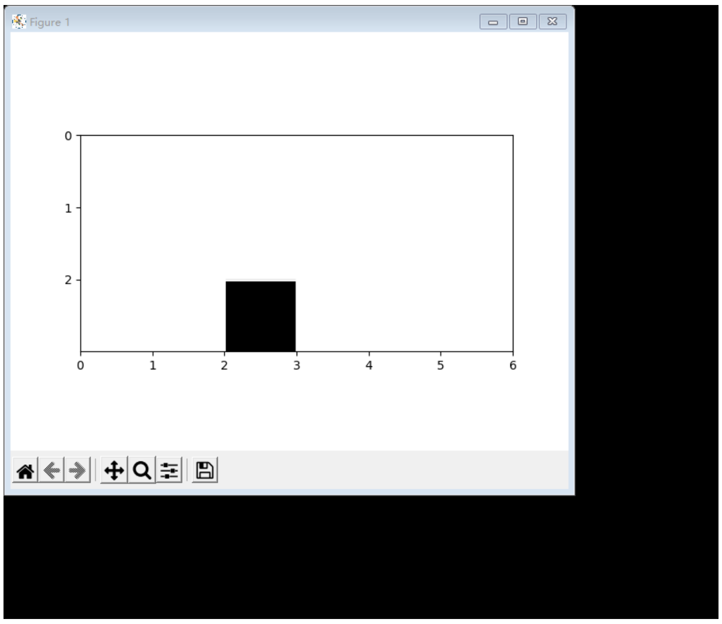
COMP7308 Introduction to unmanned systems 2021

Assignment 4

1 (50 points) Grid-map based localization

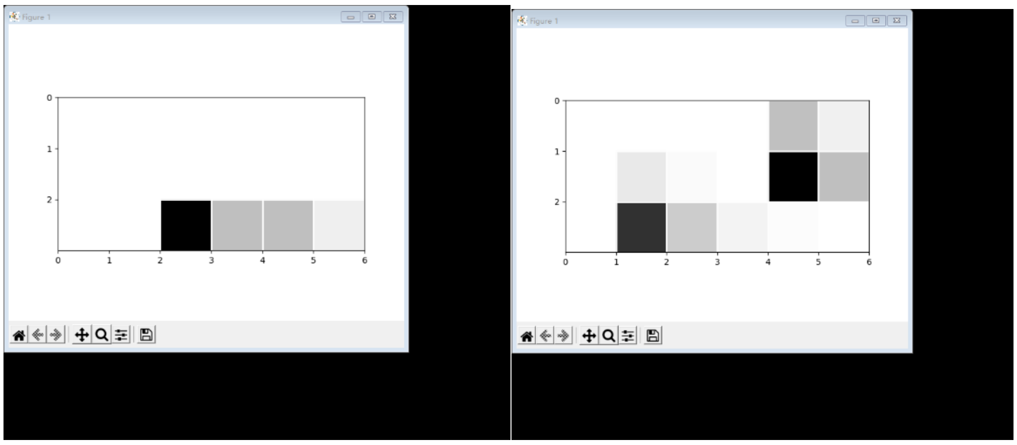
(a) (10 points) Implement the grid-based localization code based on the provided starter code.



