Homework 3

Grid Localization

1. I wrote the function `gridLocalizationStationary` which returns three items: the final probability density function of the robot belief, the x coordinates of the grid locations, and the y coordinates of the grid locations. This function starts by creating a matrix of the probability with an initial probability of $\frac{1}{n \times m}$. For each measurement input, the program loops through every grid location to check how likely the reading from `rangePredict` is the grid in which the robot sits. Following this, the program normalizes the probability and continues until it traverses all sensor readings.

2. The initial distribution is $\frac{1}{n \times m}$ which is $\frac{1}{100}$ or 0.01.

3. See Figure 1 and 2.

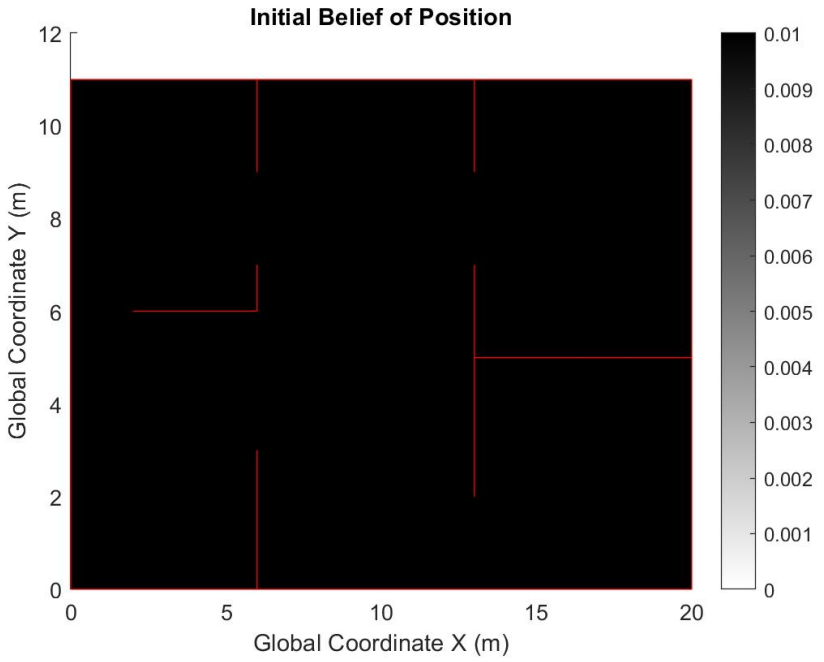

Figure 1: Initial Distribution of the Grid with Map Overlay

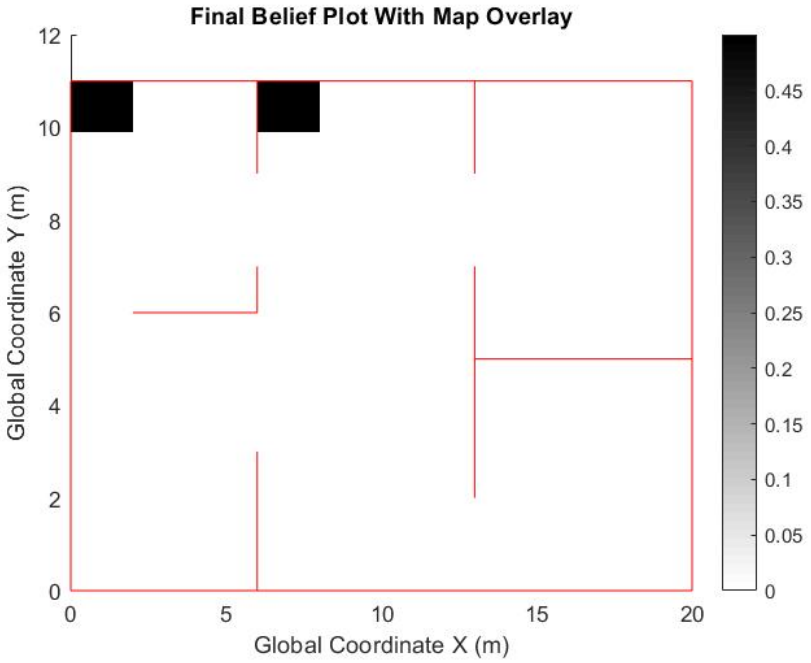

Figure 2: Final Location Belief of the Robot Position

4. Both of the locations from the 10x10 grid are included in the 40x22, but there are additional places with a slight variance in belief. From the measurements, we know that we are 1 meter away from the wall in the North and West directions and more than 3 meters away in the East and South directions. With that knowledge, it is clear that the four major locations are in the top left corners of the map in their relative corner. With the 10x10 map, we see there is an issue with the grid size being too large to allow the rightmost locations to be predicted. There is no center point that is close enough to 1 meter away from the West wall and also greater than 3 meters away from the East wall. This test shows that the grid size matters if it is too large, but additionally we notice that the 40x22 map takes much longer to run than the 10x10. I also ran a 80x44 grid as shown in Figure 5. We notice that there is almost no difference in the grid location (with a slightly more precise grid), which indicates that 40x22 is very likely large enough for finding the approximate location of the robot!

