Unconstrained Model Predictive Control for Robot Navigation under Uncertainty

Senthil Hariharan Arul[1†,2*], Jong Jin Park[1*], Vishnu Prem[1*], Yang Zhang[3*], and Dinesh Manocha[2]

***Abstract*— In this paper, we present a probabilistic and unconstrained model predictive control formulation for robot navigation under uncertainty. We present (1) a closed-form approximation of the probability of collision that naturally models the propagation of uncertainty over the planning horizon and is computationally cheap to evaluate, and (2) a collision-cost formulation which provably preserves forward invariance (i.e., keeps the robot away from obstacles) when combined with the probability formulation. Notably, our formulation avoids hard constraints by construction, which in turn avoids abrupt transitions in robot behavior around the constraint boundaries ensuring graceful navigation. Further, we present proof for the forward invariance and the stability of the approach. We compare the efficacy of our method with the baseline [1], which the proposed approach builds on. We demonstrate that the approach results in confident and safe robot navigation in tight spaces by smoothly slowing down the robot in low survivability environments (e.g., tight corridors), but also allows it to move away from obstacles safely when needed.**

I. Introduction

Robots' ability to navigate safely and efficiently in real-world scenarios is critical for their widespread deployment. The past two decades have seen significant advancement in robotics, enabling them to move beyond controlled laboratory and industrial scenes and into unstructured environments. Consequently, robot vacuums, self-driving cars, and home monitoring robots operate in everyday environments where they can interact and assist humans. Yet, existing navigation methods still encounter challenges when deployed in highly constrained settings such as public spaces and household environments [2], [3], [4], [5].

Navigating indoor environments is particularly challenging due to their constrained spaces, tight corridors, and the presence of dynamic agents such as humans and pets. Besides, the imperfect sensing and estimation errors make the robot and obstacle locations uncertain. As a result, the navigation algorithms need to consider such uncertainties for safe and reliable navigation. While a conservative robot that always stays away from obstacles is safe, it is equally important for the robots to be efficient and capable of traversing narrow spaces, making sharp turns, and accepting some risk when necessary. However, finding a compute-efficient solution that produces safe and graceful robot motion without being overly conservative in such probabilistic settings is still an open challenge [6].

A. Related Work

Velocity obstacle (VO) [7] computes a set of collision-free relative velocities between the agent and the obstacle for safe navigation. RVO [8] improves VO by assuming equal responsibility between agents in a multi-agent setting, and the idea is formulated as a linear programming problem in ORCA [9]. However, VO and its variants plan in the velocity space and can result in arbitrary acceleration between time steps. A class of methods uses model-predictive control (MPC) to produce predictive navigation behavior by planning over a horizon. In [10], [11], ORCA constraints are combined with MPC to generate collision-free trajectories with smooth velocity variations. In [12], authors present a local planner based on model predictive contouring control (MPCC) for maneuvering in unstructured environments. Recently, learning-based approaches have shown impressive navigation performance in some cluttered scenarios. GA3C-CADRL [13] presents an RL-based navigation approach that outperforms ORCA in many scenarios. Performer-MPC [14] presents a trainable MPC framework to navigate challenging scenarios using a cost function with learnable components. Although Performer-MPC outperforms conventional MPCs in the evaluated scenarios, the cost function is learned individually per scenario, thus lacking generalizability. In general, learning-based methods suffer from a lack of explainability, safety guarantees, and sim-to-real gap. Moreover, they provide limited scope for tuning the navigation behavior after the training phase, which can be desirable in real world deployment.

For safe navigation under sensing and localization uncertainties, it is imperative to consider uncertainty. In [15], [16], the authors inflate the obstacle configuration by a safety distance to account for the localization and sensing uncertainties. While inflated obstacles aid in maintaining a safe distance, they can prevent robots from entering tight corridors and doorways typical in many indoor environments.

Robust MPC planners consider a bounded set of uncertainties to plan a safe trajectory and can provide safety guarantees [17]. However, accounting for the entire uncertainty set can result in infeasible solutions in dense and crowded scenarios. Moreover, the uncertainty distribution such as from Kalman filters, could be unbounded. In [18], a nonlinear-MPC is used for trajectory planning where the

* Authors contributed equally and are listed alphabetically.
† Work done during the internship at Amazon.
[1] Authors are with Amazon Lab126, Sunnyvale, CA 94098, USA. Email: {jongpark,visprem}@amazon.edu
[2] Authors are with the University of Maryland, College Park, MD 20740, USA. Email: {sarul1,dmanocha}@umd.edu
[3] Author is with Latitude AI, 3251 Hillview Ave Building B, Palo Alto, CA 94304. Email: yangzhang@lat.ai This work does not relate to the author's position at Latitude AI.

robot's bounding volume is enlarged by $3-\sigma$ confidence bound. Like other bounding volume expansion methods they can result in conservative behavior or infeasibility.

