Trường Đại học Bách Khoa TP.HCM
Khoa Khoa học và Kỹ thuật Máy tính

KALMAN FILTER

GVHD: Phạm Hoàng Anh
SVTH: Phan Gia Anh

Mục lục

Danh mục hình ảnh

I. KALMAN FILTER LÀ GÌ?

1. Giới thiệu

Thời đại này là một thời đại của sự phát triển khi những giới hạn của con người đang dần được gỡ bỏ. Vào những ngày xưa, khoa học luôn đi trên một con đường chuẩn mực và chính xác, bất kì thứ gì không đạt được sự chính xác tuyệt đối đều không được chấp nhận và xem như là một điều sai, điều này cản trở việc tiếp cận thực tế vì mọi thứ ở thực tế đều không tuyệt đối… cho đến khi xác suất ra đời.

Từ đó mọi thứ đều có một mức độ chính xác riêng và khoa học cũng đã tiến lên rất nhanh từ đó khi đã mở rộng được tầm nhìn của mình và chấp nhận những thứ không hoàn hảo.

Hôm nay nghệ thuật của sự không chắc chắn, của sự thiếu sót sẽ được bộc lộ thông qua một thứ: **Kalman Filter.** Thường những giải thuật hay định lý, hay bất cứ một thứ gì, khi đã được đặt theo tên của một người nào đó thì bạn biết thứ đó sẽ rất là đỉnh và hay ho, và Kalman Filter cũng không phải một ngoại lệ.

Kalman Filter – hay Bộ lọc Kalman – được đặt theo tên Rudolf E. Kálmán (1930-2016), là một bộ lọc chống nhiễu dựa trên một ý tưởng nghe điên rồ nhưng lại cực kỳ hợp lí: Nếu bạn cần đo đạc một thứ không-thể-đo-trực-tiếp được, hãy đo nhiều thứ khác liên quan đến thứ đó và dùng chúng để quyết định thứ bạn không thể đo.

Ví dụ: Giờ học đã hết và bạn chưa về nhà, mẹ của bạn không biết bạn đang ở đâu và bà ấy cũng không gọi cho bạn được, nhưng nếu bà ấy biết những thông tin sau:

- Bà hàng xóm thấy bạn và đám bạn đi qua xóm trên
- Ở xóm trên có sân bóng nhân tạo
- Đôi giày đá banh của bạn không nằm trong hộc giày ở nhà
- Trong giờ học lớp kế bên nói lớp chúng nó đá banh hay nhất khối

Vâng, và không cần phải là một thiên tài để đoán ra là đang có một trận siêu kinh điển giữa lớp bạn và lớp kê bên và đúng, bạn sắp gặp rắc rối to với mẹ mình rồi!

Đấy là một ví dụ về bạn có thể dùng những thông tin liên quan khác nhau để xâu chuỗi tạo thành một thông tin "có độ tin cậy cao hơn", dù đương nhiên thông tin mới ấy có thể không chính xác 100% nhưng nó hoàn toàn đáng để tin. Và dù nói nó không hoàn toàn đáng tin nhưng sự thật là nó đã hoạt động cực kỳ ổn định, ổn định đến mức nó đã được sử dụng trong chương trình Apollo, tàu con thoi NASA, xe tự hành không gian không người lái, tên lửa hành trình,….

Figure 1. Kalman Filter được dùng trong dự án Apollo

2. Nhiệm vụ

Giả sử ta có một mô hình mô tả sự phát triển trạng thái của một hệ thống nào đó như sau:

$$x_k = Fx_{k-1} + Bu_{k-1} + w_{k-1}$$

Trong đó F là ma trận chuyển trạng thái cho vector trạng thái x_{k-1} trước đó, B là ma trận điều khiển đầu vào cho vector điều khiển u_{k-1} và w_{k-1} là vector nhiễu của mô hình, ở đây ta cho sai số Q và $w_{k-1} \sim \mathcal{N}(0, Q)$

Mô hình đo đạc:

$$z_k = Hx_k + v_k$$

Trong đó z_k vector đo đạc, H là ma trận đo đạc và v_k là vector nhiễu trong việc đo đạc có sai số là R và $v_k \sim \mathcal{N}(0, R)$

Vai trò của Kalman Filter là ước lượng vector trạng thái x_k ở thời điểm k, với trạng thái đã biết x_0 và chuỗi đo đạc $z_0, z_1, z_2, .. z_k$ và những thông tin về hệ thống như F, B, H, Q, R.

