Novel Simplex Unscented Transform and Filter

Wan-Chun Li, Ping Wei, and Xian-Ci Xiao

Abstract—In this paper, a new simplex unscented transform (UT) based Schmidt orthogonal algorithm and a new filter method based on this transform are proposed. This filter has less computation consumption than UKF (unscented Kalman filter), SUKF (simplex unscented Kalman filter) and EKF (extended Kalman filter). Computer simulation shows that this filter has the same performance as UKF and SUKF, and according to the analysis of the computational requirements of EKF, UKF and SUKF, this filter has preferable practicality value. Finally, the appendix shows the efficiency for this UT.

Index Terms—Extended Kalman filter, modify unscented Kalman filter (MUKF), simplex unscented transform, unscented Kalman filter.

1. Introduction

Nonlinear filtering is important in many fields, such as control, communication, and target tracking. The classical nonlinear filter is extended Kalman filter (EKF)\cite{1}, its filter equation is the same as linear Kalman filter equation by linearizing the nonlinear equation based on 1-order Taylor expansion at the predicted points. So the filter may diverge when the observability of the system is low. Sometimes the Jacobian matrix does not exist; in that case, the EKF can not be used. To address the deficiencies of EKF, Julier \cite{2} proposed a new unscented transform, which is efficient and unbiased for the mean and variance. Based on this transform, the unscented Kalman filter (UKF) was proposed, it is shown that this filter has 2-order approximate convergence for Taylor expansion, but it has a higher computational complexity than EKF. Julier proposed a spherical simplex sigma points, and a new simplex unscented Kalman filter (SUKF)\cite{3}, its filter performance equals to UKF, and it has a less computational complexity than EKF. Literatures\cite{4}-\cite{9} discuss the UKF and its application. In this paper, we propose a minimum sigma points set with the same weight, and present a new nonlinear filter named modify unscented Kalman filter (MUKF). By analyzing the computational requirements of EKF, UKF and SUKF in detail, it is shown that MUKF is of the least computational complexity, and the filter performance equals to UKF, SUKF. In appendix we show that the transform is 2-order efficient and unbiased for the mean and variance of state vector. The structure of this paper is organized as follows. Section 2 proposes a new simplex unscented transform; section 3 presents the MUKF algorithm; section 4 discusses the computational complexity; section 5 gives an example for DOA tracking by EKF, UKF, SUKF and MUKF; and section 6 is the conclusion.

2. A New Method for Simplex Unscented Transform

An effective transform can be constructed, as the appendix shows, if the following conditions could be satisfied:

a) For the unbiased covariance, the sigma points $\chi$ and the weights $\omega$ must satisfy

$$\chi \Lambda \chi^T = I$$

where $\Lambda = \text{diag}(\omega)$, and $\omega = [\omega_0, \omega_1, \cdots, \omega_L]^T$.

b) For the unbiased mean, the sigma points and the weights must satisfy

$$\chi \omega = 0.$$  \hspace{1cm} (2)

c) For normalize the weights, the weight must satisfy

$$\text{tr}(\Lambda) = \sum_{k=1}^{L} \omega_k = 1$$ \hspace{1cm} (3)

where $\text{tr}(\Lambda)$ denote the trace of matrix $\Lambda$.

As the dimension of state vector is $N$, let $\chi$ be a matrix with size $N \times L$. From (1), we know that the rank of $\chi$ is $N$, so $L \geq N$.

We first consider the case of $L = N$. Let $\omega_0 = 1/N$, (3) holds, and (2) becomes

$$\sum_{i=1}^{N} \chi_{il} = 0, \quad i = 1, 2, \cdots, N.$$  \hspace{1cm} (4)

Equation (1) is reduces to $\chi \chi^T = \mathbf{M}$, that is, $N^{0.5} \chi$ is an orthogonal matrix. Obviously, the above results are inconsistent mutually, so $L$ must be larger than $N$.

