International Journal of Engineering Inventions
e-ISSN: 2278-7461, **p**-ISBN: 2319-6491
Volume 2, Issue 3 (February 2013) PP: 35-40

Motion of Robots in a Non Rectangular Workspace

K Prasanna Lakshmi
Asst. Prof. in Dept of Mechanical Engineering
JNTU Hyderabad

Abstract:- The paper deals with the geometrical aspect of the motion planning of multiple robots in a non rectangular workspace. Motion planning is dynamic in nature which is a property of the decoupled technique. Every robot in the considered workspace is reaching the destination which is a property of the centralized planning approach. The boundary of the non rectangular workspace can be altered and the number of robots in the workspace can be increased or decreased. 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. The motion of the robots are purely geometrical and the dynamic aspects in terms of the physical interaction is completely ignored. The number of robots can be added and the boundaries can also be altered without much change in time for reaching the destination of every robot.

Keywords:- Workspace, Decoupled Technique, Centralized Approach, Motion Planning

I. INTRODUCTION

The concept of centralized and decentralized planning is implemented to move the robots from its initial to final destination. Some advantages of centralized planning over the decoupled techniques to some extent is globally accepted, but as the number of robot increase along with the static obstacles, the computational effort becomes very tedious and the time the robot takes to reach its destination is not that effective. The trade offs between these two approaches lie on efficiency and completeness. But centralized planning gives a proper and exact solution of every robot in the considered workspace, to reach their goals promptly, though it may take lot of time. The only advantage the decoupled technique has, that dynamic planning is possible without much computational efforts and can reach the destination in an effective time. But this technique cannot ensure the reaching of the robots to its destination. In this paper, an attempt is made to take the advantages of centralized and decoupled planning so as to reduce the expensive computational steps and ensuring every robot to reach its destination.

An attempt is made to overcome the problems dealt earlier in the motion of the multiple robots. The task accomplished in making the robot reach its target and the time of reaching the target could be minimized. But since the workspace taken is a limited area during simulation, the number of robots that could be added is also limited. If the workspace during simulation can be increased, the same operations can be performed well with more number of robots.

The position of the robot and the position of the obstacles are recognized by the tactile sensors incorporated to the geometrical aspects of the robot and its environment. Every movement of the robot with respect to the other robots at every instant is to be monitored continuously, so that the robot can take decision instantaneously about its path planning and can trace on the newly planned path which is free of collisions. Though the obstacles are stationary, the moving robot is also treated as obstacle and can be called as moving obstacles. In the real time, the moving obstacles could be the other robots, the automated guided vehicle, the human being, the overhead cranes etc. But the research contribution of this paper is limited to only the geometrical aspects rather than the physical and mechanical aspects. The dynamic properties and the issues related to the mechanical interaction between two physical objects are completely ignored. And therefore the work is purely based on only the geometrical structure.

II. LITERATURE REVIEW

Decentralized motion control system leads each robot to their individual goals while keeping connectivity with the neighbours. An experimental result is presented with a group of car like robots equipped with omni directional vision system [7].

A roadmap is constructed for one robot using probabilistic path planner which guarantees that the approach can be easily applied to different types of robots. A number of these simple roadmaps are combined into a roadmap for the composite robot. A coordinated approach is used rather than decoupled planning in which the method is probabilistically complete and can solve the problem within a finite amount of time, but is limited to five robots [6].

The paper in [2] deals with the delay in the collision checking in Probabilistic roadmap planning. The paper alludes that the PRM planner spends most of their time in performing collision checks even if it is not required. Since in the PRM planners, the algorithm will predefine the paths, considering the collision checking at each interpolated points, a connection strategy can be constructed in which while this algorithm is running, one can schedule these tests along the path in order to reveal a collision as early as possible. Since the PRM planners are used in centralized approaches, if more number of robots with more degrees of freedom is introduced, the algorithm will become computational intensive to predefine the paths and also decreasing the delay in collision checking.

"Decentralized motion planning for multiple mobile robots: The Cocktail Party Model" has dealt with mainly decentralized motion planning of robots. This programming is just like a drunkard in a party, who don't know how to move in the crowded area in order to avoid them and make a clear path [8].

