Safe Set Synthesis with Tunable Boundary Gradients via Poisson Safety Functions

Gilbert Bahati and Aaron D. Ames

***Abstract*— Robotic systems operating in the real-world are typically required to avoid collisions while exercising appropriate levels of caution when approaching or navigating near different obstacles. To achieve this, we investigate a method of generating safe sets with gradient behavior customized to specific regions of a domain. Our method leverages Poisson safety functions, which enable the generation of safe sets from perception data. Specifically, we demonstrate how adjusting the boundary conditions of the guidance field yields tunable boundary gradients, whose magnitudes capture levels of caution around obstacle surfaces. We showcase the utility of the method in simulation, and discuss how it can be integrated with semantic information to achieve context-aware safety strategies, such as robustness tailored to specific obstacles.**

I. Introduction

Modern robotic systems increasingly leverage advanced perception tools, enabling semantic understanding of surrounding objects and obstacles. In autonomous driving for example, this is particularly important when different objects represent varying levels of risk or importance (e.g., humans vs static structures). Leveraging this information for dynamic navigation requires not only detecting these objects, but also exercising the appropriate level of caution when navigating near them.

Given a functional representation of the environment that characterizes safety, Control Barrier Functions (CBFs) [1], [2] are a tool used to synthesize safe controllers by enforcing the forward invariance of a desired safe set. However, in the context of collision avoidance, traditional methods for synthesizing safe sets often impose uniform gradients on the set boundary, treating all obstacles equally, which limits the ability to tailor safety behavior to specific contexts. Such uniformity offers limited flexibility in how conservativeness can be expressed in semantically rich scenarios.

Recently, Possion safety functions [3] have been proposed to enable real-time generation of safe sets from perception data by solving a Dirichlet problem for Poisson's equation. This approach leverages a guidance field that encodes gradient information required for safety. The guidance field provides additional flexibility in defining safety by allowing boundary conditions to specify desired repulsive gradients (i.e., boundary flux) on obstacle surfaces.

This work presents a method for generating safe sets with tunable boundary gradients by adjusting the desired boundary flux in Poisson safety functions. This enables encoding different levels of desired caution around obstacles.

The authors are with the Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, CA {gbahati, ames}@caltech.edu.

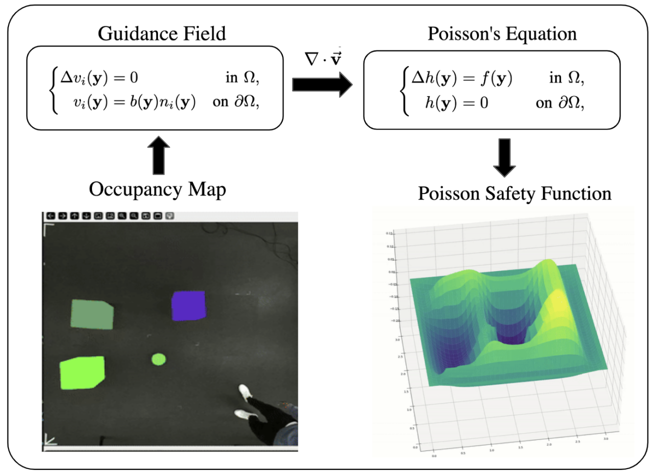

Fig. 1. Safe sets synthesis in real-time from perception data via Poisson's equation. See video here.

We validate the method in simulation and discuss how it can be integrated with semantic labeling to enable context-aware safety strategies such as obstacle-specific robustness.

II. Background

Consider the nonlinear control affine system of the form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m$ are the state and input, and $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^n, \mathbf{g} : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are assumed to be locally Lipschitz continuous functions. Given a locally Lipschitz continuous controller $\mathbf{k} : \mathbb{R}^n \to \mathbb{R}^m$, the closed-loop system $\dot{\mathbf{x}} = \mathbf{f}_{\text{cl}}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{k}(\mathbf{x})$ yields an ODE such that for any initial condition $\mathbf{x}(0) = \mathbf{x}_0 \in \mathbb{R}^n$, there exists a unique continuously differentiable solution $t \mapsto \mathbf{x}(t)$, which we assume exists for all $t \geq 0$ for ease of exposition.

