4-28-2023

Experimental Validation of Motion Planning and Control Algorithms on Ground Robot Platforms

Yifei Li
Washington University in St. Louis, liyifei@wustl.edu

Washington University in St. Louis
James McKelvey School of Engineering

Experimental Validation of Motion Planning and Control Algorithms on Ground Robot Platforms

Yifei (Bruce) Li, Department of Electrical & Systems Engineering
Research Advisor: Professor Yiannis Kantaros

INTRODUCTION

Background

Robotic systems are increasingly being used in various applications, from manufacturing and logistics to healthcare and education. However, the performance of these systems relies heavily on their ability to navigate through their environment.

Purpose

The purpose of this research project is to experimentally validate the effectiveness of a robot planning and control algorithm on a ground robot platform.

Specifically, we will use the Turtlebot 3 Waffle Pi, which is a 2-wheel ROS based robot platform with built-in 360-degree lidar distance sensor SLAM algorithms.

For the motion planning algorithm, we choose the incremental sampling-based motion planning algorithm RRT*, a improved version of RRT (Rapidly-exploring Random Tree), that has been proven to have asymptotic optimality, improved solution quality, and flexibility compared to RRT [1].

PROBLEM STATEMENT

Given

- The physical environment distributed with obstacles
- Coordinates of start and destination points
- The dimensions and available controls of the robot

Find

- The control scheme of the robot that departs from the start point and reach desired destination while avoiding obstacles

METHOD & PROCEDURE

1. Get knowledge of the environment

- Done by running built in SLAM and Teleop node on the Turtlebot
- The map captured by SLAM is saved as an Occupancy Grid Map

