Research Journal of Applied Sciences 2 (12): 1218-1222, 2007
ISSN: 1815-932X

Robots with Obstacles

[1]Eswar Reddy Chevi Reddy, [2]Rajagopal Kuderu and [3]Prasanna Lakshmi Kaujala
[1]Department of Mechanical Engineering, SVU, Tirupati, AP, India
[2]JNTU, Hyderabad, AP, India
[3]Department of Mechanical Engineering, Pulivendula, AP, India

Abstract: This study discusses about the planner in which the simulation ensures that all the robots starting from its source position are destined. Since it considers the static obstacles along with the moving obstacles which are robots itself, the path is coordinated in such a way that the trajectory is deviated when it is heading towards the obstacle and the path of the trajectory is a straight line till it comes near the obstacle and then it takes the shape of the obstacle with an offset distance of maintaining the minimum gap from the obstacle and again follows the straight line motion. When these robots are having collision with the other moving robots then, the priority is taken into consideration and the collision is avoided by changing the velocity of the robots.

Key words: Simulation, trajectory, deviation, collision, obstacle, priority

INTRODUCTION

The field of robotic research plays a very vital role in day today activities. Using a team of robots, accomplishing the tasks by different approaches has become very competitive in the market. Many sophisticated applications such as in the space, Hospitals, Manufacturing units, usage of multiple robots has become mandatory if the precision is given the top priority. When multiple robots are moving, every other moving robot becomes an obstacle for the moving robots. One approach is Centralized Planning (Schwartz and Sharir, 1983; Svestka and Mark, 1998) in which the trajectories are constructed for each robot in a specific order such that each trajectory is collision free of previous constructed trajectories. This approach ensures the solution but as the number of robots increases, the computational effort becomes very tedious. Another approach is Decentralized Planning (Steven and Seth, 1998) in which path is computed for each robot independently and a coordination diagram is used to plan a collision free trajectory for each robot along its path. In this approach, many number of robots having identical shape can be introduced without any computational effort because it is a dynamic path planning, but whether every robot reaches its destination or not is not ensured. The suggested technique in this study is to make the robot to move to their destination without colliding with each other and also with the static obstacles present if any, in the workspace namely rectangular workspace in minimum possible time. The proposed approach makes the added robots also to reach their destination without much computational effort and all the robots are reaching their destination is ensured.

If a robot is moving in an unknown environment, it has to satisfy multiple criteria. It has to acquire all the information about the unknown environment either through sensors or through path planning algorithms and generate a path to the target. A fuzzy decision making system (Boonphoapichart *et al*., 2003) is used. It is using decision levels for determining rough direction and precise trajectory. Autonomous system architecture is proposed in which a multi criteria fuzzy decision making is proposed and an alternative path detection method is used in order to avoid the collision with the static obstacle as well as moving obstacles. Multi factorial evaluation is used in which multiple objective decisions making are used.

In decentralized agent system, communication among the robots is very important. In this system, it is difficult for each robot to plan its effective motion while considering unpredictable coordination and every robot has to communicate globally to the all the other robots at the same time in order to avoid conflicts. In Takanori and Toshio (1993) while planning, every robot uses a common static world model to express a

Corresponding Author: Prasanna Lakshmi Kaujala, Department of Mechanical Engineering, Pulivendula, AP, India

common environment including public resources using this approach each robot gradually acquires experience where and when the public resources are crowded through iterative learning process and depending on the experience each robot becomes ale to plan reasonable path.

In prioritized motion planning for multiple robots is a simple approach (Jur and Mark, 2005) already introduced by Erdmann and Lozano-Perez (1987). It works as follows: Each of the robots is assigned a priority. Next, the robots are picked in order of decreasing priority. For each picked robot a trajectory is planned, avoiding collisions with the static obstacles as well as the previously picked robots, which are considered as dynamic obstacles. The approach requires two important ingredients: a method to plan motions for a single robot in known dynamic environments and a scheme to prioritize the robots. In this study, we discuss how we implemented them in our method.

