SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

April 2017, Vol. 60 042201:1–042201:11
doi: 10.1007/s11432-015-0189-9

Iterative spherical simplex unscented particle filter for CNS/Redshift integrated navigation system

Kui FU[1], Guangqiong ZHAO[2], Xiajing LI[3], Zhong-Liang TANG[1] & Wei HE[4]*

[1]*Center for Robotics, University of Electronic Science and Technology of China, Chengdu* 611731*, China;*
[2]*School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu* 611731*, China;*
[3]*School of Automation, University of Electronic Science and Technology of China, Chengdu* 611731*, China;*
[4]*School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing* 100083*, China*

Received April 19, 2016; accepted May 7, 2016; published online March 14, 2017

Abstract We propose an improved Unscented Particle Filter (UPF) algorithm for the Celestial Navigation System/Redshift (CNS/Redshift) integrated navigation system. The algorithm adopts the iterated spherical simplex unscented transformation rather than the traditional unscented transformation. The navigation performance of the proposed algorithm is assessed by several indexes. Simulation results show that the proposed UPF algorithm has advantages over the traditional UPF algorithm in terms of computation burden, navigation accuracy, and numerical stability.

Keywords CNS/Redshift, navigation system, UPF, spherical simplex, iteration

Citation Fu K, Zhao G Q, Li X J, et al. Iterative spherical simplex unscented particle filter for CNS/Redshift integrated navigation system. Sci China Inf Sci, 2017, 60(4): 042201, doi: 10.1007/s11432-015-0189-9

1 Introduction

With increasing aerospace technology and human interest in the universe, deep space exploration is an active aerospace field. Deep space exploration is becoming a new hot spot in international space activities. The basic generic technologies of deep space exploration include deep space navigation, deep space communication, power propulsion, etc. The autonomous navigation problem is one of the most basic technologies and of great importance.

At present, the main navigation methods include ground measurement and control, satellite navigation, inertial navigation and celestial navigation. In general, a deep space probe relies on radio measurements and control by a ground station for navigation, but a ground station is vulnerable during wartime, restricted by geographical conditions, low angle measurement accuracy, and prolonged time delays. All of these problems threaten the safety and reliability of a deep space probe. Conventional navigation methods have limitations that require the introduction of spacecraft orbital dynamics, which lead to a variety of complex computations. A Celestial Navigation System (CNS) can realize autonomous navigation by celestial body sensors, i.e., without any other hardware equipment [1–3]; these sensors can only obtain position and angle information between celestial bodies or between a celestial body and the probe and

* Corresponding author (email: hewei.ac@gmail.com)

cannot obtain distance and velocity information. Furthermore, limited to measurement accuracy, a CNS cannot achieve high navigation accuracy. Conversely, Redshift can calculate the relative velocity between a probe and a permanent position, but it cannot obtain all state estimations; as a result, it cannot be separately applied to autonomous navigation. The CNS/Redshift integrated navigation system proposed in this paper utilizes filter information fusion technology, merging CNS angle information, and Redshift velocity information to effectively improve the precision of a navigation system. moreover it implements probe accurate positioning in the whole segment.

Filtering methods play a significant role in navigation calculations. They can compensate for some existing hardware problems to some extent, such as fixed and random errors. For nonlinear problems, the most conventional solution to this problem is the extended Kalman filter (EKF) [4], which is a first-order linear truncation to the Taylor series expansion of a nonlinear function that ignores the remaining high-order items; thus, applying Kalman filter to a nonlinear system converts a nonlinear problem to a linear one. However this has two defects [5]. First, when the system has strong nonlinearity, filter performance is poor, which may lead to divergence, and second, it requires calculating the Jacobian matrix [6]. To address these problems, the Unscented Kalman filter (UKF) [7] has been proposed, which is based on the Unscented Transformation (UT) [8, 9]. This filter produces a more accurate result and is easier to implement compared with the EKF. Nevertheless, its limitations appear when it is applied to a general non-Gaussian distribution. Particle filters (PFs) [10,11] are known as a sequential Monte-Carlo methods, they utilize a large number of random samples to represent the probability density function(pdf) of a system and have greater ability to handle nonlinear/non-Gaussian systems. While particle degeneracy is a common problem in PFs, a modification that combines the PF the UKF algorithms is the Unscented Particle filter (UPF) [12], which- has been proposed to overcome this drawback. However, its performance has been proved satisfactory at the price of a significant increase in the computational cost. Furthermore, the UPF algorithm is a third-order approximation, which has high computational burden. In this paper, we propose the improved UPF algorithm. The conventional UPF is based on the Unscented Transformation (UT)- to handle nonlinear transformations of the mean and covariance. Since the number of Sigma points selected in the UT determines the computational burden, decreasing the required points becomes crucial. The Sphere Simplex unscented transform (SSUT) [13] was proposed by Julier in 2003. It uses $n + 2$ hypersphere distribution Sigma points, rather than $2n + 1$ symmetric distribution Sigma points in the conventional UT, and can greatly reduce the computational burden on the premise of guaranteed navigation precision. In terms of filtering performance, adaptive methods have been adopted [14–18] for improvement, such as [19] achieved with an adaptive neural network. All the filter mentioned above are single iterate approximations. Disregarding higher-order terms implies the introduction of a truncated error. Iterative methods incorporate the latest measurements to outperform the estimation. The iterated UKF (IUKF) achieves a maximum a posteriori estimate when it is applied as the generating proposal distribution in the particle filter framework; in particulr, the approximated posterior density function exhibits more precisely. We combine the SSUT and IUKF to obtain the Iterative Spherical Simplex Unscented Particle filter (ISSUPF) algorithm, which can achieve better navigation performance with lower computational burden.

