SDC Homework 3 - Kalman Filter

312512005　黃名諄

1. My Kalman filter code explain:

我根據以下 algorithm 來建構 Kalman filter

1: **Algorithm Kalman_filter**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):
2: $\bar{\mu}_t = A_t\ \mu_{t-1} + B_t\ u_t$
3: $\bar{\Sigma}_t = A_t\ \Sigma_{t-1}\ A_t^T + R_t$
4: $K_t = \bar{\Sigma}_t\ C_t^T (C_t\ \bar{\Sigma}_t\ C_t^T + Q_t)^{-1}$
5: $\mu_t = \bar{\mu}_t + K_t(z_t - C_t\ \bar{\mu}_t)$
6: $\Sigma_t = (I - K_t\ C_t)\ \bar{\Sigma}_t$
7: return μ_t, Σ_t

在程式中，將 Kalman filter 計算 belief 的分布 mean(μ_t)當作 state，所以使

kalman_filter.py × | filtered_path.py

kalman_filter.py > ...

```python
1  import numpy as np
2
3  class KalmanFilter:
4      def __init__(self, x=0, y=0, yaw=0):
5          # State [x, y, yaw]
6          self.state = np.array([x, y, yaw])
7
8          # Transition matrix
9          self.A = np.identity(3)
10         self.B = np.identity(3)
11
12         # State covariance matrix
13         self.S = np.identity(3) * 1
14
15         # Observation matrix
16         self.C = np.array([[1,0,0],[0,1,0]])
17
18         # State transition error
19         self.R = np.array([[1,0,0],[0,1,0],[0,0,1]])
20
21         # Measurement error
22         self.Q = np.array([[3,0],[0,3]])
23
24     def predict(self, u):
25         #assume that the mean of belief is state
26         self.state_predict = np.dot(self.A, self.state) + np.dot(self.B, u)
27         self.S_predict = np.dot(np.dot(self.A, self.S) , self.A.T) + self.R
28
29     def update(self, z):
30         K = np.dot(np.dot(self.S_predict, self.C.T), np.linalg.inv((np.dot(np.dot(self.C, self.S_predict), self.C.T) + self.Q)))
31         self.state = self.state_predict + np.dot(K, (z - np.dot(self.C, self.state_predict)) )
32         self.S = np.dot((np.identity(3)-np.dot(K, self.C)), self.S_predict)
33         return self.state, self.S
```

用 state 來做預測更新，表示的就是分布 mean(μ_t)的預測更新。另外 matrix C、R、Q 的設計會在後續討論。

- Step1 Prediction:

```python
24     def predict(self, u):
25         #assume that the mean of belief is state
26         self.state_predict = np.dot(self.A, self.state) + np.dot(self.B, u)
27         self.S_predict = np.dot(np.dot(self.A, self.S) , self.A.T) + self.R
```

2: $\bar{\mu}_t = A_t\ \mu_{t-1} + B_t\ u_t$
3: $\bar{\Sigma}_t = A_t\ \Sigma_{t-1}\ A_t^T + R_t$

25~26 行對應到 algorithm step 2~3，使用上一刻 state 分布以及 control(u)來預測我的 state(mean) 和 covariance

- Step2 measurement update:

```
29 def update(self, z):
30     K = np.dot(np.dot(self.S_predict, self.C.T), np.linalg.inv((np.dot(np.dot(self.C, self.S_predict), self.C.T) + self.Q)))
31     self.state = self.state_predict + np.dot(K, (z - np.dot(self.C, self.state_predict)) )
32     self.S = np.dot((np.identity(3)-np.dot(K, self.C)), self.S_predict)
33     return self.state, self.S
```

