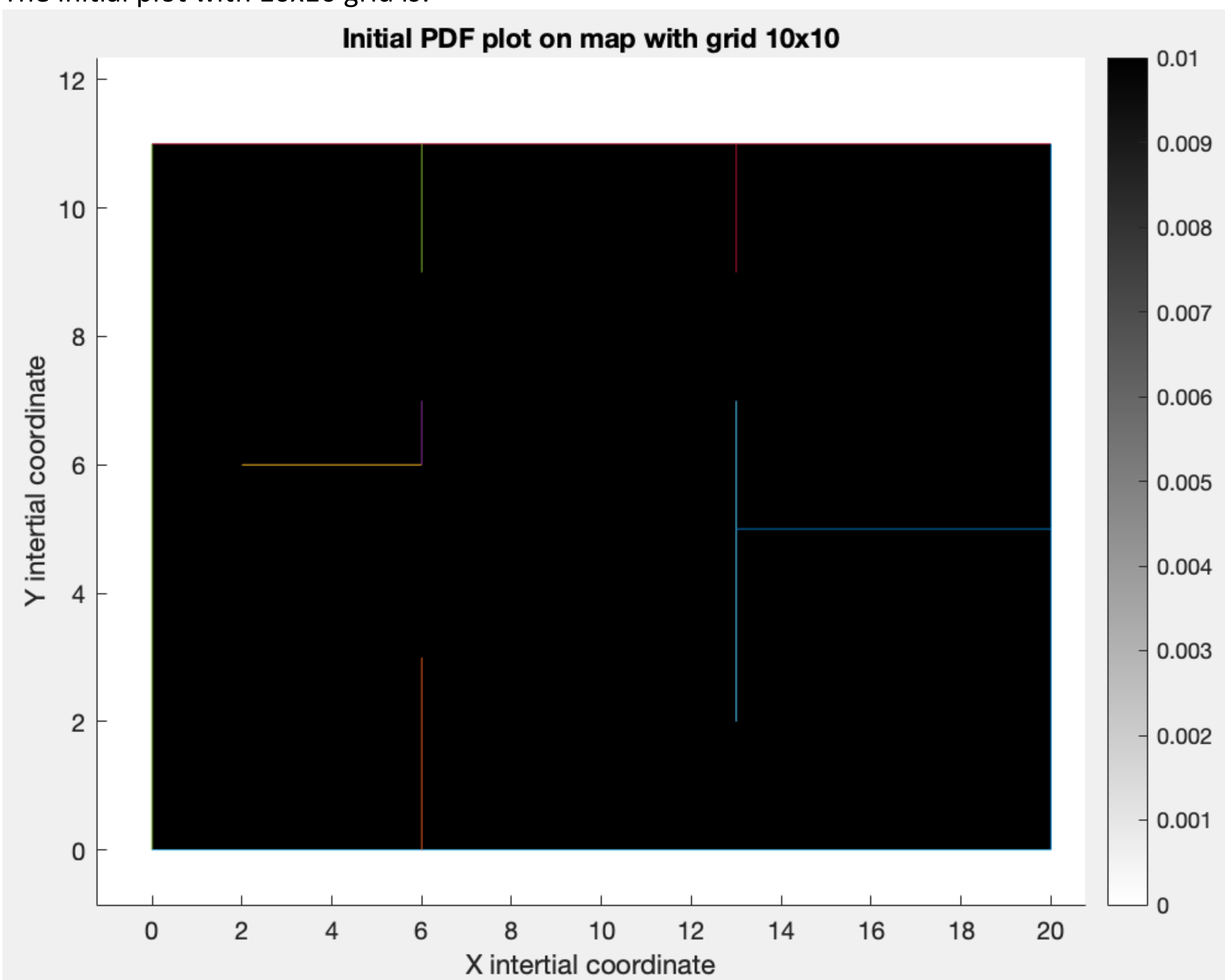
Jonathan Nusantara (jan265)
AMR
HW3

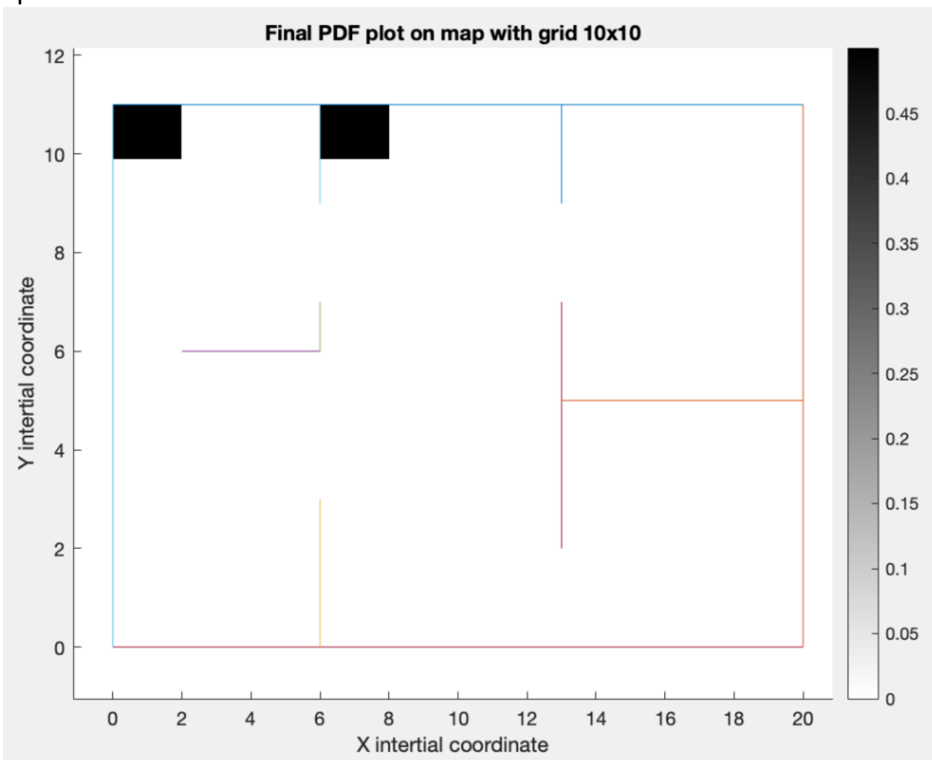
Grid Localization

1. The issue is that since we are using grids on our map, the problem is on finding out where in the grid is our robot. The location of the robot will affect the calculation of the distance of the robot to the walls. To resolve this issue, I made an assumption that the robot is always in the middle