II. GIẢI THUẬT KALMAN FILTER

Giải thuật Kalman Filter bao gồm 2 giai đoạn: **dự đoán** và **cập nhật.**
Một số quy ước:

- Những biến có ^ có nghĩa là những biến ước lượng, ví dụ $\hat{x}$ là ước lượng của x
- Những biến có chỉ số trên là – (+) có nghĩa là những ước lượng dự đoán(cập nhật)

1. Dự đoán

Ước lượng trạng thái dự đoán:

$$\widehat{x_k}^- = F\widehat{x_{k-1}}^+ + Bu_{k-1}$$

Tương quan lỗi trạng thái dự đoán:

$$P_k^- = FP_{k-1}^+F^T + Q$$

Trong đó P là Tương quan lỗi trạng thái (state error covariance)
Thường thì sau bước dự đoán chúng ta sẽ thầy P_k^- lớn hơn do có cộng vào Q, chứng tỏ rằng sau khi dự đoán thì bộ lọc càng không chắc chắn về ước lượng trạng thái. Nghe có vẻ hợp lý vì dự đoán mà, đương nhiên nó sẽ không chắc chắn rồi vì đã có số liệu gì để so sánh đâu.

2. Cập nhật

Độ lệch đo đạc:

$$\widetilde{y_k} = z_k - H\widehat{x_k}^-$$

Độ phóng đại Kalman:

$$K_k = P_k^-H^T(R + HP_k^-H^T)^{-1}$$

Ước lượng trạng thái cập nhật:

$$\widehat{x_k}^+ = \widehat{x_k}^- + K_k\widetilde{y_k}$$

Tương quan lỗi cập nhật:

$$P_k^+ = (I - K_kH)P_k^-$$

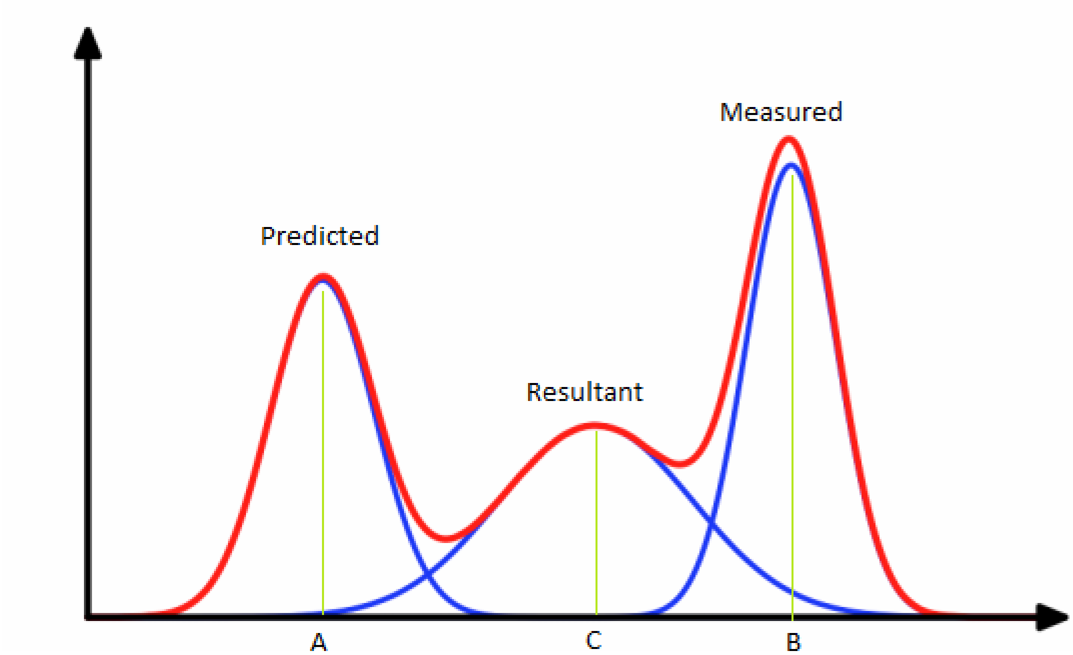

Figure 2. Sự dung hợp giữa dự đoán và thực tế

Đầu tiên ta sẽ tính độ lệch đo đạc được $\widetilde{y_k}$ là sai khác giữa giá trị đo đạc thực sự z_k và giá trị đo đạc ước lượng $H\widehat{x_k}^-$**.** Đây là một bước quan trọng vì giá trị này biểu thị sự liên kết giữa thực tế và tính toán, là cầu nối để điều chỉnh những giá trị ước lượng phía sau.
Độ sai lệch $\widetilde{y_k}$ sau đó được nhân với hệ số Kalman K_k và cộng với giá trị ước lượng trạng thái dự đoán để tạo ước lượng trạng thái cập nhật $\widehat{x_k}^+$**.**
Cuối cùng là cập nhật tương quan lỗi $P_k{}^+$**.** Ở đây ta sẽ thấy $P_k{}^+$ sẽ nhỏ hơn $P_k{}^-$, do lúc này đã có sự tham gia của giá trị z_k đo đạc thực tế nên độ chắc chắn sẽ tăng lên và giá trị tương quan lỗi $P_k{}^+$ sẽ giảm xuống.

Lưu ý rằng Kalman Filter chỉ đạt được kết quả khả quan nhất với điều kiện mô hình quá trình và mô hình đo đạc là tuyến tính, tức là có thể biểu hiện bằng những ma trận F, B, H.

3. Ví dụ thực tế

Bài toán: GPS trong xe hơi của bạn hư rồi, tệ thật! Giờ bạn phải xác định trạng thái xe hơi của bạn dùng Kalman Filter, bao gồm trạng thái về vị trí trong không gian và vận tốc của xe hơi bạn. Bắt đầu nào! (*Ví dụ lấy từ* https://www.intechopen.com/books/introduction-and-implementations-of-the-kalman-filter/introduction-to-kalman-filter-and-its-applications)

Giả sử vector trạng thái được định nghĩa:

$$x = [p^T, v^T]^T$$

Trong đó $p = \left[p_x, p_y, p_z\right]^T$ là vector vị trí và $v = \left[v_x, v_y, v_z\right]^T$ là vector vận tốc.
Vector trạng thái có thể được **dự đoán** nhờ vào những công thức vật lý đã được học như sau:

$$x_k = \begin{bmatrix} p_k \\ v_k \end{bmatrix} = \begin{bmatrix} p_{k-1} + v_{k-1}\Delta t + \frac{1}{2}\tilde{a}_{k-1}\Delta t^2 \\ v_{k-1} + \tilde{a}_{k-1}\Delta t \end{bmatrix}$$

Trong đó $\tilde{a}_{k-1}$ là gia tốc của xe hơi.
Phương trình trên có thể biến thành dạng hồi quy:

$$x_k = \begin{bmatrix} I_{3x3} & I_{3x3}\Delta t \\ 0_{3x3} & I_{3x3} \end{bmatrix} x_{k-1} + \begin{bmatrix} \frac{1}{2} I_{3x3}\Delta t^2 \\ I_{3x3}\Delta t \end{bmatrix} \tilde{a}_{k-1}$$

