Midterm Exam CS226

Stanford CS226 Statistical Algorithms in Robotics, Winter 2006

Full Name: ______________________________

Email: ______________________________

ANSWERS

Welcome to the CS223B Midterm Exam!

- The exam is 6 pages long. Make sure your exam is not missing any sheets. The exam has a maximum score of 50 points. You have 60 minutes.
- The exam is closed book, closed notes, closed cell phones, etc.
- Write your answers in the space provided. If you need extra space, use the back of the preceding sheet.
- Write clearly and be concise.
- All points will be manually counted before certification.

Question	Points
1 (20 max)	
2 (10 max)	
3 (20 max)	
total	

1 Rao-Blackwellized Filter 20pts

Suppose we have an existing (and working!) EKF implementation of feature-based localization for a mobile robot operating on a paved surface. Now suppose we operate the robot in outdoor terrain where the robot can face four very distinct surfaces: pavement (where the existing EKF works well), ice (where the wheels slip at random), water (where the robot floats), and mud (where the wheels frequently become stuck. Set up a Rao-Blackwellized EKF. *(Note: a simple yes/no is insufficient. Please explain your answers. And please keep things simple! This question is not about the dynamics of floating robots!)*

1. What is the state of the Rao-Blackwellized EKF?

   Answer: The discrete surface variable and the state of the existing EKF. It may happen that things like slip introduce additional hidden variables but we won't consider them here.

2. In the Rao-Blackwellized EKF, what elements of the state would be implemented by particle filters, what elements by Kalman filters?

   Answer: The discrete surface variable would the implemented by the particle filter, and all other state variables by the EKF.

3. What modification would one have to apply to the EKF itself?

   Answer: Extend the motion model, by introducing one for ice, one for water, and one for mud.

4. Suppose we know for a fact that the surface type remains constant throughout the robot's operation. What does this mean for the probabilistic models of the Rao-Blackwellized filter?

   Answer: The next state function for the surface type is a point mass distribution. In other words, no particle ever changes the surface type.

5. How would you compute the weight for resampling? (No equations please, just an intuitive explanation.)

   Answer: the resampling weight is the likelihood the robot associates with a measurement. Since for the four different terrain surfaces, the motion predictions will all be different, the likelihood of the measurements will be different as well. Particles with the correct surface model will survive with higher likelihood.

6. Can the filter figure out the surface type? Will it?

   Answer: Yes, by virtue of the answer to the previous question. It will eventually figure it out with probability 1 if the surface type never changes.

7. Someone proposes to replace the particle filter by a histogram filter. Is this possible? Is this a good idea? Argue why / why not.

> *Answer: This is possible: It's just like a particle filter with 4 particles (one for each terrain) which are never ever resampled. It's a good idea in the beginning, since we won't run danger that the correct terrain is discarded by accident. Later on, when we know the terrain with high certainty, it'll be a bad idea since we'll be wasting computational resources.*

8. Obviously we can re-implement the EKF using FastSLAM. What would the mean for the basic filter? What variable would be represented by particles, what variables by Gaussians?

> *Answer: The surface type and the robot pose would be represented by particles, the landmark locations by Gaussians..*

9. Should we replace the EKF by FastSLAM in this application? Argue why / why not.

> *Answer: There are two answers: No: If the EKF works well, then keep it. Yes: For large number of landmarks FastSLAM might be more efficient in its own right. Either answer is acceptable.*

10. Now suppose the terrain can changes over time, as the robot moves. Would this affect the state vector of the filter? If so, argue why and give the new state vector. If not, argue why not.

> *Answer: One might be inclined to say no, since a random chance would only affect the next state function. However, this answer is wrong. Terrain won't change randomly When the robot traverses the same terrain again, it'll be the same terrain type. To keep our world Markovian, we now need to include a terrain map in the state vector. This is feasible with algorithms like FastSLAM but would be a substantial change.*

2 Multi-Robot GraphSLAM 10pts

Suppose we run GraphSLAM for multiple robots. The robots can move along their x-, y-, and z axes, but in doing so they retain a fixed global (and known) orientation. Landmarks are points in 3-D, and the robots can measure Δx, Δy, and Δz to the landmarks. Suppose robots can see landmarks but they cannot see each other.

1. What would be an appropriate state vector? How many elements does it have? (Assume there are K robots, N landmarks, T time steps, and D total measurements.)

