Motion Planning Lecture 8

Introduction to Open Motion Planning Library

Wolfgang Hönig (TU Berlin) and Andreas Orthey (Realtime Robotics)

June 12, 2024

Recap Last Week

Last Week

- Optimal kinodynamic planning
- Planning without Steering (SST)
- Space-cost approach (AO-RRT)

Today

- Introduction to open motion planning library (OMPL)
- General structure of OMPL
- How to setup problems in OMPL

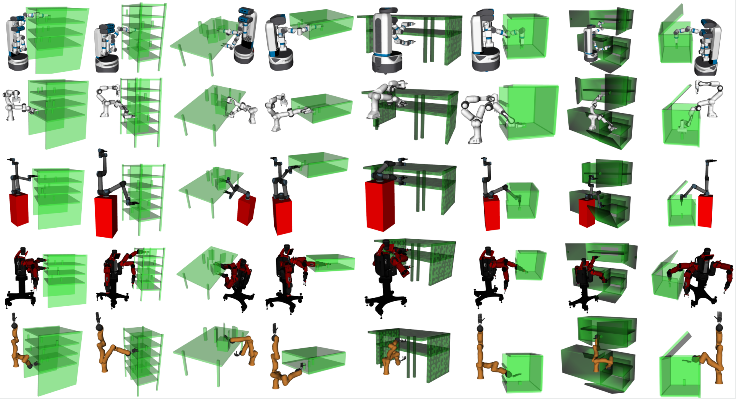
Motivation

OMPL

- Open-source software
- Collection of sampling-based planners
- Written and maintained by researchers in motion planning
- Modular, easily extendable
- Used by several robotics companies

Motivation

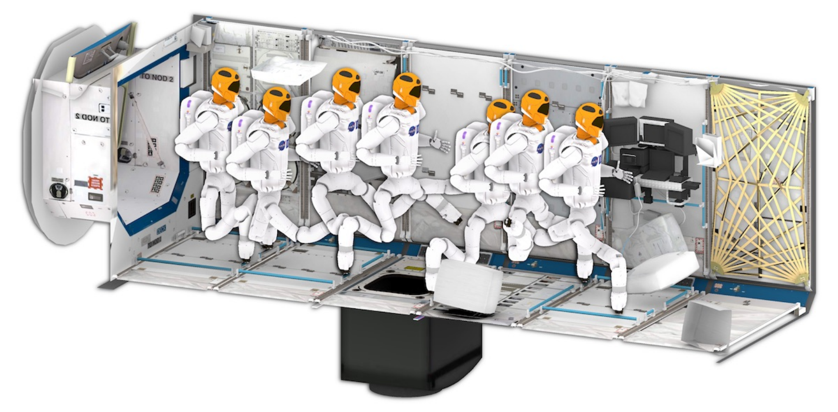
OMPL



Chamzas et al., "MotionBenchMaker: A Tool to Generate and Benchmark Motion Planning Datasets", (2022)

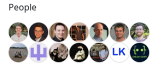
Motivation

OMPL



Kingston et al., "Exploring Implicit Spaces for Constrained Sampling-Based Planning", (2019)

Open motion planning library



History

- Created by Lydia Kavraki, Professor at Rice University, Houston, TX
- Maintained by Mark Moll, Previous Director of Research at PickNik Robotics
- Contributions by many researchers worldwide

Sucan et al., "The Open Motion Planning Library", (2012)

Open motion planning library

Version History

- Version 1.6 (Jan 7, 2023)
- Version 1.4 (Jun 25, 2018)
- Version 1.2 (Jun 19, 2016)
- Version 1.0 (Oct 25, 2014)

Open motion planning library

Capabilities

- Solving geometrical problems
- Solving optimal with arbitrary objectives
- Solving kinodynamic problems
- Benchmarking planners

