

```
warnings.simplefilter('ignore')
# Change the delimiter to ';' to match the file
data2 = np.loadtxt("data.csv", encoding='latin-1', delimiter=';', skiprows=1, usecols=(5, 6, 7), dtype=float)
df.head()
```

	Ngày	1	2	3	4	5	6	7	8	9	...	39	40	41	42	43	44	45	46	47	48
0	01/01/2021	964.4	964.4	964.4	964.4	964.4	964.4	964.4	964.4	964.4	...	964.4	964.4	964.4	964.4	964.4	964.4	964.4	964.4	964.4	964.4
1	01/02/2021	1019.7	1019.7	1019.7	1019.7	1019.7	1019.7	1019.7	1019.7	1019.7	...	1019.7	1019.7	1019.7	1019.7	1019.7	1019.7	1019.7	1019.7	1019.7	1019.7
2	01/03/2021	988.4	988.4	988.4	988.4	988.4	988.4	988.4	988.4	988.4	...	988.4	988.4	988.4	988.4	988.4	988.4	988.4	988.4	988.4	988.4
3	01/04/2021	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	1002.0	1002.1	...	1010.8	1010.8	1010.8	1010.8	1010.8	1010.8	1010.8	1010.8	1010.8	1010.8
4	01/05/2021	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	...	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5	1061.5

5 rows x 49 columns

```
# Rename the column if necessary
feats = ['Ngày', '5', '6', '7']
df[feats].head()
```

	Ngày	5	6	7
0	01/01/2021	964.4	964.4	964.4
1	01/02/2021	1019.7	1019.7	1019.7
2	01/03/2021	988.4	988.4	988.4
3	01/04/2021	1002.0	1002.0	1002.0
4	01/05/2021	1061.5	1061.5	1061.5