Recently, chance constraints have been widely used for decision-making under uncertainty [19], where the optimization maintains the probability of the collision constraint violation below a threshold. However, evaluating chance-constrained optimization is computationally intractable, and the constraints are generally approximated by assuming Gaussian uncertainty [20] or simple agent dynamics such as single integrator [21] for tractable implementation. Groot et al. [22] reformulate the chance-constrained optimization using a scenario approach and explicitly constrain the joint collision probability of the planned trajectory over the obstacles. Although chance constraints are less conservative than bounding volume expansion, continually accepting some risk could be inappropriate. Moreover, transitioning across the user-defined probability threshold can result in infeasibility, requiring drastic transitioning behavior such as aggressive braking, which is generally unsuitable for real robots.

Park et al. [1] present MPEPC, a stochastic MPC formulation that generates smooth local trajectories through a low-dimensional, unconstrained optimization. The approach desires probabilistic safety guarantees that are more suitable for real-world scenarios with uncertainty. In [23], this approach has been extend with a terminal cost formulation to reduce deadlocking behavior.

B. Main Contribution

We address the problem of over-conservatism in stochastic trajectory planning by presenting two novel contributions that build on the MPEPC [1], a stochastic MPC framework for navigation. Our main contributions are as follows:

- A closed-form approximation of the collision probability that naturally models the propagation of uncertainty over the planning horizon by incorporating the flattening of the probability distribution. This generalized collision probability improves navigation behavior by resulting in more confident (higher velocity) motion in tight corridors. (Sec. III)
- A collision cost that in addition to limiting the robot's velocity under high collision probability, encourages the robot to navigate away from obstacles when desirable. Thus reducing conservative freezing behavior. (Sec. IV)
- Further, we provide proof for the forward invariance and stability of our navigation approach. (Sec. V)

We present experimental results in both simulated and real-world complex static environments, demonstrating improved navigation behavior compared to the baseline [1]. Further, we analyze the tunable parameters of the collision probability and the collision cost function to demonstrate their effect on the navigation behavior.

II. Preliminaries

We present an overview on the trajectory parameterization and the stochastic MPC framework used in our work.

A. Closed Loop Trajectory Parameterization

Consider navigating a robot to a target pose T located at a distance r away from the robot. Let θ denote the orientation of T with respect to the line of sight between the robot and T. Let δ denote the orientation of the robot with respect to the line of sight. Then the coordinates given by (r, θ, δ) define the target T relative to the robot's pose.

Given a target T, Park et al. [24] define a pose-following control law that can navigate the robot towards (r, θ, δ). The control law is given by,

$$\omega = \frac{v}{r}\left[k_2(\delta - \arctan(-k_1\theta)) + \left(1 + \frac{k_1}{1+(k_1\theta)^2}\right)\sin\delta\right]. \quad (1)$$

Here, v and ω define the linear and angular velocity, respectively, while k_1 and k_2 are gain parameters. A curvature-dependent velocity is chosen based on the maximum velocity v_{max}. Given $z* = (r, \theta, \delta, v_{max})$, the control law completely describes the trajectory of the robot converging to T. Thus, $z*$ parameterize a set of closed-loop trajectories that are realizable by the robot and smooth.

B. Robot Navigation via Stochastic MPC

MPEPC [1] presents a local trajectory planner based on unconstrained model predictive control. At each planning cycle, the planner optimizes to choose a suitable trajectory parameter $z*$ that maximizes the robot's progress towards the goal while maintaining safety and smoothness of the trajectory. Let $q_{z*} : [0, T] \rightarrow C$ denote the continuous trajectory generated by smooth control law for the parameter $z*$ over a time horizon T. Here, $C \simeq R^2 \times S^1$ is the configuration space of the robot. For trajectory evaluation, the continuous trajectory is divided into N trajectory segments. The trajectory cost function is given by,

$$J = \sum_{i=1}^{N} p_{s_i} \cdot J_{progress_i} + J_{action_i} + (1 - p_{s_i}) \cdot J_{collision_i} \quad (2)$$

1) Survivability (p_{s_i}): The probability of collision (p_{c_i}) over the i^{th} trajectory segments is modeled as a bell-shaped function.

