Sdc_hw4

311512064 鄧書桓

1. Introduction

   Kalman filter 是一種很厲害的濾波器，它能夠通過我們擁有已知的訊息(觀察得到的數據或過去的狀態)與 belief 的分佈相結合來減輕現實生活中的不確定性，並利用誤差與 noise 來跟蹤我們所要求的目標或測量。簡單來說，就是通過現實生活中感測器的誤差與環境中的不確定性，不斷的迭待更新 belief 與預測，來降低不確定性，進而提高準確率。

2. Briefly explain your code.

   將卡爾曼濾波器的更新修正方式程式化。

```
    self.Q = np.array([[0.5, 0.0, 0.0], [0.0, 0.2, 0.0], [0.0, 0.0, 0.2]])
    # Measurement error
    # self.R = np.array([0.75]).reshape(1, 1)
    self.R = np.array([[0.53, 0], [0, 0.82]])
```

   首先，將 Q 與 R 設置成所想要的矩陣，如何求得與詳細過程會在下面第四點做說明。

   接著，根據以下公式來完成預測:

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

   相對應的 code:

```
def predict(self, u):
    self.x = np.dot(self.A, self.x) + np.dot(self.B, u)
    self.P = np.dot(np.dot(self.A, self.P), self.A.T) + self.Q
```

   利用 numpy 中的 dot()函式來實現，其中 x 為算式中的 $\bar{\mu}_t$ 、P

為 $\bar{\Sigma}_t$，值得注意的是，這邊定義的 Q 與 R 與上課教的相反(此程式的 Q 為公式中的 R)。

最後，透過以下公式來完成更新:

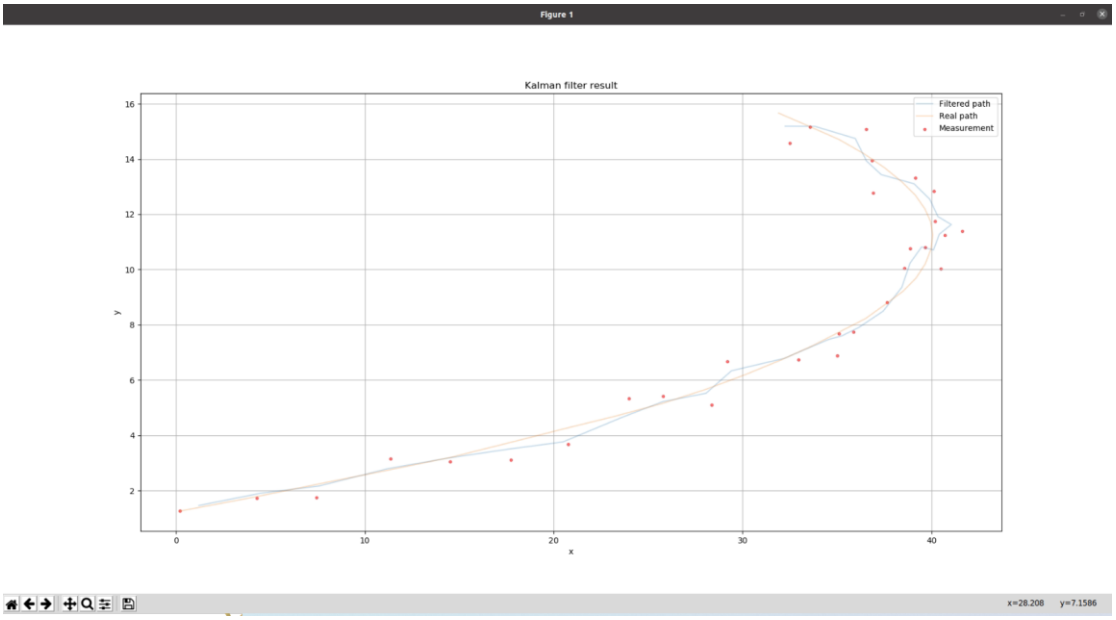
$$K_t = \bar{\Sigma}_t\, C_t^T (C_t\, \bar{\Sigma}_t\, C_t^T + Q_t)^{-1}$$
$$\mu_t = \bar{\mu}_t + K_t(z_t - C_t\, \bar{\mu}_t)$$
$$\Sigma_t = (I - K_t\, C_t)\, \bar{\Sigma}_t$$

相對應的 code:

```
def update(self, z):
    y = z - np.dot(self.H, self.x)
    S = self.R + np.dot(self.H, np.dot(self.P, self.H.T))
    # find inverse matrix
    K = np.dot(np.dot(self.P, self.H.T), np.linalg.inv(S))
    self.x = self.x + np.dot(K, y)
    I = np.eye(3)
    self.P = np.dot(I - np.dot(K, self.H), self.P)
    return self.x, self.P
```

這邊同樣使用 numpy.dot()實現，H 為算是中的 C 、x 為 $\bar{\mu}_t$ 、P 為 $\bar{\Sigma}_t$，由於怕式子太長，有將部分括號內的算式以新的變數命名，如第一行的 y(代表 $(z_t - C_t\, \bar{\mu}_t)$ 與第二行的 s(代表 $C_t\, \bar{\Sigma}_t\, C_t^T + Q_t$)，最終返回 x 與 p 方便做下一次的更新與預測。

3. Include the output image.



由此圖可知，根據所調整的 Q 與 R 能使卡爾曼濾波器製造的路徑與實境的路徑幾乎一樣，但仍有些許的誤差，若想要做到幾乎一樣，我認為不能給定固定的 Q 與 R，需要在每次迭代時一起做更新調整。

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

Q 與 R 先由以下公式推出初始值:

$$\begin{bmatrix} \sigma_x^2 & \sigma_x\sigma_y \\ \sigma_y\sigma_x & \sigma_y^2 \end{bmatrix} \qquad \begin{bmatrix} \sigma_x^2 & \sigma_x\sigma_y & \sigma_x\sigma_z \\ \sigma_y\sigma_x & \sigma_y^2 & \sigma_y\sigma_z \\ \sigma_z\sigma_x & \sigma_z\sigma_y & \sigma_z^2 \end{bmatrix}$$

其中，因為 x、y 與 yaw 相互獨立，因此 Q 與 R 為對角矩陣，又根據題目所給的標準差可得:

Q = [[0.1, 0.0, 0.0],
[0.0, 0.1, 0.0],
[0.0, 0.0, 0.1]]

R = [[0.75, 0.0],
[0.0, 0.75]]

但是，設成這樣 filter 的誤差很大，預測出來的路徑與 real path 差很多，因此還需做一些調整。其中，提高 Q 的值(上課教的 R) 可使 filter 更不相信自己的預測，也就是更相信觀測到的數據;提高 R 的值(上課教的 Q) 可使 filter 更不相信自己的觀測，也就是更相信預測的數據。藉由此觀念來進行 try and err，最終調整成第二點的數據。

5. How will the value of Q and R affect the output of Kalman filter?
   影響的方式在第四點有明確說明，簡單來說，就是他們的值會影響 filter 比較相信預測的數據還是觀測到的數據，我也是根據此原理來調整 Q 與 R。