis: The results of real experiment are in Table IV. DWA planner cannot perform well with the real quadrupedal robot. In both easy and hard scenarios, GPF-BG has better success rate and less collision than TEB. In the more complicated environment, GPF-BG has smaller reduction of success rate, which is consistent with simulation results. Fig. 7 depicts the robot actual paths for GPF-BG and TEB in one hard environment testing. GPF-BG can safely reach the goal with shorter path than TEB. These real experiments further demonstrate safer navigation for quadrupedal robots.

Fig. 7: The robot navigation paths for GPF-BG (green) and TEB (red) in a constrained environment.

VI. Conclusion

GPF-BG is a hierarchical vision-based planning framework for safe quadrupedal navigation. The hierarchical structure improves safety for legged robot navigation. *Bézier gap* local planner not only shares the advantages of potential gap to guarantee safety, but also improves the robustness to different robot models. Stabilizing perception space for collision checking offers additional safety to robots with locomotion dynamics. Both simulation and real experiments are conducted to demonstrate the safe performance of GPF-BG. In the future work, terrain traversability and multi-modal planning for different action primitives will be integrated in this vision-based navigation framework. Safely navigating on different terrains becomes possible in the all-in-one system.

References

[1] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 236–258, 2011.

[2] G. Bledt, P. M. Wensing, and S. Kim, "Policy-regularized model predictive control to stabilize diverse quadrupedal gaits for the mit cheetah," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 4102–4109.

[3] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1–9.

[4] C. D. Bellicoso, F. Jenelten, C. Gehring, and M. Hutter, "Dynamic locomotion through online nonlinear motion optimization for quadrupedal robots," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2261–2268, 2018.

[5] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[6] Y. Ding, A. Pandala, C. Li, Y.-H. Shin, and H.-W. Park, "Representation-free model predictive control for dynamic motions in quadrupeds," *IEEE Transactions on Robotics*, vol. 37, no. 4, pp. 1154–1171, 2021.

[7] Z. Zhou, B. Wingo, N. Boyd, S. Hutchinson, and Y. Zhao, "Momentum-aware trajectory optimization and control for agile quadrupedal locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7755–7762, 2022.

[8] H. Zhu, D. Wang, N. Boyd, Z. Zhou, L. Ruan, A. Zhang, N. Ding, Y. Zhao, and J. Luo, "Terrain-perception-free quadrupedal spinning locomotion on versatile terrains: Modeling, analysis, and experimental validation," *Frontiers in Robotics and AI*, vol. 8, 2021.

[9] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert, "Autonomous navigation for bigdog," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 4736–4741.

[10] A. Chilian and H. Hirschmüller, "Stereo camera based navigation of mobile robots on rough terrain," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 4571–4576.

[11] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter, "Navigation planning for legged robots in challenging terrain," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1184–1189.

[12] M. Brandao, O. B. Aladag, and I. Havoutis, "Gaitmesh: controller-aware navigation meshes for long-range legged locomotion planning in multi-layered environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3596–3603, 2020.

[13] T. Dudzik, M. Chignoli, G. Bledt, B. Lim, A. Miller, D. Kim, and S. Kim, "Robust autonomous navigation of a small-scale quadruped robot in real-world environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3664–3671.

[14] D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim, "Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2464–2470.

[15] P. Fernbach, S. Tonneau, A. Del Prete, and M. Taïx, "A kinodynamic steering-method for legged multi-contact locomotion," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 3701–3707.

[16] J. Norby and A. M. Johnson, "Fast global motion planning for dynamic legged robots," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 3829–3836.

[17] M. Chignoli, S. Morozov, and S. Kim, "Rapid and reliable trajectory planning involving omnidirectional jumping of quadruped robots," *arXiv preprint arXiv:2111.13648*, 2021.

[18] S. Gilroy, D. Lau, L. Yang, E. Izaguirre, K. Biermayer, A. Xiao, M. Sun, A. Agrawal, J. Zeng, Z. Li, *et al.*, "Autonomous navigation for quadrupedal robots with optimized jumping through constrained obstacles," in *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2021, pp. 2132–2139.

[19] M. Kulkarni, M. Dharmadhikari, M. Tranzatto, S. Zimmermann, V. Reijgwart, P. De Petris, H. Nguyen, N. Khedekar, C. Papachristos, L. Ott, *et al.*, "Autonomous teamed exploration of subterranean environments using legged and aerial robots," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 3306–3313.

[20] 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.

[21] S. Feng, F. Lyu, J. Ha Hwang, and P. A. Vela, "Ego-centric stereo navigation using stixel world," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 201–13 207.

[22] 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.

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

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

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

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

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

[35] A. Bajcsy, S. Bansal, E. Bronstein, V. Tolani, and C. J. Tomlin, "An efficient reachability-based framework for provably safe autonomous navigation in unknown environments," *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 1758–1765, 2019.

[36] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[37] 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.

[38] 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/S1877750317308906

[39] H. A. Satai, M. M. A. Zahra, Z. I. Rasool, R. S. Abd-Ali, and C. I. Pruncu, "Bézier curves-based optimal trajectory design for multirotor uavs with any-angle pathfinding algorithms," *Sensors*, vol. 21, no. 7, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/7/2460

[40] 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

[41] 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

[42] 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.