Module D: SLAM & Pure Pursuit

Part 1. Introduction to SLAM

Part 2: Particle Filter SLAM

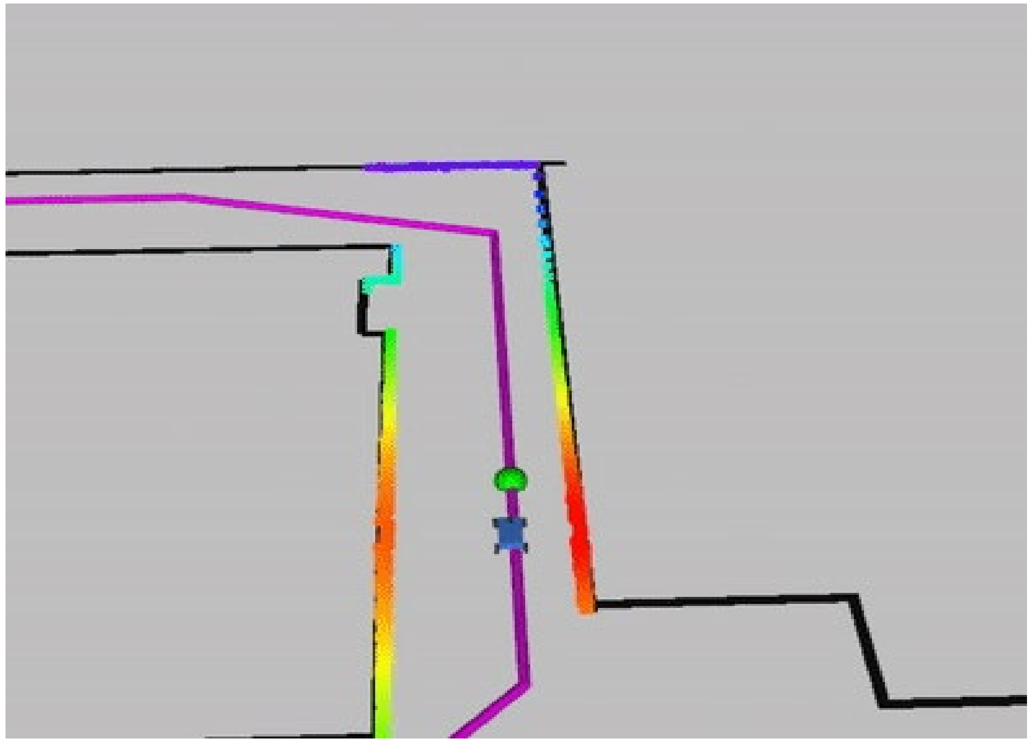
Part 3. Cartographer & Pure Pursuit

References:

https://f1tenth.org and UPenn ESE 680 Slides

https://google-cartographer.readthedocs.io/en/latest/

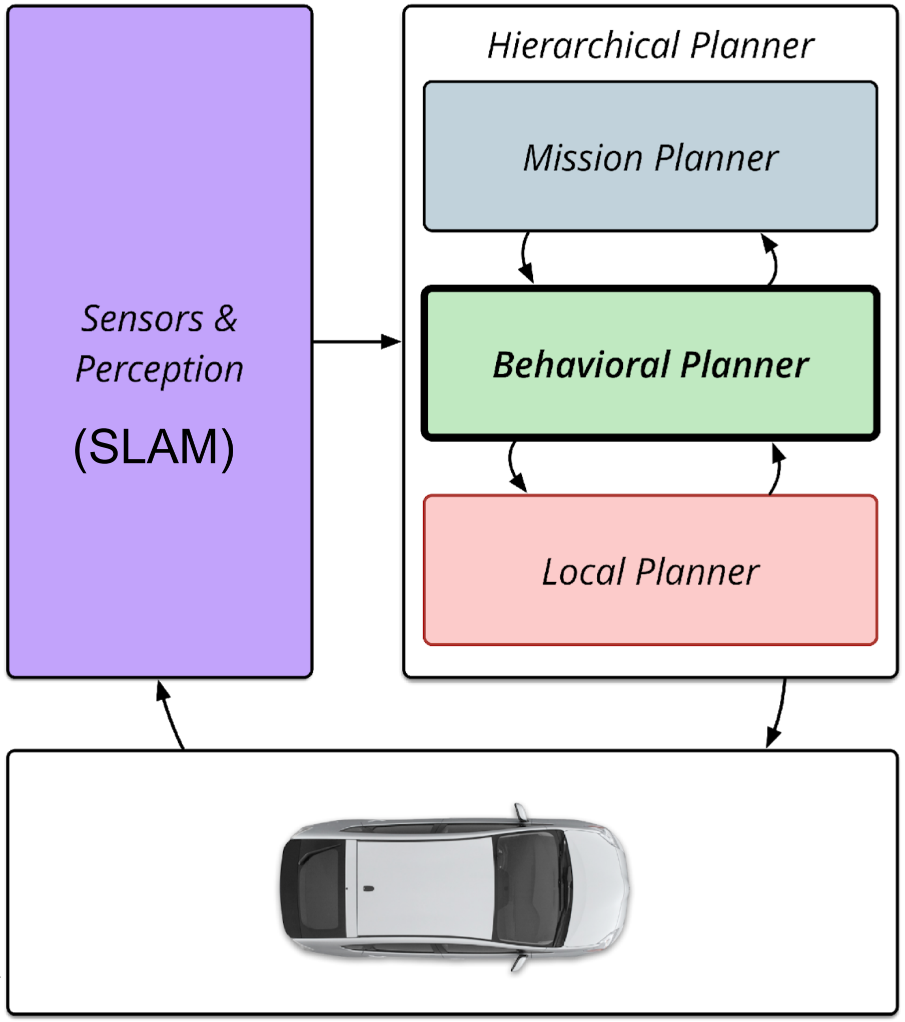
Project 4: Pure Pursuit



- Deliberative Paradigm:
  - Autonomous vehicle planning and control stack
  - Pure Pursuit Tracking Algorithm: design a race line and follow it

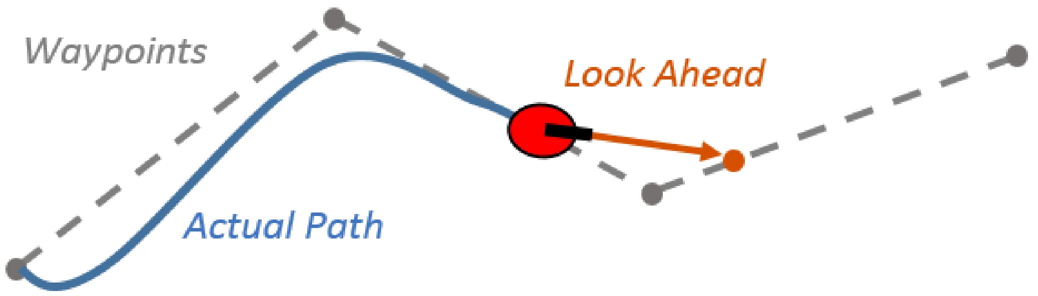
Autonomous Vehicle Planning and Control

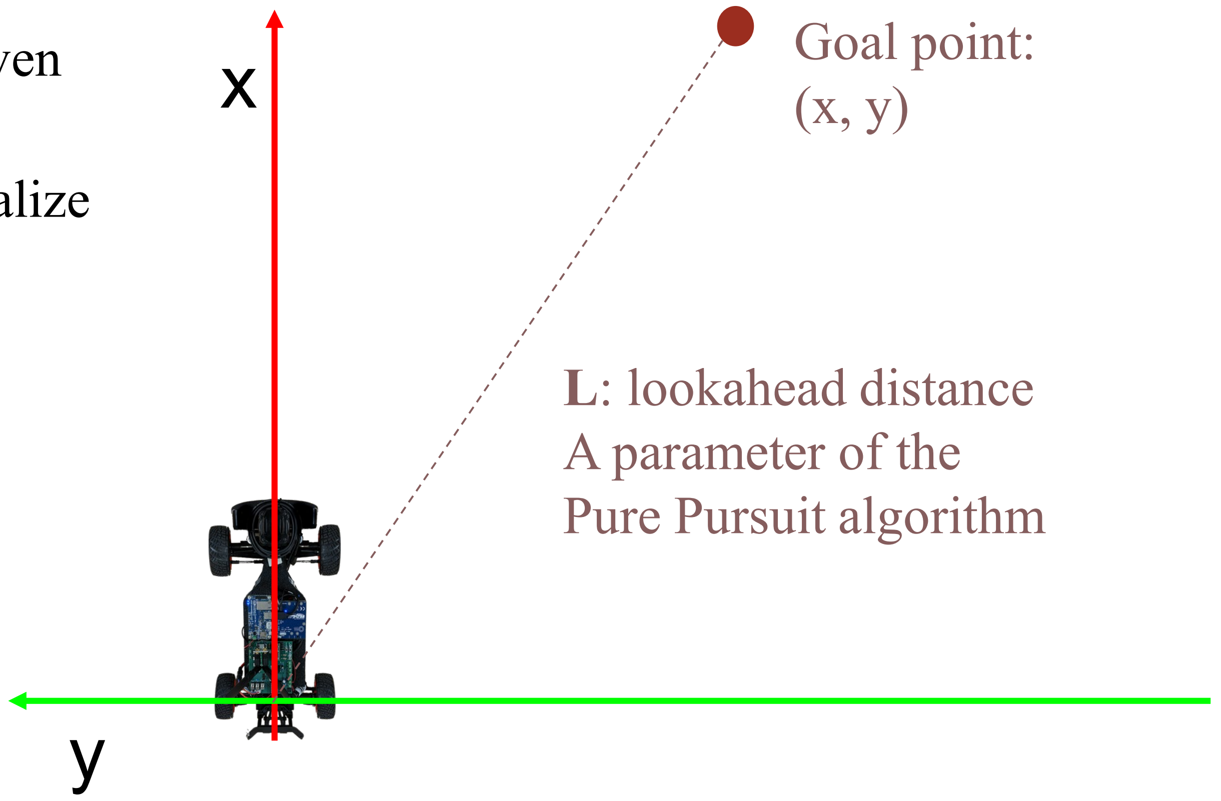
- Sensor and Perception: including SLAM.
- Planning module:
  - Mission Planner: what is the overall goal of the vehicle?
  - Behavioral Planner: what rules should the vehicle follow in different situations?
  - Local Planner: what is the optimal trajectory from position to a goal?
- Control Module
  - How do we track a given trajectory?
  - How do we correct for actuation errors?
- Reference: R. Craig Coulter, "Implementation of the Pure Pursuit Path tracking Algorithm," CMU Tech report, 1992. https://www.ri.cmu.edu/pub_files/pub3/coulter_r_craig_1992_1/coulter_r_craig_1992_1.pdf