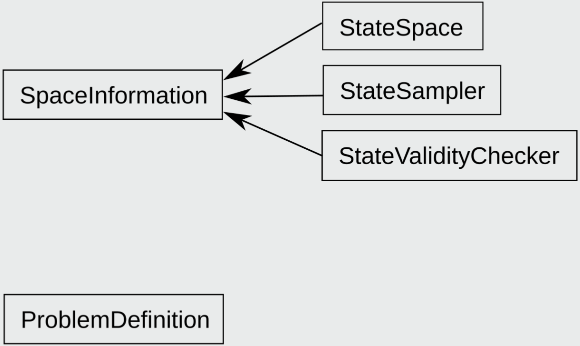
Overview OMPL

Open motion planning library

OMPL Structure

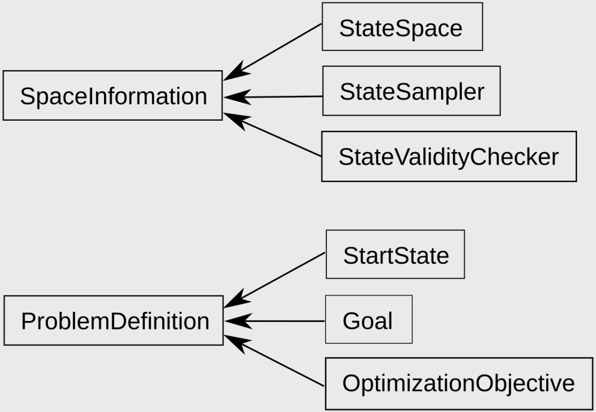
Open motion planning library

OMPL Structure



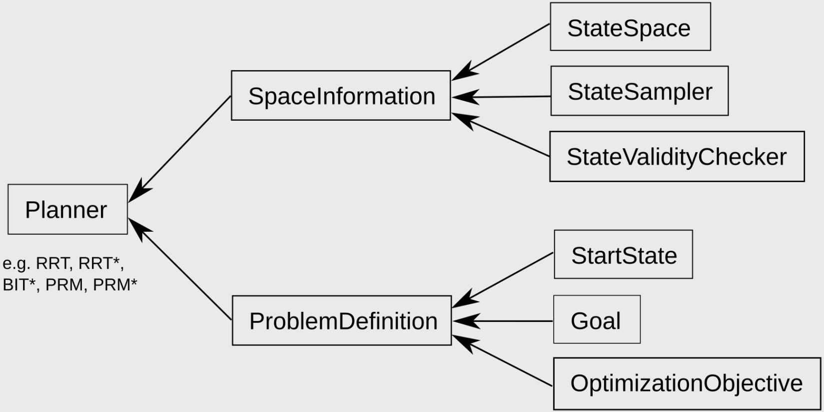
Open motion planning library

OMPL Structure



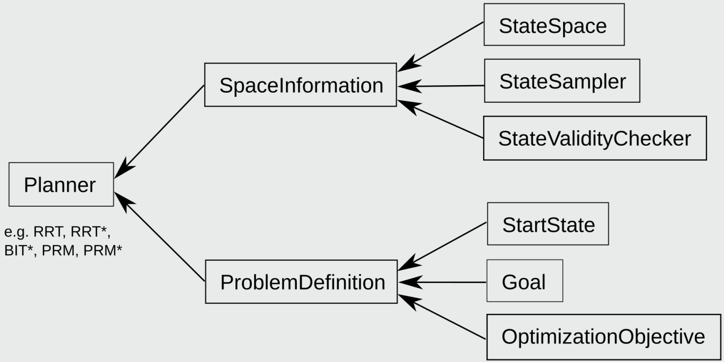
Open motion planning library

OMPL Structure



Open motion planning library

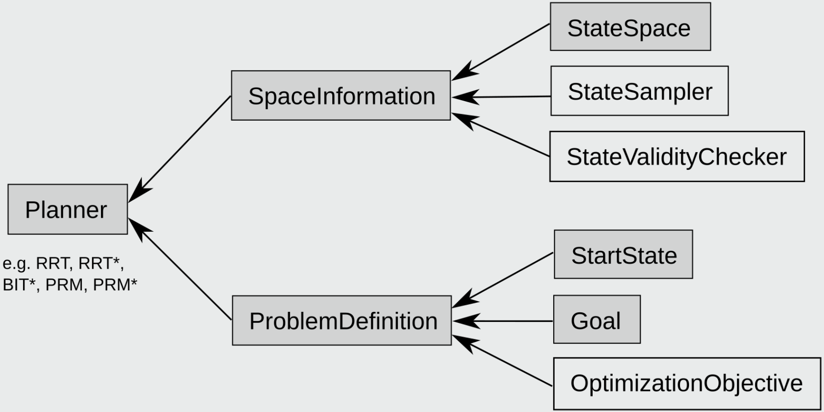
OMPL Structure



What are the necessary structures?

Open motion planning library

OMPL Structure



Overview OMPL

State Space

Open motion planning library

State Space

- RealVectorStateSpace R^n (Manipulator arm)
- SE2StateSpace $SE(2)$ (Mobile base, roomba)
- SE3StateSpace $SE(3)$ (Drone, rigid body)
- DubinsStateSpace (Car with constant forward velocity)
- TimeStateSpace (Only go forward in time)

Primitives

- Distance metric
- Uniform sampling
- Interpolation

Requirements

- Bounds
- Dimensionality

Overview OMPL

StateValidityChecker

Open motion planning library

StateValidityChecker

- Interface with collision checking libraries
- Custom code
- Flexible Collision Library (FCL) `https://github.com/flexible-collision-library/fcl`
- Proximity Queries Package (PQP) `https://github.com/GammaUNC/PQP`

Primitives

- `bool IsValid(x)`: Check that all constraints are fulfilled
- `float clearance(x)`: Return distance to nearest invalid state [Optional]

Overview OMPL

StateSampler

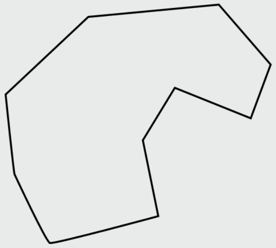
Open motion planning library

StateSampler

- Sample states from the state space
- Unbiased vs Biased
- Default: Unbiased in StateSpace
- Biased: Obstacle, Clearance, Deterministic

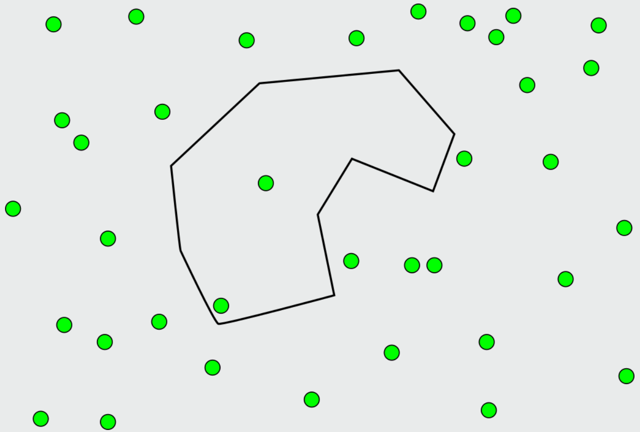
Open motion planning library

Obstacle-based sampling



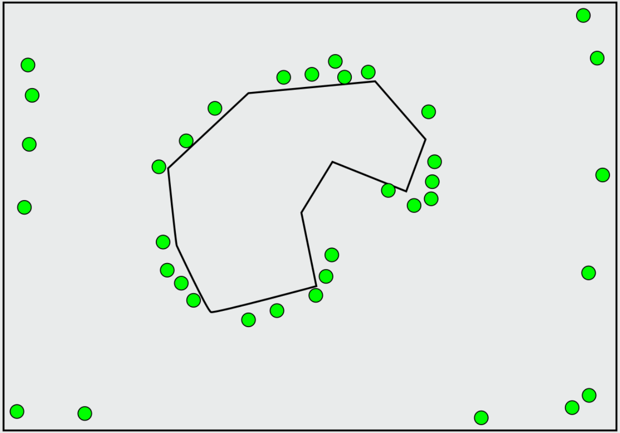
Open motion planning library

Obstacle-based sampling



Open motion planning library

Obstacle-based sampling



Open motion planning library

Question

What are the advantages of using obstacle-based sampling?