Đọc dữ liệu và chọn 3 cột 5, 6, 7 .

```

from filterpy.kalman import KalmanFilter

import numpy as np

# Function to apply Kalman Filter
def apply_kalman_filter(data):
    kf = KalmanFilter(dim_x=2, dim_z=1)
    kf.x = np.array([0., 0.]) # initial state (location and velocity)
    kf.F = np.array([[1., 1.], [0., 1.]]) # state transition matrix
    kf.H = np.array([[1., 0.]]) # Measurement function
    kf.P *= 1000. # covariance matrix
    kf.R = 5 # state uncertainty
    kf.Q = np.array([[0.1, 0.1], [0.1, 0.1]]) # process uncertainty

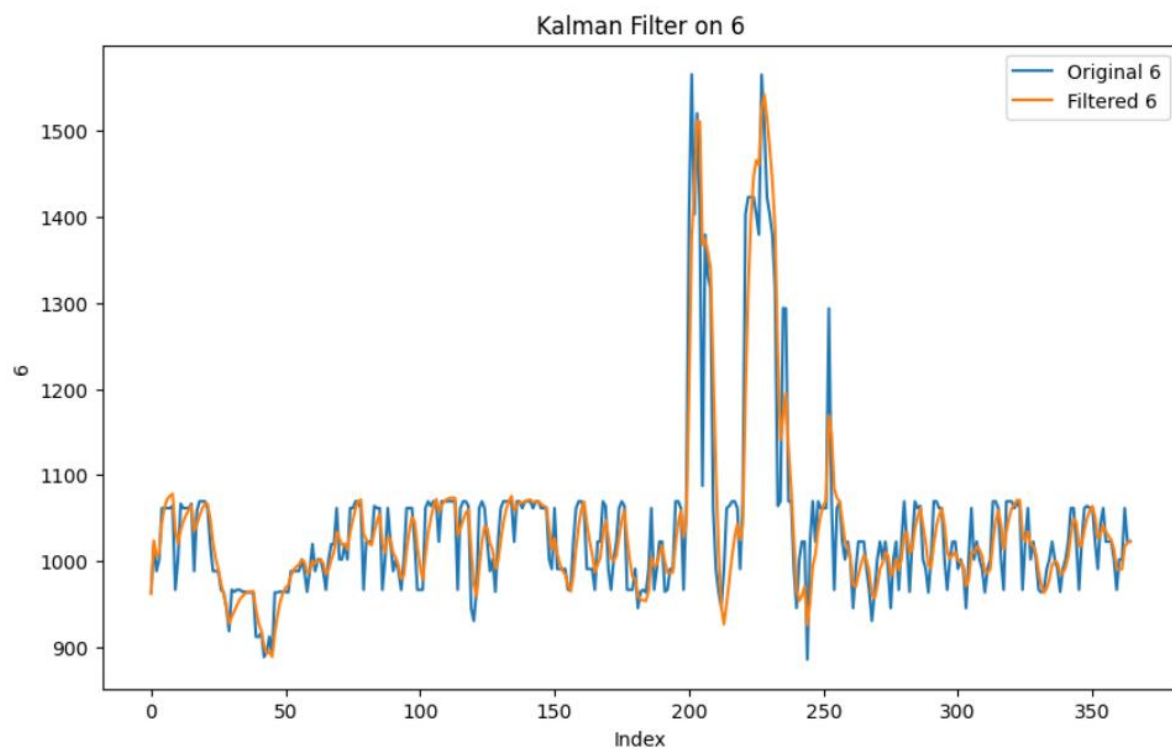
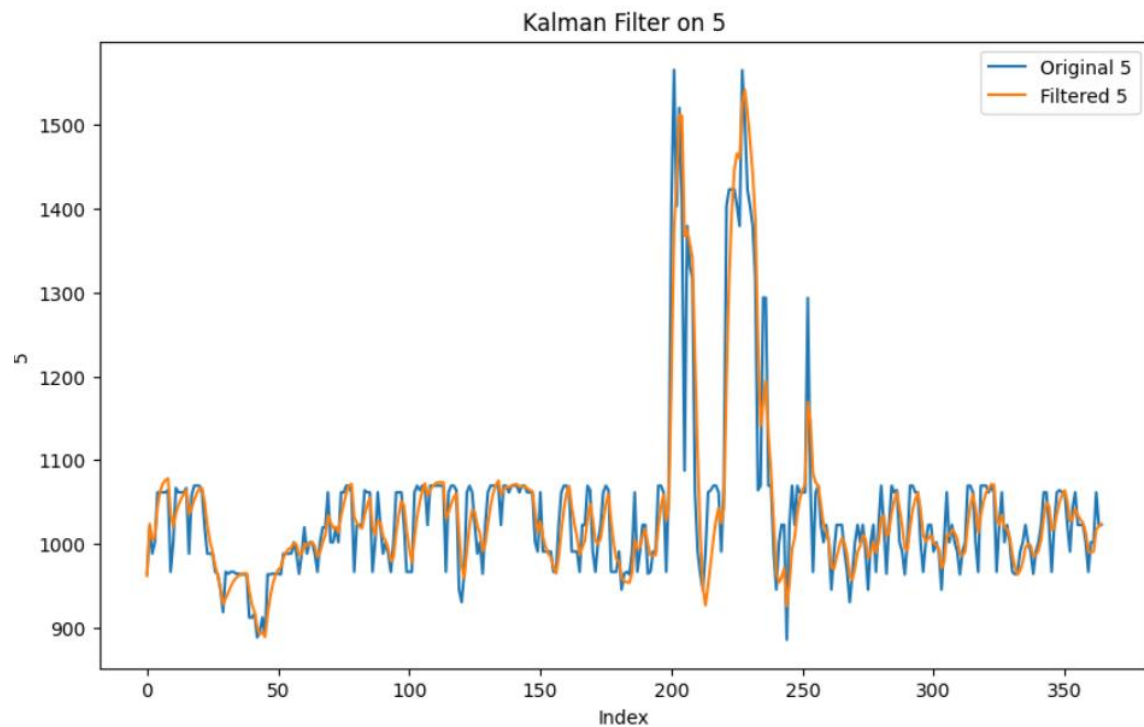
    filtered_data = []
    for z in data:
        kf.predict()
        kf.update(z)
        filtered_data.append(kf.x[0])
    return filtered_data

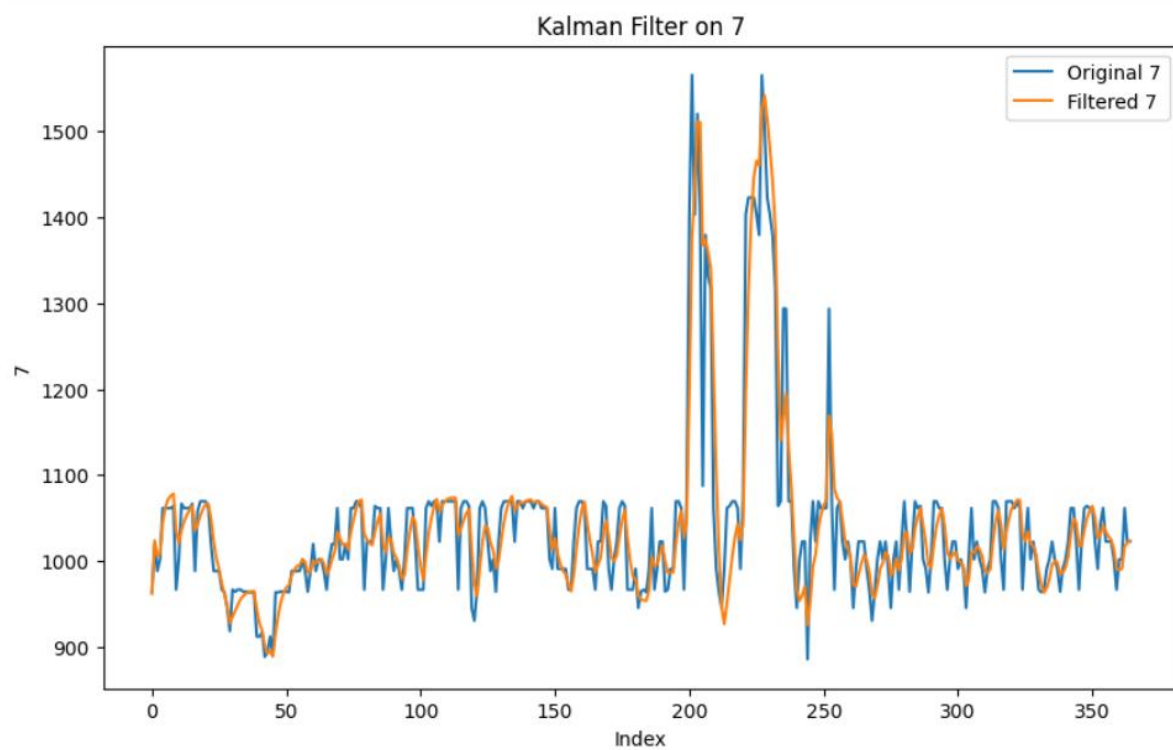
# Áp dụng Kalman Filter cho cột thứ 5
filtered_feature_5 = apply_kalman_filter(X[:, 0])

# So sánh dữ liệu gốc và dữ liệu đã lọc
plt.figure(figsize=(10, 6))
plt.plot(X[:, 0], label='Original ' + df.columns[5])
plt.plot(filtered_feature_5, label='Filtered ' + df.columns[5])
plt.title('Kalman Filter on ' + df.columns[5])
plt.xlabel('Index')
plt.ylabel(df.columns[5])
plt.legend()
plt.show()

```

Khởi tạo mô hình Kalman





Vẽ Biểu đồ kết quả của mô hình Kalman với 3 cột 5 ,6, 7



```

# Vẽ các trạng thái ẩn dự đoán
def plot_hidden_states(hidden_states, title):
    plt.figure(figsize=(10, 2))
    plt.plot(hidden_states, marker='o')
    plt.title(title)
    plt.xlabel("Time")
    plt.ylabel("Hidden State")
    plt.show()

plot_hidden_states(hidden_states, "Predicted Hidden States")

# Vẽ các quan sát mẫu
def plot_observations(observations, title):
    plt.figure(figsize=(10, 2))
    plt.plot(observations)
    plt.title(title)
    plt.xlabel("Time")
    plt.ylabel("Observation")
    plt.show()

plot_observations(X, "Sampled Observations")

```

