PythonRobotics: a Python code collection of robotics algorithms

Atsushi Sakai
https://atsushisakai.github.io/
University of California, Berkeley
Berkeley, USA

Abstract

This paper describes an Open Source Software(OSS): PythonRobotics[12].This OSS is a Python code collection of robotics algorithms, especially focusing on autonomous navigation. It aims for beginners of robotics to understand basic ideas of each algorithm. The algorithms which is widely used in academia and industry and practical are selected. Each sample code only depends some standard modules on Python 3.x. In this paper, related works of this project, some key ideas about this OSS project, and brief structure of this repository are introduced. I also discuss future works of this project.

1 Introduction

In recent years, autonomous navigation technologies have received a great deal of attention in many fields such as autonomous driving[13] and other transportation systems, drone flight navigation etc.

2 Related works

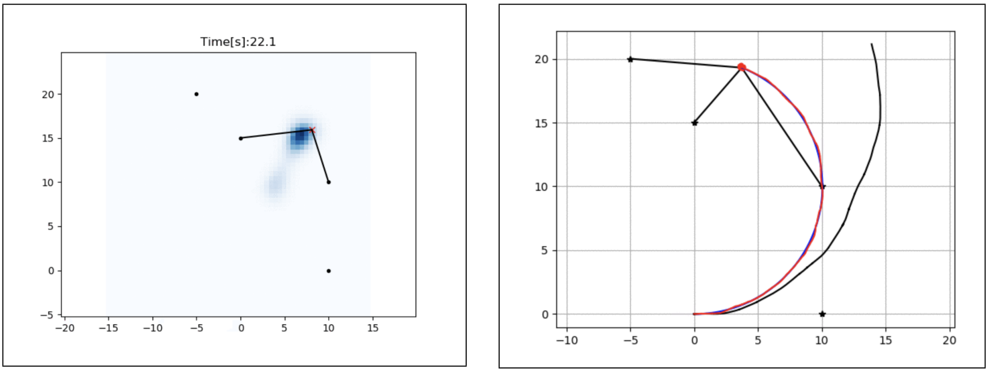
3 Philosophy

In this section, the philosophy of this project is described. This project based on three philosophies.

The first one is that the codes have to be easy to read for understanding each algorithm's basic idea. This project aims for beginners of robotics to understand basic ideas of each algorithm. Therefore, the code have to be easy to read and understand the algorithm. Programming language, Python[7] is adopted in this project because it has good code readability and it allows us to focus on algorithm itself. Python has great libraries for matrix operation, mathematical and scientific operation, and visualization. These libraries also allows us to focus on algorithm itself.

The second one is the algorithms which is widely used in academia and industry and practical are selected. For example, Kalman filters and particle filter for localization, grid mapping for mapping, dynamic programming based approaches and sampling based approaches for path planning, and optimal control based approach for path tracking.

The last philosophy is minimum dependency. It allows us to use the codes easily and to convert the codes to other programming languages such as C++, Java for practical usage. Each sample code only depends some modules on Python3 as bellows.



Histogram filter Particle filter

Figure 1: Localization simulation results

- numpy[4] for matrix operation
- scipy[8] for mathematics, science, and engineering computing
- matplotlib[3] for visualization
- pandas[6] for data analysis
- cvxpy[2] for convex optimization

These modules are OSS and could be used for free. This repository doesn't include any commercial software.

4 Repository structure

In this section, the brief structure of this project is described.

This repository has five directories which means five technical categories in autonomous navigation, Localization, Mapping, SLAM, Path planning, and Path tracking. Each directory includes several directories which has each sample code with different algorithms.

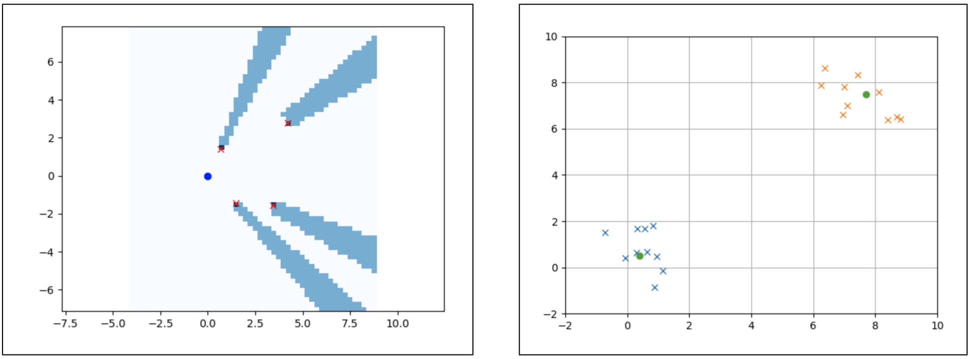
In the following subsections, some algorithms and some simulation examples described.

