13

Determination of Location and Path Planning Algorithms for Industrial Robots

Yung Ting and Ho-Chin Jar

1. Introduction

Before path planning, it is significant to determine the robot location, which very few researches have addressed in this topic. Determination of a suitable robot location is influential to prepare for the subsequent path search with better solution or even to ensure the possibility of finding a path. In particular, the environment with complex obstacles, the inspection is demanding. In this article, a method by use of the intersection theorem (Danielle & Mark, 2001) is proposed to determine the robot location.

Path planning has been studied with numerous researches on the topics of minimum time, minimum energy, and obstacle avoidance, etc. Obstacle avoidance is probably the most distinguished one investigated for many application purposes. Distance maps is one of the earlier method to divide the space by grids with equal distance. The obstacle is mapped in this 2D diagram. The rest of the area is considered to be passable and marked with integer numbers, which indicates the distance to the obstacle (Latombe, 1991; Pobil et al., 1992; Jarvis, 1993). Wave expansion method is derived based on the distance maps. It starts to mark the passable nodes with sequential integer numbers from the selected initial position to expand outward, and then begins the path search at the final position (Barraquand et al., 1992; Ralli & Hirzinger, 1994).

Configuration space concept is proposed by (Lozano-Perez, 1987; Banski, 1996; Red & Truong-Cao, 1996). It attempts to illustrate the robot manipulation geometry in terms of the joint space. For an n degree-of-freedom robot, there is n dimensional vector in the configuration space, where the collision occurs in the workspace can be expressed.

In this study, three path-planning methods, the neighboring search method, the depth-first search method, and the extensile search method, are developed. The path searching capability, manipulation steps and time are discussed with comparison.

A practical automobile baggage trunk welding process in association with an industrial robot, ABB IRB1400, is selected as an example to be investigated with simulation on the Robot Studio S4-lite software. The proposed extensile

Source: Industrial-Robotics-Theory-Modelling-Control, ISBN 3-86611-285-8, pp. 964, ARS/plV, Germany, December 2006, Edited by: Sam Cubero

neighboring search method, in particular, is more reliable to find a path and shows autonomous capability of reducing manipulation steps.

2. Determination of Robot Location

Inappropriate location of the robot may cause inconvenient operation, or even unexpected damage. Especially, it may provide no solution for path planning when dealing with complex obstacles in the working environment. Therefore, investigating the feasible location area of the robot in the Cartesian coordinate system before path planning is the primary task.

The shapes of miscellaneous obstacles are difficult to express by simple mathematical description. An easy way is to segment the obstacle into analyzable geometric shapes such as triangle or rectangle. The unavoidable error due to this approximation approach is tolerable because it does not affect the determination of robot location and the following path planning obviously. For example, the top view of an automobile baggage trunk in 2D is shown in Figure 1. The solid line represents the boundary of the baggage trunk. The segmented rectangular areas bounded by the dashed lines replace the original practical trunk shapes. For instance, the robot needs to pass the four (A,B,C,D) working points. The possible location area to cover each of the passing points (A,B,C,D) is represented R_A, R_B, R_C, and R_D, respectively. Via inspection on the intersection with the obstruction area of the obstacle, the possible location area is obtained and can be mathematically described by

$$R = (R_A \bigcap \overline{O}) \bigcap (R_B \bigcap \overline{O}) \tag{1}$$

where "O" represents the obstruction area of one of the segmented rectangular obstacles, and "R" represents the inspected possible region that the robot can be located for the working points A and B. To check each rectangular shape in sequence, the possible location areas are searched, so that the robot location is determined.

Similarly, the passable area for considering all of the segmented obstacles (O_1, O_2, …, O_n) etc., are defined as

$$\begin{aligned} R = {} & [(R_A \bigcap \overline{O_1}) \bigcap (R_B \bigcap \overline{O_1})] \bigcap [(R_A \bigcap \overline{O_2}) \bigcap \\ & (R_B \bigcap \overline{O_2})] \bigcap \cdots \bigcap [(R_A \bigcap \overline{O_n}) \bigcap (R_B \bigcap \overline{O_n})] \end{aligned} \tag{2}$$

Concerning all of the desired working passing points (A,B,C,D, …), the possible location region R is inspected with the same process in (2). In case that the intersection area is none, that is, R =[ ], then, there may not have suitable robot

location for the subsequent path planning. On the other hand, large space of R may provide better solution for searching a suitable path.
As shown in Figure 1, each of the fours bigger circles represents the possible location area while the robot stretches out to pass each of the points A,B,C,D, respectively. Similarly, the other four smaller circles represent the possible location area while the robot withdraws to pass the points A,B,C,D, respectively. Similar tests are also carried on in the other Y-Z and Z-X planes, which are not addressed in detail here. It is concluded that the shaded area is the possible region for robot location. Via numerical analysis, the point E in Figure 1 is selected as the robot location.

