Safe Planning via Receding Hamilton Jacobi Reachability

Rohan Panicker & Vattanary Tevy
University of Washington

Abstract

Safety is non-negotiable for autonomous robots operating in dynamic environments. As autonomy scales, we want to close the gap between safety and real-time efficient control. Hamilton Jacobi (HJ) Reachability is a powerful method for analyzing safety, but it is **hard to compute** in real-time with dynamic obstacles. We propose a method of computing reachability for local regions in a **receding fashion**, thus **saving compute power and time while being safe.**

Background & Motivation

Model Predictive Path Integral (MPPI) is widely used to produce fast trajectories in nonlinear systems. However, s**tandard MPPI lacks safety reasoning**, making it vulnerable in safety-critical scenarios involving dynamic obstacles or rare but dangerous events. To address this, we integrate **risk-aware safety filters**–notably receding Hamilton-Jacobi (HJ) reachability and Control Barrier Functions (CBFs)--into the MPPI framework.

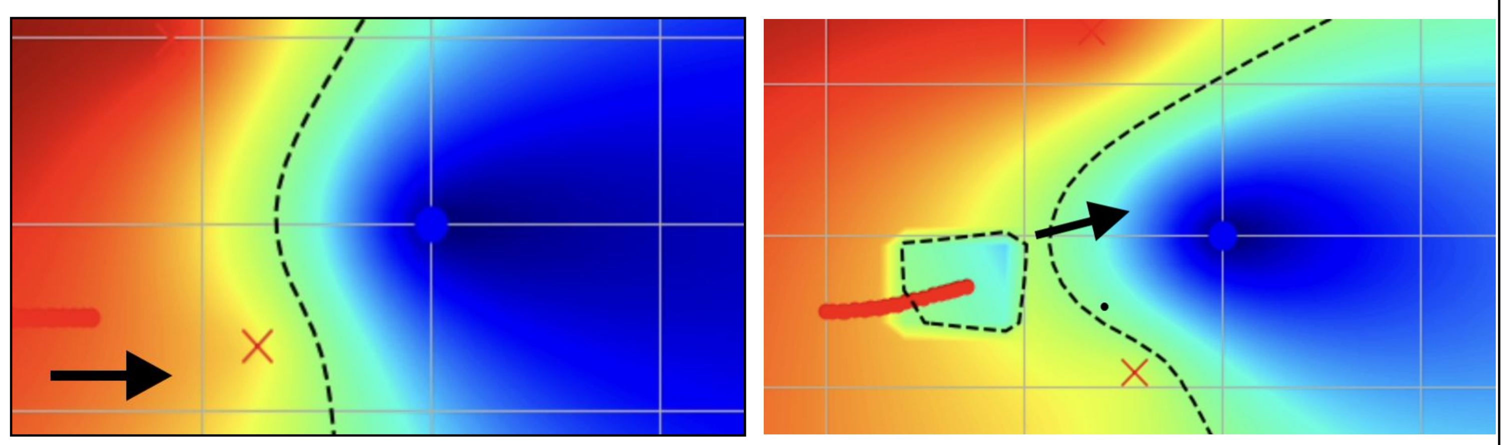

Fig. 1: (a) showcases MPPI-CBF with the standard global signed distance value function. (b) showcases MPPI-CBF with receding HJ reachability method which computes the safe sets for a time horizon within the local map. Both the subfigures are taken at for the same timestep.

Proposed Approach

With a receding local approach to HJ reachability, this patch-wise solution **reduces computation**, **enables adaptation to dynamic environments**, and **maintains safety immediately ahead** of the robot.

Core Concept:

- Baseline method
  - MPPI with CBF safety filter.
- Extension
  - Baseline + Receding HJ

Algorithm 1 Receding Horizon HJI-MPPI Planning

Require: Global value function V_{global}, time horizon T, initial state x_0
Ensure: Executed trajectory of the agent

```
x ← x_0
while not GoalReached(x) do
    (V_local, G_local) ← ExtractLocalPatch(V_global, x)
    τ ← 0
    while τ < T do
        V_local ← HJI_Backward_Update(V_local, G_local)
        τ ← τ + Δt
    end while
    u ← MPPI_Control(x, V_local)
    x ← StepDynamics(x, u)
    V_global ← UpdateGlobalValue(x, V_global)
end while
return Trajectory =0
```

$$\frac{\partial V_{\text{local}}(x,t)}{\partial t} + \min_{u \in \mathscr{U}} \max_{d \in \mathscr{D}} \left\{ \nabla V_{\text{local}}(x,t)^\top f(x,u,d) \right\} = 0$$

$$\text{where } x \in \mathscr{X}_{\text{patch}}, \quad t \in [t_0, t_0 + \Delta T]$$

