Unscented Kalman Filter (UKF)

Dr. **Yuan-Li Cai**

Spring • 2024

0. Outline

考虑如下非线性系统:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, \boldsymbol{w}_k, k) \tag{1}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, k) + \boldsymbol{v}_k \tag{2}$$

其中, $\boldsymbol{x}_k \in \boldsymbol{R}^n$, $\boldsymbol{y}_k \in \boldsymbol{R}^m$, $\boldsymbol{w}_k \in \boldsymbol{R}^q$, $\boldsymbol{v}_k \in \boldsymbol{R}^m$。此外, $E[\boldsymbol{w}_k] = \mathbf{0}$, $E[\boldsymbol{v}_k] = \mathbf{0}$, $E[\boldsymbol{w}_k \boldsymbol{w}_j^{\mathrm{T}}] = \boldsymbol{Q}_k \delta_{kj}$, $E[\boldsymbol{v}_k \boldsymbol{v}_j^{\mathrm{T}}] = \boldsymbol{R}_k \delta_{kj}$, $E[\boldsymbol{w}_k \boldsymbol{v}_j^{\mathrm{T}}] = \mathbf{0}$, $\forall k, j$。

1. EKF 算法回顾

简记

$$\hat{\boldsymbol{x}}_k = \hat{\boldsymbol{x}}_{k|k}, \quad \boldsymbol{P}_k = \boldsymbol{P}_{k|k}$$

$$\hat{\boldsymbol{x}}_k^- = \hat{\boldsymbol{x}}_{k|k-1}, \quad \boldsymbol{P}_k^- = \boldsymbol{P}_{k|k-1}$$

(1) 初始化：$\hat{\boldsymbol{x}}_0 = E\boldsymbol{x}_0, \quad \boldsymbol{P}_0 = \mathrm{var}(\boldsymbol{x}_0)$

(2) 时间修正（time update）：

$$\hat{\boldsymbol{x}}_{k+1}^{-} = \boldsymbol{f}(\hat{\boldsymbol{x}}_k, 0, k) \tag{3}$$

$$\boldsymbol{P}_{k+1}^{-} = \boldsymbol{F}_k \boldsymbol{P}_k \boldsymbol{F}_k^{\mathrm{T}} + \boldsymbol{G}_k \boldsymbol{Q}_k \boldsymbol{G}_k^{\mathrm{T}} \tag{4}$$

(3) 量测修正（measurement update）:

$$\hat{\boldsymbol{y}}_{k+1}^{-} = \boldsymbol{h}(\hat{\boldsymbol{x}}_{k+1}^{-}, k+1) \tag{5}$$

$$\boldsymbol{P}_{k+1}^{\boldsymbol{yy}} = \boldsymbol{H}_{k+1}\boldsymbol{P}_{k+1}^{-}\boldsymbol{H}_{k+1}^{\mathrm{T}} + \boldsymbol{R}_{k+1} \tag{6}$$

$$\boldsymbol{P}_{k+1}^{\boldsymbol{xy}} = \boldsymbol{P}_{k+1}^{-}\boldsymbol{H}_{k+1}^{\mathrm{T}} \tag{7}$$

$$\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1}^{\boldsymbol{xy}}(\boldsymbol{P}_{k+1}^{\boldsymbol{yy}})^{-1} \tag{8}$$

$$\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_{k+1}^{-} + \boldsymbol{K}_{k+1}(\boldsymbol{y}_{k+1} - \hat{\boldsymbol{y}}_{k+1}^{-}) \tag{9}$$

$$\boldsymbol{P}_{k+1} = \boldsymbol{P}_{k+1}^{-} - \boldsymbol{K}_{k+1}\boldsymbol{P}_{k+1}^{\boldsymbol{yy}}\boldsymbol{K}_{k+1}^{\mathrm{T}} \tag{10}$$