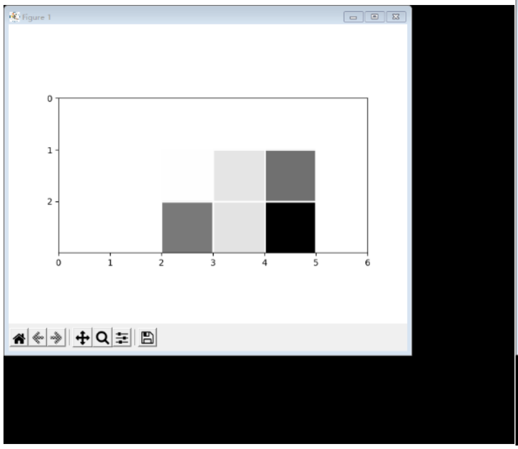
code is in the attachment

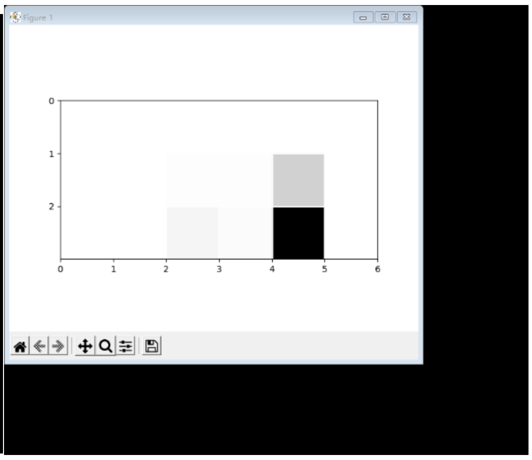
(b) (20 points) Try changing the probabilities that the sensor and action fail with different values. How does this affect the robot's understanding of where it is? Explain your observations.

action fail:0.2 sense fail 0:

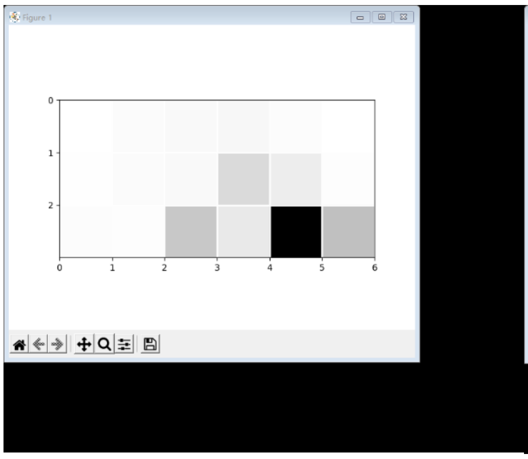


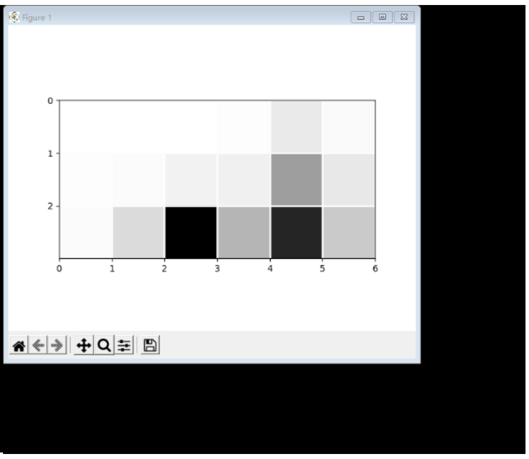
action fail:0 sense fail 0.2:



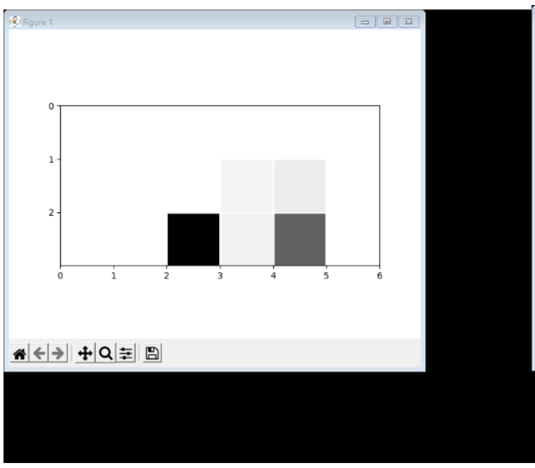


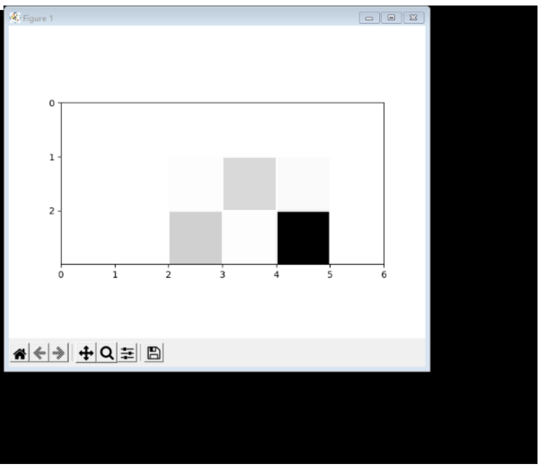
action fail:0.2 sense fail 0.2:



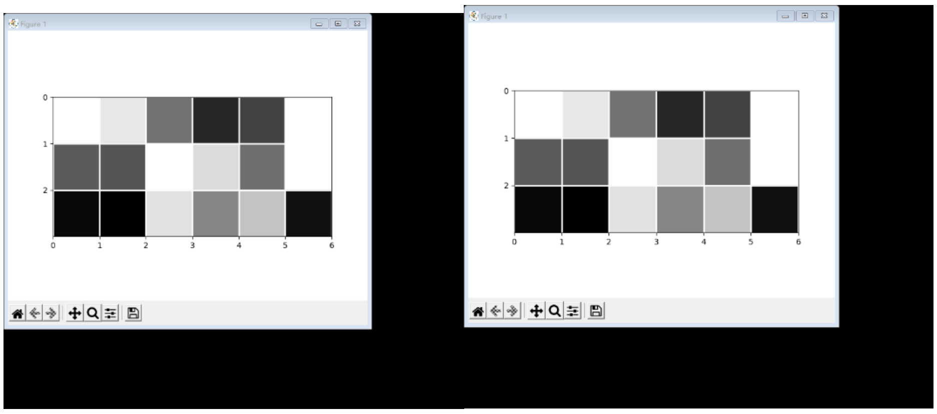


action fail:0 sense fail 0.8:



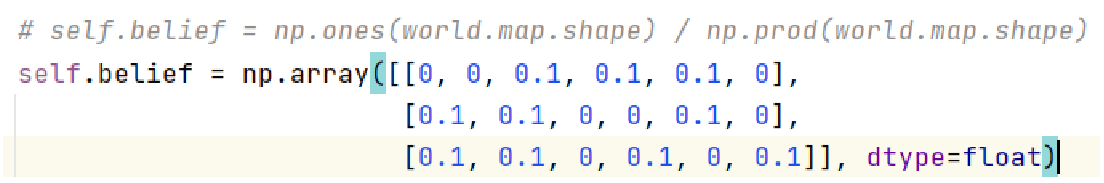


action fail:0.8 sense fail 0:



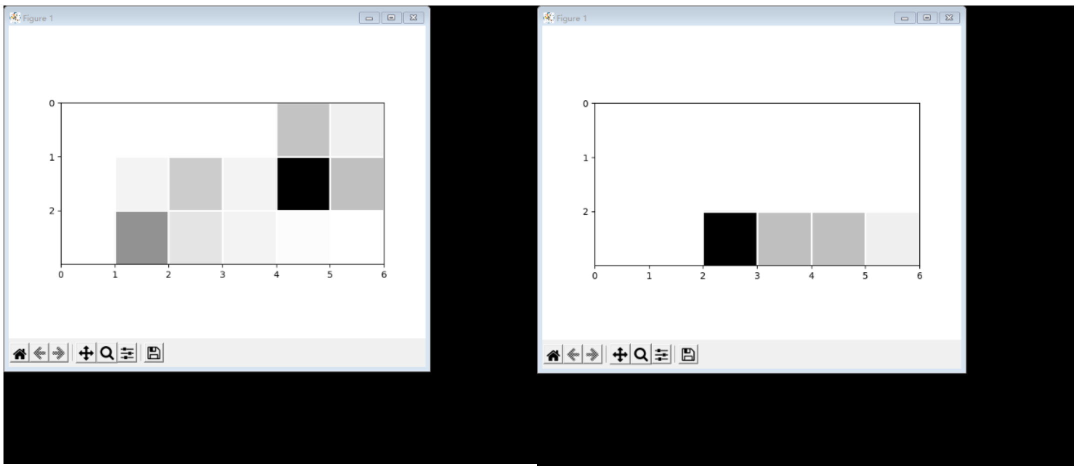
When sense fail and action fail are small, the sensor will affect more than action, because there are only two valuse of the sensor and it will directly change the correction peogress. When sense fail and action fail are big, the action is more important because it means the robot seems not to move at all, so the information from the sensor is not important.