> *Answer: It contains all poses $s_t^i = (x_t^i, y_t^i, z_t^i)$ where i is a robot index, and all landmarks $m_k^j = (x^j, y^j, z^j)$, where j is the landmark index. With K robots, N landmarks and T time steps, we get $3(TK + N)$.*

2. What would be the appropriate motion model? Be consise.

> *Answer: it would decompose into local motion models. Assume independent Gaussian noise, we would get*

$$\begin{pmatrix} x_{t+1}^i \\ y_{t+1}^i \\ z_{t+1}^i \end{pmatrix} = \begin{pmatrix} x_t^i + \Delta_t^i(x) \\ y_t^i + \Delta_t^i(y) \\ z_t^i + \Delta_t^i(z) \end{pmatrix} + \mathcal{N}(0, \Sigma) \tag{1}$$

> *where $\Delta_t^i()$ is the actual motion, and Σ is a diagonal covariance matrix that characterizes the motion noise. The joint motion model is defined through a high-dim matrix that is block diagonal, where each block is of the type described here.*

3. For GraphSLAM, would the robots be forced to exchange raw sensor measurements when building a joint map, or could they all retain their own local statistics and communicate those? If the latter, what statistics would that be?

   Answer: They would not. It would suffice to compute the local information matrices and vectors and communicate those.

4. Now suppose that robots can see each other. When one robot sees another, what would happen to the joint information matrix?

   Answer: That would add an element in the joint off-diagonal between different robot paths.

5. Could the robots still compute a local map after seeing each other? Argue why / why not.

   Answer: They can, but they have to ignore this information.

3 True or False? 20pts

Correct answer is $+1$ point per question; a false answer results in -1 point.

FALSE If two variables A and B are independent, then they remain independent given knowledge of any other variable C.
Answer: False, this is discussed in the very beginning of the book.

FALSE If the posterior belief is unimodal, EKFs are the method of choice from the varius filters in the book.
Answer: False. No for a number of reasons. Usually UKFs are superior, and when dealing with heavy tail distributions neither of them might work sufficiently well.

TRUE A particle filter that observes the position of a moving object can infer its velocity
Answer: True - just make velocity part of the state vector as we discussed in class

TRUE FastSLAM provides a sound solution to the data association problem in that each particle pursues its own data association
Answer: Obviously true

FALSE The covariance matrix of EKF SLAM is rank deficient
Answer: False, no clue why this should be true, this makes no sense

FALSE The Kalman gain K measures the amount of surprise of a specific sensor measurement
Answer: False - the sensor measurement isn't used in its calculation

TRUE In certain degenerate cases a particle filter could still work even with a single particle.
Answer: True - the degenerate case would be the one of fully deterministic motion

TRUE A histogram filter approximates the posterior belief over a continuous space by a distribution over a finite decomposition of this space
Answer: True, this is its definition

TRUE In occupancy grid maps, adding a large number has a stronger effect on the posterior probability than adding a small number
Answer: True, we added more evidence.

TRUE The log-odds transform is invertible.
Answer: True, we can go from log-odds to probability and back

TRUE Occupancy grid maps in log-odds form are numerically more stable than in probability form
Answer: True, log odds go from minus to plus infinity, probabilities go from 0 to 1 (with 1 being the unstable value)

TRUE Given a GraphSLAM information matrix and vector, computing the map requires linear time so as long as the size of the largest loop is independent of N, the number of features in the map.
Answer: True, the size of the loop will then be a constant factor

FALSE Apply EKF to global localization, and assume the robot observes the range and bearing to four different landmarks in its very first sensor measurement (assuming known correspondence, and a map is available). EKF will then very likely succeed in globally localizing the robot.
Answer: False, the linearization may fail, since it relies on an assumed orientation (which may be badly wrong).

TRUE The information matrix is symmetric.
Answer: True, it's a quadratic matrix

FALSE In EKF SLAM, in the limit the robot can determine the exact locations of all landmarks with arbitrary accuracy
Answer: False, there remains uncertainty about the absolute location, it's only the relative locations that will be known with full certainty

TRUE Bayes filters assume conditional independence of sensor measurements taken at different points in time given the current and all past states.
Answer: True

FALSE Bayes filters assume conditional independence of sensor measurements taken at different points in time given the current state (but not past states).
Answer: False, since past measurements are still dependent if past state is known.

FALSE The importance weight of a particle is the same (modulo some random noise) as the determinant of the covariance in a Kalman filter
Answer: False, this is plain nonsense.

TRUE The update time of FastSLAM with known correspondence is in $O(N \log N)$.
Answer: True, assuming the tree implementation

FALSE When implementing FastSLAM on a physical robot, it is always good to use at least 10,000 particles just to be sure
Answer: False, the more particles we use, the less often we can update the filter. Sometimes we need a small number of particles to keep track with the incoming data.