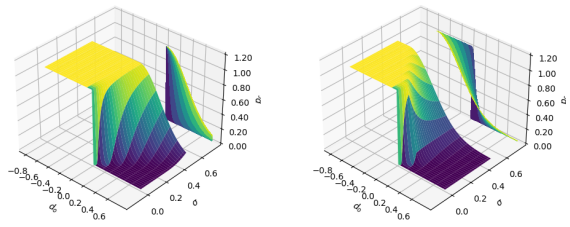
$$p_{c_i}(d_{o_i}, \sigma_i) = \exp\left(-\frac{d_{o_i}^2}{\sigma_i^2}\right) \quad (3)$$

where d_{o_i} is the minimum distance (non-negative) between the robot and the closest obstacle and σ_i is an uncertainty measure associated with the robot's localization. The collision probability is used to define the notion of survivability (p_{s_i}), which is the probability of the trajectory being collision-free until the current segment.

$$p_{s_i} = \prod_{k=1}^{i}(1 - p_{c_k}) \quad (4)$$

2) Uncertainty Estimation: Given an initial uncertainty $\sigma(t_0)$ associated with the robot's localization, the uncertainty grows over the planning horizon and is modelled as a function of the robot's speed [25]. The uncertainty at a time t_i at the end of the i^{th} trajectory segment is given by,

$$\sigma(t_i) = \min(\sigma(t_0) + \textstyle\sum_{k=1}^{i} \sqrt{\lambda_v \cdot \bar{v}_k^2 + \lambda_\omega \cdot \bar{\omega_k}^2}, \sigma_{max}). \quad (5)$$



(a) Baseline p_c **(b)** Generalized p_c (proposed)

Fig. 1: The surface of the collision probability as a function of distance to obstacle and uncertainty values. From the contour plot for the σ axis, we can observe that, for the Baseline p_c (Eqn. 3), the area under the curve monotonically increases with increasing uncertainty value, while for the generalized p_c (Eqn. 8), the probability distribution flattens with increasing uncertainty value.

$\bar{v}$ and $\bar{\omega}$ are the average linear and angular velocities of the robot over the trajectory segment. λ_v and λ_ω are constant weights. The maximum uncertainty is given by σ_{max}. Equation 5 models the intuition that, for a stationary robot, it is not entirely accurate for the uncertainty to grow over the planning horizon. In addition, for feedback controlled systems, the uncertainty is bounded by some σ_{max}.

3) Progress Cost ($J_{progress}$): The progress term measures the distance travelled towards the target T over the trajectory segment. NF is the navigation function; in our implementation it is the Euclidean distance to goal.

$$J_{progress_i} = NF(q_{z^*}(t_i)) - NF(q_{z^*}(t_{i-1}))$$

4) Action Cost (J_{action}): The action cost is a quadratic function on the velocity and limits large actions.

$$J_{action_i} = (c_v \cdot v_i^2 + c_\omega \cdot \omega_i^2) \cdot h$$

Here, c_v, c_ω are constant weights and h denotes the time duration of the trajectory segment.

5) Collision Cost ($J_{collision}$): The collision cost is a function on the robot's velocities and discourages motion when the robot is under a high probability of collision.

$$J_{collision_i} = r_0 + r_v\big(|v_i| + |\omega_i|\big) \cdot h.$$

Here, r_0, r_v are constant weights and h denotes the time duration of the trajectory segment.

III. Generalized Collision Probability

As the uncertainty propagates over the planning horizon, we are less confident about the robot's localization. Thus, the uncertainty in the robot's relative position grows, leading to a flattening of the collision probability distribution. That is, given a long enough planning horizon, the distribution eventually flattens to a uniform distribution.

A limitation of the bell-shaped collision model (Eqn. 3) is its inability to model this propagation of uncertainty. As a result, the probability of collision for any finite distance d increases monotonically with growing uncertainty. In practice, this deficiency manifests itself as a robot behaving conservatively in narrow spaces when it is physically safe to move faster. To address this issue, we present a generalized collision probability which is closed-form approximation that naturally models the flattening behavior with a minimal number of tunable parameters.

A. Modeling Flattening of Collision Probability

In cases where uncertainty is high, an estimated distance of zero to an obstacle no longer guarantees a collision. Thus, we require the robot to significantly overlap with the obstacle to be certain of a collision (i.e., $p_c = 1$). That is, we must have a certain level of penetration or minimum collision depth to be certain of a collision.