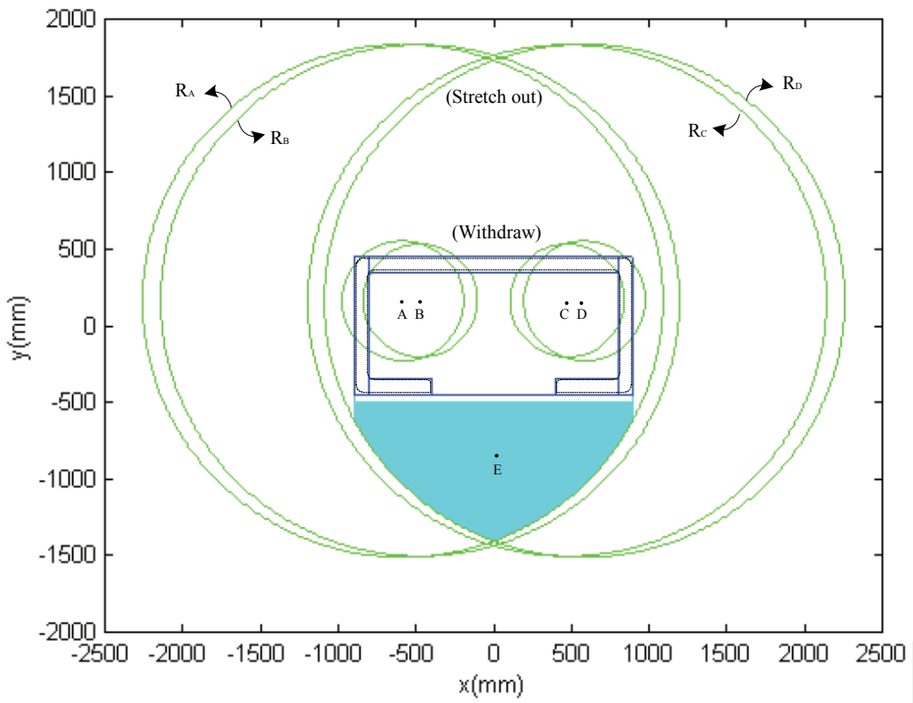


Figure 1. Segmented Obstacles and Robot Location

3. Collision Inspection

The robot and the obstacle need to be defined in the geometric space. Since the last three joints inherited with limited manipulation space for most of the six DOF industrial robots, the robot structure can be lumped to express by the first three arms of the elbow (Ting et al., 2002).
The robot and the obstacle can be described in a discretized space by the distance maps method. In this 2D map, the perpendicular grids intersect on the nodes and cut the space into numerous equivalent squares (Pobil et al., 1992).

Configuration space method is a tool to transfer the robot manipulation geometry into the joint space so that robot collision inspection can be achieved in the same space. The configuration space is established by the joint variables, which is tantamount to the dimension of degree-of-freedom of the robot. Thus, it is convenient to transform the robot and the obstacle structure in the distance maps into the configuration space for further investigation.
According to the robot shape, the boundary function Plane (P) is used to check an arbitrary point P whether the collision appears (Ting et al., 2002). Via collision inspection, the nodes of the obstacle and the unreachable region of the robot in the configuration space are marked –1, and those of the movable range are marked 0.

4. Path Planning

Wave expansion method provides an appropriate approach for path planning (Pobil et al., 1992; Ting et al., 2002). Via the previous collision inspection results, the passable area marked 0 can be expanded outward either from the initial position or the final position and marked with a specified integer number in the configuration space. The number is marked with 1 at the chosen start node, and gradually increased to n at the end node (Ting et al., 2002). For example, the passable nodes are marked with numbers shown in Figure 2.

