ICRA 2024 ThAT33.03

Safe Receding Horizon Motion Planning with Infinitesimal Update Interval

Inkyu Jang, Sunwoo Hwang, Jeonghyun Byun, and H. Jin Kim
Department of Aerospace Engineering, Seoul National University, Seoul, Korea.

Summary

- We take **infinitesimal update interval** for a safety-aware receding horizon motion planning problem to get a **real-time safety filter**.
- The proposed algorithm is **safe**, **recursively feasible**, and runs in **real-time**.
- The proposed motion planning method was validated through **hardware experiments**.

Notes on the Replanning Interval

- It denotes **how promptly the system can respond** to possibly changing environmental conditions.
- ➙ It is favorable to keep the interval **as short as possible**.
- It serves as the **time budget to complete searching** for the next trajectory.
- ➙ It **cannot be shortened** beyond the robot's onboard computation capacity.

In this work, we consider the limit ***replanning interval* ↘ *0*.**

Trajectory parameter optimization

$$\begin{aligned} \min_{k}\quad & J(k) && \text{(trajectory cost)} \\ \text{s.t.}\quad & \mathrm{x}(k,0) = x && \text{(initial condition)} \\ & w(k) \geq 0 && \text{(safety)} \\ & \rho(k) \geq 0 && \text{(parameter feasibility)} \end{aligned}$$

- Feasibility not guaranteed, **needs an emergency brake maneuver** for safety.
- **Nonlinear, nonconvex** optimization, heavy computation.
- The update interval cannot be shortened beyond the robot's onboard computation power.

Trajectory parameter *change rate* optimization

$$\begin{aligned} \min_{u,\dot{k},\dot{\tau}}\quad & \partial_k J(k)\cdot\dot{k} + \lambda\dot{\tau} + \frac{1}{2}\mu_k\|\dot{k}\|^2 + \frac{1}{2}\mu_\tau\dot{\tau}^2 && \text{(trajectory cost change rate)} \\ \text{s.t.}\quad & \partial_k \mathrm{x}(k,\tau)\cdot\dot{k} + \partial_\tau \mathrm{x}(k,\tau) = f(x) + g(x)u && \text{(initial condition)} \\ & \partial_k w(k)\cdot\dot{k} + \alpha(w(k)) \geq 0 && \text{(safety)} \\ & \partial_k \rho(k)\cdot\dot{k} + \gamma(\rho(k)) \geq 0 && \text{(parameter feasibility)} \\ & \dot{\tau} + \sigma(\tau) \geq 0 && \text{(trajectory starting point feasibility)} \\ & u \in U && \text{(input feasibility)} \end{aligned}$$

- The trajectory cost is replaced with its time derivative, plus some auxiliary terms for numerical stability.
- Equality constraint satisfied indirectly using time derivatives.
- Inequality constraint satisfied using CBF-QP-style class-K functions.

✓ **Convex quadratic program** (QP), which can be solved in **real-time**.
✓ **Safety filter** that directly gives the input u.
✓ **Recursively feasible**, given the trajectories satisfy final stop conditions.

x, u	: State and input variables	$J(k)$	: Differentiable trajectory cost
k	: Trajectory parameter	$\mathrm{x}(k,\tau)$	: Parametrized trajectory with final stop condition
τ	: Indicator at which time point on the parametrized trajectory the robot will start	w, ρ	: Differentiable safety and feasibility constraints
		α, γ, σ	: Element-wise class-K functions

