Spring 2016 :: 16662 :: Robot Autonomy :: Team 7

Motion Planning for Autonomous All-Terrain Vehicle

Guan-Horng Liu, Samuel Wang, Shu-Kai Lin, Chris Wang, Tiffany May
Advisor : Mr. George Kantor

OUTLINE

- **Platform Introduction**
- **Motivation & Challenges**
- **Problem Formulation**
- **OMPL Framework**
- **ROS Planning PipeLine**
  - Collision Check Module
  - RRT-Based Planner Module
- **Final Demo**

Platform Introduction

Project Overview

General purpose autonomous technology development for off-road driving in wilderness environment.

Vehicle Platform

2016 YAMAHA Viking VI side-by-side ATV

On-board Sensors

Novatel, Velodyne 64, Multisense S21

Software Modules

Classification, Pose Estimation,
Global Planner, Local Planner

Parter

Field Robotics Center
YAMAHA Motor Company, Japan
YAMAHA Motor Corporation, USA

Motivation & Challenges

Motivation

Propose a new local planner for off-road navigation with

- Static obstacles avoidance
- High-speed maneuvering in complex vehicle dynamic

Kinodynamic Planning in Control Space

Challenges

Uncertainty in Vehicle Dynamic Response Modeling
e.g. Wheel-terrain interaction...

Real-Time Implementation
e.g. Anytime planning, Computational efficiency

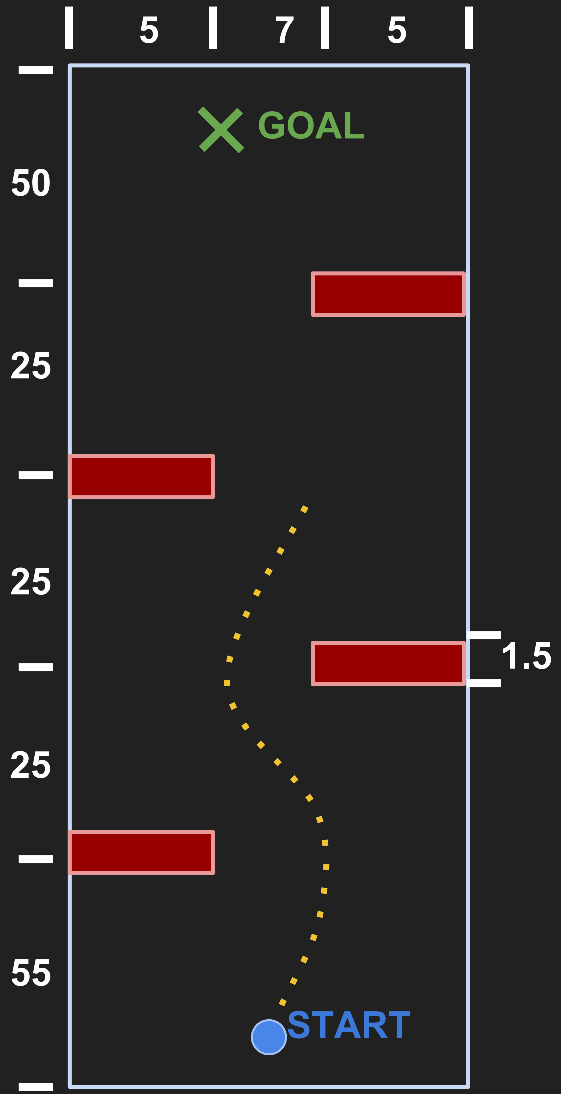
Problem Formulation

Testing Scenario Design

S-shape maneuvering with static obstacles avoidance
Vehicle velocity with at least 20kph
Model-based planner

Available Module/Sensor

YAMAHA Velocity Controller (YVCA)
LiDAR (point cloud), GPS/INS (position & velocity)