4.1 Localization

Localization is an ability of a robot to know it's position and orientation with sensors such as GNSS and IMU. In localization, Bayesian Filters such as Kalman filters, histogram filter, and particle filter are widely used[20]. This repository includes some sample codes using these algorithms. Fig.1 shows localization simulation results using histogram filter and particle filter.

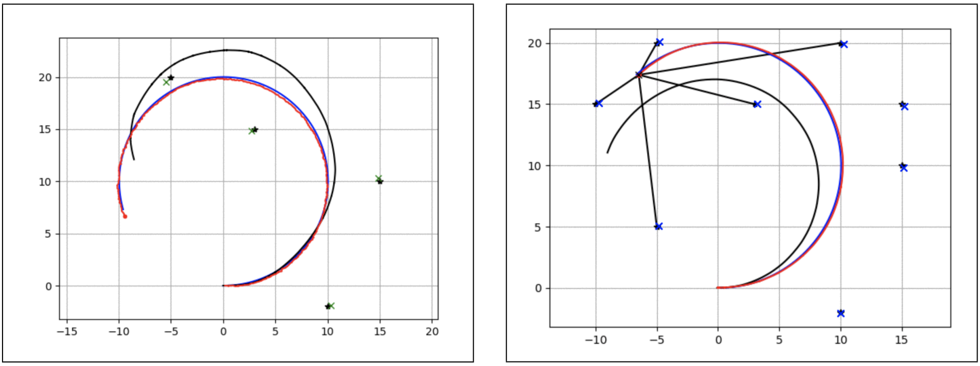
4.2 Mapping

Mapping is an ability of a robot to understand surroundings with sensors such as LIDAR and imaging sensor. Robots have to recognize obstacle positions and it' shape for obstacle avoidance. In mapping, Grid map, machine learning algorithms are widely used[20][9].



Grid mapping with 2D ray casting 2D object clustering with k-means algorithm

Figure 2: Mapping simulation results



Extended Kalman Filter based SLAM FastSLAM 2.0 based SLAM

Figure 3: SLAM simulation results

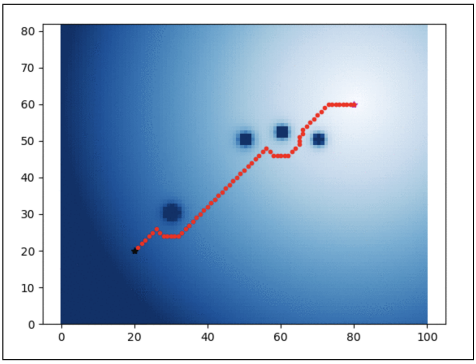
This repository includes some sample codes using these algorithms. Fig.2 shows mapping simulation results using Grid mapping with 2D ray casting and 2D object clustering with k-means algorithm.

4.3 SLAM

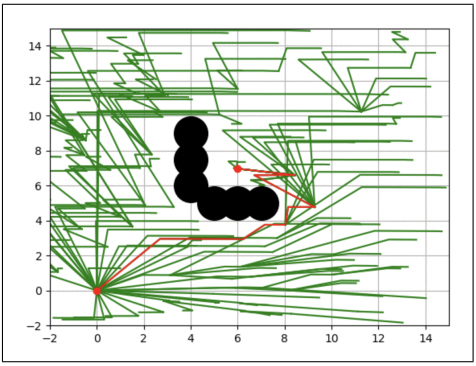
Simultaneous Localization and Mapping: SLAM is an ability of a robot to estimate the pose of a robot and the map of the environment at the same time. SLAM problem is hard to solve, because a map is needed for localization and a good localization is needed for mapping, which is a kind of chicken and egg problem. Popular SLAM solution methods include the extended Kalman filter, particle filter, and Fast SLAM algorithm[20]. This repository includes some sample codes using these algorithms. Fig.3 shows SLAM simulation results using extended kalman filter and results using FastSLAM2.0[20].