Let σ_c denote the critical uncertainty threshold at which the flattening effect begins to show; then minimum collision depth as a function of uncertainty can be represented by

$$\tilde{d}(\sigma|\sigma_c) \equiv \lambda_d \cdot \max(\sigma - \sigma_c,\ 0). \tag{6}$$

In addition, flattening the probability distribution peak also causes a spread of the distribution's tail, which can be modelled by an effective uncertainty value. The effective uncertainty σ_{eff} is given by

$$\sigma_{eff} \equiv (1 + \lambda_d \cdot \lambda_\sigma) \cdot \sigma. \tag{7}$$

Here, λ_d and λ_σ are weight parameters for collision depth and tail spread respectively. Incorporating the collision depth and effective uncertainty with the bell-shaped collision model (Eqn. 3), the generalized collision probability is given by

$$p_c(d, \tilde{d}, \sigma_{eff}) = \exp\left(-\frac{\max(d + \tilde{d},\ 0)^2}{\sigma_{eff}^2}\right). \tag{8}$$

Where d is the minimum separating distance (or penetration depth) between the robot and obstacle. For an uncertainty $\sigma < \sigma_c$, the effective uncertainty $\sigma_{eff} > \sigma$, and the collision probability is equivalent to the original collision probability function (Eqn. 3) with a larger uncertainty. While for $\sigma > \sigma_c$, the effect of peak flattening and tail spread begins to show. Thus, our proposed collision probability provides a larger gradient (than Eqn. 3) to move the robot away from the obstacle in the initial segments of the trajectory, while enabling the planner to choose longer trajectories (especially in tight spaces) due to the modeling of flattening and tail spread in the later segments of the trajectory. We highlight that, for parameters $\sigma_c, \lambda_d, \lambda_\sigma$ set to zero, our proposed collision model reduces to the baseline model (Eqn. 3). The baseline and generalized collision probabilities (proposed) are illustrated in Figure 1. Moreover, with a reliable motion predictor for dynamic objects, we can extend the formulation to dynamic obstacles, factoring in relative velocity and distance over time.

IV. Unfreezing the Robot

Under a high probability of collision (from Eqn. 4, $p_s \to 0$), the collision cost is weighed larger (Eqn. 2) and dictates the robot's navigation behavior. The cost function reaches a minimum when the robot's velocity is zero, encouraging the robot to stop or freeze. We propose a collision cost that smoothly slows down the robot in low survivability condition, but also allows it to move away from obstacles when needed, while provably preserving forward invariance.

Let's consider a scenario with a robot near an obstacle tasked with reaching a goal location in the free space such

that a safe trajectory to the goal exists. Due to the low survivability at the initial state ($p_{s_1} \to 0$), the robot remains stationary, resulting in an overly conservative navigation behavior. We can argue that even under such scenarios, trajectories that increase the distance to the obstacle are safe as they improve the overall safety.

Our proposed cost function models this behavior in static environments, and the collision cost is a function of the robot's velocity and the change in distance to the static obstacle (Δd_o) over the trajectory segment.

We present *active* and *passive* collision cost function which differ in their ability to induce motion in the robot in the absence of any progress cost. The first case maintains the collision term *passive* and merely helps the robot move towards the goal in desirable cases in the presence of a negative progress cost. In the second case, the term is *active* and the cost can repulse the robot from obstacles to improve safety even in the absence of a negative progress cost.

A. Case 1: Passive Collision Term

Here, we maintain the range of collision cost as non-negative (i.e., $J_{collision} \geq 0$). The linear velocity term $|v| \cdot h$ is the distance traversed in that time step. The Δd_o term is a fraction of the $|v| \cdot h$ term for static obstacles. Thus, weighing Δd_{obs} term lower than $|v| \cdot h$ ensures the collision term will remain non-negative, but its positive value is reduced in trajectory segments that increase the distance to obstacles. Given a sufficient fraction of the progress term is acting, the robot would be able to move towards the goal.

$$J_{collision_i} = (c \cdot [|v_i| + |\omega_i|] \cdot h - c' \cdot \Delta d_{o_i}) \tag{9}$$

$$0 \leq c' \leq c \leq 1 \tag{10}$$

B. Case 2: Active Collision Term

Here, we weigh the Δd_o larger than the velocity term and can actively (over-)compensate for the velocity term. The collision term can have a negative range and provides a gradient maneuvering the robot away from obstacles even without a negative progress cost.

However, naively allowing a negative value for our collision cost by having $c' > c$ can result in unsafe behavior. Consider a trajectory that passes through an obstacle initially but eventually moves to a safe distance away from the obstacle. In this case, the negative collision cost from moving away from the obstacle can compensate for the positive cost from a few trajectory segments in collisions. That is, even though the trajectory is colliding, the summation of the collision cost terms over the entire trajectory is negative. Ideally, we require the robot to be stationary when in a collision to not worsen the situation by moving.

To ensure this safe behavior, we introduce a function I_{coll} that indicates if the trajectory until the current trajectory segment is collision-free. If not, the Δd_o term is reduced to zero. This ensures that the summation of collision cost terms over a trajectory is negative only when the robot moves away from the obstacle using a non-colliding trajectory. The collision cost is given by,

$$\mathrm{J}_{collision_i} = (c \cdot \big[|v_i| + |\omega_i|\big] \cdot h - c' \cdot I_{coll}(i) \cdot \Delta d_{o_i}). \tag{11}$$

The weights c, c' and the indicator I_{coll} are defined as,

$$0 \leq c \leq 1, \quad c' > c \tag{12}$$

$$I_{coll}(i) = I_{coll}(i-1) \cdot \mathbb{1}(p_c(i) > p_c^*). \tag{13}$$

p_c^* is user-defined probability threshold to indicate collision.