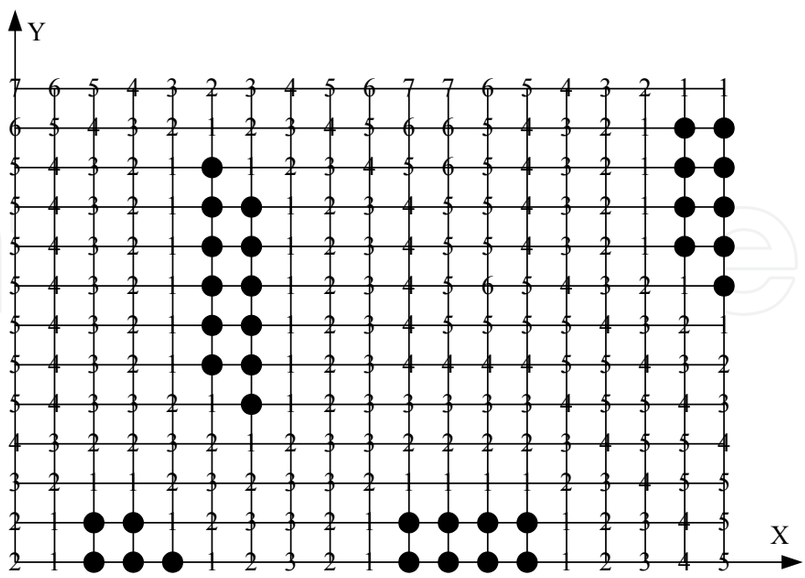


Figure 1. Marked numbers on the nodes

4.1 Neighboring Search Method

While searching for suitable path, it is suggested to start from the node of final position with the largest marked number since there is no guarantee to have a solution to start on the initial position. All of the passable nodes with one unit distance around the start node, 8 nodes at most for two dimension and 26 nodes at most for three dimensions, are checked to see whether there exists at least one passable neighboring node. If the marked number of the start node is n, the node around it with the smallest number, for example, n-1 or n-2, is the desired node as the next passable destination. Then, from that node, the subsequent search is continued until the marked number of the desired passable node is 1. To conjoin these searched nodes, thus determines the path. In case that the chosen passable node does not eventually search a path successfully, it needs to switch to another node with smaller number, and then continues the path search from there again.

4.2 Depth-first Search Method

The depth-first method is derived based upon the data structure concept of the computer system. It starts at the node of the final position with the largest marked integer number n, and searched the neighboring node whose marked number (n–1) must be smaller than it by one. For instance, the searched path in the tree structure is illustrated in Figure 3. The white nodes are the likely nodes to pass through, and the black nodes indicate the obstructive area or the robot unreachable area. The start node means the initial position to begin wave expansion, and the end node means the final position to begin the marked number. The Null indicates the termination of search with no solution.

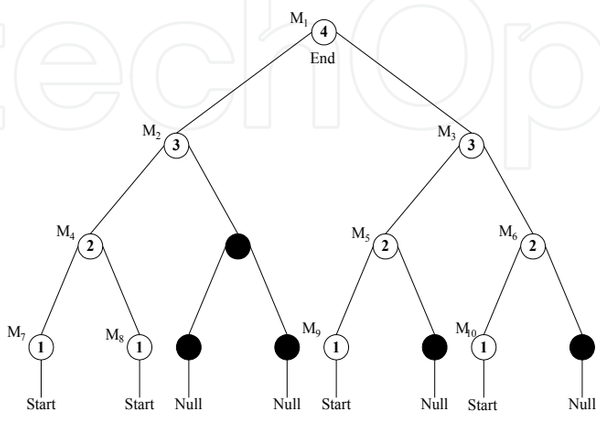


Figure 3. Depth-first search method

As the depth-first search method is used in data structure, a path is obtained from the top to the bottom and from left to right on the tree structure (Tarjan, 1972; Ting & Lei, 1999). In this example, the path searching process is followed by

$M_1 \rightarrow M_2 \rightarrow M_4 \rightarrow M_7 \rightarrow M_8 \rightarrow M_3 \rightarrow M_5 \rightarrow M_9 \rightarrow M_6 \rightarrow M_{10}$

Hence, several likely paths are concluded as below.

(1) $M_1 \rightarrow M_2 \rightarrow M_4 \rightarrow M_7$ (2) $M_1 \rightarrow M_2 \rightarrow M_4 \rightarrow M_8$

(3) $M_1 \rightarrow M_3 \rightarrow M_5 \rightarrow M_9$ (4) $M_1 \rightarrow M_3 \rightarrow M_5 \rightarrow M_{10}$

To reduce the robot manipulation steps, an easy way is to merge the nodes in the same manipulation direction (Lei, 1999).

4.3 Extensile Neighboring Search Method

Via many experimental testing, the neighboring search method does not obtain a good path in comparison with the depth-first search method. It is interesting to dig out why the former one searching 26 neighboring nodes along various directions cannot obtain better outcome than the latter one searching only 6 neighboring nodes. It is attempted to develop an extensile neighboring search method that outperforms the depth-first search method by solving the defects of the neighboring search method as described below.