In order to construct an simplex and effective transform, we choose $L = N + 1$ in our consideration, and therefore we have $\chi \in \mathbb{R}^{N}$.\hspace{1cm} (5)

Next, we consider the calculations of weights and sigma...
points. For simplicity, we choose the equal weights 
\[ \omega_i = 1/(N+1), \]
and
\[ \omega = \frac{1}{N+1} [1, 1, \ldots, 1]^T. \] (5)

To satisfy (2), the simplest matrix \( \chi_i \) is selected as
\[ \chi_i = [1, -1] \] (6)
where \( I = [1, 1, \ldots, 1]^T \) is an \( N \) dimension vector.

From (1) and (5), we have
\[ XX^T = (N+1)I, \] (7)
so \((N+1)^{-0.5} \chi \) is an orthogonal matrix.

Let \( \chi_i = [x_i^1, x_i^2, \ldots, x_i^n]^T \), where \( x_i \) is the \( i \)th row vector of matrix \( \chi_i \), \( x_i \in \mathbb{R}^{n(N+1)} \).

Using Schmidt orthogonal transform to (6), we have
\[ \chi = \sqrt{N+1} \left[ \begin{array}{cccc} 2^{-0.5} & -2^{-0.5} & 0 & 0 \cdots & 0 \\ 6^{-0.5} & 6^{-0.5} & -(3/2)^{-0.5} & 0 \cdots & 0 \\ 12^{-0.5} & 12^{-0.5} & 12^{-0.5} & -(4/3)^{-0.5} \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \end{array} \right]. \]

3. Modifying Unscented Kalman Filter

The nonlinear discrete time system is modeled as
\[ x_k = f(x_{k-1}) + w_{k-1} \] (8a)
\[ z_k = h(x_k) + v_k \] (8b)
where \( x \) is \( n \)-dimensional state vector, \( z \) is \( m \)-dimensional observation vector, \( w \) is \( n \)-dimensional state noise vector, \( v \) is \( m \)-dimensional observation noise vector. Assume that the noise vectors \( w \) and \( v \) are zero-mean, and
\[ E(w_i^j) = 0, \quad E(w_i^j)^T = \delta_{ij}Q \]
\[ E(v_i^j) = \delta_{ij}R, \quad E(v_0^j) = \hat{x}_0 \]
\[ E[(x_0 - E(x_0))(x_0 - E(x_0))^T] = P_0 \]
where when \( k = l, \delta_{ij} = 1; \) else \( \delta_{ij} = 0. \)

At time \( k-1 \), the estimation of state vector is \( \hat{x}_{k-1} \) and state covariance is \( P_{k-1} \). Because \( P_{k-1} \) is a positive definite matrix, the Cholesky decomposition is given by
\[ P_{k-1} = A_{k-1}A_{k-1}^T \] (9)
where \( A_{k-1} \) is a positive upper triangular matrix.

Select \( n+1 \) sigma points and weights as
\[ \omega = [\omega_1, \omega_2, \ldots, \omega_{n+1}]^T \]
\[ \Lambda = \text{diag}(\omega_i), \quad i = 1, 2, \ldots, n+1 \] (10)
where \( \otimes \) represents Kronecker product.

Decompose the positive definite matrix \( P_0 \) as
\[ P_0 = A_0A_0^T. \] (11)
The initial sigma points are created by
\[ \Psi_0 = I^T \otimes \hat{x}_0 + A_0\chi. \] (12)
The transformed set \( \Psi'_k \) is given by instantiating each point through the state model:
\[ \Psi'_k = f(\Psi'_{k-1}) \] (13)
where \( \Psi'_{k-1} \) represents the \( i \)th row vector of matrix \( \Psi_{k-1} \) \( (i = 1, 2, \ldots, n+1) \).

The predicted mean \( \hat{x}_{k-1} \) is computed as
\[ \hat{x}_{k-1} = \Psi'_{k-1} \omega \] (14)
The predicted covariance \( P_{k-1} \) is given by
\[ P_{k-1} = (\Psi_{k-1} - I^T \otimes \hat{x}_{k-1})A' \left( \Psi_{k-1} - I^T \otimes \hat{x}_{k-1} \right)^T + Q_{k-1}. \] (15)

Instantiate each of the prediction points \( x_i \) through the observation model:
\[ \zeta_{k|k-1} = h(\Psi_{k|k-1}). \] (16)

