Research Paper

Volume : 2 | Issue : 5 | May 2013 • ISSN No 2277 - 8179

Engineering

KEYWORDS :

Motion Planning of Moving Robots Amongst Static Obstacles

Dr. K Prasanna Lakshmi JNTUH College of Engg. Manthani, A.P., INDIA

ABSTRACT

Every robot is designed in such a way that it can accomplish the task assigned to it in the minimum time, counterfeiting the obstacles in its path. How efficiently the robot can move in a cluttered environment where many other robots are moving along with the stationary obstacles is important. The paper deals with the technique used in the context of moving the robot from its initial position to the goal position minimizing the time and counterbalancing the hindrances caused by the obstacles by deviating from its pre planned trajectory.

Introduction

Multiple robot motion has either decoupled approach or centralized approach. Centralized approach defines the pre planned paths and any modification in the system, rescheduling the pre planned paths, thereby increasing the computational effort, but ensures that the robot reaches its target. Decoupled approach plans the paths dynamically and therefore any modification do not alter the computational effort, but every robot in the considered workspace reaches the target or not, is not ensured. Therefore a balanced optimized value between the computational effort and reaching the target is chosen.

Previous work

[1], The new strategy outperforms both the centralized and decoupled planning approach, indicating that in practice it can be a better choice, than both the centralized and decoupled approach, for solving multi-robot planning problems. [2], The Directive circle prevents the robot from being trapped in deadlocks or local minima. It is assumed that the target's velocity is known, while the speeds of dynamic obstacles, as well as the locations of static obstacles, are to be calculated online. [3], The static obstacles along with the moving obstacles which are robots itself, takes the coordinated path 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. [4], The motion of various robots to reach one's destination within veiy less time so as to make all the robots reach. its goal position in a minimum possible time. The planner will provide the path to the robot in a straight line without encountering any obstacle and the other planner will resolve the deadlock. situation where-in no robot can move forward. [5] A complement to the traditional panel method is introduced to generate a more effective harmonic potential field for obstacle avoidance in dynamically changing environments and a group of mobile robots working in an environment containing stationary and moving obstacles is considered. The group is not forced to maintain a formation during the motion. Every robot considers the other robots of the group as moving obstacles and hence the physical dimensions of the robots are also taken into account. [6], Coordination is achieved by planning robot velocities along the paths through a velocity-optimization process. An objective function for minimizing formation errors is established and solved by a linear interactive and general optimizer. Motion planning can be further adjusted online to address emergent demands such as avoiding suddenly appearing obstacles. [7], Motion planning is dynamic in nature which is a property of the decoupled technique. An attempt is made to utilize the advantages of both centralized and decentralized approach and thereby overcoming to some extent, the limitations of both the approaches. An extension of the [7] is brought out in the Methodology of this paper.

Methodology:

The robots are represented as a rectangular plane having some area and every robot in the system is given an initial position and target position to approach to. The robot's rectangular workspace area while moving from original position to goal position is checked in every incremental value either in the x direction or y direction for the intersection with the area of the obstacles or intersection with the area of the workspace area of the other robots.

How collision is detected ?

Line Intersection command in Java applet is used to detect the intersection between all the boundary lines of the robot's workspace with the circumferential lines of the obstacles. The direction of the robot is decided based on the slope of the movement from initial to final position. If the slope is positive, the robot motion is decided to move in the upward and forward direction. If the slope is negative, the robot motion is decided to move in the downward and forward direction. If the slope is zero, obviously the robot motion is parallel. Based on the slope, when the robot travels from initial to final position, the upward, forward and downward motion, the incremental pixel value of the robot movement is based on the following calculations :

$$X_{inc}=\frac{ctL-crL}{10} \quad \text{Eq 1}$$

Where ctL and crL are the coordinates of the left position of the workspace of robot's target and initial corners respectively

$$Y_{inc}=\frac{ctT-crT}{X_{inc}} \quad \text{Eq 2}$$

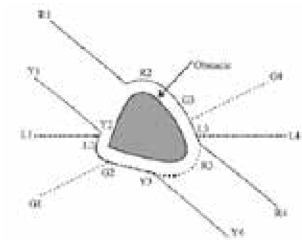
Where ctT and crT are the coordinates of the top position of the workspace of robot's target and initial corners respectively.

The robots in the workspace will continue to move until it reaches its destination. In the considered workspace, the robots move only in forward direction. Every one increment of one of the robot will check its collision with the workspace of other robots, boundary lines and obstacle's workspace. The movement of the any of the robot depends upon the priority of the task assigned. The first prior task to be accomplished by the robot must and should have its identity in such a way that, the most prioritized identified robot moves early as compared to the less prioritized robots. Therefore the priority as a fundamental criteria should be related to the identity of the robot. Once the priority is known, according to their corresponding identity the robots will move.