of the grid. So, when calculating the distance of the robot to a wall, I calculated the distance of the current cell or grid center to the wall.
2. The initial distribution is a uniform distribution across the 100 cells. So, the confidence of probability at each location is 0.01.
3. The initial plot with 10x10 grid is:



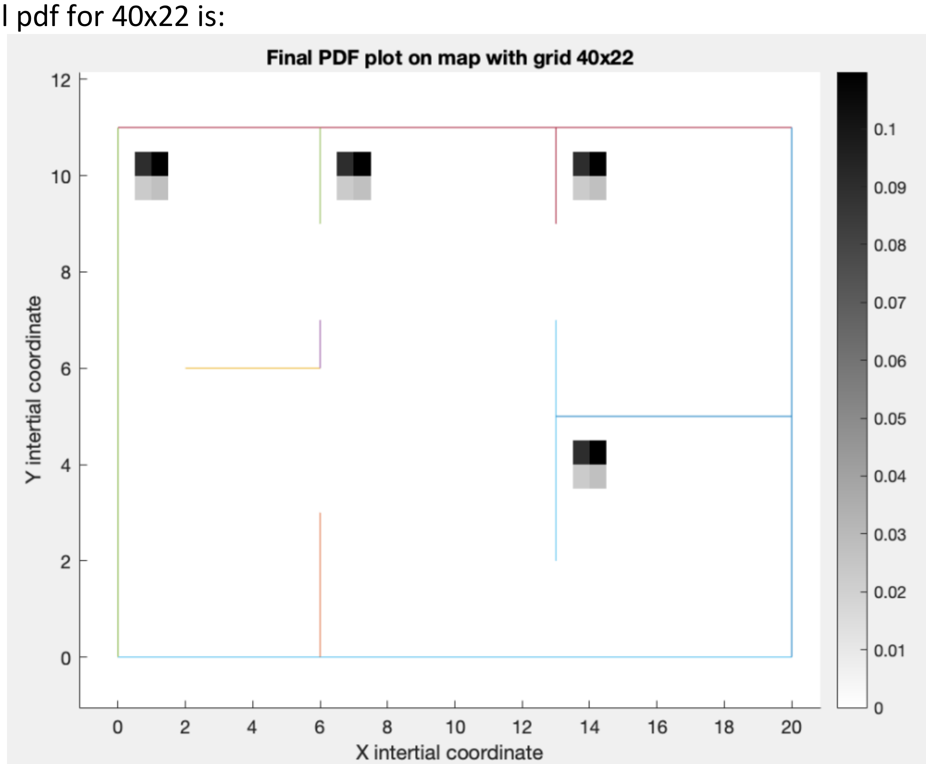
We can see how there is an equal distribution throughout the entire map.