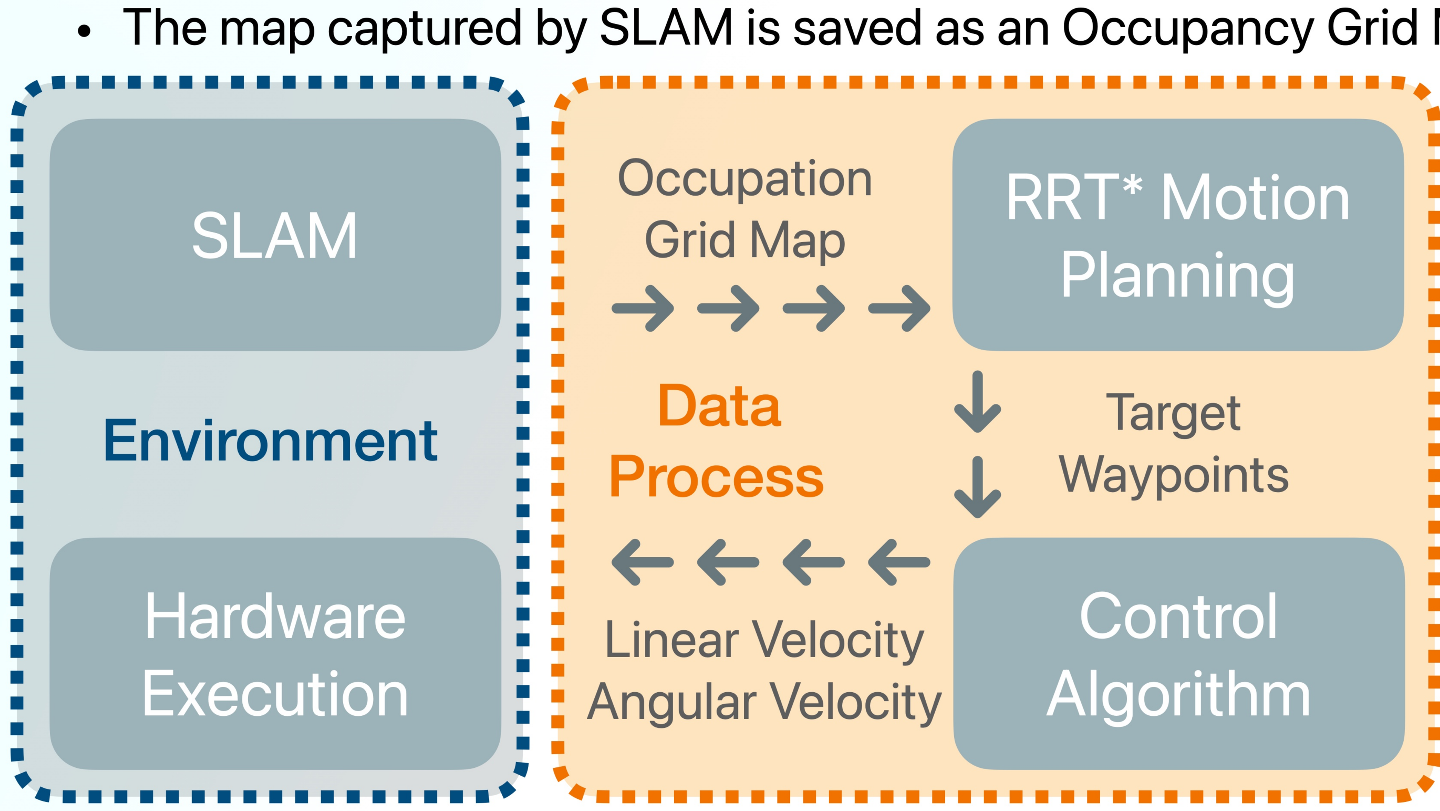

Fig. 1 a block diagram of the procedure of solving the problem

2. Plan the motion path

- Given a existing map
- Given start and target point
- Use RRT* planning [1]

Algorithm 4: Extend$_{RRT^*}$

$V' \leftarrow V$; $E' \leftarrow E$;
$x_{\text{nearest}} \leftarrow$ Nearest(G, x);
$x_{\text{new}} \leftarrow$ Steer(x_{nearest}, x);
if ObstacleFree$(x_{\text{nearest}}, x_{\text{new}})$ **then**
  $V' \leftarrow V' \cup \{x_{\text{new}}\}$;
  $x_{\text{min}} \leftarrow x_{\text{nearest}}$;
  $X_{\text{near}} \leftarrow$ Near$(G, x_{\text{new}}, |V|)$;
  for *all* $x_{\text{near}} \in X_{\text{near}}$ **do**
    if ObstacleFree$(x_{\text{near}}, x_{\text{new}})$ **then**
      $c' \leftarrow$ Cost$(x_{\text{near}}) + c($Line$(x_{\text{near}}, x_{\text{new}}))$;
      if $c' <$ Cost(x_{new}) **then**
        $x_{\text{min}} \leftarrow x_{\text{near}}$;
  $E' \leftarrow E' \cup \{(x_{\text{min}}, x_{\text{new}})\}$;
  for *all* $x_{near} \in X_{\text{near}} \setminus \{x_{\text{min}}\}$ **do**
    if ObstacleFree$(x_{\text{new}}, x_{\text{near}})$ *and* Cost$(x_{\text{near}}) >$ Cost$(x_{\text{new}}) + c($Line$(x_{\text{new}}, x_{\text{near}}))$ **then**
      $x_{\text{parent}} \leftarrow$ Parent(x_{near});
      $E' \leftarrow E' \setminus \{(x_{\text{parent}}, x_{\text{near}})\}$;
      $E' \leftarrow E' \cup \{(x_{\text{new}}, x_{\text{near}})\}$;
return $G' = (V', E')$

- Present planned path as a sequence of waypoints

3. Drive the robot

- Given the current position and a target position
- Compute next action of angular and linear velocity that the robot need to perform with following transfer function

Algorithm 1: Algorithm to get the next action for a robot

Input: Current state $s = [x, y, \theta]$ and goal $goal = [goal\_x, goal\_y, goal\_theta]$
Output: Next action for the robot in the form of $[linear\_vel, angular\_vel]$
Function *get_next_action s, goal*:
  $x, y, \theta \leftarrow s$
  $goal\_x, goal\_y, goal\_theta \leftarrow goal$