V. Safety and Stability

In this section, we prove the forward invariance and stability of our navigation approach in static environments. For simplicity, we assume a circular agent (ω does not affect the distance to goal or obstacle) and we consider only the progress and collision cost terms in the following proofs. The cost function is given by,

$$J = \sum_1^N p_{s_i} \cdot \Delta d_{p_i} + (1 - p_{s_i}) \cdot (c \cdot |v_i| \cdot h - c' \Delta d_{o_i}),$$

where, Δd_{p_i} is the progress made towards the goal, and Δd_{o_i} is the change in distance to the static obstacle over the trajectory segment. A negative Δd_{p_i} implies the robot moves towards the goal and a negative Δd_{o_i} implies the robot moves towards the obstacle.

Proposition 1. *The $(1 - p_{s_i})$ term is monotonically non-decreasing over time. That is, $(1 - p_{s_1}) \leq (1 - p_{s_2}) \leq ... \leq (1 - p_{s_n})$. Proof: From p_s definition (Eqn. 4).*

Proposition 2. *Goal progress (d_{p_i}) and the change in distance to the static obstacles (d_{o_i}) are bounded by the distance moved over the trajectory segment (product of robot's absolute velocity and time step).*

$$\Delta d_{p_i} = \alpha_i \cdot |v_i| \cdot h \leq |v_i| \cdot h, \quad -1 \leq \alpha_i \leq 1 \tag{14}$$

$$\Delta d_{o_i} = \beta_i \cdot |v_i| \cdot h \leq |v_i| \cdot h, \quad -1 \leq \beta_i \leq 1 \tag{15}$$

Definition 1. *Let $\mathcal{C} \subset \mathbb{R}^n$ be a set defined as the super-level set of a continuously differentiable function $h : \mathcal{D} \subset \mathbb{R}^n \to \mathbb{R}$. Then any trajectory $q_z* = \arg\min J$ for the system $\dot{x} = f(x)$ render the set $\mathcal{C}$ invariant. Additionally, minimizing J outside $\mathcal{C}$ either retains or reduces the distance to set $\mathcal{C}$, showing stability.*

Lemma 1. Forward Invariance: *A robot starting within a set $\mathcal{C}$ (safe set) continues to remain within $\mathcal{C}$ provided the trajectory is selected by minimizing cost function J.*
Proof: The trajectory cost can be written as,

$$J = \sum^N \Big(p_{s_i} \cdot \big(\Delta d_{p_i} - (c \cdot |v_i| \cdot h - c' \cdot \Delta d_{o_i})\big) + c \cdot |v_i| \cdot h \Big) - c' \cdot \sum^N \Delta d_{o_i}.$$

To prove forward invariance, let us consider the robot at the end of the trajectory has moved closer to the obstacle. That is, $\sum^N \Delta d_{o_i} < 0$. This results in $-c' \cdot \sum^N \Delta d_{o_i} > 0$. For $|v_i| = 0$, we have $J = 0$. Since we minimize J, for the

(a) $t = 0$s **(b)** $t = 10$s **(c)** $t = 20$s

Fig. 2: A robot navigating through a narrow, L-shaped corridor using the baseline p_c formulation in MPEPC [1]. The over-estimation of the collision probability hinders the robot from confidently navigating through the narrow corridor.

cost function to induce motion, J should be negative. Thus, for one or more trajectory segments i, we need

$$p_{s_i} \cdot \left(\Delta d_{p_i} - (c \cdot |v_i| \cdot h - c' \cdot \Delta d_{o_i})\right) + (c \cdot |v_i| \cdot h) < 0.$$

From proposition 2, we can write

$$\Rightarrow \left(p_{s_i} \cdot \left(\alpha_i - c + c' \cdot \beta_i\right) + c\right) \cdot |v_i| \cdot h < 0$$

Since $|v_i| \cdot h \geq 0$, and by setting the minimum value for $\alpha, \beta = -1$, let us identify the constraint on p_{s_i} which would always keep the term non-negative. We get,

$$p_{s_i} \leq \frac{c}{1 + c + c'}$$

With initial trajectory segment $i = 1$ above this p_{s_i} threshold (then $p_{s_i} < p_{s_1}, \quad \forall i > 1$ from Prop. 1), the cost function will not induce motion and this loose bound defines the boundary of the safety set.

$$\mathcal{C} = \left\{ x \in \mathcal{D} \;\middle|\; p_{s_1} \geq \frac{c}{1 + c + c'} \right\}$$

We highlight that the I_{coll} in the active collision term does not affect the proof. As even if a trajectory passes through the obstacle (at a trajectory segment k) starting from the safe set $\mathcal{C}$, the $\sum^k \Delta d_{o_i} < 0$ is true, since the robot moves towards the obstacle.

