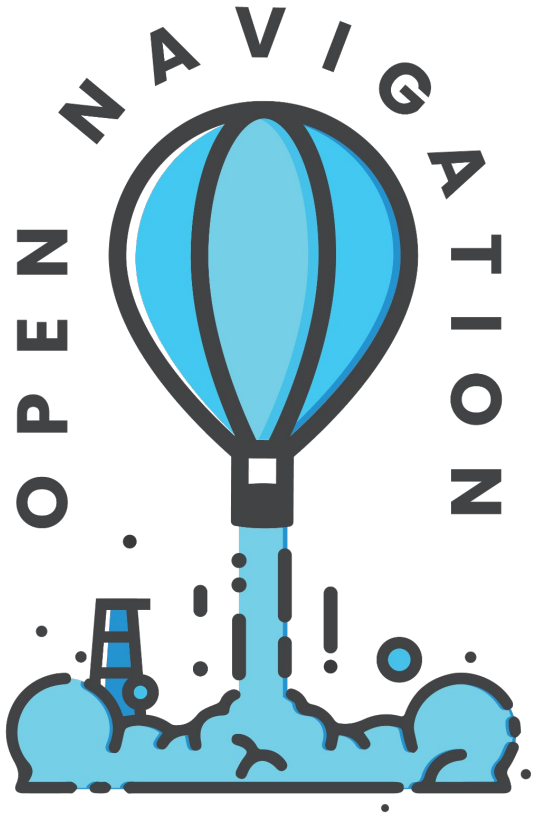
On Use of Nav2 MPPI Controller

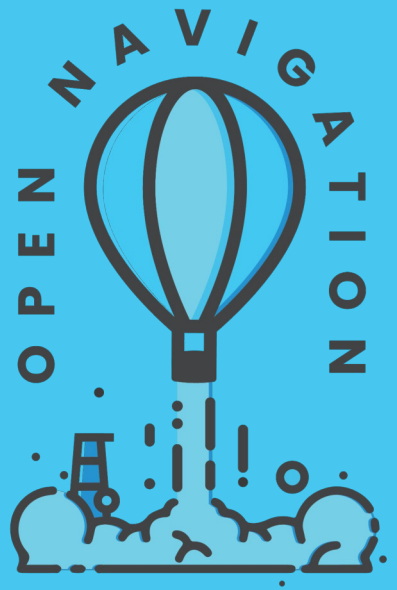


Steve Macenski, Open Navigation LLC

Open Navigation LLC

Launching Nav2, ROS, and the Community into the Future!

- Your Friendly Neighborhood Navigators!
- Develop & Maintain Open-Source Mobile Robotics, Long-Term
- Sponsored by Those Using ROS / Nav2 for Products & Services
- Offering Support Services to Power Users' Success

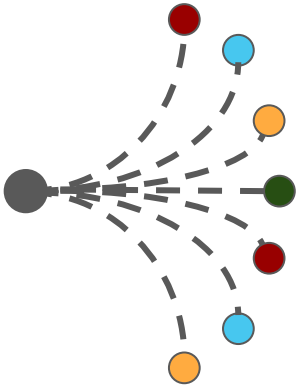


Looking for Additional Sponsors and Partners!

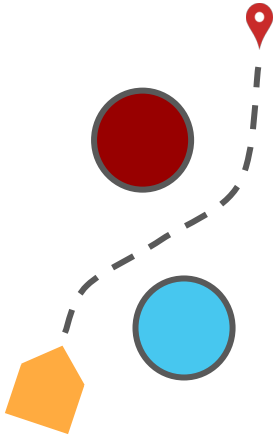
- Find me after the talk or email me at **info@opennav.org**

Overview

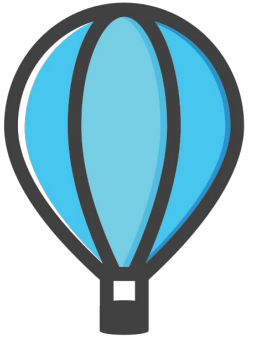
Background



MPPI Overview



Nav2 MPPI Controller



Background - Trajectory Planning

“How Do I Follow This Objective?”