(c) (20 points) In the `__init__` function in the `StateEstimator` class, try changing initial belief from being uniform over the entire grid map to being uniform over the free region in the occupancy map. How does this change the robot's understanding of where it is? Explain your observations.

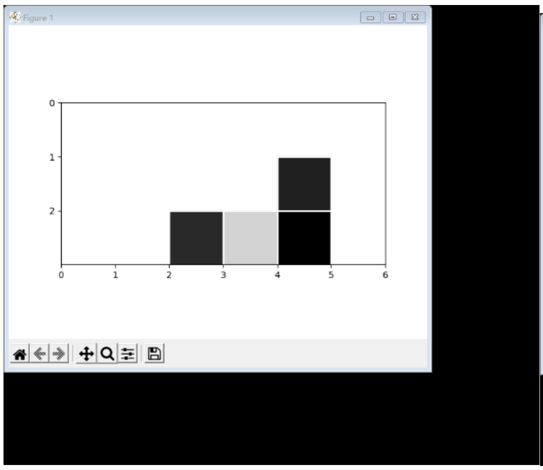


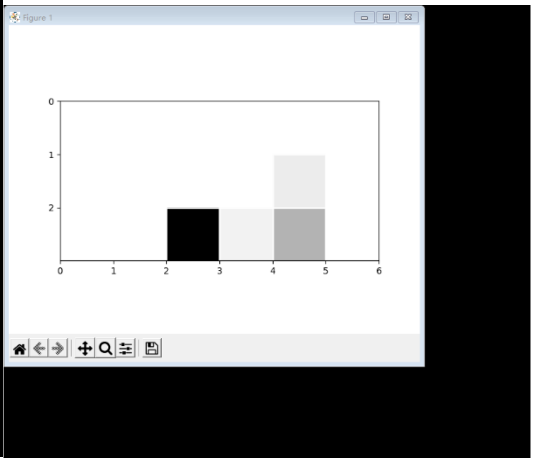
```
# self.belief = np.ones(world.map.shape) / np.prod(world.map.shape)
self.belief = np.array([[0, 0, 0.1, 0.1, 0.1, 0],
                        [0.1, 0.1, 0, 0, 0.1, 0],
                        [0.1, 0.1, 0, 0.1, 0, 0.1]], dtype=float)
```

action fail:0.2 sense fail 0:

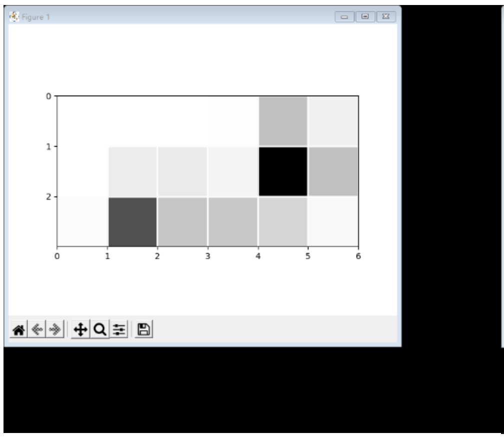


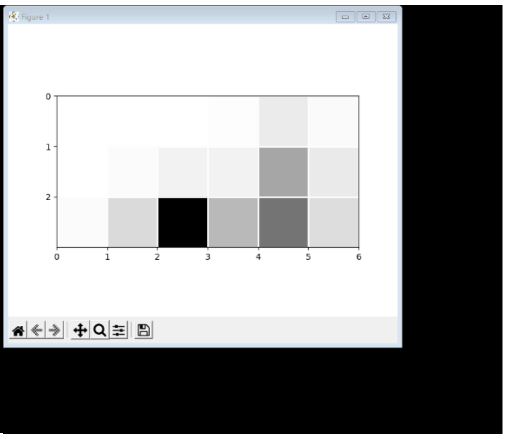
action fail:0 sense fail 0.2:





action fail:0.2 sense fail 0.2:





I don’t think this affect the robot a lot.