其中:

$$
\boldsymbol{F}_k = \left.\frac{\partial \boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{w}_k, k)}{\partial \boldsymbol{x}_k^{\mathrm{T}}}\right|_{\boldsymbol{x}_k=\hat{\boldsymbol{x}}_k, \boldsymbol{w}_k=0}
$$

$$
\boldsymbol{G}_k = \left.\frac{\partial \boldsymbol{F}(\boldsymbol{x}_k, \boldsymbol{w}_k, k)}{\partial \boldsymbol{w}_k^{\mathrm{T}}}\right|_{\boldsymbol{x}_k=\hat{\boldsymbol{x}}_k, \boldsymbol{w}_k=0}
$$

$$
\boldsymbol{H}_{k+1} = \left.\frac{\partial \boldsymbol{h}(\boldsymbol{x}_{k+1}, k+1)}{\partial \boldsymbol{x}_{k+1}^{\mathrm{T}}}\right|_{\boldsymbol{x}_{k+1}=\hat{\boldsymbol{x}}_{k+1}^{-}}
$$

按简记符号，$\hat{\boldsymbol{x}}_{k+1}^{-}$、$\hat{\boldsymbol{y}}_{k+1}^{-}$ 分别表示系统状态及量测的（一步）预测估计。

2. UT 变换

UT: Unscented Transformation

考虑非线性映射:

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) \tag{11}$$

其中, $\boldsymbol{x} \in \boldsymbol{R}^n \sim \mathcal{N}(\bar{x}, \boldsymbol{P}_{\boldsymbol{x}})$, $\boldsymbol{y} \in \boldsymbol{R}^m$。

构造如下加权 sigma 点集:

$$\boldsymbol{\chi}_0 = \bar{\boldsymbol{x}}, \qquad w_0 = \frac{\kappa}{n+\kappa}, i = 0 \tag{12}$$

$$\boldsymbol{\chi}_i = \bar{\boldsymbol{x}} + (\sqrt{(n+\kappa)\boldsymbol{P}_{\boldsymbol{x}}})_i, \qquad w_i = \frac{1}{2(n+\kappa)}, i = 1, \cdots, n \tag{13}$$

$$\boldsymbol{\chi}_i = \bar{\boldsymbol{x}} - (\sqrt{(n+\kappa)\boldsymbol{P}_{\boldsymbol{x}}})_{i-n}, \qquad w_i = \frac{1}{2(n+\kappa)}, i = n+1, \cdots, 2n \tag{14}$$

其中，κ 是可调参数。当 $\boldsymbol{x}$ 为正态分布时，$\kappa = 3 - n$。

那么，精确到 2 阶以上（泰勒级数）有

$$\bar{\boldsymbol{y}} = \sum_{i=0}^{2n} w_i \boldsymbol{y}_i \tag{15}$$

$$\boldsymbol{P}_{\boldsymbol{y}} = \sum_{i=0}^{2n} w_i (\boldsymbol{y}_i - \bar{\boldsymbol{y}})(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^{\mathrm{T}} \tag{16}$$

其中，$\boldsymbol{y}_i = \boldsymbol{h}(\boldsymbol{\chi}_i)$，$i = 0, \cdots, 2n$。

3. SUT 变换

SUT: Scaled Unscented Transformation

$$\boldsymbol{\chi}_0 = \bar{\boldsymbol{x}}, \quad i = 0 \tag{17}$$

$$\boldsymbol{\chi}_i = \bar{\boldsymbol{x}} + (\sqrt{(n+\lambda)\boldsymbol{P}_{\boldsymbol{x}}})_i, \quad i = 1, \cdots, n \tag{18}$$

$$\boldsymbol{\chi}_i = \bar{\boldsymbol{x}} - (\sqrt{(n+\lambda)\boldsymbol{P}_{\boldsymbol{x}}})_{i-n}, \quad i = n+1, \cdots, 2n \tag{19}$$

$$w_i^{(m)} = \begin{cases} \lambda/(n+\lambda), & i=0 \\ 1/2(n+\lambda), & i=1,2,\cdots,2n \end{cases} \tag{20}$$

$$w_i^{(c)} = \begin{cases} \lambda/(n+\lambda)+(1-\alpha^2+\beta), & i=0 \\ 1/2(n+\lambda), & i=1,2,\cdots,2n \end{cases} \tag{21}$$

其中，可调参数 $0 \leq \alpha \leq 1$(一般可选 $\alpha = 10^{-3}$)，β 根据 $\boldsymbol{x}$ 先验知识选取(正态分布时，$\beta = 2$ 为最优)，κ 通常取为 0，$\lambda = (n+\kappa)\alpha^2 - n$.