A. Safety and Control Barrier Functions

We define the concept of safety through the notion of forward invariance, where trajectories $t \mapsto \mathbf{x}(t)$ must be kept in a desired safe set for all $t \geq 0$. In particular, we consider safe sets defined as the 0-superlevel set of a continuously differentiable function $h_{\mathcal{S}} : \mathbb{R}^n \to \mathbb{R}$ as:

$$\mathcal{S} = \left\{ \mathbf{x} \in \mathbb{R}^n \,\middle|\, h_{\mathcal{S}}(\mathbf{x}) \geq 0 \right\}. \tag{2}$$

Control Barrier Functions (CBFs) are a constructive tool that can be used to design controllers for (1) that enforce the forward invariance of the set $\mathcal{C}$.

Definition 1. (Control Barrier Functions [1]) We call a function $h_{\mathcal{S}} : \mathbb{R}^n \to \mathbb{R}$ a Control Barrier Function (CBF) for (1) if there exists[1] $\gamma \in \mathcal{K}_\infty^e$ such that for all $\mathbf{x} \in \mathbb{R}^n$, the following condition holds:

$$\sup_{\mathbf{u}\in\mathbb{R}^m} \{ \underbrace{\nabla h_{\mathcal{S}}(\mathbf{x})\cdot\mathbf{f}(\mathbf{x})}_{L_{\mathbf{f}} h_{\mathcal{S}}(\mathbf{x})} + \underbrace{\nabla h_{\mathcal{S}}(\mathbf{x})\cdot\mathbf{g}(\mathbf{x})}_{L_{\mathbf{g}} h_{\mathcal{S}}(\mathbf{x})} \mathbf{u}\} > -\gamma(h_{\mathcal{S}}(\mathbf{x})) \tag{3}$$

Given a nominal controller k_{nom}, and a CBF h, a typical way of synthesizing safe controllers is through quadratic programming-based safety filters, which adjust k_{nom} to the nearest safe action:

$$\begin{aligned} \mathbf{k}(\mathbf{x}) = \underset{\mathbf{u}\in\mathbb{R}^m}{\arg\min} \quad & \|\mathbf{u} - \mathbf{k}_{\text{nom}}(\mathbf{x})\|_2^2 \\ \text{s.t.} \quad & L_{\mathbf{f}} h_{\mathcal{S}}(\mathbf{x}) + L_{\mathbf{g}} h_{\mathcal{S}}(\mathbf{x})\mathbf{u} \geq -\gamma(h_{\mathcal{S}}(\mathbf{x})). \end{aligned} \tag{Safety-Filter}$$

Next, we discuss a method of synthesizing CBFs for environmentally relevant safety specifications, as presented in [3].

B. Poisson Safety Function

We focus on systems for which safety specifications are described in spatial coordinates $\mathbf{y} \coloneqq \mathbf{y}(\mathbf{x}) = (x, y, z) \in \mathbb{R}^3$. Given an occupancy map, let Ω be a smooth, open, bounded and connected set representing unoccupied regions and $\partial\Omega$ represent the surfaces of occupied regions. Specifically, $\partial\Omega = \bigcup_i^{n_\text{o}} \partial\Gamma_i$ where Γ_i is an open, bounded and connected set corresponding to the interior of an occupied region with n_o denoting the total number of unoccupied regions. A safety function provides a functional representation of safety for an environment, defined as follows.