Since all the robots will move simultaneously, according to the incremented value, every robot's position is known, the first incremented value of prioritized robot's position is checked with all the other robot's incremented value. If any collision is detected by the line intersection method, then the second prioritized robot will not be incremented to new position and it retains its previous position. Likewise, the checking is done for all the robots in the one to one fashion. Therefore the programming is dynamic because any modification in the position of the robot is fed back to the system and the collision detection takes place with the latest information available within the system.

Volume : 2 | Issue : 5 | May 2013 • ISSN No 2277 - 8179

While moving in the pre determined path which is liable to change with every movement dynamically, the robot's workspace may come across the obstacles's worskspace. Since the obstacles are the stationary hindrances and the position of the obstacles are static too, the robots cannot move in its workspace. Whether robot is moving in the upward direction or downward direction is known by the Eqn. 2(where Y_{inc} is negative), the upward motion makes the robot move in the forward and upward direction, and if it come across the position of the obstacle, the path of the robot is changed. It also depends upon the shape of the obstacle. If the obstacle is a curved shaped as shown in Fig. A, the [3],Prasanna, the path is followed.



$$Y = Ae^{Bx},$$
$$\ln y = \ln A + Bx,$$
$$A = \exp(a)$$
$$B = b$$
$$a = \frac{\sum_{i=1}^{n} \ln y_i \sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} x_i \ln y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$
$$b = \frac{n\sum_{i=1}^{n} x_i \ln y_i \sum_{i=1}^{n} x_i - \sum \ln y_i}{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}$$

Fig A.

If the obstacle is a polygonal shape, then it is virtually inscribed in a rectangle and the four corners of the rectangle decides the shape of the obstacle. When the robot moving in the upward direction counteracts the obstacle, then the path of the robot changes and modified path illustrates the motion of the robot from its present position to the left and topmost position of the obstacle, and then the path continues to its destination as shown in the Figure from Fig 1 to Fig 10.

Similarly when the robot is moving in the downward direction(where Y_{inc} is positive), counteracts the obstacle, then the path of the robot motion changes from its previous trajectory to the left and bottommost position of the obstacle, and then the path continues to its destination as shown in the Figure from Fig 11 to Fig 14.

Extra robots can be added to the existing workspace with the same programming concept and ensures that the robot will reach its destination.

Results : The results shows the time taken by each robot keeping in consideration of the priorities assigned to the Robots.

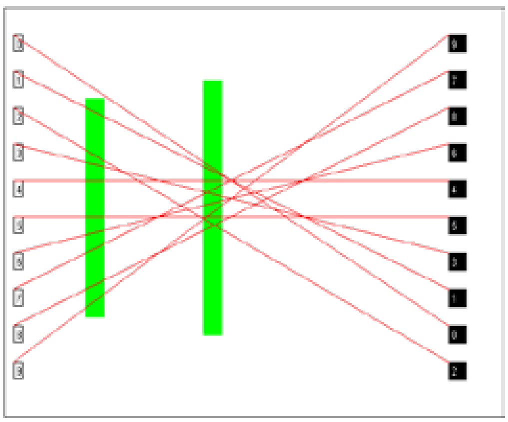


Fig 1 : Robots with initial and final positions

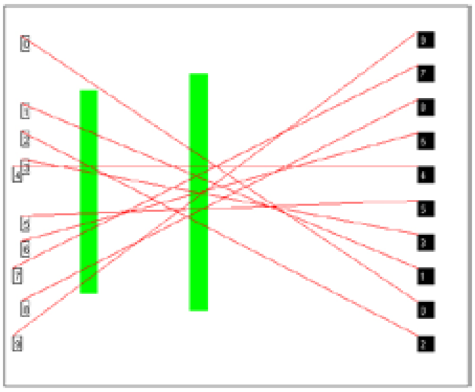


Fig 2 : Robots moving with incremented values.

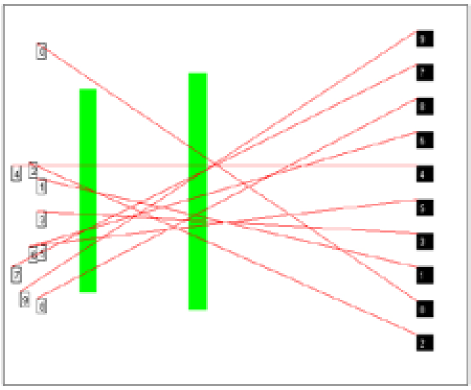


Fig 3 : Robots moving with priority like in Robot No. 1 & Robot No. 3.