The remainder of this paper is arranged as follows. In Section 2, we introduce the orbit dynamics model of probe motion. In Section 3, the CNS/Redshift integrated navigation system is proposed. Next in Section 4, we describe the ISSUPF in detail. Section 5 describes our simulation. Then Section 6, we give some system assessments. Finally, in Section 7, conclusion is drawn.

2 Orbit dynamics model

The Mars probe orbit dynamic model can be considered as a circular restricted four segments model, which must consider the impact of the center of gravity of the sun, Mars, and Earth on the probe and the gravitational force of other celestial bodies as perturbing factors. Assuming that Mars and Earth orbit the sun in uniform circular motion, the radius is the average distance between the sun and Mars r_{sm} and

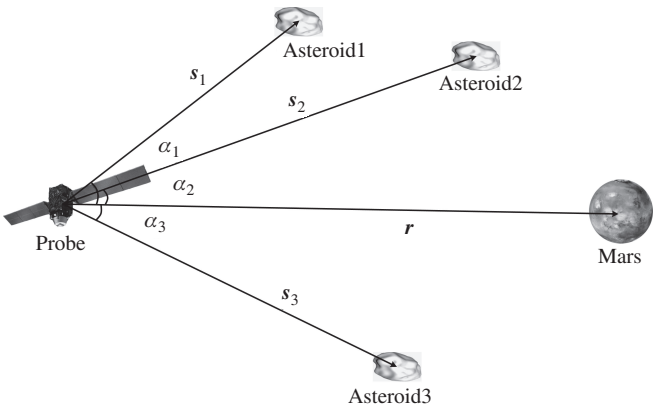

Figure 1 Measurement model.

an astronomical unit (AU); the motion can be expressed as follows [20]:

$$\ddot{\boldsymbol{r}}_{ps} = -\mu_s \frac{\boldsymbol{r}_{ps}}{r_{ps}^3} - \mu_m \left[\frac{\boldsymbol{r}_{pm}}{r_{pm}^3} - \frac{\boldsymbol{r}_{sm}}{r_{sm}^3} \right] - \mu_e \left[\frac{\boldsymbol{r}_{pe}}{r_{pe}^3} - \frac{\boldsymbol{r}_{se}}{r_{se}^3} \right], \tag{1}$$

where μ_s, μ_m, and μ_e refer to heliocentric, center of Mars and the geocentric gravitation constant, respectively; $\boldsymbol{r}_{ps}$ is the vector form of heliocentric to the probe, $\boldsymbol{r}_{pm}$ is the vector from Mars to the probe, $\boldsymbol{r}_{sm}$ is the vector from center of Mars to heliocentric, $\boldsymbol{r}_{pe}$ is the vector from Earth to the probe, and $\boldsymbol{r}_{se}$ is the vector from geocentric to heliocentric.

3 CNS/Redshift integrated navigation system

In this section, we review the CNS and Redshift navigation systems. Then we propose a CNS/Redshift navigation system.

3.1 Celestial navigation system

In a deep space mission to Mars, the probe will enter Low Earth Orbit (LEO) after launch; after that, it enters a transfer orbit between Earth and Mars and then flies to Mars. The probe in transfer orbit has not been out of the Earth's gravity and is mainly affected by the gravity of Mars, Earth, and the sun; other celestial bodies have little impact on it, so the problem can be regarded as circular restricted four segments model as discussed in Section 2. The main obstacle of automation celestial navigation is the problems of the nonlinearity of the system and the non-Gaussian distribution of noise. We use a four segments orbit dynamic model as given by (1), choose state vector $\boldsymbol{X} = [x, y, z, v_x, x_y, v_z]^{\mathrm{T}}$, and chose state model noise $\boldsymbol{W} = [w_x, w_y, w_z, w_{v_x}, w_{v_y}, w_{v_z}]^{\mathrm{T}}$. Then, the state equation can be written as

$$\dot{\boldsymbol{X}}(t) = f(\boldsymbol{X}, t) + \boldsymbol{W}(t). \tag{2}$$

In an autonomous navigation system, we usually use starlight angular distance as the measurement value. The measurement model is shown in Figure 1.

Here, α_1, α_2, and α_3 denote three starlight angular measurements, which indicate the angles between the vector from the probe to Mars and the vector from the probe to asteroids. According to the geometrical relationship in Figure 1, we obtain the following expressions for the starlight angular distance:

$$\alpha_1 = \arccos\left(\frac{\boldsymbol{r} \cdot \boldsymbol{s}_1}{r}\right) + v_{\alpha_1}, \tag{3}$$

$$\alpha_2 = \arccos\left(\frac{\boldsymbol{r} \cdot \boldsymbol{s}_2}{r}\right) + v_{\alpha_2}, \tag{4}$$

$$\alpha_3 = \arccos\left(\frac{\boldsymbol{r} \cdot \boldsymbol{s}_3}{r}\right) + v_{\alpha_3}, \tag{5}$$

where v_{α_1}, v_{α_2}, and v_{α_3} represent noise measurements.