Pure Pursuit Tracking Algorithm

- Assumptions:
  - Vehicle is given a sequence of 2D positions, i.e. waypoints, to follow, vehicle knows where the given waypoints are in the vehicle's frame of reference;
  - Underlying assumption is that the vehicle can localize itself and the waypoints are transformed from the global world frame to device frame;
- Goal:
  - Follow the waypoints as much as possible





Figures From Upenn ESE680 Slides

How to compute the steering angle?

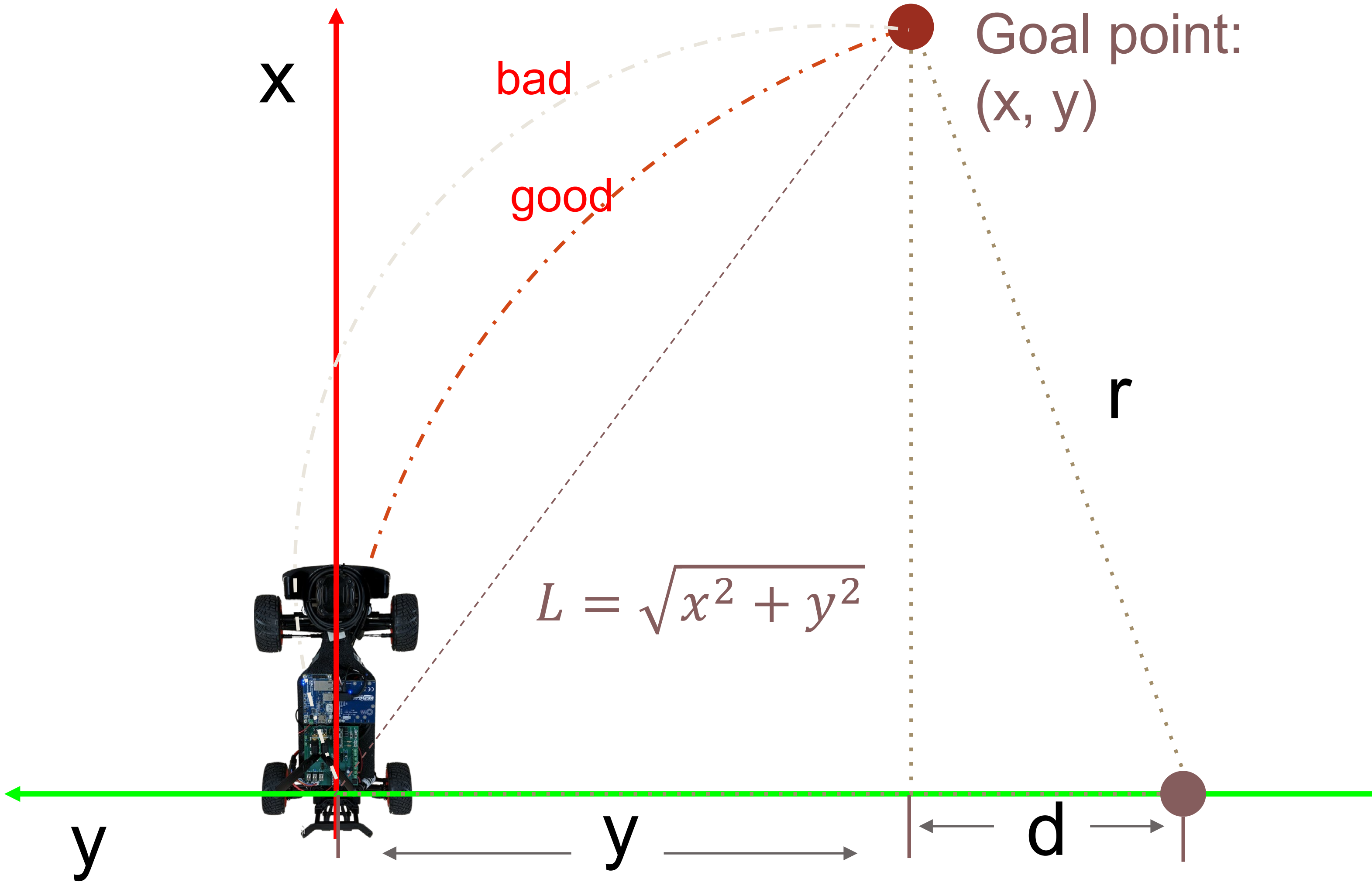
Constrain the center of the arc to be on the y-axis