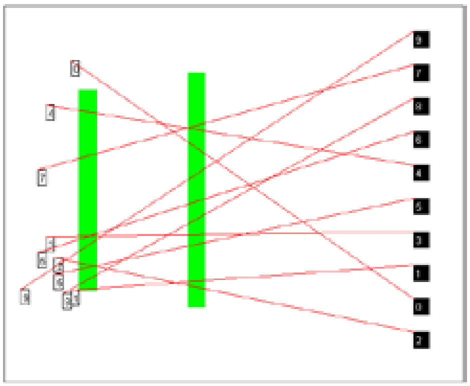


Fig 4 : Robots changing its path due to hindrance by the 1st obstacle.

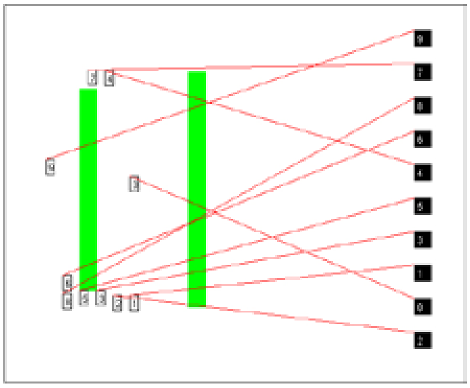


Fig 5 : Robots counteracted with the 1st obstacle and encountering with the 2nd one.

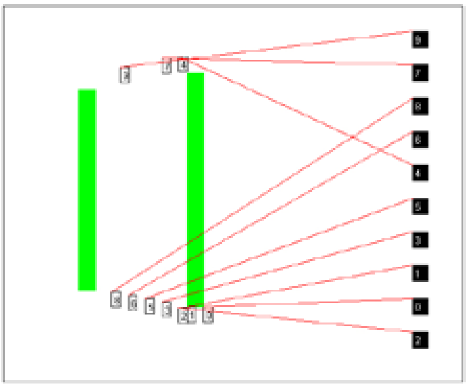


Fig 6 : Robots changing its path due to hindrance by the 2nd obstacle.

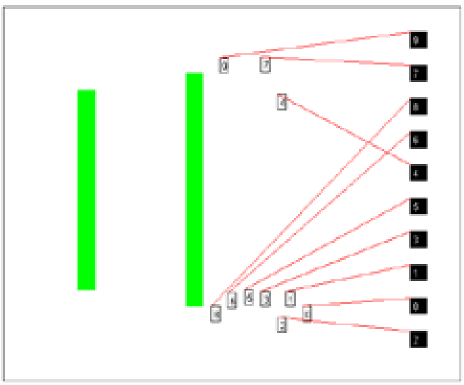


Fig 7 : Robots cleared the blockage by the obstacles and reaching its destination.

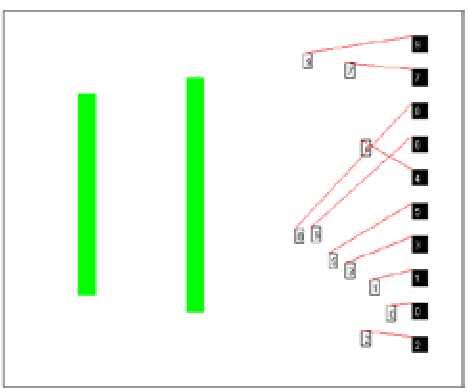


Fig 8 : Robots approaching their destination.

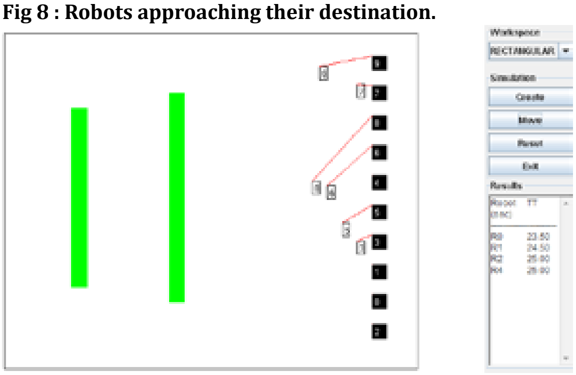


Fig 9 : Some of the robots reached their destination with indicated time.

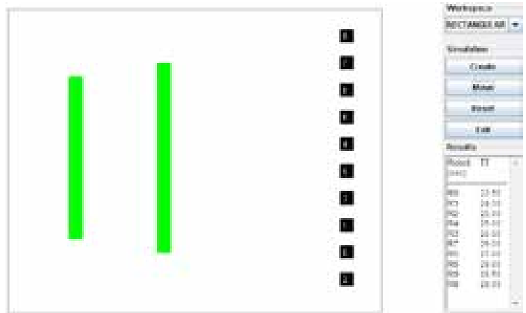


Fig 10 : All the robots reached their destination with indicated time.