While using the wave expansion method, the start and the end nodes are selected to be either the initial or the final positions. Once the initial position is chosen, the wave expansion begins at that nodal point. An investigation is carried out to examine whether there is different wave expansion solution by exchanging the initial and the final destinations. As illustrated in Figure 4, two paths, the solid and the dotted lines, are obtained by interchanging the initial with the final destinations. The bold line represents the passable region of both cases. It is obviously to see that the searched two paths are not the same.

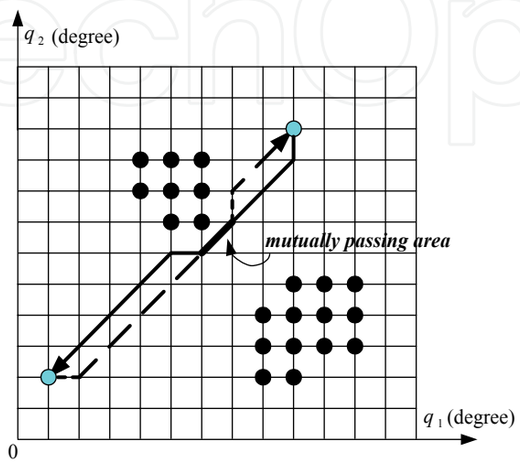


Figure 4. Two searched paths by exchanging the initial with the final positions

Therefore, the selection of the start node for wave expansion and the end point for backward numbering is important, and may seriously affect the search result. This situation is more obvious in the 3D environment for the search neighboring nodes are increased from 8 (2D) to 26 directions. To double-check on the searched path by interchanging the initial with the final position is necessary.

In Figure 4, a solid line is searched on condition that the furthest left node is selected as the initial position to start wave expansion process, and the furthest right node is the final position to begin the path search. On contrary, the initial and the final positions are exchanged to obtain the dotted path. It is seen that the difference of the searched paths between the two cases appears two parallelogram areas. In these areas, it is for sure that no collision occurs. Thus, it is likely to merge the two searched paths into one as shown in Figure 5.

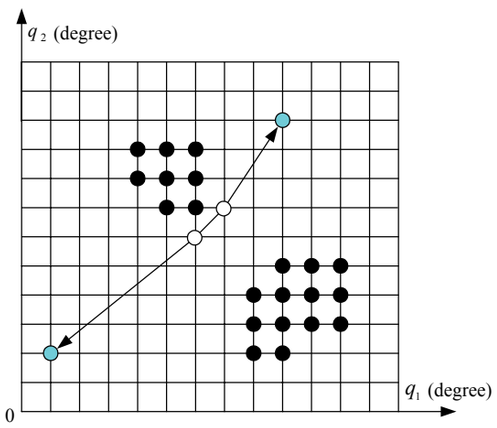


Figure 6. Merge of the two paths into one solution

While using the neighboring search method, the robot path must pass the grid nodes of the same square so that a saw tooth type of path is likely obtained. Thus, it limits the searched path no matter how small the grid is planned. This defect may result in the searched path is not flexible and the manipulation time is elongated because of passing more nodes. This phenomenon is illustrated in Figure 6.

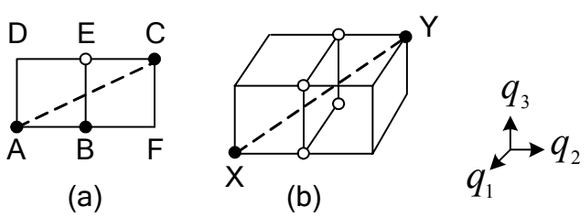


Figure 6. Reduction of path in 2D and 3D

On the assumption that the path $A \rightarrow B \rightarrow C$ shown in Figure 6(a) is the original searched path, it can be reduced to $A \rightarrow C$. It is not solvable by the neighboring search method since the searched path has to move along the grid nodes in the range of a unit square. The nodes D and E on the unit square ABED are considered to inspect collision (Ting et al., 2002). Since the triangle ΔABE includes the dotted line across the parallelogram, it is necessary to check only the node E. To extend to the 3D environment shown in Figure 6(b), the path $X \rightarrow Y$ is desired, and it only needs to check the area is enveloped in the grid nodes includes the dotted line. This method is useful to reduce the path. For example, once the circled grid nodes shown in Figure 7 are ensured no collision occurs, then the path is defined to be $P \rightarrow Q$, which thus saves many manipulation steps.

