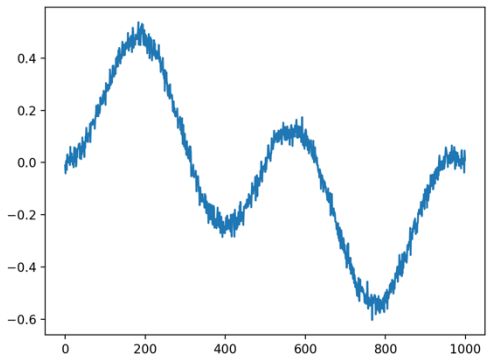
李少琪 人工智慧學程 0860908

Optimization Theory and Application HW3

題目：$x_{cor} = x^{\wedge} + v$, 從 x_{cor} 找出 $x^{\wedge}$。

1. 讀出 x_{cor}

x_{cor}：




```
xcor = np.load('denoise.npy')

x_axis = np.linspace(0, len(xcor)-1, len(xcor))
plt.plot(x_axis, xcor)
plt.show()
```

2. $X^{\wedge} = (I + \lambda D^T D)^{-1} X_{cor}$
   - 設定D

$$D = \begin{bmatrix} -1 & 1 & 0 & \cdots & 0 \\ 0 & -1 & 1 & 0 \cdots & 0 \\ & & \cdot & & \\ & & \cdot & & \\ & & \cdot & & \\ 0 & \cdots & & 0 & -1 \quad 1 \end{bmatrix} : 999 \times 1000$$

```
D = np.zeros((999, 1000))
for i in range(D.shape[0]):
    D[i][i] = -1
    D[i][i+1] = 1
```

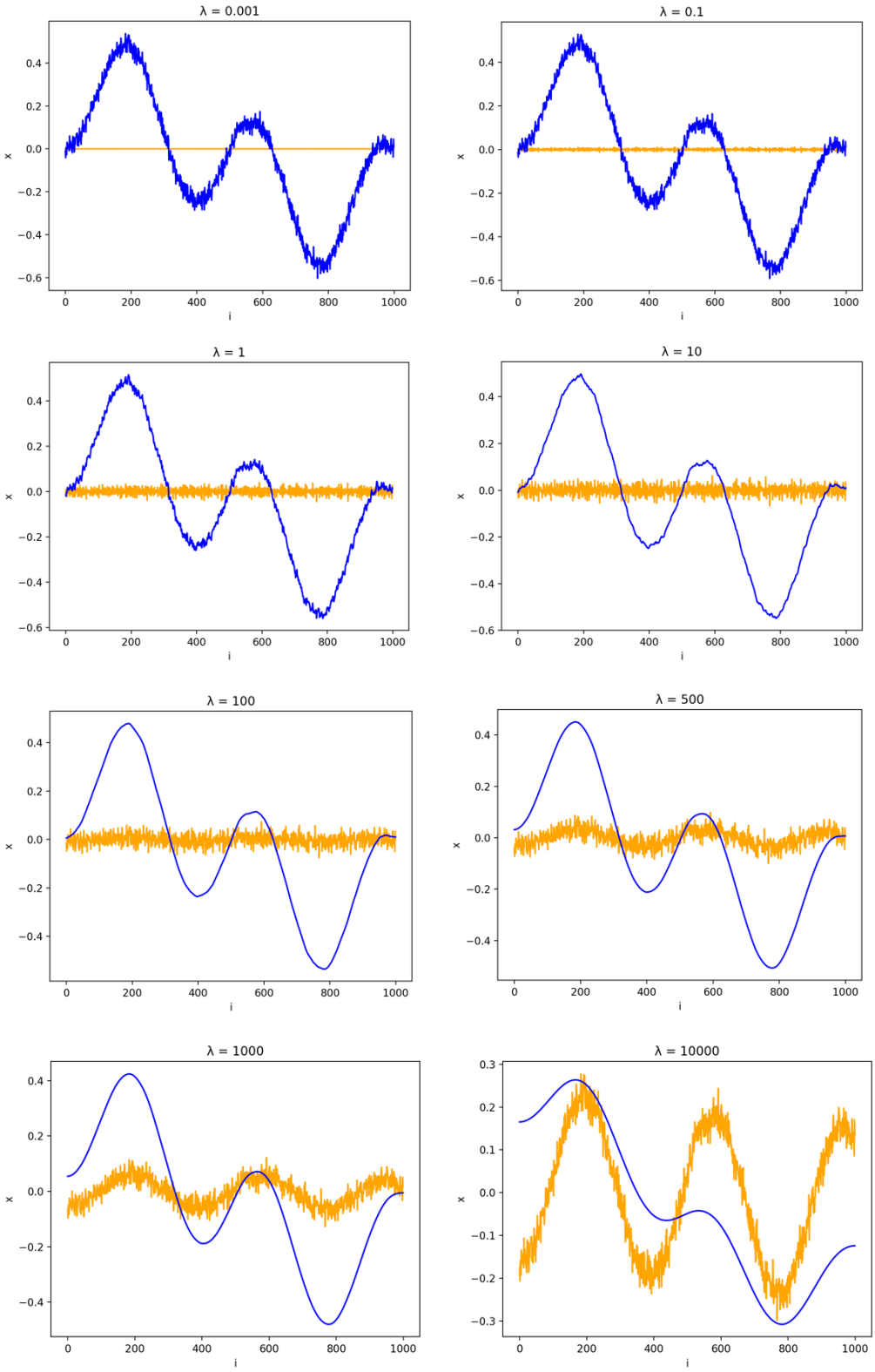
   - $X^{\wedge} = (I + \lambda D^T D)^{-1} X_{cor}$

```
I = np.identity((1000))
lamda = 100
X = np.linalg.inv(I + lamda * D.T.dot(D)).dot(xcor)

plt.plot(x_axis, X)
plt.show()
```