$$r = \frac{L^2}{2|y|}$$

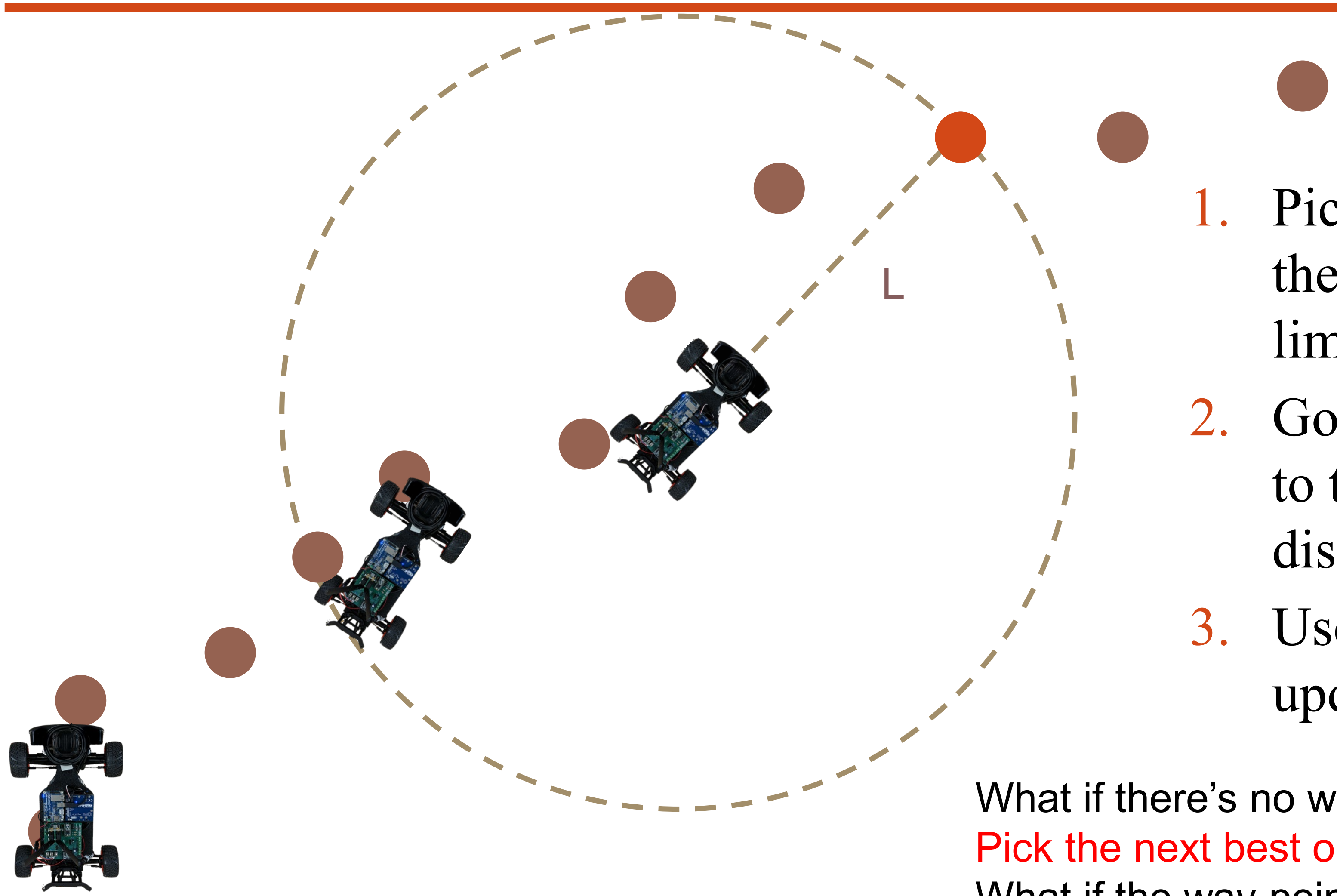
Curvature is the inverse of radius

$$\gamma = \frac{1}{r} = \frac{2|y|}{L^2}$$

Steering angle should be **proportional** to the curvature of the arc → use P control