Eq. 1: Backward Reachability Tube (BRT) for a local region.

$$u^\star = \arg\min_u \ \|u - u_{\text{nom}}\|_2^2$$

$$\text{s.t.} \quad \nabla b(x)^\top f(x,u) \geq -\alpha(b(x))$$

Eq. 2: Control Barrier Functions (CBF) based safety filter. Project nominal control u_{nom} onto the set of inputs that keep the system within a safe set defined by $b(x) \geq 0$.

$$\mathbf{u}_j^* = \mathbf{u}_j + \boldsymbol{\mathcal{H}}^{-1}\boldsymbol{\mathcal{G}} \left(\mathbb{E}_{q_u^\nu} \left[\frac{\exp(-\frac{1}{\lambda}\tilde{S}(\tau))\frac{\epsilon_j}{\sqrt{\Delta t}}}{\mathbb{E}_{q_u^\nu}\left[\exp\left(-\frac{1}{\lambda}\tilde{S}(\tau)\right)\right]} \right] \right)$$

Eq. 3: Model Predictive Path Integral (MPPI) update rule. Compute control $\mathbf{u}_j^*$ as a weighted average of noise perturbations.

Experimental Results

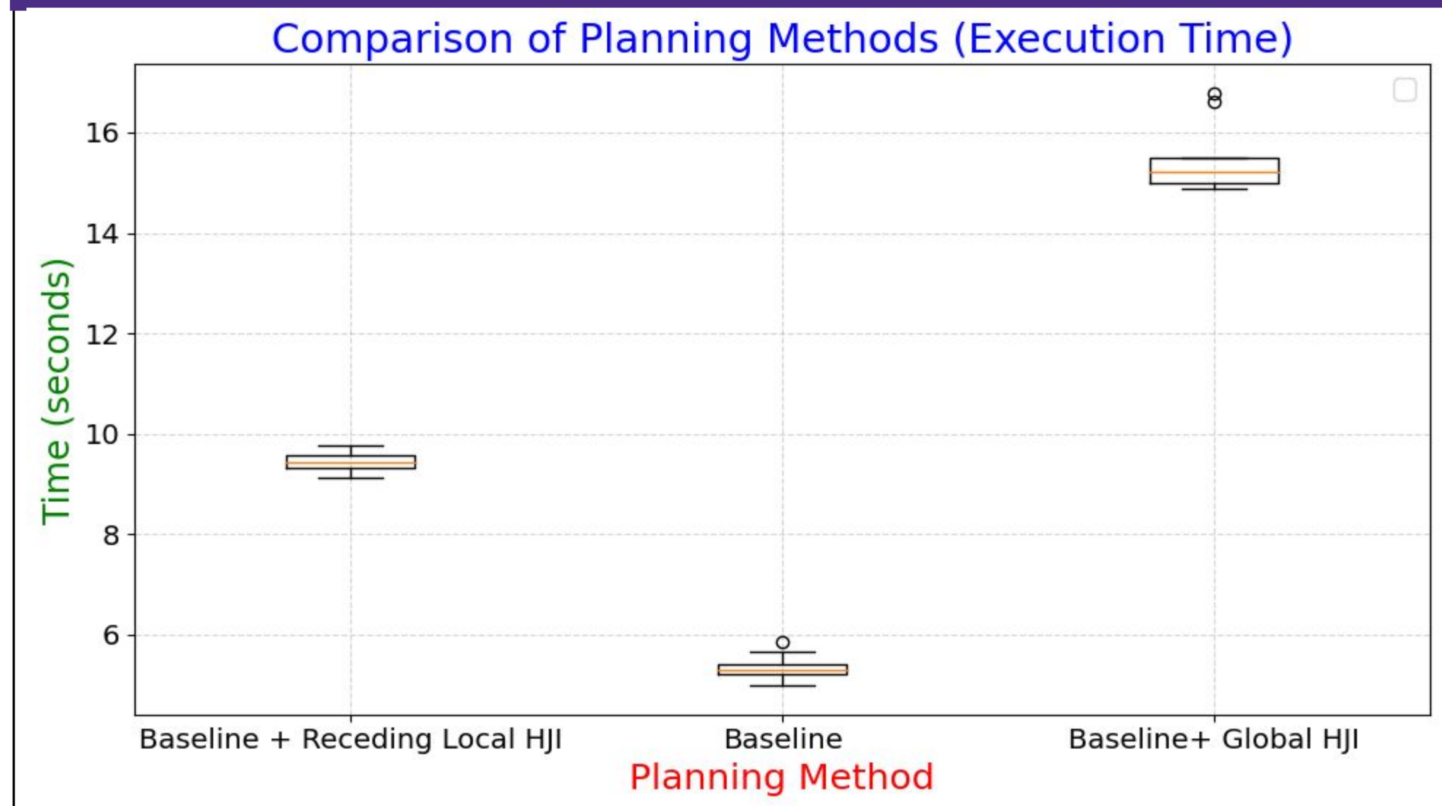

Fig. 2: Execution Speed comparison between baseline & variants.

