# Spacecraft Relative State Tracking Based on Genetic Algorithm-embedded Particle Filter

WenhuaCheng[1, a], Yasheng Zhang[1, b], Hong Yao[1, c]

[1]Equipment Academy of PLA, Beijing101416, PR China

[a]cheng007wenhua@163.com, [b]lizhizys@263.net, [c]momoridy@163.com

**Keywords:** Relative State, Genetic Algorithm, Particle Filter, GA-PF

**Abstract.** Spacecraft Relative State Tracking is not only the application foundation of space-based space surveillance system, but alsothe important prerequisite for space autonomous rendezvous and on-orbit servicing.Through making genetic algorithm (GA)embeddedPF resampling process, the particle degradation can be solved by using evolutionary thinking. Simulation result shows that GA-PF can achieve a good tracking for space target relative state.

## 1. Introduction

Spacecraft Relative State Tracking is not only the application foundation of Space-Based Space Surveillance (SBSS) system, as well as the important prerequisite for Space Autonomous Rendezvous and On-Orbit Servicing (OOS) [1-3]. Through the improvement of the relative position and attitude dynamics model and the precision of measurement equipment to improve the relative state tracking accuracy has attracted a lot of attention all around the world. However, due to limited cognitive ability, a high precision model still cannot be build, and the measurement data also has a variety of error. Therefore, the use of mathematical processing means is still the main way to improve the tracking accuracy [4].

For white Gaussian noise linear system, KalmanFilter (KF) is optimal, and the most extensive state estimation algorithm. For nonlinear systems, KFcan also make the state estimation via linearizing the model (Extended KalmanFilter, EKF), but the Particle Filter (PF) can provide a better estimation. For non-Gaussian systems, the KF is still the optimal linear filter algorithm, but notas effective as PF. The biggest problem of PF is that it's a particularly large amount of calculation. However, since the PFis easy and suitable for parallel computing, with the development of digital computer technology, PF will have broad application prospects [5].

The most common problem of PFis the particledegradation. Using resampling algorithm can solve the particle degradation, but it also affects the system's robust performance and reduces the diversity of particlesin practice, thus leading the PF algorithm performance lower, even leading the algorithm diverge. Through making genetic algorithm (GA)embedded PF resampling process, the particle degradation can be solved by using evolutionary thinking. Thus, an approach that not only guaranteeing the effective nessbut also increasing the particle diversity is proposed for the conflicting issues of effectiveness and diversity within resampling process [6].

## 2. Characteristics of GA-PF

As a random search technique based on natural selection and genetic, Genetic Algorithm makes the population updated and optimized by selection, crossover and mutation. This is exactly analogous to the filtering process of PF, Table.1 shows a comparison of PF and GA, by contrast, it can be found that both have highly compatible [7].

Table.1 Comparison of Particle Filter and Genetic Algorithm

| *PF* | *GA* |
|---|---|
| Time*k* | Generation*k* |
| Particle | population |
| Filter | Select |
| System noise | Crossover and mutation |
| Proportion | Fitness |

The basic idea of GA-PF is that taking particles as the chromosomal in GA, GA-embedded in the PF. For every particle's state with the proportion, firstly, coding the state as chromosome, then, making the corresponding calculation of particles follow the preset selection, crossover and mutation probability, to achieve the evolution of the particles. After that, the particles end up resampling by decoding chromosomes. The GA-embedded PF resampling process, not only can use the selection operator to select of outstanding individuals, but also can produce new individualsvia crossover and mutation. Therefore, setting the appropriate selection, crossover and mutation operators can ensure both of the effectiveness and diversity of particles [8].

## 3. Problem definition

### 3.1. Description of GA-PF

System state equation and measurement equation can be illustrated as:

$$\begin{aligned} \boldsymbol{X_{k+1}} &= \boldsymbol{X_k} + \boldsymbol{f}(\boldsymbol{X_k}, \boldsymbol{u_k}) + \boldsymbol{w_k} \\ \boldsymbol{Y_k} &= \boldsymbol{h}(\boldsymbol{X_k}) + \boldsymbol{v_k} \end{aligned} \quad (1)$$

Where $\mathrm{X_k}$ denotes the state of k times, $\mathrm{Y_k}$ denotes the measurement value of k times, $\mathrm{u_k}$ denotes the control impact, $\mathrm{w_k}$ and $\mathrm{v_k}$ denotes the random noise.