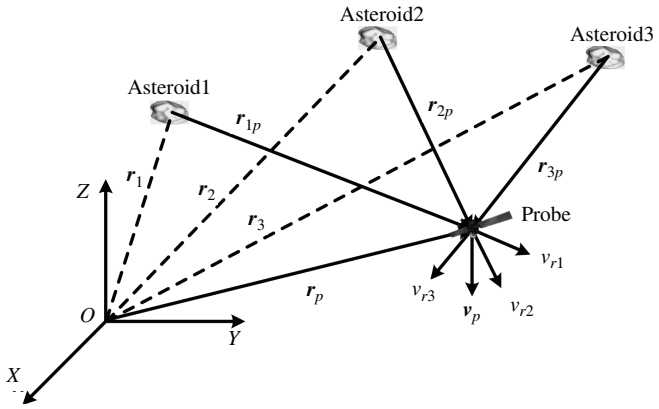

Figure 2 The operating principle of Redshift navigation.

3.2 Redshift navigation system

When a wave source moves toward or away from the receiver, the frequency of the wave measured by the receiver becomes bigger or smaller, which is called the Doppler effect. On the basis of the Doppler effect, the probe Redshift autonomous navigation method utilizes the light signal of celestial bodies in the solar system as a navigation information source and combines this information with ephemeris information of celestial bodies and probe attitude information. According to the Redshift effect measurement, we obtain the velocity of the probe in the inertial coordinate system, and by integration, we further obtain the position information of the probe [2,21]. In the probe automation navigation system, we select three asteroids as light sources. When relative motion happens between the probe and a light source, the frequency measured on the probe will have some change compared with that measured on the Earth. The relationship between the two frequencies and speeds can be described as follows [22]:

$$f_m = f_0 \frac{\sqrt{1-|\boldsymbol{v}|^2/c^2}}{1+|\boldsymbol{v}|\cos\theta/c}, \tag{6}$$

where f_m is the frequency measured on the probe, f_0 is the frequency measured on Earth, $\boldsymbol{v}$ represents the velocity vector of the probe relative to the light source, θ is the angle between the light source vector and the velocity vector, and c is the velocity of light in a vacuum. The operating principle of Redshift navigation [23] is shown in Figure 2.

We use three non-collinear asteroids as light sources. Another formation of (6) can be described by

$$\boldsymbol{v}_r = \frac{f_o}{f_m}\sqrt{c^2-|\boldsymbol{v}_p|^2} - c. \tag{7}$$

Together with Figure 2, Eq. (7) becomes (8);

$$\begin{cases} v_{r1} = \frac{f_{01}}{f_{m1}}\sqrt{c^2-|\boldsymbol{v}_p|^2} - c, \\ v_{r2} = \frac{f_{02}}{f_{m2}}\sqrt{c^2-|\boldsymbol{v}_p|^2} - c, \\ v_{r3} = \frac{f_{03}}{f_{m3}}\sqrt{c^2-|\boldsymbol{v}_p|^2} - c. \end{cases} \tag{8}$$

The geometric relation can be described by (9):

$$\begin{cases} v_{r1} = (\boldsymbol{v}_p - \boldsymbol{v}_1)\cdot\boldsymbol{u}_1, \\ v_{r2} = (\boldsymbol{v}_p - \boldsymbol{v}_2)\cdot\boldsymbol{u}_2, \\ v_{r3} = (\boldsymbol{v}_p - \boldsymbol{v}_3)\cdot\boldsymbol{u}_3, \end{cases} \tag{9}$$

where v_{r1}, v_{r2}, and v_{r3} denote the velocity vectors of the navigation celestial bodies in the inertial frame, and $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, and $\boldsymbol{v}_3$ are the unit vectors of the probe position in the inertial frame. From above, we obtain

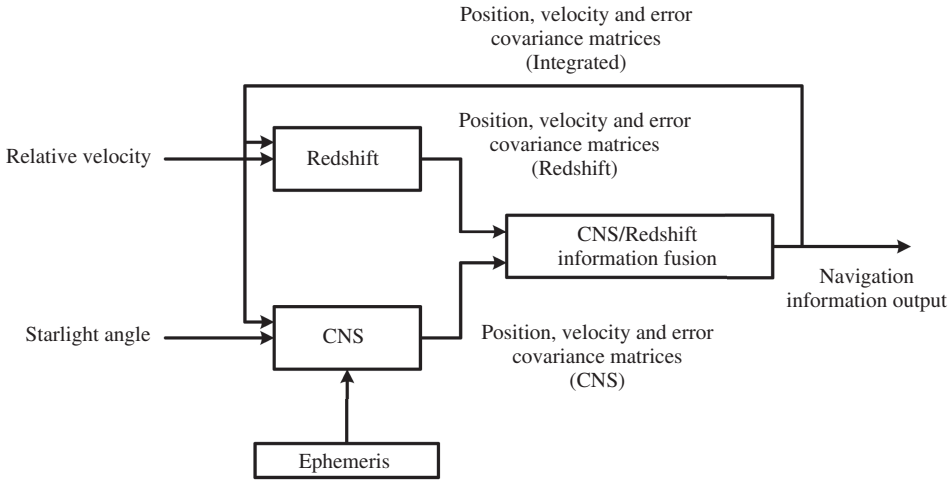

Figure 3 CNS/Redshift integrated navigation system.

the state estimation equations of the position and velocity as follows:

$$\begin{cases} \boldsymbol{v}_p = f(\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3, v_{r1}, v_{r2}, v_{r3}), \\ \boldsymbol{r}_p = \int \boldsymbol{v}_p \mathrm{d}t. \end{cases} \tag{10}$$

By solving these equations, we obtain information about $\boldsymbol{v}_p$ and $\boldsymbol{r}_p$.

