Fast Motion Planning of UAVs: Implementation of the SE-SCP Algorithm in ROS

Kaixi Wang[1], Dr. Amir Rahmani[2]
[1]The University of Texas at Austin [2]NASA Jet Propulsion Laboratory

BACKGROUND

Fast Motion Planning (FMP) of autonomous vehicles has been advanced rapidly for robotics research, particularly for trajectory planning of spacecraft. The FMP team at JPL and the California Institute of Technology has developed an algorithm (called SE-SCP) for autonomous vehicles in environments with many fixed obstacles. Current development and simulation of this algorithm is still being tested with software such at MATLAB as well as the collaborative robotics software called the Robot Operating System (ROS). By developing the SE-SCP algorithm in ROS, a ROS package can be created and uploaded online. This will provide opportunities for the public to easily utilize the software and apply the SE-SCP algorithm for motion planning to their own autonomous vehicles.

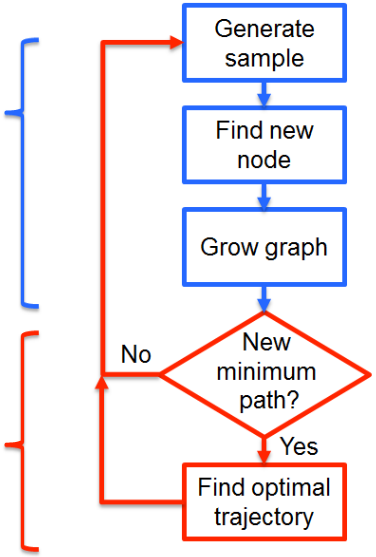
METHOD: SE-SCP

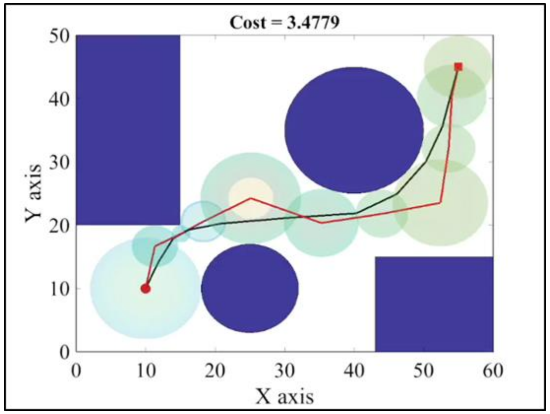
The spherical expansion and sequential convex programming (SE-SCP) algorithm is computationally efficient and guarantees any-time local optimality for a given function on top of being faster than other sampling-based motion planning methods. Spherical expansion (SE) is a randomized sampling to explore the workspace of the autonomous vehicle and it finds and initial cost-minimized path. Sequential convex programming (SCP) then optimizes this path and computes a locally optimal trajectory.

RESULTS

1. Spherical Expansion Step

2. Sequential Convex Optimization Step





Example of simulated trajectory planning using the SE-SCP algorithm

DISCUSSION: ROS

ROS is an open source, free, and flexible framework for writing robot software. It allows for collaborative software and has both hardware and software implementations. ROS includes a collection of tools, libraries, and conventions specifically for robotics improvement for the general public. Therefore it is advantageous over other software because ROS avoids the need to develop your own code for many aspects of robot control.

Both C++ and Python code was used for the implementation of SE-SCP in ROS. The following was updated in order to build a functional SE-SCP simulation:

- Update for multiple iterations of SE-SCP
- Update different types of dynamics
  - Distance Minimizer
  - Single Integrator
  - Double Integrator

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation through the Robert Noyce Teacher Scholarship Program under Grant # 1340110. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. The research was also made possible by the California State University STEM Teacher and Researcher Program, in partnership with Chevron (www.chevron.com), the National Marine Sanctuary Foundation (www.marinesanctuary.org), and the Jet Propulsion Laboratory.