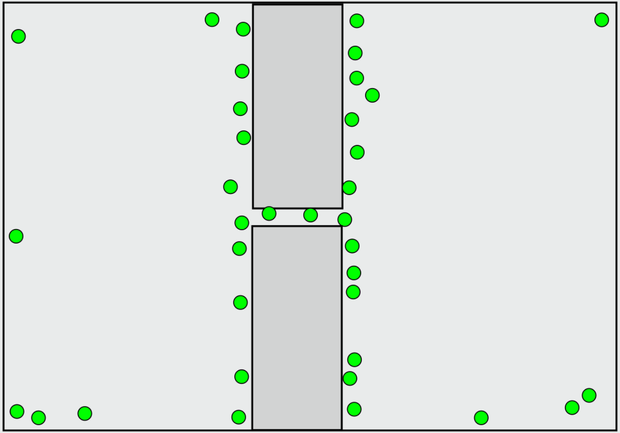
Open motion planning library

Obstacle-based sampling

- Narrow passages
- Path length bias

Open motion planning library

Obstacle-based sampling



Open motion planning library

Obstacle-based sampling

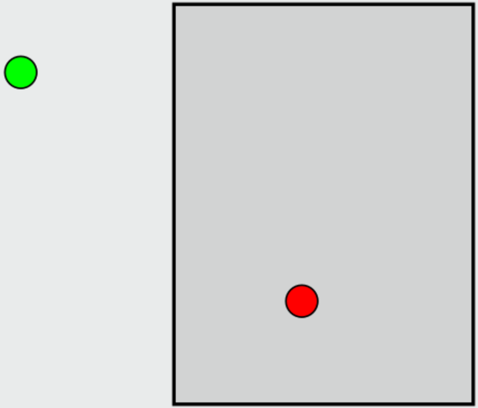
- ObstacleBasedValidStateSampler
- GaussianValidStateSampler
- BridgeTestValidStateSampler