3.3 Integration navigation system

The CNS/Redshift integrated navigation system utilizes information fusion methods fusing celestial information and Redshift information, reducing the complexity of equipment, greatly improving navigation accuracy, and realizing accurate positioning in the whole segment. Its basic principle diagram is depicted in Figure 3.

In this integrated navigation system, we use (2) as the state equation. Furthermore, we assume that $\boldsymbol{Z} = [\alpha_1, \alpha_2, \alpha_3, v_{r1}, v_{r2}, v_{r3}]$; in this case, the measurement model can be expressed as

$$\boldsymbol{Z}(t) = \boldsymbol{H}(\boldsymbol{X}(t), t) + \boldsymbol{V}(t), \tag{11}$$

where $\boldsymbol{V}$ is the noise measurement.

4 Iterative spherical simplex unscented particle filter

4.1 Traditional unscented particle filter

The traditional UPF is consists of importance sampling and resampling. In importance sampling, the UPF adopts the UT to obtain the proposal distribution of each particle. The basic UKF algorithm can be described as follows.

1. Initialize:

$$\hat{x}_0 = E[x_0], \quad P_0 = \left[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^{\mathrm{T}}\right]. \tag{12}$$

2. Calculate Sigma points:

$$\chi_{k-1} = \left[\hat{x}_{k-1}\ \hat{x}_{k-1} + \eta\sqrt{P_{k-1}}\ \hat{x}_{k-1} - \eta\sqrt{P_{k-1}}\right]. \tag{13}$$

3. Time update:

$$\chi_{i,k|k-1} = f(\chi_{k-1}), \tag{14}$$

$$\hat{x}_k^- = \sum_{i=0}^{2L} W_i^m \chi_{i,k|k-1}, \tag{15}$$

$$P_k^- = \sum_{i=0}^{2L} W_i^m [\chi_{i,k|k-1} - \hat{x}_i^-][\chi_{i,k|k-1} - \hat{x}_i^-]^{\mathrm{T}} + Q_k. \tag{16}$$

4. Measurement update:

$$\gamma_{i,k|k-1} = h(\chi_{i,k|k-1}), \tag{17}$$

$$\hat{y}_k^- = \sum_{i=0}^{2L} W_i^m \gamma_{i,k|k-1}, \tag{18}$$

$$P_{y_k y_k} = \sum_{i=0}^{2L} W_i^c (\gamma_{i,k|k-1} - \hat{y}_k^-)(\gamma_{i,k|k-1} - \hat{y}_k^-)^{\mathrm{T}} + R_k, \tag{19}$$

$$P_{x_k y_k} = \sum_{i=0}^{2L} W_i^c (\chi_{i,k|k-1} - \hat{x}_k^-)(\gamma_{i,k|k-1} - \hat{y}_k^-)^{\mathrm{T}}, \tag{20}$$

$$\kappa_k = P_{x_k y_k} P_{y_k y_k}^{-1}, \tag{21}$$

$$\hat{x}_k = \hat{x}_k^- + \kappa_k (y_k - \hat{y}_k^-), \tag{22}$$

$$P_k = P_k^- - \kappa_k P_{y_k y_k} \kappa_k^{\mathrm{T}}. \tag{23}$$

The basic UPF algorithm can be described as follows.

1. Initialization, $t = 0$:

   For $i = 1, \ldots, N$, draw particle $x_0^i \sim p(x_0)$ and set $t = 1$.

2. Importance sampling step:

   For $i = 1, \ldots, N$, use UKF to generate the importance distribution $N(\overline{x}_t^i, P_t^i)$ from particle $\overline{x}_t^i$. Sample $x_t^i \sim N(\overline{x}_t^i, P_t^i)$.

3. Importance weight step:

   For $i = 1, \ldots, N$, evaluate the importance weights $\tilde{\omega}_t^i = \frac{p(y_t x_t^i) p(x_t^i | x_{t-1}^i)}{N(\overline{x}_t^i, P_t^i)}$;

   For $i = 1, \ldots, N$, normalize the importance weights $\omega_t^i = \frac{\tilde{\omega}_t^i}{\sum_{j=1}^N \tilde{\omega}_t^i}$.

4. Resampling step:

   Resample N particles $\tilde{x}_{1:t}^i$ from $x_{1:t}^i$ according to the normalized importance weights. Set $\omega_t^i = \frac{1}{N}$.