The paper mainly deals with prioritized motion for multiple robots. The approach is based on a powerful method of motion planning in dynamic environments. The priorities on the robots can be assigned based on the importance of the task. Still finding the order in which the trajectories are planned can have a significant influence on how optimal the resulting path is .As the number of robots is increased, the run time and quality of the paths produced may come down drastically [5].

The paper mainly deals with prioritized motion for multiple robots. The approach is based on a powerful method of motion planning in dynamic environments. The paper has considered 24 robots along with the fixed obstacles and fixed source and target positions. And all these robots do not have any limitation on the number of degrees of freedom. But if the obstacle positions are changed and any arbitrary shaped obstacles are considered, this approach may give the solution but the robots might collide with obstacles. As the number of robots is increased, the run time and quality of the paths produced may come down drastically [4]. "Motion Planning of Multiple Robots Using Dynamic Groups" is a paper that proposes a concept of "groups" in motion planning. The groups are classified into static groups and dynamic groups. Thin, a grouping algorithm is implemented in Virtual Impedance Method. Static group is taken as a group in which members don't change and dynamic group is taken as the one in which members do change. This grouping is a bit irrelevant to the practical conditions because there are hardly any static groups. The robots need to be changed according to the necessity. This group approach needs to be overlapped with advantages taken from individual approach and this combination may make the paper complete [3].

The proposed techniques in paper [1], uses implementation of dynamic networks. When any two robots are within a communication range of each other, they establish a communication link. A network of robots is formed when two or more robots establish links between one another. Two robots in a network can then exchange information not just through direct communication links, but through any series of communication links within the network. Once the robots share the information, it broadcasts its plan to all the other robots in the network and therefore the best plan is implemented from the plans received from all the other robots. The communication range complexities increases with the increase in the number of robots.

III. PRESENT WORK A. COLLISION DETECTION

The collision is detected by checking the x and y coordinates of the present moved robot (assumed movement) with the earlier coordinates of the other robots. The robot's direction decides whether the robot is moving in the upward or downward direction. If only rectangular workspace is considered, the robot always moves forward and is incremented by 20 pixels. But based on the final position of the robot with its present position, the robot will either move in the upward direction or downward direction. If the final y coordinate of the target position is higher than its initial y coordinate, the robot will move in the forward and downward direction and if the finaly coordinate of the target position is lower than its initialy coordinate, the robot will move in the forward and upward direction. Then for the movement of the robot, it is placed 20 pixel forward, i.e, Width difference –Xf-Xi Xinc – 20 Lmove- Xf-Xi /20 Height difference –Yf-Yi Yinc– Yf-Yi/move So the robot will move by (Xi + Xinc , Yi + Yinc) Therefore the 1st robot's, first incremented position would be

(X_i, Y_i), where $X_i = X_i + X_{inc}$ and

Yi= Yi + Yinc

Generally, let us say the 1st robot's coordinate is (x,y). But it is occupying (x+20,y+20) area in the workspace, which means that in this area no other robot's coordinate should exist. The other corner coordinate of the 1st robot is say (ex,ey). Where ex=x+20 ey=y+20 If the 1st robot moves by one increment step ahead, it is checking the collision with the following conditions with every other robot on the workspace. One by one checking is taking place. That means the 1st robot's new position is checked with the earlier positions of the 2nd, 3rd, 4th till 10th. The 2nd robot's coordinate is say (nx,ny) and it occupies upto (enx,eny). Where enx=nx+20 eny=ny+20

DC(Detect Collision Method)
{If the nx value lies between or equal to x and ex, then the value of ny is checked. If the ny value lies between or equal to y and ey,
If any of the above condition comes out to be true, then there is a collision, if the above condition is false, then the following condition is checked
If the enx value lies between or equal to x and ex, then the value of eny is checked
If the eny value lies between or equal to y and ey,
If any of the above condition is true, then there is a collision. The process continues for the 1st robot with all the other robots.}

If there is no collision, since all the four conditions above are true, then the assumed incremented value is considered, and that particular robot is placed in its incremented position, and this would be its new position.

If there is collision, then the assumed incremented value is not considered and that particular robot is placed in its earlier position, since the new position of the robot is likely to collide, where it was.