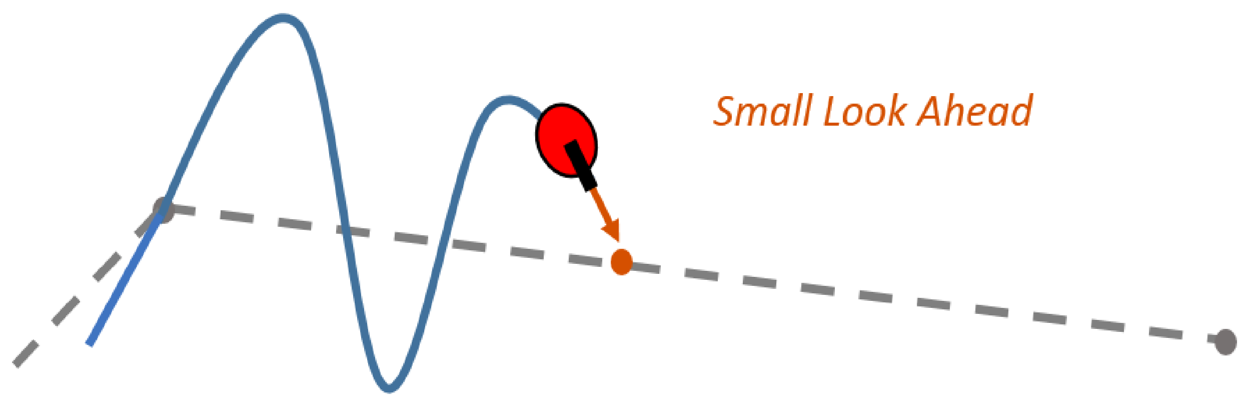
Choose Waypoint and Update Goal

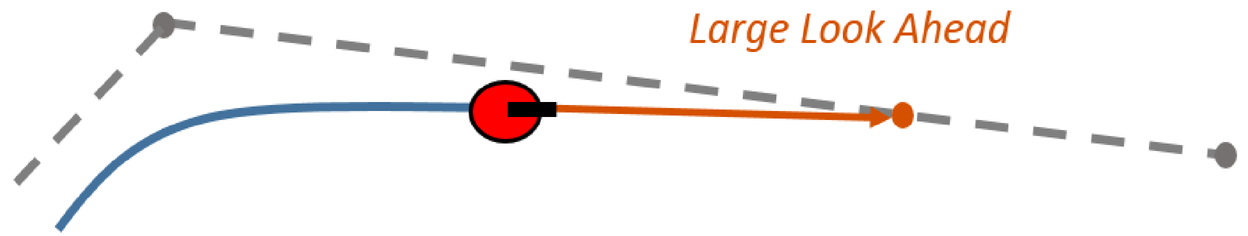


1. Pick the waypoint that is closest to the car: note the steering angle is limited to $\pm 24°$
2. Go up to the waypoint until you get to the one that is one lookahead distance away from the car
3. Use that as the current waypoint, update the goal point

What if there's no waypoints exactly L away from the car?
Pick the next best one (closest to one L away)
What if the way points are too far away?
Interpolate the waypoints

