A Horizontal Tracking Algorithm Suitable for Airborne Collision Avoidance System

Yuchen Wang, Liangfu Peng*

College of Electronic Information, Southwest Minzu University, Chendu, China
Email: 490680108@qq.com

Abstract. The horizontal tracker is essential for the reliable operation of the Airborne Collision Avoidance System (ACAS). In ACAS target tracking, for the non-linear state estimation problem of using sensor measurement values to track in Cartesian coordinates, this paper proposes a horizontal tracking algorithm based on the Augmented Unscented Kalman Filter (AUKF) to achieve the horizontal of the target. Accurate tracking of direction. Firstly, the tracking model of the relative horizontal state of the target aircraft is established, and then the data is processed by AUKF. In order to verify the effectiveness of the horizontal tracking algorithm, the computer simulation method is used to simulate the track of the local aircraft and the intruder in the horizontal direction, and noise is added to the measured values. The traditional Kalman Filter In Polar Coordinates (KFPC) and AUKF algorithm are used to filter the ACAS horizontal tracking. The simulation results show that AUKF can achieve more accurate target tracking.

Keywords: airborne collision avoidance system (ACAS); horizontal tracker; coordinate transformation; augmented unscented Kalman filter (AUKF)

1 Introduction

Airborne Collision Avoidance System (ACAS) is an air traffic collision avoidance system that does not rely on ground equipment. It can detect aircraft equipped with ACAS transponders in its adjacent airspace and report potential air traffic conflicts to the pilot. And provide instructions for anti-collision measures. The ACAS monitoring module obtains information about the altitude, distance and azimuth angle of the target aircraft through the secondary radar and ACAS receiver [1], and then the tracking module filters and predicts the monitoring data, and finally forms a complete track tracking of the target aircraft information. The accurate tracking of the horizontal tracker is essential to the reliable operation of ACAS.

The tracking of surrounding targets is a three-dimensional space target tracking problem. For this tracking model, some scholars proposed to separate the aircraft anti-collision model into horizontal anti-collision and vertical anti-collision, simplifying the complex three-dimensional anti-collision problem into two two-dimensional anti-collision problems [2-3]. In ACAS target tracking, the Cartesian coordinate system is usually used for modeling, and the measured value can be directly obtained in the original sensor coordinate system [4]. Therefore, using sensor coordinate measurements to track in Cartesian coordinates is actually a nonlinear state estimation problem. For this problem, some scholars have proposed the Kalman Filter In Polar Coordinates (KFPC) to avoid the problem of nonlinear coordinate conversion [5]. KFPC avoids the coordinate conversion error, reduces the amount of calculation, improves the positioning accuracy, and has a better working effect when the model is inaccurate. Some scholars have proposed that the particle filter (Particle Filter, PF) gets rid of the constraints of the Gaussian model [6]. PF can adapt to the accurate estimation of the nonlinear state, but it will bring huge computational complexity and particle degradation, and its computational complexity will be Hinder its application in real-time applications [7]. Extended Kalman Filter (EKF) is widely used to determine the target state [8], and some scholars have proposed that EKF that overcomes nonlinearity should be applied to ACAS target tracking [9]. EKF can solve the nonlinear problem of ACAS well, but when the system is highly nonlinear, EKF will produce larger errors. Unscented Kalman Filter (Unscented Kalman Filter, UKF) uses Sigma points to approximate the target state instead of a linearization process. This feature makes it more adaptive than EKF when dealing with nonlinear estimations [10].

Based on the sensor error of the ACAS principle and the uncertainty of the aircraft's future trajectory, this paper proposes a horizontal tracking algorithm based on AUKF to achieve accurate tracking of the target's horizontal direction.

2 ACAS Horizontal Tracking Model

In the ACAS active monitoring and tracking module, its position measurement and angle measurement respectively use the tracker to receive the raw measurement data of the sensor. As shown in Table 1, the raw measurement data of the sensor received by the tracker.

Table 1. Sensor raw measurement data.

