Assignment 2, Due November 3rd, 2010
(Worth 10%)

1. **20%**) Using the occupancy grid produced from the real sonar data in assignment 1, plan an efficient path from the location of the first sonar scan to the location of the last sonar scan using a Wavefront planner.

2. **20%**) 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):** $\|F_r^i\| = \begin{Bmatrix} \dfrac{\alpha}{(d_i - d_{safe})^2} \text{ if } d_i < \beta \\ 0 \text{ otherwise} \end{Bmatrix}$
    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, and β are constants defined by you and d_{safe} is a short safety distance, for example 0.5 m.
Hint, check for division by zero.
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. *Alternatively*, keep a history of the locations visited by the robot and add a repulsive force proportional to the time spend on them.

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 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.

Use the source code "*leader.cc*" as a starting point input parameters are: "-h localhost" the machine player runs on; "–p 6665" the port player listens to; "-i 0" the leaders id; "-n 1" the team size; and "-destX, -destY" the destination coordinates defined by the user. Experiment with "*cave_single.cfg*", and "*obstacle_single.cfg*" worlds.

3. **10%**) Using the potential field of question 2, implement a random walk by selecting a new destination randomly at short intervals. The only modification is at regular intervals, every 5 seconds for example, create a new goal:

   $X_{goal} = \left[x_r + r\cos(\theta) \quad y_r + r\sin(\theta)\right]^T$ where $r = rand(1) + 0.5$, and $\theta = rand(0, 2\pi)$

   Use the code from question 2, no need to enter a destination. Experiment with "*cave_single.cfg*", and "*obstacle_single.cfg*" worlds.

4. **50%**) Use a potential field to guide a team of robots to swarm and follow a leader robot. The leader (robot0) should implement the behavior from question 3. The rest of the robots (followers) should use repulsive forces for obstacles and also for the other followers. Robots are detected by sonars. Filter out the sonar returns from the leader (do not be repulsed by robot 0). The attractive force for robot "j" should be:

$$\left\|F_a^j\right\| = \left\{ \begin{array}{ll} \dfrac{\alpha}{d_{j\to 0}^2}, & \text{if } d_{j\to 0} < d_{safe} \\ 0, \ \text{if } d_{safe} < d_{j\to 0} < d_{far} & \\ \gamma \cdot d_{j\to 0}^2, & \text{otherwise} \end{array} \right\}$$

   Where $d_{j\to 0}$ is the distance between $robot_j$ and $robot_0$ minus twice the radius of the robots. A safe distance could be 1 m, and a too far distance would be 3 m. The above are approximate numbers.

   Extend the code from question 3, create a new program "follower.cc" input parameters are: "-h localhost" the machine player runs on; "–p 6665" the port player listens to; "-i [1-3]" the followers id; "-n 4" the team size. Run the leader program and three follower programs in different terminals (or tabs). Experiment with "*cave_multi.cfg*", and "*obstacle_multi.cfg*" worlds.