Trajectory planning pdf

Continue

Trajectory planning is moving from point A to point B while avoiding collisions over time. This can be computed in both discrete and continuous methods. Trajectory planning is a major area in robotics as it gives way to autonomous vehicles. Trajectory planning is sometimes referred to as motion planning and erroneously as path planning. Trajectory planning is distinct from path planning in that it is parametrized by time. Essentially trajectory planning encompasses path planning in addition to planning how to move based on velocity, time, and kinematics. Problem Constraints[edit | edit source] Holonomicity[edit | edit source] Holonomicity is the relationship between the controllable degrees of freedom of the robot and the total degrees of freedom of the robot. If the number of controllable degrees of freedom are greater than or equal to the total degrees of freedom a robot is said to be holonomic. By using a holonomic robot many movements are much easier to make and return to a past pose is much easier. A car would be non-holonomic, as it has no way to move laterally. This makes certain movements, such as parallel parking, difficult. An example of a holonomic vehicle would be one using mecanum wheels, such as the new Segway RMP.[1] Dynamic Environments[edit | edit source] In dynamic environments, such as the real world, many possible collision objects are not stationary. This makes trajectory planning more difficult as time is constantly changing and objects are moving. A robot cannot simply move backward in time as it might simply back away from a stationary collision. In addition to this many choices are completely irreversible due to terrain, such as moving off of a cliff. Concepts[edit | edit source] Concepts of Trajectory Planning Trajectory planning gives a path from a starting configuration S to a goal configuration G avoiding collisions in a 2D or 3D space. A configuration is the pose of a robot describing its position. Configuration Space C, is the set of all configurations. For instance, in two dimensions a robot's configuration would be described by coordinates (x, y) and angle θ. Whereas in three dimensions a robot's configuration would be described by coordinates (x, y, z) and angles (α, β, γ). Free space Cfree is the set of all configurations that are collision-free. Computing the shape of Cfree is not efficient, however, computing if a given configuration is a collision free is by simply using kinematics and collision detection from sensors. Target space is a linear subspace of free space which we want robot go there. In global motion planning, target space is observable by robot's sensors. However, in local motion planning, robot cannot observe the target space in some states. To solve problem, robot assume several virtual target space which is located in observable area (around robot). The virtual target space is called sub-goal.[2] Planning Algorithms[edit | edit source] Artificial Potential Field[edit | edit source] Example of a Potential Field Artificial Potential Field Planning places values over the map with the goal having the lowest(Highest) value raising(falling) the value depending on the distance from the goal. Obstacles are defined to have an incredibly high(low) value. The robot then simply moves to the lowest(highest) potential value adjacent to it, which should lead it to the goal. However this technique often gets trapped in local minima. Certain techniques can be used to avoid this, such as wavefront potential field planning. Artificial potential fields can be achieved by direct equation similar to electrostatic potential fields or can be drive by set of linguistic rules.[3] Sampling Based Planning[edit | edit source] Roadmap method is one sampling based planning method. First a sample of N configurations in C as milestones. Then a line PQ is formed between all milestones as long as the line PQ is completely in Cfree. Then graph search algorithms can be used to find a path from start to the goal. As N grows better solutions are found, however this increases computation time. Grid Based Planning[edit | edit source] Example of Grid Based Planning Grid Based planning overlays a grid on the map. Every configuration then corresponds with a grid pixel. The robot can move from one grid pixel to any adjacent grid pixels as long as that grid pixel is in Cfree. Then a search algorithm such as A* can be used to find a path to get from start to the goal. One potential tradeoff with this method is with a lower resolution grid(bigger pixels) the search will be faster, however it may miss paths through narrow spaces of Cfree. In addition as the resolution of the grid increases memory usage increases exponentially, therefore in large areas using another path planning algorithm may be necessary. Reward-Based Planning[edit | edit source] Reward-Based Algorithms assume that robot in each state (position and internal state include direction) can choose between different action (motion). However, the result of each action is not definite. In the other word, outcomes (displacement) are partly random and partly under the control of the robot. Robot gets positive reward when it reach to the target and get negative reward if collide with obstacle. These Algorithms try to find a path which maximized cumulative future rewards. Markov decision processes (MDPs) is a popular mathematical framework which is used in many of Reward-Based Algorithms. Advantage of MDPs over other Reward-Based Algorithms is that it generate optimal path. Disadvantage of MDPs is that it limit robot to choose from a finite set of action; Therefore, the path is not smooth (similar to Grid-based approaches). Fuzzy Markov decision processes (FDMPs)is an extension of MDPs which generate smooth path with using an fuzzy inference system.