  $dx \leftarrow goal\_x - x$
  $dy \leftarrow goal\_y - y$
  $goal\_dist \leftarrow \sqrt{dx^2 + dy^2}$
  $goal\_angle \leftarrow tan^{-1}(dy, dx)$
  $angle\_error \leftarrow goal\_angle - \theta$

  if $|angle\_error| > threshold$ **then**
    $angular\_vel \leftarrow angle\_error$
  end
  else
    $angular\_vel \leftarrow 0.0$
    $linear\_vel \leftarrow 0.5 \times goal\_dist$
  end
  return $[linear\_vel, angular\_vel]$
end

RESULTS

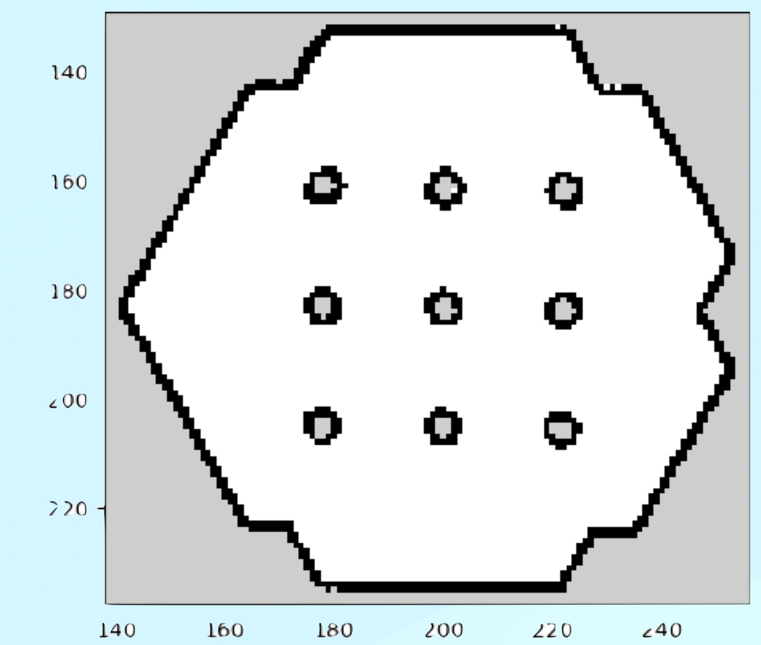

Fig. 2 the occupancy grid map saved as the result of SLAM mapping through a simulator world

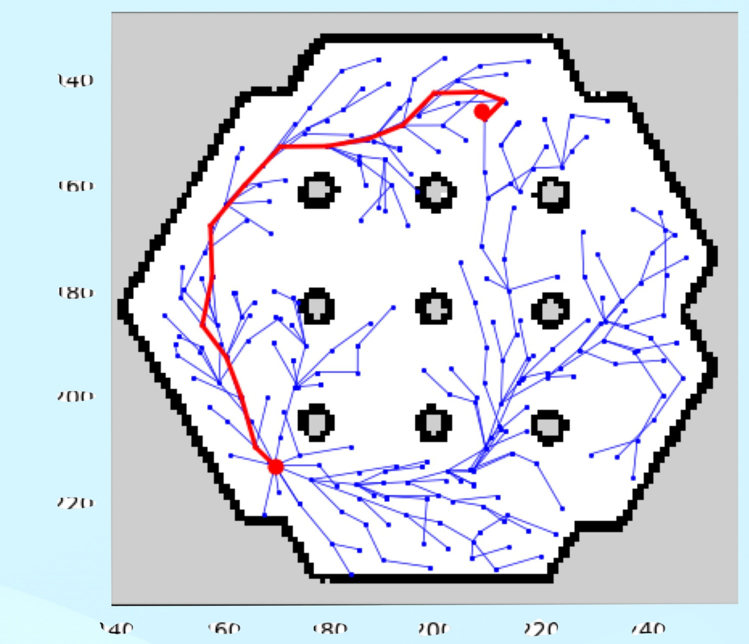

Fig. 3 the explored locations and returned path of the RRT* algorithm with 5000 iteration