Experiment Result

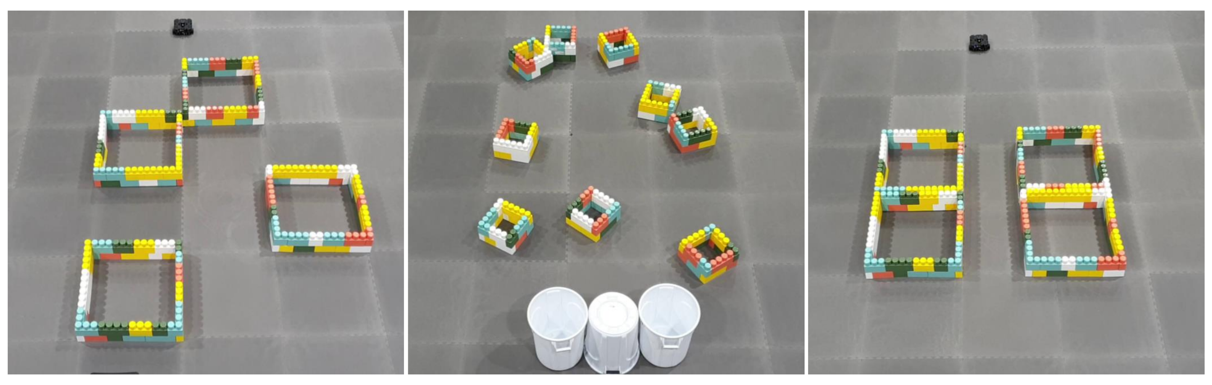
Experiment setting

- Diff-drive robot, controlled by linear / angular accelerations.
- 2D LiDAR sensor which scans obstacle distance from all directions. (up to 360 points)

Safety requirements

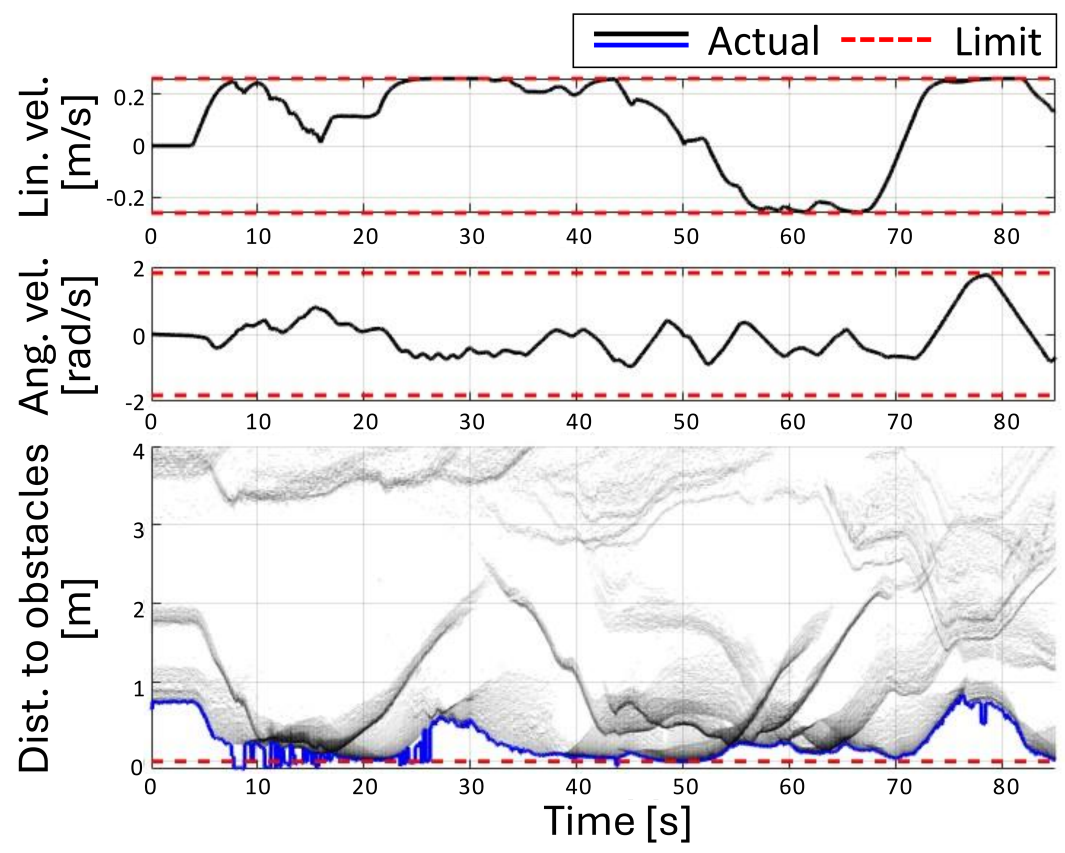
- Collision avoidance with all the LiDAR points.
- Linear / angular speed limits

Experiment environments



Exp. video @ YouTube

- The **trajectory cost was given in an aggressive way** that the robot is driven towards the obstacles.
- Even under aggressive trajectory cost, the robot was able to **avoid hundreds of LiDAR points simultaneously** using onboard computation only.



A snapshot from the experiment.