Random Based Narrow Space Path Planning For Arm Manipulation With the Fixed Orientation Constraints

ZHENG Jingyi[1], LI En[1], LIANG Zize[1]

1. Institute of Automation, Chinese Academy of Sciences, Beijing 100190
E-mail: jingyi.zheng@ia.ac.cn

Abstract: Arm manipulation is a skill for robot working in cluttered environment. In this work, we focus on arm manipulation in narrow space with the end-effector to be a certain orientation. In this paper, we present a random, smooth path planning algorithm to avoid the obstacles considering every link of the arm, moving the arm from the starting point to the destination. Our planning algorithm can make the path similar in different planning process because the algorithm is end-point oriented. The simulation result validates the effectiveness of the algorithm.

Key Words: Arm manipulation, path planning, Obstacle Avoidance, Path Smoothness

1 Introduction

Manipulation in narrow space is a challenging task, especially for arms with several links. We must ensure the collision avoidance of every links.

Our focus is, in particular, on tasks where the robot is carrying an object with a certain orientation constraint in the whole process.

The motion planning algorithm is in Cartesian space. It is more vivid to construct the relationship between every link of the arm with the environment, thus it is easier to avoid obstacles.

The robot is required to carry a workpiece from the starting point to the end point, avoiding the obstacles. And the end effector holds a certain orientation in the process. The whole process is composed of several parts including the detection of the workpiece, motion planning and the control of the manipulation. In our work, what we focused on is the motion planning keeping the end effector to be a certain orientation, including calculating the path, collision checking and path smoothing.

Motion planning holding the initial roll and pitch of the workpiece is a constrained task. Because of the redundancy of the arm, there are many inverse kinematic solutions for one configuration. Thus, we add the constraints to the task to get the only set of joint angles. Also, we have to check the collision between every link and the obstacles as the internal constraints.

Robot collision-free path planning algorithm has not been solved well until now. Random methods, which includes the Probabilistic roadmap method(PRM) and Rapidly-exploring Random Trees (RRT)[1], are traditional ones. PRM consists of an off-line roadmap construction step and an on-line search step to generate the path. Probabilistic roadmap method, which is easy to realize, is effective in solving multi-degree of freedom path planning. However, PRM is not suitable in dynamic and rapidly changing environment and it is not good for nonholonomic path planning.

The RRT planner builds a tree on-line from the starting point to the end point. Some papers adopt bi-directional search[2] based on greedy collision detecting to plan the manipulator's motion in real time, which do not need to calculate the roadmap in advance. Randomized motion planners[3] were used for planning re-grasping actions. One of the approaches[4] to planning of dual-arm manipulation was demonstrated using a randomized planner. [5] proposed enhanced RRT algorithm added some heuristics to make the path compatible with the human operator. [6] presented a nonholonomic path planning and the optimal path planning within the RRT framework. RRT planner may be far from optimal. It lacks of certainty of the path planning due to the randomness, and the path is not smooth because of the node of the path is random. However, it is fast to plan a path.

In our method, we plan a path of smooth and comparatively steady with the random search method. We present a goal oriented random path planning algorithm to ensure the path for the same task is similar.

In the field of path smoothing, SRRT[7] method guarantees continuity of curvature considering both the obstacles in the environment and the internal constraints due to the kinematic or dynamic constraints. Dubins curves are one of the most popular method for path smoothing [8,9,10]. Clothoids[11,12] method and splines [13] have also been applied for path smoothing. Because of lacking of the differential constraints, such approaches are generally inefficient.

In this paper, we focus on the random based motion planning in the environment with obstacles and generating a smooth and steady path of every joint. In the whole process, we set a certain orientation constraint on the end-effector.

2 The Platform Introduction

2.1 The Overall Introduction

The arm(see Fig. 1) hungs on a horizontal slide, and it works on a plane when the horizontal axis has been set. The arm has three degrees of freedom and thus has a redundant degree of freedom.

As we can see from Fig.1, there are many obstacles in the environment. When the end-effector avoids the obstacles, it

*This work is supported by National High-tech R&D Program(863 Program) Under Grant 2013AA041002-1.

cannot guarantee the obstacle avoidance of the three links of the arm. Thus, it must execute the obstacle avoidance algorithm.

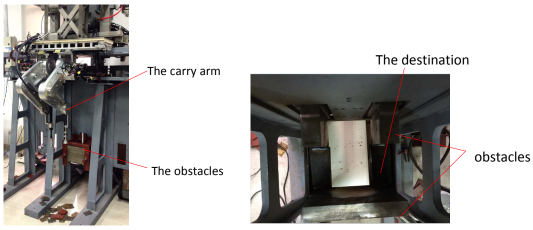


Fig. 1: The robot and the environment

2.2 The Constraints of the Path Planning

The arm is a redundancy arm. When the horizontal axis is fixed, the arm with three degrees of freedom works on a plane. Thus, there is one redundant degree of freedom. The redundant joint is parameterized by the other two joint angles. The redundant joint we choose is the end-effector.