此时

$$\bar{\boldsymbol{y}} = \sum_{i=0}^{2n} w_i^{(m)} \boldsymbol{y}_i \tag{22}$$

$$\boldsymbol{P}_{\boldsymbol{y}} = \sum_{i=0}^{2n} w_i^{(c)} (\boldsymbol{y}_i - \bar{\boldsymbol{y}})(\boldsymbol{y}_i - \bar{\boldsymbol{y}})^{\mathsf{T}} \tag{23}$$

其中，$\boldsymbol{y}_i = \boldsymbol{h}(\boldsymbol{\chi}_i)$，$i = 0, \cdots, 2n$。

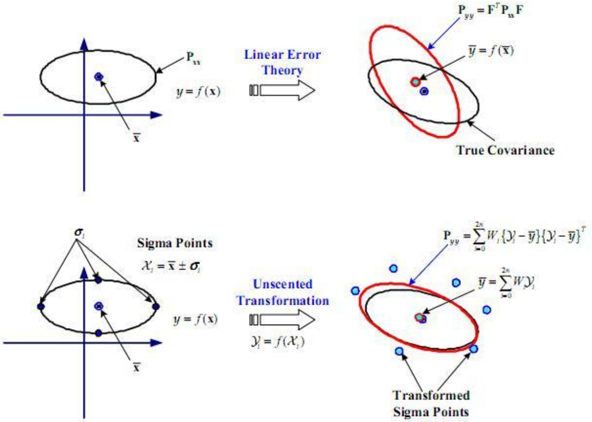

Figure 1: UT 变换示意图

4. UKF 算法

(1) 扩展状态:

$$\boldsymbol{x}_k^a = \begin{bmatrix} \boldsymbol{x}_k \\ \boldsymbol{w}_k \end{bmatrix}, \boldsymbol{P}_k^a = \begin{bmatrix} \boldsymbol{P}_k & \boldsymbol{P}_k^{\boldsymbol{xw}} \\ \boldsymbol{P}_k^{\boldsymbol{wx}} & \boldsymbol{Q}_k \end{bmatrix} \tag{24}$$

(2) 选择 sigma 点: 依据 $(\hat{\boldsymbol{x}}_k^a, \boldsymbol{P}_k^a)$ 生成 $L(= 2(n+q)) + 1$ 个 sigma 点 $(\boldsymbol{\chi}_{i,k}^a)$, $i = 0, 1, \cdots, L$。

(3) 时间修正:

$$\boldsymbol{\chi}_{i,k+1} = \boldsymbol{f}(\chi_{i,k}^{a}, k) \tag{25}$$

$$\hat{\boldsymbol{x}}_{k+1}^{-} = \sum_{i=0}^{L} w_i^{(m)} \boldsymbol{\chi}_{i,k+1} \tag{26}$$

$$\boldsymbol{P}_{k+1}^{-} = \sum_{i=0}^{L} w_i^{(c)} (\boldsymbol{\chi}_{i,k+1} - \hat{\boldsymbol{x}}_{k+1}^{-})(\boldsymbol{\chi}_{i,k+1} - \hat{\boldsymbol{x}}_{k+1}^{-})^{\mathrm{T}} \tag{27}$$

(4) 量测修正:

$$\boldsymbol{y}_{i,k+1} = \boldsymbol{h}(\boldsymbol{\chi}_{i,k+1}, k+1) \tag{28}$$

$$\hat{\boldsymbol{y}}_{k+1}^{-} = \sum_{i=0}^{L} w_i^{(m)} \boldsymbol{y}_{i,k+1} \tag{29}$$

$$\boldsymbol{P}_{k+1}^{\boldsymbol{yy}} = \sum_{i=0}^{L} w_i^{(c)} (\boldsymbol{y}_{i,k+1} - \hat{\boldsymbol{y}}_{k+1}^{-})(\boldsymbol{y}_{i,k+1} - \hat{\boldsymbol{y}}_{k+1}^{-})^{\mathrm{T}} + \boldsymbol{R}_{k+1} \tag{30}$$