Trajectory Planning Aims to:

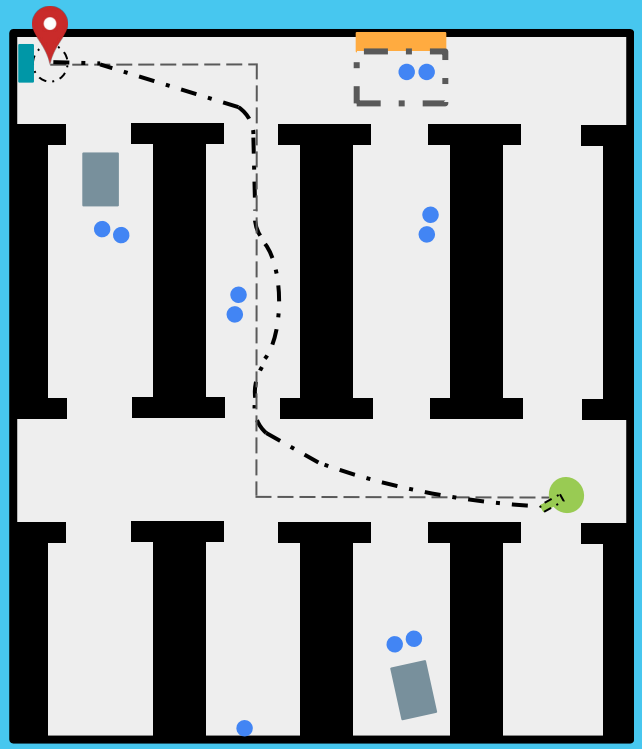
- Track paths or goals
- Avoid obstacles
- Compute feasible commands
- Optimize for behavioral objectives
- Achieve goal(s) with high accuracy

This is the Behavior of a Robot a Person Sees

- Important beyond engineering metrics!

Many Types of Approaches Exist

- Solve for 1+ or all of the *aims*
- MPC, DWA, PP, Splines, Primitives, APF



Background - ROS Trajectory Planning

What Options Did We Have Before?

Navigation Stack
- Base Local Planner
- DWA Local Planner

Nav2
- DWB Controller
- Reg. Pure Pursuit Controller
- Rotation Shim

The Community
- TEB Local Planner
- Graceful

→ No Contemporary Approaches

Common Complaints

"The Nav Stack keeps running into walls."

"Why does my robot get stuck so much?"

"God, tuning this is such a pain!"

"Recoveries are constantly triggering"

"I wish I went to medical school instead…"

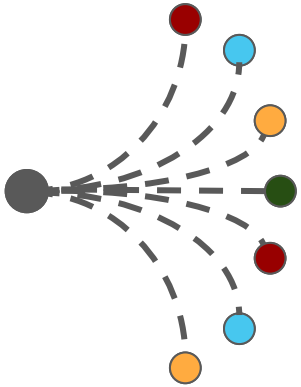
The root cause is *usually* the trajectory planner *

See a full comparison *"From the Desks of ROS Maintainers: A Survey of Modern & Capable Mobile Robotics Algorithms in the Robot Operating System 2"*