Variable	Description
r_{slant}	Slope distance to an intruder
h_0	The absolute air pressure altitude of the own relative to 29.92 inches of mercury
h_1	The absolute air pressure altitude of the intruder relative to 29.92 inches of mercury
χ	The azimuth angle of the own relative to the intruder
ψ	The heading angle of the aircraft (the direction of the nose)

The sensor output information frequency is 1Hz. The output of the tracker is the horizontal distance r of the intruder, which is different from the direction of the intruder θ. The space where the aircraft flies is three-dimensional space [11], as shown in Figure 1 is the three-dimensional space collision avoidance model of the aircraft.

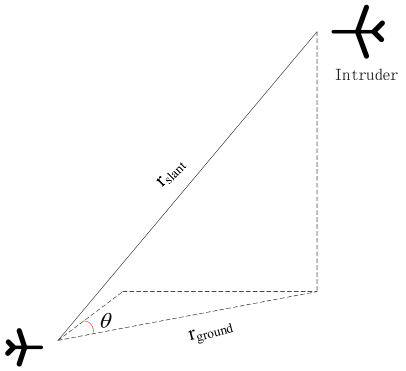

Figure 1. Aircraft three-dimensional space collision avoidance model.

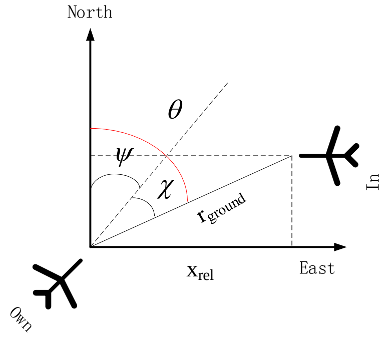

Figure 2. Horizontal collision avoidance model in Cartesian coordinates.

After projecting onto the horizontal plane, the distance between the two planes in the relative north and relative east directions can be calculated. As shown in Figure 2, the Cartesian coordinate system is established with the own as the center in the east and north directions.

3 AUKF

3.1 Unscented Transformation

Unscented Transformation (UT) transformation is the core of UKF. UKF is used to deal with systems in which state equations or observation equations are nonlinear. UT transform provides a method of approximating the probability distribution, using a set of deterministic points selected based on prior conditions, these points are called sigma points. Substitute the sigma points into the equation to obtain the corresponding point sets, and then use these point sets to obtain the transformed mean and covariance.

For the nonlinear transformation $y = f\left(x\right)$, the state x is n-dimensional, its mean is $\overline{X}$, and the variance is Pxx. 2n+1 sigma points are obtained through UT transformation:

$$\begin{cases} X^{(0)} = \overline{X}, i = 0 \\ X^{(i)} = \overline{X} + \left(\sqrt{(\mathrm{n} + \lambda)P_{xx}}\right)_{i}, i = 1 \sim n \\ X^{(i)} = \overline{X} - \left(\sqrt{(\mathrm{n} + \lambda)P_{xx}}\right)_{i}, i = n + 1 \sim 2n \end{cases} \tag{1}$$

Calculate the weight of each sigma point:

$$\begin{cases} \omega_{m}^{(\mathrm{i})} = \dfrac{\lambda}{n + \lambda}, i = 0 \\ \omega_{c}^{(\mathrm{i})} = \dfrac{\lambda}{n + \lambda} + \left(1 - \alpha^{2} + \beta\right), i = 0 \\ \omega_{m}^{(\mathrm{i})} = \omega_{c}^{(\mathrm{i})} = \dfrac{\lambda}{2(n + \lambda)}, i = 1 \sim 2n \end{cases} \tag{2}$$

where: $\omega_{m}^{(\mathrm{i})}$ represents the weighted weight of the mean, $\omega_{c}^{(\mathrm{i})}$ represents the weighted weight of the covariance.

The selection of the sigma point is based on the prior mean and the prior covariance, and the transformation of the sigma point is based on the selected UT scaling parameter. The parameter $\lambda = \alpha^{2}\left(n + \kappa\right) - n$ is a scaling parameter. The scaling of UT transform can be completely determined by three scaling parameters [12]. The main parameter α is the scale factor, which controls the distribution of sampling points. Usually a small amount greater than zero is selected to make the sigma point distribution more compact. The second parameter β is a non-negative weight coefficient. For Gaussian distribution, β=2 is the best choice. When the state dimension is greater than or equal to 3, the third parameter κ is usually set to 0.

