



PATH PLANNING USAGE FOR MOBILE ROBOTS

Edwards Valbahs

Riga Technical University, Department of System Theory and Design
Meza street 1/4-547, Riga, LV 1048, Latvia; ph.: +(371)67089584, e-mail: err@inbox.lv

Abstract. *The paper deals with path planning programme for mobile robots. The aim of the paper is to develop a path planning programme for mobile robots that is based on simulation software development. The results of the careful analysis were considered for optimal path planning algorithms. The experimental evidence was proposed to demonstrate the effectiveness of the algorithm for steady covered space. The results described in this work can be extended in a number of directions, and applied to other algorithms.*

Keywords: *robotic, robot, RRT, RRT*, path planning.*

Introduction

In order to achieve the wide range of the robotic application it is necessary to provide iterative motions among points of the goals. For instance, in the industry mobile robots can replace any components between a storehouse and an assembly department. Ammunition replacement is widely used in military services. Working place is possible in ports, airports, waste site and etc. Mobile robots can be used for monitoring if it is necessary to observe control points in the secret place. There are a lot of scenarios where the iterative motion is applied [1].

The environment for such mobile robots used is complicated, various, non – structural and dynamic by nature. Robots must get rid of the obstacles which are different in size, form, location and can appear or disappear at any time. At that time avoiding the obstacles the possible of collision risk, sensor information and movement planning imprecision, error location and uneven surface are always foreseen. Robot can be damaged, trapped or embed in any construction etc. Environmental imprecision is always very hazardous for mobile robots [2].

At the same time the effective utilization is required from robots. Robots should work as much as possible and fast and safely [1, 3, 4]. Although the safety of people and the place of their living is primary and the same can be said about the safety of robots as well.

Algorithms for motion planning have proved themselves as optimal methods in this planning. The best conditions are normally measured with a distance. But it is possible to measure the conditions mentioned differently. For instance, in order to find safe ways space robots take into consideration surface, roughness and slope. However, the effectiveness of complex, dynamic and partly unknown spaces was not investigated for a long time. Nowadays, it is little research in the choice of motion planning in the dynamic conditions. Approaches [5, 6, 7] admit that environmental structure is known as *a priori*. In source [8] it is admittable that unknown environment is static and unchangeable during some period of time. Source [9] reflects uncertainty except the two previous statements.

Commonly, in order to work out any motion planning system which is used mobile robotic platform and it is necessary to do the following tasks:

- to familiarize with the essence of motion planning task i.e. to observe the algorithm of classical and modern planning;
- to identify the advantages and disadvantages of the algorithm;
- to select the most relevant algorithm;
- to work out system design.

The representation of working area for the robot can vary from continuous geometrical behavior to approved decomposition of topological maps. The first step for any motion

planning system is permanent environmental model transformation in the map relevant to the motion planning algorithm chosen if it is possible. The motion planning distinguishes due to the influence on the discrete decomposition. The three basic decomposition strategies can be mentioned [1]:

- Motion maps: to show the package of motion in free space;
- Unit decomposition: to distinguish empty from occupied units;
- Potential field: robot reaches the target unit if it follows the least resistance direction. That function is sometimes called as navigation function if additional conditions are added.

Then some motion planning algorithms can be briefly envisaged i.e. RRT and RRT*.

Goals

According to the mentioned above statement to compare RRT and RRT* algorithms by means of simulation is becoming rather interesting.

The aim of the research paper is to analyse motion planning algorithms that contain the design of modelling programme. The programme is needed as environment modelling to obtain the simulation data. The programme product is based on RRT algorithm. The simulation data give the possibility to conduct the wide analyses for selected algorithm. Analysis means the simulation data interpretation and comparison with other data obtained using the RRT* algorithms for motion planning.

Methods

RRT – in the last decade the motion planning algorithms on the basis of step sample have been proved as efficient ones and worked out theoretically as probable definiteness. The theoretical limits of these algorithms have not solved so far. It is proved that RRT algorithms always converge but do not insure the optimal magnitude.

RRT* - the algorithm always converges and insures the optimal motion planning. RRT* saves asymptotical optimality and holds the tree structure as RRT. Both algorithms are different in means of construction of motion planning maps [1, 10].

The use in practice and the necessity of it is very connected to autonomous robots that move in the space and are able to plan route on their own. One of such robots existing in our everyday life is autonomous vacuum cleaner. Autonomous vacuum cleaners do not usually use the motion planning algorithm. They are based on different simple algorithms, for example cleaning in a spiral, crossing the premises avoiding the walls and their moving is casual after touching the walls. The philosophy of this design was proposed by the scientists of Massachusetts Institute of Technology. Robots must be like insects having primitive controlling devices aimed at the environment. As a result, an autonomous vacuum cleaner is very effective in cleaning premises, but much more time is required for them as compared with work made by a person. There is a drawback, the autonomous vacuum cleaner tidies one and the same places many times but other places are cleaned only once. The use of motion planning algorithms can raise the effectiveness of an autonomous vacuum cleaner.