4.2 Improved unscented particle filter

The traditional UPF algorithm uses UKF to obtain the importance distribution and combines the latest observations; it is a third-order approximation of the state. Its computational burden largely depend on the number of Sigma points. The standred UT, which adopts Sigma symmetric sampling, requires $2n+1$ Sigma points. The computational burden increases over the vector dimension. The ISSUPF uses the Iterative Spherical Simplex Unscented Kalman Filter (ISSUKF) to obtain the importance distribution and uses $2n+1$ Sigma points that are noncentrosymmetrical, which can greatly reduce the computational burden. In ISSUKF, sampling spherical simplex points are required. We assume that $\boldsymbol{\chi}_i^j$ is the ith Sigma point in the set for the jth dimensional space, without loss of generality, and that $\overline{x} = 0$ and $P_{xx} = I$. We first consider a single dimension, where the three points are $\boldsymbol{\chi}_0^1 = [0]$, $\boldsymbol{\chi}_1^1 = [-x_1]$, and $\boldsymbol{\chi}_2^1 = [x_2]$. The corresponding weights are W_0, W_1 and W_2, respectively. According to these conditions, we obtain the following equations:

$$W_0 + W_1 + W_2 = 1, \tag{24}$$

$$-W_1x_1 + W_2x_2 = 0, \tag{25}$$

$$W_1x_1^2 + W_2x_2^2 = 1. \tag{26}$$

Under the assumption that these simplex points are the same distance from the origin, it is very appropriate to add the constraint that $x_1 = x_2$. By solving the above equations, it is easy to conclude that $W_1 = W_2 = (1 - W_0)/2$, $x_1 = x_2 = 1/\sqrt{W_1}$, and W_0 is a free parameter. Extending this to two dimensions, a new point $(0, s_3x_3)$ with weight W_3 is introduced with scaling factor s_3. Then, we obtain the following:

$$W_0 + 2W_1 + W_3 = 1, \tag{27}$$

$$-2W_1x_1 + W_3x_3 = 0, \tag{28}$$

$$2W_1x_1^2 + W_3s_3^2x_3^2 = 1. \tag{29}$$

Each point is the same distance from the origin, so $W_1 = W_3$. By solving the above equations, we obtain $s_3 = 2$, $x_1^2 = 1/2W_1$ and $x_3^2 = 1/6W_1$. The point selection algorithm for the SSUT can be described as follows:

1. Choose $0 < W_0 < 1$.
2. Choose weight sequence
$$W_i = (1 - W_0)/(n + 1).$$
3. Initialize vector sequence
$$\boldsymbol{\chi}_0^1 = [0], \quad \boldsymbol{\chi}_1^1 = \left[-\frac{1}{\sqrt{2W_1}}\right], \quad \boldsymbol{\chi}_2^1 = \left[-\frac{1}{\sqrt{2W_1}}\right].$$
4. Expand vector sequence for $j = 2, \ldots, n$ according to
$$\boldsymbol{\chi}_i^j = \begin{cases} \begin{bmatrix} \boldsymbol{\chi}_0^{j-1} \\ 0 \end{bmatrix}, & \text{for } i = 0, \\ \begin{bmatrix} \boldsymbol{\chi}_i^{j-1} \\ -\frac{1}{\sqrt{j(j+1)W_1}} \end{bmatrix}, & \text{for } i = 1, \ldots, j, \\ \begin{bmatrix} 0^{j-1} \\ \frac{j}{\sqrt{j(j+1)W_1}} \end{bmatrix}, & \text{for } i = j + 1. \end{cases} \tag{30}$$

This algorithm demonstrates two main issues: the weight for each Sigma point is the same and is relative to $(1 - W_0)/(n + 1)$, all Sigma points lie on the hypersphere of radius $\sqrt{n}/(1 - W_0)$. According to the SSUT, we obtain the ISSUKF algorithm as follows.

1. Initialize:
$$\hat{x}_0 = E[x_0], \quad P_0 = E\left[(x_0 - \hat{x}_0)(x_0 - \hat{x}_0)^{\mathrm{T}}\right]. \tag{31}$$
2. Calculate Sigma points using (30).
3. Time update:
$$\boldsymbol{\chi}_{i,k|k-1} = f(\boldsymbol{\chi}_{k-1}), \tag{32}$$
$$\hat{x}_k^- = \sum_{i=0}^{L+1} W_i^m \boldsymbol{\chi}_{i,k|k-1}, \tag{33}$$
$$P_k^- = \sum_{i=0}^{L+1} W_i^m [\boldsymbol{\chi}_{i,k|k-1} - \hat{x}_k^-][\boldsymbol{\chi}_{i,k|k-1} - \hat{x}_k^-]^{\mathrm{T}} + Q_k. \tag{34}$$

4. Measurement update:

$$\boldsymbol{\gamma}_{i,k|k-1} = h(\boldsymbol{\chi}_{\boldsymbol{k}|\boldsymbol{k-1}}), \tag{35}$$

$$\hat{y}_k^- = \sum_{i=0}^{L+1} W_i^m \boldsymbol{\gamma}_{i,k|k-1}, \tag{36}$$

$$P_{y_k y_k} = \sum_{i=0}^{L+1} W_i^c (\boldsymbol{\gamma}_{i,k|k-1} - \hat{y}_k^-)(\boldsymbol{\gamma}_{i,k|k-1} - \hat{y}_k^-)^{\mathrm{T}} + R_k, \tag{37}$$

$$P_{x_k y_k} = \sum_{i=0}^{L+1} W_i^c (\boldsymbol{\chi}_{i,k|k-1} - \hat{x}_k^-)(\boldsymbol{\gamma}_{i,k|k-1} - \hat{y}_k^-)^{\mathrm{T}}, \tag{38}$$

$$\boldsymbol{\kappa}_k = P_{x_k y_k} P_{y_k y_k}^{-1}, \tag{39}$$

$$\hat{x}_k = \hat{x}_k^- + \boldsymbol{\kappa}_k (y_k - \hat{y}_k^-), \tag{40}$$

$$P_k = P_k^- - \boldsymbol{\kappa}_k P_{y_k y_k} \boldsymbol{\kappa}_k^{\mathrm{T}}. \tag{41}$$