Khi đó độ nhiễu hệ thống xuất phát từ giá trị gia tốc:

$$a_{k-1} = \tilde{a}_{k-1} + e_{k-1}$$

Ở đây độ nhiễu được định nghĩa $e_{k-1} \sim \mathcal{N}(0, I_{3x3}\sigma_e^2)$. Với ma trận tương quan được định nghĩa $Cov(Ax) = ARA^T$, $Cov(x) = R$, ta nhận được tương quan Q:

$$Q = Cov\left(\begin{bmatrix} \frac{1}{2} I_{3x3}\Delta t^2 \\ I_{3x3}\Delta t \end{bmatrix} I_{3x3}\sigma_e^2\right) = \begin{bmatrix} \frac{1}{2} I_{3x3}\Delta t^2 \\ I_{3x3}\Delta t \end{bmatrix} I_{3x3}\sigma_e^2 \begin{bmatrix} \frac{1}{2} I_{3x3}\Delta t^2 \\ I_{3x3}\Delta t \end{bmatrix}^T$$

$$= \begin{bmatrix} \frac{1}{4} I_{3x3}\Delta t^4 & 0_{3x3} \\ 0_{3x3} & I_{3x3}\Delta t^2 \end{bmatrix} \sigma_e^2$$

Giờ đây ta đã có mô hình trạng thái như sau:

$$x_k = Fx_{k-1} + Bu_{k-1} + w_{k-1}$$

Trong đó

$$F = \begin{bmatrix} I_{3x3} & I_{3x3}\Delta t \\ 0_{3x3} & I_{3x3} \end{bmatrix}$$

$$B = \begin{bmatrix} \frac{1}{2} I_{3x3} \Delta t^2 \\ I_{3x3} \Delta t \end{bmatrix}$$

$$w_{k-1} \sim \mathcal{N}(0, Q)$$

Với:

$$Q = \begin{bmatrix} \frac{1}{4} I_{3x3} \Delta t^4 & 0_{3x3} \\ 0_{3x3} & I_{3x3} \Delta t^2 \end{bmatrix} \sigma_e^2$$

Kết quả đo đạc sẽ là:

$$z_k = \begin{bmatrix} p_k \\ v_k \end{bmatrix} + m_k$$

Tức là theo mô hình đo đạc:

$$z_k = H x_k + m_k$$

thì ta sẽ giả sử:

$$H = I_{6x6}$$

$$m_k \sim \mathcal{N}(0, R)$$

Sau khi đã có những mô hình phân tích, việc tiếp theo là giả lập những giá trị ban đầu:

- $k = 1, 2, 3, \dots, 20$: Ta sẽ xem xét trạng thái ở 20 bước chạy đầu tiên.
- $\Delta t = 1$: Bước nhảy thời gian là 1s cập nhật 1 lần.
- $v_k = [5,5,0]^T$ $(k = 1,2,3, \dots, 20)$: Ở đây để đơn giản ta có thể giả sử xe chạy đều không gia tốc, trên thực tế việc có thêm gia tốc sẽ không ảnh hưởng nhiều đến Kalman Filter.
- $p_0 = [0,0,0]^T$: Vị trí bắt đầu được đặt làm gốc.
- Độ nhiễu của đo đạc gia tốc, vị trí và vận tốc được giả định với phương sai lần lượt là: $0.3^2, 3^2, 0.03^2$. Khi đó ma trận tương quan có giá trị:

$$Q = \begin{bmatrix} \frac{1}{4} I_{3x3} & 0_{3x3} \\ 0_{3x3} & I_{3x3} \end{bmatrix} 0.3^2$$

$$R = \begin{bmatrix} I_{3x3}3^2 & 0_{3x3} \\ 0_{3x3} & I_{3x3}0.03^2 \end{bmatrix}$$

- Giá trị ước lượng ban đầu và tương quan lỗi ban đầu. Ở đây bạn có thể chọn bất cứ giá trị nào, vì các giá trị này đều mang tính dự đoán nên nếu Kalman Filter hoạt động đúng, chúng rốt cuộc cũng sẽ quy về một đường xác định, gợi ý là nên chọn giá trị có độ lệch cao, khi đó sẽ dễ quan sát được hiệu quả của Kalman Filter:

$$\widehat{x_0}^+ = [2, -2, 0, 5, 5.1, 0.1]$$

$$P_0^+ = \begin{bmatrix} I_{3x3}4^2 & 0_{3x3} \\ 0_{3x3} & I_{3x3}0.04^2 \end{bmatrix}$$