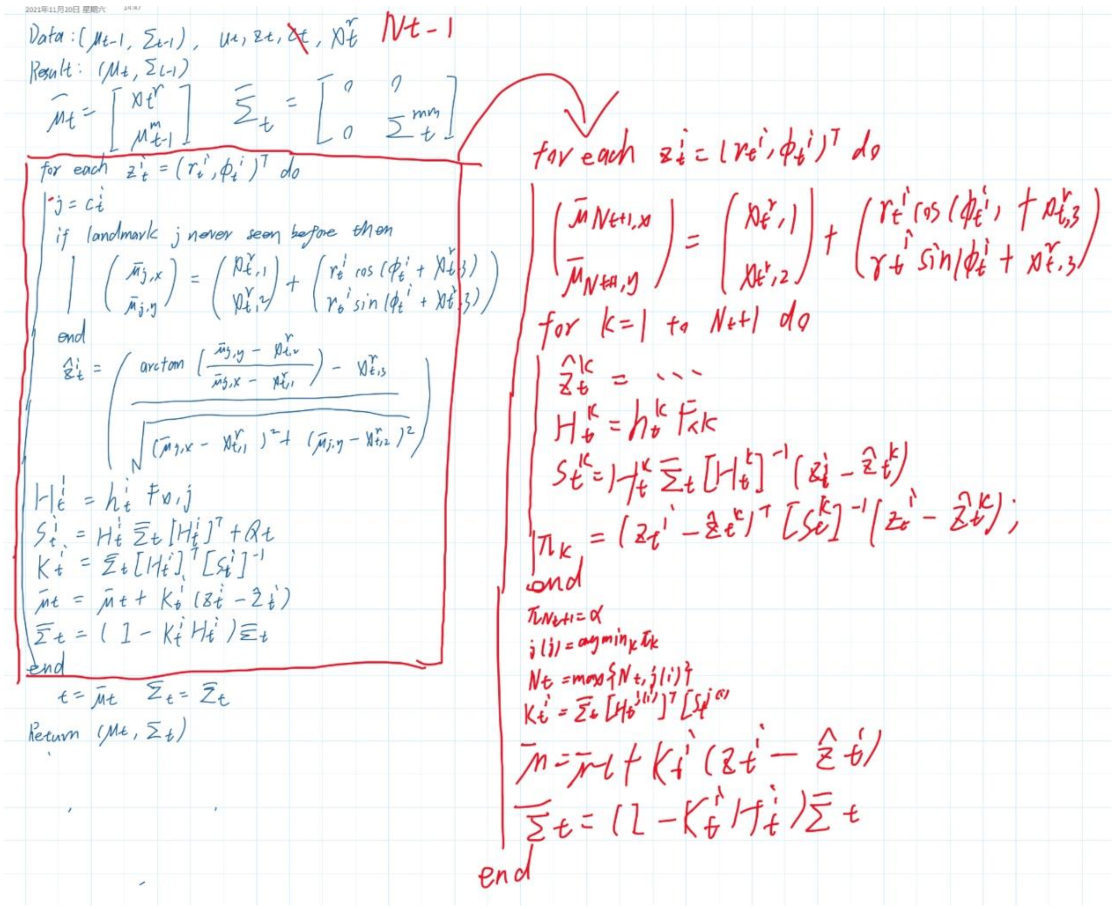
2 (50 points) EKF-based mapping

(a) (15 points) Please write down the EKF-based mapping for the cases when correspondence is known. You should get an algorithm of the similar form as the EKF-SLAM on page 37 of lecture 9.

```
Data: $(\mu_{t-1}, \Sigma_{t-1})$, $u_t, z_t, c_t, x_t^r$
Result: $(\mu_t, \Sigma_{t-1})$
$\bar{\mu}_t = \begin{bmatrix} x_t^r \\ \mu_{t-1}^m \end{bmatrix}$    $\bar{\Sigma}_t = \begin{bmatrix} 0 & 0 \\ 0 & \Sigma_t^{mm} \end{bmatrix}$
  for each $z_t^i = (r_t^i, \phi_t^i)^T$ do
    $j = c_t^i$
    if landmark $j$ never seen before then
      $\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \end{pmatrix} = \begin{pmatrix} x_{t,1}^r \\ x_{t,2}^r \end{pmatrix} + \begin{pmatrix} r_t^i \cos(\phi_t^i + x_{t,3}^r) \\ r_t^i \sin(\phi_t^i + x_{t,3}^r) \end{pmatrix}$
    end