If the system is out of control, then the equation turns to:

$$\begin{aligned} \boldsymbol{X_{k+1}} &= \boldsymbol{X_k} + \boldsymbol{f}(\boldsymbol{X_k}) + \boldsymbol{w_k} \\ \boldsymbol{Y_k} &= \boldsymbol{h}(\boldsymbol{X_k}) + \boldsymbol{v_k} \end{aligned} \quad (2)$$

Process of classical PF is described in [9,10]. Here gives the steps of space target state tracking algorithm based on GA-PF.

**Step 1: Particles Creation**

Let the initial state be denoted by $\mathrm{X_0}$.Then, N particles are created based on possibility density function (PDF) of initial state, and denoted as $\mathrm{X^+_{0,i}}(\mathrm{i} = 1, \cdots, \mathrm{N})$, and $\widehat{\mathrm{X}}^+_{0,\mathrm{i}} = \mathrm{X}^+_{0,\mathrm{i}}$.

**Step 2: Particle State Estimation Update**

$$\widehat{\boldsymbol{X}}^-_{\boldsymbol{k+1,i}} = \widehat{\boldsymbol{X}}^+_{\boldsymbol{k,i}} + \boldsymbol{f}\left(\widehat{\boldsymbol{X}}^+_{\boldsymbol{k,i}}\right) + \boldsymbol{w_{k,i}} \quad (3)$$

Where $\mathrm{w_{k,i}}$ is based on $\mathrm{w_k}$ which PDF is known.

**Step 3: Particle PDF Update**

Each particle's PDF is calculated based on measurement value $\mathrm{Y_{k+1}}$ via the non-linear measurement equation:

$$q_{k+1,i} = p\left(\boldsymbol{Y_{k+1}} | \widehat{\boldsymbol{X}}^-_{\boldsymbol{k+1,i}}\right) \quad (4)$$

Then

$$q_{k+1,i} = \frac{q_{k+1,i}}{\sum q_{k+1,i}} \quad (5)$$

**Step 4: GA-PF resampling**

Let the particles are taken as a group. Then, the operations which based on genetic system are done with the group, such as selection, crossover and mutation.

*Selection*

Let fitness of each chromosome equals to particle's PDF. Selection is achieved via the Roulette Theory, that is, the fitness higher, the selection easier.

*Crossover*

Here, crossover adopts arithmetic cross. Arithmetic cross creates new populations via linear combination of two old populations:

$$\hat{\boldsymbol{X}}_{\boldsymbol{k+1,i}}^{+} = \alpha\hat{\boldsymbol{X}}_{\boldsymbol{k+1,i}}^{-} + (1-\alpha)\hat{\boldsymbol{X}}_{\boldsymbol{k+1,j}}^{-}$$
$$\hat{\boldsymbol{X}}_{\boldsymbol{k+1,j}}^{+} = \alpha\hat{\boldsymbol{X}}_{\boldsymbol{k+1,j}}^{-} + (1-\alpha)\hat{\boldsymbol{X}}_{\boldsymbol{k+1,i}}^{-} \quad (6)$$

Where $\alpha$ is a random number between 0 and 1.

*Mutation*

Mutation is to change values of chromosome randomly. Here, mutation adopts real number mutation:

$$\begin{cases}\hat{\boldsymbol{X}}_{\boldsymbol{k+1,i}}^{+} = (1+\beta)\hat{\boldsymbol{X}}_{\boldsymbol{k+1,i}}^{-}, \alpha < 0.5 \\ \hat{\boldsymbol{X}}_{\boldsymbol{k+1,i}}^{+} = (1-\beta)\hat{\boldsymbol{X}}_{\boldsymbol{k+1,i}}^{-}, \alpha \geq 0.5\end{cases} \quad (7)$$

Where $\beta$ is a random real number.