Open motion planning library

ObstacleBasedValidStateSampler

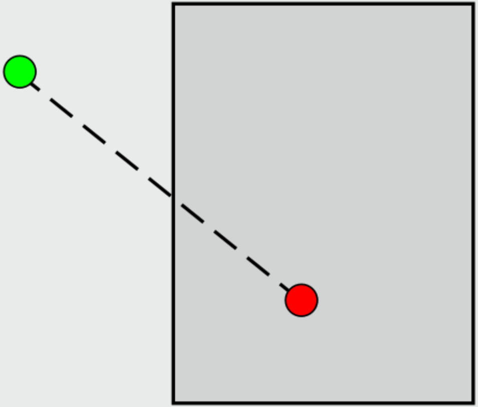
Open motion planning library

ObstacleBasedValidStateSampler



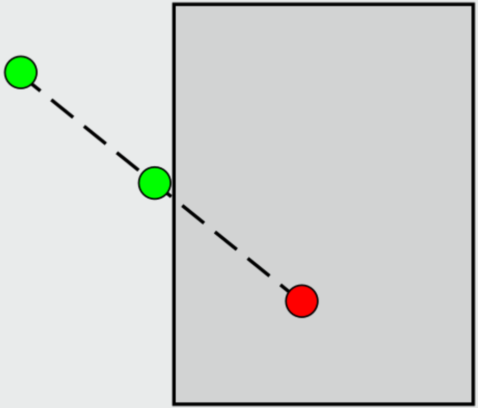
Open motion planning library

ObstacleBasedValidStateSampler



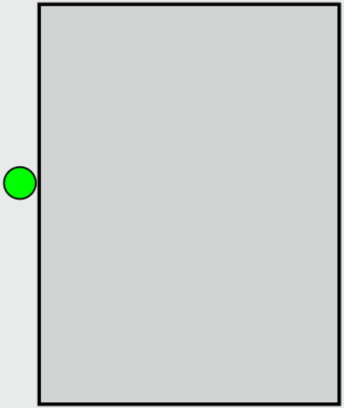
Open motion planning library

ObstacleBasedValidStateSampler

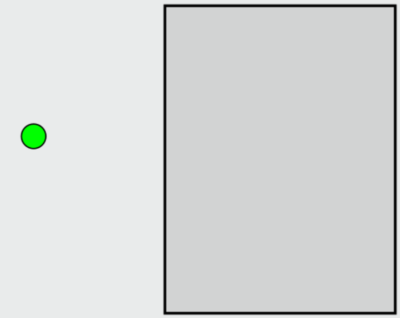


Open motion planning library

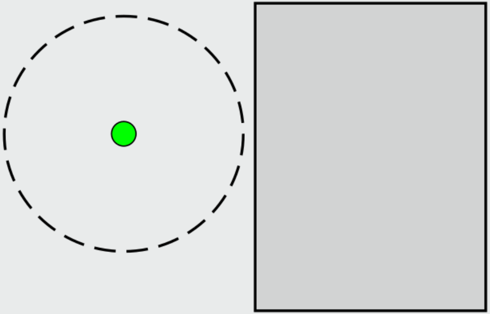
ObstacleBasedValidStateSampler



GaussianValidStateSampler

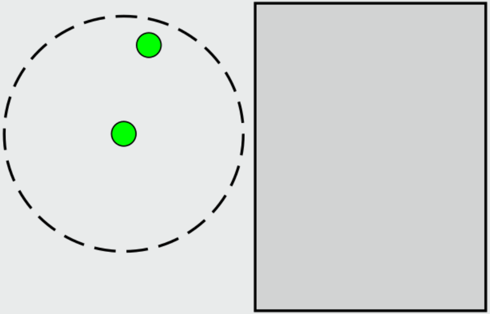


GaussianValidStateSampler



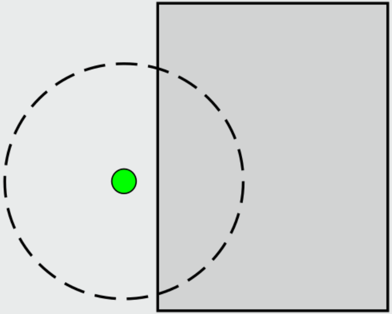
Open motion planning library

GaussianValidStateSampler



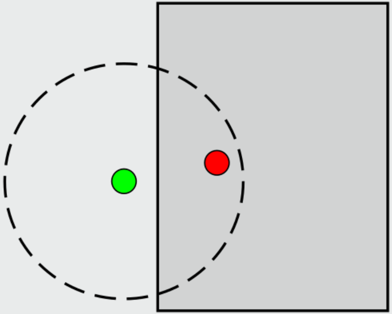
Open motion planning library

GaussianValidStateSampler



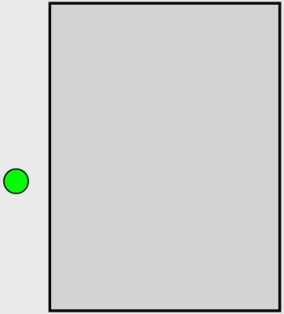
Open motion planning library

GaussianValidStateSampler



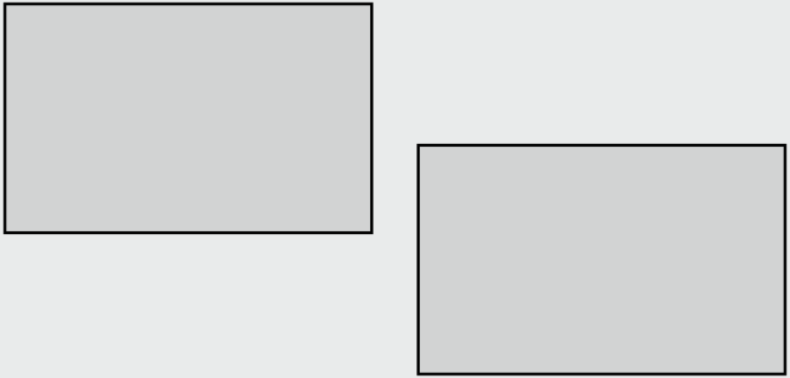
Open motion planning library

GaussianValidStateSampler



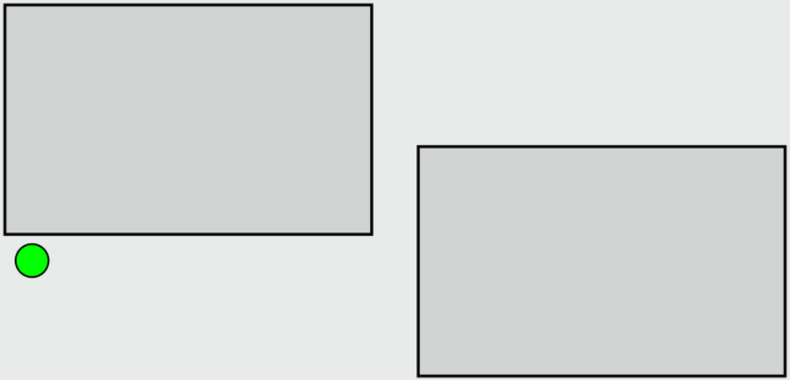