Homework 3

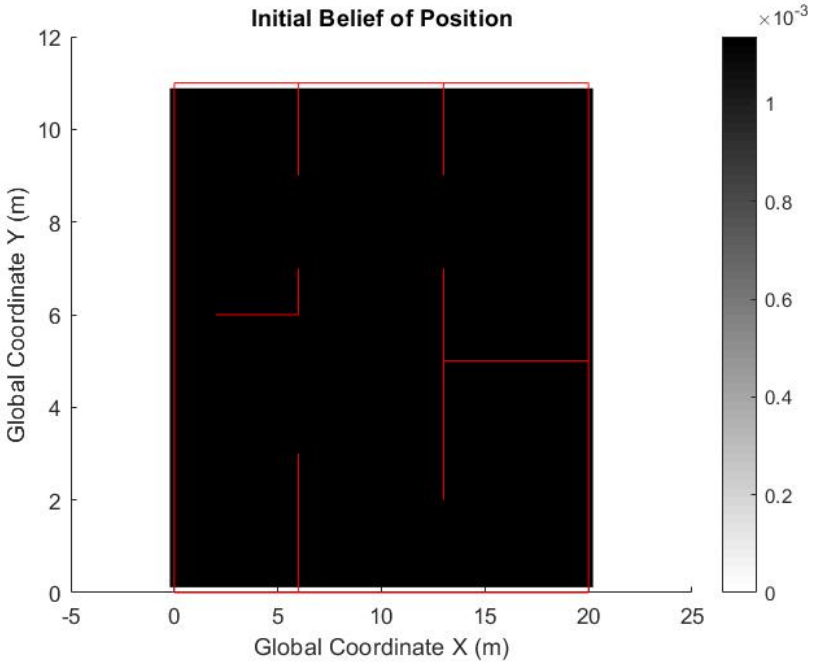

Figure 3: Initial Location Belief of the Robot Position

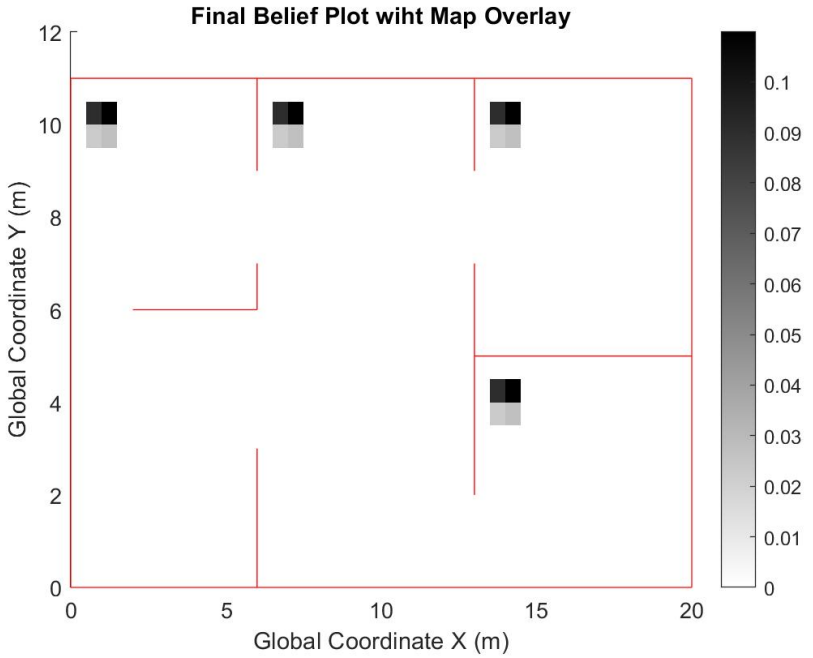

Figure 4: Final Location Belief of the Robot Position

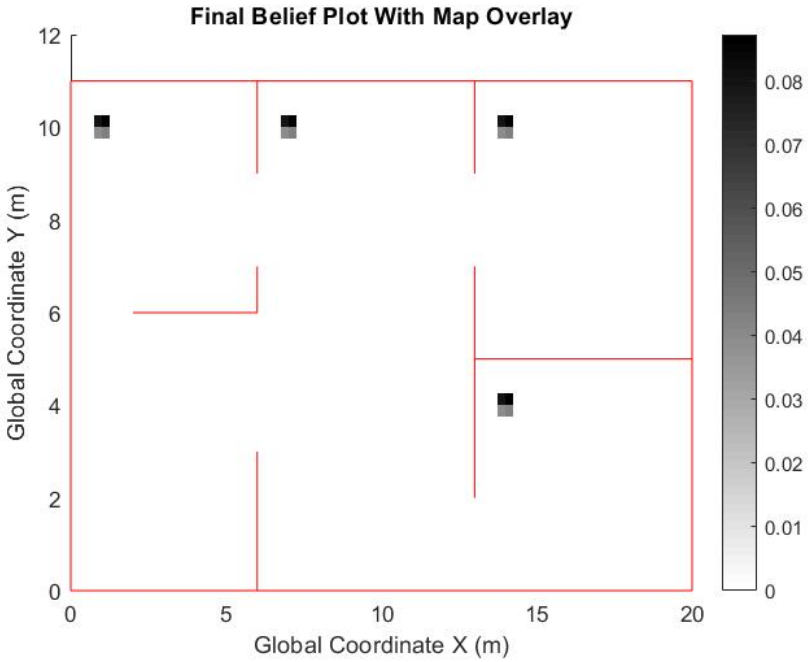

Figure 5: Final Location Belief of the Robot Position

5. Completed in Matlab.

Kalman Filter

1. Compmleted in Matlab.
2. My prediction step follows as:

$$\bar{u_t} = A\mu_t + B\mu_t + \epsilon \tag{1}$$

$$\bar{\Sigma} = A\Sigma_{t-1}A^T + R \tag{2}$$

but since B, ϵ, and R are the 0's matrix and A is the identity matrix, there is no real use for the update step with a stationary object.

3. The update is performed as followed:

$$\mu_t = \bar{\mu}K_t(Z_t - C\bar{\mu_t}) \tag{3}$$

where $Z_t - C\bar{\mu_t}$ is the difference in our sensor reading and map range measurement and K_t is

$$K_t = \bar{\Sigma_t}C^T(C\bar{\Sigma_t}C^T + Q)^{-1} \tag{4}$$

and

$$\Sigma_t = (I - K_t C)\bar{\Sigma}_t \tag{5}$$

4. My initial distribution was the Identity matrix times 50 to ensure the program knew there was 0 certainty in the position. In Figure 10, we see the covariance is only 1 with location of [1,1] initial guess, and it is a quite large range.