After that, the new group$\hat{X}_{k+1,i}^{+}(i = 1, \cdots, N)$ is put out.

**Step 5: Repetition**

Repeat step 2~step 4, and get state estimation of each time.

$$\hat{\boldsymbol{X}}_{k} = \sum q_{k,i}\hat{\boldsymbol{X}}_{k,i}^{-} \quad (8)$$

The flow chart is shown in Fig.1.

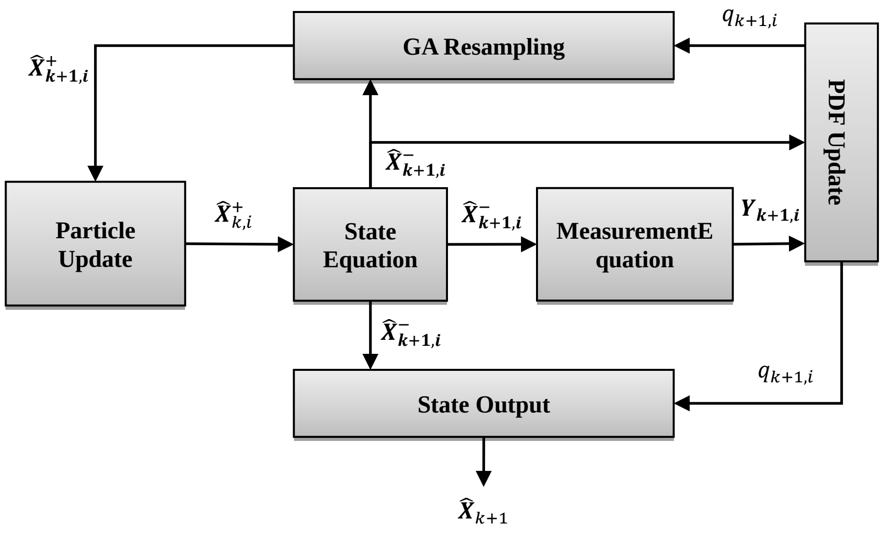


Fig.1 flow chart of GA-PF

## 3.2. Relative State Model

The relative state model is proposed in [11], as follows.

$$\dot{\boldsymbol{X}} = \begin{bmatrix}\dot{\boldsymbol{R}}\\ \dot{\boldsymbol{v}}\\ \dot{\boldsymbol{q}}\\ \dot{\boldsymbol{\omega}}\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{v}}\\ \frac{\mu}{{R_S}^3}\left(\frac{3x}{R_S}\boldsymbol{R}_S\right) - 2\boldsymbol{\omega}_0 \times \dot{\boldsymbol{R}} - \dot{\boldsymbol{\omega}}_0|_S \times \boldsymbol{R} - \boldsymbol{\omega}_0 \times (\boldsymbol{\omega}_0 \times \boldsymbol{R}) - \boldsymbol{f}_S\\ \frac{1}{2}\begin{bmatrix}\boldsymbol{\omega} & -[\boldsymbol{\omega}\times]\\ 0 & -\boldsymbol{\omega}^T\end{bmatrix}\boldsymbol{q}\\ \boldsymbol{D}(\boldsymbol{q})\boldsymbol{I}_{\boldsymbol{T}}^{-1}[-(\boldsymbol{\omega}_{\boldsymbol{S}}+\boldsymbol{\omega}) \times \boldsymbol{D}(\boldsymbol{q})\boldsymbol{I}_{\boldsymbol{T}}(\boldsymbol{\omega}_{\boldsymbol{S}}+\boldsymbol{\omega})] - \boldsymbol{I}_{\boldsymbol{S}}^{-1}(\boldsymbol{T}_{\boldsymbol{S}} - \boldsymbol{\omega}_{\boldsymbol{S}} \times \boldsymbol{I}_{\boldsymbol{S}}\boldsymbol{\omega}_{\boldsymbol{S}})\end{bmatrix} \quad (9)$$