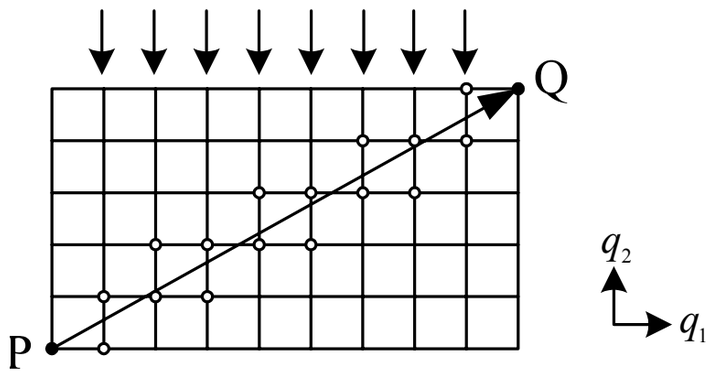


Figure 7. Further reduction of searched path by checking the circled nodes

It is quite difficult to intuitively choose the grid nodes for checking in the 3D space. A recursive program is developed to systematically inspect the collision of the nodes around the unit square where the considered path intersects. The algorithm is described as below.

Recursive Algorithm:

Assuming the k^{th} grid node is located at (a,b,c), and the $(k+2)^{th}$ grid point is at (x,y,z). Provided that the distance along the X-Y-Z direction is defined with $|x-a| \geq |y-b| \geq |z-c|$, and the grid unit distance is designated with d degree. Let the grid number N is defined as $N = |x-a|/d$, and the checked nodes (p,q,r) is defined as $(p,q,r)=(x-a,y-b,z-c)/N$, and the plus or minus symbol of the checked nodes is defined by $(u,v,w)=sgn(p,q,r)$, respectively. By use of the above definitions, three conditions are defined as below.

1. While |p|=|q|=|r|, then inspects the nodes at (a+du, b+dv, c+dw),…, (a+(N-1)du, b+(N-1)dv, c+(N-1)dw).
2. While |p|=|q|≠|r|, then inspects nodes at (a+du, b+dv, c), (a+du, b+dv, c+dw),…, (a+(N-1)du, b+(N-1)dv, c+(N-2)dw), (a+(N-1)du, b+(N-1)dv, c+(N-1)dw).
3. While |p|≠|q|≠|r|, then inspects (a+du, b, c), (a+du, b+dv, c), (a+du, b, c+dw), (a+du, b+dv, c+dw),…, (a+(N-1)du, b+(N-2)dv, c+(N-2)dw), (a+(N-1)du, b+(N-1)dv, c+(N-2)dw), (a+(N-1)du, b+(N-2)dv, c+(N-1)dw), (a+(N-1)du, b+(N-1)dv, c+(N-1)dw).

After the inspection procedures described above, the original path k→k+1→k+2 can be simplified to be k→k+2 on condition that there is no collision occurs at the checked nodes around the unit square. Thus, if the path is searched to pass n nodes, it may be reduced at most (n-1) nodes.

The extensile neighboring search method is able to obtain a path near the obstacle. It is advantageous to deal with a complicated environment such as complex obstacles or limited passable area. Also, unlike the neighboring and the depth-first search methods need intuitive decision to reduce the manipulation steps along the same moving direction, it is autonomous to complete path search by the developed recursive algorithm. Moreover, there may be a situation that the depth-first search method is infeasible. This method inspects the up, down, left, and right directions in 2D. For instance, the obstacle is transferred into the configuration space as depicted in Figure 8.

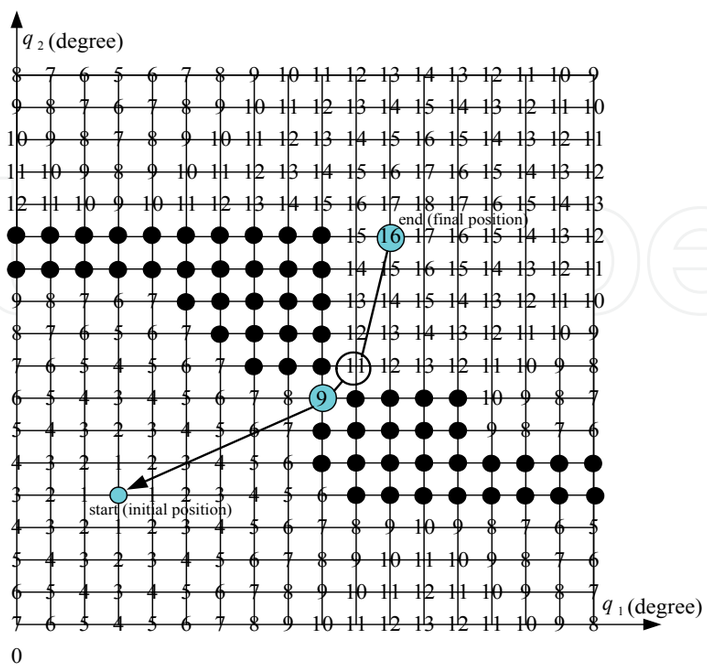


Figure 8. Unsuccessful path planning by depth-first search

It is seen that there is no subsequent node to continue with when the path search meets the grid number 11. That is, the depth-first search method is not workable if the grid node of number (n-1), number 10 in this example, does not neighbor to the current searched node. On the other hand, the extensile neighboring search method checks the surrounding nodes with number (n-2), number 9 in this example, so that it is able to find a path.