When multiple robots move, the collision of robot to robot is different than the collision of robot with the static object. A Coordinate Space (CS) can be used to make the robots move along specified geometric paths in Cartesian Space. When the number of robots is restricted to two, this CS (Zeungnam and Jihong, 1992) can be used to detect the collision between the robots. the concept of coordinate space is used to find the time optimal trajectory of robot 1 and robot 2, i.e., plan the time optimal trajectory of robot 2 using the trajectory of robot 1 as a constraint and time optimal trajectory of robot 1 using the trajectory of robot 2 as a constraint.

When planning the motion of an actual robot, various dynamic constraints are to be considered which restricts the motion capability of robot. This study (Fraichard, 1999) provides the solution to planning a geometric path that avoids the stationary obstacles and planning the velocity along this path so as to avoid the moving obstacles.

PROPOSED TECHNIQUE

A rectangular workspace is taken in which the initial positions and target position of the robots is already specified. One can change the coordinates of the robots either in the initial position or in the final position. The same program may be modified by taking the static obstacle and without colliding with these static obstacles, the robots should reach their target position.

The program is so flexible that any time before the simulation starts,

- The position of the robot can be changed.
- The position of the obstacle can be changed.
- The size of the obstacle can also be changed.
- The number of obstacle can be increased.

The robot can reach its destination in minimum time if and only if it moves in a straight line which is formed by joining the coordinates of initial and target position.

Let S = $\{S_1, S_2, S_3, S_4, \ldots\}$ Where S- source

$S(x_i,y_i)$ i.e, i = 1,2,3 .. etc. $S(x_i,y_i)$-the coordinates of the Source position

Let T = $\{T_1, T_2, T_3, T_4, \ldots\}$ Where T- Target

$T(x_i,y_i)$ i.e, I = 1,2,3 .. etc. $T(x_i,y_i)$ -the coordinates of the Target positions are

Since the robots would be moving in a straight line, the length can be found out by

$L_i = \sqrt{(Tx_i-Sx_i)^2 - (Ty_i-Sy_i)^2}$ where I= 1, 2, 3….etc.

If the 3rd (Lavendar) and 8th (Blue) robot collides with each other, at some intersection point, then before it gets intersected, the velocity is calculated dynamically

$V_i = L_i/t$ and $V_j = L_j/t$, say I = 3 and j = 8, where V is the velocity and t is the time.

And according to the priority, whether i^{th} or j^{th} robot should reach early to its destination is decided and if i^{th} robot has to reach first, then j^{th} robot's velocity slowly decelerates. Once the i^{th} robot is away from the collision range of j^{th} robot, j^{th} robot starts accelerating and reaches its target position as soon as possible.

Let the robot be circular or square, the coordinates implies the centre of the circle or the centre of the square which is formed by the intersection of the diagonals of the square. The obstacles can be placed anywhere in the workspace, but its size and shape should be known before the simulation starts. The placement of the obstacle is very important because the path of the robot in a straight line is completely depended on the position of the obstacles if any is present in its path. The algorithm takes care about the movement of the robots.

STRAIGHT LINE AND CURVE ALGORITHM

The robots are moving on a straight line based on Bresenham's line algorithm and reaching their goals. If in its path any other mobile robot or static obstacles are present, then the robot will slowly decelerate and stop. The priority (Jur and Mark, 2005) is taken into consideration and according to the priority, the robot will move to its destination in the straight line path. The