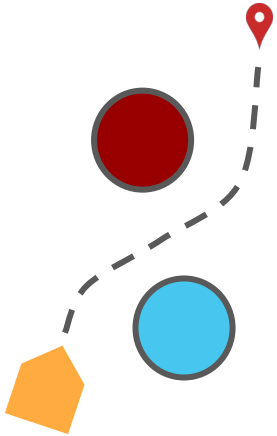
* Or a desire for a career change

Overview

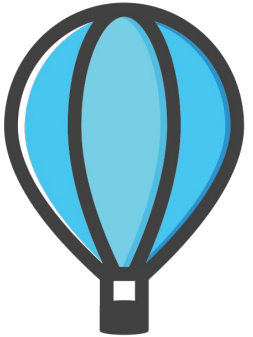
Background



MPPI Overview



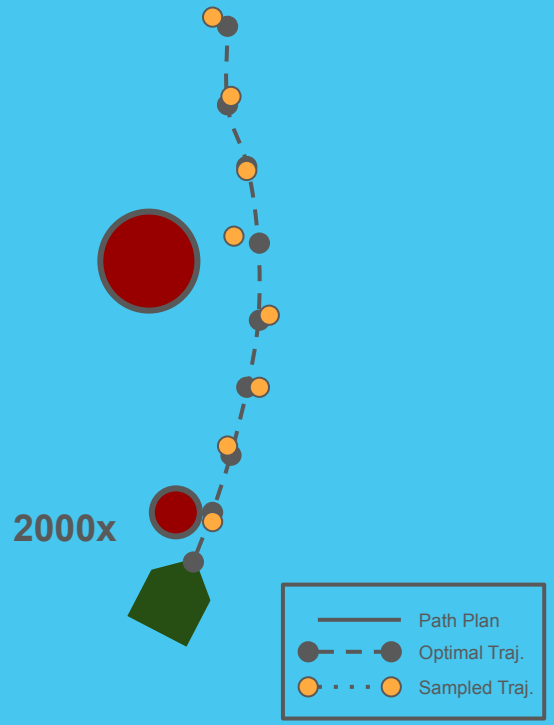
Nav2 MPPI Controller



MPPI Controller - Concept

Sampling-Based Model Predictive Control

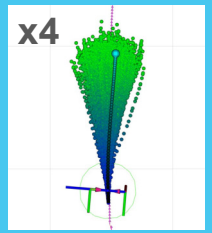
1. **Add Noises to Prior Optimal Trajectory's Controls**
2. Apply Dynamics to Controls & Rollout Trajectories
3. Score Noised Trajectories Via Objective Functions
4. Compute New Optimal Control Sequence
5. Execute First Control, Shift Optimal Control. Repeat.



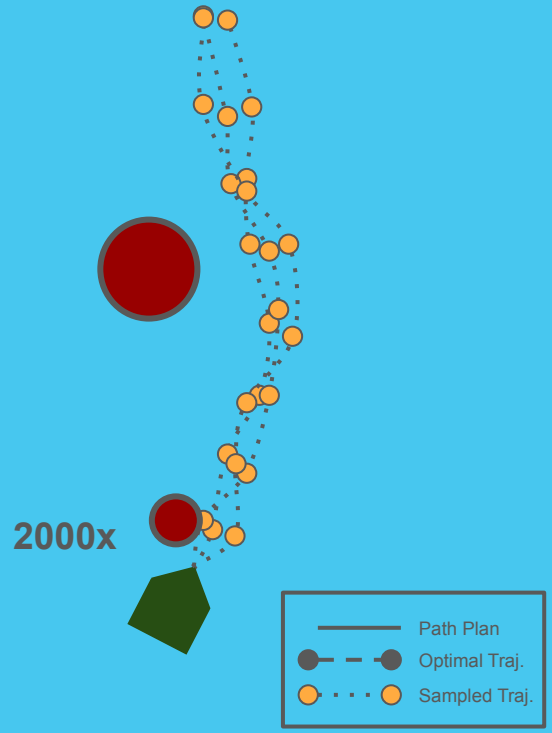
MPPI Controller - Concept

Sampling-Based Model Predictive Control

1. Add Noises to Prior Optimal Trajectory's Controls
2. **Apply Dynamics to Controls & Rollout Trajectories**
3. Score Noised Trajectories Via Objective Functions
4. Compute New Optimal Control Sequence
5. Execute First Control, Shift Optimal Control. Repeat.



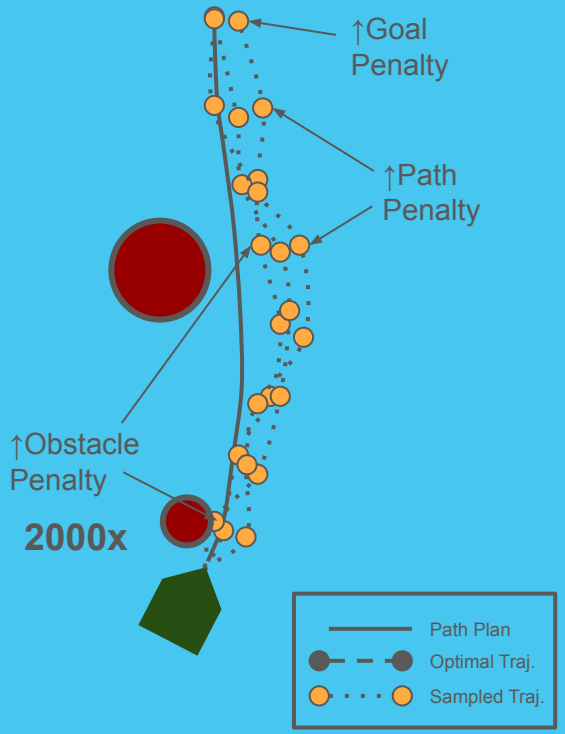
Looks like this!



