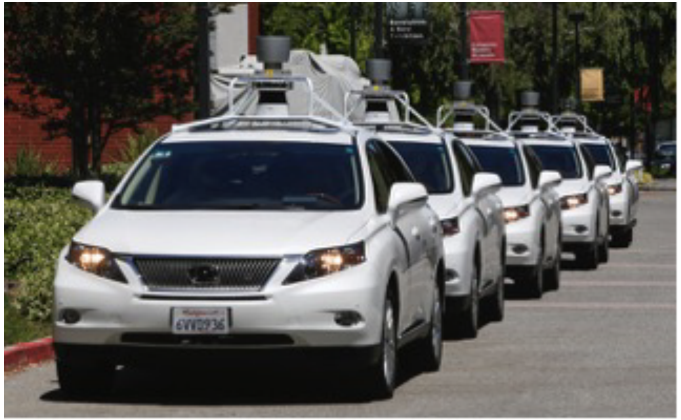
Safe Navigation in Dynamic, Unknown, Continuous, and Cluttered Environments

Mike D'Arcy, Pooyan Fazli, and Dan Simon
People and Robots Laboratory (PeRL)
Electrical Engineering and Computer Science Department
Cleveland State University

Introduction

- Safe Navigation through Dynamic , Unknown, Continuous, and Cluttered environments is a crucial ability for autonomous robots

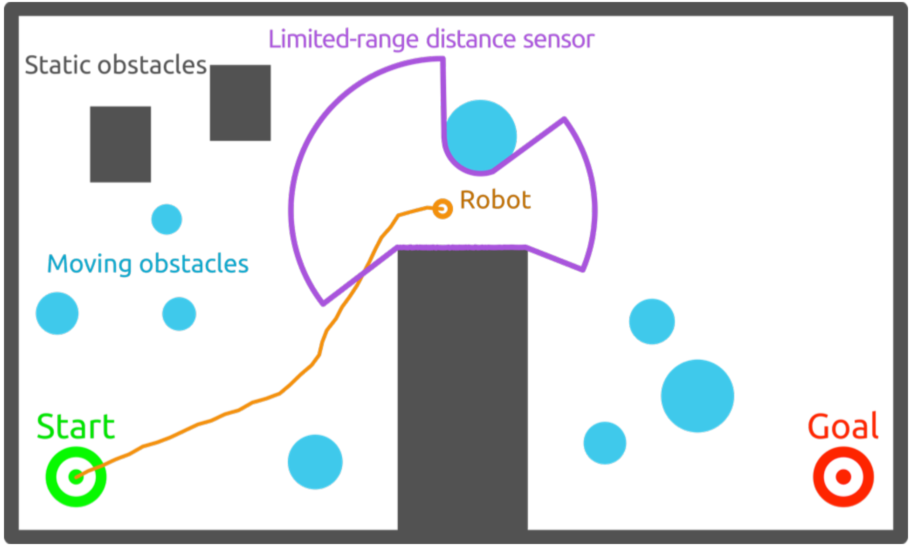


Google Autonomous Cars

Amazon, Kiva Systems Robots

Problem Definition

- 2D unknown configuration space
- Static and moving obstacles
- Point-sized holonomic robot
- Limited-range distance sensor (e.g. lidar)
- Knows its own position and goal position



Objective: To navigate from an initial point to the goal without colliding with any of the dynamic or static obstacles

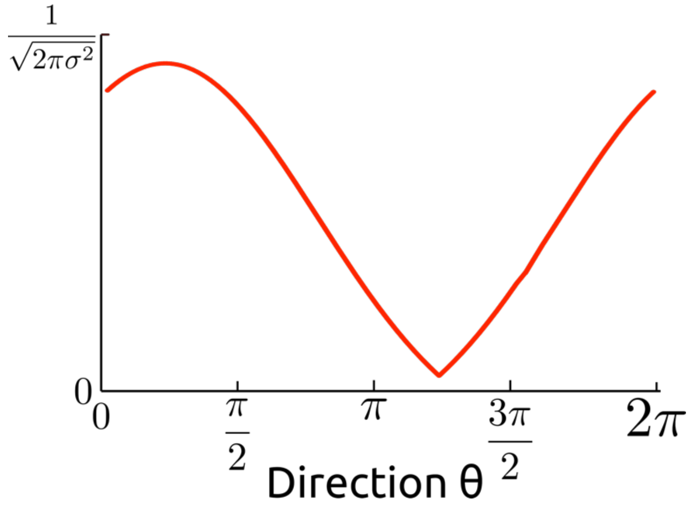
Probabilistic Local Planner (ProbLP)

