Assignment 3

The objective of this assignment is for you to become familiar with the Turtlebot 2 robot.

1. Ensure that your assigned robot is functional.

Look over the tutorials available in:
http://wiki.ros.org/Robots/TurtleBot
Your robot is using the create base.
Follow the following articles:
http://spectrum.ieee.org/automaton/robotics/diy/new-turtlebot-tutorials
http://spectrum.ieee.org/automaton/robotics/diy/turtlebot-2-unboxing
http://www.cse.sc.edu/~yiannisr/574/notes-turtlebot.pdf

Especially the following link:
https://cse.sc.edu/~yiannisr/574/2018Fall/notes_turtlebot2.pdf

Follow the tutorials, in particular the depth image to laser scan:
http://wiki.ros.org/depthimage_to_laserscan

The robots are using the create base but they have a Raspberry Pi computer the same as the ones using the Kabuki base.

2. Random Walk 50%

Modify the code from Assignment 1 to perform random walk with the actual robot. Your robot should never hit an obstacle. Always ensure you have adequate clearance.

3. (Individual) Potential Fields Path Planning (start) 50%

Create a new ros package, then place potential_field.cpp into ./src/

Instructions: search for the 2 occurrences of "TODO" in the source code, and then address them.

Use a potential field to guide a robot to a user selected destination. Use the following functions:

a. **Attractive force:** $\|F_a\| = \gamma \cdot \|X_{goal} - X_{robot}\|^2$
b. **Repulsive force (per sensor reading):**

$$\left\|F_r^i\right\| = \begin{cases} \dfrac{\alpha}{\left(d_i - d_{safe}\right)^2} & \text{if} \quad d_{safe} + \varepsilon < d_i < \beta \\ \dfrac{\alpha}{\varepsilon^2} & \text{if} \quad d_i < d_{safe} + \varepsilon \\ 0 & \text{otherwise} \end{cases}$$

c. Use vector summation for the forces of all sensor readings as well the attractive force: $F = \sum_i \vec{F}_r^{\,i} + \vec{F}_a$

where γ, α, β, and ε are constants defined by you and d_{safe} is a short safety distance. For example d_{safe} = 0.5 m, $\gamma = \alpha = 1$, $\beta = 4$ m, and ε = 0.05 m.

Local Minima: In order for the robot to move out of local minima, introduce a random force. Monitor the robots position and increase the magnitude of the random force the more the robot stays near the same place.

From forces to velocity commands:

- **Angular velocity**: $\omega = \kappa(\theta_F - \theta_r)$, where κ is a scaling constant, θ_F is the orientation of the cumulative force, and θ_F is the orientation of the robot. *Hint*, use the atan2 function for your calculations, during subtraction take into account the 0, 2π discontinuity.
- **Linear velocity**: project the cumulative force along the orientation of the robot and use the projected force. Remember to truncate accordingly, to account for maximum allowed speed. If the vector points behind the robot, set the linear velocity to 0.

Use the source code "*potential_field.cc"* as a starting point command line parameters are: "id #robots destX destY": the robot's id; the team size; the X coordinate of the destination; and the Y coordinate of the destination. Experiment with "*cave_single.world*", and "*obstacle_single.world*" worlds.

4. Bonus question 20%

Consider negative obstacles such as steps or drop offs. Monitor the floor plane and anything that drops lower than a few cm consider it an obstacle. Test you code in the lab, ensure that you are close enough to the robot to catch it if it miscalculates. Stay out of the field of view.

Evaluation:

Prepare a report presenting screenshots of your work, provide a brief overview on how the program works and also discuss the choices made. Place the report in the base directory with your name and zip everything into a single file. Submit in the dropbox. I will arrange with every team to see a demo of the random walk behaviour.