ME652: Mobile Robotics
Homework #5 (10 points)

Due: June 29 (Mon), 2020

Problem 1. (5 points) Consider the equations of motion for a 2D kinematic vehicle.

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{V} \end{bmatrix} = \begin{bmatrix} V\cos\psi \\ V\sin\psi \\ u_1 \\ u_2 \end{bmatrix}$$

Implement a SLAM algorithm using an extended Kalman filter (EKF) using the above-shown kinematic model and the measurement equation given below

$$\dot{\mathbf{x}} = \begin{bmatrix} \psi \\ V \\ \beta_1 \\ \rho_1 \\ \beta_2 \\ \rho_2 \\ \vdots \end{bmatrix} + \mathbf{v}$$

where β_i is the bearing angle and ρ_i is the range to the ith landmark. The vehicle starts from (1,1) with the heading angle of 45° and its initial position is assumed to be known perfectly (*i.e.*, zero initial position uncertainty, P(1,1) = P(2,2) = 0). In addition, a total of five landmarks are assumed, and their initial position estimates are given with non-zero uncertainty values. For this assignment, a skeleton code (SLAM_skeleton.m) is provided, which contains the detailed settings of the filter parameters.

(a) Create your own SLAM code using the given skeleton code (SLAM_skeleton.m).
(b) Try different values for the sensor accuracy settings, and then discuss how the performance of SLAM changes.

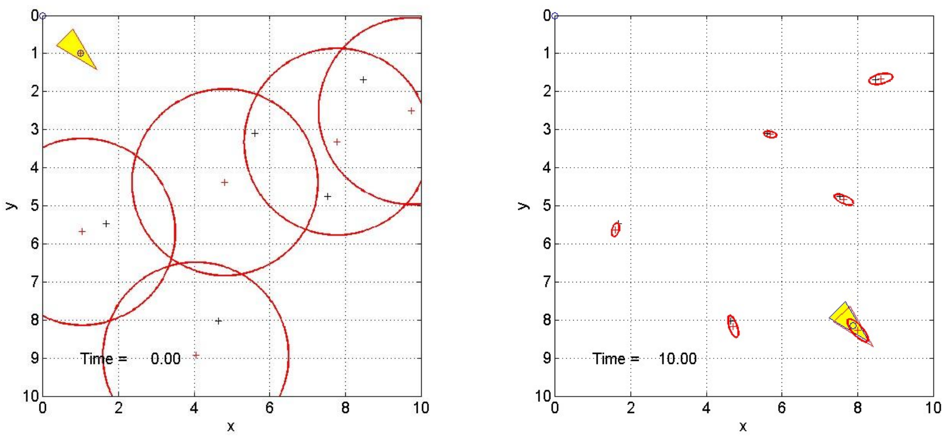
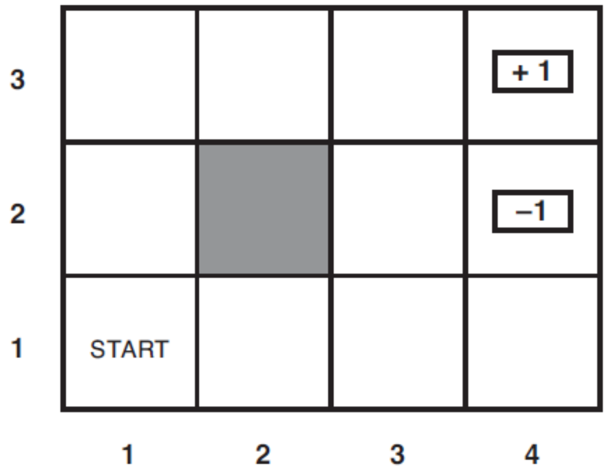
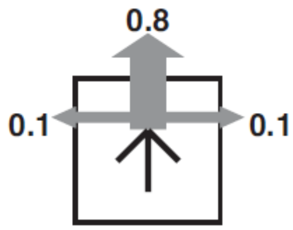


Figure 1 Example plots are shown with 95% confidence ellipses in the above. The initial setting is shown on the left, and an example SLAM result is shown on the right.

Problem 2. (5 points) This is the canonical MDP problem we are going to discuss in class. You are required to implement the MATLAB code for value iteration and policy iteration using the provided skeleton code. Suppose that a robot is situated in the 4×3environment shown in the following figure.





Beginning in the start state, the robot must choose an action at each time step. The interaction with the environment terminates when the agent reaches one of the goal states, marked +1 or –1. The actions available to the agent in each state are Up, Down, Left, and Right. Each action achieves the intended effect with probability 0.8, but the rest of the time, the action moves the agent at right angles to the intended direction either to the left with probability 0.1 or to the right with probability 0.1. If the agent bumps into a wall, it stays in the same square. Assume that the discount factor $\gamma = 1$.

(a) Complete the MATLAB code for value iteration by filling out two functions "value_iteration" and "extract_policy" in the provided skeleton code (MDP_valueIteration_skeleton.m). Set the immediate reward $\mathrm{R} = -0.04$ for all non-terminal states, and check if your results are correct (see below).

Optimal value function and optimal policy (R=-0.04)

```
0.8116    0.8678    0.9178    1.0000          "R"    "R"    "R"    "X"
0.7616         0    0.6603   -1.0000          "U"    "X"    "U"    "X"
0.7053    0.6553    0.6114    0.3879          "U"    "L"    "L"    "L"
```

(b) Investigate the change of optimal value function and policy by changing the value of reward in non-terminal states and discount factor γ. Then, discuss the results.

(c) Implement the MATLAB code for policy-iteration by modifying the code you implemented in (a).

* Submit the MATLAB code for each problem.