The predicted observation \( \hat{z}_{k|k-1} \) is calculated by
\[
\hat{z}_{k|k-1} = \zeta_{k|k-1} = \Psi_{k|k-1} \omega. \] (17)

The cross correlation matrix \( \Psi_{k}^{xz} \) is determined by
\[
\Psi_{k}^{xz} = (\Psi_{k|k-1} - 1^T \otimes \hat{x}_{k|k-1}) \Lambda(\zeta_{k|k-1} - 1^T \otimes \hat{z}_{k|k-1})^T \] (18)

Because the observation noise is additive and independent, the innovation covariance \( \Psi_{k}^{z} \) has the form:
\[
\Psi_{k}^{z} = (\zeta_{k|k-1} - 1^T \otimes \hat{z}_{k|k-1}) \Lambda(\zeta_{k|k-1} - 1^T \otimes \hat{z}_{k|k-1})^T + R_{k} \] (19)

So the Kalman gain \( \Omega_{k} \) is
\[
\Omega_{k} = \Psi_{k}^{xz} (\Psi_{k}^{z})^{-1}. \] (20)

The updated state covariance \( P_{k} \) is computed by
\[
P_{k} = P_{k|k-1} - \Omega_{k} \Psi_{k}^{z} \Omega_{k}^T. \] (21)

The updated state mean \( \hat{x}_{k} \) is given by
\[
\hat{x}_{k} = \hat{x}_{k|k-1} + \Omega_{k}(z_{k} - \hat{z}_{k|k-1}). \] (22)

This algorithm is given in Fig. 1.

1. Initialization:
\[
P_{k|1} = A_{k|1} A_{k|1}^T \\
\Psi_{k|1} = 1^T \otimes \hat{x}_{k|1} + A_{k|1} \chi
\]

2. Prediction:
\[
\Psi_{k|k-1} = f(\Psi_{k|k-1}) \\
\hat{x}_{k|k-1} = \Psi_{k|k-1} \omega \\
P_{k|k-1} = (\Psi_{k|k-1} - 1^T \otimes \hat{x}_{k|k-1}) \Lambda(\Psi_{k|k-1} - 1^T \otimes \hat{x}_{k|k-1})^T + Q_{k-1} \\
\zeta_{k|k-1} = h(\Psi_{k|k-1}) \\
\hat{z}_{k|k-1} = \zeta_{k|k-1} \omega
\]

3. Filter:
\[
P_{k}^{xz} = (\Psi_{k|k-1} - 1^T \otimes \hat{x}_{k|k-1}) \Lambda(\zeta_{k|k-1} - 1^T \otimes \hat{z}_{k|k-1})^T \\
P_{k}^{z} = (\zeta_{k|k-1} - 1^T \otimes \hat{z}_{k|k-1}) \Lambda(\zeta_{k|k-1} - 1^T \otimes \hat{z}_{k|k-1})^T + R_{k} \Omega_{k} = (P_{k}^{xz})^{-1} \\
P_{k} = P_{k}^{xz} - \Omega_{k} P_{k}^{z} \Omega_{k}^T \\
\hat{x}_{k} = \hat{x}_{k|k-1} + \Omega_{k}(z_{k} - \hat{z}_{k|k-1})
\]

Fig. 1. Summary of MUKF algorithm.

4. Computational Requirements

In this section, we compare the computational requirements for EKF, UKF, SUKF and MUKF in the following table, \( N \) denotes the size of state vector, \( M \) denotes the size of measure vector.

<table>
<thead>
<tr>
<th>Computational requirements for EKF, UKF, SUKF and MUKF</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Additions</strong></td>
</tr>
<tr>
<td>EKF</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>UKF</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>SUKF</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>MUKF</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

The complexity of the proposed algorithm can be assessed by the table 1 and the matrix inverse and extra computational requirements. For matrix inverse, all the filters need: additions: \((N-1)(N+1)/2\); multiplications: \(N(N^2+2N+3)/2\), and the extra computational requirements include the evaluation, allot sigma points, etc. The EKF’s computational requirements are twice as MUKF, and the computational requirements of SUKF equal to EKF. So MUKF has the least computational requirements.