- Những giá trị đo đạc z_k sẽ được generate ngẫu nhiên từ mô phỏng Monte Carlo

  Cuối cùng là quan sát biểu hiện của các trạng thái qua 2 lần chạy mô phỏng

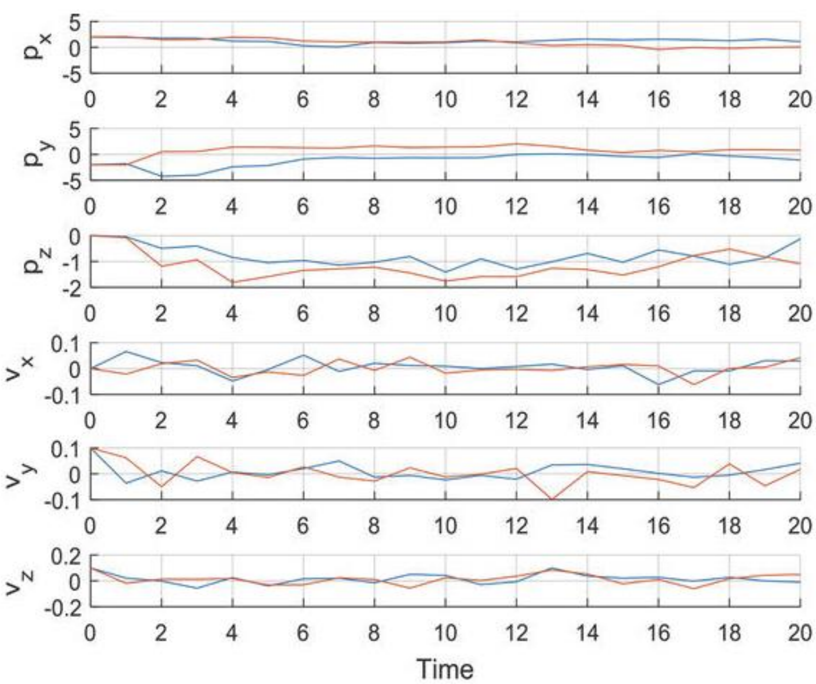

Figure 3. Mô phỏng các giá trị trạng thái trong 2 lần chạy mô phỏng

Ta nhận thấy rằng cả 2 lần mô phỏng đều ra kết quả khác nhau dù điều kiện ban đầu được thiết lập giống nhau, tuy nhiên giá trị vẫn xoay quanh một đường chung là đường đồ thị của các giá trị đo được.