- ProbLP is a local planner
- Based on trajectory sampling, but…
- We construct a probability distribution of promising directions
- Sampling from this distribution finds safe and efficient trajectories faster

Constructing the Probability Distribution

Target distribution

- Gaussian centered on the direction to the goal



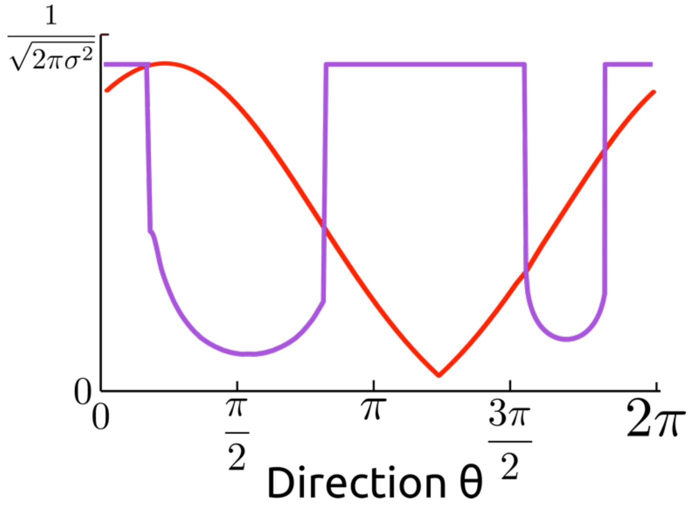
Constructing the Probability Distribution

Target distribution

- Gaussian centered on the direction to the goal

Obstacle distribution

- The range sensor data



Constructing the Probability Distribution

Target distribution

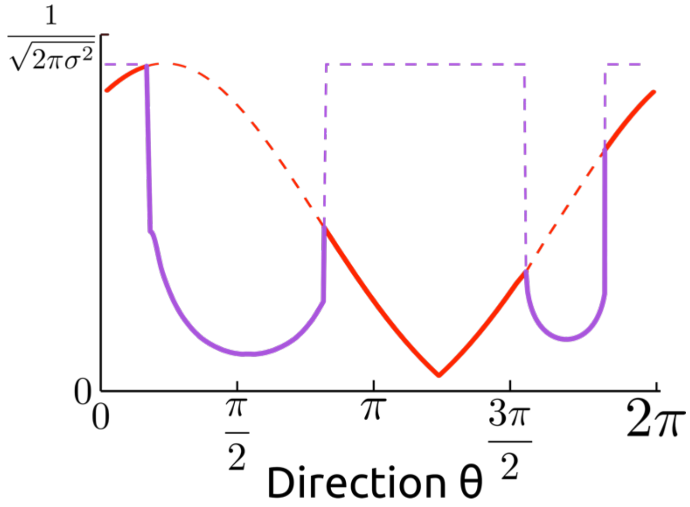
- Gaussian centered on the direction to the goal

Obstacle distribution

- The range sensor data

Final distribution

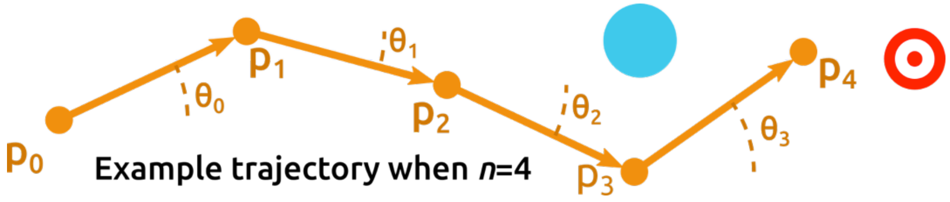
- Minimum of the target and obstacle distributions
- Area normalized to 1



Selecting the Candidate Trajectories

Starting at the current location p_0:

- Pick a direction θ from the distribution
- Put p_{i+1} a small distance from p_i in the direction of θ
- Repeat up to p_n



Scoring Trajectories

- Distance score
  - Distribution value at the angle between p_0 and p_n, multiplied by the ratio of trajectory length to straight-line distance from p_0 to p_n
- Safety score
  - Use obstacle predictor to get the probability of a collision if *p* is followed
- Take a weighted sum of distance and safety scores