5. Simulation

Computer simulation is a target tracking problem with DOA, assumed that the target is constant velocity in 2-dimension plane, the measurement parameters are DOAs by three radars, their positions are (100, 0) km, (-50, 86.6) km and (-50, -86.6) km, the measurement errors are zero-mean and their covariances are 1 degree. The sampling interval time is 10 second, the target’s initial position is (100, 120) km, and its velocity is (-150, 100) m/s. Fig. 2 is averaged from 100000 independent Monte Carlo simulations. In this figure, UKF, SUKF and MUKF are overlapped sketched in solid line, which means that they have the same tracking performance.
The state and the measure equations are given by
\[
\begin{align*}
x_k &= \Phi x_{k-1} + w_{k-1} \\
\theta_k &= h(x_k) + v_k
\end{align*}
\]
where the parameters are given by
\[
\begin{align*}
\Phi &= \begin{bmatrix} I_2 & T \end{bmatrix} \\
\theta &= \begin{bmatrix} 0 \end{bmatrix} \begin{bmatrix} I \end{bmatrix}
\end{align*}
\]

From the simulation, we know that the tracking performance: the UKF, SUKF and MUKF have same performance, and all they have better performance than that of EKF. In the appendix, we will explain the performance of UKF, SUKF and MUKF.

**6. Conclusions**

We present a new simplex unscented transforms which achieve a lower computational requirements than sphere and skew unscented transforms, and propose a new Kalman filter MUKF, which has the same performance as UKF and SUKF, and is better than that of EKF, and it is the least computational requirements, especially, when the size of state vector or measurement vector are large, it will show a very efficient method for filter, and it is more suitable for nonlinear filter. In the furture, the MUKF will be used in many industry fields.

**Appendix**

In this appendix, we show that the transform has 2-order efficient and unbiased for the mean and variance.

The random vector $x$ have a mean $\bar{x}$ and covariance $Dx$; we discuss the random vector $y = f(x)$, to compute the mean $\bar{y}$ and covariance $Dy$.

Take the Taylor series expansion of $\bar{x}$ as
\[
y = f(x) + \frac{1}{2} \nabla f(x) + \frac{1}{4!} \nabla^3 f(x) + \cdots \quad (F1)
\]
where $\delta x$ is a zero mean with covariance $Dx$, $\nabla^m f(x)$ is the appropriate $m$th order term in the multidimensional Taylor series.

Take expectations, yields
\[
\bar{y} = f(\bar{x}) + \frac{1}{2} \nabla f(\bar{x})Dx + \frac{1}{3!} \nabla^3 f(\bar{x})Dx^2 + \cdots
\]
\[
Dy = \nabla f(\bar{x})Dx\nabla f(\bar{x}) + \frac{1}{2} \nabla^2 f(\bar{x})Dx^2 + \frac{1}{3!} \nabla^3 f(\bar{x})Dx^3 + \cdots
\]

Neglecting the second and higher order terms of $\delta x$, yields
\[
\bar{y} = f(\bar{x}), \quad (F2)
\]
\[
Dy = \nabla f(\bar{x})Dx\nabla f(\bar{x}), \quad (F3)
\]

From (7), we get
\[
Dx = AA^T.
\]

The sigma points are created by
\[
\Psi = I^T \otimes x + A\chi
\]

The transformed set is given by instantiating each point through the state model:
\[
\Psi_j = f(\Psi_j).
\]

Using the transform:
\[
\hat{\Psi} = \Psi_{\omega} \quad (F4)
\]
where $(A\chi)_\omega$ represents the $\omega$th arrow of matrix $A\chi$.

Take the Taylor series expansion about $\bar{x}$:
\[
\hat{\Psi} = f(\bar{x}) + \frac{1}{2} \nabla f(\bar{x}) + \frac{1}{3!} \nabla^3 f(\bar{x}) + \cdots
\]
\[
= f(\bar{x}) + \nabla f(\bar{x})\omega + \frac{1}{N+1} \sum_{k=1}^{N+1} \left\{ \frac{1}{2} \nabla^2 f((\bar{x})_k) \right\}^2 + \cdots
\]

Neglecting the second and higher order terms of $(A\chi)_\omega$ and noting $\omega = 0$, yields
\[
\hat{\Psi} = f(\bar{x}) = \bar{y}, \quad (F5)
\]