Definition 2. (Safety Function [3]) Let $\mathbf{y} = (x, y, z) \in \mathbb{R}^3$ represent coordinates in three dimensional space. We call a function $h : \overline{\Omega} \to \mathbb{R}$ a safety function of order k on $\overline{\Omega}$ if h is k-times differentiable, $Dh(\mathbf{y}) \neq 0$ when $h(\mathbf{y}) = 0$, and the 0-superlevel set of h characterizes a safe set:

$$\mathcal{C} = \{\mathbf{y} \in \overline{\Omega} : h(\mathbf{y}) \geq 0\}, \tag{4a}$$
$$\partial\mathcal{C} = \{\mathbf{y} \in \overline{\Omega} : h(\mathbf{y}) = 0\}, \tag{4b}$$
$$\text{int}(\mathcal{C}) = \{\mathbf{y} \in \overline{\Omega} : h(\mathbf{y}) > 0\}. \tag{4c}$$

Given environmental data characterizing the domain $\overline{\Omega}$ through an occupancy map, Poisson safety functions [3] define safe sets satisfying Def. 2 as solutions to the Dirichlet problem for Poisson's equation:

$$\begin{cases} \Delta h(\mathbf{y}) = f(\mathbf{y}) & \text{in } \Omega, \\ h(\mathbf{y}) = 0 & \text{on } \partial\Omega, \end{cases} \tag{5}$$

where $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ is the *Laplacian* and $f : \Omega \to \mathbb{R}_{<0}$ is a given forcing function. As discussed in [4], under appropriate regularity assumptions on Ω, a smooth forcing function $f \in C^\infty(\overline{\Omega})$ yields a smooth solution $h \in C^\infty(\overline{\Omega})$ to (5). As demonstrated in [3], this smooth solution, h, characterizes the safe set $\mathcal{C}$ such that $\Omega = \text{int}(\mathcal{C}), \partial\mathcal{C} = \partial\Omega$, and may be used to construct safety filters yielding safe control actions for (1) under the appropriate relative degree assumptions [5]. One way of generating a smooth h is by constructing a smooth guidance field, which we discuss next.

[1]A continuous function $\gamma : \mathbb{R} \to \mathbb{R}$ is an *extended class* $\mathcal{K}$, $\mathcal{K}_\infty^e$, ($\gamma \in \mathcal{K}_\infty^e$) if γ is monotonically increasing, $\gamma(0) = 0$, $\lim_{s\to\infty} \gamma(s) = \infty$, and $\lim_{s\to-\infty} \gamma(s) = -\infty$.

C. Guidance Field

A guidance field provides a way of designing a forcing function f that encodes desired gradient information required for safety [3]. To construct a smooth guidance field, we consider the vector field $\vec{\mathbf{v}} = (v_x, v_y, v_z) : \overline{\Omega} \to \mathbb{R}^3$, with each component satisfying Laplace's equation subject to Dirichlet boundary conditions:

$$\begin{cases} \Delta v_i(\mathbf{y}) = 0 & \text{in } \Omega, \\ v_i(\mathbf{y}) = b(\mathbf{y}) n_i(\mathbf{y}) & \text{on } \partial\Omega, \end{cases} \tag{6}$$

for $i \in \{x, y, z\}$, where $\hat{\mathbf{n}} = (n_x, n_y, n_z) : \partial\Omega \to \mathbb{R}^3$ denotes the outward unit normal vector such that $\vec{\mathbf{v}}(\mathbf{y}) = b(\mathbf{y})\hat{\mathbf{n}}(\mathbf{y})$ on $\partial\Omega$, and $b : \partial\Omega \to \mathbb{R}_{<0}$ prescribes the outward directional derivative encoding the desired boundary flux. The boundary flux is negative to encode repulsive gradients on the boundary.

Given a guidance field $\vec{\mathbf{v}}$ as the solution to (6), one can verify that using the forcing function $f(\mathbf{y}) = \nabla \cdot \vec{\mathbf{v}}$ in (5) yields a smooth solution h whose gradient, ∇h, is the least squares approximation of $\vec{\mathbf{v}}$, with the error in the boundary flux given as $e(\mathbf{y}) = \nabla h(\mathbf{y}) \cdot \hat{\mathbf{n}}(\mathbf{y}) - b(\mathbf{y})$ on $\partial\Omega$. Furthermore, to guarantee h satisfies (4a), the following adjustment[2]:

$$f(\mathbf{y}) = -\frac{1}{\beta} \ln(1 + \exp^{-\nabla\cdot\vec{\mathbf{v}}(\mathbf{y})\beta}). \tag{7}$$

ensures that $f(\mathbf{y}) < 0$ for all $\mathbf{y} \in \Omega$, yielding in $h \in C^\infty(\overline{\Omega}; \mathbb{R}_{\geq 0})$ with $h(\mathbf{y}) = 0$ only on $\partial\Omega$. The next section presents our main contribution, highlighting the benefits of the guidance field in achieving more customized safety.