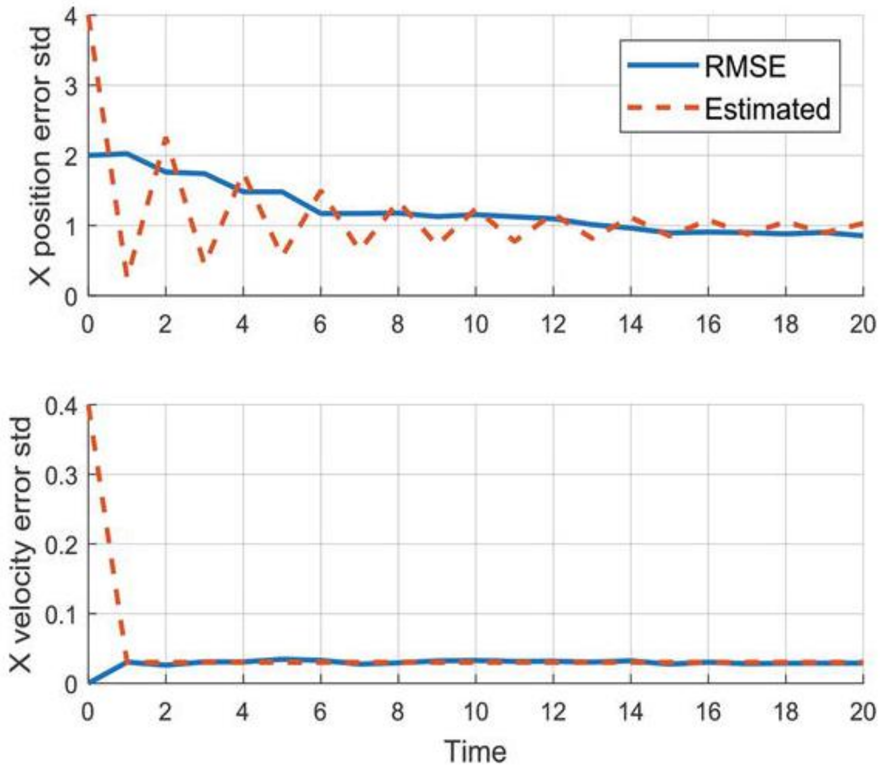

Figure 4. Giá trị RSME(tính toán) và giá trị ước lượng

RSME(Root Mean Square Error) được tính toán dựa vào độ lệch chuẩn của các lỗi ở mỗi bước nhảy thời gian.
Ở đồ thị trên ta thấy Kalman Filter hoạt động khá hiệu quả và tiệm cận với thực tế.

III. MÔ PHỎNG SCILAB

Ta sẽ hiện thực Kalman Filter với nhiệm vụ là lọc nhiễu cho một đồ thị sinusoid

1. Hàm KalmanFilter

Áp dụng những công thức đã xây dựng ở trên, ta xây dựng hàm *KalmanFilter* như sau:

```
function [X, P]= KalmanFilter(zk, x1, p1, F, B, H, Q, R)
    // Prediction
    xkp = F * x1;
    Pkp = F * p1 * F' + Q;
    
    // Calculate residual and Kalman gain
    res = zk - H * xkp;
    K = Pkp * H' / (H * Pkp * H' + R);
    
    // Update
    //X(1:i) = x1;
    X = xkp + K * res;
    //P(1:i) = p1;
    P = Pkp - K * H * Pkp;
endfunction
```

Figure 5. Hàm Kalman Filter hiện thực trong Scilab

Với:

- x1, p1: giá trị vector trạng thái và vector tương quan lỗi tại thời điểm i
- zk: Giá trị đo được
- F, B, H: Ma trận của hệ thống
- Q, R: Ma trận sai số
- X, P: giá trị vector trạng thái và vector tương quan lỗi tại thời điểm i+1

2. Thông số mô phỏng

```
// System Parameter
n = 2;
F = [cos(w*T) -sin(w*T); sin(w*T) cos(w*T)];
B = 0;
H = [1 0];
p0 = [1000 0; 0 0];
R = 1;
Q = 0;
x0 = zeros(n, 1);

// Initialize for loop
x1 = zeros(2, length(t));
p1 = zeros(2, 2, length(t));
p1(:,:,1) = [1000 0; 0 0];
```