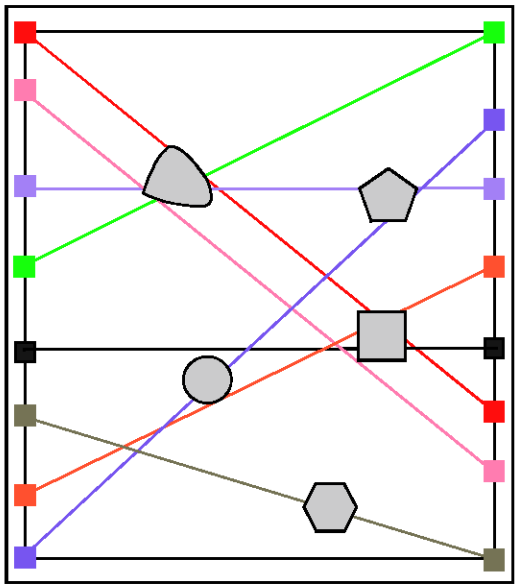

Fig. 1: Path planning without detecting the obstacles

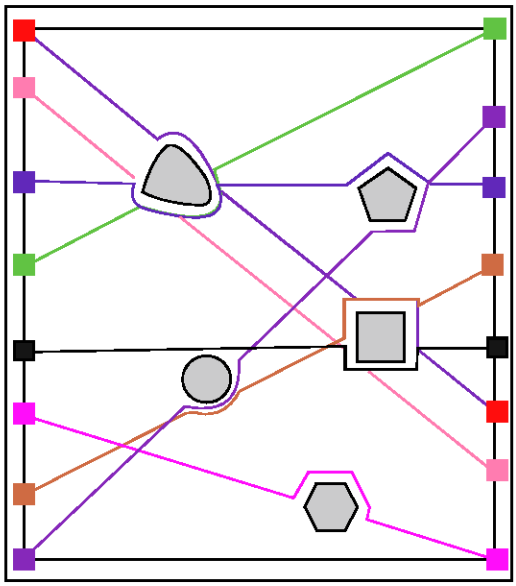

Fig. 2: Path planning detecting the obstacles

approach here taken is a decentralized approach (Lumelsky and Harinarayan, 1997) in which the robots will act dynamically. Any number of robots can be taken according to the workspace taken. And before the simulation starts, any number of robots can be added with its source and destination coordinates. Since, lot of collisions might occur, any of the following criteria can be taken to solve the collision.

- Giving the priority to the robots in the collision range and according to the priority, the robots will move.
- Giving the priority to the robots in the collision range and according to the priority, the velocity of the robot will change.
- Without giving the priority, which ever robot is nearest, it will move first.

The robot would be moving on the path forming a linear equation

$$Y = y_{si} + (y_{ti}-y_{si})\,(X-x_{si}) / (x_{ti}-x_{si}) \text{ where } I - 1, 2, 3 \ldots .\text{etc.}$$

If the obstacle presence is not considered, then the movement of the robot would be as shown in Fig. 1.

If this robot comes across any obstacle, the robot start moving on the edge of this obstacle keeping a gap of 1 pixel and it continue to move around the edge of the obstacle as shown in Fig. 2, till again it reaches its own path which is formed linearly.

Flexibility of the algorithm: The algorithm is so flexible that any number of obstacles can be placed anywhere in the workspace. If there is a situation in which the robots are bound to collide with each other, then the priority is given to the robot which should reach early. Even if 2 or more robots are very near to each other and also to the obstacle, then also the priority is assigned as if which robot should move first.

Once a collision is detected, based on the priority, the highest priority robot will move on its path without stopping and the other robot which was moving with some acceleration, will slowly decelerate. If it happens with 2 or more robots, the highest priority robot continues its journey and the other robots slowly decelerate and once the other robot is away from the collision range, it also starts moving. When the robot is encountered with an obstacle, it has to detect the shape of the obstacle, so that the same peripheral shape path can be planned and move on this path till it reaches its original straight line motion.

Based on the position of the static obstacle having different shapes, before the simulation starts, the path is planned accordingly, so that it do not collide with the new static obstacle position and also follow the shortest path. In this shortest path if it encounters any obstacle, then the path is deflected, so that it surrounds the obstacle maintaining minimum distance. There are two ways that this robot can move around the obstacle which is shown in the Fig. 3. If the obstacle's shape is a perfect rectangle,

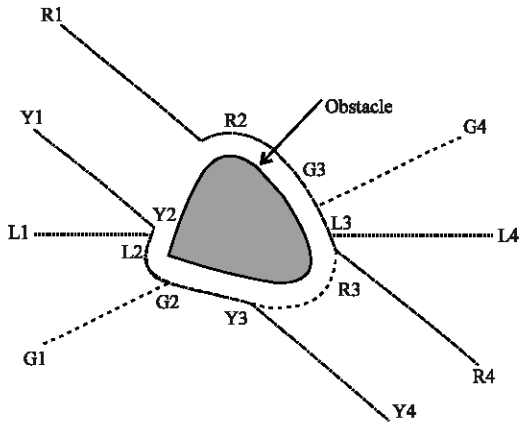

Fig. 3: Deviated path of the robot due to the obstacle

or square or circle, then a proper equation can be formed, but if its shape is an arbitrary one as shown in Fig. 3, then the every edge coordinates has to be known in the workspace. Since, the algorithm used is able to detect all the robots as well as the obstacles as an image and since image is having a boundary in the workspace, the boundaries coordinates are also known. If the edge coordinate is known , a curve is best fitted based on the theorem of least square method. Since, the shortest path is a straight line, the robot will follow the straight line path till it comes across the obstacle.