$$P_{k+1}^{\boldsymbol{xy}} = \sum_{i=0}^{L} w_i^{(c)} (\boldsymbol{\chi}_{i,k+1} - \hat{\boldsymbol{x}}_{k+1}^{-})(\boldsymbol{y}_{i,k+1} - \hat{\boldsymbol{y}}_{k+1}^{-})^{\mathrm{T}} \tag{31}$$

$$\begin{cases} \boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1}^{\boldsymbol{xy}}(\boldsymbol{P}_{k+1}^{\boldsymbol{yy}})^{-1} \\ \hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_{k+1}^{-} + \boldsymbol{K}_{k+1}(\boldsymbol{y}_{k+1} - \hat{\boldsymbol{y}}_{k+1}^{-}) \\ \boldsymbol{P}_{k+1} = \boldsymbol{P}_{k+1}^{-} - \boldsymbol{K}_{k+1}\boldsymbol{P}_{k+1}^{\boldsymbol{yy}}\boldsymbol{K}_{k+1}^{\mathrm{T}} \end{cases} \tag{32}$$

5. UKF 简化算法

考虑如下加性噪声非线性系统:

$$\boldsymbol{x}_{k+1} = \boldsymbol{f}(\boldsymbol{x}_k, k) + \boldsymbol{w}_k \tag{33}$$

$$\boldsymbol{y}_k = \boldsymbol{h}(\boldsymbol{x}_k, k) + \boldsymbol{v}_k \tag{34}$$

(1) 选择 sigma 点: 依据 $(\hat{\boldsymbol{x}}_k, \boldsymbol{P}_k)$ 生成 $L(=2n)+1$ 个 sigma 点 $\boldsymbol{\chi}_{i,k}$, $i = 0, 1, \cdots, L$。

(2) 时间修正:

$$\boldsymbol{\chi}_{i,k+1} = \boldsymbol{f}(\boldsymbol{\chi}_{i,k}, k) \tag{35}$$

$$\hat{\boldsymbol{x}}_{k+1}^{-} = \sum_{i=0}^{L} \boldsymbol{w}_i^{(m)} \boldsymbol{\chi}_{i,k+1} \tag{36}$$

$$\boldsymbol{P}_{k+1}^{-} = \sum_{i=0}^{L} \boldsymbol{w}_i^{(c)} (\boldsymbol{\chi}_{i,k+1} - \hat{\boldsymbol{x}}_{k+1}^{-})(\boldsymbol{\chi}_{i,k+1} - \hat{\boldsymbol{x}}_{k+1}^{-})^{\mathrm{T}} + \boldsymbol{Q}_k \tag{37}$$

(3) 量测修正:

$$\boldsymbol{y}_{i,k+1} = \boldsymbol{h}(\boldsymbol{\chi}_{i,k+1}, k+1) \tag{38}$$

$$\hat{\boldsymbol{y}}_{k+1}^{-} = \sum_{i=0}^{L} \boldsymbol{w}_i^{(m)} \boldsymbol{y}_{i,k+1} \tag{39}$$

$$\boldsymbol{P}_{k+1}^{\boldsymbol{yy}} = \sum_{i=0}^{L} \boldsymbol{w}_i^{(c)} (\boldsymbol{y}_{i,k+1} - \hat{\boldsymbol{y}}_{k+1}^{-})(\boldsymbol{y}_{i,k+1} - \hat{\boldsymbol{y}}_{k+1}^{-})^{\mathrm{T}} + \boldsymbol{R}_{k+1} \tag{40}$$

$$\boldsymbol{P}_{k+1}^{\boldsymbol{xy}} = \sum_{i=0}^{L} \boldsymbol{w}_i^{(c)} (\boldsymbol{\chi}_{i,k+1} - \hat{\boldsymbol{x}}_{k+1}^{-})(\boldsymbol{y}_{i,k+1} - \hat{\boldsymbol{y}}_{k+1}^{-})^{\mathrm{T}} \tag{41}$$