Open motion planning library

BridgeTestValidStateSampler



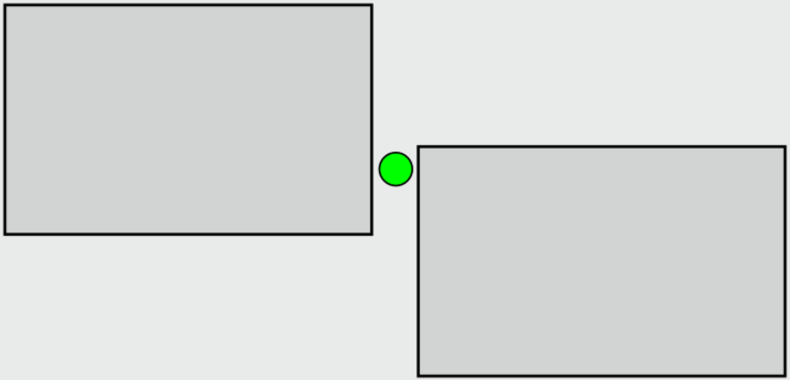
Open motion planning library

BridgeTestValidStateSampler



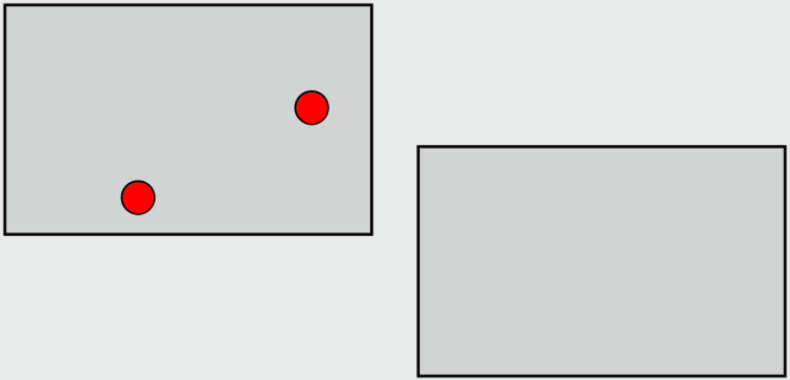
Open motion planning library

BridgeTestValidStateSampler



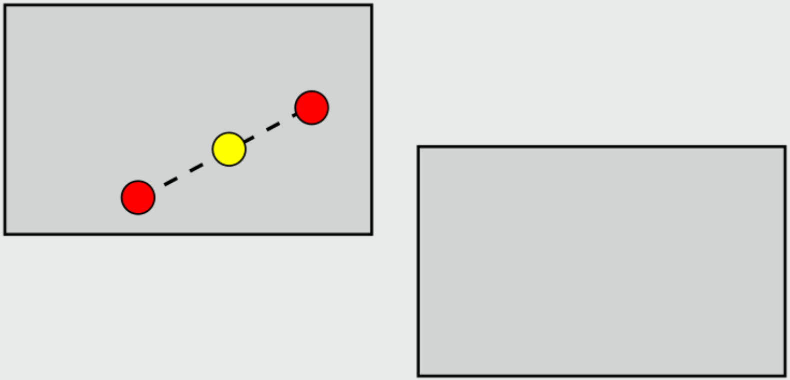
Open motion planning library

BridgeTestValidStateSampler



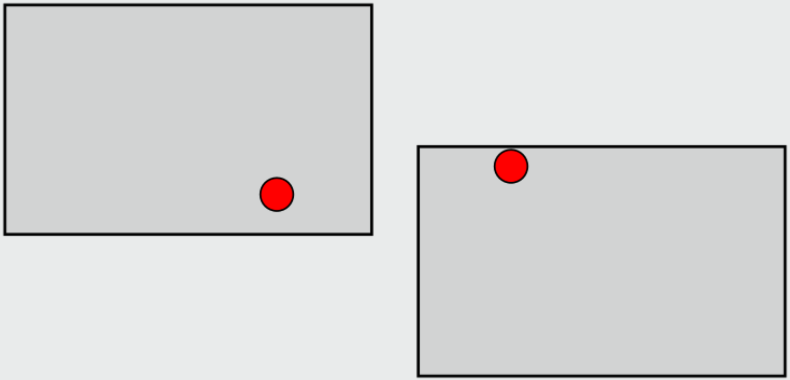
Open motion planning library

BridgeTestValidStateSampler



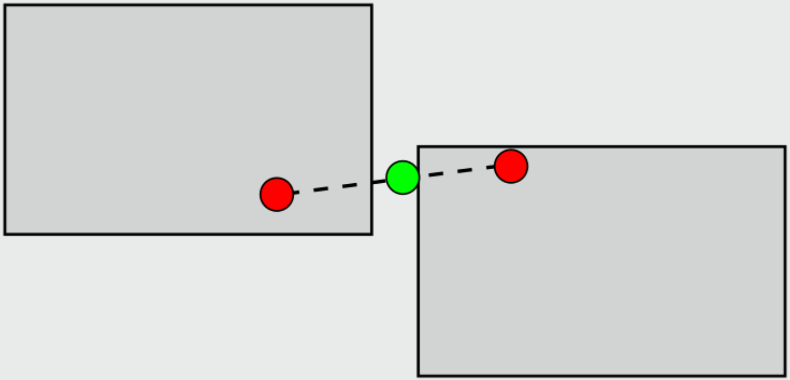
Open motion planning library

BridgeTestValidStateSampler



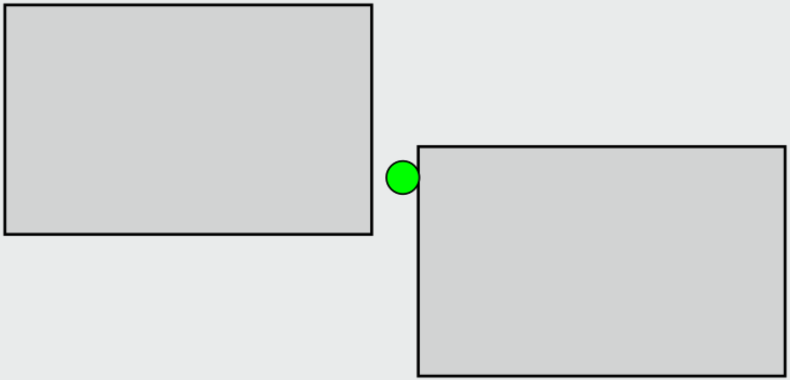
Open motion planning library

BridgeTestValidStateSampler



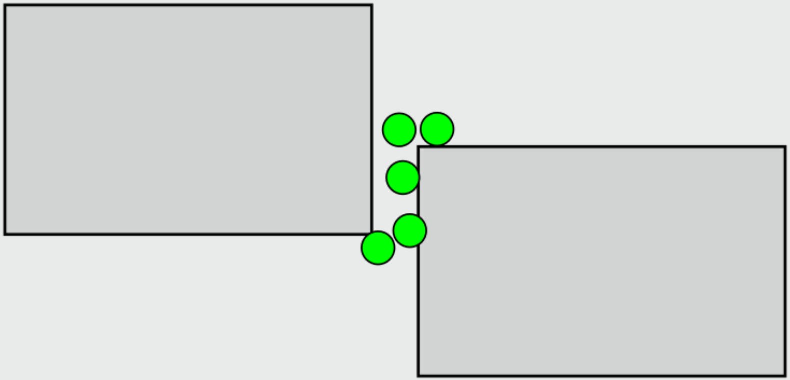
Open motion planning library

BridgeTestValidStateSampler



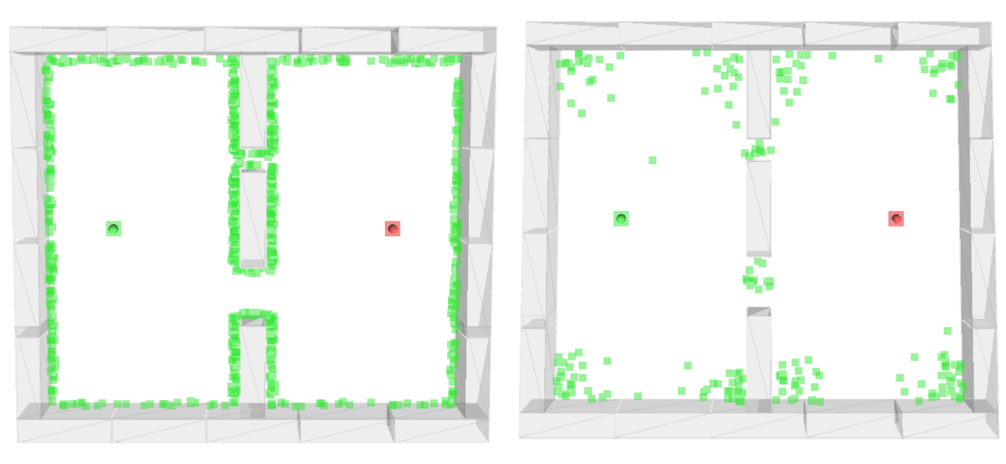
Open motion planning library

BridgeTestValidStateSampler



Open motion planning library

BridgeTestValidStateSampler



Open motion planning library

Question

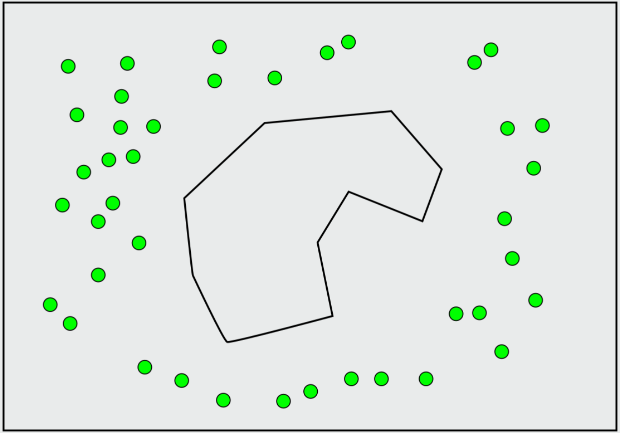
What are the disadvantages of using obstacle-based sampling?