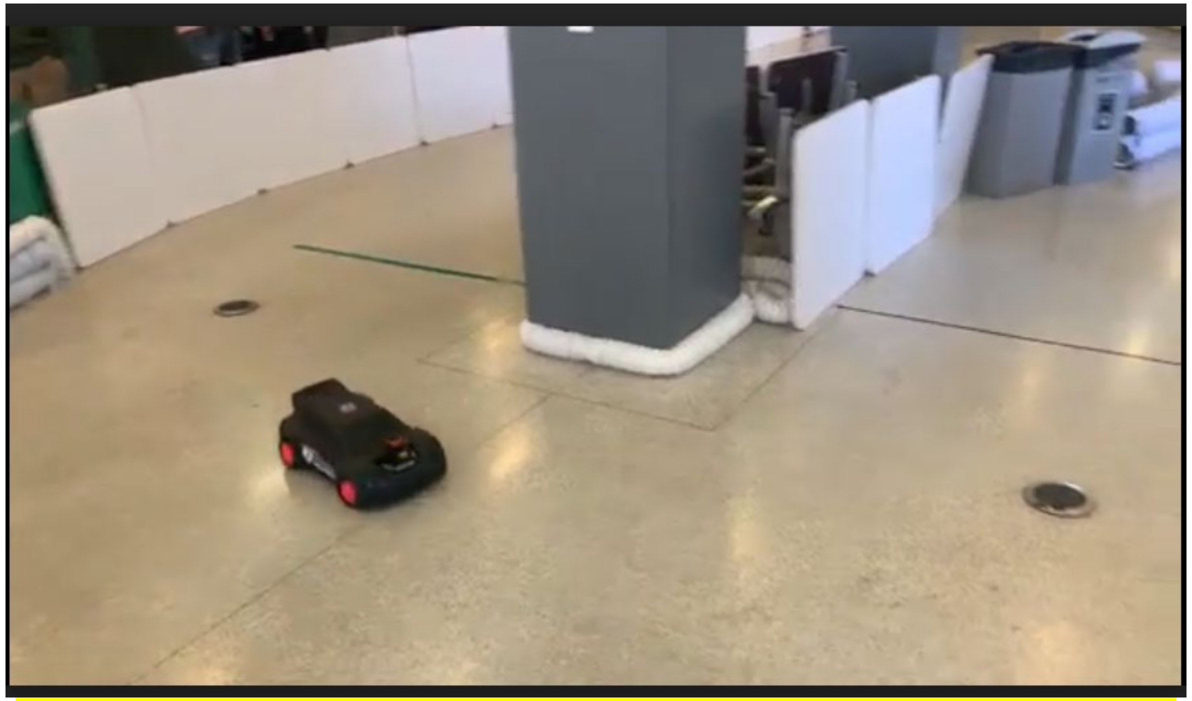
Tuning the Parameters





- The parameter L (lookahead distance) is the most important parameter of pure pursuit.
- Smaller L leads to more aggressive maneuvering to track tighter arc, and the tighter arcs might be against dynamical limits of the car.
- Larger L leads to smoother trajectory but larger tracking errors, might lead to close calls with obstacles.
- The parameter also changes with velocity which also changes with steering angle

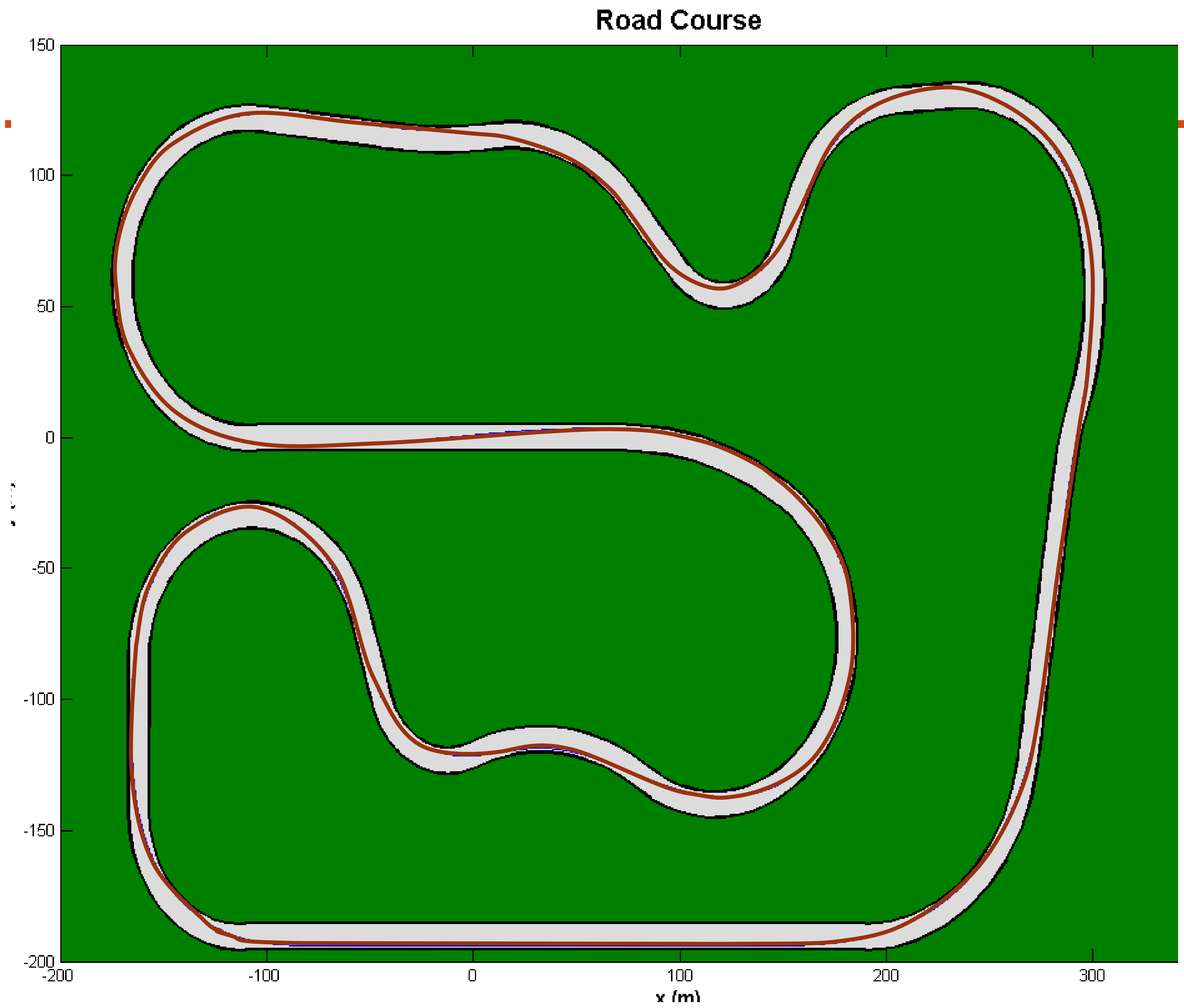
Project 4: Pure Pursuit Tasks



UPenn video
https://drive.google.com/file/d/1JUTJvDrONAKWd5wNnuphJCJByNb4FKfs/view