In order to get the only inverse kinematics, the joint limits on the robot's joints are also taken into account. According to three constraints: the joint limits, the smaller the difference of the adjacent joints, the better and the certain number of the sum of three joint angles, we can get the single inverse kinematics solution for a given end-effector pose. These are the constraints for the arm's single configuration.

We use $(x, y, z, \theta_{yaw}, \theta_1)$ to represent the configuration of the end effector(see Fig. 2), where (x, y, z) is the position of the end-effector, θ_{yaw} is the angle of inclination and θ_1 is the parameter to hold the end-effector to be a certain orientation. The cubic in Fig. 2 represents the workpiece.

In order to keep the orientation of the end-effector, θ_1 is set to be the initial sum of the angles at the starting point. This is another constraint for the planning algorithm.

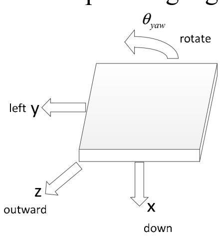


Fig. 2: Configuration of the end-effector

The above constraints are the internal constraints, and the external constraints are the collision checking.

3 Motion Planning Algorithm

The task of planning search is to find a path in the constructed environment, from the state of the current configuration of the arm, to any state of the desired location and orientation. In other words, we consider the problem of finding a motion that gets the workpiece from the current position to the destination.

In the following sections, we demonstrate all the components of the algorithm, including the environment construction, obstacle avoidance, the path generation and the path smoothing.

3.1 The Construction of the Environment

When planning path for manipulation, some represented the configuration space of the arm in joint space[3]. In our method of path planning, we present the environment in the Cartesian space in order to get more vivid collision checking. Because of the redundancy of one freedom, we construct the two dimension space to describe the obstacles and the arm on one plane (see Fig. 3). The two rectangles are the obstacles, and the three links are composed of the arm. The green point is the starting point, and the red point is the end point.

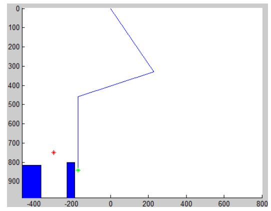


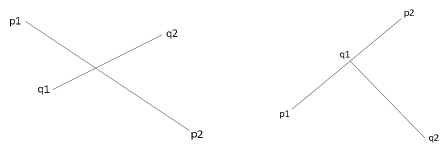
Fig. 3: The environment

3.2 The Obstacle Avoidance Algorithm

There are two rectangles, and three links of the arm in the environment. When the end-effector moves along the path, it must ensure that every link doesn't collide with the two rectangles. In this paper, we use two steps to check the collision[14].

Step1: preliminary collision checking. The line p1p2 represents one link of the arm, and line q1q2 represents one edge of the obstacles. Check if line p1p2 and line q1q2 satisfies the relationship in Fig.4(a) or (b). If it is yes, then go to step2; otherwise the two line segments do not collide(see Fig. 4).

Step2: line segments crossing test. If two line segments p1p2, q1q2 intersects with each other, the line p1q1 and q1p2 locate at the two sides of line segments q1q2(see Fig.5).



(a) (b)

Fig. 4: The rectangle collision test

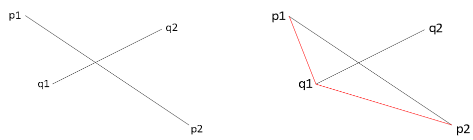


Fig. 5: The crossing test

3.3 Path Smoothing

In the step of path smoothing, we use cubic Bezier curve at every turning points. Cubic Bezier curve is a common method used in robotics where the motion of the arm should be smooth to avoid unnecessary wear. One of the control points is the turning point, and the other two control points are selected between the two turning points.

As we can see from Fig. 6, the red line is the generated path, and the green curve is path after smoothing. Point A is the turning point, which is a control point, point B and point C are two control points, too. Point C is calculated by the turning point A and turning point B.

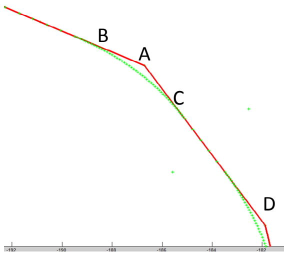


Fig. 6: Path smoothing

3.4 The Planning Algorithm

The motion planning algorithm we describe in this paper is to find a path from the starting point to the end point with obstacle avoidance. We use a random path planning algorithm to generate the feasible path. In our algorithm, the path is prone to the line from the starting point to the destination. And it can ensure the obstacle avoidance of every link, not only the end-effector. The algorithm is as follows:

Step1: set the start node, end node, the obstacles and the initial configuration of the arm for the planning problem. Set the step length from one node to the next node.

Step2: collision checking from the start node to the end node along the line between the two points. If there is no node which collides with the obstacles and the distance between the start node and the end node is less than the step, the path is the start node and the end node. If any node collides with the environment, go to step3.

Step3: generate new node, and check the collision. The tree node is the start node.

i. generate a rand point to be in the line between the starting point to the end point with a certain possibility or the rand point distributes in other places.

iii. generate a new point in the line between the rand point to the last tree node.

iv. collision checking between the rand point, the three links and the obstacles. If there is no collision, save the rand point as the tree node, its parent tree node's number and the configuration.

v. repeat the step i to iv until the tree node reaches the end node or the distance between the tree node and the end node is less than the step.

vi. add the end node to the tree node set to ensure the path can reach the destination.

Step4: According to the parent tree node, find the feasible path and every path node's configuration.

Step5: delete some nodes if the turning point angle is too small.

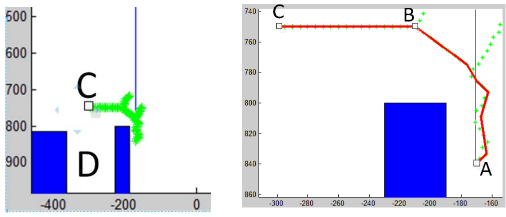
Step6: path smoothing using Bezier curve at every remaining turning point.

Step7: check the collision at the interpolation points, delete the collision points and calculate the configuration of every joint angle.

4 Simulation Results

In this section, we present simulation results which illustrate the performance of the proposed path planning algorithm.

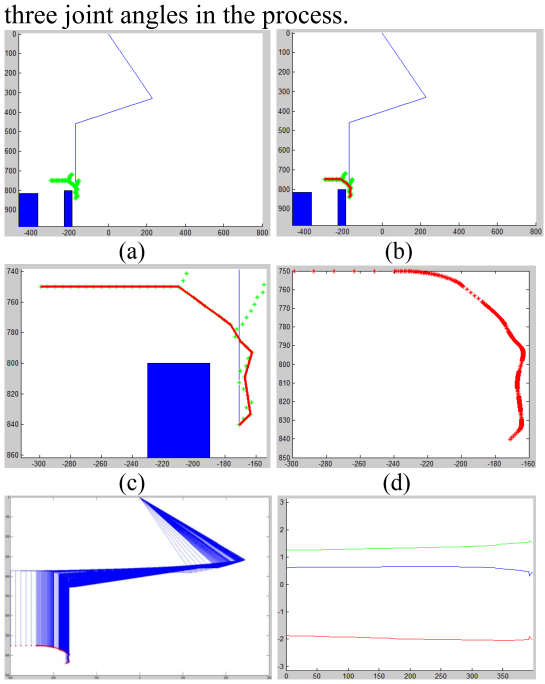
In order to accommodate to the narrow space path planning, we adopt the multi segments path planning algorithm. And this method also accelerates the searching algorithm. From Fig. 7(a), the working space is narrow, and the workpiece has the only way to put to the destination by move the end effector vertically from C to D. From Fig. 7(b), the whole planning is segmented into two parts, point A to point B, and point B to point C.



(a) (b)

Fig. 7: The multi segments path planning

We assume the end effector to be vertical. Thus, θ_1 is the initial sum of the three joints, and θ_{yaw} is zero. As can be seen from Fig. 8, (a) the green nodes represent the generated tree nodes. (b)The red nodes are the path. (c) is the detailed path. (d) is the result of path smoothness. (e) is the configuration of the arm in the process of the planning. (f) is the three joint angles in the process.

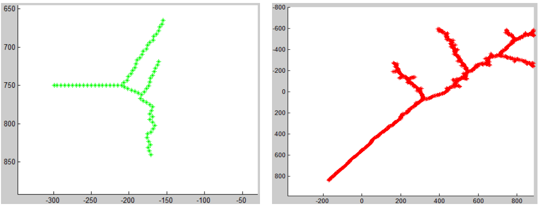


(a) (b) (c) (d) (e) (f)

Fig. 8: The simulation results

In the whole process, the configurations are smooth, and the joint angles are smooth. And the path avoids the obstacles successfully. Thus, our planning algorithm has a reasonable solution.

Compared to the traditional RRT, our method is easy to converge to the goal point, because our method is oriented to the goal point. In Fig. 9(a), our method only generates 76 tree nodes and it successfully finds the path from the starting point to the destination. Because RRT selects the minimum distance of the new node, it may stuck in the local minimum and fail to fulfill the path planning (see Fig. 9). It generates 823 tree nodes, however, the plan fails.



(a) (b)

Fig. 9: Our method and the traditional RRT

If the smoothing step is not used, the path is the red line (see Fig. 10). There are too many turning points, which is not suitable in the experiment. From Fig. 8(d), we can see a smooth path.

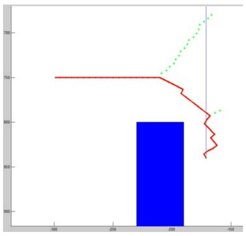
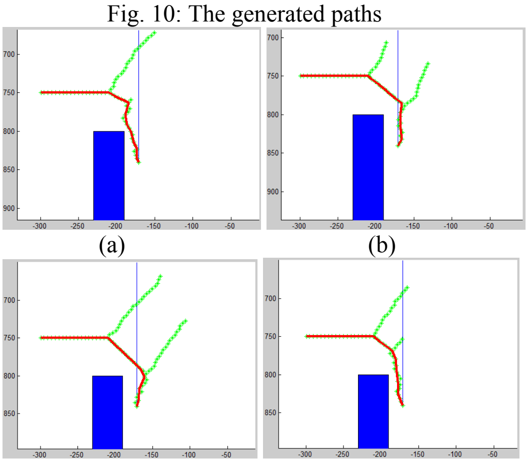


Fig. 10: The generated paths



(a) (b) (c) (d)

Fig. 11: The generated paths

Because of the random points distribute in the line between the starting point to the end point, the trend of the path goes to the similar direction (see Fig. 11). This advantage makes the motion of the arm stable.

5 Conclusions

In this paper, we present a random motion planning algorithm for workpiece manipulation with a certain end-effector to be a certain orientation. In our approach, we consider the problem of obstacle avoidance path planning. Because of the narrow space, we segment the path into two parts. Compared to the traditional RRT algorithm, our method generates the path oriented to the end point and the path is similar. In the process of obstacle avoidance, we not only consider the end effector, but also every link. Also, we use the cubic Bezier curve to make the path smooth. The simulation result validates the effectiveness of our algorithm with a steady, smooth and collision free path in narrow space.

References

[1] La Valle, S.M., Kuffner, J. Randomized kinodynamic planning. In Int. J. Robot. Res. 20(50), 378-400,2001.

[2] Kuffner, J., La Valle, S. RRT-connect: an efficient approach to single-query path planning. In IEEE International Conference on Robotics and Automation, 2000.

[3] Nikolaus Vahrenkamp, Dmitry Berenson, Tamim Asfour, James Kuffner, and Rudiger Dillmann. Humanoid motion planning for dual-arm manipulation and re-grasping tasks. In IEEE/RSJ International Conference on Intelligent Robots and Systems, October 2009.

[4] Yoshihito Koga and Jean-Claude Latombe. Experiments in dual-arm manipulation planning. In IEEE International Conference on Robotics and Automation, May 1992.

[5] Jorge Nieto, Emanuel Slawinski, Vicente Mut, Bernardo Wagner, Online path planning based on rapidly-exploring random trees, IEEE International Conference on Industry Technology,2010.

[6] K.Yang, SK Gan, J Huh, S Joo, Optimal spline-based RRT path planning using probabilistic map, International Conference on Control, Automation and System, 2014.

[7] Kwanjin Yang, Sangwoo Moon, Seunghoon Yoo. Spline-based RRT Path Planner for Non-Holonomic Robots, In Journal of Intelligent & Robotic Systems, 2014,73(1-4):763-782.

[8] Jacobs, P., Canny J. Planning smooth paths for mobile robots, In IEEE International Conference on Robotics and Automation,1989.

[9] Fraichard, T. Smooth trajectory planning for a car in a structured world. In IEEE International Conference on Robotics and Automation, 1991.

[10] Anderson, E., Beard, R., McLain, T. Real-time dynamic trajectory smoothing for unmanned air vehicles. In IEEE Trans. Control. Syst. Technol. 13(3), 471-477,2005.

[11] Fraichard, T., Scheuer, A. From Reeds and Shepp's to continuous-curvature paths. In IEEE Trans. Robot. 20(6),1025-1035(2004).

[12] Kito, T., Ota, J., Katsuiu, R., Mizuta, T., Arai, T., Ueyama, T.,Nishiyama, T. Smooth path planning by using visibility graph-like method. In IEEE International Conference on Robotics and Automation, 2003.

[13] Connors, J., Elkaim, G. Analysis of a spline based obstacle avoiding path planning algorithm. In IEEE 65th Vehicular Technology Conference,2007.

[14] De Berg, Mark, Cheong, Otfried, Van Kreveld, Marc, Computational Geometry: Algorighthms and Applications, Springer,2010.