From (F3), we get
\[
\hat{\mathbf{D}}\mathbf{y} = [\Psi - 1^T \otimes (\Psi; \omega)] \Lambda [\Psi - 1^T \otimes (\Psi; \omega)]^T.
\]

For \( \Lambda = \frac{1}{N+1} \mathbf{I} \),

\[
\hat{\mathbf{D}}\mathbf{y} = \frac{1}{N+1} \sum_{k=1}^{N+1} \left\{ \mathbf{f} \left[ \mathbf{x} + (A\chi)_k \right] - \frac{1}{N+1} \sum_{k=1}^{N+1} \mathbf{f} \left[ \mathbf{x} + (A\chi)_k \right] \right\}^T
\]

\[
\times \left\{ \mathbf{f} \left[ \mathbf{x} + (A\chi)_k \right] - \frac{1}{N+1} \sum_{k=1}^{N+1} \mathbf{f} \left[ \mathbf{x} + (A\chi)_k \right] \right\}^T \right\}
\]

(F6)

and

\[
\mathbf{f} \left[ \mathbf{x} + (A\chi)_k \right] = \frac{1}{N+1} \sum_{k=1}^{N+1} \mathbf{f} \left[ \mathbf{x} + (A\chi)_k \right]
\]

\[
= \mathbf{f} \left( \mathbf{x} \right) + \nabla \mathbf{f} \left[ (A\chi)_k \right] + \frac{1}{2} \nabla^2 \mathbf{f} \left[ (A\chi)_k \right]^2 + \frac{1}{3!} \nabla^3 \mathbf{f} \left[ (A\chi)_k \right]^3 + \cdots
\]

\[
- \frac{1}{N+1} \sum_{k=1}^{N+1} \left\{ \mathbf{f} \left( \mathbf{x} \right) + \nabla \mathbf{f} \left[ (A\chi)_k \right] \right\}
\]

\[
+ \frac{1}{2} \nabla^2 \mathbf{f} \left[ (A\chi)_k \right]^2 + \frac{1}{3!} \nabla^3 \mathbf{f} \left[ (A\chi)_k \right]^3 + \cdots
\]

\[
= \nabla \mathbf{f} \left( A\chi_k \right) + \frac{1}{2} \nabla^2 \mathbf{f} \left[ (A\chi)_k \right]^2 + \frac{1}{3!} \nabla^3 \mathbf{f} \left[ (A\chi)_k \right]^3 + \cdots
\]

\[
- \frac{1}{N+1} \sum_{k=1}^{N+1} \left[ \frac{1}{2} \nabla^2 \mathbf{f} \left( (A\chi)_k \right)^2 + \frac{1}{3!} \nabla^3 \mathbf{f} \left( (A\chi)_k \right)^3 + \cdots \right]
\]

(F7)

Substituting (F6) into (F7) and neglecting the second and higher order terms of \( (A\chi)_k \),

\[
\hat{\mathbf{D}}\mathbf{y} = \frac{1}{N+1} \sum_{k=1}^{N+1} \left[ \nabla \mathbf{f} \left( A\chi_k \right) \right] \left[ \nabla \mathbf{f} \left( A\chi_k \right) \right]^T
\]

\[
= \nabla \mathbf{f} A\chi A\chi^T \Lambda^T \nabla \mathbf{f}^T
\]

Noting \( \chi A\chi^T = \mathbf{I} \) and \( \mathbf{A} \mathbf{A}^T = \mathbf{D} \), yields:

\[
\hat{\mathbf{D}}\mathbf{y} = \nabla \mathbf{f} \mathbf{D} \mathbf{x} \nabla \mathbf{f}^T = \mathbf{D} \mathbf{y}
\]

So this transform estimator is 2-order efficient and unbiased for the mean and variance. \( \text{QED} \)

References


Wan-Chun Li was born in Sichuan Province, China, in 1978. He received the B.S. and M.S. degrees from University of Electronic Science and Technology of China (UESTC), Chengdu, in 1999 and 2002, respectively. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, UESTC. His research interests include spectral estimation, passive location, and information theory.