$$
\begin{cases}
\boldsymbol{K}_{k+1} = \boldsymbol{P}_{k+1}^{\boldsymbol{xy}} (\boldsymbol{P}_{k+1}^{\boldsymbol{yy}})^{-1} \\
\hat{\boldsymbol{x}}_{k+1} = \hat{\boldsymbol{x}}_{k+1}^{-} + \boldsymbol{K}_{k+1} (\boldsymbol{y}_{k+1} - \hat{\boldsymbol{y}}_{k+1}^{-}) \\
\boldsymbol{P}_{k+1} = \boldsymbol{P}_{k+1}^{-} - \boldsymbol{K}_{k+1} \boldsymbol{P}_{k+1}^{\boldsymbol{yy}} \boldsymbol{K}_{k+1}^{\mathrm{T}}
\end{cases}
\tag{42}
$$

[Remark]

- 矩阵平方根可以采用 Cholesky 分解算法;
- UKF 与 EKF 计算量相当或更小，但不需要计算 Jacob 矩阵，滤波精度及计算稳定性均更好;
- 有许多改进，例如平方根滤波算法、自适应算法等;
- 除应用于非线性状态估计外，也可以应用于参数估计、机器学习、信号处理、时间序列预测、图像处理等领域。

6. 仿真算例

[例 1] 考虑非线性系统

$$x_{k+1} = 1 + \sin\frac{k\pi}{25} + \frac{1}{2}x_k + w_k$$

$$y_k = \begin{cases} \frac{1}{2}x_k^2 + v_k, k \leq 30 \\ \frac{1}{2}x_k + v_k, k > 30 \end{cases}$$

其中，$w_k \sim \mathcal{N}(0, 10^{-5})$，$v_k \sim \Gamma(3, 0.5)$，$x_0 \sim \mathcal{N}(1, \frac{3}{4})$。

UKF 算法参数: $\alpha = 1, \beta = 2, \kappa = 0$。均方根误差 (root-mean-square-error, RMSE) 比较见表1.

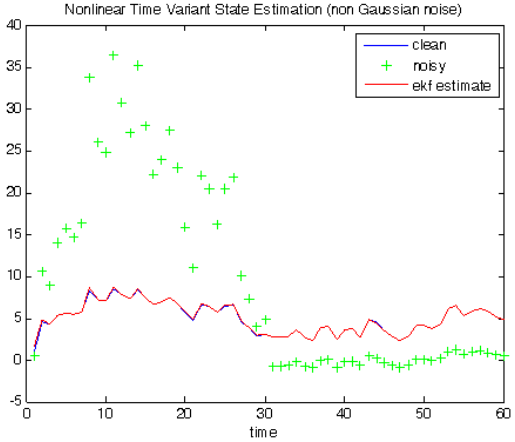

Figure 2: EKF 滤波

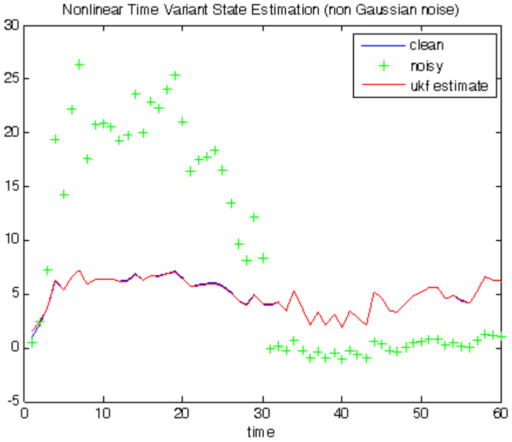

Figure 3: UKF 滤波

Table 1: 均方根误差比较

算法	RMSE	备注
EKF	0.113	图2
UKF	0.043	图3

[例 2] 考虑如下非线性系统

$$x_{k+1} = \frac{x_k}{2} + \frac{5x_k}{1+x_k^2} + 8\cos(0.4k) + w_k$$

$$y_k = \frac{x_k^2}{20} + v_k$$

其中，$x_0 \sim \mathcal{N}(0,5)$；$w_k \sim \mathcal{N}(0,10)$，$v_k \sim \mathcal{N}(0,1)$，二者均为白噪声，相互独立。仿真结果见图4.