B. Collision detected in non rectangular area without obstacles

If the workspace considered is a non rectangular workspace consisting of horizontal and vertical straight lines, as such the vertical and horizontal straight lines inside the rectangular region are taken as the obstacles and the robot motion is traced in such a way that the robots do not touch these lines which are the boundaries of the workspace and reach their targets in minimum possible time.

A virtual path is created from the initial to the final position of all the robots. Since the workspace considered is a non rectangular workspace, the robot movement could be either from left to right or from right to left in the confined workspace.

So completely there are four directions in which the robot can move
1)robot moving in the forward and upward direction
2)robot moving in the forward and downward direction
3)robot moving in the backward and upward direction
4)robot moving in the backward and downward direction

While following the motions, if the horizontal boundary or vertical boundary comes across its way, a virtual path is created which is collision free path.

The horizontal or vertical lines in the way of the movement of the robots are considered as the obstacles. So the path planning is in such a way that no robot should collide with each other and also should not collide with the boundaries coming in the way of the robots.

Every robot has four sides and four corners. In every moment of the robot, a constant check has to be made, so that the left, right, top and bottom of the robot should not cross the vertical and horizontal boundaries coming in its path.

The shortest path from the initial to the final coordinate is a straight line. If there are no obstacles, the robot will move in this shortest straight line. If in this path, the vertical line boundaries or the horizontal line boundaries intersect, the path of the robot will change.

The vertical boundaries coordinates are such that the topmost point of the vertical line is its initial coordinate and the bottommost point of the vertical line is its final coordinate. The horizontal boundary coordinates are such that the leftmost point is its initial coordinate and the right most point of the horizontal line is its final coordinate.

If the robot is translating in the forward direction towards its destination and comes across vertical line boundary, then 20 pixels is reduced from the initial x coordinate and 20 pixel from the initial y coordinate of the vertical boundary and this new coordinate becomes the next step of this robot. So the path of the robot will change and this path is included in the virtual path. From this coordinate, the robot again translates towards its destination.

If the robot is moving in the backward direction towards its destination and comes across vertical line boundary, then the new x position of the robot is the initial x coordinate of the vertical boundary and y position is 20 pixel reduced from the initial y coordinate of the vertical boundary. If the robot is moving in the forward direction towards its destination and comes across horizontal line boundary, then the new x position of the robot is final x coordinate of the horizontal line and the new y position of the robot is 20 pixel reduced from the final y coordinate of the horizontal boundary. If the robot is moving in the backward direction towards its destination and comes across horizontal line boundary, then the new x position of the robot is 20 pixel reduced from the initial x coordinate of the horizontal line and the new y position of the robot is 20 pixel reduced from the initial y coordinate of the horizontal line.

Likewise, all the new positions of the robots are saved in a path, and the robots move accordingly

in the new path, which is free from the collision with the horizontal and vertical boundary.
When the robots are moving in this path, again it has to check the collision with the other robots.
Width difference – X_f-X_i X_{inc} – 20
Lmove- X_f-X_i /20 Height difference –Y_f-Y_i Y_{inc}– Y_f-Y_i/move
If X_f is less than X_i, the robot will move in the backward direction, so the new position of the robot in x coordinate, X= X_i - X_{inc}
If X_f is greater than X_i , the robot will move in the forward direction
The new position of the robot in x coordinate is X_{inc} X= X_i + X_{inc}

Whether the robot will move in the upward direction or downward direction, is decided by the final and initial y coordinate, soIf final y coordinate is greater than initial y coordinate, the robot will move in downward direction Height difference – Y_f-Y_i
Y_{inc}– Y_f-Y_i/move Y= Y_i + Y_{inc}
If the final y coordinate is less than initial y coordinate, the robot will move in upward direction
Y_{inc} =(Y_i - Y_f)/Lmove Y- Y_i - Y_{inc}
Every movement in the virtual path with the incremented values, the collision is checked, and accordingly, the robots are moving. Since the workspace considered is a non rectangular one, the direction of the robot should be checked at every instant with the above mentioned logic.