Measurement model:

$$x_p = \frac{f}{Z_P} X_P$$
$$y_p = \frac{f}{Z_P} Y_P \tag{10}$$

Then, define state variables as

$$\boldsymbol{X} = [\boldsymbol{R}^T, \boldsymbol{v}^T, \boldsymbol{q}^T, \boldsymbol{\omega}^T]^T$$

Measurement variables

$$\boldsymbol{Y} = \left[x_{p1}, y_{p1}, \cdots, x_{pN}, y_{pN}\right]^T$$

After all above, the filter model can be represented as

$$\dot{\boldsymbol{X}} = \boldsymbol{f}(\boldsymbol{X}, \boldsymbol{u}) + \boldsymbol{w}$$
$$\boldsymbol{Y} = \boldsymbol{h}(\boldsymbol{X}) + \boldsymbol{v} \tag{11}$$

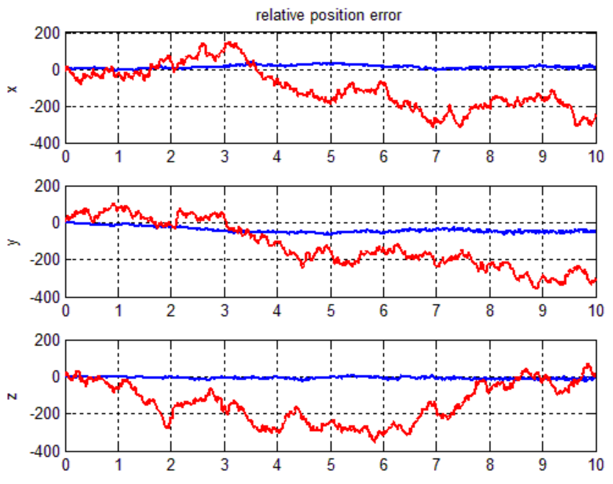
Where u denotes $f_S$ and $T_S$.

## 4. Simulation and result

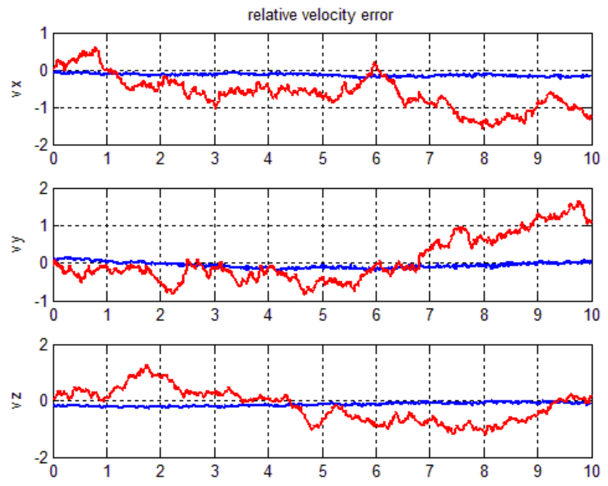
In this section, the GA-PF algorithm will be validated via a simulation example. Initialization conditions as follows.