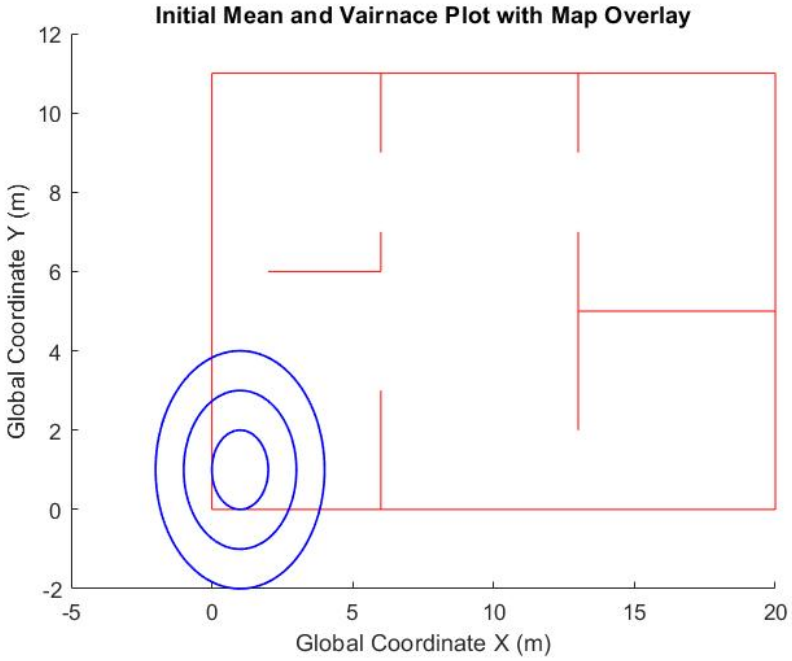

Figure 6: Initial Distribution with Covariance of 1

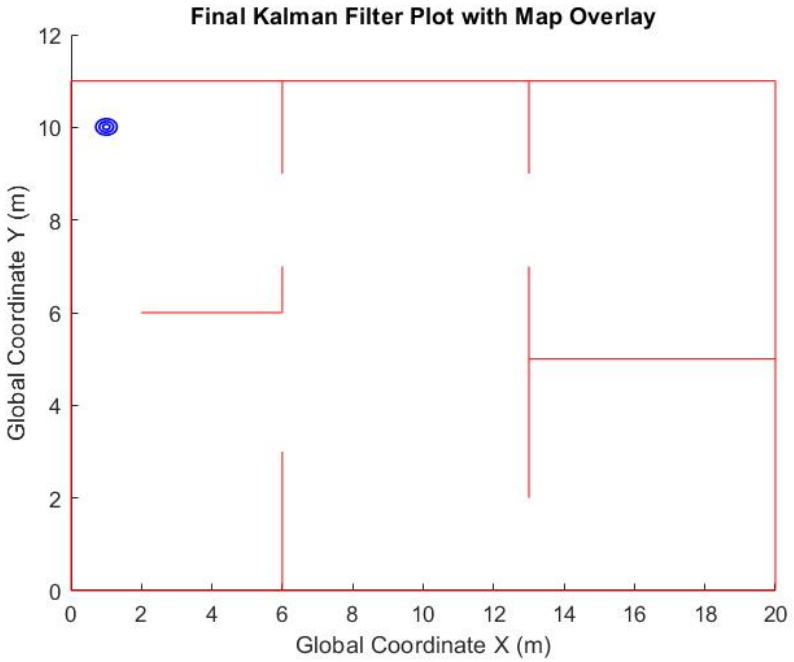

Figure 7: Final Distribution

5. See Figure 10.

6. Here I tried two different locations close to the right side of the map, and both ended up in the same location with no location or covariance change