IV. RESULTS AND DISCUSSION

Since the simulation is completely geometric, every pixel movement is taken in terms of the system time, and accordingly the system time is calculated from the initial position to the final position. The results are shown in the right bottom corner. As and when the robots are reaching their destination, the robot along with its number, against priority, the time is mentioned. Since the robots are moving in either forward or backward direction, and if priority is considered, the overall time of the robots reaching their destination may be more than if the priority is not considered. The motion of the robots is shown in the following figures. The transparent numbered robots are the initial positions and the opaque numbered robots are the final destination targets. The boundary of the workspace is the non rectangular area with the horizontal and vertical lines. In this workspace, there are totally 5 horizontal lines and 5 vertical lines with differ=

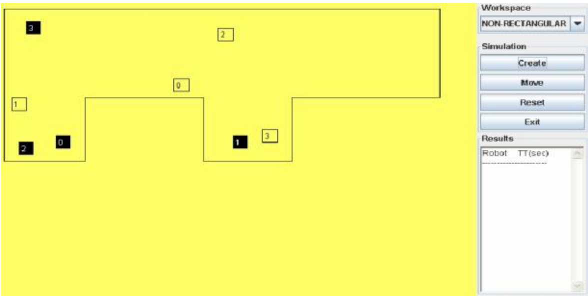


Fig. 1 The non rectangular space with the robot inside The red lines shown in Fig. is the shortest path of the robot from its initial point to its final point.

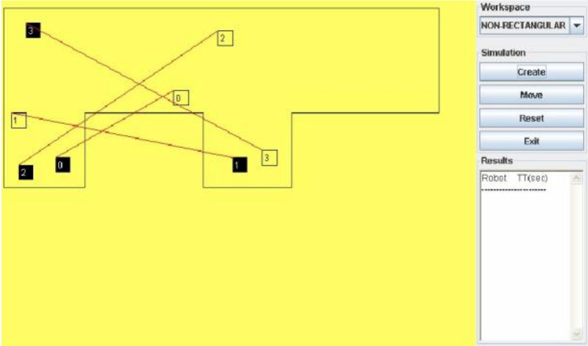
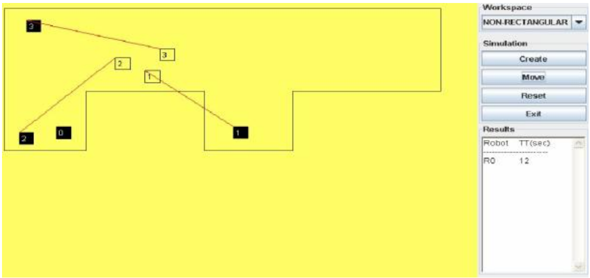


Fig. 2 The transparent robots are the initial position and the opaque robots are the final position



Workspace: NON-RECTANGULAR

Robot	TT(sec)
R0	12

Fig. 3 The robot 0 has reached its destination in 12 seconds, which is shown in the result box,on the right bottom side of the figure.

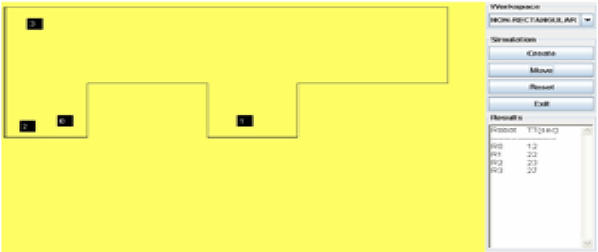


Fig. 4 The robot 2 has reached its destination in 23 seconds

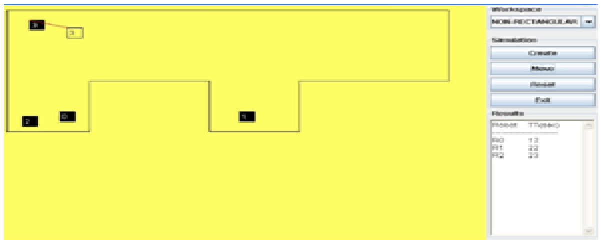
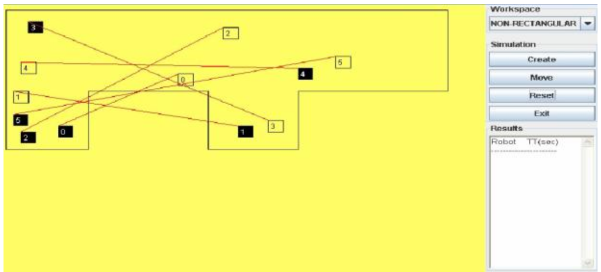