| *Motion parameters* | |
|---|---|
| Servicer orbit parameters | $\{\boldsymbol{7000km} \quad \boldsymbol{0.1} \quad \boldsymbol{60°} \quad \boldsymbol{100°} \quad \boldsymbol{30°} \quad \boldsymbol{0s}\}$ |
| Inertia matrix | $I_S = I_T = diag[100 \quad 110 \quad 120](kgm^2)$ |
| Initial relative position | $R_0 = [0 \quad 0 \quad 50000]^T(m)$ |
| Initial relative velocity | $v_0 = [100 \quad 100 \quad 100]^T(m/s)$ |
| Initial relative attitude | $q_0 = [0.5 \quad 0.5 \quad 0.5 \quad 0.5]^T$ |
| Initial relative angular velocity | $\omega_0 = [0.01 \quad 0.01 \quad 0.01]^T(rad/s)$ |
| *Filter parameters* | |
| System noise | $\boldsymbol{w_k} \sim \boldsymbol{N}(\boldsymbol{0}, [\boldsymbol{10} \quad \boldsymbol{1} \quad \boldsymbol{0.01} \quad \boldsymbol{0.001}])$ |
| Measurement noise | $v_k \sim N(0, 0.00001)$ |
| Initial particle number | $N = 100$ |
| Initial particles' state | $p(R_{0,i}) \sim N(R_0, 10)$, $p(v_{0,i}) \sim N(v_0, 1)$ $p(q_{0,i}) \sim N(q_0, 0.01)$, $p(\omega_{0,i}) \sim N(\omega_0, 0.001)$ |
| *GA parameters* | |
| The number of iterations | $\boldsymbol{n = 10}$ |
| Crossover possibility | $P_c = 0.6$ |
| Mutation possibility | $P_m = 0.01$ |
| *Simulation parameters* | |
| Simulation time | $\boldsymbol{t = 10s}$ |
| Simulation step size | $h = 0.01s$ |

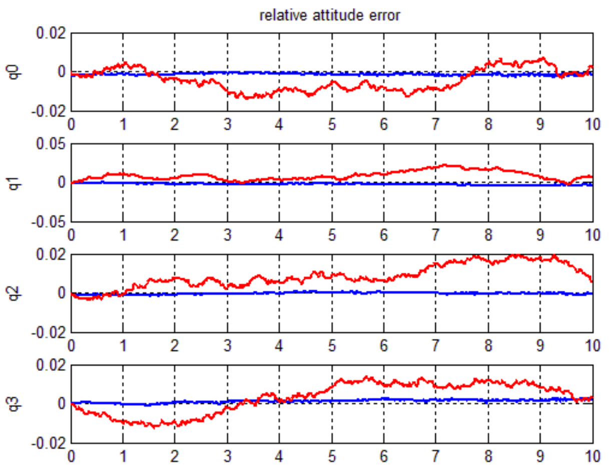
The simulation results are illustrated in Fig.2 and Fig.3.