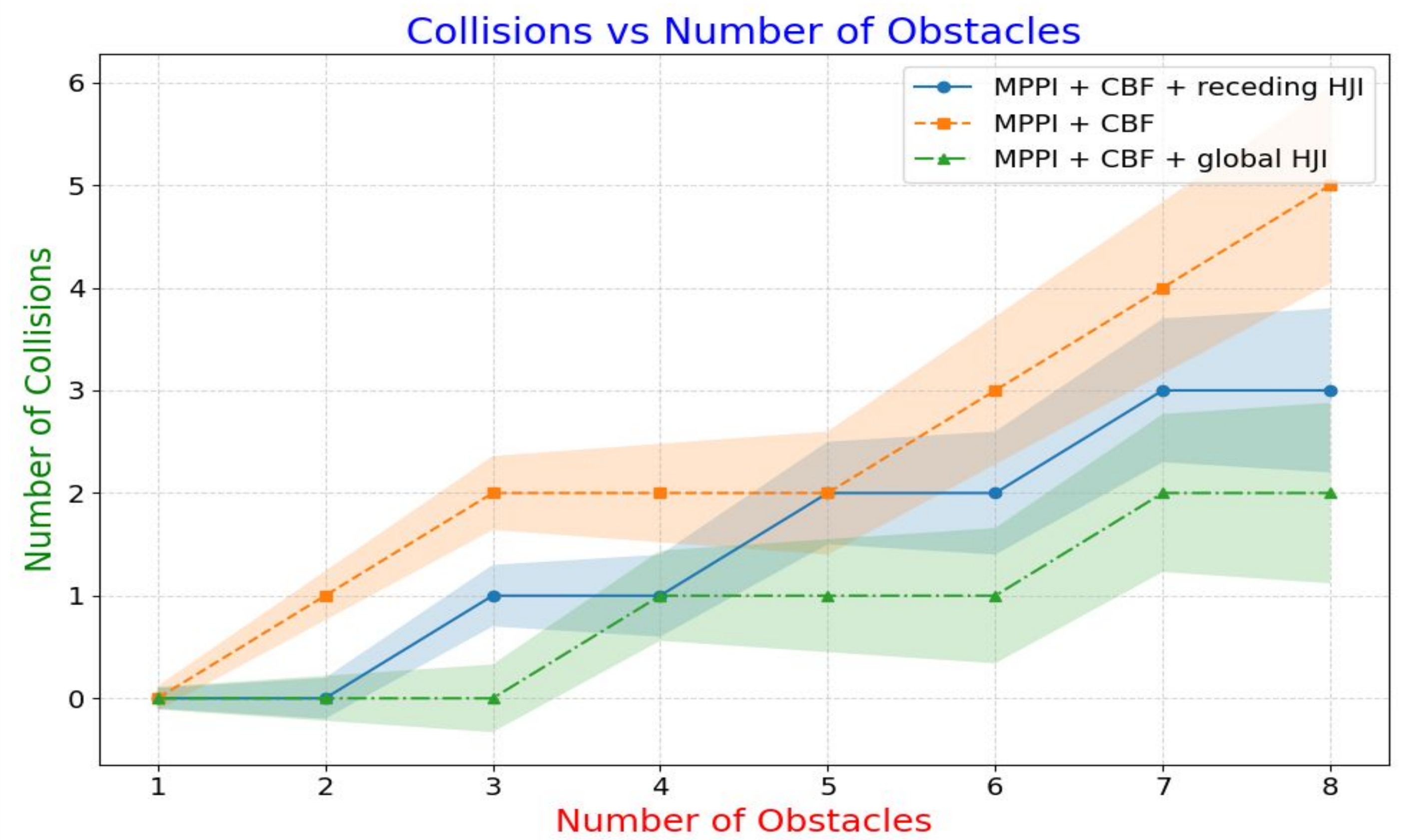

Fig. 3: Collisions vs number of obstacles for baseline & variants.

Conclusion

Takeaways:

- **Algorithmic and mathematical** way to to reason about safety and goal-directed behavior under worst-case disturbances
- Potential to integrate reachability based algorithms **close to real-time.**

Future Work:

- Since the **local reachability solution may be suboptimal due to incorrect boundary conditions for long time horizons,** function approximators can be used to mitigate this problem.