5. Manipulation Time

As a matter of fact, the manipulation time is more concerned than the manipulation steps. In general, any type of motion profile can be planned by motion programming method (Tesar & Matthew, 1976). In this study, for example, according to the driving motor of ABB IRB-1400 industrial robot (ABB, 1996), a polynomial form to express the motion profile is given by

$$S(t)=C_5t^5+ C_4t^4+ C_3t^3+ C_2t^2+ C_1t^1+ C_0 \quad (3)$$

Two cases of motion profiles are presented in Figure 9. In Figure 9(a), the initial time is at t_0. The time between t_0 and t_1 is the time needs to arrive at the maximum speed. The time between t_1 and t_2 is the time to manipulate with the maximum speed. The time between t_2 and t_3 is the time needs to reduce the speed to zero. In case the motor of the robot does not reach its maximum speed due to too short distance of the manipulation step, the motion profile is illustrated in Figure 9(b). Once the coefficients in (3) are defined, the manipulation time following the designed motion profile is then computed for both cases. It is noted that the distance of each piece of straight line in the joint space formed by the passable nodes may be different. For example, some passable nodes are connected to form a straight line. That is, more manipulation steps are accompanied with. Therefore, the motion profile planned for each piece of straight line is different. The entire manipulation time is the sum of the manipulation time of each piece of the passing straight line.

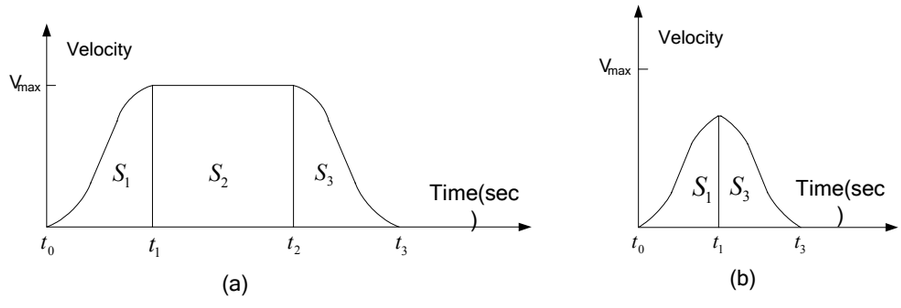


Figure 9. Motion profile for different situation

6. Simulation and Results

An ABB IRB-1400 industrial robot with 6 DOF is selected as an example to investigate the path planning. The maximum motor speed of the robot is limited to be 30 cm/sec. The shape of the first three links is assumed to be cylinder with radius r_1, r_2, r_3 and corresponding link length l_1, l_2, l_3. The last three joints of the wrist of most the industrial robots have limited manipulation range. For example, the workspace of the fifth joint of ABB IRB-1400 industrial robot is negligible because of short link length and small joint working area, and ineffective to the rest two joints. The radius r_3 of the third link is intended to expand to include the workspace of the fifth link. Hence, the simplified first three joints with joint variables (q_1, q_2, q_3) of the elbow are considered for the path planning (Ting et al., 2002).

A practical automobile baggage trunk is depicted with simplified picture shown in Figure 10. The dimension of this obstacle is about 900mm×1800mm× 500mm. The designated passing points are assumed to be A, B, C, D, E, F, where points A and F are the start and the end positions, respectively, and the rest of them are the welding positions. These points mapped into the configuration space in terms of the three joints are (90^o, 0^o, 0^o), (30^o, 20^o, 10^o), (25^o, 15^o, 10^o), (-25^o, 15^o, 10^o), (-30^o, 20^o, 10^o), and (-90^o, 0^o, 0^o), respectively, in reference to the robot base.

In this example, it is quite uneasy to find a solution by robot teaching method from many practical experimental tests. The searched path is critical and inefficient for the robot. Especially, inappropriate location of the robot may cause the path planning unlikely. The obstacles are arbitrarily segmented into several rectangular pieces with a 2D top view shown in Figure 2. The location of the robot is investigated by (2) with all the passed points. According to the offline computation, the robot is located at (0, -850) in reference to the center of the obstacle (0,0) in the X-Y Cartesian coordinate.

