Proceedings of the 39nd ISR(International Symposium on Robotics), 15~17 October 2008

Probabilistic Roadmap Method and Real Time Gait Changing Technique Implementation for Travel Time Optimization on a Designed Six-legged Robot.

Aamir Ahmad[1] and Nirjhar Dhang[2]
[1]Institute for Systems and Robotics,
Department of Electrical and Computer Engineering,
Instituo Superior Técnico, Lisbon, Portugal
[2]Indian Institute of Technology, Kharagpur, India
E-mail : aamir.iitkgp@gmail.com

Speaker: Aamir Ahmad, Indian Institute of Technology, Kharagpur, India (aamir.iitkgp@gmail.com)

Topic: 6-1 Cognitive Robotics: AI

Keywords: hexapod, motion planning, probabilistic roadmap, multi dof robot, legged robot

Abstract

This paper presents design and development of a six legged robot with a total of 12 degrees of freedom, two in each limb and then an implementation of 'obstacle and undulated terrain-based' probabilistic roadmap method for motion planning of this hexaped which is able to negotiate large undulations as obstacles. The novelty in this implementation is that, it doesnt require the complete view of the robot's configuration space at any given time during the traversal. It generates a map of the area that is in visibility range and finds the best suitable point in that field of view to make it as the next node of the algorithm. A particular category of undulations which are small enough are automatically 'run-over' as a part of the terrain and not considered as obstacles. The traversal between the nodes is optimized by taking the shortest path and the most optimum gait at that instance which the hexaped can assume. This is again a novel approach to have a real time gait changing technique to optimize the travel time. The hexaped limb can swing in the robot's X-Y plane and the lower link of the limb can move in robot's Z plane by an implementation of a four-bar mechanism. A GUI based server 'Yellow Ladybird' eventually which is the name of the hexaped, is made for real time monitoring and communicating to it the final destination co-ordinates.

1. Introduction

A lot of research is already done in the field of Path planning, navigation and localization related problems of mobile robots. In past decade the most important issue that has come up is regarding the increasing number of degrees of freedom of robots. Cell decomposition, Potential field method and probabilistic roadmap planning[1, 2, 3, 4] are some of the most widely acclaimed methods for path planning. Research show that the probabilistic roadmap method is often suggested and applicable with high degrees of freedom, but is computationally expensive as the degrees of freedom increase. The problem of localization is another challenge which has to be met while having effective path planning. Although the localization problem owes mostly to the hardware imprecisions, effective algorithms and feedback methods are used to negotiate with such errors.

In this paper we, present a application of probabilistic roadmap method on a six legged robot which has twelve degrees of freedom, two in each leg. The novelty lies in the use of dynamic gait changing mechanism of the robot which is done in real time. This technique is implemented in the preprocessing phase of the probabilistic roadmap method. The local planner in the query phase ultimately finds the exact path which the robot has to follow

The Localization problem of the robot is neglected for all practical purposes in this paper because experimentally it was found that the error amounts to 0.5mm in 10cm . A simulator for the robot is made in C++ environment which maps the obstacles, free and obstructed configuration spaces and finally displays the path robot follows graphically. The results at the end of paper show a comparison between actual robot motion in the arena and the simulator results. The slight difference owes to the minor localization problem of the robot. The rest of the paper is divided into five major sections, the hexapod description, map generation and decomposition, preprocessing phase which includes gait[5] changing technique, query phase and results followed by reference section and authors curriculum vitae.

2. The Hexapod

The hexapod[6] has two degrees of freedom in each leg. One in the robot's x-y plane and the the other in the z-y plane, considering the robot placed on level ground of x-y plane. The robot can assume two different gaits[5] at any time of the motion, the alternating tripod gait and the crab gait. In [Figure 1] alternating tripod gait is presented. In the crab gait the distance moved in sideways is 2 cm and the robot structure remains as the normal state of [Figure 1].