5. Iteration: Let $\hat{x}_{k,0} = \hat{x}_k^-$, $P_{k,0} = p_k^-$ and $\hat{x}_{k,1} = \hat{x}_k$, $P_{k,1} = P_k$. Furthermore, let $g = 1$ and $j = 2$. Then, generate new Sigma points $bm\chi_j$ by (30). Recalculate (33)–(41):

$$\hat{x}_{k,j}^- = \sum_{i=0}^{2L} W_i^m \boldsymbol{\chi}_{i,j}, \tag{42}$$

$$\boldsymbol{\gamma}_j = h(\boldsymbol{\chi}_j), \tag{43}$$

$$\hat{y}_{k,j}^- = \sum_{i=0}^{2L} W_j^m \boldsymbol{\gamma}_{i,j}, \tag{44}$$

$$P_{y_k y_k j} = \sum_{i=0}^{2L} W_i^c [\boldsymbol{\gamma}_{i,j} - \hat{y}_{k,j}^-][\boldsymbol{\gamma}_{i,j} - \hat{y}_{k,j}^-]^{\mathrm{T}} + R_k, \tag{45}$$

$$P_{x_k y_k j} = \sum_{i=0}^{2L} W_i^c [\boldsymbol{\chi}_{i,j} - \hat{x}_{k,j}^-][\boldsymbol{\gamma}_{i,j} - \hat{y}_{k,j}^-]^{\mathrm{T}}, \tag{46}$$

$$\boldsymbol{\kappa}_{k,j} = P_{x_k y_k j} P_{y_k y_k j}^{-1}, \tag{47}$$

$$\hat{x}_{k,j} = \hat{x}_{k,j} + g \cdot \boldsymbol{\kappa}_{k,j} (y_k - \hat{y}_{k,j}), \tag{48}$$

$$P_{k,j} = P_{k,j-1} - \boldsymbol{\kappa}_{k,j} P_{y_k y_k j} \boldsymbol{\kappa}_{k,j}^{-1}, \tag{49}$$

where the subscript j denotes the jth iterate; $\boldsymbol{\gamma}_{i,j}$ denotes the ith component of $\boldsymbol{\gamma}_j$. We define $\hat{y}_{k,j} = h(\hat{x}_{k,j})$, $\tilde{x}_{k,j} = \hat{x}_{k,j} - \hat{x}_{k,j-1}$, and $\tilde{y}_{k,j} = y_k - \hat{y}_{k,j}$. If the inequality (50) is satisfied

$$\tilde{x}_{k,j}^{\mathrm{T}} P_{k,j-1}^{-1} \tilde{x}_{k,j} + \tilde{y}_{k,j}^{\mathrm{T}} R_k^{-1} \tilde{y}_{k,j} < \tilde{y}_{k,j-1}^{\mathrm{T}} R_k^{-1} \tilde{y}_{k,j-1}, \tag{50}$$

and $j \geqslant N$. Then, update $g = \eta \cdot g$, $j = j + 1$ and continue to the next iteration; otherwise, set $\hat{x}_k = \hat{x}_{k,j}$ and $P_k = P_{k,j}$, and terminate the iteration.

We adopt the ISSUKF algorithm as mentioned above rather than standard UKF to generate importance distribution in standard UPF, then we can obtain ISSUPF algorithm.

5 Simulation

To illustrate the performance of ISSUPF, we consider the Redshift navigation system discussed in Section 3. The initial simulation conditions are shown in Table 1.

In Figure 4, note the flightpaths of Mars, Earth and the probe, and the forecast paths of UPF, SSUPF, IUPF, and ISSUPF, which are also shown in this figure.

From Figures 5–7, we can compare the position error of the UPF, SSUPF, IUPF, and ISSUPF in x, y, and z directions.

Table 1 The initial parameters

Total flight time		20332680 s
Total sampling points		338878
Sample interval time		60 s
Initial state	Position	$(0.7061, 1.2307, 0.5333) \times 10^{11}$ m
	Velocity	$(-2.8251, 1.4514, 0.6232) \times 10^{4}$ m/s
Model error		diag([3 3 3 3×10^{-3} 3×10^{-3} 3×10^{-3}])
Measurement error		diag([6×10^{2} 6×10^{2} 6×10^{2} 1×10^{2} 1×10^{2} 1×10^{2}])

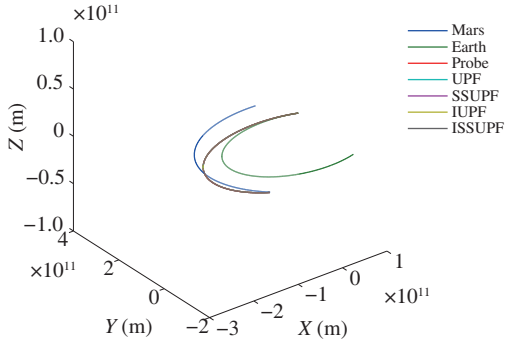

Figure 4 The total flightpath.