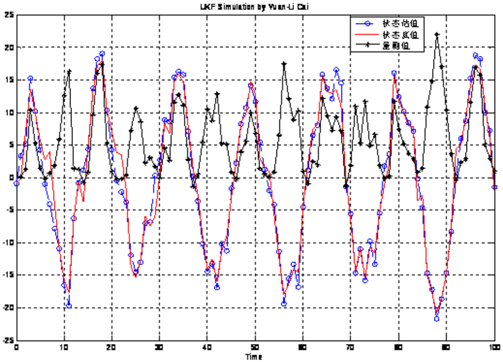

Figure 4: UKF 仿真 (RMS=2.0)

7. MATLAB code

```
%UKF algorithm 1
function [xhat] = UPF_A1(f, h, Q, R, x0, P0, y)
tf = size(y,2);
x =x0;
P = P0;

n = size(P, 2); m = size(R, 2);

for t = 1:tf
    % time update
    [sigm, wm, wc] = OnSigmaPoint(x, P); % generate sigma points

    sigmaX = feval(f, sigm, t);
    xhat_minus = sum(sigmaX.*repmat(wm,n,1));
```

```
    {\bm{P}}_minus=Q+
      sum(((sigmaX-xhat_minus)*(sigmaX-xhat_minus)').*repmat(wc,n,1));

    sigmaY = feval(h, sigmaX);
    \bm{y}_minus = sum(sigmaY.*repmat(wm,n,1));

    {\bm{P}}_{\bm{y}}y=R + sum(((sigmaY-\bm{y}_minus)*(sigmaY-\bm{y}_minus)').
    {\bm{P}}_{\bm{x}}y=sum(((sigmaX-xhat_minus)*(sigmaY-\bm{y}_minus)').*repma

    Kg = {\bm{P}}_{\bm{x}}y*inv({\bm{P}}_{\bm{y}}y);
    xhat(t) = xhat_minus + Kg*(y(t) - \bm{y}_minus);
    x = xhat(t);
    P = {\bm{P}}_minus - Kg*{\bm{P}}_{\bm{y}}y*Kg';
end
```

```
function [sigma, wm, wc] = OnSigmaPoint(x, P)
alph = 1.0;
beta = 2.0;
ka = 0;

L = size(P, 2);
lmd = (L+ka)*alph*alph - L;

PS = sqrtm((L+lmd)*P);

wm(1) = lmd/(L+lmd);
wc(1) = wm(1) + (1-alph*alph+beta);
```

```
sigma(1) = x;

for i=2:(2*L+1)
    wm(i) = 1.0/(2*(L+lmd));
    wc(i) = wm(i);
    if i <= L+1;
        sigma(i) = x + PS(:,i-1);
    else
        sigma(i) = x - PS(:,i-1-L);
    end
end;
```

```
%UKF simulation example
clear all;

P0 = 5; % Initial noise covariance
Q = 10; % Process noise covariance
R = 1; % Measurement noise covariance
tf = 100; % Final time

h = inline('(x.^2)/20');
f = inline('0.5*x + 5*x./(1+x.^2) + 8*cos(0.4*t)','x','t');

randn('state',0);
x0 =sqrtm(P0)*randn(1); % Initial state value
```

```
for t = 1:tf % Simulate the system
    if t == 1
        x(t) = feval(f,x0,0) + sqrtm(Q)*randn(1);
    else
        x(t) = feval(f,x(t-1),t-1) + sqrtm(Q)*randn(1);
    end
    y(t) = feval(h,x(t)) + sqrtm(R)*randn(1);
end

xTrue = [x0, x];
xhat = UKF_A1(f, h, Q, R, x0, P0, y);
xhat = [x0, xhat];

plot(0:tf,xhat,'bo--',0:tf, xTrue,'r', 1:tf, y, 'k*-');
```

```
xlabel('Time');
legend('状态估值','状态真值','量测值');
title('UKF Simulation by Yuan-Li Cai');
grid on;

rms = sqrt(sum(((xTrue - xhat).^2)/tf))
```

□

Questions?