    $\hat{z}_t^i = \begin{pmatrix} \arctan\left(\frac{\bar{\mu}_{j,y} - x_{t,2}^r}{\bar{\mu}_{j,x} - x_{t,1}^r}\right) - x_{t,3}^r \\ \sqrt{(\bar{\mu}_{j,x} - x_{t,1}^r)^2 + (\bar{\mu}_{j,y} - x_{t,2}^r)^2} \end{pmatrix}$
    $H_t^i = h_t^i F_{x,j}$
    $S_t^i = H_t^i \bar{\Sigma}_t [H_t^i]^T + Q_t$
    $K_t^i = \bar{\Sigma}_t [H_t^i]^T [S_t^i]^{-1}$
    $\bar{\mu}_t = \bar{\mu}_t + K_t^i (z_t^i - \hat{z}_t^i)$
    $\bar{\Sigma}_t = (I - K_t^i H_t^i) \bar{\Sigma}_t$
  end
  $\mu_t = \bar{\mu}_t$   $\Sigma_t = \bar{\Sigma}_t$
Return $(\mu_t, \Sigma_t)$
```

(b) (15 points) Please write down the EKF-based mapping for the cases when correspondence is unknown. You should get an algorithm of the similar form as the EKF-SLAM on page 45 of lecture 9.

The idea is seem to (a), that is , remove EKF prediction, and replace robot state with xt.