The final pdf is:



We can see that the probability or guess comes down to two grid cell locations.

4. The final pdf for 40x22 is:



We can see the difference in final pdf of using 10x10 grid vs 40x22 grid. In a 10x10, there are only 2 possible locations. In 40x22, there are 16 possible cells across 4 regions. 2 of the regions are covered when using 10x10, while the two others are new. This happen because 40x22 allows a smaller grid, which there are more coordinates of the center of the grid. As a result, there are more coordinate that matches better with the sensor measurements. As we change the number of grids, we may always be able to get different answers, as the center coordinates in the grid will change.

Kalman Filter

1. Code is submitted.
2. No, we do not need one because the robot is stationary. Prediction step is generally on calculating how our belief changes based on previous location and controls. If there are no controls then prediction is not needed. The formula for predictions step is as below:

$$\underset{\mathbb{R}^{2\times1}}{\bar{\mu}_t} = \underset{\mathbb{R}^{2\times2}}{A} \cdot \underset{\mathbb{R}^{2\times1}}{\mu_{t-1}} + \underset{\mathbb{R}^{2\times2}}{B} \cdot \underset{\mathbb{R}^{2\times1}}{u_t} \rightarrow = 0$$

$$\begin{bmatrix} \bar{\mu}_{t\_x} \\ \bar{\mu}_{t\_y} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \mu_{t-1\_x} \\ \mu_{t-1\_y} \end{bmatrix} + \underbrace{0 \cdot 0}_{\text{0 because robot is stationary}}$$

$$\overset{\mathbb{R}^{2\times2}}{\bar{\Sigma}_t} = \overset{\mathbb{R}^{2\times2}}{A} \cdot \overset{\mathbb{R}^{2\times2}}{\Sigma_{t-1}} \cdot \overset{\mathbb{R}^{2\times2}}{A^T} + \overset{\mathbb{R}^{2\times2}}{R} \rightarrow = 0$$

process noise = 0

because robot is stationary

The prediction step calculates the mu_t_bar and sigma_t_bar. The dimensions are listed as above. The B and R matrix is 0, because the robot is stationary. So, the mu_t_bar = mu_t-1 and sigma_t_bar = sigma_t-1, and this step is not needed.

3. The update is performed as below:

Calculate kalman gain

NESW

$$\overset{\mathbb{R}^{2\times4}}{K_t} = \overset{\mathbb{R}^{2\times2}}{\bar{\Sigma}_t} \cdot \overset{\mathbb{R}^{2\times4}}{C^T} \left(\overset{\mathbb{R}^{4\times2}}{C} \cdot \overset{\mathbb{R}^{2\times2}}{\bar{\Sigma}_t} \cdot \overset{\mathbb{R}^{2\times4}}{C^T} + \overset{\mathbb{R}^{4\times4}}{Q} \right)^{-1}$$

$$C = \begin{bmatrix} 0 & -1 \\ -1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \rightarrow \text{effects of each direction on } x \text{ and } y$$

sensor noise

$$Q = \begin{bmatrix} 0.1 & 0 & 0 & 0 \\ 0 & 0.3 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.3 \end{bmatrix}$$