4: $K_t = \bar{\Sigma}_t\, C_t^T (C_t\, \bar{\Sigma}_t\, C_t^T + Q_t)^{-1}$

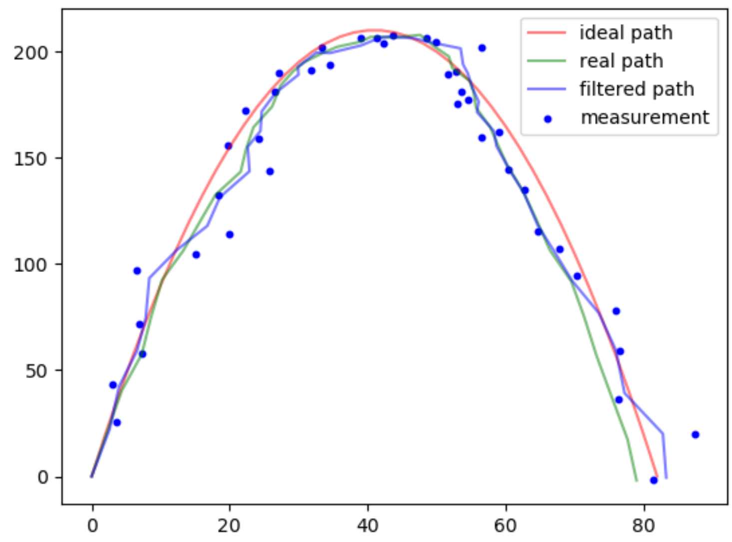
5: $\mu_t = \bar{\mu}_t + K_t (z_t - C_t\, \bar{\mu}_t)$

6: $\Sigma_t = (I - K_t\, C_t)\, \bar{\Sigma}_t$

7: return μ_t, Σ_t

30~33 行對應到 algorithm step 4~7，使用上一步預測結果，計算 kalman gain (K)，再進一步利用這一刻的 measurement(z) to update，最後 return 更新後 belief 的分布情形，其 state(mean) 和 covariance

2. Filtered path result:



3. How you design the observation matrix (C)?

由課本中$Z_t = C_t x_t + \delta_t$，其表示 state(X)到 measurement(z)之間的線性轉換關係且包含 uncertainty，C 則是 observation matrixx96 來將 state(X) 轉換到 measurement(z)，在助教給的 filtered_path.py 中，生成 real path 的部分，其 measurement 只有 x,y，且是直接將 state x,y 加上 uncertainty，表示 state(X)到 measurement(z)的轉換可寫為以下形式

$$\begin{bmatrix} z_x \\ z_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ yaw \end{bmatrix} + \delta_t$$

```
measure_x = x + np.random.normal(0, self.measurement_var, 1)[0]
measure_y = y + np.random.normal(0, self.measurement_var, 1)[0]
```

因此$C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$

4. How you design the covariance matrices(Q, R)?

i. design R:

control and state transition $X_t = A_t x_{t-1} + B_t u_t + \varepsilon_t$ 中，uncertainty 項ε_t是一個 multivariate gaussian distribution 之 noise，而 R 為ε_t之 covariance matrix

根據題目 Control term (u): displacement of robot and yaw change [delta_x, delta_y, delta_yaw] (with **0 mean**, **1 variance** Gaussian noise added to delta_x, delta_y, and delta_yaw, respectively)

Assume 3 個 state 變數的 noise($\varepsilon_x, \varepsilon_y, \varepsilon_{yaw}$)互相獨立，不同變數間 covariance=0，則根據 covariance matrix 定義

$$R = \begin{bmatrix} cov(\varepsilon_x, \varepsilon_x) & cov(\varepsilon_x, \varepsilon_y) & cov(\varepsilon_x, \varepsilon_{yaw}) \\ cov(\varepsilon_y, \varepsilon_x) & cov(\varepsilon_y, \varepsilon_y) & cov(\varepsilon_y, \varepsilon_{yaw}) \\ cov(\varepsilon_{yaw}, \varepsilon_x) & cov(\varepsilon_{yaw}, \varepsilon_y) & cov(\varepsilon_{yaw}, \varepsilon_{yaw}) \end{bmatrix}$$

$$= \begin{bmatrix} var(\varepsilon_x) & 0 & 0 \\ 0 & var(\varepsilon_y) & 0 \\ 0 & 0 & var(\varepsilon_{yaw}) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