Show Time First

20kph

OUTLINE

Platform Introduction

Motivation & Challenges

Problem Formulation

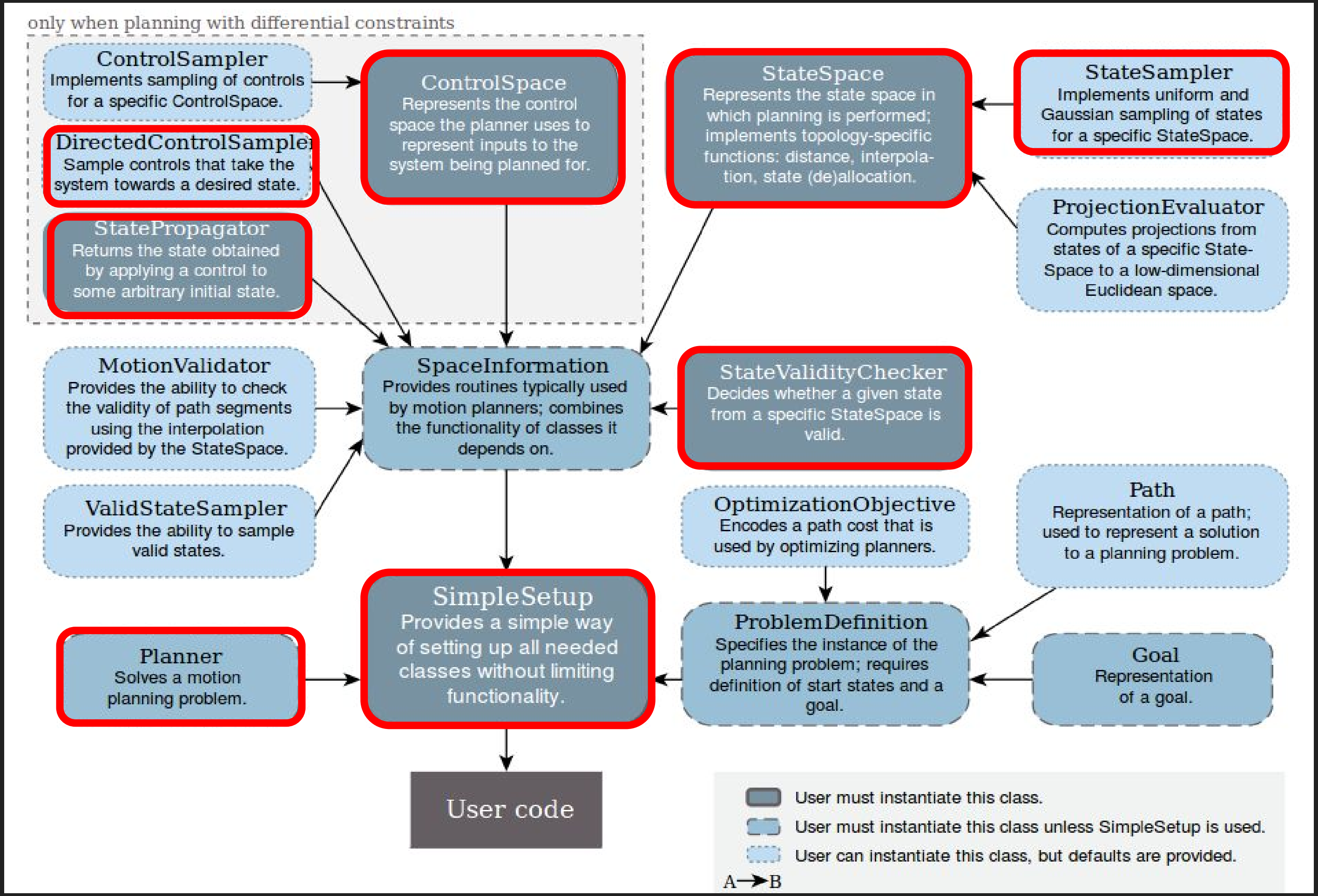
OMPL Framework

ROS Planning PipeLine

- Collision Check Module
- RRT-Based Planner Module

Final Demo

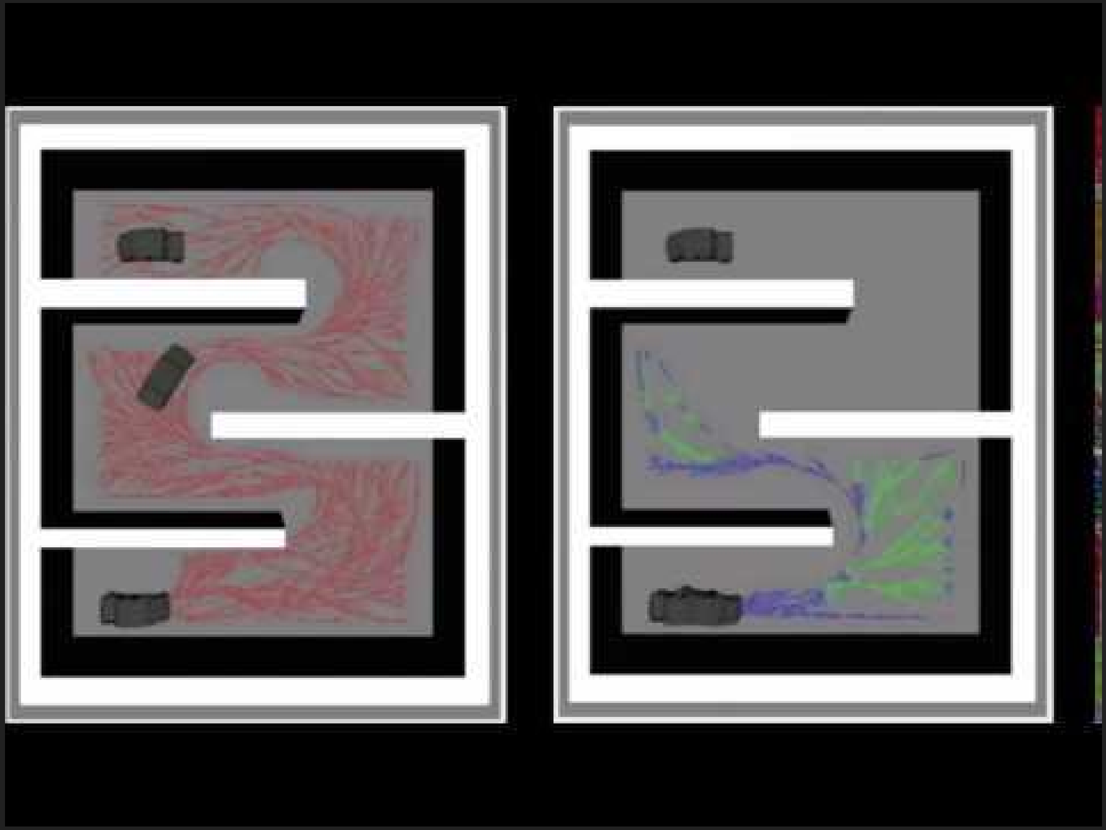
OMPL :: Framework



OMPL :: OMPL.app

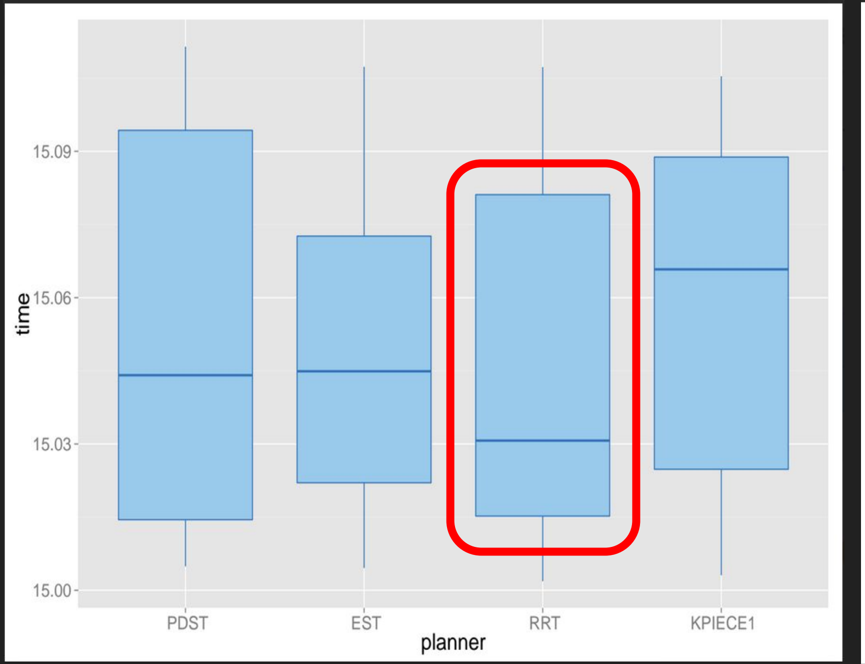
RRT Planner

KPIECE Planner

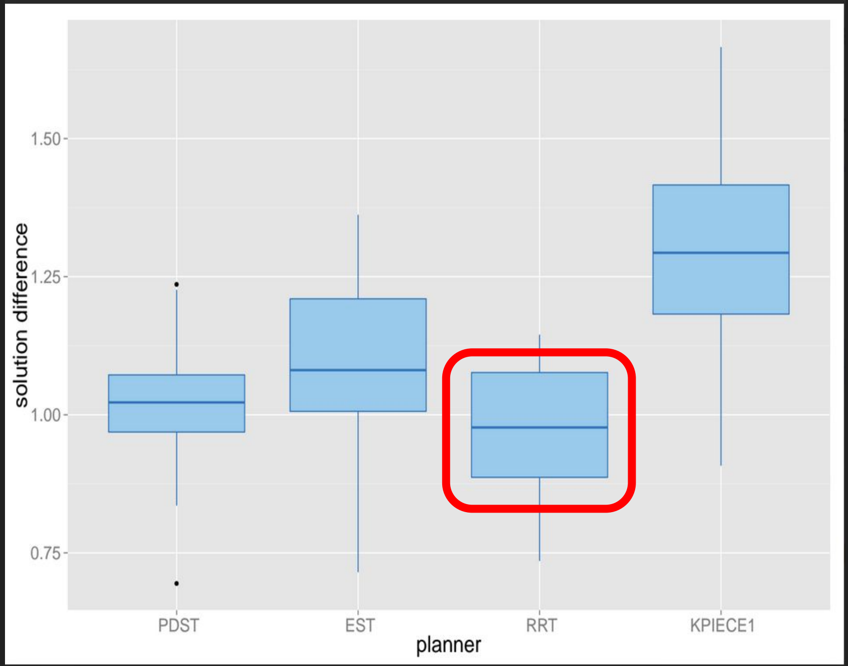