[2] [4][5][6][7][8] References[edit | edit source] Brady M. (1982) Trajectory planning. In Robot Motion:Planning and Control. M. Brady et al. (Eds.), MIT Press, Cambridge, Mass., pp. 221–2 Google Scholar Craig J.J. (1989) Introduction to Robotics:Mechanics and Control. 2nd ed., Addison-Wesley, Reading, Mass.MATH Google Scholar De Boor C. (1978) A Practical Guide to Splines. Springer-Verlag, New York.MATH CrossRef Google Scholar De Luca A. (1986) A Spline Generator for Robot Arms. Tech. rep. RAL 68, Rensselaer Polytechnic Institute, Department of Electrical, Computer, and Systems Engineering. Google Scholar Fu K.S., Gonzalez R.C., Lee C.S.G. (1987) Robotics: Control, Sensing, Vision, and Intelligence. McGraw-Hill, New York. Google Scholar Hollerbach J.M. (1984) Dynamic scaling of manipulator trajectories. ASME J. Dynamic Systems, Measurement, and Control. 106:102–106.MATH CrossRef Google Scholar Lin C.S., Chang P.R., Luh J.Y.S. (1983) Formulation and optimization of cubic polynomial joint trajectories for industrial robots. IEEE Trans. Automatic Control. 28:1066–1073.MATH CrossRef Google Scholar Paul R.P (1972) Modelling, Trajectory Calculation, and Servoing of a Computer Controlled Arm. Memo. AIM 177, Stanford Artificial Intelligence Laboratory. Google Scholar Paul R.P. (1979) Manipulator Cartesian path control. IEEE Trans. Systems, Man, and Cybernetics. 9:702–711.MATH CrossRef Google Scholar Paul R.P. (1981) Robot Manipulators: Mathematics, Programming, and Control. MIT Press, Cambridge, Mass. Google Scholar Taylor R.H. (1979) Planning and execution of straight line manipulator trajectories. IBM J. Research and Development. 23:424–436.CrossRef Google Scholar hbayerlein/uav_data_harvesting • • 1 Jul 2020 Autonomous deployment of unmanned aerial vehicles (UAVs) supporting next-generation communication networks requires efficient trajectory planning methods. mcapino/adpp-journal • 8 Sep 2014 In this paper we a) propose a revised version of prioritized planning and characterize the class of instances that are provably solvable by the algorithm and b) propose an asynchronous decentralized variant of prioritized planning, which maintains the desirable properties of the centralized version and in the same time exploits the distributed computational power of the individual robots, which in most situations allows to find the joint trajectories faster. mit-acl/faster • • 9 Jan 2020 The standard approaches that ensure safety by enforcing a "stop" condition in the free-known space can severely limit the speed of the vehicle, especially in situations where much of the world is unknown. qwerty35/swarm_simulator • 23 Sep 2019 This paper presents a new efficient algorithm which guarantees a solution for a class of multi-agent trajectory planning problems in obstacle-dense environments. LorenzoFederici/RobustTrajectoryDesignbyRL • • 19 Aug 2020 This paper investigates the use of Reinforcement Learning for the robust design of low-thrust interplanetary trajectories in presence of severe disturbances, modeled alternatively as Gaussian additive process noise, observation noise, control actuation errors on thrust magnitude and direction, and possibly multiple missed thrust events. andreadegiorgio/cylindrical-astar • 23 Sep 2020 Since path optimization is the core of any search algorithms, including A*, the 4D cylindrical grid provides for a search space that can embed further knowledge in form of cell properties, including the presence of obstacles and volumetric occupancy of the entire industrial robot body for obstacle avoidance applications. Page 2 ZJU-FAST-Lab/visPlanner • 11 Mar 2021 The visibility of targets determines performance and even success rate of various applications, such as active slam, exploration, and target tracking. wangxiao5791509/DeepMTA_PyTorch • • 18 Oct 2021 Recently, there has been tremendous progress in developing each individual module of the standard perception-planning robot autonomy pipeline, including detection, tracking, prediction of other agents' trajectories, and ego-agent trajectory planning. imanlab/deep_movement_primitives • • 14 Feb 2022 This paper presents a novel approach to manipulation path/trajectory planning called deep Movement Primitives that successfully generates the movements of a manipulator to reach a breast phantom and perform the palpation. OpenPerceptionX/TCP • 16 Jun 2022 The two branches are connected so that the control branch receives corresponding guidance from the trajectory branch at each time step. jasonarmitage-res/pm-vln • • 24 Jul 2022 In a busy city street, a pedestrian surrounded by distractions can pick out a single sign if it is relevant to their route.