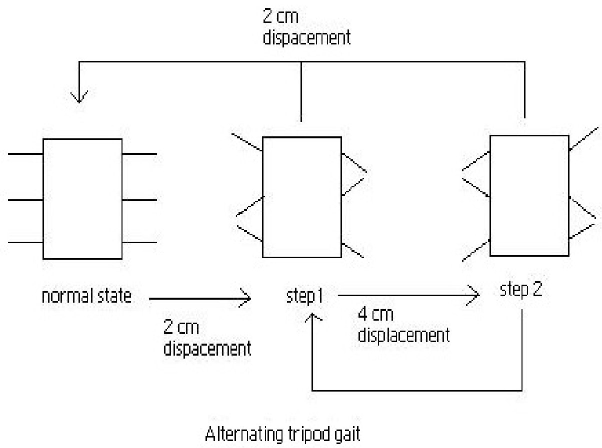

Figure 1 : Alternating tripod gait

3. Obstacle Identification & Map generation

In the simulator, a map of 10m by 10m is created. A scale of 2cm by 2cm represented by 1 pixel is chosen for a clear view. The user of the application can manually create obstacles on the map by drawing it in the same way as done in Microsoft paint application. The simulator identifies the arbitrary obstacles and bounds them by a fitting rectangle. In the memory map, these regions are marked as C_{obs} . The rest of the region is treated as C_{free} in which again C_{obs} are found depending on the size and orientation of robot at any given point. The snapshots of the simulator as shown in [Figue 2] give the detail of the above explained statements.

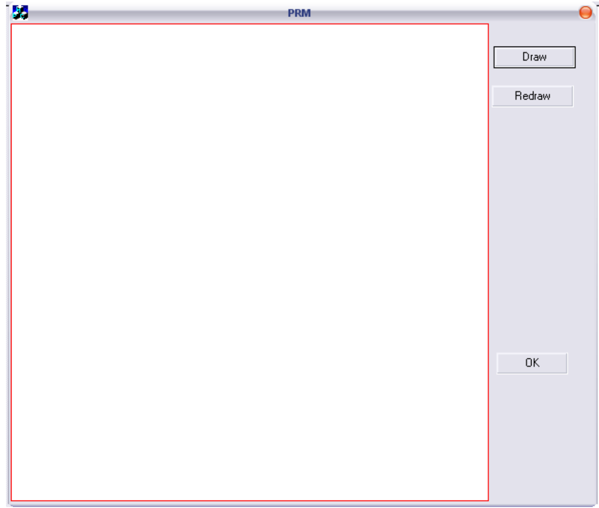

Figure 2.a

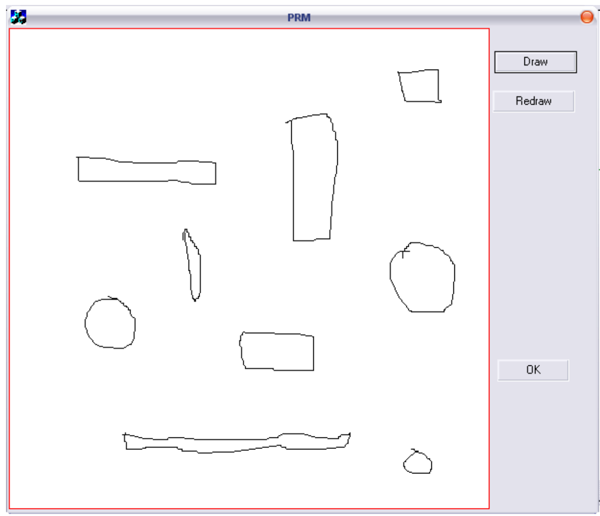

Figure 2.b

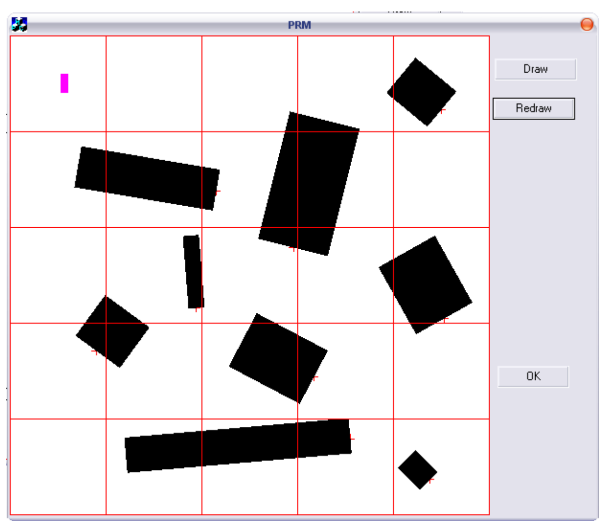

Figure 2.c

Figure 2 : a)10m by 10m map b) obstacles drawn free hand c) obstacles identified and bounded by fitting rectangles, small pink rectangle represents the robot.