Figure 6. Thông số giả lập dùng cho mô phỏng

3. *Kết quả mô phỏng*

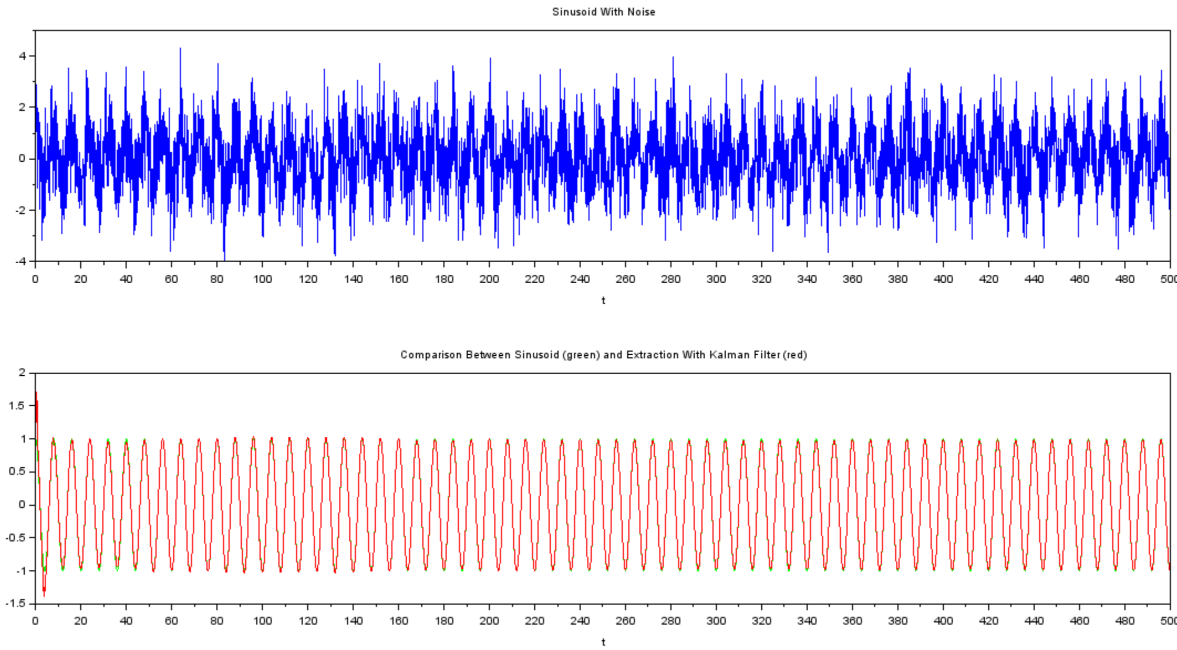

Figure 7. Kết quả mô phỏng loại bỏ tín hiệu nhiễu của Sinusoid

Như đã thấy, sau khi lọc nhiễu dùng Kalman Filter thì tín hiệu đầu ra(màu xanh lá) đã sạch và gần với tín hiệu gốc của sinusoid(màu đỏ).

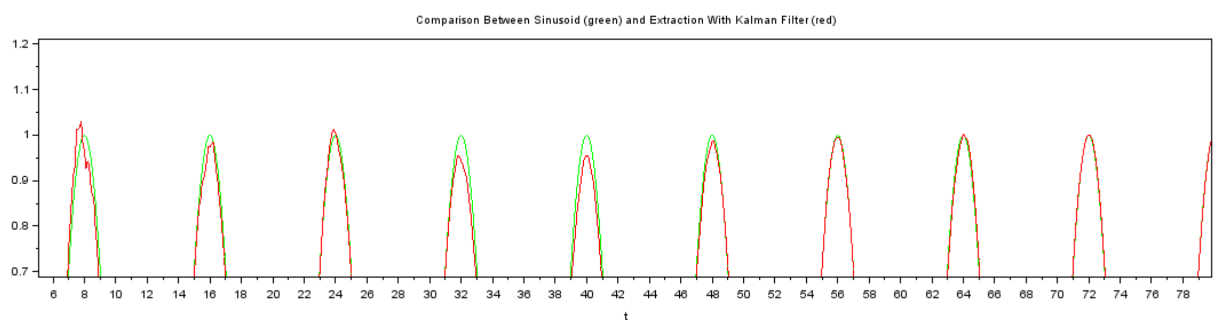

Figure 8. Độ lệch của tín hiệu dùng Kalman Filter và tín hiệu gốc

Sai số là có nhưng rất nhỏ.

IV. PHỤ LỤC VÀ TÀI LIỆU THAM KHẢO

Bạn có thể tham khảo thêm về Extended Kalman Filter tại đây: https://towardsdatascience.com/extended-kalman-filter-43e52b16757d?gi=edd9ffd37d62

[1] Introduction To Kalman Filter and Its Applications
https://www.intechopen.com/books/introduction-and-implementations-of-the-kalman-filter/introduction-to-kalman-filter-and-its-applications
[2] Understanding Kalman Filters
https://www.youtube.com/watch?v=mwn8xhgNpFY&list=PLn8PRpmsu08pzi6EMiYnR-076Mh-q3tWr
[3] How a Kalman filter works, in pictures
https://www.bzarg.com/p/how-a-kalman-filter-works-in-pictures/