EL2320 - Applied Estimation
GraphSLAM

Student Name: Riccardo Periotto
Student ID: 19980915-T633

Part I - The SLAM problem

Question 1

GPS measurements typically include the following information:

- latitude, the receiver's north-south position, measured in degrees;
- longitude, the receiver's east-west position, measured in degrees;
- altitude, the receiver's height above sea level, measured in meters;
- time: the time at which the measurements were taken, typically given in Coordinated Universal Time (UTC).

We can assume to have a function that converts a GPS measurement into the x, y coordinates of a local reference frame, for example using the UTM (Universal Transverse Mercator) coordinate system. In this way, the requested function has the following shape:

$$\mathbf{z}_{GPS} = h_{GPS}(\mathbf{x}) + \eta_{GPS} = [x, y]^\top + \eta_{GPS}$$

As can be seen, the formula excludes the time information and does not provide indications about the robot's orientation, but just regarding its position.

Question 2

The compass measures the absolute heading of the robot with respect to the magnetic North. We can assume to have a function which is able to convert the compass measurement to the orientation of the robot θ with respect to the adopted reference frame. In this case, the measurement model becomes:

$$\mathbf{z}_{compass} = h_{compass}(\mathbf{x}) + \eta_{compass} = \theta + \eta_{compass}$$

Question 3

The measurement model can be:

$$z_{gyro} = h_{gyro}(\mathbf{x}) + \eta_{gyro} = \omega + \eta_{gyro}$$

The formula holds assuming the state vector $\mathbf{x}$ includes the angular velocities about axis z, ω.

Modifying the system state in this way, it is possible to calculate the turned angle $\Delta\theta$ assuming a constant velocity in each timestep, getting $\Delta\theta = \Delta t * \omega$. In addition, it is also possible to fuse the information from the gyro and from the wheel to make a better prediction of the angle turned.

Question 4

Assuming again that the control input $\mathbf{u}$ gives the change in the distance travelled Δs and the angle turned $\Delta\theta$, we have:

$$\mathbf{g} = (x_{i-1} + (\Delta s)cos(\theta_{i-1}), y_{i-1} + (\Delta s)sin(\theta_{i-1}), \theta_{i-1} + \Delta\theta)^\top$$

$$p(\mathbf{x_i}|\mathbf{x_{i-1}}, \mathbf{u_i}) = \frac{1}{\sqrt{|2\pi R|}} exp - \frac{1}{2}(\mathbf{x_i} - \mathbf{g})^\top R^{-1}(\mathbf{x_i} - \mathbf{g})$$

Question 5

With $\sigma \sim N(0, Q)$ and $Q = 1$, the explicit measurement model becomes:

$$\bar{z_k} = h(\mathbf{x_i}, \mathbf{m_j}) = \sqrt{(m^x_{f(k)} - x^x_{p(k)})^2 + (m^y_{f(k)} - x^y_{p(k)})^2}$$

$$p(z_k | x_{p(k)}, m_{f(k)}) = \frac{1}{\sqrt{2\pi\sigma^2}} exp^{-\frac{1}{2}(\frac{z_k - \bar{z_k}}{\sigma})^2}$$

Question 6

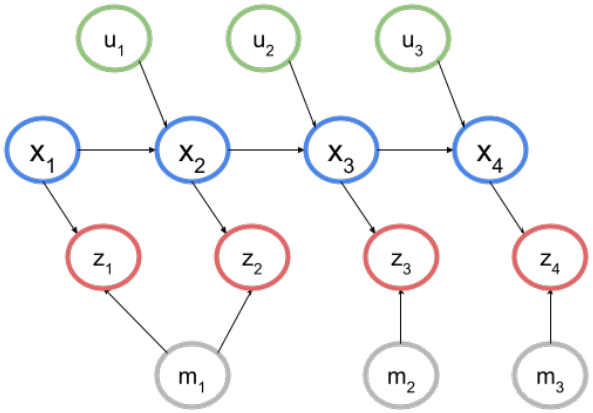

Figure 1: SLAM problem as Bayesian belief network.

