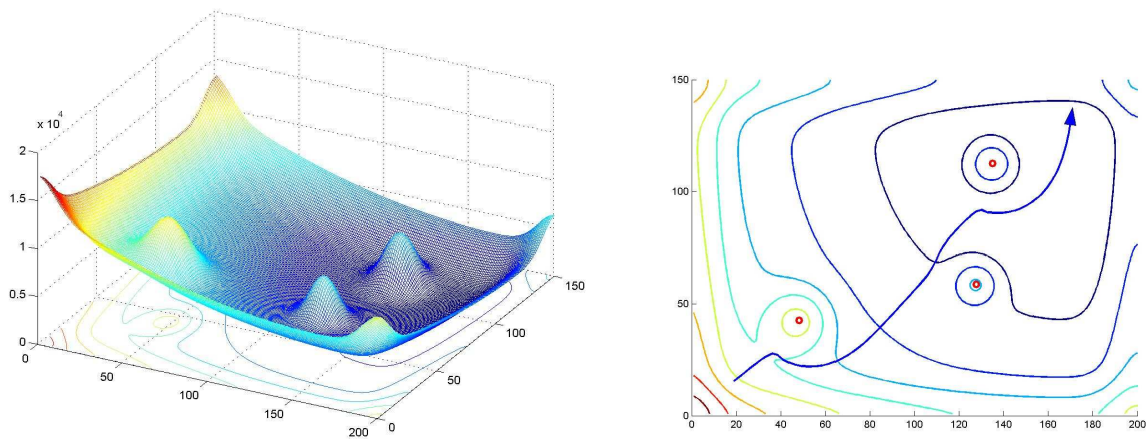


**CSE 397/498-013**  
**Introduction to Mobile Robotics**

**Homework: Motion Planning via Potential Fields**

**Report Due Date: Tuesday, 20 Dec 05 submitted via Blackboard NLT 1045**  
**(NO ANALOG SUBMISSIONS)**



**Figure 1: Sample Potential Field Simulation**

**A. Objectives:** Following the model developed in R. Beard's notes, as well as the class lectures, you are to develop a potential field based motion planner that allows for autonomous navigation in a cluttered environment

**B. Requirements:**

1. You **MUST** use Matlab to complete this assignment.
2. This is an individual assignment. Each student is required to submit his/her own work in order to receive credit.
3. As stated on the web page, if you turn this assignment in late without coordinating with me first you will receive a 0.

**C. The Assignment:**

1. There is a skeleton `hwk7.m` file that everyone **MUST** use as the basis for their homeworks. Exceptions will not be accepted. Download this from the course web page.
2. Initial values for the motion model parameters are included in this `.m` file. You **MAY MODIFY THESE** as you deem necessary.

3. Assume a closed configuration space (the bounds of the figure are walls). This requires the simulator to model the goal position, obstacles and walls with APPROPRIATE potential functions.
4. Parameters for these functions can be chosen at your discretion. However, they should confine the robot trajectory to the configuration space (inside the figure) for all but the worst goal placements (e.g. on the wall).
5. The user will input obstacle positions from the mouse, a robot position, and then a goal position
6. We are assuming that the obstacles can be modeled as points.
7. We are assuming a non-holonomic robot. In other words, the robot **must** reflect orientation. You can do this by assuming that the robot is pointing in the same as the negative gradient of the potential field
8. You may assume that robot localization and motion are perfect (there is no need to introduce noise).
9. Your motion planner should update at a rate of 10Hz.
10. After user input, contours of the potential function are to be drawn on the figure (see matlab function contour) and the robot will follow the trajectory dictated by the potential field (example in Figure 1 (right)).
11. Plot the surface-contour plot of the potential function used for the simulation (example in Figure 1 (left)) using the built in Matlab meshc function.
12. The simulation will terminate once the robot reaches a certain proximity to the goal, or when a local minimum is detected (constrained to a certain proximity for several iterations movement).

**D. Turn In:** A write up, to include:

1. Images from simulation trials.
2. Your Matlab source code.
3. Answers to the following additional questions:
  - a) Your fellow student makes the observation that you could improve the performance of the potential field by instead of using exponentials which possess local minima, using a potential function with only a

single minimum even in the presence of obstacles. Do you think this would be possible? Why or why not?

- b) List possible solutions for recovering from local minima.
- c) How would you change the mathematical representation for obstacles if you had to model them as two-dimensional shapes?
- d) Explain how you might adapt your potential field model for local obstacle avoidance (without the presence of a map).