Lemma 2. Stability: *A robot starting in $\mathcal{D}$ outside the safe set $\mathcal{C}$, selecting a trajectory by minimizing cost J reduces its distance to the safe set at trajectory end state when it moves.*
Proof: Consider the robot is located outside set $\mathcal{C}$, and the robot moves, that is, $J < 0$. From the above lemma we know the when the bound on p_1 is not satisfied, the following term is non-negative. That is,

$$\sum^N \left(p_{s_i} \cdot \left(\Delta d_{p_i} - (c \cdot |v_i| \cdot h - c' \cdot \Delta d_{o_i})\right) + c \cdot |v_i| \cdot h\right) \geq 0.$$

Hence, for $J < 0$, we need

$$-c' \cdot \sum^N \Delta d_{o_i} < 0 \Rightarrow \sum^N \Delta d_{o_i} > 0.$$

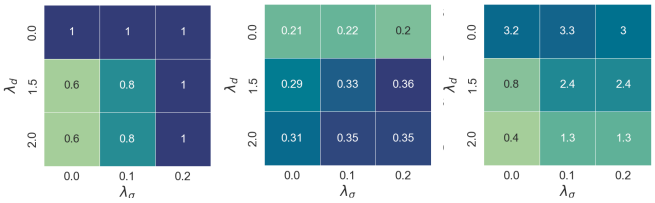
That is, the robot moves away from the obstacle. In the active case, if the trajectory passes through an obstacle then $\sum^N \Delta d_{o_i} > 0$ is not possible because of the I_{coll}.

VI. Results and Discussion

In this section, we evaluate our approach against the baseline (MPEPC [1]) in both real-world and Gazebo simulations using a non-holonomic ground robot measuring ≈ 50cm $\times$ 25cm. Our implementation utilizes NLOPT [26], an off-the-shelf optimization package.

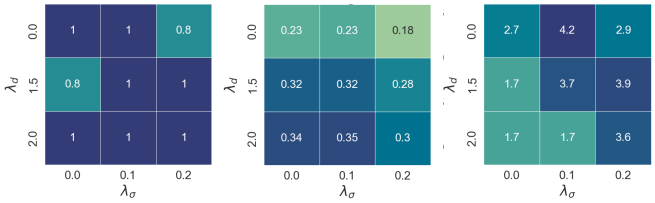
(a) $t = 0$s **(b)** $t = 10$s **(c)** $t = 20$s

Fig. 3: The figure illustrates the robot navigating through a narrow, L-shaped corridor. The proposed p_c formulation successfully navigates the robot to the goal.



(a) Success Rate **(b)** Mean Velocity **(c)** Min. d_{obs}

Fig. 4: Heat maps (for $\sigma_c = 0$) illustrating the effect on navigation performance with λ_d and λ_σ values while navigating a narrow U-shaped corridor. We observe increasing λ_d improves robots mean velocity while also affecting success. While increasing λ_σ increases the minimum distance to obstacle.



(a) Success Rate **(b)** Mean Velocity **(c)** Min. d_{obs}

Fig. 5: Heat maps (for $\sigma_c = 0.01$): We observe increasing the σ_c value from Fig. 4 improves the success rate while maintaining a higher mean velocity than [1] ($\lambda_d = 0$, $\lambda_\sigma = 0$ case in Fig. 4).

(a) MPEPC [1] **(b)** w/ Passive Term **(c)** w/ Active Term

Fig. 6: This figure compares the performance of the original MPEPC formulation with our proposed approach with passive (center) and active (right) collision terms. The star indicates the goal location.

A. Narrow Space Navigation

Figs. 2 and 3 illustrates the robot's motion through a narrow, L-shaped corridor. The original collision probability formulation (Fig. 2) overestimates the collision probability and behaves conservatively, failing to guide the robot to the goal. In contrast, the proposed collision probability formulation (Fig. 3) swiftly navigates the robot through the corridor. In addition, the proposed formulation results in confident navigation (higher speed) in the straight section of the corridor (between $t = 0$ to 10s).