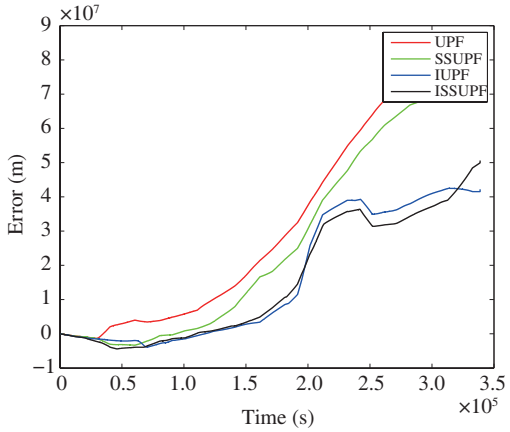

Figure 5 The position error of x direction.

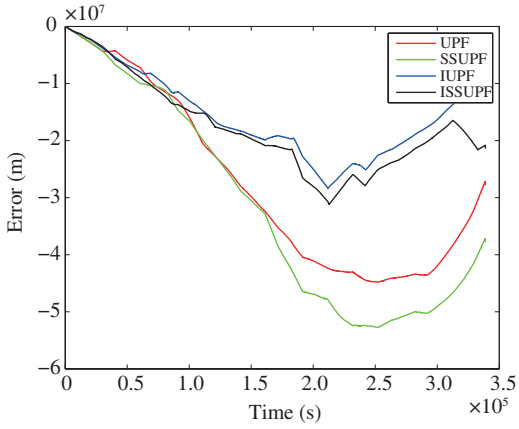

Figure 6 The position error of y direction.

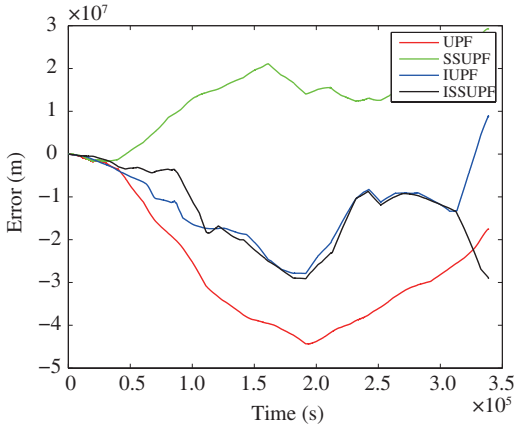

Figure 7 The position error of z direction.

Table 2 System assessment

	Runtime	UPF	SSUPF	IUPF	ISSUPF
		1.1331×10^4	8.0038×10^3	2.0864×10^4	1.7643×10^4
Position	RMS	6.1300×10^7	5.4563×10^7	3.0470×10^7	3.2191×10^7
	Available point	141627	177982	336435	313770
	Availability	0.41793	0.52521	0.99279	0.92591
	Continuity point	315959	315918	315917	315903
	Continuity	0.93237	0.93225	0.93225	0.93221
Velocity	RMS	4.4231	4.1242	5.7946	5.3550
	Available point	274559	267418	246301	239445
	Availability	0.8102	0.78913	0.72681	0.70658
	Continuity point	164361	178593	338371	338429
	Continuity	0.48502	0.52701	0.99851	0.99868

6 System assessment

We evaluate navigation performance using the root mean square (RMS):

$$\mathrm{RMS} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{1}{m_i}\sum_{j=1}^{m_i}\triangle_{ij}^2}, \tag{51}$$

where n is the quantity of available experiment data, m_i is the quantity of sampling points in one experiment, and $\triangle_{ij}$ is the error at these sampling points. We evaluate the service availability of the navigation system using Service Availability Thresholds (SATs):

$$\sqrt{(x_k - \hat{x}_k)^2 + (y_k - \hat{y}_k)^2 + (z_k - \hat{z}_k)^2} < \mathrm{SAT}_p. \tag{52}$$

If position point $(\hat{x}_k, \hat{y}_k, \hat{z}_k)$ satisfies (52), then the position point

$$\sqrt{(v_{xk} - \hat{v}_{xk})^2 + (v_{yk} - \hat{v}_{yk})^2 + (v_{zk} - \hat{v}_{zk})^2} < \mathrm{SAT}_v \tag{53}$$

is valid. If the velocity point $(\hat{v}_{xk}, \hat{v}_{yk}, \hat{v}_{zk})$ satisfies (53), then the velocity point is valid. We use the Accuracy Limit (AL) to evaluate the service availability of navigation the system; specifically,

$$\sqrt{(\hat{x}_k - \hat{x}_{k-1})^2 + (\hat{y}_k - \hat{y}_{k-1})^2 + (\hat{z}_k - \hat{z}_{k-1})^2} < \mathrm{AL}_p. \tag{54}$$

If position point $(\hat{x}_k, \hat{y}_k, \hat{z}_k)$ satisfies (54), then the navigation data is continuous at this point, i.e.,

$$\sqrt{(\hat{v}_{xk} - \hat{v}_{x(k-1)})^2 + (\hat{v}_{yk} - \hat{v}_{y(k-1)})^2 + (\hat{v}_{zk} - \hat{v}_{z(k-1)})^2} < \mathrm{AL}_v. \tag{55}$$

If velocity point $(\hat{v}_{xk}, \hat{v}_{yk}, \hat{v}_{zk})$ satisfies (55), then the navigation data is continuous at this point. We set $\mathrm{SAT}_p = 4.9 \times 10^7$, $\mathrm{SAT}_v = 7.5$, $\mathrm{AL}_p = 1.86 \times 10^6$, and $\mathrm{AL}_v = 0.48$, this yields the assessment data in Table 2.