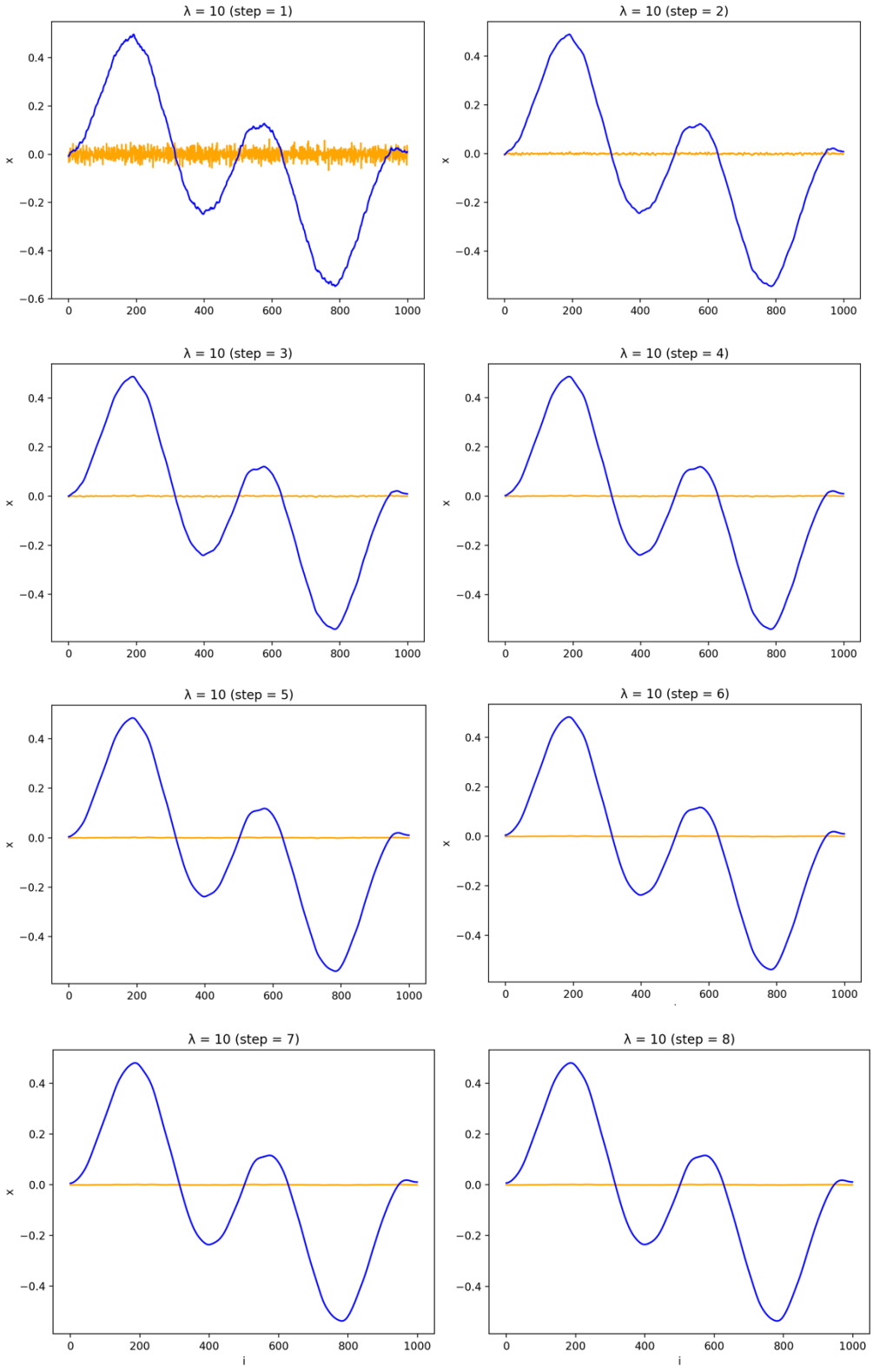
   - $v = x_{cov} - x^{\wedge}$

3. 結果



4. 分析

由上述結果可以發現，當 λ 越小時，可以找出的 noise 越小，但當 λ 過大時，可能會連原本的訊號一起被濾掉。若想避免 λ 設定的問題，我嘗試了將 λ 固定在較小的值，將還原出來的訊號重複做 least squares，但也要避免訓練過多次，結果如下。


λ = 10 (step = 1)
λ = 10 (step = 2)
λ = 10 (step = 3)
λ = 10 (step = 4)
λ = 10 (step = 5)
λ = 10 (step = 6)
λ = 10 (step = 7)
λ = 10 (step = 8)
x
i

```
import numpy as np
from matplotlib import pyplot as plt

# read the data
xcor = np.load('denoise.npy')
# n = np.random.rand(1000)/-100+np.random.rand(1000)/-10+np.random.rand(1000)/10
x_axis = np.linspace(0, len(xcor)-1, len(xcor))
plt.plot(x_axis, xcor)
plt.show()

# set matrix D
D = np.zeros((999, 1000))
for i in range(D.shape[0]):
    D[i][i] = -1
    D[i][i+1] = 1

# calculate X
I = np.identity((1000))
lamda = [0.001, 0.1, 1, 10, 100, 500, 1000, 10000]
for l in lamda:
    X = np.linalg.inv(I + l * D.T.dot(D)).dot(xcor)
    noise = xcor - X

    plt.plot(x_axis, noise, c='orange')
    plt.plot(x_axis, X, c='blue')
    plt.title("λ = {}".format(l))
    plt.xlabel('i')
    plt.ylabel('x')
    plt.show()

# sequential learning
X = xcor
for step in range(0,8):
    XX = np.linalg.inv(I + 10 * D.T.dot(D)).dot(X)
    noise = X - XX

    plt.plot(x_axis, noise, c='orange')
    plt.plot(x_axis, XX, c='blue')
    plt.title("λ = 10 (step = {})".format(step + 1))
    plt.xlabel('i')
    plt.ylabel('x')
    plt.show()

    X = XX
```