The tunable parameters $(\sigma_c, \lambda_d, \lambda_\sigma)$ in p_c affect the speed and safety of the navigation. Through Figs. 4 and 5, we

(a) $t = 0$s **(b)** $t = 2$s **(c)** $t = 4$s **(d)** $t = 0$s **(e)** $t = 2$s **(f)** $t = 4$s

Fig. 7: Real-World Experiments: Snapshots of the robot navigating through a narrow space between the wall and the couch. **[Figs. (a)-(c)]** The baseline (MPEPC [1]) is used in this case, which overestimates the collision probability, preventing the robot from navigating through the passage. **[Figs. (d)-(f)]** The proposed generalized collision probability is used in this case, which models the propagation of uncertainty over the planning horizon, helping the planner choose longer trajectories. The robot is seen to navigate confidently through the passage. The parameter values used are $(\lambda_d = 1.5, \lambda_\sigma = 0.1, \sigma_c = 0.01)$.

(a) MPEPC [1] **(b)** w/ Passive Term **(c)** w/ Active Term

Fig. 8: Real-World Experiments: We evaluate our approach in the scenario with the robot initially at the goal. Our approach with the active collision terms is seen to move the robot away from the wall to improve safety.

demonstrate the influence of tunable parameters $(\sigma_c, \lambda_d, \lambda_\sigma)$ on the navigation performance, particularly the success rate, the mean velocity of the robot, and the minimum distance to the obstacle. We consider a U-shaped corridor for this evaluation, and we present the results as a set of 2D heatmaps for each σ_c value. We choose three values each for λ_d and λ_σ, which gives us nine permutations for a given σ_c value. Each heatmap entry represents an average value from five simulation runs.

Fig. 4 represents the case with $\sigma_c = 0$. The parameters $\lambda_d = 0, \lambda_\sigma = 0$ represent the baseline p_c. We observe that the original formulation navigates the robot to the goal safely, but the robot navigates with a mean velocity of ~ 0.2m/s, which is lower than the other cases. In addition, the robot maintains a minimum distance of ~ 3.2cm to the obstacles, which is larger than the other cases.

On increasing the value for λ_d, the minimum collision depth required to confirm a collision increases. This reduces the collision probability in comparison to the original formulation, resulting in a slightly lower success rate (in Fig. 4.a)), higher mean velocity (in Fig. 4.b)), and lower minimum distance to the obstacles (in Fig. 4.c) in comparison to the original formulation. Increasing the value for λ_σ increases the tail spread (increases the effective uncertainty) and shows a high success rate and increased minimum distance to the obstacle.

σ_c determines when the effect of minimum collision depth starts to act. From Fig. 5, we observe that for $\sigma_c = 0.01$, the success rate increases while still maintaining the improved mean robot velocity. Hence, it results in confident navigation while still maintaining safety.

B. Unfreezing the robot

In Fig. 6, we consider a scenario with the robot located in close proximity to a wall. The collision probability at the initial state of the trajectory is near one due to the relatively small distance to the wall and the localization uncertainties. For this evaluation, the we use the value $c = 0.5$ and $c' = 0.49$ for the passive collision term, and $c = 0.5$ and $c' = 0.6$ for the active collision term.

In Fig. 6, the robot is tasked with reaching a goal in front of it (represented by a star). We observe the baseline MPEPC formulation fails to navigate the robot to the goal, as the collision cost minimizes the robot's velocity to remain safe. In the case of our proposed passive and active collision term (Fig 6.b-6c), successfully navigates the robot to the goal. We note that chance constraint methods [20] can be impractical in such a scenario as the initially high collision probability leads to infeasible constraints, causing robot freezing.

C. Real-world Experiments

Fig. 7 show a sequence of snapshots of a robot navigating through a narrow passage between a wall and the couch. We observe in the case with the proposed approach (Fig. 7.d-7.f), the robot navigates confidently through the passage, while with MPEPC (Fig. 7.a- 7.c) the robot hesitates to enter the passage. Our proposed method distinctly outperforms the baseline in this case.

In Fig. 8, the robot is already at the goal near a wall. From a safety standpoint, it is desirable for the robot to increase its distance to the wall. With the MPEPC formulation and our proposed approach with passive collision cost, the robot remains stationary, while the approach with active collision cost provides a gradient to repulse the robot from the wall.

VII. Conclusion and Future Work

In this paper, we address the problem of over-conservatism in probabilistic trajectory planners. We introduced a closed-form approximation for the collision probability, which results in confident navigation (i.e., higher velocity) in tight spaces such as narrow corridors. In addition, we presented two formulations for collision cost that both slow robots down in low survivability and allows them to move away from obstacles. Further, we showed our approach maintains forward invariance and stability. As a future work, we plan address more dynamic cases involving multiple pedestrians.

REFERENCES

[1] J. J. Park, C. Johnson, and B. Kuipers, "Robot navigation with model predictive equilibrium point control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 4945–4952.