Also MIT video
https://www.youtube.com/watch?v=lQ1GtNJvBjg

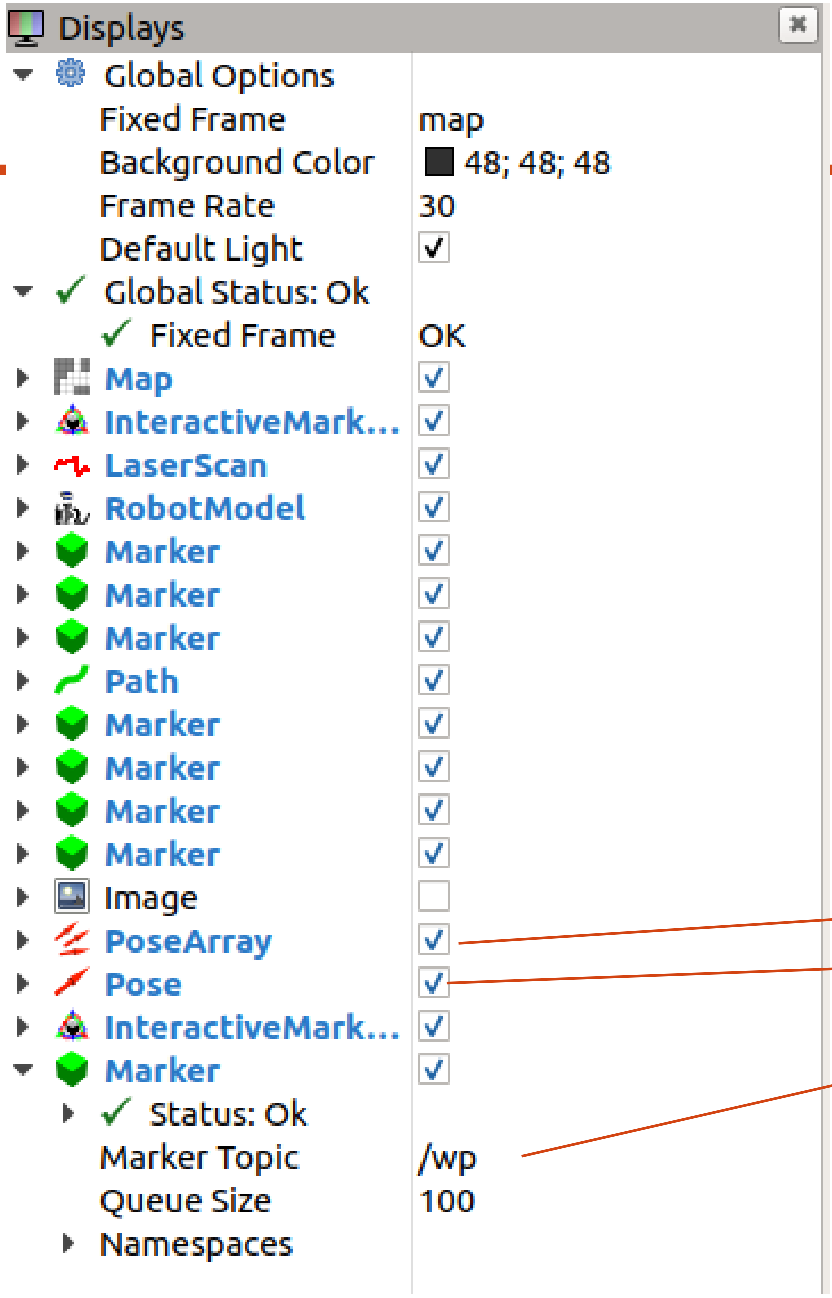
- Create a map and waypoints:
  - Select a map in simulator or particle_filter packages or any online databases
  - Create a list of waypoints on the track:
    - drive the robot along the track using an existing nav method with particle_filter.py
    - Visualize the waypoints and modify them
- Design the Pure_Pursuit node:
  - Pick/edit waypoints to track at each frame
  - Set steering angle to track the current waypoint
  - Update the waypoint to track as you go
  - Tune the velocity and lookahead distance *L* which will change the behavior of pure pursuit the most.