III. Contribution: Tunable Boundary Gradients

In this work, we examine the advantages of designing forcing functions (7) with the guidance field $\vec{\mathbf{v}}$ satisfying (6). In particular, we illustrate the provided flexibility in assigning desired gradient behavior on $\partial\Omega$ through the boundary conditions in (6), and its utility to achieve customized safety in the context of collision avoidance.

A. Obstacle-Specific Gradients

The boundary flux of $\vec{\mathbf{v}}$, dictated by the function b in (6), prescribes the desired magnitude of repulsive gradients ∇h along the boundary $\partial\Omega$. Large values of b yield steeper boundary gradients, while small values of b lead to shallow gradients. This ability to tune boundary gradients leads to varying magnitudes of repulsive gradients around obstacle surfaces, enabling the ability encode *obstacle-specific gradients*. In what follows, we demonstrate the yielded benefits of obstacle-specific gradients in the performance of synthesized safety filters with the use of a working example.

[2]Inspired by softplus function.

For ease of exposition, let $\Omega \subset \mathbb{R}^2$ such that $\vec{\mathbf{v}} = (v_x, v_y) = b\hat{\mathbf{n}}$ with $\hat{\mathbf{n}} = [n_x, n_y]^\top$ on $\partial\Omega$, and consider the single integrator:

$$\dot{\mathbf{y}} = \mathbf{w} \tag{8}$$

where $\mathbf{y} = (x, y) \in \mathbb{R}^2$ represents 2D spatial coodinates. We consider the task of stabilizing (8) from an initial condition $\mathbf{y}_0 \in \mathbb{R}^2$ to a goal position $\mathbf{y}_\mathrm{d} \in \mathbb{R}^2$, while avoiding collisions with obstacles.

Let $h \in C^\infty(\overline{\Omega})$ be the solution to (5) with the forcing function (7), characterizing a safe set $\mathcal{C}$ as in (4a). To ensure the system (8) avoids collisions with the obstacles (as shown in Fig 2.) and remains within $\mathcal{C}$, we filter a stabilizing nominal controller $\mathbf{k}_\mathrm{nom}(\mathbf{y}) = -K_\mathrm{p}(\mathbf{y} - \mathbf{y}_\mathrm{d})$ where $K_\mathrm{p} \in \mathbb{R}^{2\times 2}$ is a diagonal, positive definite matrix, to the nearest safe action with the following QP-based controller:

$$\mathbf{k}(\mathbf{y}) = \underset{\mathbf{w}\in\mathbb{R}^2}{\arg\min} \quad \|\mathbf{w} - \mathbf{k}_\mathrm{nom}(\mathbf{y})\|_2^2 \tag{9}$$

$$\text{s.t.} \quad \nabla h(\mathbf{y})^\top \mathbf{w} \geq -\gamma h(\mathbf{y}) \tag{10}$$

with $\gamma > 0$.