此即為 R 之設計方法

ii. design Q:

在 observation function $Z_t = C_t x_t + \delta_t$ 中，uncertainty 項δ_t是一個 multivariate gaussian distribution 之 measurement noise，而 Q 為δ_t之 covariance matrix

根據題目 Measurement (z): Position of [x, y] (with **0 mean**, **3 variance** added to x and y, respectively)

Assume x 和 y 的 measurement noise(δ_x, δ_y)互相獨立，此兩變數間 covariance=0，則根據 covariance matrix 定義

$$Q = \begin{bmatrix} cov(\delta_x, \delta_x) & cov(\delta_x, \delta_y) \\ cov(\delta_y, \delta_x) & cov(\delta_y, \delta_y) \end{bmatrix}$$

$$= \begin{bmatrix} var(\delta_x) & 0 \\ 0 & var(\delta_y) \end{bmatrix} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$$

此即為 Q 之設計方法

5. How will the value of Q and R affect the output of Kalman filter?

，在 Bayes 基礎上可以知道最後輸出的 belief 是由 measurement 及 belief of prediction step (control dynamic)相乘得出，由上課所學及課堂 ppt 中知，在 Kalman filter 中，兩者是以常態分佈的形式表示，如下所示:

$$
bel(x_t) = \eta \quad \underset{\sim N(z_t; C_t x_t, Q_t)}{\underset{\Downarrow}{p(z_t \mid x_t)}} \qquad \underset{\sim N(x_t; \overline{\mu}_t, \overline{\Sigma}_t)}{\underset{\Downarrow}{\overline{bel}(x_t)}}
$$

接著由高斯分布相乘之性質:

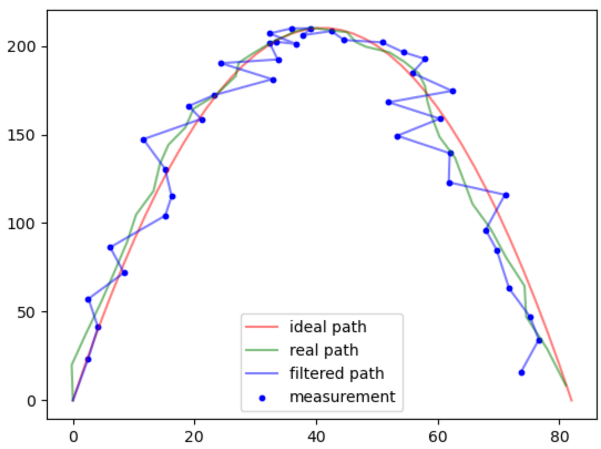
$$
\left.\begin{array}{l} X_1 \sim N(\mu_1, \sigma_1^2) \\ X_2 \sim N(\mu_2, \sigma_2^2) \end{array}\right\} \Rightarrow p(X_1)\cdot p(X_2) \sim N\left(\frac{\sigma_2^2}{\sigma_1^2+\sigma_2^2}\mu_1 + \frac{\sigma_1^2}{\sigma_1^2+\sigma_2^2}\mu_2, \quad \frac{1}{\sigma_1^{-2}+\sigma_2^{-2}}\right)
$$

相乘結果的 mean 是兩者 mean 根據 covariance 的比例組合，covariance 大者(表 uncertainty 較大)，其 mean 佔比會較低，老師上課也有提到，在結果上簡單來說就是結果會更相信 control dynamic prediction 及 measurement 兩者 uncertainty 較低的那個，而 control dynamic prediction 及 measurement 兩者 uncertainty 分別和 R 及 Q 有關因此實際調整兩 matrix R 及 Q 之大小比例來驗證以上推論