Assumptions

In order to fulfill the aim of the research paper the following conditions are introduced for:

- premises where an object moves;
- robot (or object) moves around the premises;
- path the robot moves on in the premises.

The premises are presented as two – dimensional plane. The plane of premises is equally divided into the checks. The check dimensions are equal to robot dimension that moves in the premises. The robot moves only one check forward and back. During one motion the object

can move to the one check from empty eight ones (eight checks around one check) paying attention to that check is not occupied by the obstacle. RRT algorithm is introduced to the programme product as the only one which calculates the motion planning task (fully covers all empty space).

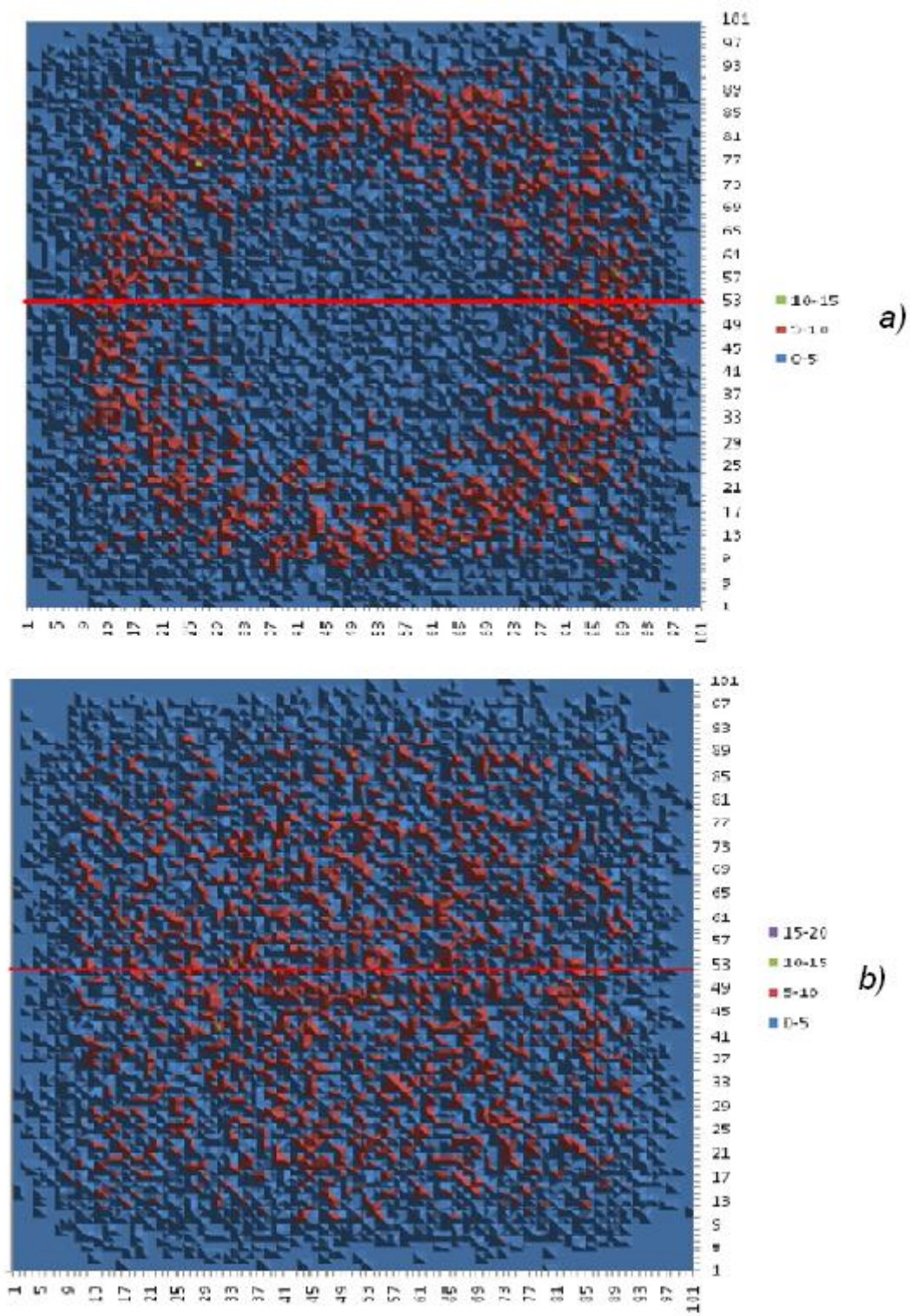


Fig.1. Density of the covering (a – for RRT; b – for RRT*; and where a red line is longitudinal section)

The results were compared with RRT* algorithm data. In order to receive the data based on RRT* algorithm, the new programme product was not developed from the very beginning but

an open source programme was extended and coded in C programming language. The changes had to be made because the programme calculated the path of motion planning between two points. After the first section which is the only the programme did not continue the motion planning i.e. to choose the next point freely from empty points (with the way covered or obstacle). A drawback of the programme (taking into account the requirements claimed before) was successfully kept away extending the programme code.

