Probabilistic Mobile Robotics - EE698G

Practice Problems

---

1. **Radar detection**: If an aircraft is present in a certain area, a radar correctly registers its presence with probability 0.99. If it is not present, the radar falsely registers an aircraft presence with probability 0.10. We assume that an aircraft is present with probability 0.05. What is the probability of false alarm (a false indication of aircraft presence), and the probability of missed detection (nothing registers, even though an aircraft is present)?

2. John is driving from Boston to the New York area, a distance of 180 miles. His average speed is uniformly distributed between 30 and 60 miles per hour. What is the PDF of the duration of the trip?

3. When a matrix is said to be diagonalizable? Is symmetric matrix always diagonalizable ? If so prove it.

4. Prove the following :
   (i) Eigen vectors corresponding to distinct eigen values of any square matrix are independent.
   (ii)Eigen vectors of a symmetric matrix are always orthogonal.

5. Comment on the performance of all filtering techniques (discussed in class) in solving SLAM problem in terms of their order of complexity (i.e $\mathcal{O}(.)$). Report various issues relating inconsistency with EKF-SLAM.

6. Explain the nonlinear least squares formulation of Graph SLAM and obtain its standard least squares approximation by the method of linearization, i.e express in this form $\boldsymbol{\theta^* = \arg\min_{\theta} \|A\theta - b\|^2}$ and define clearly the matrices A, b

7. In this programming assignment you have to perform EKF-SLAM, while a mobile robot is traversing in a static environment (with landmarks). You are given snippets of MATLAB files that help you in getting the task done. All you have to do is to code the **predict.m, update.m, dataassociate.m** steps of the filter.

   Note :

   - You have to run '**main.m**' file after coding the above mentioned files.
   - Open '**ekf_slam_sim.m**' and go through the comments while observing the code, to understand the flow. In these files you will encounter functions '**predict**', '**update**' and '**dataassociate**' that are to be coded.
   - The rest of the files contain some helpful functions. Interested people can go through them. To gain good understanding of the flow, do read the comments in each file.
   - The configuration of the EKF-simulator is managed by the script file '**configfile.m**'. You have the flexibility to alter the parameters of the vehicle, sensors, etc.
   - You can really play with the simulator by changing the parameters in '**configfile.m**'. If you don't disturb this file the filter runs with the default parameters.
   - For augmenting the new landmarks to the state vector **'augment.m'** is already implemented.

   For you information the true robot path and the estimated robot path will be automatically saved in the workspace. Perform the following detail analysis of your output :

   - Plot the x-coordinates of true pose and the estimated pose against time, along with the 3-σ bound.
   - Similarly do it for y-coordinates and theta of true pose and the estimated pose in separate figures.

- Also plot the determinant of covariance of estimated pose against time. For your information determinant of covariance matrix is the generalized variance and moreover it captures the volume of your data cloud.
- Change the noise parameters of motion model and measurement model in '**configfile.m**' file and comment on the performance and computation complexity of the filter.

Hints :

- **Prediction :** This time we use a slightly different **non-linear** motion model :

$$\mathbf{x_t} = \mathbf{f}(\mathbf{x_{t-1}}, \mathbf{u_t}) \Rightarrow \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} + V * dt * cos(G + \theta_{t-1}) \\ y_{t-1} + V * dt * sin(G + \theta_{t-1}) \\ \textbf{pi\_to\_pi}(\theta_{t-1} + V * dt * sin(G)/W_B) \end{bmatrix} \tag{1}$$

  where V, G are the noisy controls which represents the linear velocity and angular rotation respectively and W_B is wheel base

  Note : '**pi_to_pi**' function is given to you to keep the orientations (angles) in the range $[-\pi \;\; \pi]$. Make sure you use this function when representing the robot heading.
- **Update :** Here we use the non-linear measurement model as follows :

$$\mathbf{z_t} = \mathbf{h}(\mathbf{x_t}) + \eta \tag{2}$$

  where $\mathbf{h}(\mathbf{x_t}) = \begin{bmatrix} r \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{(x_t - \ell_x)^2 + (y_t - \ell_y)^2} \\ atan2(\frac{y_t - \ell_y}{x_t - \ell_x}) - \theta_t \end{bmatrix}$ and $\ell = \begin{pmatrix} \ell_x \\ \ell_y \end{pmatrix}$ is the true pose of the landmark.

Note : See the '**configfile.m**' to know about the noise parameters associated with prediction and measurement model.

8. A robot uses a range sensor that can measure ranges from $0mts$ and $3mts$. For simplicity assume the ranges are distributed uniformly in the interval. Unfortunately, the sensor can be faulty. When the sensor is faulty, it constantly outputs a range below $1mts$, regardless of the actual range in the sensor measurement cone. We know that the prior probability for a sensor to be faulty is $p = 0.01$.
   Suppose the robot queried its sensor N times, and every single time the measurement value is below $1mts$. What is the probability of a sensor fault, for $N = 1, 2, \ldots, 10$. Formulate the corresponding probabilistic model.