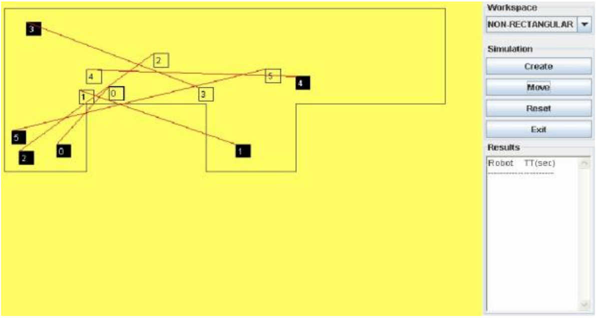
Fig. 5 The robot 3 has reached the destination in 27 seconds



Workspace: NON-RECTANGULAR

Robot TT(sec)

Fig. 6 An extra robot with number 5 is added.



Workspace: NON-RECTANGULAR

Robot TT(sec)

Fig. 7 The robots are moving from its initial position to the final position

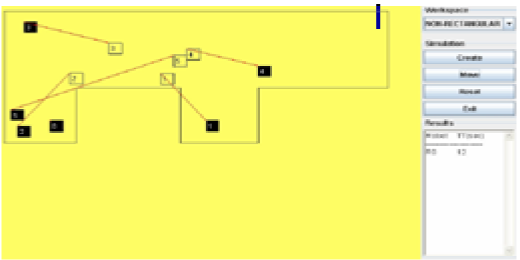
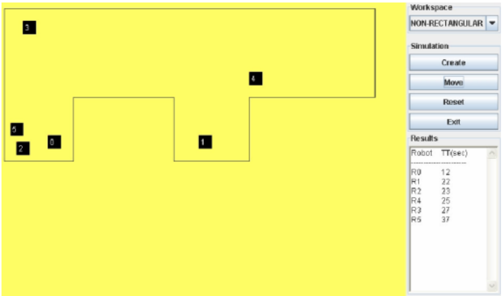


Fig. 8 Robots are moving towards their destination



```
Robot    TT(sec)
-----------------
R0       12
R1       22
R2       23
R4       25
R3       27
R5       37
```

Fig. 9 All the robots have reached within a time of 27 seconds.

V. CONCLUSION

Many researchers have contributed many approaches. In this paper, the work done is completely geometric based and the main concept is even if the number of robots are increased in the confined given workspace, the time each robot is taking to reach its destination is not much. But the workspace taken is a very limited space. Therefore the number of robot that can be increased is also a very limited one. And also in this contribution, the stationary obstacles are not considered. In future, there is a scope for the researchers to add stationary obstacles and find the collision free path without much change in the time of reaching their destination.

REFERENCE

1) Christopher M. Clark, Stephen M. Rock and J.C. Latombe, "Dynamic Networks for motion planning in Multi Robot Space System"
2) G.Sanchez and J.C. Latombe, "On delaying collision checking in PRM planning : Application to multi robot coordination", International Journal of Robotics Research, Vol
3) 21(1), 2002, pp 5-26.
4) J. Ota and T. Arai, "Motion planning of multiple mobile robots using dynamic groups", International Conference on Robotics and Automation, 1993, pp. 28-33.
5) Jur P. Van Den Berg and Mark H. Overmars, "Prioritized motion planning for multiple robots", International Conference on IRS, 2005, 2-6 Aug., 430-435.
6) Jur P. Van Den Berg and Mark H. Overmars, "Roadmap based motion planning in dynamic environment", International Conference on IRS, 2004, 28 Sep - 2 Oct, Vol 2,
7) 1598-1605
8) P. Svestka, Mark H. Overmars, "Coordinated Path Planning for Multiple Robots", Robotics and Autonomous Systems, Vol 23, 1998, 125-152.
9) Pereira G.A.S., Aveek K. Das, R.Vijay Kumar and Mario F.M.C., "Decentralized motion planning for multiple robots subject to sensing and communication constraints", Proceedings in International workshop on Multi Robot System, 2003, Vol II, 267-278.
10) V.J. Lumelesky and K.R. Harinarayan, "Decentralized motion planning for multiple mobile robots: The Cocktail Party Model", Kluwer Academic Publishers, 1997, 121-135.