OMPL :: Benchmark

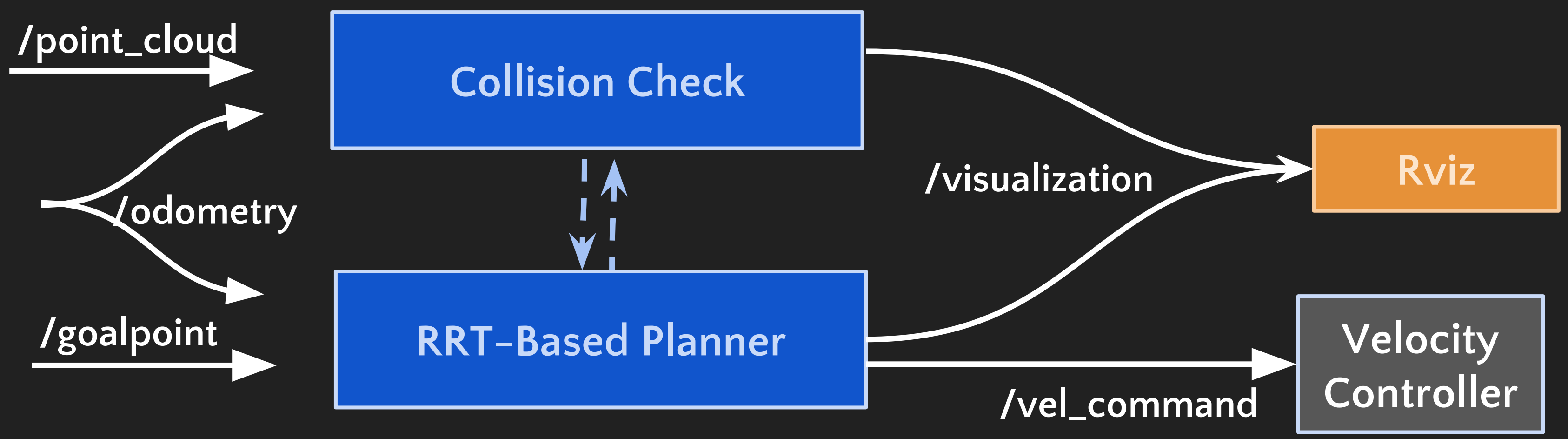
Solving Time among multi planners



Path Difference among multi planners



ROS Planning PipeLine



Collision Check Module

Simplified Occupancy Grid

Mesh

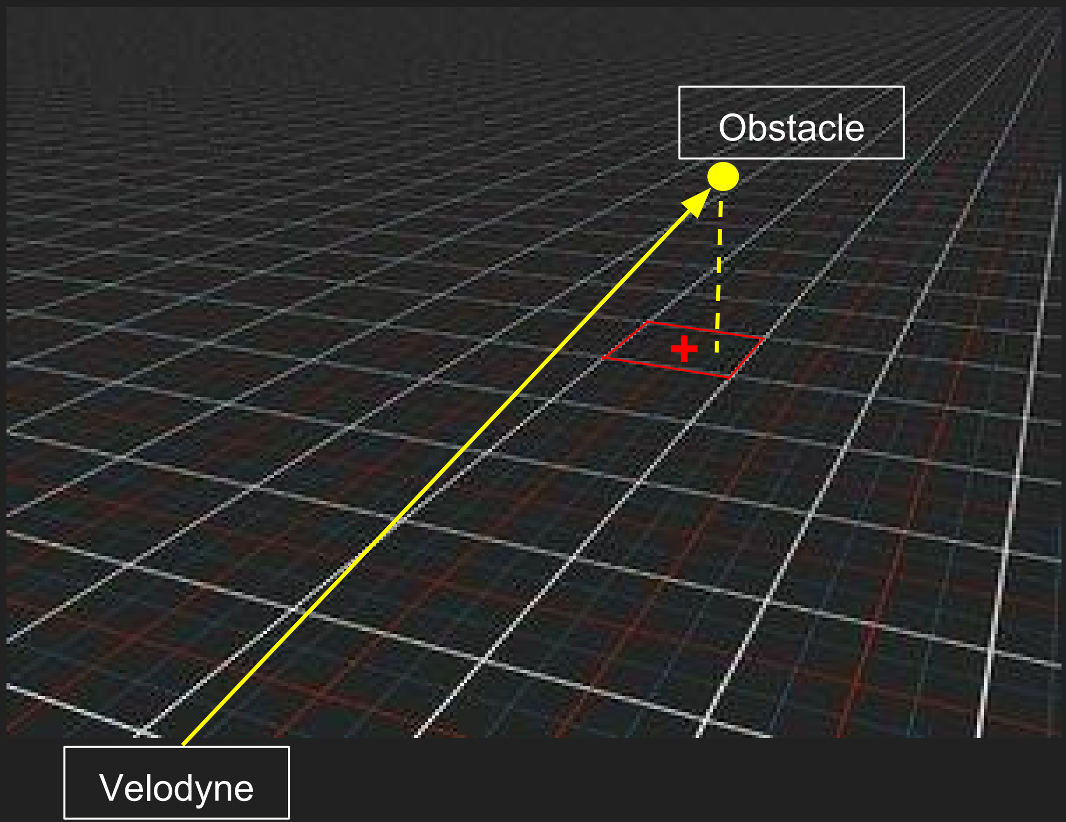
RANSAC Segmentation

Height Map Algorithm

Implemented Approach

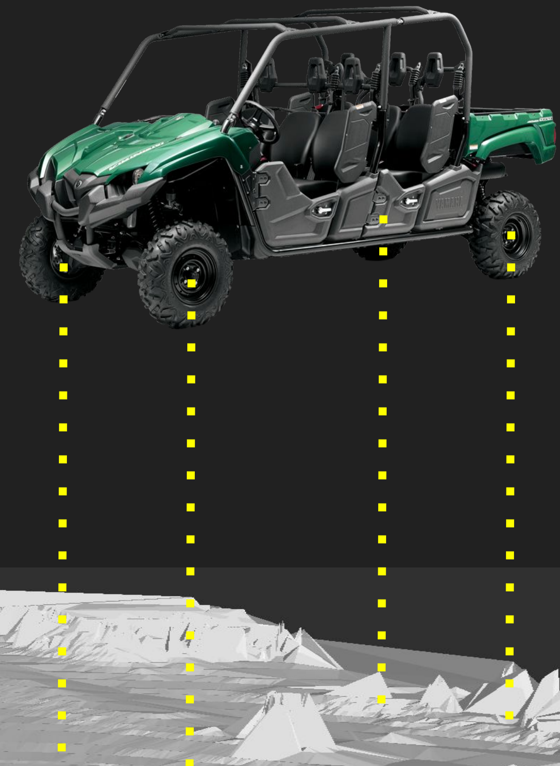
Collision Check :: Simplified Occupancy Grid