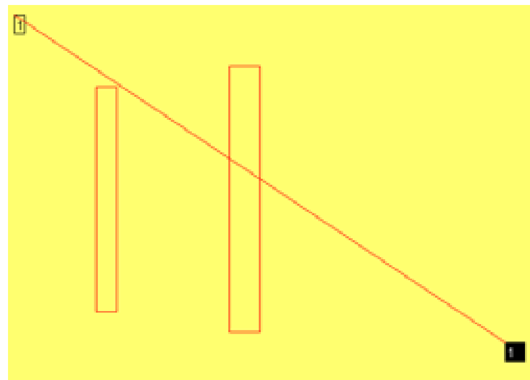


Fig 11 : Inclined line indicates the the path from its intial position to its goal.

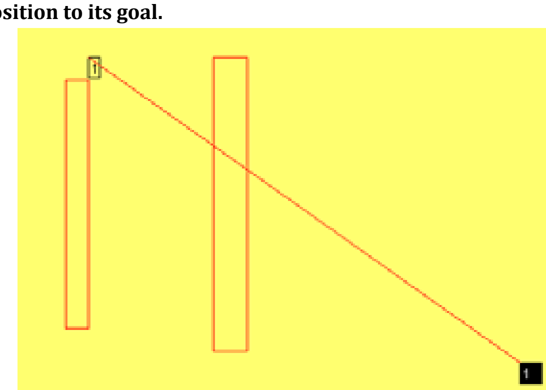


Fig 12 : The robot has changed its path due to collision with the 1st obstacle.

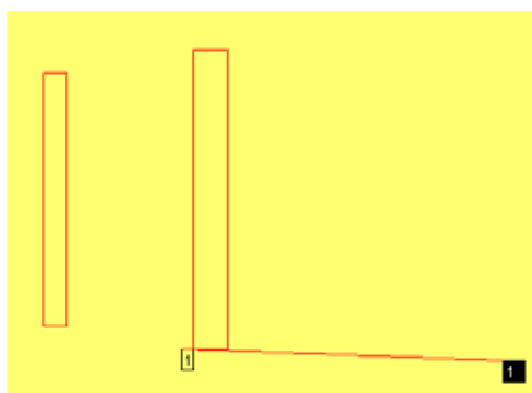


Fig 13 : The robot has changed its path due to collision detection by the 2nd obstacle.

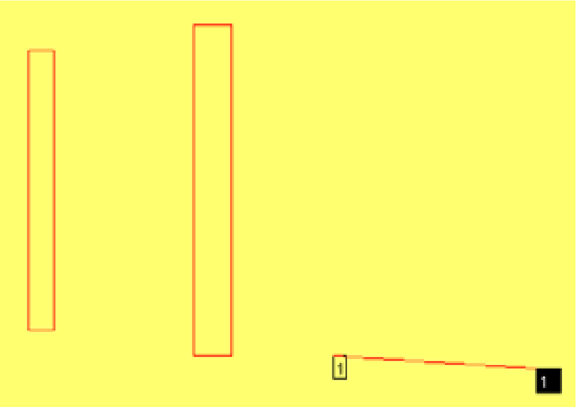


Fig 14 : The robot approaching its destination.

Conclusion & Future Scope :

In the decoupled approach, the robots reaching their destination is ensured and in the centralized approach, the computational effort doesn't vary with respect to the robots getting added, which is the biggest disadvantage in the research area, when multiple motion of the robots are considered. More complicated shaped obstacles and more number of hindrances, may be considered as the future scope.

REFERENCE

1. Multi-Robot Motion Planning by Incremental Coordination, International Conference on Intelligent Robots and Systems, IEEE, pp. 5960 – 5963, Saha, M. , Isto,P., CA , 9-15 Oct., 2006. | 2. Robot Motion Planning in dynamic environments with moving obstacles and target, World Academy of Science, Engineering and Technology, Ellips Masehian & Yalda Katebi, 2007. | 3. Robots with Obstacles, Research Journal of Applied Sciences, Medwell Journals, Vol No. 2 (12), pp. 1218-1222, Prasanna Lakshmi k, Eswar Reddy C & Rajagopal K, December 2007. | 4. Robots moving with Sense, Spectrum, Vol. No. 3, pp. 16-21, Prasanna Lakshmi k, Eswar Reddy C & Rajagopal K, March 2008. | 5. Real time obstacle avoidance for multiple mobile robots, Robotica, Vol. 27(2), pp. 189-198, Farbor Fahimi, C. Nataraj & Hashem Ashrafiouon, April 2008. | 6. Coordinated Motion Planning for Multiple Mobile Robots Along Designed Paths With Formation Requirement, IEEE Transactions on Mechatronics, Vol No. 6(6), pp. 1021-1031, Dong Sun & Changan Zhu, December 2011. | 7. Motion of Robots in Non Rectangular Workspace, International Journal of Engineering Inventions, Vol No. 2(3), pp. 35-40, Prasanna Lakshmi K, February 2013.