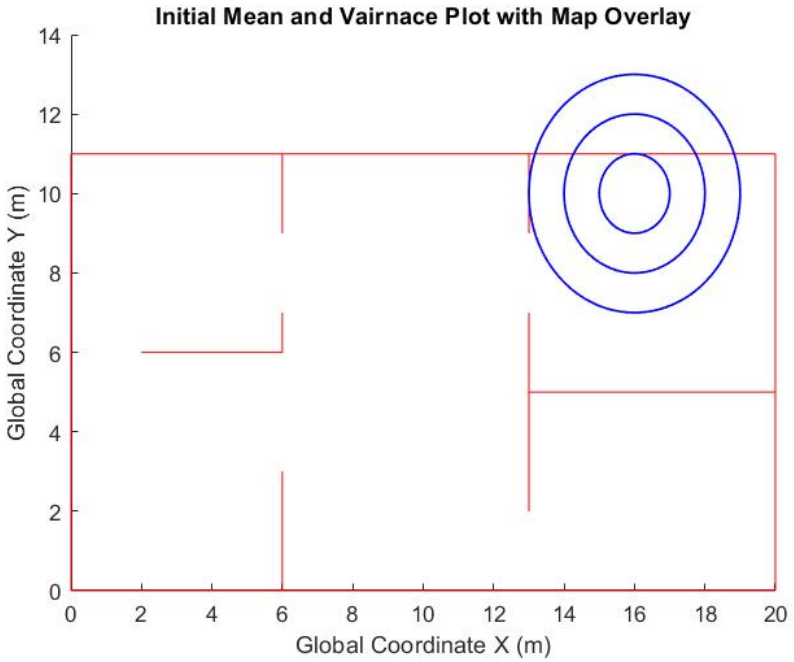

Figure 8: Initial Distribution with Position 2

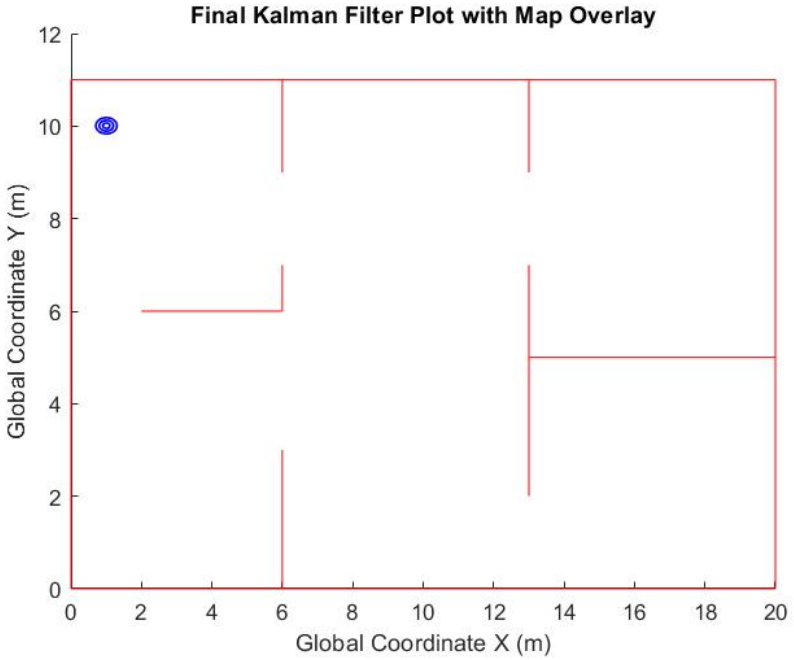

Figure 9: Final Distribution

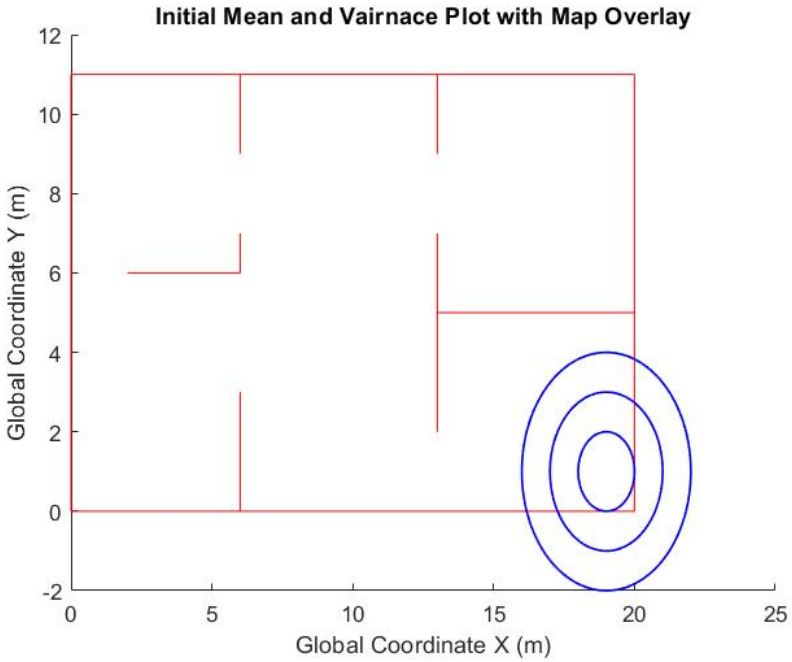

Figure 10: Initial Distribution with Position 3

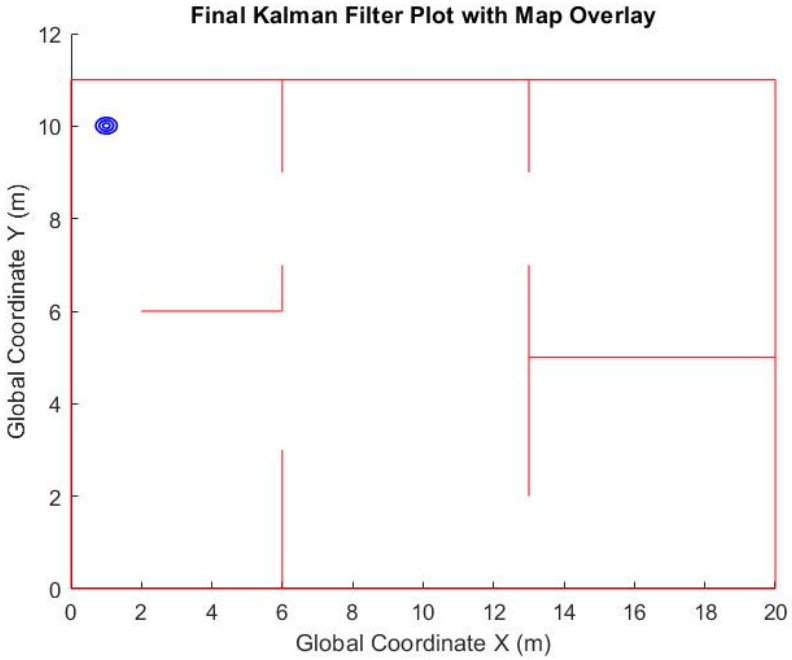

Figure 11: Final Distribution

7. Completed in Matlab.

8. As this is stationary, the Kalman Filter makes more sense than the Extended Kalman Filter. The point of the EKF is to linearize non-linear data, but in this case there is no non-linear data to process.

Location Estimate

1. If there was enough information and the map had unique features to differentiate from, both filters would be able to estimate the location of the map. The pitfall of these stationary methods is that it is possible there are two areas or rooms that have the same NESW measurements. In this scenario there is now way to differentiate the two rooms. Although this is true, we would still be given an equal probability for the grid localization and the Kalman filter will pick the one closest to the initial position typically.

2. Yes you could because there are four possible locations, and only one of them has a y coordinate lower than 8. That location is most likely in the grid of [X,y] = [4.25,14.25]. Additionally there are three also possible positions to the South, Southwest, and South directions, but their likelihood is lower than at the first position.