$$\overset{\mathbb{R}^{2\times1}}{\mu_t} = \overset{\mathbb{R}^{2\times1}}{\bar{\mu}_t} + \overset{\mathbb{R}^{2\times4}}{K_t} \Big(\underset{\text{sensor measurement}}{\overset{\mathbb{R}^{4\times1}}{z_t}} - \underbrace{\overset{\mathbb{R}^{4\times2}}{C} \cdot \overset{\mathbb{R}^{2\times1}}{\bar{\mu}_t}}_{\text{replace with calculated measurement from map \& pose}} \Big)$$

$$\overset{\mathbb{R}^{2\times2}}{\Sigma_t} = \Big(\overset{\mathbb{R}^{2\times2}}{I} - \overset{\mathbb{R}^{2\times4}}{K_t} \cdot \overset{\mathbb{R}^{4\times2}}{C} \Big) \cdot \overset{\mathbb{R}^{2\times2}}{\bar{\Sigma}_t}$$

The above shows the calculation of kalma gain, mu_t, and sigma_t. Note that the mu_t_bar = mu_t and sigma_t_bar = sigma_t. The update is performed by first calculating the kalman gain. It is calculated based on the noise of the sensor measurement Q, and the matrix C, which is the matrix that shows the relationship between measurements and robot pose. Then, we calculate mu_t using the formula stated above. However, we actually replace the C*mu_t_bar with the expected measurement. This expected measurement is our calculation on the distance between the robot location (in the middle of the cell) to the wall that it is facing, depending on the direction. The sigma_t is then calculated with the formula above.

4. For the initial distribution, the mu that is chosen is [10, 5.5]. This x and y value is chosen because it is in the middle of the map. The middle of the map is a good guess given the map.

   Next, we would have to determine sigma. In our application, sigma is the variance. Since the initial guess is a wild guess, we would want a variance that covers the entire map. The x should have an uncertainty of 10, (half of the width), and y should have the uncertainty of 5.5 (half of height), so that it extends from the map's end to end. We know that in gaussian, +- 3 standard deviation will provide 99.7% of the probabilities, almost a 100%. So, we set this uncertainty to be our 3rd standard deviation. Then, we can then calculate the variance based on this:

$X \to \pm 10 \to \frac{1}{2}$ of width

$3\ \text{std} = 10$

$\text{std} = \frac{10}{3}$

$(\text{std})^2 = \text{Var} = \left(\frac{10}{3}\right)^2$

So, $\sigma = \begin{bmatrix} \left(\frac{10}{3}\right)^2 & 0 \\ 0 & \left(\frac{5.5}{3}\right)^2 \end{bmatrix}$