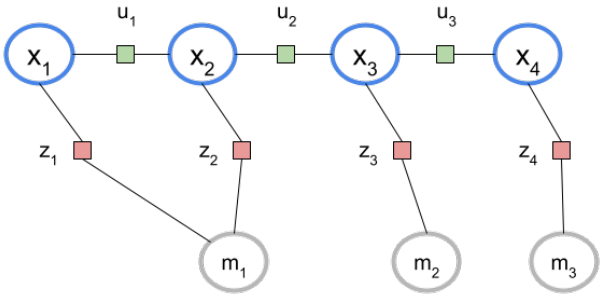

Figure 2: SLAM problem as a factor graph.

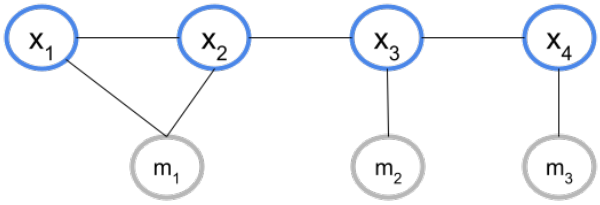

Figure 3: SLAM problem as Markov random field.

Question 7

The assumptions behind the equation for the probability distribution are:

- the prior distribution on the initial state is independent of the other distribution involved;
- the motion model probability for every state is independent of the one in other states; given the previous state and the received input.
- each measurement is independent of the others once the state and the landmark location are known;
- the association problem has been already solved.

Question 8

When a loop closure is detected, the structure of Ω remains the same, but the content of the entries involved in the closure change value.

As explained in the book, the possibility to solve the inference problem by computing the calculation $\Omega^{-1}\xi$ in linear time is only present when the matrix can be made band-diagonal. With loop closures, both before and after applying the reduction step, the matrix is filled with values out of the diagonal and the algorithm performing the inversion on such matrices has higher complexity.

Question 9

The pose to pose factor is represented by the distribution $p(\mathbf{x_i}|\mathbf{x_{i-1}}, \mathbf{u_i})$ reported also above. Computing the logarithm of that function, we obtain:

$$ln(p(\mathbf{x_i}|\mathbf{x_{i-1}}, \mathbf{u_i})) = const + \frac{1}{2}(\mathbf{x_i} - g(\mathbf{u_i}, \mathbf{x_{i-1}}))^\top R^{-1}(\mathbf{x_i} - g(\mathbf{u_i}, \mathbf{x_{i-1}}))$$

Linearizing g as in the book, it is possible to write it as:

$$g(\mathbf{u_i}, \mathbf{x_{i-1}})) \approx g(\mathbf{u_i}, \bar{\mu}_{\mathbf{i-1}})) + G(\mathbf{x_{i-1}} - \bar{\mu}_{\mathbf{i-1}})$$

where G is the Jacobian of the motion model with respect to the state vector, while $\bar{\mu}_t$ is the estimate of the state for $\mathbf{x_{t-1}}$.

Following again the book by substituting the linearization in the previous equation and collecting the terms related to the pose terms at time t and time $t-1$, the formula becomes:

$$const + \frac{1}{2}\mathbf{x}_{\mathbf{i-1}}^\top G^\top R^{-1} G\mathbf{x_{i-1}} + \frac{1}{2}\mathbf{x}_\mathbf{i}^\top R^{-1}\mathbf{x_i} - \mathbf{x}_\mathbf{i}^\top R^{-1} G\mathbf{x_{i-1}}$$

$$-\mathbf{x}_{\mathbf{i-1}}^\top G^\top R^{-1}(G\bar{\mu}_{i-1} - g(u_i, \bar{\mu}_{i-1})) + \mathbf{x}_\mathbf{i}^\top R^{-1}(G\bar{\mu}_{i-1} - g(u_i, \bar{\mu}_{i-1}))$$

