Path Planning for A Special Robot Working in Narrow Space

Jingyi Zheng[1], Hao Lin[1], Guodong Yang[1],En Li[1],Guibin Bian[1],Zize Liang[1]

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

Abstract: This paper has designed a special robot working in a narrow space. Planning an appropriate path is one of the significant problems. In this paper, a mathematical expression of the G value considering obstacles in the environment based on A* algorithm in low-dimensionality is proposed. The algorithm is a approaching path planning algorithm in static, known environments. We plan the path of the end-effector after it moves near the obstacles. As only the end-effector changes drastically, it is safe for other joints working in the collision environment if the other joints are obtained using inverse kinematics. The simulation result validates the effectiveness of our algorithm.

Key Words: Heuristic-based planning algorithm, Obstacle free path, Approaching path planning

1 INTRODUCTION

Planning in human environment for manipulators is challenging and difficult, as the environment has certain obstacles that the manipulators must find the collision free path. Working in a narrow space for human beings is tedious and dangerous. Thus, a special robot(figure 1) is designed to fulfill the task with high efficiency and high safety.

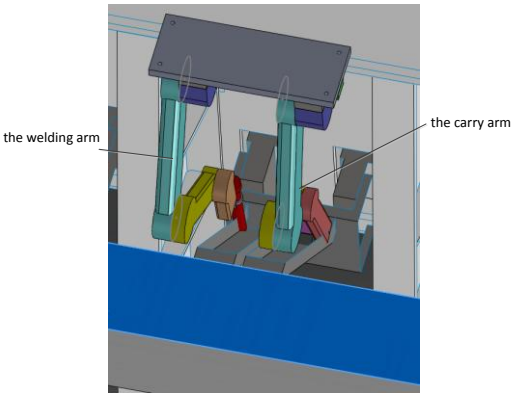

Fig1. the overview of the dual-arm welding robot

The whole execution of the task would be composed of multiple components including detection of the square workpiece, determination of the grasping point and path planning in the narrow space. In this paper, what we are concerned about is the task of path planning — the computation of collision free path from the starting point to the destination while maintaining the position of the end-effector.

The task that we would like to execute is that the end-effector of the carry arm moves from one place to the destination to carry the square workpiece with the fix orientation.

In the environment with obstacles, the artificial potential field algorithm[1] is a common method. This method is used in real-time obstacle avoidance. However, it is prone to stuck in local minima.

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

Eldershaw and Yim[2]have demonstrated a graph-based planner for legged vehicles based on PRM(Probabilistic Roadmaps), a randomized algorithm. However, the PRM suffers from the shortcoming that the generated plans, though feasible, may be far from optimal. Furthermore, the PRM-based planner is part of a greater planning hierarchy, which increases the complexity of the overall architecture.
Some algorithms have developed to minimize the cost of the solutions through optimization techniques[3][4]. The Covariant Hamiltonian Optimization and Motion Planning(CHOMP) algorithm[3] works by creating a naive initial trajectory from start to goal, and then uses a method similar to gradient descent to minimize the cost function. However, the gradient descent makes the approach vulnerable to local minima in the cost function. The Stochastic Trajectory Optimization for Motion Planning(STOMP) algorithm[4] relies on generating noisy trajectories to explore the space around a naive initial trajectory. It then combines these trajectories to produce an updated trajectory with low cost. A cost function which combines the obstacles and smoothness cost is optimized in each iteration. The scholastic nature of the approach makes it less vulnerable to the local minima in the cost function.
[5][6][7] provide methods which are computationally less expensive. However, they do not generalize well to action cost computation in maps with non-binary costs. Also, they are often limited to a particular shape or class of obstacles.
[8] designs the static fuzzy controller for the partially unknown environment and the simulation result illustrates its effectiveness.[9] finds the path in indoor environment based on Colony Optimization algorithm.[10] proposes a new genetic algorithm to overcome the weakness of the traditional genetic algorithm, and simulation result indicates that the method can reduce the scale of the population, minimize the searching scope and improve the velocity of the convergence.
Our approach to this problem is a graph search-based algorithm on the foundation of A* algorithm[11]. A* algorithm is simple, and efficient in most cases when the problem is in lower dimension. A* algorithm is a search-based algorithm. Search-based algorithms have the

advantages of good cost minimization and guarantees on completeness[12]. The two goals of the cost function is to minimize the length of the path and also maximize the distance to the obstacles along the path.

A* algorithm has a long history, and a lot of planning algorithm is based on it. D* algorithm is a kind of dynamic A* algorithm, which repairs paths to the robot's state in real time. Focussed D* algorithm[13] extends D*, and it focusses the repairs to significantly reduce the total time required for the initial path calculation and subsequent replanning operations. Min-Max LRTA*[14] proposes a real-time heuristic search method to solve planning tasks in non-deterministic domains efficiently. SetA*[15] combines the goal directed search of A* with the ability of BDDs to traverse an exponential number of states in polynomial time. Groundhog[16] uses A* algorithm to navigate in the mines, and the planning algorithm is in 3D C-Space maps.[17] presents an algorithm for planning goal-directed footstep navigation strategies through obstacle-filled environment and uneven ground. [18] shows a continuous, randomized version of A* along with an empirical analysis showing planning time convergence rates in the robotic manipulation domain. There is an anytime algorithm(ARA*)[19] to produce bounded suboptimal solutions in an anytime fashion. [20] develops a variable sized grid map, in which the size of the grid cells further from the robot increased. Only the area closest to the robot needs to be searched carefully, areas further from the robot can be searched coarsely. In [21] Honda ASIMO uses A* algorithm with three cost functions: location cost, step cost and expected cost. Sampled Composition A*[22] is used to solve low-dimensional trajectory planning. [23] proposes a solution to the problem of finding an effective yet admissible heuristic function for A* by precomputing a look-up table of solutions. [24] thought that A* were often ill-suited to solve kinodynamic motion planning problems. They proposed some methods for approximating state space obstacles to make search-based planning faster and safer. Focused A* Heuristics Recomputation[25] enhances A* search that can detect and correct large discrepancies between the heuristic cost-to-go estimate and the true cost function.

A major problem with A* and related algorithms had been that admissible heuristics result in examination of prohibitively large portions of the configuration space, whereas inflated heuristics cause significantly suboptimal behavior[26].Likhachev et al.[27] presents a framework of efficiently updating an A* search while smoothly reducing heuristic inflation, allowing resolution complete search in an anytime fashion on a broader variety of problems than previously computed.

The main contribution of this work is to propose a cost function considering the possibility of the collision with obstacles. Section 2 describes the design of the robot. Section 3 gives the description of the planning task. Section 4 shows the cost function and the simulation result. Finally, section 5 is the conclusion.

2 ROBOT STRUCTURE

One of the characteristics of the robot is that it is used in narrow space. The narrow space is shown in figure 2. The product box has four obstacles: the outer front side, the inner front side, the back side, and the bottom. Also, the ceiling is a obstacle.

The design of the robot is to imitate the process of our human beings in the factory. There are two arms with three joints to imitate the function of upper limb of our human beings.

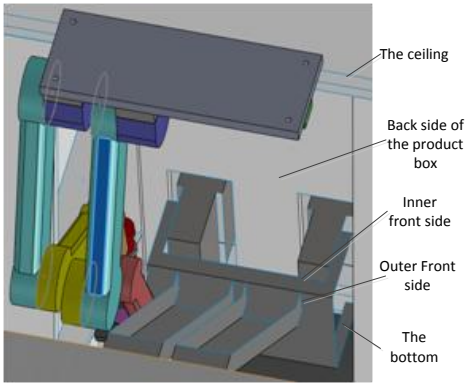

Fig2. the narrow space

The special robot is a kind of compact cylindrical coordinates robot (figure 1). There are two manipulators and one rail. Each arm moves around the rail on a plane, and also moves along the rail to reach another plane. One arm is for welding, and the other is for carrying and placing. Each arm is a Four-DOF arm, including one rail, three bars and three rotating links. The end of the welding arm is a welding gun; and the end of the carrying arm is suction cup.

3 DISCRIPTION OF THE PLANNING TASK

3.1. Problem Description

The problem is to move the carry arm(figure 1) to the destination and carry the square workpiece(figure 1), finding the collision free path when the end-effector approaches the obstacles. In the process, the angle of the end-effector is invariant because the suction plate must hold horizontally. In this paper, we only plan the path of the end-effector in Cartesian space after it moves near the front side of the product box. Because when the manipulator approaches the obstacles, only the end-effector is prone to collide with the obstacles in our environment.

3.2. Modeling of The Environment

We consider the path planning in the workspace, and what we are only concerned about is the position of the end-effector working in one plane.

For any path planning algorithm, the dimension of the planning is very important. Thus, we consider the fewer degrees of freedom to decrease the time complexity. The manipulator works in a plane when the position in rail is fixed. When the end-effector's attitude is fixed, the path planning is a two dimensional problem, which is a low-dimensional planning problem.

In figure 3, the starting point and the goal are denoted as ‘S’ and ’D’ respectively. There are five obstacles: obstacle1, obstacle 2, obstacle 3, obstacle 4 and obstacle 5. Obstacle 1 is the outer front side of the product box; obstacle 2 is the inner front side of the product box; obstacle 3 is the back side. Obstacle 4 is the ceiling, and obstacle 5 is the bottom. The scale of collision probability is obstacle 1, obstacle 2, obstacle 3, obstacle 4 and obstacle 5, from large to small.

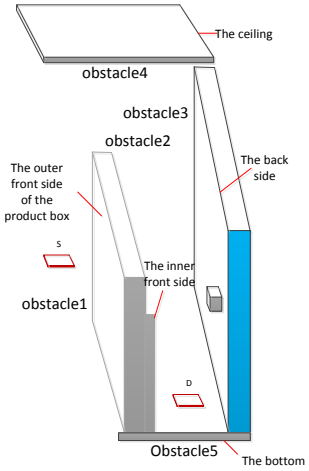

Fig 3. The scheme of path planning

3.3. Graph Construction

According to the model of the problem, we construct a graph with 20*20 squares on a plane(figure 4). The black nodes are the obstacles, the green node is the starting point, and the yellow node is the goal.

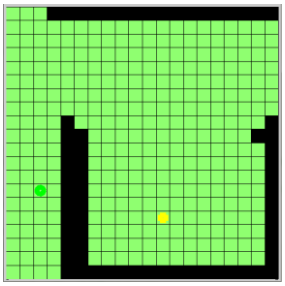

Fig4. The planning graph

4 PATH PLANNING ALGORITHM BASED ON A* ALGORITHM

Since the path planning problem is a two dimensional problem, the algorithm we propose is based on A* algorithm. We find the path from the four neighbours of a node, which is the left, right, up and down, not including the diagonal of the node. This is also called four-neighbor search[25].Our A* algorithm considers the distance between the end-effector and the obstacles.

4.1. Cost Function

As the path must have certain distance from the obstacles, we use a mathematical method to generate every G value to describe the distance to the obstacles. The dimension of the grid is m*n, and the number of the obstacles is q.

Assume that every checking node p_A belongs to the set of A, and every node p_O of the obstacles belongs to the set of O. The shortest distance between the checking node and one obstacle is as follows:

$$D_{\min}(A,O)=\min\{d(p_A,p_O)\}=\min(|x_i-x_{or}|+|y_j-y_{or}|) \quad (1)$$

in equation (1),

(x_i, y_j) is the position of every node p_A, i = 1,2,…,m, j = 1,2…,n;

(x_{or}, y_{or}) is the position of obstacles p_O, r = 1,2,…,q;

Thus, the expression of the G value is equation (2).

$$G(i,j)=\frac{C}{\omega_r * D_{\min}(A,O)} \quad (2)$$

Where,

ω_r is the weight, showing the importance of every obstacle, and the value is set according to the possibility of collision;

C is the adjustment parameter, which is used to adjust the G value bigger than 1.

4.2. Simulation

There are three experiments and the grid size is 20*20. Experiment 3 is our algorithm, and Experiment 1 and experiment 2 are used as comparison.

Experiment 1:A* that do not consider the distribution of obstacles. If we do not consider the probability of obstacle collision. The G value is 1, and the simulation result is in figure 5. The black line denotes the path of the end-effector. We can see that this method is not a good one because the path is near the obstacles.

Experiment 2:A* that uses empirical G value. We give G value of every node, especially G value is zero in ‘S’ and ‘D ’, and black nodes are the obstacles. Figure 6 gives the result.

Experiment3:A* that uses mathematical G value. Considering the obstacles, we use the mathematical expression of the G value, and the result is in figure 7.

In our experiment, we compare the duration time and number of steps using these algorithms in 20*20 space in Table 1. A* with mathematical expression of G value increases the number of searching time slightly in 20*20 space. However, our method decreases the searching steps. Also, setting the empirical value is a intractable and time-consuming task. Our algorithm using G(i,j) can decrease the workload.

The comparison between different sizes of the space is in figure 8 using G(i,j) and empirical value.

From figure 8, we can see that the number of steps of G value in mathematical expression is a little bit more than the empirical G value in 10*10 space and 12*12 space. However, when the space becomes larger, the steps of G value in mathematical expression is less than the steps of empirical G value.

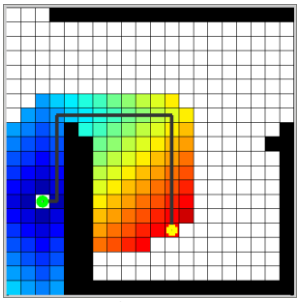

Fig 5. Result for that G value is 1

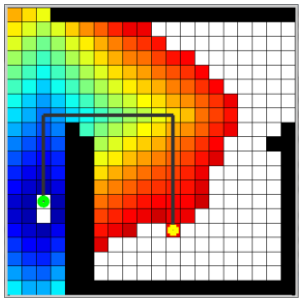

Fig 6.Empirical G value result

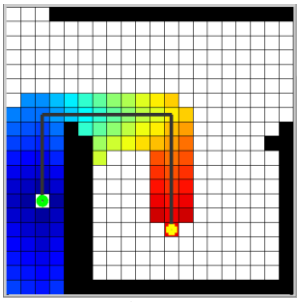

Fig 7. Result of mathematical G value

Table1. Comparison between three experiment

Method	Duration time(s)	Number of steps
A*without collision avoidance (G value is 1)	0.270679	106
A* with collision avoidance(empirical G value)	0.270876	191
A*with mathematical G value(G(i,j))	0.277456	76

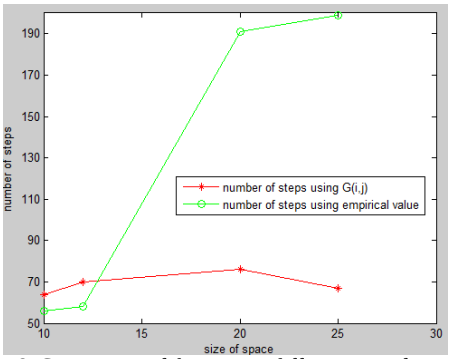

Fig 8. Comparison of the steps in different size of space

5 CONCLUSIONS

In this paper, we contribute to path planning algorithm for the end-effector. We use the mathematical expression of the G value(G(i,j)) in the cost function based on A* algorithm, finding a collision free path. G(i,j) is a precise expression of the collision, and releases the pressure of setting the G value empirically and it can also decreases the searching steps to some extent. The simulation result validates the effectiveness. The platform of the algorithm is a special robot for carrying and welding. In future work, we will use sensors to detect the position of the obstacles, and use path planning algorithm to find the collision free path.

REFERENCES

[1] Khatib.O, Real-Time obstacle Avoidance for Manipulators and Mobile Robots, IEEE International Conference on Robotics and Automation, May, 1985.

[2] C. Eldershaw, M. Yim, Motion planning of legged vehicles in an unstructured environment, Proceedings of the IEEE International Conference on Robotics and Automation, 2001.

[3] N. Ratliff, M. Zucker, J. A. D. Bagnell, and S. Srinivasa, Chomp:Gradient optimization techniques for efficient motion planning, IEEE International Conference on Robotics and Automation, 2009.

[4] M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, Stomp: Stochastic trajectory optimization for motion planning, International Conference on Robotics and Automation, 2011.

[5] M.D.C. Amezquita Benitez, K.K. Gupta, and B. Bhattacharya. Eodm-a novel representation for collision detection, IEEE International Conference on Robotics and Automation, 2000.

[6] Ming Chieh Lin, Efficient collision detection for animation and robotics. PhD thesis, University of California, Berkeley, 1993. Chair-Canny, John F.

[7] T. Lozano-Perez, Spatial planning: A configuration space approach, IEEE Transactions on Computers, C-32(2):108–120, Feb. 1983.