[2] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *J. Hum.-Robot Interact.*, vol. 12, no. 3, apr 2023. [Online]. Available: https://doi.org/10.1145/3583741

[3] X. Xiao, Z. Xu, Z. Wang, Y. Song, G. Warnell, P. Stone, T. Zhang, S. Ravi, G. Wang, H. Karnan, J. Biswas, N. Mohammad, L. Bramblett, R. Peddi, N. Bezzo, Z. Xie, and P. Dames, "Autonomous ground navigation in highly constrained spaces: Lessons learned from the benchmark autonomous robot navigation challenge at icra 2022 [competitions]," *IEEE Robotics Automation Magazine*, vol. 29, no. 4, pp. 148–156, 2022.

[4] R. Mirsky, X. Xiao, J. Hart, and P. Stone, "Prevention and resolution of conflicts in social navigation–a survey."

[5] X. Xiao, Z. Xu, G. Warnell, P. Stone, F. G. Guinjoan, R. T. Rodrigues, H. Bruyninckx, H. Mandala, G. Christmann, J. L. Blanco-Claraco, *et al.*, "Autonomous ground navigation in highly constrained spaces: Lessons learned from the 2nd barn challenge at icra 2023," *arXiv preprint arXiv:2308.03205*, 2023.

[6] Y. Yang, J. Pan, and W. Wan, "Survey of optimal motion planning," *IET Cyber-Systems and Robotics*, vol. 1, no. 1, pp. 13–19, 2019. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/iet-csr.2018.0003

[7] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998. [Online]. Available: https://doi.org/10.1177/027836499801700706

[8] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 1928–1935.

[9] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*, C. Pradalier, R. Siegwart, and G. Hirzinger, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 3–19.

[10] H. Cheng, Q. Zhu, Z. Liu, T. Xu, and L. Lin, "Decentralized navigation of multiple agents based on orca and model predictive control," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3446–3451.

[11] S. H. Arul and D. Manocha, "Dcad: Decentralized collision avoidance with dynamics constraints for agile quadrotor swarms," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1191–1198, 2020.

[12] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4459–4466, 2019.

[13] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, Sept. 2018. [Online]. Available: https://arxiv.org/pdf/1805.01956.pdf

[14] X. Xiao, T. Zhang, K. M. Choromanski, T.-W. E. Lee, A. Francis, J. Varley, S. Tu, S. Singh, P. Xu, F. Xia, L. Takayama, R. Frostig, J. Tan, C. Parada, and V. Sindhwani, "Learning model predictive controllers with real-time attention for real-world navigation," in *6th Annual Conference on Robot Learning*, 2022. [Online]. Available: https://openreview.net/forum?id=7Nwds2LjN1s

[15] D. Hsu, R. Kindel, J.-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," *The International Journal of Robotics Research*, vol. 21, no. 3, pp. 233–255, 2002.

[16] W. Chung, G. Kim, M. Kim, and C. Lee, "Integrated navigation system for indoor service robots in large-scale environments," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 5, 2004, pp. 5099–5104.

[17] A. Ben-Tal and A. Nemirovski, "Robust convex optimization," *Mathematics of Operations Research*, vol. 23, no. 4, pp. 769–805, 1998. [Online]. Available: http://www.jstor.org/stable/3690632

[18] M. Kamel, J. Alonso-Mora, R. Siegwart, and J. Nieto, "Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 236–243.

[19] D. Lenz, T. Kessler, and A. Knoll, "Stochastic model predictive controller with chance constraints for comfortable and safe driving behavior of autonomous vehicles," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, 2015, pp. 292–297.

[20] H. Zhu and J. Alonso-Mora, "Chance-constrained collision avoidance for mavs in dynamic environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 776–783, 2019.

[21] B. Gopalakrishnan, A. K. Singh, M. Kaushik, K. M. Krishna, and D. Manocha, "Prvo: Probabilistic reciprocal velocity obstacle for multi robot navigation under uncertainty," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1089–1096.

[22] O. de Groot, L. Ferranti, D. Gavrila, and J. Alonso-Mora, "Scenario-based motion planning with bounded probability of collision," *arXiv preprint arXiv:2307.01070*, 2023.

[23] S. H. Arul, J. J. Park, and D. Manocha, "Ds-mpepc: Safe and deadlock-avoiding robot navigation in cluttered dynamic scenes," *ArXiv*, vol. abs/2303.10133, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257622799

[24] J. J. Park and B. Kuipers, "A smooth control law for graceful motion of differential wheeled mobile robots in 2d environment," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 4896–4902.

[25] J. J. Park, "Graceful navigation for mobile robots in dynamic and uncertain environments." 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:196089984

[26] S. G. Johnson, "The NLopt nonlinear-optimization package," https://github.com/stevengj/nlopt, 2007.