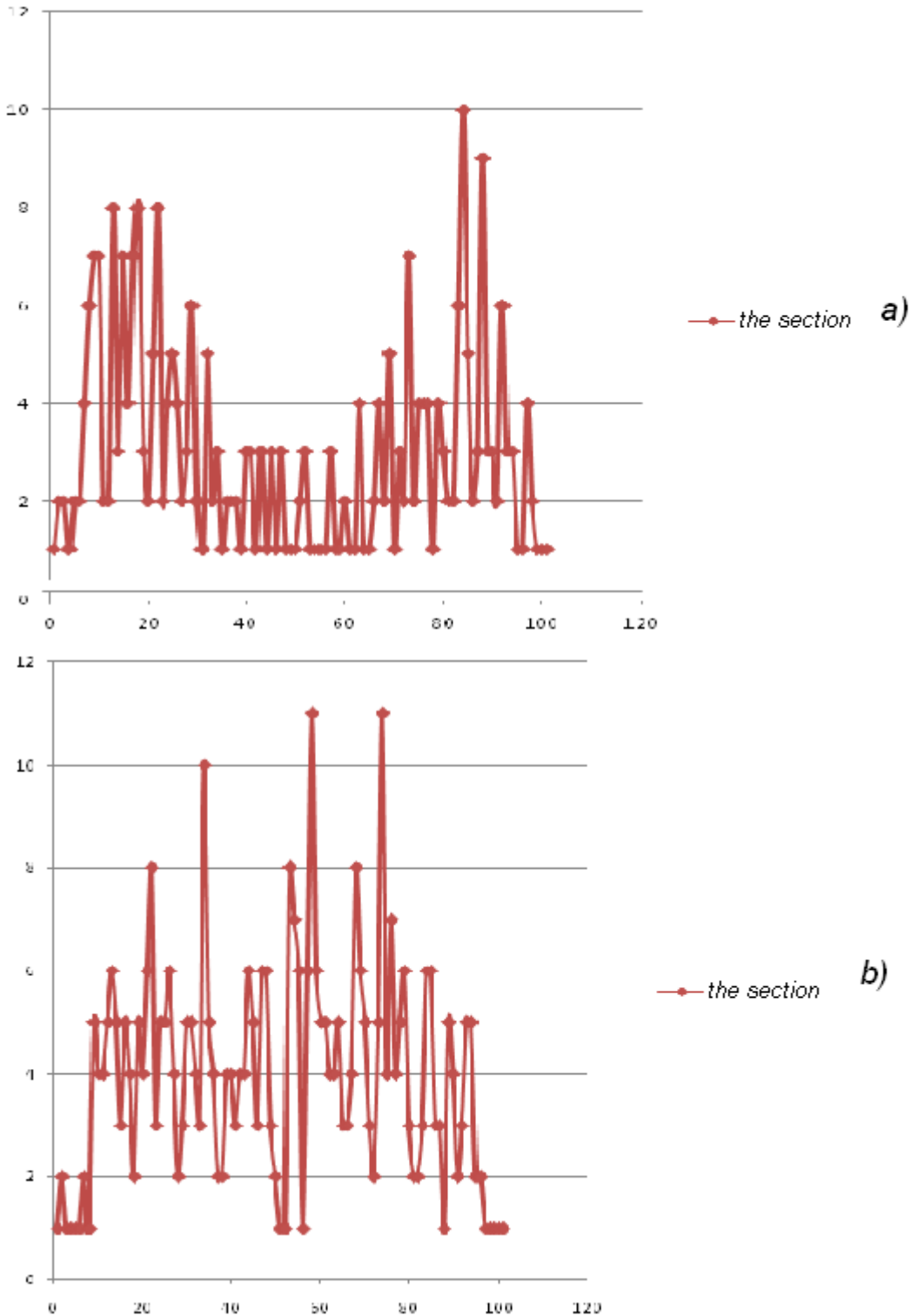


Fig.2. Density of the covering in longitudinal sections
(see Fig.1; a – for RRT; b – for RRT*)

In this research paper both algorithms were compared practically using and combining different placement of obstacles in the unchangeable two-dimensional space. All the results were obtained on one and the same computer (2.66 GHz processor and 2GB RAM), operating systems (Microsoft Windows XP or Ubuntu 10.10 Linux were used). The following information was collected about:

- the number of covering for each check;
- the time which was necessary for both algorithms to plan the route.