Obstacle-based sampling

- Trade-off quality vs. runtime
- Path length

Open motion planning library

Clearance-based sampling



Open motion planning library

Question

What are the advantages of using clearance-based sampling?

Open motion planning library

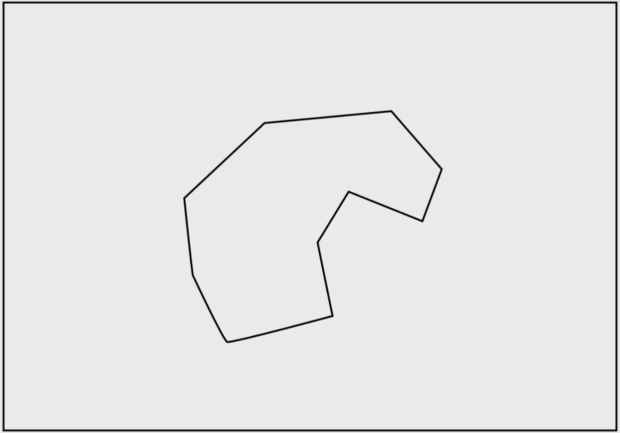
Clearance-based sampling

- Execution uncertainty
- Clearance as safety fence

Types of clearance-based sampling

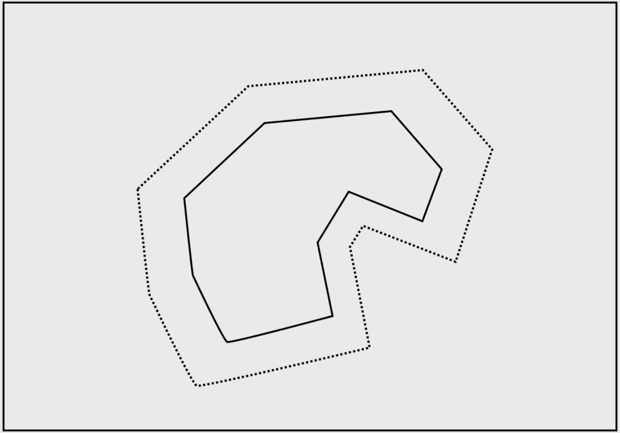
- Minimum clearance
- Maximize clearance