7 Conclusion

In this paper, we presented a feasible autonomous navigation method, namely, the CNS/Redshift integrated navigation, for deep space mission in the solar system. An improved UPF algorithm was proposed by combining the iterative strategy and spherical single algorithm. Using the iterative strategy, we achieved higher navigation accuracy, at the same time, spherical simplex algorithm reduce the computational burden. Results show that UPF and SSUPF have equal position accuracy, and IUPF has the same position accuracy. Furthermore, we showed that the iterative strategy can have a positive role in

improving the position accuracy. Simulation results also showed that the ISSUPF has better position availability compared with the UPF and SSUPF and lower computational burden when compared with the IUPF. Moreover, the ISSUPF and IUPF have better velocity continuity compared with the UPF and SSUPF. Overall, the proposed ISSUPF algorithm can be successfully applied in integrated navigation systems for deep space missions, and can achieve better position navigation performance.

Acknowledgements This work was supported by National Basic Research Program of China (973 Program) (Grant No. 2014CB744206). The authors would like to thank the anonymous reviewers for their constructive comments in improving the quality and presentation of this paper.

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Ning X L, Fang J C. An autonomous celestial navigation method for {LEO} satellite based on unscented Kalman filter and information fusion. Aerosp Sci Technol, 2007, 11: 222–228
2 Ning X L, Fang J C. Spacecraft autonomous navigation using unscented particle filter-based celestial/Doppler information fusion. Meas Sci Technol, 2008, 19: 095203
3 Ning X L, Fang J C. A new autonomous celestial navigation method for the lunar rover. Robot Auton Syst, 2009, 57: 48–54
4 Zhuang Y, Gu M W, Wang W, et al. Multi-robot cooperative localization based on autonomous motion state estimation and laser data interaction. Sci China Inf Sci, 2010, 53: 2240–2250
5 St-Pierre M, Gingras D. Comparison between the unscented Kalman filter and the extended Kalman filter for the position estimation module of an integrated navigation information system. In: Proceedings of IEEE Intelligent Vehicles Symposium, Parma, 2004. 831–835
6 Huang Y, Chen Z J, Wei C. Least trace extended set-membership filter. Sci China Inf Sci, 2010, 53: 258–270
7 Julier S J, Uhlmann J K. Unscented filtering and nonlinear estimation. Proc IEEE, 2004, 92: 401–422
8 Wang J L, Feng X Y, Zhao L Q, et al. Unscented transformation based robust Kalman filter and its applications in fermentation process. Chin J Chem Eng, 2010, 18: 412–418
9 Morelande M R, Moran B. An unscented transformation for conditionally linear models. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Honolulu, 2007. 3: 1417–1420
10 Arulampalam M S, Maskell S, Gordon N, et al. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. IEEE Trans Signal Process, 2002, 50: 174–188
11 Wang L, Wan J W, Liu Y H, et al. Cooperative localization method for multi-robot based on pf-ekf. Sci China Ser-F: Inf Sci, 2008, 51: 1125–1137
12 van der Merwe R, Doucet A, de Freitas N, et al. The unscented particle filter. In: Proceedings of Conference on Neural Information Processing Systems, Denver, 2000. 584–590
13 Julier S J. The spherical simplex unscented transformation. In: Proceedings of American Control Conference, Denver, 2003. 3: 2430–2434
14 He W, Ge S S. Robust adaptive boundary control of a vibrating string under unknown time-varying disturbance. IEEE Trans Control Syst Technol, 2012, 20: 48–58
15 He W, He X, Ge S S. Vibration control of flexible marine riser systems with input saturation. IEEE/ASME Trans Mechatron, 2016, 21: 254–265
16 Zhan R H, Wan J W. Iterated unscented Kalman filter for passive target tracking. IEEE Trans Aerosp Electron Syst, 2007, 43: 1155–1163
17 He W, Ge S S, Huang D Q. Modeling and vibration control for a nonlinear moving string with output constraint. IEEE/ASME Trans Mechatron, 2015, 20: 1886–1897
18 Zhang S, He W, Huang D Q. Active vibration control for a flexible string system with input backlash. IET Control Theory Appl, 2016, 10: 800–805
19 He W, Ge S S, Li Y N, et al. Neural network control of a rehabilitation robot by state and output feedback. J Intell Robot Syst, 2015, 80: 15–31
20 Song T T, Ning X L, Yu W B. A new autonomous celestial navigation method for Mars probe. In: Proceedings of 6th International Symposium on Instrumentation and Control Technology: Signal Analysis, Measurement Theory, Photo-Electronic technology, and Artificial Intelligence, Beijing, 2006. 63574N
21 Li P, Cui H T, Cui P Y. Upf based autonomous navigation scheme for deep space probe. J Syst Eng Electron, 2008, 19: 529–536
22 Valarmathi J, Emmanuel D S, Christopher S. Velocity tracking based on interpolated adaptive Doppler filter. In: Proceedings of 15th International Conference on Information Fusion (FUSION), Singapore, 2012. 1511–1518
23 Krysik P, Wielgo M, Misiurewicz J, et al. Doppler-only tracking in GSM-based passive radar. In: Proceedings of 17th International Conference on Information Fusion (FUSION), Salamanca, 2014. 1–7