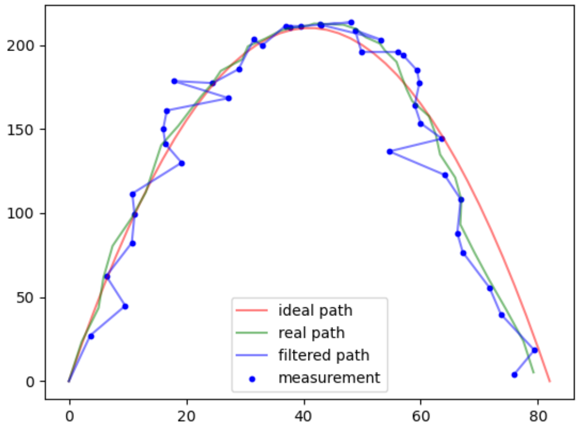
- R 大 Q 小之情形:

  此情形下 control dynamic prediction 之 uncertainty 較 measurement 來的大，使結果更相信 measurement

  i. 設 $R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$， $Q = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$



ii. 設 $R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$， $Q = \begin{bmatrix} 0.001 & 0 \\ 0 & 0.001 \end{bmatrix}$

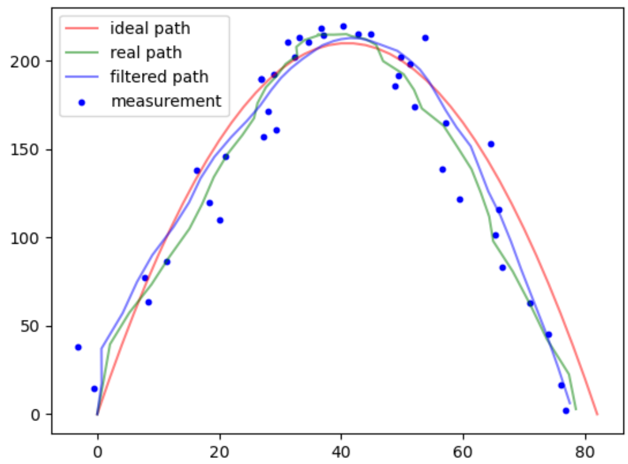


由結果可知，filter 結果更加偏向於 measurement，符合先前的推論

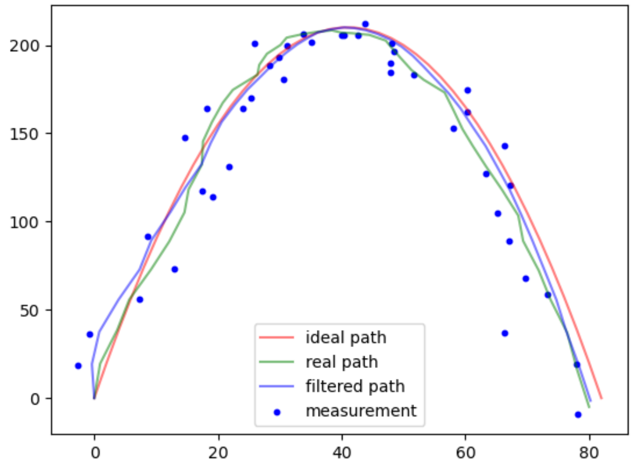
- R 小 Q 大之情形:

  此情形下 measurement 之 uncertainty 較 control dynamic prediction 來的大，使結果更相信 control dynamic prediction

i. 設 $R = \begin{bmatrix} 0.01 & 0 & 0 \\ 0 & 0.01 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}$， $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



ii. 設 $R = \begin{bmatrix} 0.001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.001 \end{bmatrix}$，$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$



由結果可知，filter 結果更加偏向於 control 對 state 之影響，而較不相信 measurement，符合先前的推論，且 R 越小，表 control 的 uncertainty 越小，使結果更偏向 ideal path