MPPI Controller - Concept

Sampling-Based Model Predictive Control

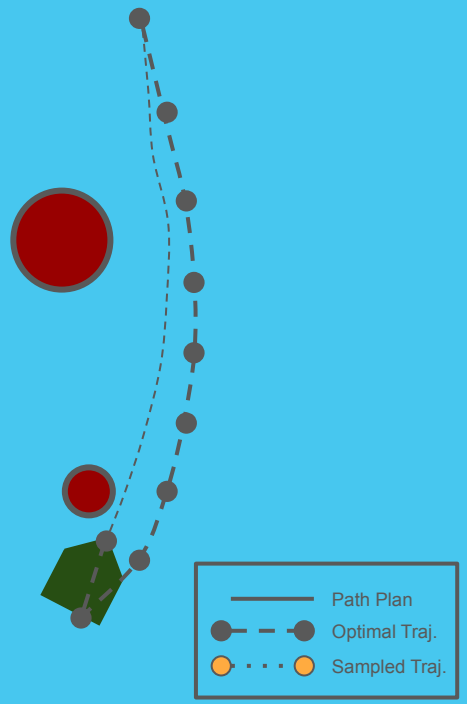
1. Add Noises to Prior Optimal Trajectory's Controls
2. Apply Dynamics to Controls & Rollout Trajectories
3. **Score Noised Trajectories Via Objective Functions**
4. Compute New Optimal Control Sequence
5. Execute First Control, Shift Optimal Control. Repeat.



MPPI Controller - Concept

Sampling-Based Model Predictive Control

1. Add Noises to Prior Optimal Trajectory's Controls
2. Apply Dynamics to Controls & Rollout Trajectories
3. Score Noised Trajectories Via Objective Functions
4. **Compute New Optimal Control Sequence**
5. Execute First Control, Shift Optimal Control. Repeat.



Find more information: *"Information Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving"*

MPPI Controller - Concept

Sampling-Based Model Predictive Control

1. Add Noises to Prior Optimal Trajectory's Controls

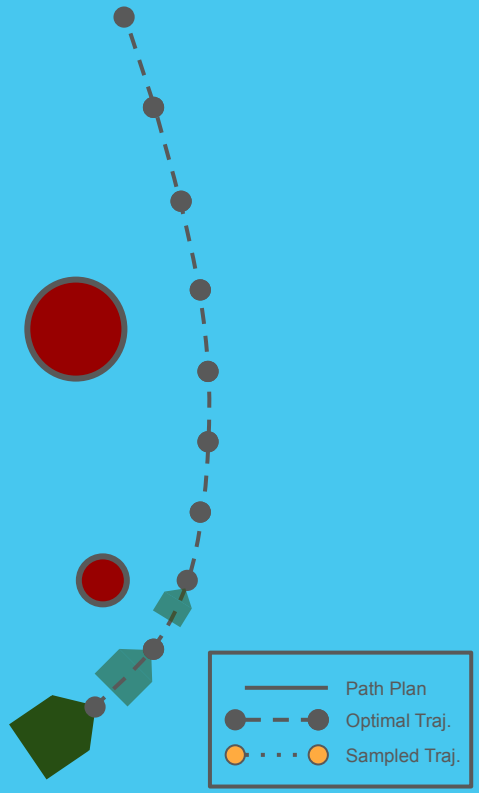
2. Apply Dynamics to Controls & Rollout Trajectories

3. Score Noised Trajectories Via Objective Functions

4. Compute New Optimal Control Sequence

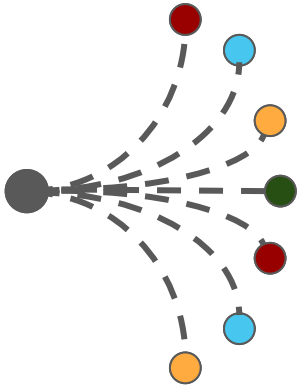
5. **Execute First Control, Shift Optimal Control. Repeat.**

→ No Derivatives or Required Objectives Convexity
→ Arbitrary Dynamics, Constraints, and Objectives