One can verify that small values of values of γ result in more conservative (i.e., cautious) behaviors near the boundary $\partial\mathcal{C} = \partial\Omega$. However, this leads to the same conservative behavior around all obstacles surfaces on $\partial\Omega$, resulting in *global* conservatism. In contrast, we find empirically that by assigning small values of b to specific obstacle surfaces on $\partial\Omega$ results in *local* conservative behaviors with respect to those obstacles. This provides a mechanism for encoding obstacle-specific gradients. To illustrate this, let Γ_obs correspond to the interior of the top right obstacle in Fig 2., we consider the following choices of desired boundary flux b in the boundary conditions of (6):

$$b_1(\mathbf{y}) = \begin{cases} 2 & \text{if } \mathbf{y} \in \partial\Gamma_\mathrm{obs} \\ 1, & \text{else} \end{cases}, \tag{11}$$

and

$$b_2(\mathbf{y}) = \begin{cases} 0.2 & \text{if } \mathbf{y} \in \partial\Gamma_\mathrm{obs} \\ 1, & \text{else} \end{cases}. \tag{12}$$

Figure 2. demonstrates the resulting trajectories from various initial positions, where we see (more clearly with the green trajectory) the effect of the flux values around $\partial\Gamma_\mathrm{obs}$.

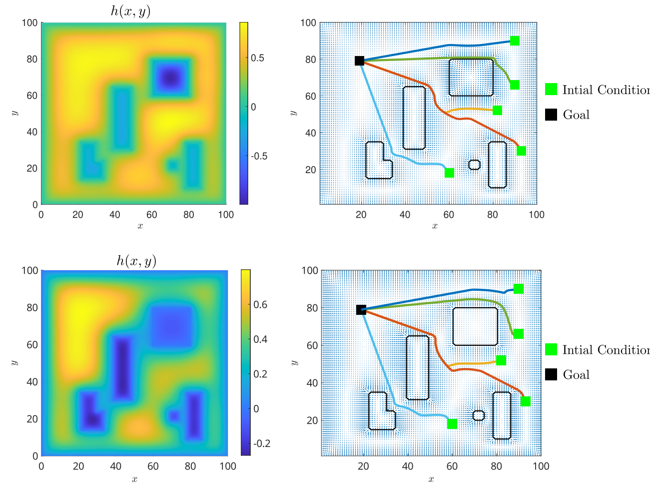

Fig. 2. (Top Row) By leveraging b_1, we increase the magnitude of the gradients (i.e., make them steeper) at the boundary of the top right obstacle and show the resulting trajectories around this obstacle. (Bottom Row) By leveraging b_2, we reduce the magnitude of gradients (make them shallower) and show the resulting conservative behavior.

IV. Discussion and Future Work

The ability to assign obstacle-specific gradients offers several benefits. Among them, it has been shown in [3] to help yield trajectories that avoid deadlocks (e.g., undesired equilibria) resulting from synthesized safety filters. Beyond this, the approach enables the use of semantic labeling—allowing different levels of caution to be assigned based on obstacle type. We believe that this work creates a pathway for semantic-aware safety, where certain obstacles (e.g., humans, fragile objects) are treated with higher priority. More specifically, this provides a pathway for encoding robustness with respect to specific obstacles, enabling the ability to incorporate more context in safety objectives. Future work involves exploring these pathways by integrating semantic information to achieve obstacle-specific robustness, with hardware applications on various robotic platforms.

V. Conclusion

We have presented a method for synthesizing safe sets with tunable boundary gradients using Poisson safety functions. We showed that adjusting the desired boundary flux yields tunable boundary gradients, providing the ability to encode varying gradient magnitudes around obstacle surfaces, and have demonstrated this via empirical results with simulations. Future work will focus on semantic integration and hardware implementation.

References

[1] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[2] A. Alan, A. J. Taylor, C. R. He, G. Orosz, and A. D. Ames, "Safe controller synthesis with tunable input-to-state safe control barrier functions," *IEEE Control Systems Letters*, vol. 6, pp. 908–913, 2022.

[3] G. Bahati, R. M. Bena, and A. D. Ames, "Dynamic safety in complex environments: Synthesizing safety filters with poisson's equation," in *Robotics: Science and Systems*, 2025 (To Appear). Paper link.

[4] D. Gilbarg, N. S. Trudinger, D. Gilbarg, and N. Trudinger, *Elliptic partial differential equations of second order*, vol. 224. Springer, 1977.

[5] A. Isidori, *Nonlinear control systems: an introduction*. Springer, 1985.