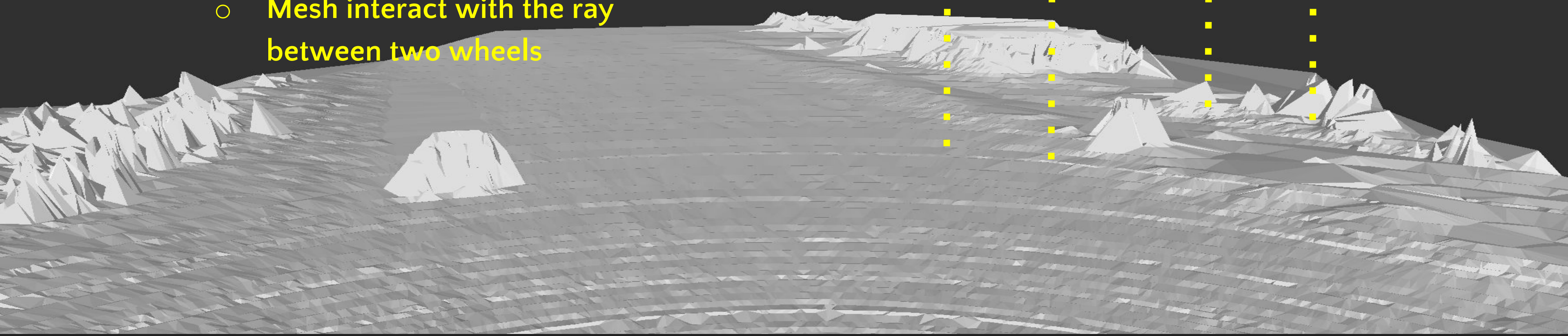
- Count only increases
- If (count > threshold)
  → obstacle



Collision Check :: Mesh

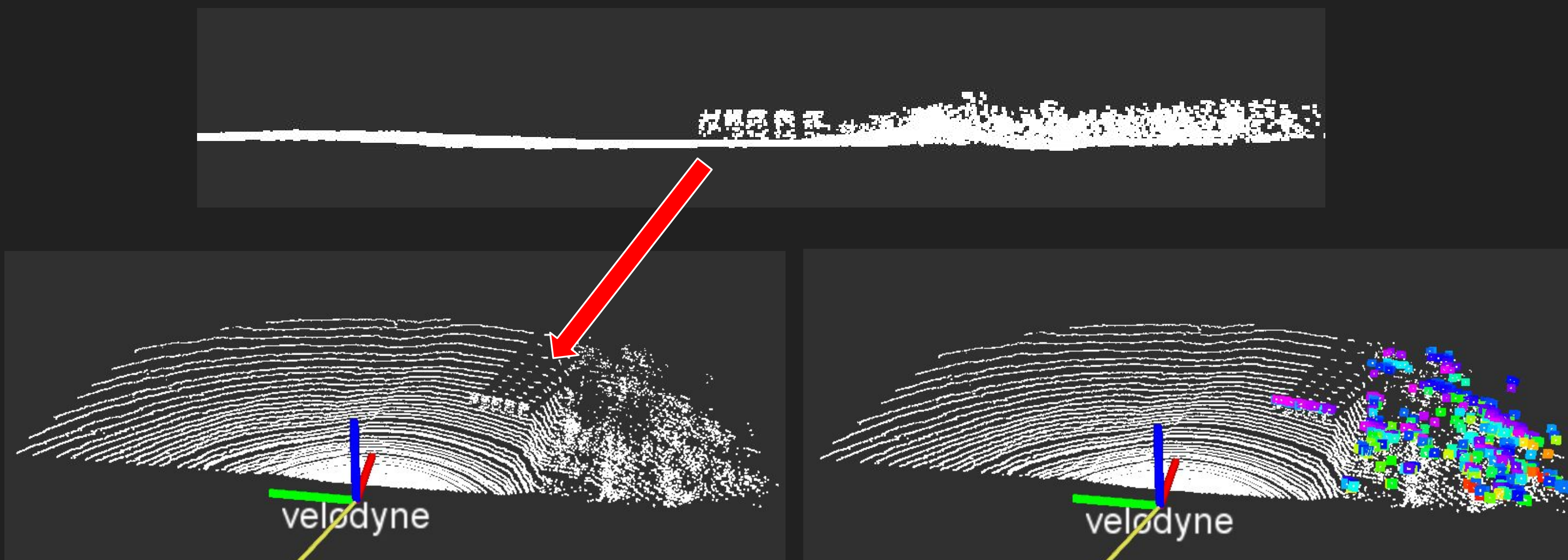
- Use Open Dyanmic Engine (ODE) to cast ray from 4 wheels down to the mesh
- Calculate vehicle pitch and roll from 4 contact points
- Collision condition:
  - Pitch > threshold
  - Roll > threshold
  - Mesh interact with the ray between two wheels