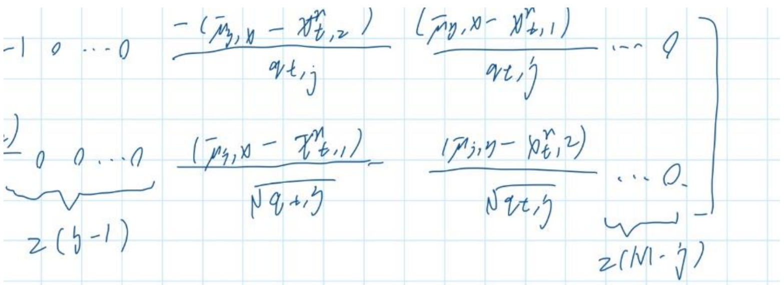


2021年11月20日 星期六

```
Data: $(\mu_{t-1}, \Sigma_{t-1})$, $u_t, z_t,$ ~~$c_t$~~, $x_t^r$   $N_{t-1}$
Result: $(\mu_t, \Sigma_{t-1})$
  $\bar{\mu}_t = \begin{bmatrix} x_t^r \\ \mu_{t-1}^m \end{bmatrix}$   $\bar{\Sigma}_t = \begin{bmatrix} 0 & 0 \\ 0 & \Sigma_t^{mm} \end{bmatrix}$
for each $z_t^i = (r_t^i, \phi_t^i)^T$ do
    $j = c_t^i$
    if landmark $j$ never seen before then
        $\begin{pmatrix} \bar{\mu}_{j,x} \\ \bar{\mu}_{j,y} \end{pmatrix} = \begin{pmatrix} x_{t,1}^r \\ x_{t,2}^r \end{pmatrix} + \begin{pmatrix} r_t^i \cos(\phi_t^i + x_{t,3}^r) \\ r_t^i \sin(\phi_t^i + x_{t,3}^r) \end{pmatrix}$
    end
    $\hat{z}_t^i = \begin{pmatrix} \arctan\left(\frac{\bar{\mu}_{j,y} - x_{t,2}^r}{\bar{\mu}_{j,x} - x_{t,1}^r}\right) - x_{t,3}^r \\ \sqrt{(\bar{\mu}_{j,x} - x_{t,1}^r)^2 + (\bar{\mu}_{j,y} - x_{t,2}^r)^2} \end{pmatrix}$
    $H_t^i = h_t^i F_{x,j}$
    $S_t^i = H_t^i \bar{\Sigma}_t [H_t^i]^T + Q_t$
    $K_t^i = \bar{\Sigma}_t [H_t^i]^T [S_t^i]^{-1}$
    $\bar{\mu}_t = \bar{\mu}_t + K_t^i (z_t^i - \hat{z}_t^i)$
    $\bar{\Sigma}_t = (I - K_t^i H_t^i) \bar{\Sigma}_t$
end
    $\mu_t = \bar{\mu}_t$   $\Sigma_t = \bar{\Sigma}_t$
Return $(\mu_t, \Sigma_t)$
```

```
for each $z_t^i = (r_t^i, \phi_t^i)^T$ do
    $\begin{pmatrix} \bar{\mu}_{N_t+1,x} \\ \bar{\mu}_{N_t+1,y} \end{pmatrix} = \begin{pmatrix} x_{t,1}^r \\ x_{t,2}^r \end{pmatrix} + \begin{pmatrix} r_t^i \cos(\phi_t^i + x_{t,3}^r) \\ r_t^i \sin(\phi_t^i + x_{t,3}^r) \end{pmatrix}$
    for $k = 1$ to $N_t + 1$ do
        $\hat{z}_t^k = \ldots$
        $H_t^k = h_t^k F_{x,k}$
        $S_t^k = H_t^k \bar{\Sigma}_t [H_t^k]^{-1} (z_t^i - \hat{z}_t^k)$
        $\pi_k = (z_t^i - \hat{z}_t^k)^T [S_t^k]^{-1} (z_t^i - \hat{z}_t^k)$;
    end
    $\pi_{N_t+1} = \alpha$
    $j(i) = \operatorname{argmin}_k \pi_k$
    $N_t = \max\{N_t, j(i)\}$
    $K_t^i = \bar{\Sigma}_t [H_t^{j(i)}]^T [S_t^{j(i)}]$
    $\bar{\mu} = \bar{\mu}_t + K_t^i (z_t^i - \hat{z}_t)$
    $\bar{\Sigma}_t = (I - K_t^i H_t^i) \bar{\Sigma}_t$
end
```

(c) (10 points) Write down the H_t^i matrix in these two cases.



$$\begin{bmatrix} -1 & 0 \cdots 0 & \frac{-(\bar{\mu}_{j,x} - x_{t,2}^r)}{q_{t,j}} & \frac{(\bar{\mu}_{j,x} - x_{t,1}^r)}{q_{t,j}} & \cdots 0 \\ ) & \underbrace{0 \quad 0 \cdots 0}_{2(j-1)} & \frac{(\bar{\mu}_{j,x} - x_{t,1}^r)}{\sqrt{q_{t,j}}} & \frac{(\bar{\mu}_{j,y} - x_{t,2}^r)}{\sqrt{q_{t,j}}} & \underbrace{\cdots 0}_{2(M-j)} \end{bmatrix}$$

(d) (10 points) In occupancy-based grid mapping, we mentioned that one important trick is assuming the occupancy states of different cells are conditionally independent. Do you think such independency is leveraged in EKF mapping that you get in (a) and (b)? Explain your judgement.

Yes. If we don't use landmark independency, the sigma will be much bigger (now it is 2*2). So it can save computation complexity.