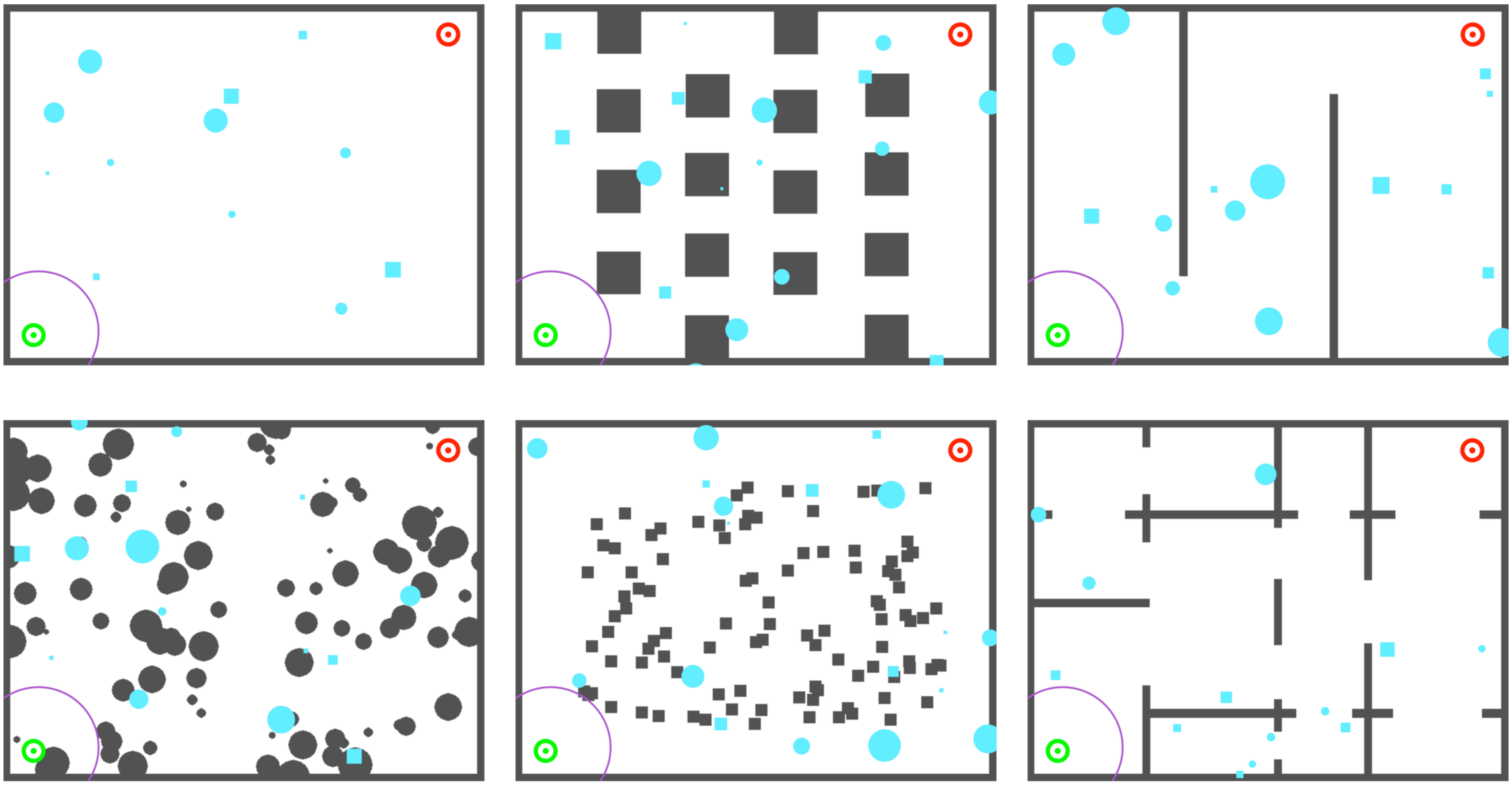
Adding a Global Planner

- ProbLP alone has no memory, so it gets stuck in dead ends
- Global planner plans a sequence of subgoals for ProbLP to follow
- We use Dynamic Rapidly-exploring Random Trees (DRRT)
- The environment is still unknown: DRRT builds a map as it goes

Experiments

- Four speed modes
  - **SP-1:** All obstacles move 0.5 m/s (slower than the robot)
  - **SP-2:** All obstacles move 1.0 m/s (same as the robot)
  - **SP-3:** All obstacles move 1.5 m/s (faster than the robot)
  - **SP-4:** Each obstacle moves randomly between 0.5-1.5 m/s

- Compared DRRT+ProbLP vs DRRT+Straight-Line Local Planner (DRRT-SLLP)



Results

- DRRT-ProbLP averaged 0.27 collisions vs 1.20 for DRRT-SLLP
- Only 10% longer paths to achieve the safety benefit

Conclusion and Future Work

- ProbLP is a reactive planner for navigating in dynamic, unknown, continuous, and cluttered environments
- In the future, we would like to
  - Work in 3D environments
  - Better handle kinematic constraints
  - Optimize algorithm parameters automatically