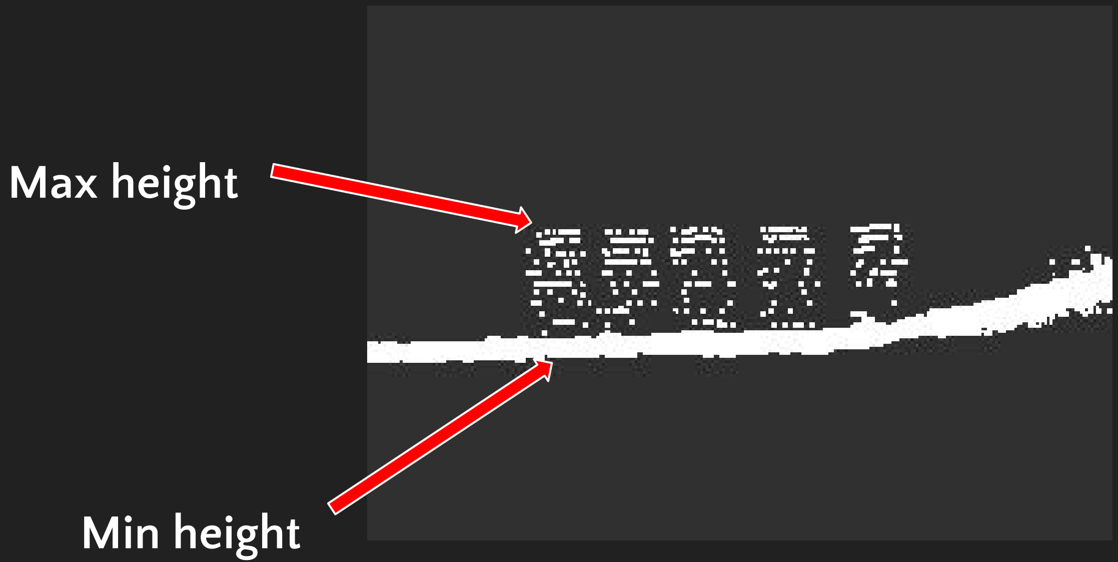
Collision Check :: RANSAC Segmentation

- Use RANSAC to obtain a cloud fitting to the plane model
- Get point cloud outliers to extract obstacles



Collision Check :: Height Map Algorithm

- Calculate the height difference of multiple points within one grid
- If (height_difference > threshold)
  → It's an obstacle



Collision Check :: Implemented Approach

- Requirement: Efficiency + Reliability
  - Efficiency

    1. Using height map algorithm as final approach for efficiency
    2. Bit-wise operation for faster multiplication

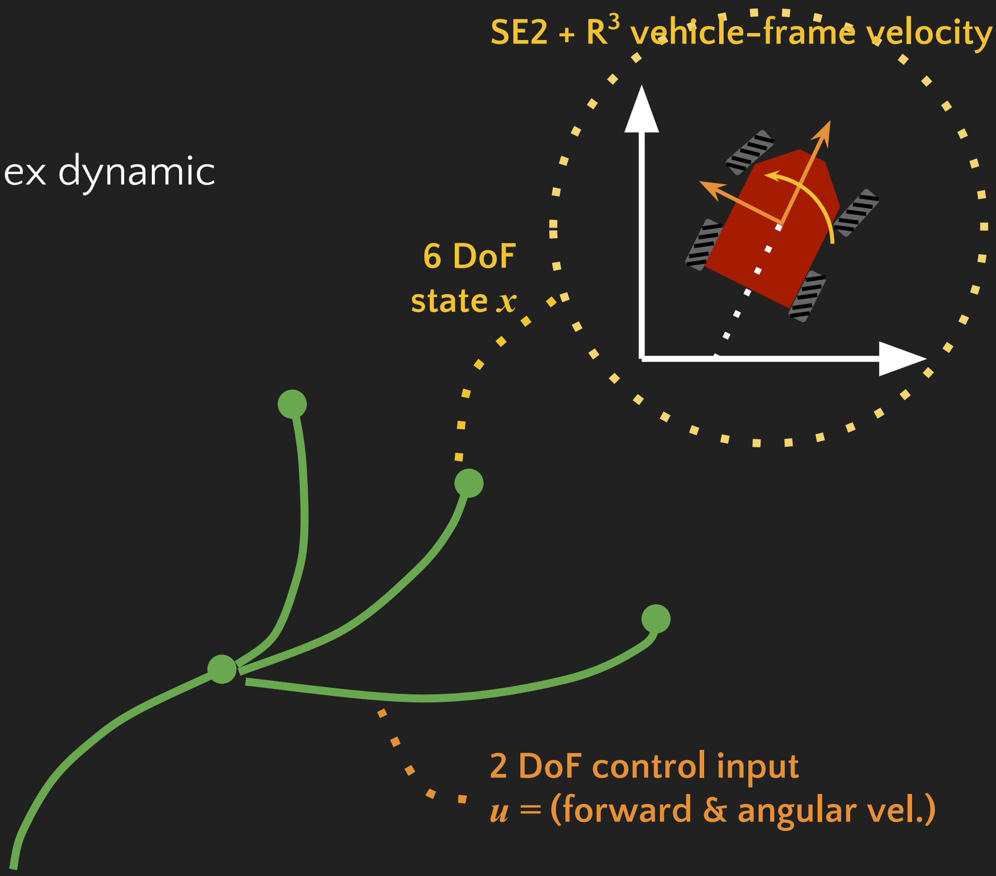
  - Reliability

    1. Using obstacle counter for more robust detection
    2. Dilate obstacle size

RRT-Based Planner Design

Why RRT ?