9. In this you are asked to design a Kalman filter for a simple dynamical system : a car with linear dynamics moving in a linear environment. Assume $\Delta t = 1$ for simplicity. The position of the car at time t is given by x_t. Its velocity is $\dot{x}_t$, and its acceleration is $\ddot{x}_t$. Suppose the acceleration is set randomly at each point i time, according to a Gaussian with zero mean and covariance $\sigma^2 = 1$.
   (a) What is a minimal state vector for the Kalman filter so that the resulting system is Markovian?
   (b) For your state vector, design the state transition probability $p(x_t|u_t, x_{t-1})$.
   (c) What will happen to the correlation between x_t and $\dot{x}_t$ as $t \uparrow \infty$?
   We will now add measurements to our Kalman filter. Suppose at time t, we can receive a noisy observation of x. In expectation, our sensor measures the true location. However, this measurement is corrupted by Gaussian noise with covariance $\sigma^2 = 10$.
   (d) Define the measurement model.

10. Show that the importance weights of the particles in the particle filter are proportional to the measurement model. What is the effect of using $M = 1, 2$ particles in the particle filtering? Can you give an example where the posterior will be biased? If so by what amount?

11. The binary filter assumes that a cell is either occupied or unoccupied, and the sensor provides noisy evidence for the correct hypothesis. In this question, you will be asked to build an alternative estimator for a grid cell: Suppose the sensor can only measure "0 = *unoccupied*" or "1 = *occupied*", and it receives a sequence $0, 0, 1, 0, 1, 1, 1, 0, 1, 0$.
    What is the maximum likelihood probability p for the next reading to be 1? Provide an incremental formula for a general maximum likelihood estimator for this probability p. Discuss the difference of this estimator to the binary Bayes filter (all for a single cell only).

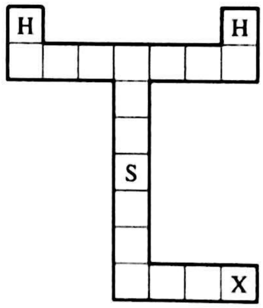

Figure 1

12. Prove that for Graph SLAM the distance between any two landmarks converges to the correct distance with probability 1.

13. Name three key, distinct advantages for each of the following SLAM algorithms : EKF, Graph SLAM and Fast SLAM. Describe a set of circumstances under which FastSLAM will fail to converge to a correct map.

14. This problem is known as tiger problem. A person faces two doors. Behind one is a tiger, behind the other a reward of $+10$. The person can either listen or open one of the doors. When opening the door with a tiger, the person will be eaten, which has an associated cost of -20. Listening costs -1. When listening, the person will hear a roaring noise that indicates the presence of the tiger, but only with 0.85 probability will the person be able to localize the noise correctly. With 0.15 probability, the noise will appear as if it came from the door hiding the reward. Now answer the following questions :
(a) Provide the formal model of the POMDP, in which you define the state, action and measurement spaces, the cost function and the associated probability functions.
(b) What is the expected cumulative payoff/cost of the open-loop action sequence : "Listen, listen, open door 1"? Explain your calculation.
(c) What is the expected cumulative payoff/cost of the open-loop action sequence : "Listen, then open the door for which we did nor hear a noise"? Again, explain your calculation.
(d)Manually perform the one-step backup operation of the POMDP.
(e) Manually perform the second backup and provide all explanations.

15. In this you have to extend dynamic programming to an environment with a single hidden state variable. The environment is a maze with a designated start marked "S", and two possible goal stated, both marked "H." What the agent does not know is which if the two goal stats provides a positive reward. One will give $+100$, whereas the other will give -100. There is a 0.5 probability that either of those situations is true. The cost of moving is -1; the agent can only move into four directions north, south, east, and west. Once a state labeled "H" has been reached, the play is over.
(a) Implement value iteration algorithm for this scenario ignoring the label "X" in figure1. What is optimal policy ?
(b) Modify your algorithm to accommodate a probabilistic motion model : with 0.9 chance the agent moves as desired; with 0.1 chance it will select any of the three other directions at random. Run your value iteration algorithm again, and compute both the value of the starting state, and the optimal policy.
(c) Now suppose the location labeled "X" contains a sign that informs the agent of the correct assignment of rewards to the two states labeled "H". How does this affect the optimal policy?
(d) How can you modify your value iteration algorithm to find the optimal policy? Be concise. State any modifications to the space over which the value function is defined.
(e) Implement your modification, and compute both the value of the starting state and the optimal policy.