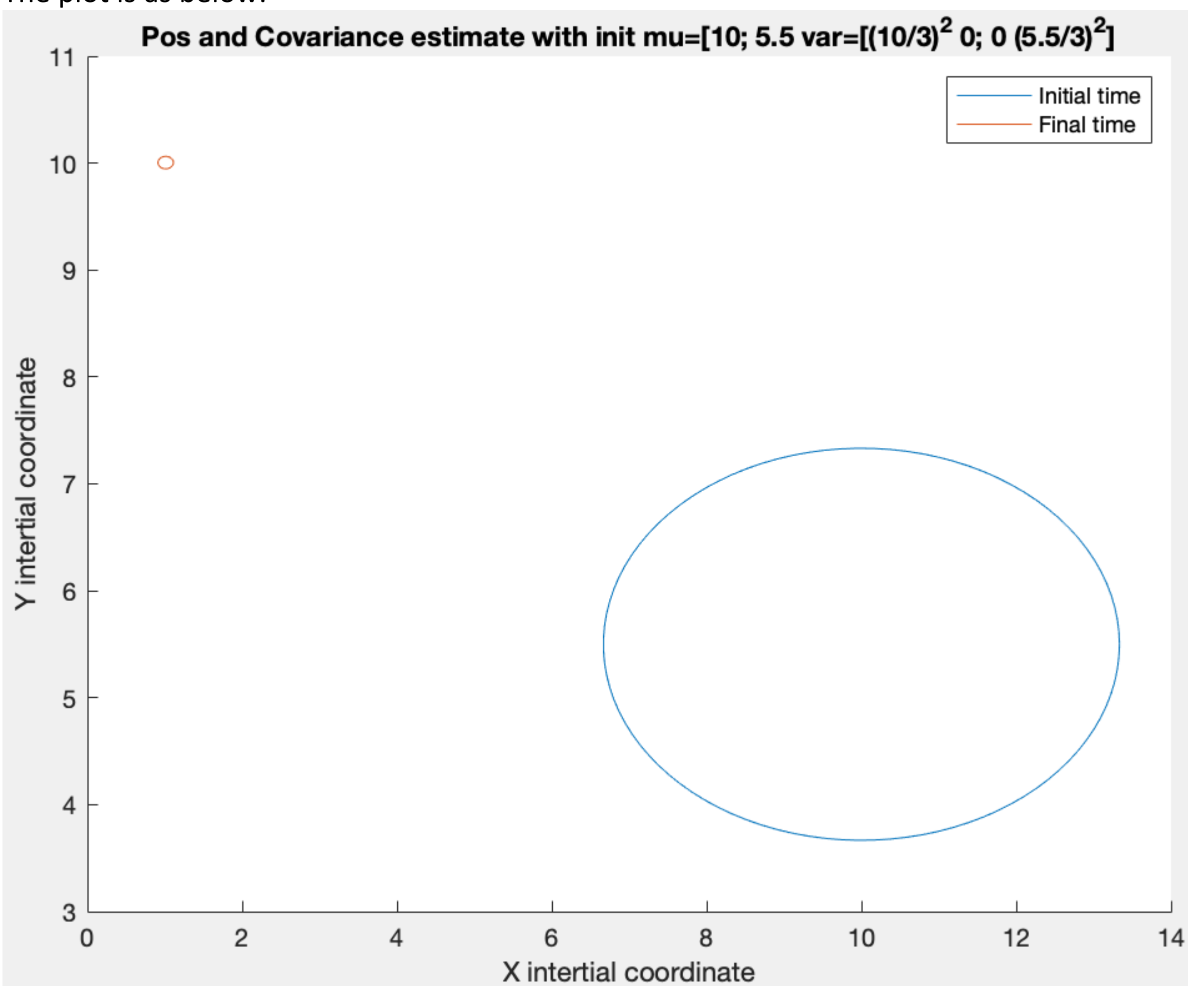
$Y \to \pm 5.5 \to \frac{1}{2}$ of height