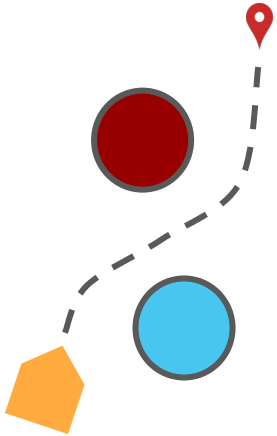


Overview

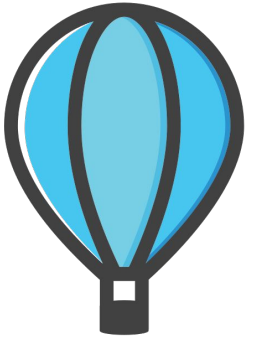
Background



MPPI Overview



Nav2 MPPI Controller



MPPI Controller - Overview

Model Predictive Path Integral Controller

- Cutting-Edge Predictive Controller Used in AV's
- Emergent Intelligent Behaviors - ***Far*** Fewer Recoveries
- Circular, Non-Circular, Legged, Ackermann, Diff & Omni
- Plugin-Based Critic Functions For Behavioral Expansion
- 97% Unit Test Coverage, Used in Production Today

MPPI Controller - More Details

Designed to be the Last I'll Ever Write - famous last words

Tuned for Great Out-of-the-Box Performance!

Superior in Tricky Situations, Highly Flexible

Accepts Feasible Paths, Respects Path Inversions

CPU-Only @ 30-50 Hz, No GPU Lock-In

Future Improvements + Innovations in Research

→ CPU speed, GPU support, smoothness, dynamics models, addtl critics, …

Major Thanks to Alexey Budyakov (@artofnothingness)

Original Prototype Author & Proactive Collaborator

MPPI Controller - Configuration

See README For Details & Guidance

10 Optional Critics

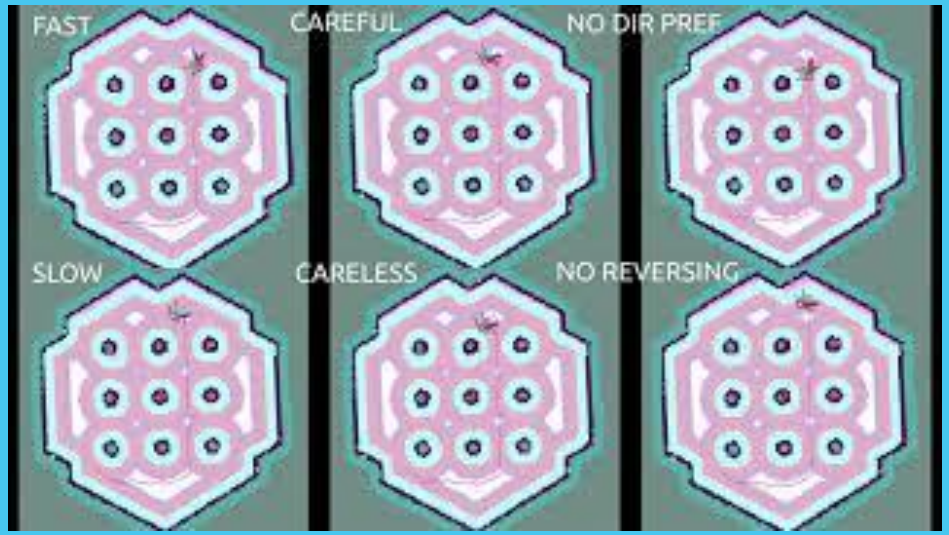
3 Motion Models

Batch Size - # of Sampled Trajectories
Time Steps - # of Time Steps in Trajectories
Model dt - Δt between Trajectory Time Steps

Tuned: 30Hz @ 2000, 50Hz @ 1000

Costmap Smooth Inflation Critical! (like Smac)

Sampling STD {Vx, Vy, Wz} - Sampling Noise



In The Wild

Our Sponsors

DEXORY

POLYMATH ROBOTICS

STEREOLABS

CONFIDENTIAL

Powered By

OPEN NAVIGATION

Repository, Documentation, and Issue Tracker:

https://github.com/ros-planning/navigation2

https://discourse.ros.org/c/navigation

https://navigation.ros.org

Interested in Getting Involved or Sponsoring?
Join our Slack: https://bit.ly/3ssxidP
Find out more at: https://opennav.org

Steve Macenski
Open Navigation LLC
info@opennav.org

P.S. Ask for rad stickers 

Our Sponsors

DEXORY

POLYMATH ROBOTICS

STEREOLABS

CONFIDENTIAL