4.4 Path planning

Path planning is an ability of a robot to search feasible and efficient path to the goal. The path have to satisfy some constraints based on movement model and obstacle positions and optimize some objective function such as time to goal and distance to obstacle etc. In path plan-

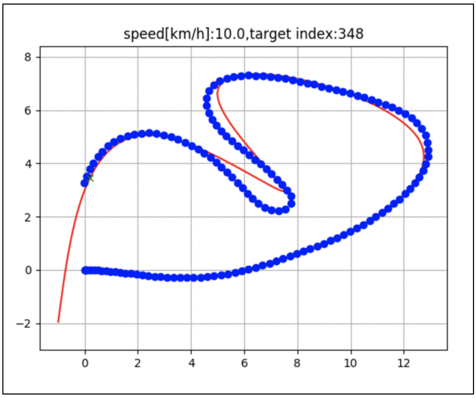


Potential Field path planning

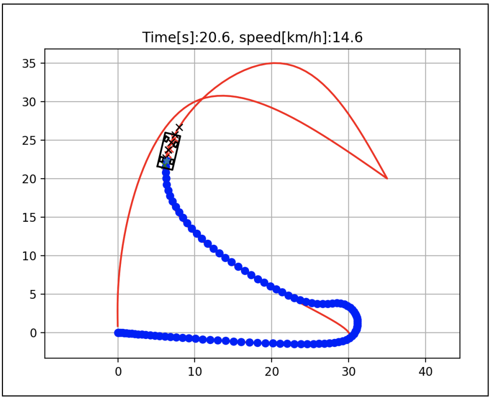


LQR-RRT* path planning[17]

Figure 4: Path planning simulation results



Rear wheel feedback steering control and PID speed control [15]



Iterative linear model predictive control

Figure 5: Path tracking simulation results

ning, dynamic programming based approaches and sampling based approaches are widely used[13]. This project includes some sample codes using these algorithms. Fig.5 shows potential field path planning and LQR-RRT* path planning[17].

4.5 Path tracking

Path tracking is an ability of a robot to follow the reference path from path planning algorithms. The role of the path tracking controller is to stabilize to the reference path or trajectory which has modeling error and other forms of uncertainty. In path tracking, feedback control techniques and optimization based control techniques are widely used[13]. This project includes some sample codes using these algorithms. Fig.5 shows Rear wheel feedback steering control and PID speed control and iterative linear model predictive path tracking control[17].

5 Conclusion and future work

In this paper, I introduced an OSS which is a Python code collection of robotics algorithms, especially for autonomous navigation. Related works of this project, some key ideas about this OSS project, and brief structure of this repository were described.

The future works of this project is as followed:

- Technical and mathematical documentation with Jupyter notebook[1].
- Simple image processing samples for autonomous navigation only using OpenCV[5].
- Simple multi-robots simulations.

If readers were interested in these future projects, contributions are welcome.

6 Acknowledgments

I appreciate all contributors: Daniel Ingram[14], Joe Dinius[11], Karan Chala[10], Antonin RAFFIN[18], and Alexis Paques[16]. This is my GitHub account[19]

References

[1] Project jupyter. http://jupyter.org/.

[2] Cvxpy. http://www.cvxpy.org/index.html.

[3] Matplotlib. https://matplotlib.org/.

[4] Numpy. http://www.numpy.org/.

[5] Opencv. https://opencv.org/.

[6] pandas. https://pandas.pydata.org/.

[7] Python. https://www.python.org.

[8] Scipy. https://www.scipy.org/.

[9] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

[10] Karan Chawla. Github account. https://github.com/karanchawla.

[11] Joe Dinius. Github account. https://github.com/jwdinius.

[12] Atsushi Sakai et al. Atsushisakai/pythonrobotics: Python sample codes for robotics algorithms. https://github.com/AtsushiSakai/PythonRobotics.

[13] David Gonzalez Bautista, JoshuÃľ PÃľrez, Vicente Milanes, and Fawzi Nashashibi. A review of motion planning techniques for automated vehicles. pages 1–11, 11 2015.

[14] Daniel Ingram. Github account. https://github.com/daniel-s-ingram.

[15] Brian Paden, Michal Cap, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. 2016.

[16] Alexis Paques. Github account. https://github.com/AlexisTM.

[17] Alejandro Perez, Robert Platt Jr, George Konidaris, Leslie P. Kaelbling, and Tomas Lozano-Perez. Lqr-rrt*: Optimal sampling-based motion planning with automatically derived extension heuristics. pages 2537–2542, 05 2012.

[18] Antonin RAFFIN. Github account. https://github.com/araffin.

[19] Atsushi Sakai. Github account. https://github.com/AtsushiSakai.

[20] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.