The results of path planning via the neighboring, the depth-first and the extensile neighboring methods are presented in Figures 11, 12 and 13, respectively. Though, these methods are able to search a path, the first one needs 33 manipulation steps, the second one needs 16 manipulation steps, and the third one needs 13 steps. In terms of the manipulation time by (3), the first one spends 18.7661 seconds, the second one spends 9.6716 seconds, and the third one spends 8.5054 seconds. It is obvious that the extensile neighboring method saves more manipulation steps, even better than the depth-first search method.

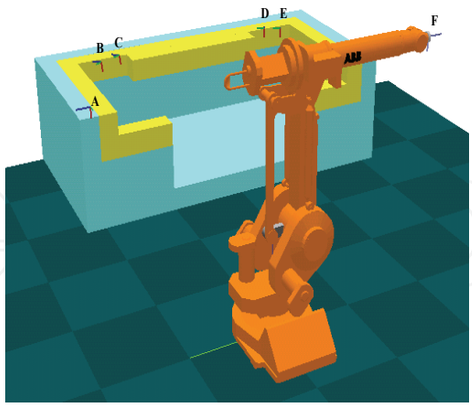


Figure 10. Diagram of automobile baggage trunk

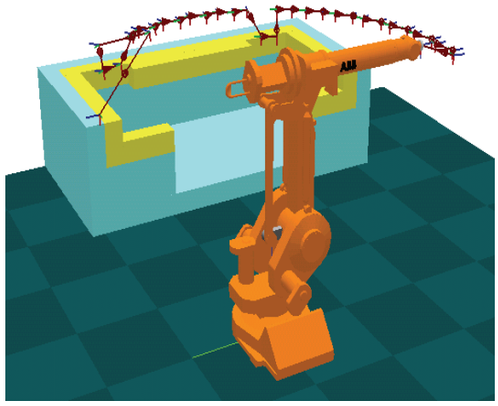


Figure 11. Path planning by neighboring search

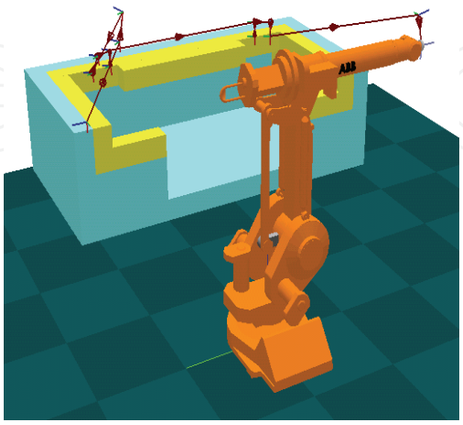


Figure 12. Path planning by depth-first search

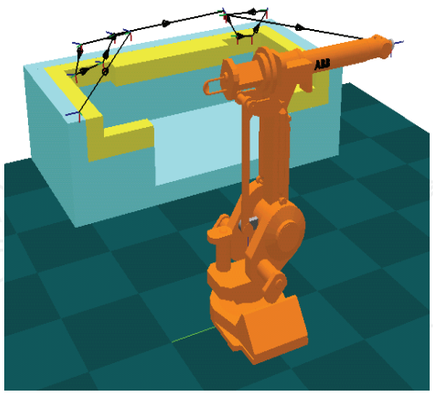


Figure 13. Path planning by extensile neighboring search

7. Conclusion

Investigation of robot location is a necessary procedure previous to path planning. Checking the passable region *R* by (2) is significant that ensures whether the later path planning is feasible. As to the measure of R in terms of a performance index, large space of *R* may provide more selections of robot location.

Via the wave expansion method, the passable nodes are marked with numbers. Three methods are proposed for path planning of the industrial robots. The plain neighboring search method is expected to find the path with more manipulation steps. The depth-first method can search a path with fewer manipulation steps, however, it may fail when there does not exist a neighboring node with marked number fewer than the number of the current node. Extensile neighboring search method provides a further reduction of manipulation steps. In general, the searched path needs fewer manipulation steps implies less manipulation time. Also, it is convenient to use the recursive search algorithm to find a path with fewer manipulation steps without the need of intuitively merging the path in the same direction either by the neighboring or the depth-first search methods (Ting et al., 2002). This method, above all, has better performance on searching a path successfully.