Given that, it is possible to extract the provided and requested values for Ω and ξ. In particular, considering also an expansion of the equation 8 provided, we have:

$$\Delta\Omega_{i-1,i-1} = G^\top R^{-1} G$$

$$\Delta\Omega_{i,i} = R^{-1}$$

$$\Delta\Omega_{i,i-1} = R^{-1} G$$

$$\Delta\xi_{i-1} = G^\top R^{-1}(G\bar{\mu}_{i-1} - g(u_i, \bar{\mu}_{i-1}))$$

$$\Delta\xi_i = R^{-1}(G\bar{\mu}_{i-1} - g(u_i, \bar{\mu}_{i-1}))$$

Question 10

A possibility is to check whether the difference between the solution obtained in the previous timestep and the new one is below a certain convergence threshold. If this is the case, we can assume that the solution has converged into something close to the optimal state. On the other side, if the difference is large, it means that there is still the possibility that running the algorithm again, starting from the new position, will further improve the solution.

*nQuestion 11 The factors that might influence the number of iterations required by the algorithm are:

- the initial estimate of the robot's pose;
- the accuracy of the motion and measurement models;
- the number of landmarks (the more landmarks, the more the data to process);

Question 12

When might this implementation of a GraphSLAM solution fail to con- verge? 10 The implementation could fail in the following cases:

- the initial estimate is poor;
- the motion and measurement model is not accurate;
- insufficient collected data;
- incorrect constraint (e.g. wrong data association);
- complex non-linearity;
- motion and measurements noises.

Part II - Practical

Question 1

To the last two questions, it is possible to answer in general.

- The linearise system is a reasonable approximation when the robot's state estimate is close to the true state and the system can be considered "locally linear" around that point. If these conditions are not satisfied, or if the motion model is not accurate, then the linearisation will not be a good approximation.
- When the robot closes a loop, it adds new constraints to the factor graph by connecting the current robot pose and the previous pose that observed the same landmark. This is reflected by a new entry in the information matrix used by the algorithm. The new constraint reduces the uncertainty in the state and map estimation, leading to a more accurate solution.

Dataset 1

- The map is small with a small number of landmarks.

- There are no signs of local minima or weak minima, indeed the execution of the algorithm always converges to the ground truth.
- The density of the map and the sensor range does not affect the convergence.
- The noise does not affect the convergence of the algorithm significantly, which indeed converges in 5 iterations with the given parameters. Also when increasing the noise out of proportion (i.e. in the order of meters and tens of degrees), the algorithm will still be able to converge in less than 10 iterations.
- With the provided data, the algorithm is also safe with respect to the initial position. For example, with an initial position completely distant from the origin (i.e. 10,10), the algorithm provides really bad results in the first few iterations, but it is able to converge again in less than 10 iterations.

Dataset 2

- The more complexity with respect to the previous dataset is given by the larger denser map, which contains more than 4 times the landmarks than the first dataset.
- Independently from the initial condition, the algorithm converges. There are no signs of local minima.
- With a denser map (with respect to Dataset 1), the overall uncertainty and errors are lower.
- The noise does not affect the result, even setting values far larger than those provided. The number of iterations to converge to a low error however is lower than before.
- As in the previous case, the initial condition simply makes the estimation to be bad in the first iterations, but the algorithm converges soon anyway.

Dataset 3

- The map is not dense andresemble that of Dataset 1. However, the path of the robot is more complex and the same landmarks are seen multiple times.
- The noises strongly affect the result and indeed Question 3 require to tune them appropriately to make the algorithm to converge to a suitable solution.

Question 2

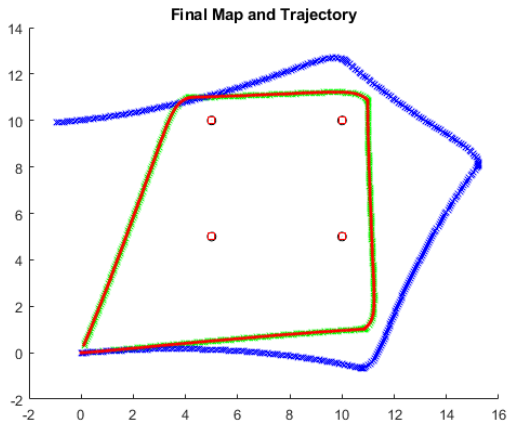

Figure 4: Best map estimation for Dataset 1.