MinimumClearanceValidStateSampler



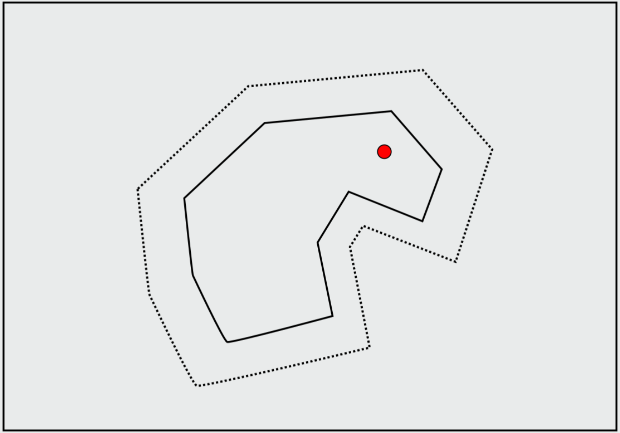
Open motion planning library

MinimumClearanceValidStateSampler



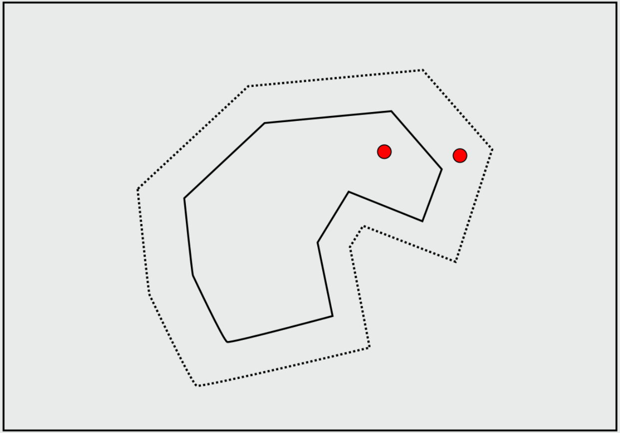
Open motion planning library

MinimumClearanceValidStateSampler



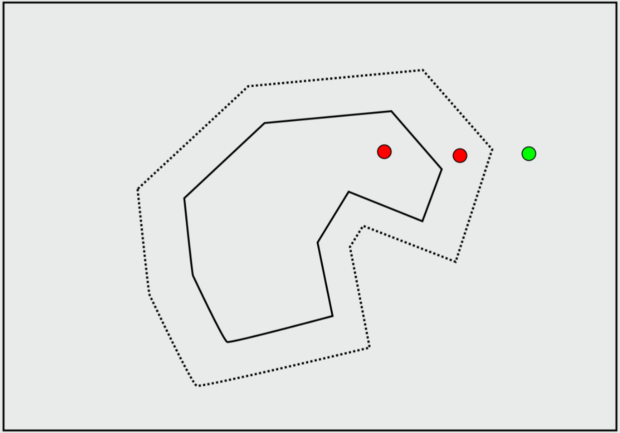
Open motion planning library

MinimumClearanceValidStateSampler

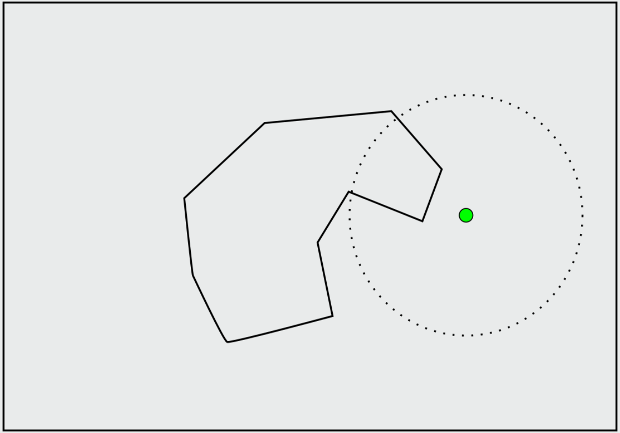


Open motion planning library

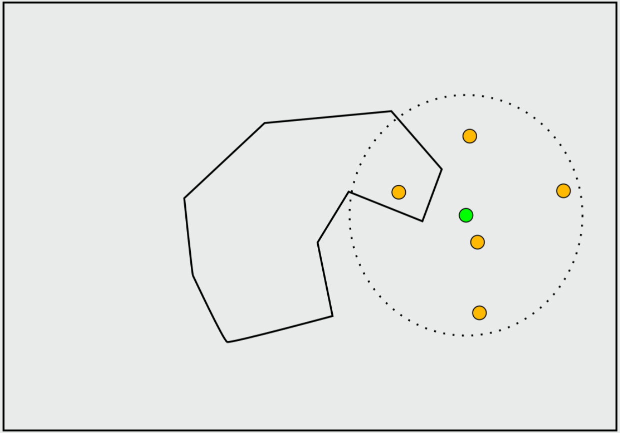
MinimumClearanceValidStateSampler



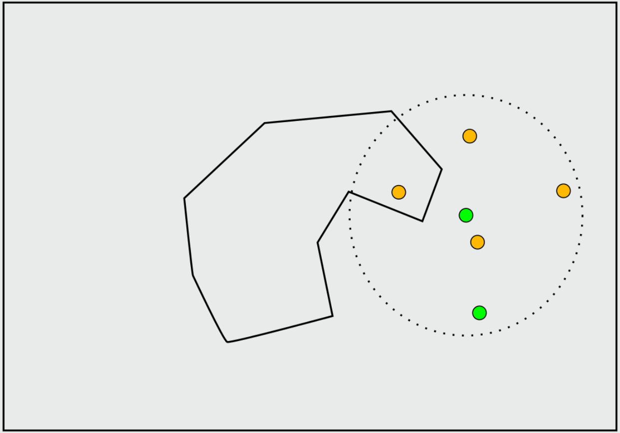
MaximizeClearanceValidStateSampler



MaximizeClearanceValidStateSampler

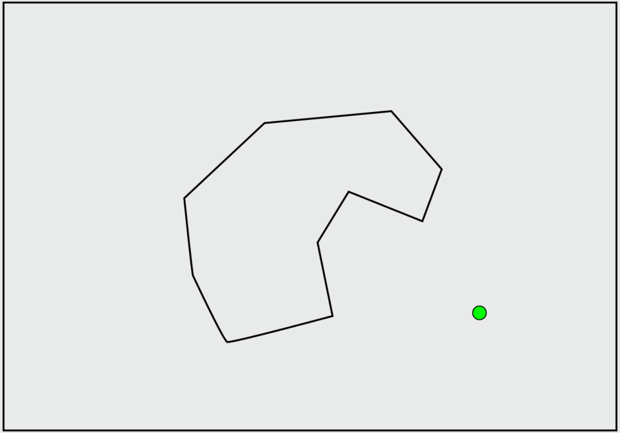


MaximizeClearanceValidStateSampler



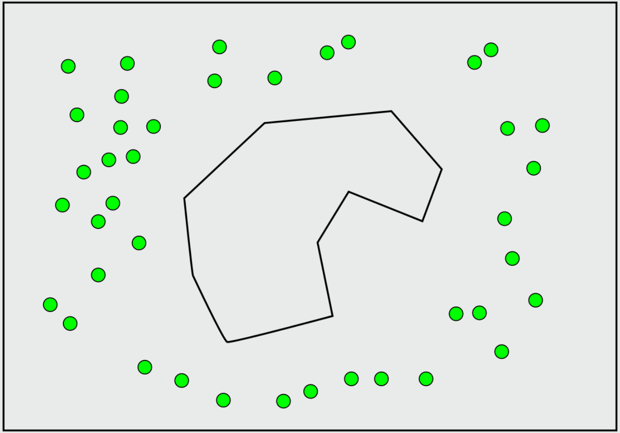
Open motion planning library

MaximizeClearanceValidStateSampler



Open motion planning library

MaximizeClearanceValidStateSampler



Open motion planning library

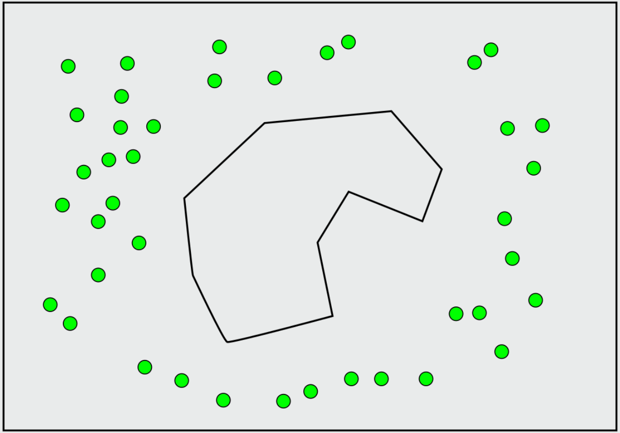
Question

What are the disadvantages of using clearance-based sampling?

Disadvantages clearance-based sampling

- Narrow passages
- Clearance cost

Deterministic sampling



Question

What are the advantages of using deterministic sampling?

Open motion planning library

Advantages deterministic sampling

- Predictability
- Better distributed
- Low-discrepancy

Open motion planning library

Halton sampling

Open motion planning library

Halton sampling

Open motion planning library

Halton sampling

Open motion planning library

Halton sampling

Open motion planning library

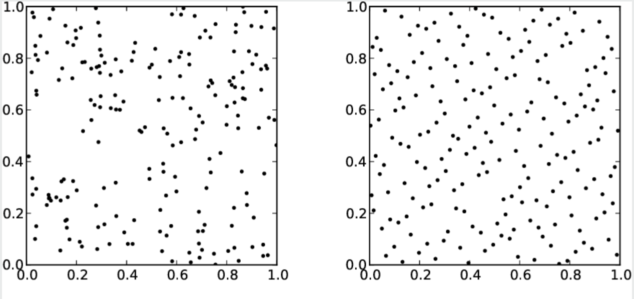
Halton sampling

Open motion planning library

Halton sampling

Open motion planning library

Halton sampling



Open motion planning library

Question

What are the disadvantages of using deterministic sampling?