3.1 Map Decomposition

The whole map is subdivided into twenty five segments as represented in [Figure 2.c]. Each segment is considered to be the visibility range of the robot. In the experimental example of this paper, the robot is placed at the center of the top left segment and has to reach the center of the bottom right segment. The roadmap method takes each segment independently and figures out the nearest node in the segment to the destination node. It is very obvious that as the number of segments increase, the path calculated becomes even shorter, but will be computationally heavy

4. Probabilistic Roadmap : Preprocessing phase

Each segment explained above is dealt separately. The preprocessing phase was done in two steps, roadmap construction and roadmap refinement. The roadmap construction step was the basis of the planner. The refinement phase tried to improve the roadmap by making connections that the construction step might have missed.

For the following algorithm description, let R= (N,E) be the roadmap graph. Let C define the whole configuration space consisting of C_{free} , the free configuration spaces and C_{obs} , the blocked configuration spaces. N is the set of nodes that correspond to configurations in C_{free}. E is the set of edges connecting the nodes from N. D: C x C -> $|R^{+}_{0}$ is a pseudo-metric (i.e. symmetrical and non-degenerate) that defines a distance between any two configurations.

4.1 Algorithm for generating the roadmap

```
1.   N:={};
2.   E:={};

     repeat
3.      q:=random configuration from C;
4.      if q is free (i.e. no collision) then
        begin
5.          add q to N;
6.          choose subset Nq of N with candidate neighbors for
q

7.          for all q' in Nq sorted ascending by D(q',q) do
            begin
8.              if local planner can connect q with q'
9.              then add (q,q') to E
            end;
        end;
10.  until limit is reached;
```

The algorithm is quite simple. Beginning with an empty roadmap (steps 1 and 2), it repeatedly adds nodes to the roadmap until a certain limit is reached (step 10). This limit is chosen based on obstacle complexity here. Another possibility would be to terminate the algorithm after a specified time has elapsed. Both methods don't take into account that depending on the given environment more time/nodes might be required for creating an adequate roadmap. The choice of limit is not crucial as the algorithm is incremental and thus can be called again anytime should it turn out that not enough nodes have been created.

To improve the roadmap nodes are added in difficult regions, especially if this allows connecting previously unconnected components within the graph. Again a probabilistic method is used. We used random-bounce walks. A configuration q is picked at random and from this starting point a path is generated by walking into a random direction till an obstacle is hit. On collision a different direction is chosen at random, and so on. This generates paths that have the potential of crossing difficult regions that the local planner failed at, like moving around a corner in a narrow path. We assign every node q in the graph a certain weight w(q) so that w(q) is high if q lies in a region where planning is hard for the local planner, it is usually a good idea to choose nodes with a high value for w(q) with a higher probability than those with lower weights. That way, random-bounce paths are not needlessly generated in regions where the local planner is sufficient. This is important because, unlike the paths generated by the local planner which can be regenerated any time by running the local planner again, the paths generated by random-bounce walks have to be stored explicitly in the roadmap. Just inserting an edge is not sufficient. An effective way that we used for storing a random-bounce path is to store the seed for the random number generator used. Once the new node q' that is the endpoint of the random-bounce walk, together with the path connecting it to q, has been stored in the roadmap, it should be attempted to connect q' with components previously unconnected to that of node q. There is the chance that the random-bounce walk has cleared a difficult obstacle and reached a point from where the local planner is able to connect to previously unreachable nodes. Connections are attempted with nodes close to q' from components not connected to that of q.

4.2 Real Time Gait Changing

Real time gait changing is done where the robot has to change direction of its motion or has to negotiate with a narrow passage. Each time local planner fails to connect any two nodes it makes a second attempt by changing the gait through the narrow available passage. The alternating tripod gait usually increases the size of robot by 10 cm while the crab gait used for sideways walking keeps the size normal

(20 cm in length) , but the crab gait makes the robot motion slower by half the speed of alternating tripod gait.

4.3 Probabilistic Roadmap : Query phase

In the query phase, the roadmap answer requests for connections between given start and goal configurations. Let q_{start} be a given start configuration and q_{goal} be a given goal configuration. If q_{start} and/or q_{goal} are not among the configurations explicitly stored in the roadmap, the first step is to connect them to nodes q'_{start} and q'_{goal} within the roadmap. In order to do this for node q_{start}, nodes from the roadmap are chosen in ascending order of distance from q_{start}. If a q'_{start} is found that can be connected to q_{start} by the local planner, the same is attempted for q_{goal}. If it is not possible to find appropriate q'_{start} and q'_{goal} configurations, random-bounce walks is attempted starting at q_{start} and q_{goal} in order to reach configurations from which the local planner can connect to suitable q'_{start} and q'_{goal} respectively. If this also fails the whole query fails.