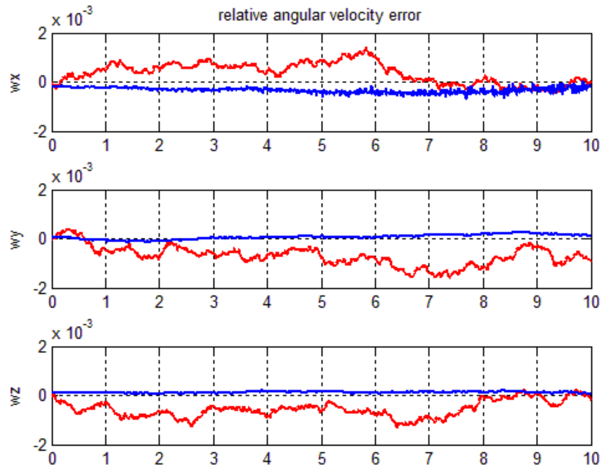
(a) relative position



(b) relative velocity



(c) relative attitude



(d) relative angular velocity

Fig.2 relative state estimation error curve

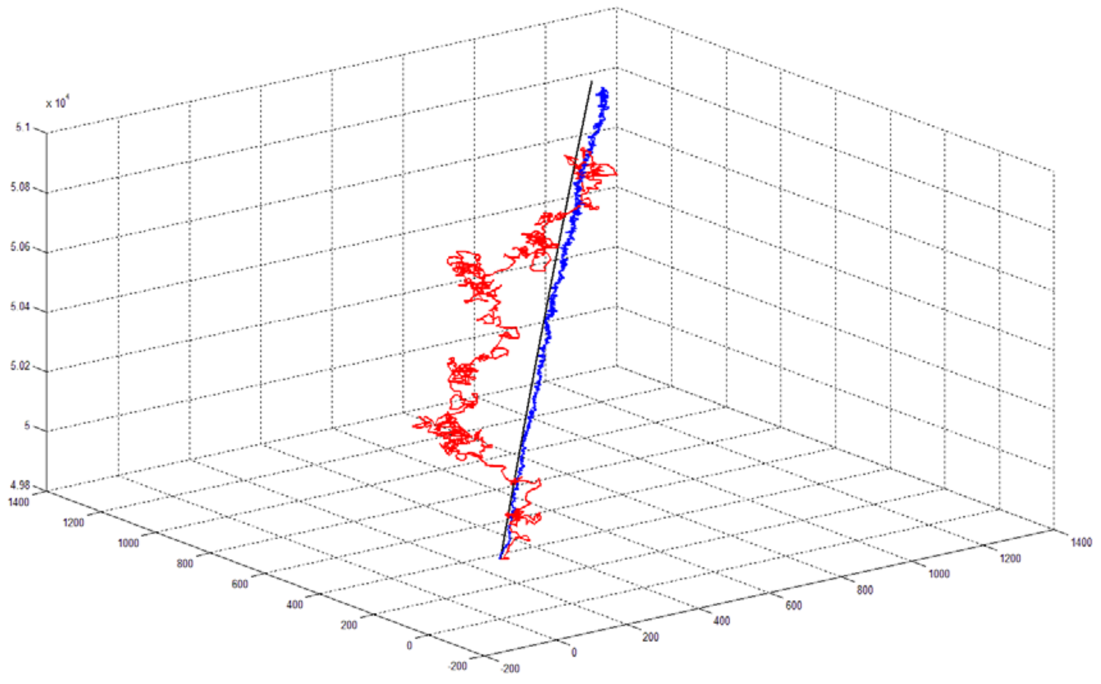


Fig.3 relative spatial position

In Fig.2 and Fig.3, the red curve shows the relative state estimation error without the filter, and the blue curve represents the relative state change with GA-PF, and the black curve (in Fig.3) represents the real state. Contrast from the picture without the filter and the filter can be seen that: in the absence of filter, the relative state error is large, and after GA-PF, the relative state error are controlled in a very small range near 0, indicating that GA-PF can achieve a good tracking for space target relative state.

## 5. Conclusion

In this paper, the GA-PF algorithm's characteristics are analysed. Then, the description of GA-PF algorithm is proposed. After that, via the example simulation, the feasibility and effectiveness of the algorithm are validated. The simulation result shows that GA-PF can achieve a good tracking for space target relative state.

## Reference

[1] Fehse W. Automated rendezvous and docking of spacecraft[M]. Cambridge university press, 2003.

[2] On-Orbit Satellite Servicing Study Project Report[R]. NASA Goddard Space Flight Center(GSFC), 2012.

[3] Jayant S, Cur-von B. toward operational space-based space surveillance[J]. Lincoln laboratory Journal, 2002, 13(2): 309-344.

[4] Liao ying, Liu Guangming, Wen Yuanlan, et al. Passive Tracking Technology of Non-Cooperative Space Target and Application[M]. National Defense Industry Press, 2014.

[5] Simon D. Optimal state estimation: Kalman, H infinity, and nonlinear approaches[M]. John Wiley & Sons, 2006.

[6] Zhan Ronghui, Zhang Jun, et al. Nonlinear Filtering Theory with Target Tracking Application[M]. National Defense Industry Press, 2014.

[7] Zhang Jinghai. Research on Particle Filtering Based on Genetic Algorithm[D]. Tianjin University, 2009.

[8] Ye Long, Wang Jingling, Zhang Qin. Genetic Resampling Particle Filter[J]. ACTA AUTOMATICA SINICA, 2010, 33(8):885-887.(in Chinese)

[9] Wang Zhixian. Optimal Estimation and System Identification[M]. Northwestern Polytechnical University Press, 2004.

[10] Crassidis J L, Junkins J L. Optimal Estimation of Dynamic Systems[M]. CRC press, 2011.

[11] Zhou Ding. Estimation of Relative State Between Non-cooperative Spacecraft Based on Stereo-Vision[D]. Harbin Institute of Technology, 2015.