Let us say the red robot in Fig. 3 moves on the linear line from the coordinate R1 to R2. From R2 onwards, it can deviate its path in the non linear fashion either in the right side direction or left side direction to move to R3. Which ever path is taking less time, It will prefer that path . It continues to move till it reaches coordinate R3 and again follow the straight line motion up to its destination through coordinate R4. The deviated curved path is best fitted by the curve fitting which is exponential function, in which the vertical distance is y and the horizontal distance in x. and this path is set to an offset distance away from the obstacle and this offset distance is the minimum distance to be maintained from the edge of the robot, so that the nearest edge of the robot do not collide with the nearest edge of the obstacle. Similarly the green, yellow and lavender robots will follow the exponential functional curve after and before the linear interpolation path to reach to its destination.

The curve traced around the irregular object is formed using Exponential Functions:

$$Y = Ae^{Bx},$$

$$\ln y = \ln A + Bx,$$

$$A \equiv \exp(\alpha)$$

$$B \equiv b$$

$$\alpha = \frac{\sum_{i=1}^{n} \ln y_i \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i \ln y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

$$b = \frac{n\sum_{i=1}^{n} x_i \ln y_i \sum_{i=1}^{n} x_i - \sum \ln y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

RESULTS AND DISCUSSION

This research considers a coordinated path based planning of a robot. Specifically, it addresses the issue of multiple robots operating safely without collisions in a domain with velocity and priority constraints. A numerical method is obtained to find the curved paths to make the robot move in minimum time to reach its destination around the obstacles shape. Hence we are able to achiever very fast preprocessing and real time motion coordination and obstacle avoidance even with large number of unstructured agents. The future research could be to make the path still more optimal and in the real time, taking any number of robots should not increase the computation cost and it should also give the guarantee that the robots are reaching its destination which is the basic concept of decentralized motion planning. The present research is carried out on a X86 based PC Intel Corporation and on the Microsoft Windows XP Professional as the operating system. The simulation of robot movement is carried out using JAVA as software having a version of 1.5.

REFERENCES

Boonphoapichart, S., S. Komada, T. Hori and W.A. Gruver, 2003. Robot Motion Decision Making System In Unknown Environments. In: Proc. IEEE. Int. Conf. Robotics and Automation, Taipei, pp: 4197-4202.

Erdmann, M. and T. Lozano-Perez, 1987. On multiple moving objects, Algorithmica 2. Artificial Intelligence Lab., 2: 477-521, 1419-1424.

Fraichard, Th., 1999, Trajectory Planning in a Dynamic Workspace: A 'State-Time' Approach. Advanced Robotics, 13: 75-94.

Jur Van Den Berg, P. and H. Mark Overmars, 2005. Prioritized Motion Planning For Multiple Robots. In: Proc. Int. Conf. Intelligent robots and Sys. IROS., pp: 2217-2222.

Lumelsky, V.J. and K.R. Harinarayan, 1997. Decentralized Motion Planning for Multiple Mobile Robots: The Cocktail Party Model. Autonomous Robots, 4: 121-135.

Schwartz, J.T., M. Sharir, 1983, On the Piano Movers problem-III. Coordinating the motion of several independent Bodies moving among polygonal Obstacles. Int. J. Robotics Res., 3: 46-75.

Steven Lavalle, M. and A. Seth Hutchinson, 1998. Optimal Motion Planning For Multiple Robots Having Independent Goals, Robotics and Automation, 14: 912-925.

Svestka, P. and H. Mark Overmars, 1998. Coordinated Path Planning for Multiple Robots. Robotics and Autonomous Systems, Elsevier, 23: 125-152.

Takanori Shibata and Toshio Fukuda, 1993. Coordinative Behavior in Evolutionary Multi Agent Robot System. In: Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Sys., pp: 26-30, 448-453.

Zeungnam, B. and Jihong Lee, 1992. A Minimum Time Trajectory Planning Method for Two Robots. Robotics and Automation, 8: 414-418.