[8] Qing Li, Chao Zhang, Caiwei Han, Yinmei Xu, Yixin Yin, and Weicun Zhang, Path Planning Based on Fuzzy Logic Algorithm for Mobile Robots in Static Environment, the Chinese Control and Decision Conference,2013.

[9] Yufeng He, Qinghua Zeng, Jianye Liu, Guili Xu, Xiaoyi Deng, Path Planning for Indoor UAV Based on Ant Colony Optimization, the Chinese Control and Decision Conference,2013.

[10] Zhou Yongnian, Zheng Lifang, and Li Yongping, An Improved Genetic Algorithm for Mobile Robotic Path Planning, the Chinese Control and Decision Conference,2012.

[11] P.E. Hart, N.J.Nilsson, and B.Raphae, A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems, Science, and Cybernetics, SSC-4(2):100-107,1968.

[12] J. Pearl, Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-wesley, 1984.

[13] Anthony Stentz, The Focussed D* Algorithm for Real-time Replanning, Proceedings of the International Joint Conference on Artificial Intelligence, August, 1995.

[14] Sven Koenig, Reid Simmons, Solving Robot Navigation Problems with Initial Pose Unvertainty Using Real-time Heuristic Search, Proceedings of the International Conference on Artificial Intelligence Planning Systems,1998, pp. 144-153.

[15] Rune Jensen, Randy Bryant, and Manuela Veloso, SetA*: An efficient BDD-based heuristic search algorithm, *Proceedings of AAAI-2002*, August, 2002.

[16] David Ferguson, Aaron Christopher Morris, Dirk Haehnel, Christopher R. Baker, Zachary Omohundro, Carlos Reverte, Scott Thayer, William L. Whittaker, Chuck Whittaker, Wolfram Burgard, and Sebastian Thrun, An Autonomous Robotic System for Mapping Abandoned Mines, Advances

in Neural Information Processing Systems(NIPS 2003),2003.

[17] Joel Chestnutt, James Kuffner, Koichi Nishiwaki, and Satoshi Kagami, Planning Biped Navigation Strategies in Complex Environments, Proceedings of the International Conference on Humanoid Robots, October, 2003.

[18] Rosen Diankov, James Kuffner, Randomized Statistical Path Planning, Proceedings of IEEE/RSJ Internatioanl Conference on Robots and Systems, October, 2007.

[19] Maxim Likhachev, David Ferguson, Geoffrey Gordon, Anthony Stentz, and Sebastain Thrun, Anytime Dynamic A*: An Anytime, Replanning Algorithm, Proceedings of the International Conference on Automated Planning and Scheduling, June, 2005.

[20] Rachel Kirby, Reid Simmons, and Jodi Forlizzi, Variable Sized Grid Cells for Rapid Replanning in Dynamic Environments, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, September, 2009.

[21] Joel Chestnutt, Manfred Lau, Kong Man Cheung, James Kuffner, Jessica K Hodgins, and Takeo Kanade, Footstep Planning for the Honda ASIMO Humanoid, Proceedings of the IEEE International Conference on Robotics and Automation, April, 2005.

[22] Christopher Dellin, Siddhartha Srinivasa, A Framework for Extreme Locomotion Planning, IEEE International Conference on Robotics and Automation, May, 2012.

[23] Ross Alan Knepper, Alonzo Kelly, High Performance State Lattice Planning Using Heuristic Look-up Tables, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, October, 2006.

[24] Matthew Zucker, Approximating State-Space Obstacles for Non-Holonomic Motion Planning, tech. report CMU-RI-TR-06-27, Robotics Institute, Carnegie Mellon University, May, 2006.

[25] Matthew McNaughton, Christopher Urmson, FAHR:Focused A* Heuristic Recomputation, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009.

[26] Mather Zucker, Nathan Ratliff, Anca Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher Dellin, J.Andrew Bagnell, and Siddhartha Srinivasa, CHOMP: Covariant Hamiltonian Optimization for Motion Planning, International Journal of Robotics Research, May, 2013.

[27] M. Likhachev, G. Gordon, and S. Thrun, ARA*: Anytime a* with provable bounds on suboptimality, In Workshop on Neural Information Processing Systems, 2004.

[28] James J. Kuffner, Efficient Optimal Search of Euclidean-cost Grids and Lattices, IEEE/RSJ International Conference on Intelligent Robots and Systems, 2004.