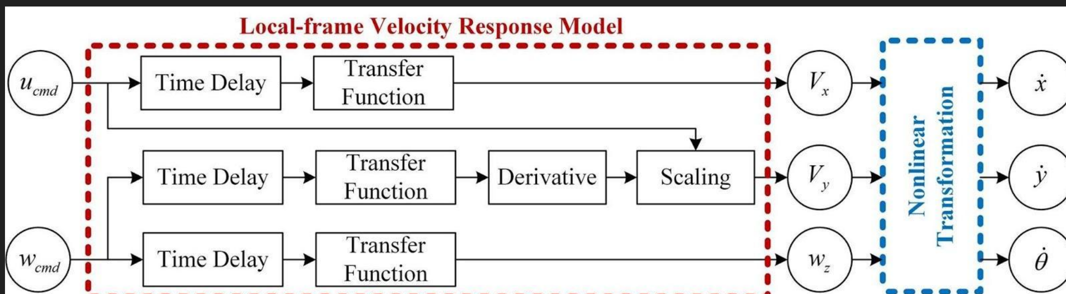
- High dimensional planning w/ complex dynamic model
- Smooth maneuvering

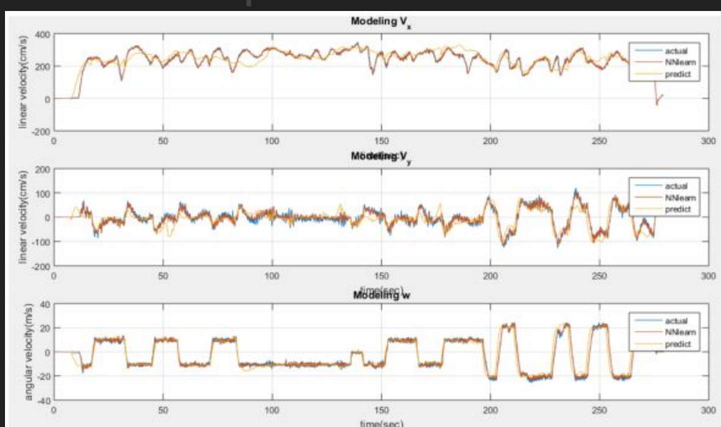


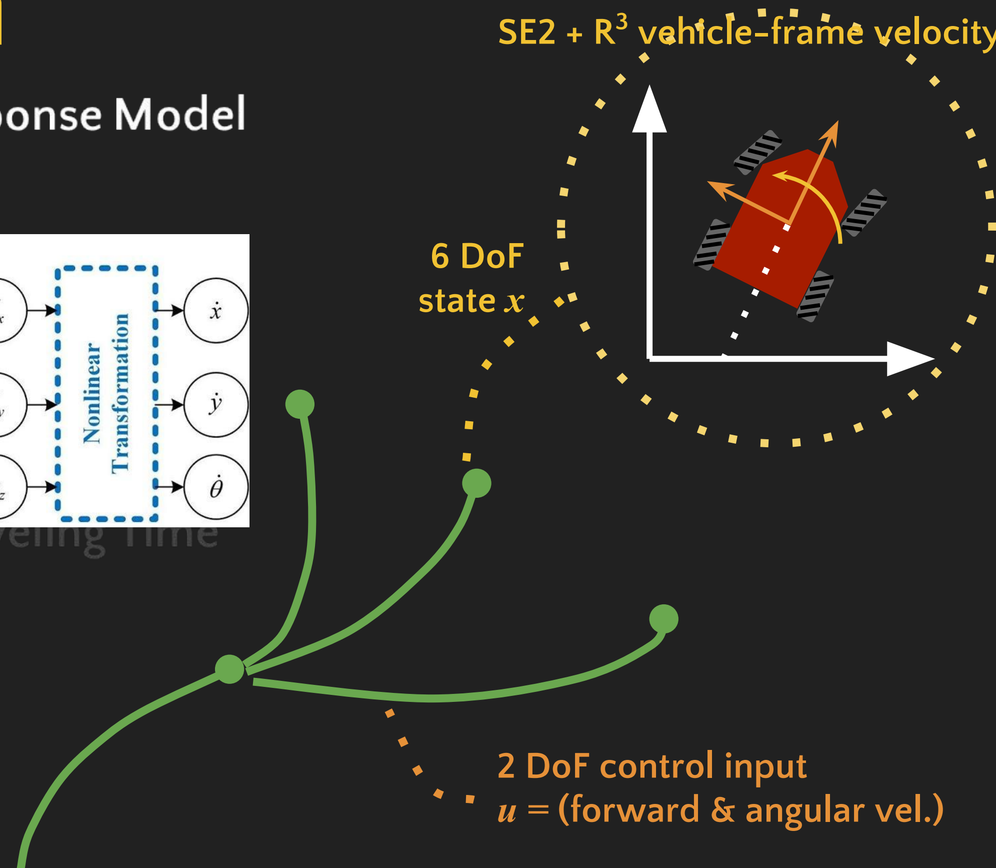
RRT-Based Planner Design

1. Data-Driven Vehicle Dynamic Response Model

Predictive Model : $\dot{x} = f(x, u, t)$







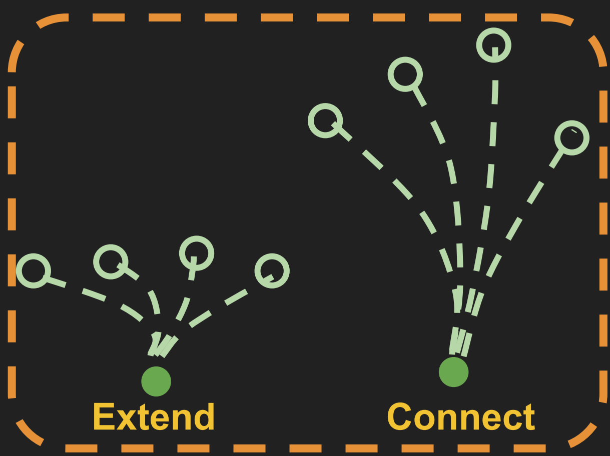
RRT-Based Planner Design

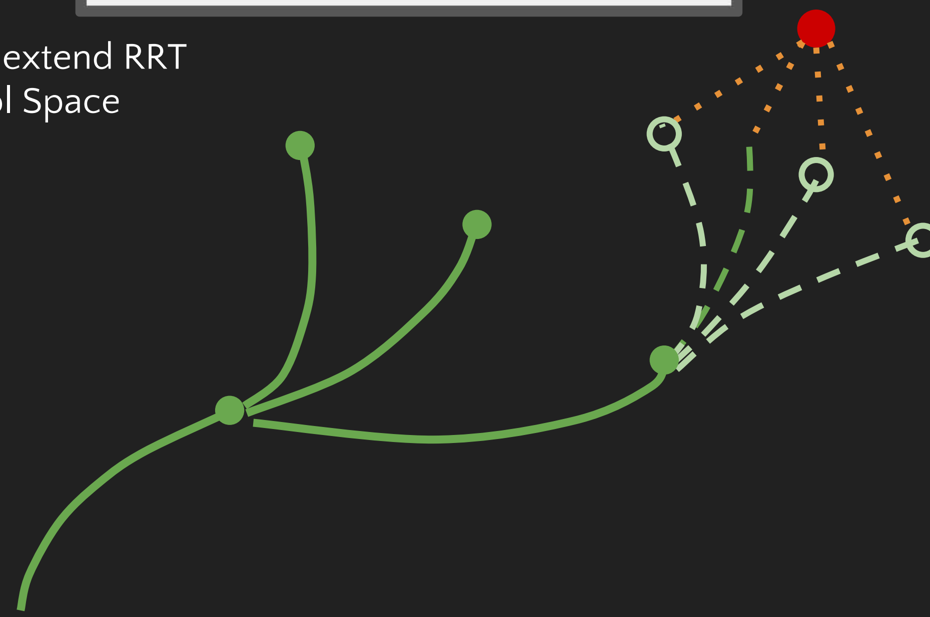
1. Data-Driven Vehicle Dynamic Respons

2. Control Shooting Method
   w/ Random Control Duration

   Flexible constraints between connect and extend RRT