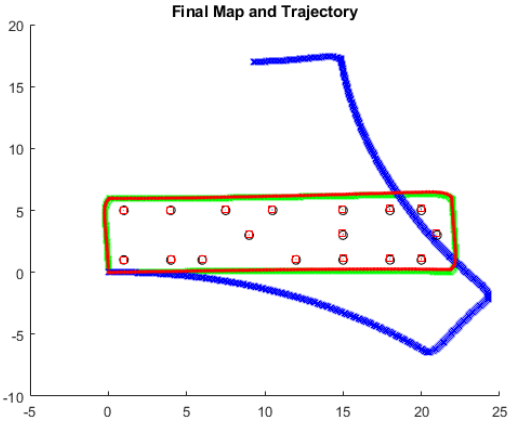

Figure 5: Best map estimation for Dataset 2.

Question 3

Outliers and spurious measurements may negatively impact the performance of the algorithm. A way to remove and clean them is through data preprocessing techniques, such as outliers detection and filtering.

Oscillation and overshooting can occur in GraphSLAM when the algorithm is trying to optimize the graph and converge to a solution, but it can't find the right balance between making progress and avoiding errors. The solutions to avoid these problem in GraphSLAM usually consist in adapting the step size along with the optimization.

GraphSLAM can be developed in a way that automatically deal with unknown data association or feature indices. This is done through the Maximum Likelihood Estimation approach, which is able to label the unknown aspect in the most likely way given the available data. In is also possible to develop the algorithm in such a way that it changes the association made before considering the update performed in the estimation.

Question 4

After the first run, I wanted the algorithm to take in consideration more the odometry, as the shape of the path performed by the robot was at least similar to it. To do that, I reduced the uncertainty of the motion model noise matrix. The best result I could achieve is the following. The algorithm converges to a value below the fixed threshold at iteration 22, but I had to increment the iteration limit to 30 to let this happening (default was 20).

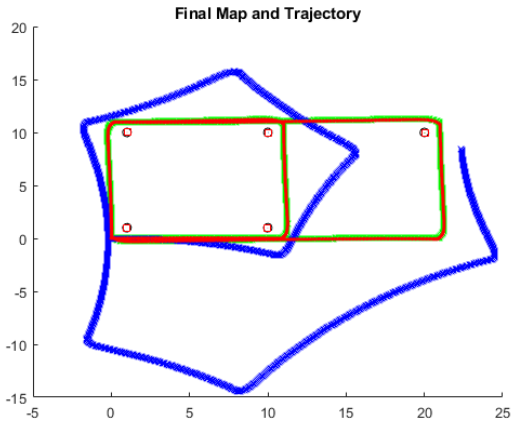

Figure 6: Best map estimation for Dataset 3.

The behaviour of the error clearly shows how the algorithm is able to find a minimum after a significant increment of the error in the first few iterations (see the following figure).

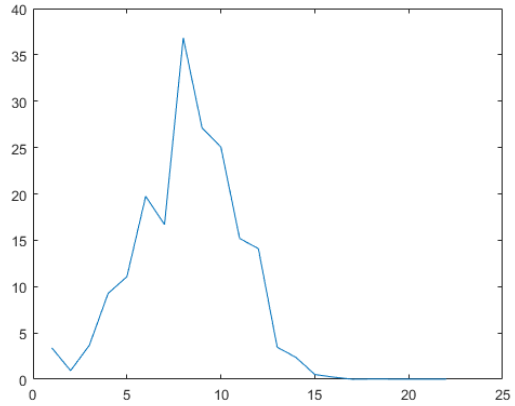

Figure 7: Algorithm error trend for Dataset 3.

The resulting mean errors and mean absolute error (MAE) are respectively [-0.015820, 0.033661, 0.003713] and [0.021150, 0.036568, 0.004497].