If appropriate configurations q'_{start} and q'_{goal} are found, the next thing is to connect q'_{start} and q'_{goal} within the graph. This yields a sequence of waypoints. Finally the requested path can be constructed by using the local planner to regenerate the path from q_{start} to q'_{start} and to regenerate the paths between the waypoints. Finally the path from q'_{goal} to q_{goal} is generated by reversing the path from q_{goal} to q'_{goal} that is regenerated using the local planner.

5. Results

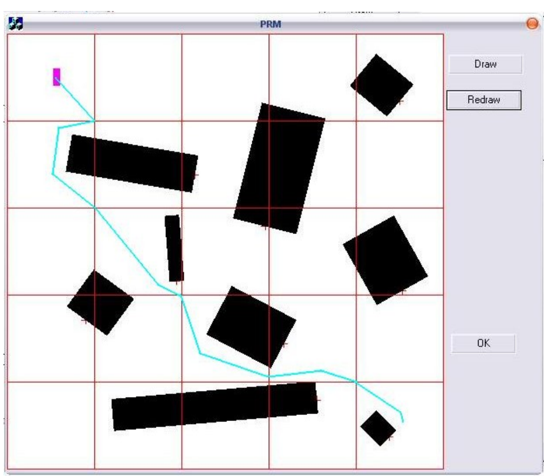

Figure 3.a

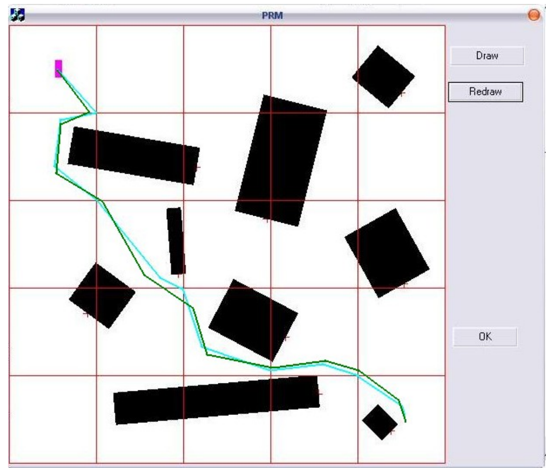

Figure 3.b

Figure 3 a) path generated in simulator b) deep green path is the path followed by the robot

6. Conclusion

As it is seen from the [Fgure 3.a] the simulator generates a path each time attempting to take the robot to nearest point to the destination in that particular segment. The [Figure 3.b] has a green path made which is the path followed by actual robot on the arena given the simulated obstacles. The minor errors in the actual robot path is due to small localization errors on the robot. With the increase in the number of segments of the map, the path followed by the robot becomes smoother.

7. References

[1] L. Kavraki, P. Svestka, J.-C. Latombe, and M. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pp. 566–580, 1996.

[2] Frank Lingelbach, Centre for Autonomous Systems, Royal Institute of Technology, SE-100 44 Stockholm, Sweden. "Path Planning using Probabilistic Cell Decomposition" Proceedings, ICRA '04. 2004 IEEE International Conference on Robotics and Automation, 2004

[3] L. K. Dale and N. M. Amato, "Probabilistic roadmaps - putting it all together," in Proceedings of the International Conference on Robotics and Automation (ICRA), 2001, pp. 1940–1947.

[4] Visibility based probabilistic roadmaps for motion planning, T Simeon, J.-P. Laumond. and C. Nissoux. To appear in Advanced Robotics Journal 14(6) (2000).

[5] S .T. Venkataraman Department of Electrical Engineering,Yale University,"A Model of Legged Locoimotion Gaits", Proceedings of the 1996 IEEE International Conference on Robotics and Automation Minneapolis, Minnesota - April 1996.

[6] Edward Z. Moore, Department of Mechanical Engineering,McGill University, Montreal, Canada' "Leg Design and Stair Climbing Control for the RHex Robotic Hexapod" A Thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements of the degree of Master of Engineering,January 2002

8. Acknowledgement

This work was supported by Fundação para a Ciência e a Tecnologia (ISR/IST pluriannual funding) through the POS_Conhecimento Program that includes FEDER funds.

9. Brief curriculum vitae

Aamir Ahmad
Graduate Student (2008-2012)
Institute for Systems and Robotics,
Department of Electrical and Computer Engineering,
Instituto Superior Técnico, Lisbon, Portugal

B-Tech (Hons.), 2008
Department of Civil Engineering
Indian Institute of Technology, Kharagpur, India

Dr. Nirjhar Dhang
Ph.D
Associate Professor
Department of Civil Engineering
Indian Institute of Technology, Kharagpur, India