A practical automobile baggage trunk is studied to show the capability of determination of robot location and path planning with the developed methods. The extensile neighboring search method not only can trace 26 directions in 3D space, but also can autonomously reduce manipulation steps; therefore, it is an ideal candidate for path planning of industrial robots.

Acknowledgement

This research is supported by NSC90-2212-E-033- 009.

9. References

ABB Robotics, Inc. (1996). *Training Manual S4Lite Robot Studio*

Banski, B. (1996). Local Motion Planning Manipulators Based on Shrinking and Growing Geometry Models, *IEEE International Conference on Robotics and Automation*, pp. 3303-3308, ISSN:1050-4729, Minneapolis, Apr 22-28, USA

Barraquand J.; Langlois B. & Latombe J. C. (1992). Numerical Potential Field Techniques for Robot Path Planning, *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 22, No. 2 (March/ April), pp. 222-241, ISSN:0018-9472

Danielle, S. & Mark, H. O. (2001). Motion Planning in Environments with Danger zones, *IEEE International Conference on Robotics and Automation*, pp.1488-1493, ISSN:1050-4729, Seoul, May 21-26, Korea

Jarvis, R. A. (1993). Distance Transform Based Path Planning for Robot Navigation, *Recent Trends in Mobil Robotics*

Latombe, J. C. (1991). *Robot Motion Planning*, Kluwer Academic Publication, Boston

Lei W.-I. (1999). *The Study of Obstacle Avoidance Methods for Industrial Robots*, Master Thesis, Department of Mechanical Engineering, Chung Yuan Christian University, Taiwan

Lozano-Perez, T. (1987). A Simple Motion Planning Algorithm for General Robot Manipulators, *IEEE Journal of Robotics and Automation*, Vol. 3, No. 3 (June), pp. 224-238, ISSN:0882-4967

Pobil, A. P. D. ; Serna, M. A. & Liovet, J. (1992). A New Representation for Collision Avoidance and Detection, *IEEE International Conference on Robotics and Automation*, pp. 246-251, Nice, May 12-14, France

Ralli E. & Hirzinger G. (1994). Fast Path Planning for Robot Manipulators Using Numerical Potential Fields in the Configuration Space, *Proceedings of IEEE/RSJ IROS*, pp.1922-1929, Munich, Sep 12-16, Germany

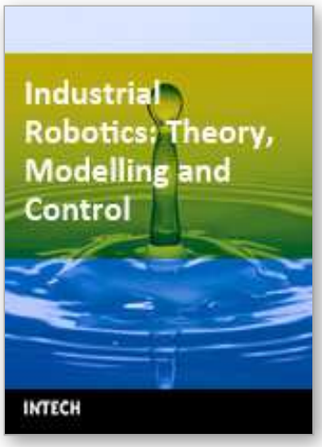
Red, W. E. & Truong-Cao, H. V. (1996). Unifying Configuration Space and Sensor Space for Vision-Based Motion Planning, *IEEE International Conference on Robotics and Automation*, pp. 3572-3577, Minneapolis, Apr 22-28, USA

Tarjan R. (1972). Depth-first Search and Linear Graph Algorithms, *SIAM Journal of Computing*, Vol. 1, No. 2, pp.146-149

Tesar, D. & Matthew, G. K. (1976). *The dynamic synthesis, analysis, and design of modeled cam systems*, Lexington, MA: Lexington Books, USA

Ting Y. & Lei W.-I. (1999). Using the Hierarchy Tree Method for Robot Path Planning, *IASTED International Conference on Robotics and Applications*, pp. 153~157, Santa Barbara, USA

Ting, Y. ; Lei, W.-I. & Jar, H.-C. (2002). A Path Planning Algorithm for Industrial Robots, *International Journal of Computers & Industrial Engineering*, Vol. 42, No. 2-4 (April), pp. 299~308, ISSN:0360-8352



Industrial Robotics: Theory, Modelling and Control

Edited by Sam Cubero

ISBN 3-86611-285-8

Hard cover, 964 pages

Publisher Pro Literatur Verlag, Germany / ARS, Austria

Published online 01, December, 2006

Published in print edition December, 2006

This book covers a wide range of topics relating to advanced industrial robotics, sensors and automation technologies. Although being highly technical and complex in nature, the papers presented in this book represent some of the latest cutting edge technologies and advancements in industrial robotics technology. This book covers topics such as networking, properties of manipulators, forward and inverse robot arm kinematics, motion path-planning, machine vision and many other practical topics too numerous to list here. The authors and editor of this book wish to inspire people, especially young ones, to get involved with robotic and mechatronic engineering technology and to develop new and exciting practical applications, perhaps using the ideas and concepts presented herein.

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen