

第 6 回演習問題

機械知能工学専攻 知能制御工学コース 17344219 二宮 悠二

問題

テキスト p.112 以降を参考にして例題 6.1 を MATLAB の利用によって解いてみよ。もしくは、他の言語やツールの利用も認める。手計算は認めない。ソースファイルと実行結果をレポートにして提出せよ。

例題 6.1

次の離散時間方程式

$$x(k+1) = x(k) + v(k), \quad x(0) = 0 \quad (1)$$

$$y(k) = x(k) + w(k) \quad (2)$$

について、カルマンフィルタのアルゴリズムを用いて、(2) 式で与えられる観測値 $y(k)$ から (1) 式の状態 $x(k)$ を推定する問題を考える。

システム雑音と観測雑音の分散をそれぞれ $\sigma_v^2 = 1$, $\sigma_w^2 = 2$ とし、初期値を $\hat{x}(0) = 0$, $p(0) = 0$ としたとき、カルマンフィルタのアルゴリズムの流れを計算せよ。なお、この例では、共分散行列、カルマンゲインなど、すべての量はスカラーである。

解答

本課題について、 $k = 300$ までのデータに対しシミュレーションを実行した。その結果を **Fig.1** に、その時のソースコードを Listing 1 に示す。なお、本課題では Python を用いてプログラムの作成を行なった。

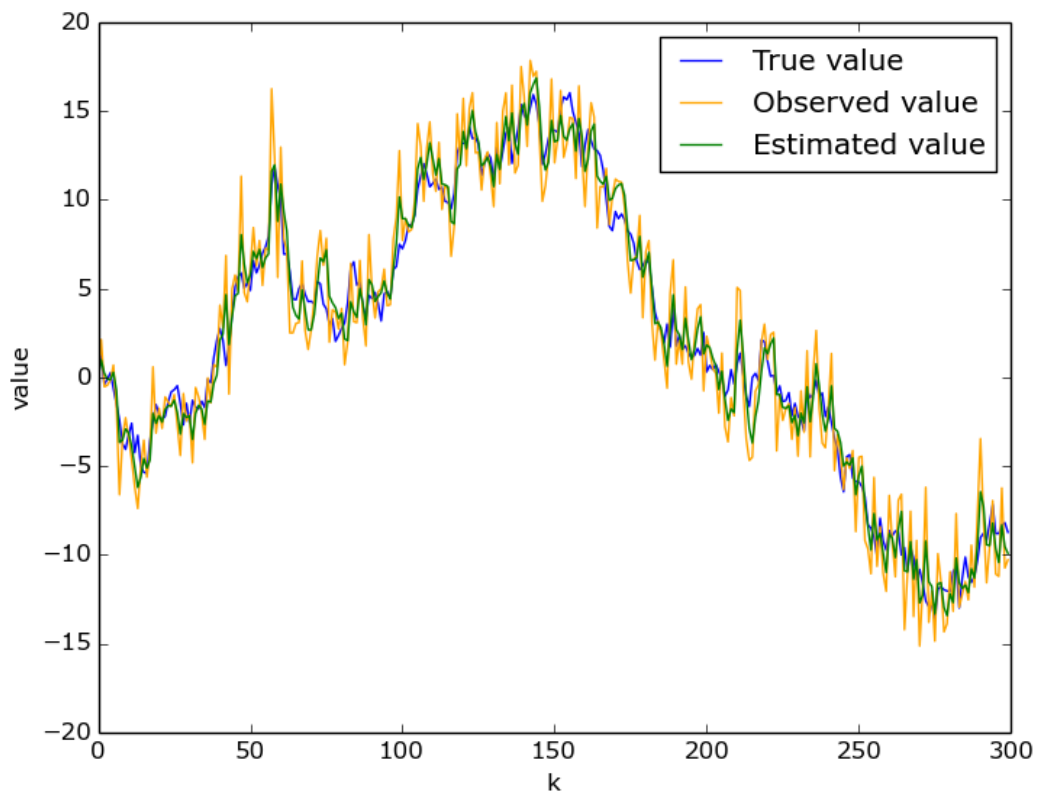


Fig. 1 例題 6.1 のシミュレーション結果

Listing 1: ソースコード

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3  # python3.5
4
5  import numpy as np
6  import matplotlib.pyplot as plt
7
8  def lkf(y,sigmax2,sigmay2,A,b,c,p): # Define the Kalman Filter
9
10     x_hat = []
11     x_hat_pri = 0.0
12
13     for y_k in y:
14         x_pri = A * x_hat_pri
15         p_pri = A * p + sigmax2 * b * b
16         g = (p_pri * c) / (c * p_pri * c + sigmay2)
17         x_hat_next = x_pri + g * (y_k - c * x_pri)
18         p = (1 - g * c) * p_pri
19         x_hat_pri = x_hat_next
20         x_hat.append(x_hat_next)
21     np.array([x_hat])
22

```

```

23     return x_hat
24
25 def main():
26
27     # Set parameter
28     A = 1.0
29     b = 1.0
30     c = 1.0
31
32     p = 0.0
33
34     sigmav2 = 1.0
35     sigmaw2 = 2.0
36
37     # Number of sample
38     n = 300
39     N = np.linspace(0, n-1, n)
40
41     # Make noise
42     v = np.random.normal(0, sigmav2, n)
43     w = np.random.normal(0, sigmaw2, n)
44
45     # Create dataset
46     x = []
47     sum_v = 0
48     for v_k in v:
49         sum_v += v_k
50         x.append(sum_v)
51         np.array([x])
52
53     y = x + w
54
55     # Kalman Filter
56     x_chil = lkf(y, sigmav2, sigmaw2, A, b, c, p)
57
58     # Plot
59     plt.figure(figsize=(8, 6))
60     plt.plot(N, x, label="True_value", color="blue", linewidth=1)
61     plt.plot(N, y, label="Observed_value", color="orange", linewidth=1)
62     plt.plot(N, x_chil, label="Estimated_value", color="green", linewidth=1)
63     plt.xlabel("k")
64     plt.ylabel("value")
65     plt.legend()
66     plt.show()
67
68 if __name__ == '__main__':
69     main()

```

参考文献

- [1] 足立 修一・丸田 一郎, "カルマンフィルタの基礎", 東京電機大学出版局, pp.112-117, 2012.