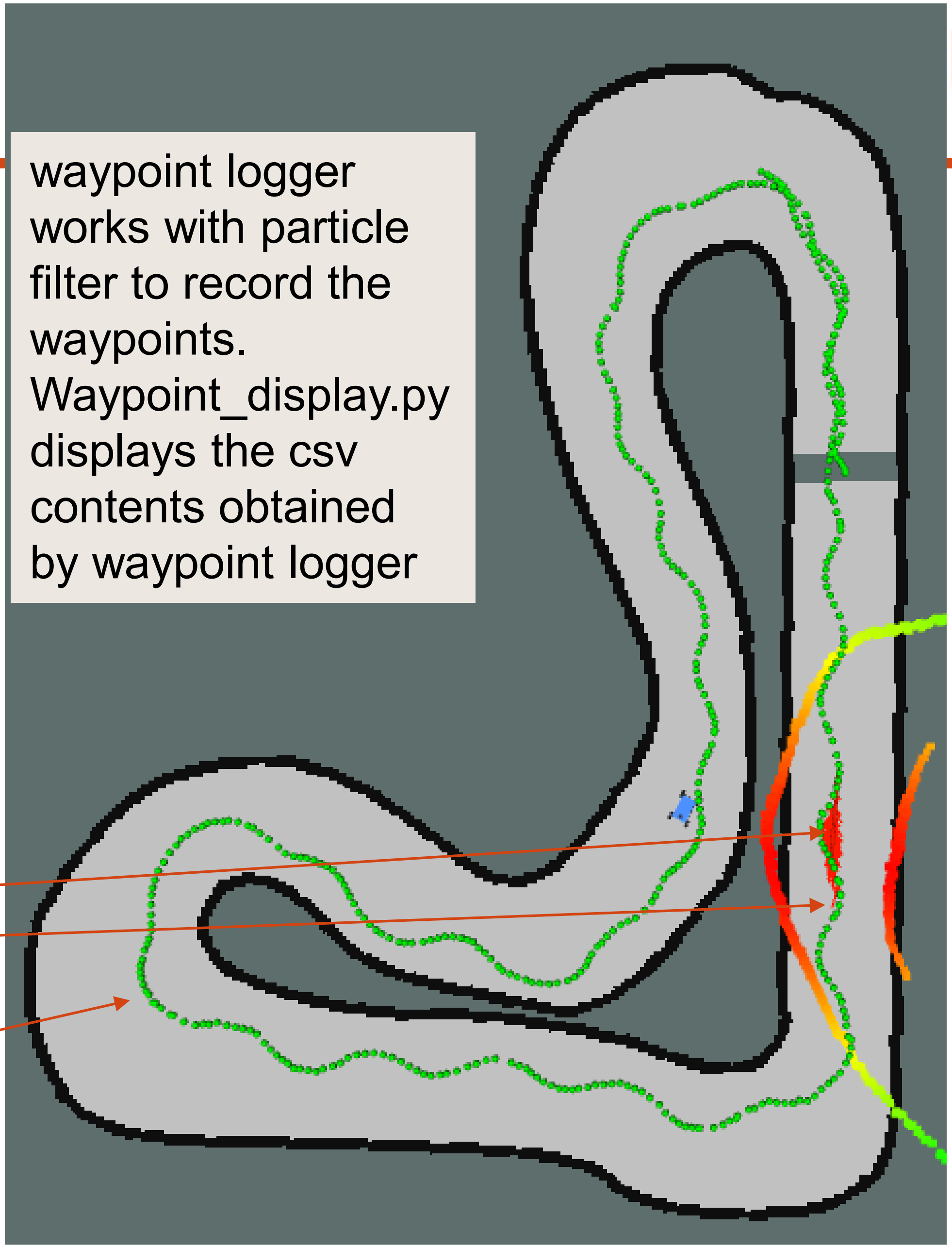
More control strategies are available at:

1. MotorTrend Channel: the Racing Line Ep. 4
https://www.youtube.com/watch?v=3HwnJ1VkWA4
Thanks to Valentin Post for sharing this link.

2. J. M. Snider, "Automatic Steering Methods for Autonomous Automobile Path Tracking," CMU-RI-TR-09-08, 2009
https://www.ri.cmu.edu/pub_files/2009/2/Automatic_Steering_Methods_for_Autonomous_Automobile_Path_Tracking.pdf



Displays
- Global Options
  - Fixed Frame: map
  - Background Color: ■ 48; 48; 48
  - Frame Rate: 30
  - Default Light: ✓
- ✓ Global Status: Ok
  - ✓ Fixed Frame: OK
- Map ✓
- InteractiveMark... ✓
- LaserScan ✓
- RobotModel ✓
- Marker ✓
- Marker ✓
- Marker ✓
- Path ✓
- Marker ✓
- Marker ✓
- Marker ✓
- Marker ✓
- Image ☐
- PoseArray ✓
- Pose ✓
- InteractiveMark... ✓
- Marker ✓
  - ✓ Status: Ok
  - Marker Topic: /wp
  - Queue Size: 100
  - Namespaces



waypoint logger works with particle filter to record the waypoints. Waypoint_display.py displays the csv contents obtained by waypoint logger

LEHIGH UNIVERSITY

The particle filter publishes the particles and inferred pose. Before the SLAM convergence, the particles and inferred pose are far away from the actual car location. The inferred map is misaligned with the actual map.