   Steering Mechanisum Constraint in Control Space

$$\boldsymbol{Distance} = \frac{\sqrt{(x_1 - x_2)^2 - (y_1 - y_2)^2}}{\sqrt{(\dot{x}_1 - \dot{x}_2)^2 - (\dot{y}_1 - \dot{y}_2)^2}}$$



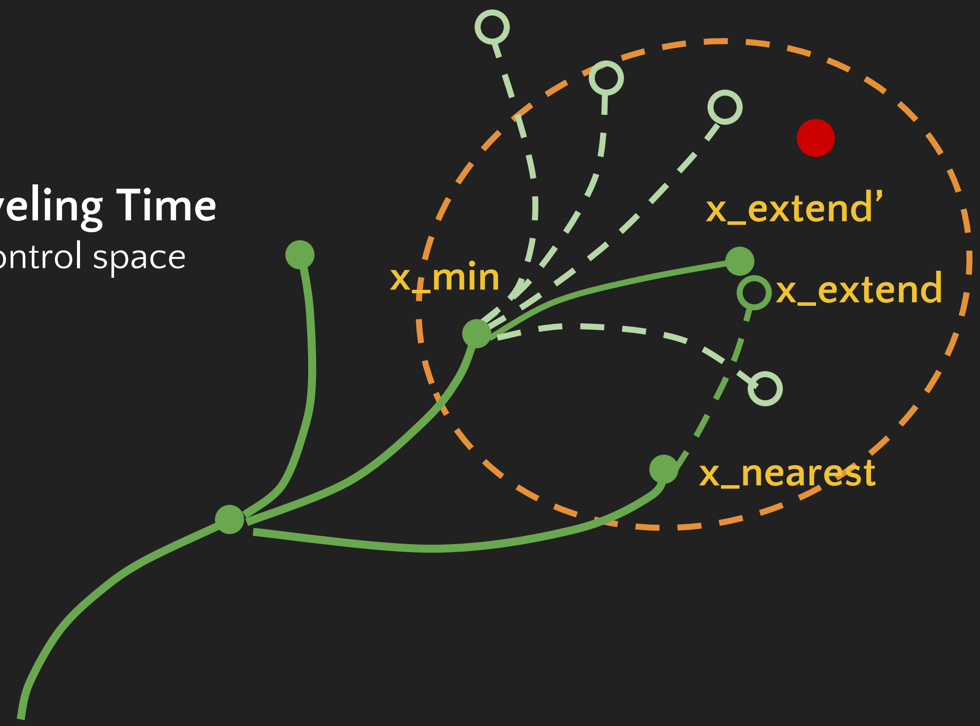


RRT-Based Planner Design

1. Data-Driven Vehicle Dynamic Response Model

2. Control Shooting Method
   w/ Random Propagation Steps

3. Sub-Optimization w/ Minimal Traveling Time
   - One more step needed to reach RRT* in control space
   - Connect extened node to minimum node



[1] Jeong hwan Jeon, Emilio Frazzoli, “Anytime Computation of Time-Optimal Off-Road Vehicle Maneuvers using RRT*”