Overview OMPL

Goal

Open motion planning library

Goal

- GoalState: State plus an ϵ neighborhood
- GoalStates: Multiple states (e.g. grasping)
- GoalRegions: Subspace of state space (manipulator arm on mobile base)

Primitives

- `bool isSatisfied(x)`: Check if a state satisfies the goal constraints

Overview OMPL

OptimizationObjective

Open motion planning library

OptimizationObjective

- PathLength
- MaximizeMinClearance
- MinimizeArrivalTime

Primitives

- `float motionCost(x, y)`: Compute the cost to go from x to y.

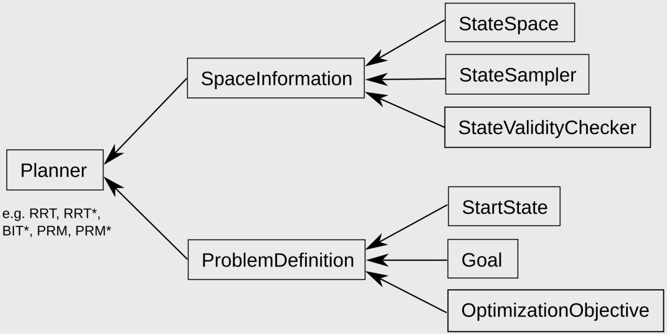
Overview OMPL

Coding Demo VL8-demo1.py

Kinodynamic Planning in OMPL

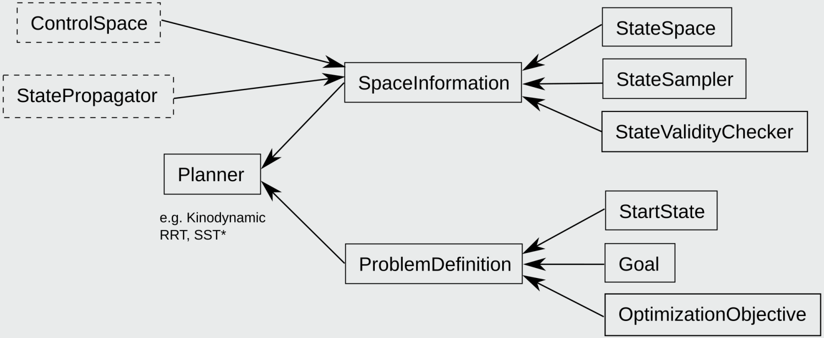
Open motion planning library

Kinodynamic planning



Open motion planning library

Kinodynamic planning



Kinodynamic Planning in OMPL

Control Space

Open motion planning library

Control Space

- RealVectorControlSpace: R^n plus bounds
- DiscreteControlSpace: Predefined set of controls (motion primitives)

Methods

- Uniform sampling

Requirements

- Bounds
- Dimensionality
- Set of controls

Kinodynamic Planning in OMPL

State propagator

State propagator

- `State propagate(x, c, t)`: Start at x, and propagate system forward with control c for duration t.
- `Control steer(x, y)`: Compute control c and duration t to move state from x to y [Optional]

Kinodynamic Planning in OMPL

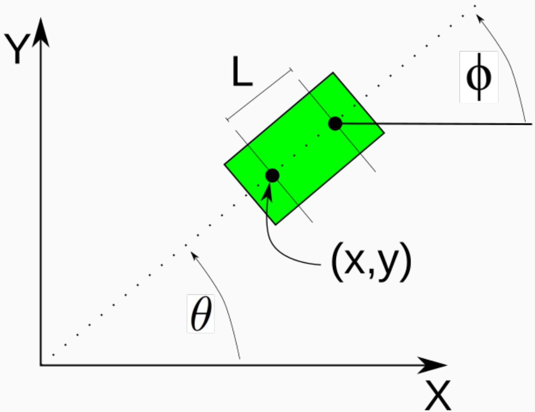
Kinematic Car

Open motion planning library

Kinematic car

- State space $SE(2) = (x, y, \theta)$
- Control space $U = [-1, +1] \times (\phi_{\text{min}}, \phi_{\text{max}}) = (u_1, u_2)$
- Dynamics $\dot{x} = f(x, y, \theta, u_1, u_2)$

Open motion planning library



Kinematic car

$$\dot{x} = \begin{bmatrix} u_1 \cos(\theta) \\ u_2 \sin(\theta) \\ \frac{\tan(u_2)}{L} u_1 \end{bmatrix}$$

Summary

OMPL

- Overview of OMPL
- Class structures, options
- Setting up kinodynamic problems
- Coding examples

Open motion planning library

Links

- OMPL `https://ompl.kavrakilab.org/`: Main website
- OMPL on github `https://github.com/ompl/ompl`: Main repository on github
- Planner Arena `https://plannerarena.org/`: Can open database (db) files to display performance of planners
- Webapp `http://omplapp.kavrakilab.org/`: Run planners on a set of classical scenarios

References i

[1] Zachary Kingston, Mark Moll, and Lydia E. Kavraki. "Exploring Implicit Spaces for Constrained Sampling-Based Planning". In: *Intl. J. of Robotics Research* 38.10–11 (Sept. 2019), pp. 1151–1178. DOI: 10.1177/0278364919868530.

[2] Mark Moll, Ioan A. Şucan, and Lydia E. Kavraki. "Benchmarking Motion Planning Algorithms: An Extensible Infrastructure for Analysis and Visualization". In: *IEEE Robotics & Automation Magazine* 22.3 (Sept. 2015), pp. 96–102. DOI: 10.1109/MRA.2015.2448276.

[3] Ioan A. Şucan, Mark Moll, and Lydia E. Kavraki. "The Open Motion Planning Library". In: *IEEE Robotics & Automation Magazine* 19.4 (Dec. 2012). https://ompl.kavrakilab.org, pp. 72–82. DOI: 10.1109/MRA.2012.2205651.

Kinodynamic Planning in OMPL

Coding Demo VL8-demo2.py