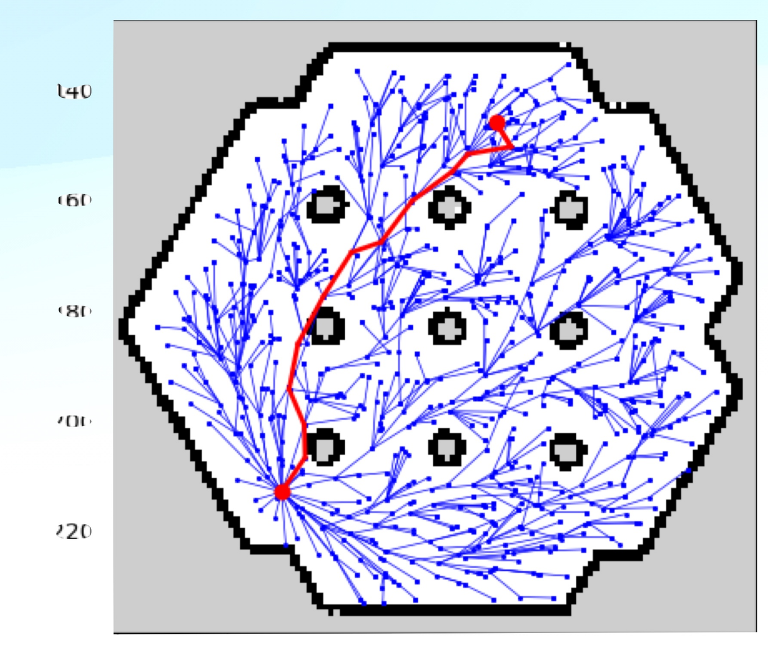

Fig. 4 the explored locations and returned path of the RRT* algorithm with 20000 iteration

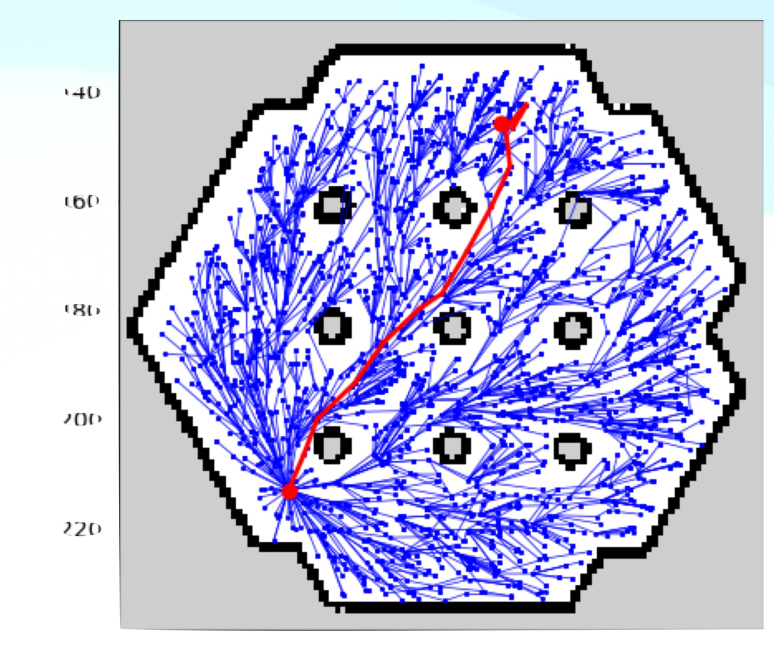

Fig. 5 the explored locations and returned path of the RRT* algorithm with 50000 iteration

We can observe that as the iteration of RRT* exploration increases, the returned path converges to an optimal solution

CONCLUSION & FUTURE DIRECTIONS

In this research project, we presented an experimental validation of robot planning and control algorithms on a ground robot platform.

Our results demonstrated that the proposed approach was effective in generating feasible paths that successfully navigated the robot to the desired destination while avoiding collisions with obstacles

By validating the effectiveness of the RRT* algorithm in a real-world setting with the specific model of robot, we can contribute to the development of more advanced and effective robotic controls.

Future directions could be investigate the effectiveness of the proposed approach with more complex obstacles; extend the proposed approach to plan in unknown environments; and update the planning online while the robot is in motion.

BIBLIOGRAPHY

[1] S. Karaman and E. Frazzoli, "Incremental sampling-based algorithms for optimal motion planning," Robotics: Science and Systems VI, 2010.