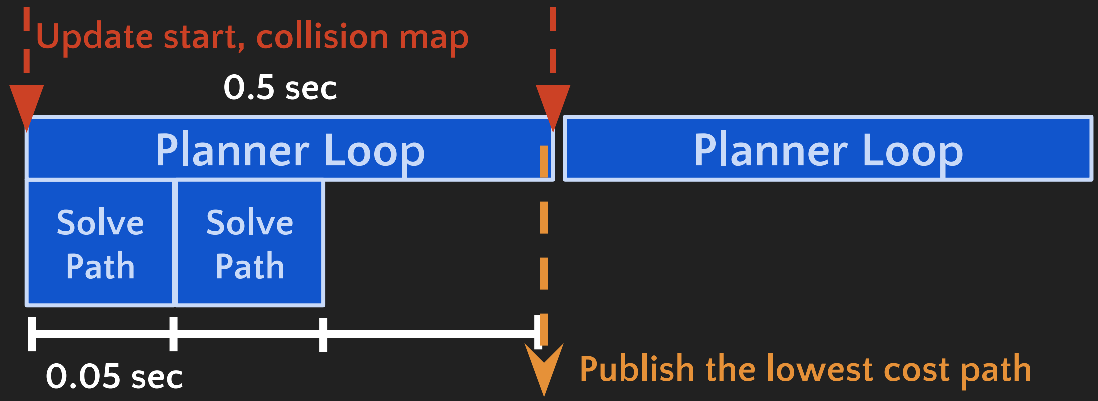
RRT-Based Planner Design

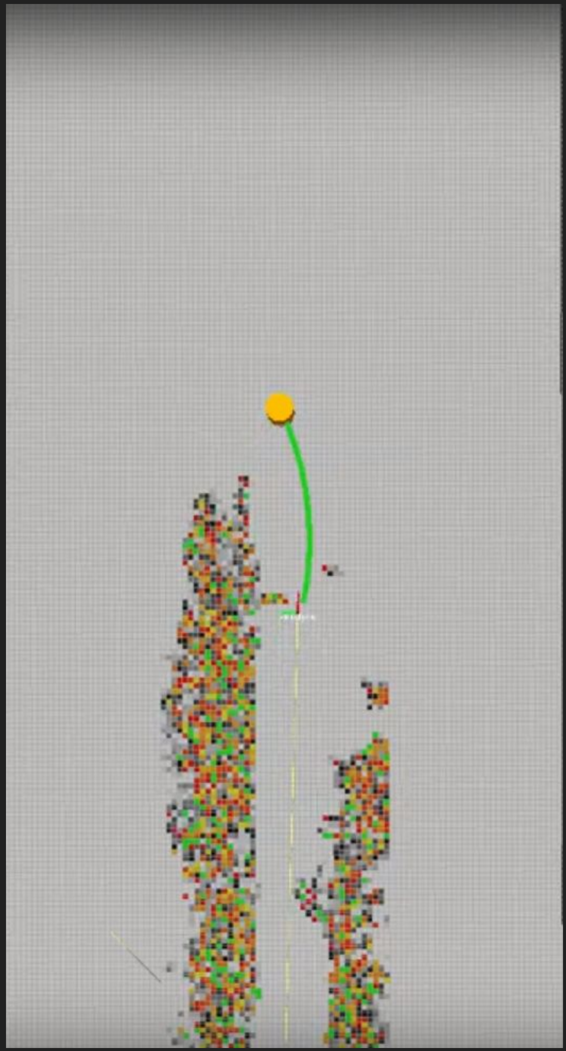
1. Data-Driven Vehicle Dynamic Response Model
2. Control Shooting Method
   w/ Random Propagation Steps
3. Sub-Optimization w/ Minimal Traveling Time
4. **Replanning**

   Solving loop in 20Hz, planner loop in 2Hz

   Estimate start state using dynamic model in **1.**

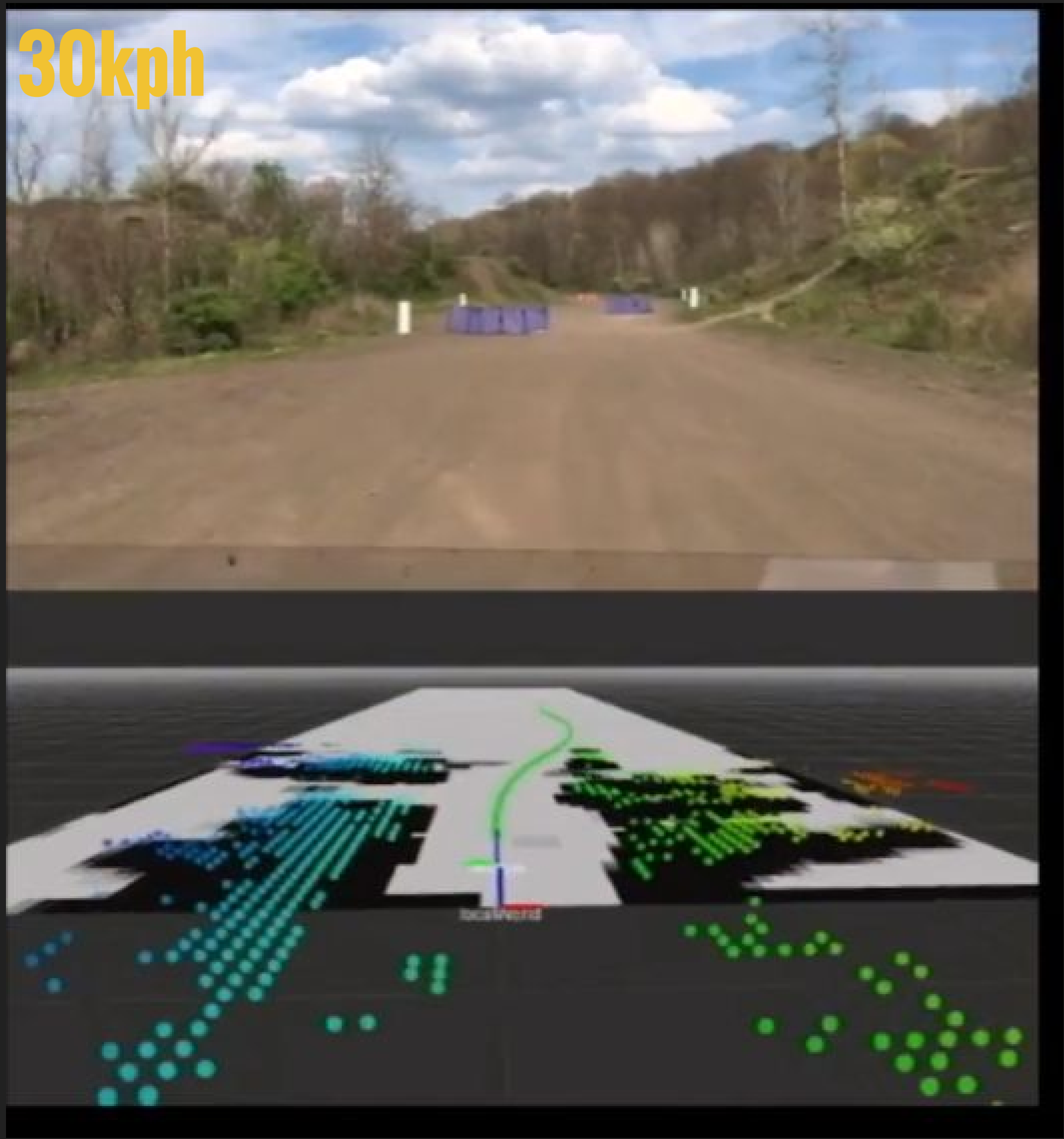
   Path consistence among each planner loop





Final Demo Video

Vehicle operation vel:
  10 ~ 40 kph
Base line requirement:
  20 kph
Final Demo here:
  ~ 30 kph



Any Questions?