CS3105 Artificial Intelligence

AI in Action
Practical 1

Aim
This practical aims to find, compare, and contrast paths for obstacle navigation by a Robot using Potential Field techniques.

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

Specification
Robot overview
Suppose a robot has to navigate 2-D obstacle courses to reach various goal positions from various start positions. The robot is to plan and execute its motion using one of two types of potential field which is selected according to a user command given at the start of each journey. The potential field planner is to use sensors able to detect front-facing obstacle edges and the 2-D goal location rather than a global map.

The robot's heading is given by its last transition or by an initial direction 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 should 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. A supplied Graphical User Interface is to be used. Ensure the interface displays
(i) the directional forwards semi-circle in use at any moment and the sensor rays hitting the nearby obstacles as the robot moves at an easy-viewing speed.
(ii) the relevant planning path segments in use at any moment.

Part 1 Free Space Travel
Design and implement an extended Robot motion planning and execution system in Java, capable of being *complete* in a significant region of Free Space. That is, complete in the sense that a goal-connecting path can be found between any pair of locations in the region using any start orientation. Use the given Graphical User Interface (GUI) to show development of the planned and executed paths. One (greedy) field planner is given and uses (forwards) steepest gradient descent, and the second (non-greedy) field planner is to use fractional progress. Extend the given system to complete travel in free space using the second type of field (one field only for each run).

The robot is to move using a pair of wheels with one on each of its left and right sides. The robot is to move in arcs at continuous speed through a given linear version of differential drive. The goal potential used should be able to switch between two different forms of distance to goal between runs: one is linear Euclidean distance for the greedy field and the other is an arc-based distance for the non-greedy field. The arc-based measure contains three arcs at each stage, one arc for the next move and two further arcs for the estimated future path.

Compare the efficiency of the two planners where each uses the most appropriate of the two goal distance measures. Justify which is the most appropriate. Give an idea of how efficient each planner is in Free Space by reporting on the number of moves made and the lengths and total turns of the paths formed. Report on the principles by which the move parameters are set. Which is your optimal set-up across all the planners for Free Space travel and why?

Part 2 Obstacle Navigation
Extend your system to obstacle navigation with the same costing as for free space travel except that obstacle potential is now added in to the costing in appropriate ways for each type of field. Justify the appropriateness. The obstacle potential should use an exponential fall-off range. Make a preliminary assessment of how relatively effective and efficient the two types of field are using small robots and easily navigated obstacles. After this, try obstacles and robots of sufficient size to significantly affect the shape of the travel path and see if your assessment changes. This assessment does not need to be reported in detail, a brief summary of the main points will do.

Create a minimally sufficient range of obstacle courses to show your system working and to illustrate the pro's and con's of the two planners. The range should include static obstacles of the following type: circular, Shallow-C, Severe-C, for a variety of start and goal postions and initial heading. Use the given 5-point obstacle for the C-shapes.

Report on the effectiveness and efficiency shown across the range. Conclude as to which of the two types of planner appears best to you for obstacle navigation giving your reasons. Make it clear as to whether any reason is due to your particular settings or is an inherent feature of the field type.

Various enhancements are possible, e.g.
(i) Significantly different new obstacle types such as narrow passages.
(ii) Use of strain as a distance measure to generate low curvature paths.
(iii) Swap of a real differential drive for the linear version to see if the behaviour is affected significantly.

Overview Questions

What is/are the likely reason(s) as to why the set book by Russell and Norvig says potential fields suffer from the Local Minimum problem?
How would you develop your system further to deal with any problems you found with your final optimal system?

Report requirements

- Your report should contain, in screenshots and/or through simple fully explicit click-and-type running instructions, sufficient I/O behaviour to demonstrate that the program works as described. *Write your report so the attached check table elements are clearly and directly addressed by using the left hand column of the rows as section headings.* Incorporate the table into the report on its own page to show which aspects have been done and provide a cross-reference into the report describing the relevant aspect where feasible. 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. You must provide clear description of which files have been changed and where they have been changed. 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. You should provide clear details of these. Remember that there is a limited amount of time to spend on the practical so make sure you round off what you have managed to achieve in plenty of time to write up properly.

- 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 and a page limit of 8 pages (excluding the check table) 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 may well be treated as not compiling or running.**

The deadline for this report is 9pm, Friday week 7, 15th March.

What to hand-in: Submit the report, jar file, and source code, via MMS. The format of the report should be that of PDF. The submission must be zipped into a single archive for MMS.
Before submitting, remember that the University policy on Good Academic Practice applies:
https://www.st-andrews.ac.uk/students/rules/academicpractice/

Marking:
See the standard mark descriptors in the School Student Handbook:
http://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/feedback.html#Mark_Descriptors
This assignment is worth 50% of the coursework marks and 20% of the marks for the module as a whole.

Remember: Late work will be penalised by MMS, See
http://info.cs.st-andrews.ac.uk/student-handbook/learning-teaching/assessment.html#lateness-penalties

MKW
7/2/19

Robotics Practical 1 Check Table:

Part 1		Done?
Completeness	A goal-connecting path can be found between any pair of locations in the displayable region.	
Graphical User Interface	The interface shows development of the relevant paths.	
Potential and motion mechanisms	The elements used to make up the potentials and motion mechanisms involved are clearly described with justification.	
Efficiency	The number of moves made and the lengths and total turns of the paths formed by each measure is reported as per the spec.	
Move parameters	The principles for setting them is reported.	
Set-up superiority	Preference given for one of the set-ups with reasons.	
Part 2		
Obstacle courses	A wide variety of obstacle shapes and courses for testing.	
Potential and motion mechanism:	The elements used to make up the potentials and motion mechanisms involved are clearly described with justification.	
Efficiency	As for Part 1, but across the range of obstacle courses.	
Planner type superiority	Ranking of the planner types with reasons.	
Enhancements	Described, implemented, and reported.	
Code and documentation for Parts 1& 2	Commented source code and jar file supplied. *Clear AI design description in report* Sufficient I/O behaviour shown in report and/or run examples. Programming working and enhancements statement. Answers to all questions. Word and page limits adhered to. Typical screen snapshot of running program with statement as to how to run program. Documentation is clear upon read-through. Documentation appears comprehensive upon read-through. Documentation shows understanding and effort upon read-through.	