The given illustrations show the density of the covering (how often a robot covers each check), but the cutting is demonstrated with a red line on a definite axis (Fig.1 and 2).

Results

Different coverings of the premises using RRT and RRT* algorithms in the identical conditions are shown in the research paper. The measure of the premises during simulation was not changed (100 x 100 checks) but the dimensions, placement and the number of obstacles in the premises were changed. The simplest case is when the premises do not contain any obstacles i.e. completely empty. So, for example the simulation was done in the empty premises using the both algorithms. This process was repeated several times. It is very important to stress that the principle of previously mentioned process was used for the placement of any obstacles in the premises i.e. not only in the case of utterly empty premises. The simulation data were calculated as average ratio to the number of simulation repetition. During each simulation the following characteristics were calculated:

- the number of initial empty checks;
- the time which was necessary to plan the route to cover the whole premises;
- the total number of steps taking into account the facts that a robot can move one and the same check several times.

Table 1.

Some typical comparison values between RRT and RRT*

Number of free checks	t_{mp_RRT} (where t_{mp} - period of time of motion planning)	N_{RRT} (where N - number of steps)	$t_{mp_RRT^*}$ (where t_{mp} - period of time of motion planning)	N_{RRT^*} (where N - number of steps)	$t_{mp_RRT^*}/t_{mp_RRT}$	N_p (%) (where N_p - least number of steps for RRT*)
10201	185	32303	1020	31782	5.51	1.62
7800	170	26064	600	4283	3.53	6.83
4582	172	14147	960	12761	5.58	9.90
5191	180	12596	1000	10829	5.56	14.03
5510	180	11441	840	9681	4.67	15.38

Periods of time of motion planning for RRT and RRT* algorithms were compared (i.e. RRT* period of time were divided with RRT interval of time). The results of division were between 3.53 and 5.58 times for different placements of obstacles. It means that motion planning for RRT algorithm is 6.14 times faster at least. It is very important to stress that periods of time of motion planning were between 3 and 15 minutes (15 minutes applies to RRT*). Route was planned better with RRT* in comparison with RRT and premises were covered more regularly that was observed from Surface diagram. RRT* showed the better results than RRT did in case the number of occupied checks were increased. The least number of steps was needed in the case of RRT*, i.e. between 1.62% and 15.38% least number of steps. It is very important to stress that the number of steps was oscillated between 9681 and 32303. The total

period of time for route can be between 9681 and 32303 s (2.69 and 8.97 hours) if considering that real robot performs 1 step a second (Table 1).

Conclusion

The period of time for route is bigger than period of time for motion planning and it restricts all the process. The period of time for route planning can be decreased if computational power is increased. That is why the effectiveness of robot depends on effectiveness of route planning, i.e. from the number of steps. The using of RRT* can improve real robot effectiveness from 1.67% to 15.38%. It is very important to stress that effectiveness of process is strongly connected to energy saving and it is possible to prove that 15.38% of electric power can be saved.

Both algorithms can be successfully used in the requirements of autonomous tasks for robots, military technology and dynamic games.

Therefore, it is possible to use motion planning algorithms in other fields of studies which are out of route planning.

References

1. Roland Siegwart, Illah R. Nourbakhsh. Introduction to Autonomous Mobile Robots, A Bradford Book The MIT Press Cambridge, Massachusetts London, England, 2004.
2. Agris Nikitenko. Knowledge-based robot control, International Journal "Information Theories & Applications". Vol.15, 2008, 126 p.
3. Valdis Laan. Pullbacks and flatness properties of acts. Tartu, 1999, 90 p.
4. Krista Fisher. Structural mean models for analyzing the effect of compliance in clinical trials. Tartu, 1999, 124 p.
5. Mart Poldvere. Subspaces of Banach spaces having Phelps' uniqueness property. Tartu, 1999, 74 p.
6. Jelena Ausekle. Compactness of operators in Lorentz and Orlicz sequence space. Tartu, 1999, 72 p.
7. Helger Lipmaa. Secure and efficient time-stamping systems. Tartu, 1999, 56 p.
8. Jan Villemson. Size-efficient interval time stamps. Tartu, 2002, 82 p.
9. Olga Sokratova. O-rings, their flat and projective acts with some applications. Tartu, 2000, 120 p.
10. S. Karaman and E. Frazzoli. Incremental Sampling-based Algorithms for Optimal Motion Planning. *Int. Journal of Robotics Research*, 2010.