CS3105 Artificial Intelligence

AI in Action
Practical 1

Aim
This practical aims to find paths for obstacle navigation by a Robot using Potential Fields and Rapidly Exploring Random Trees (RRT's).

Objectives
After doing the practical you should be able to
- Design, implement, and document Potential Field and RRT approaches.

Specification
Robot overview
Suppose a robot has to navigate a 2-D obstacle course to a goal G. The robot is to plan its motion using either potential fields or RRT's throughout each journey according to a user command given at the start of the journey. The potential field planner is to use sensors able to detect obstacles and the 2-D goal location and the RRT planner is to use a global map.

The robot's heading is given by its last transition or by an initial direction towards the goal if no transition has yet been made. The line perpendicular to the heading divides the forwards from the backwards directions. The robot moves forward at all times. The simulation can be kept simple at first by making the robot and obstacles have sufficiently small size to not significantly impact on the shape of the paths. For RRT's, a given tree factory should be used to support tree creation. For both planners, a supplied Graphical User Interface is to be used.

Part 1 Free Space Travel
Design and implement a Robot motion planning and execution system in Java, capable of being *complete* in a significant region of Free Space, i.e. so that a goal-connecting path can be found between any pair of locations in the region. Use the given Graphical User Interface to show development of the planned and executed paths.

For both planners, avoid the robot spinning on the spot during execution by using multiple turn-and-go movements. Give an idea of how efficient the two planners are in Free Space by reporting on the number of moves made and the lengths and total turns of the paths formed. For the RRT, report on (i) the ratio of unused to used nodes for execution, (ii) the effect of adding goal bias. For both planners, report on the principles by which the move parameters are set. Which of the two planners appears best to you for Free Space travel and why?

Part 2 Obstacle Navigation
Extend your system to obstacle navigation. Use the above suggestion for an initial simple simulation to make a preliminary assessment of how efficient RRT's and Potential Fields are. After this, try obstacles and robots of sufficient size to significantly affect the shape of the travel path and see if your assessment changes. What problems occur for RRT's when obstacles are not known except through local sensing?

Create a select range of obstacle courses to show your system working and to illustrate the pro's and con's of the two planners. Report on the efficiency shown across the range. Which of the two planners appears best to you for obstacle navigation and why?

Various enhancements are possible, e.g.
(i) Obstacles and robots of differing sizes could be tried as could obstacles of various shapes and numbers.
(ii) The working of the Potential Field planner could be shown by displaying the forwards semi-circle in use at any moment and the sensor rays hitting the nearby obstacles as the robot moves.
(iii) The RRT planner could be made to rely just on the same sensors as the Potential Field.

Overview Questions
How would you develop your system further to deal with the problems you found with your final system?
How would you develop your system to deal with moving obstacles?

Report requirements

- For handing in, you should supply your code together with sufficient I/O behaviour from them to demonstrate that they work correctly. The programming language is Java and the program submitted should be a jar file. Name the jar file as "Robot.jar" and ensure it runs using a double-click if possible and at least "java –jar Robot.jar". Your source code should be well structured and well commented. Make sure you also include the answers to the questions.

- You need to state if your program works fully, if it has enhancements, or if there are non-working aspects. If the latter is the case, you should provide clear details.

- Documentation should as usual be such as enables straightforward understanding and running of the programs on a specified lab machine. A word limit of 4800 words is not to be exceeded. Instructions for compiling and running *must* be confirmed through clear screen snapshots of them working. **If your instructions fail, your program will be treated as not compiling or running.**

- The start date for this practical is Wednesday of week 2, i.e. 4th February. An electronic copy of your program and report should be placed in MMS by Friday of week 7, i.e. 13th March. The documents must be zipped into a single archive for MMS.

Feedback and Marks:
See the School Handbook for the section on feedback and marks.

Remember: Late work will be penalised by MMS.

MKW
4/2/15