$3\ \text{std} = 5.5$

$\text{std} = \frac{5.5}{3}$

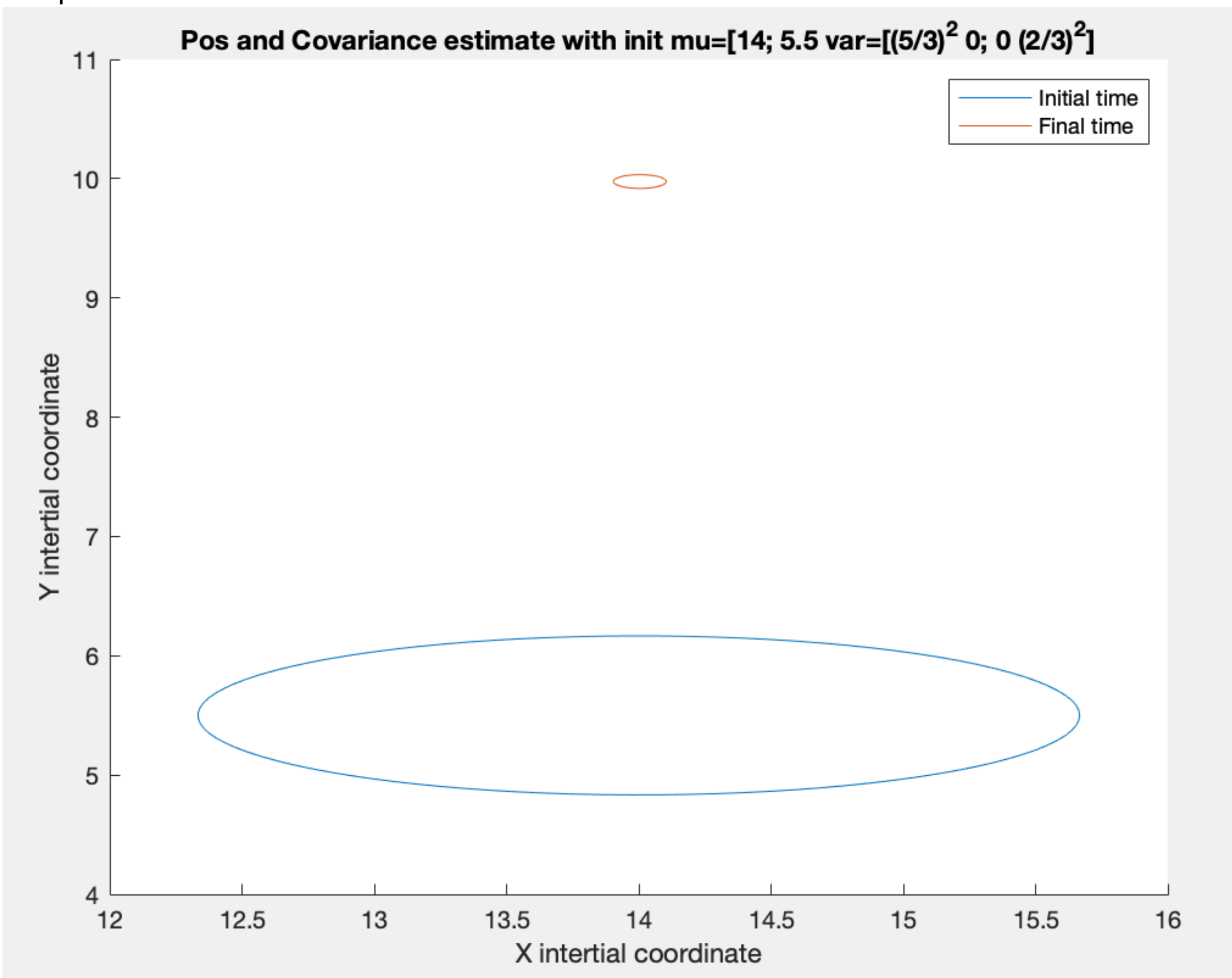
$(\text{std})^2 = \text{Var} = \left(\frac{5.5}{3}\right)^2$

5. The plot is as below:



From the plot, we can see how with the initial distribution at time 0 shown with blue plot, the variance is pretty big. As we go through time, the variance gets significantly smaller through incorporating sensor measurements and expected measurements. The final position is at [1.012 10.0048].

6. The plot is as below:



In this plot, we changed our initial distribution's mu to [14; 5.5] and the variance to [(5/3)^2 (2/3)^2]. As a result, the final position changes. With the lower variance, the program will look into the area that is within a closer range, as the lower variance means that it is more confident that the right location is closer. Since the chosen mu has the value of x to be 14, the final position that is calculated is 14. This is because the program will find a region that is closer to the initial distribution. As seen from the pdf from grid 40x22, there are possible regions at around x=1, 7, 14. Since the initial distribution chose x=14, then the program will focus to find a suitable position that is near that area, and so we got around [14 10].

7. In this problem, we are given that the robot is stationary. As a result, the system will be linear and we are using Kalman Filter. We can also use Extended Kalman Filter, which can handle both linear and non-linear system, but it would not be more appropriate for linear system. EKF will perform linearization, and it would be meaningless to be done in an already linear system. Only if the robot is not stationary should using Extended Kalman Filter be the better option. When the robot is moving, the system is non-linear as the robot position in X and Y will be based on cosine or sine of the theta.

Location Estimate

1. No. For grid localization, it would be impossible to make an exact guess that is right. This is because grid localization will always have the uncertainty on where the robot is on each grid cell. This would affect the calculation, and it would be impossible to always get the exact location given the perfect range or sensor information. For Kalman filter, despite getting the perfect range measurement, the initial distribution will affect the final result. There may be too many suitable locations based on the sensor measurements, and the robot will decide the estimation based on the initial distribution. So it would also be impossible to find the exact location.
2. Yes. After getting information from the pdfs, if we are told some true information about the location of the robot, we can estimate the real location of the robot. If we are told that y is less than 8, by looking at the 40x22 pdf plot, we can estimate that the robot would be at around the coordinate (14,4), which are shaded in the 40x22 pdf plot. This shows how plotting pdfs may be useful, as it would give you the confidence to make reasonable estimations.