3.2 AUKF

In linear systems, Kalman Filter (KF) is the optimal filter [13]. But there are always various nonlinearities in ACAS tracking. Both EKF and UKF are extensions of KF. UKF provides a method that uses the sigma point of the UT transformation to approximate the target state, which can estimate the probability distribution generated by the nonlinear transformation [10]. EKF provides another way, that is, instead of approximating the probability distribution of the result, but approximating the nonlinear transformation itself through some local approximation. This requires calculating the derivative, and the accuracy of the approximation is determined by the highest order of the calculated derivative. In ACAS horizontal tracking, UKF is preferred because it does not require these calculations, and when the system is highly non-linear, UKF usually produces more accurate results. For the case where the process equation and the measurement equation are nonlinear with respect to the noise, in order to improve the tracking accuracy, AUKF (Augmented Unscented Kalman Filter, AUKF) is used to improve the filtering accuracy.

When the process equation and measurement equation are nonlinear with respect to noise, the system equation is:

$$\begin{cases} X(k+1) = f[x(k), W(k)] \\ Z(k) = h[x(k), V(k)] \end{cases} \tag{3}$$

where: $X\left(k\right)$ is the system state vector, $Z\left(k\right)$ is the observation vector of the system, $W\left(k\right)$ is the system noise, and d is the observation noise. Suppose $W\left(k\right)N\left(0, Q\left(k\right)\right), V\left(k\right)N\left(0, R\left(k\right)\right)$.

The filtering algorithm is:

1) Extend the noise to the state vector to obtain the expanded dimension state:

$$X_a\left(k\right) = \left[x(k)\ W(k)\ V(k)\right]^T \tag{4}$$

2) Use UKF to estimate the expansion state, the initial value of UKF is set as:

$$\begin{cases} \hat{X}_a\left(k-1\right) = \left[\hat{x}(0 \mid 0)\ 0\ 0\right]^T \\ P_a(0|0) = \begin{bmatrix} \mathrm{E}\left[(x - \hat{x}(0|0))(x - \hat{x}(0|0))^{\mathrm{T}}\right] & 0 & 0 \\ 0 & Q(0) & 0 \\ 0 & 0 & R(0) \end{bmatrix} \end{cases} \tag{5}$$

3) Use UT transformation, calculate the extended dimension sigma point:

$$\begin{cases} X_a^0(k-1) = \hat{X}_a(k-1), i = 0 \\ X_a^{(i)}(k-1) = \hat{X}_a(k-1) + \left(\sqrt{(n+\lambda)P_a(k-1)}\right)_i, i = 1 \sim n \\ X_a^{(i)}(k-1) = \hat{X}_a(k-1) - \left(\sqrt{(n+\lambda)P_a(k-1)}\right)_i, i = n+1 \sim 2n \end{cases} \tag{6}$$

4) Time update:

$$\begin{cases} X_X^{(i)}(k \mid k-1) = f\left(X_X^{(i)}(k-1), X_w^{(i)}(k-1)\right) \\ \hat{X}(k \mid k-1) = \sum_{i=0}^{2n} W_m^{(i)} X_X^{(i)}(k \mid k-1) \\ P_X(k \mid k-1) = \sum_{i=0}^{2n} W_c^{(i)} \left[X_X^{(i)}(k \mid k-1) - \hat{X}(k \mid k-1)\right]\left[X_X^{(i)}(k \mid k-1) - \hat{X}(k \mid k-1)\right]^{\mathrm{T}} \\ Z_X^{(i)}(k \mid k-1) = h\left(X_x^{(i)}(k-1), X_v^{(i)}(k-1)\right) \\ \hat{Z}(k \mid k-1) = \sum_{i=0}^{2n} W_m^{(i)} Z_X^{(i)}(k \mid k-1) \end{cases} \tag{7}$$

5) Calculate the variance and covariance of the observations predicted by the system:

$$\begin{cases} P_{ZZ}(k \mid k) = \sum_{i=0}^{2n} W_c^{(i)} \left[\hat{Z}^{(i)}(k \mid k-1) - \hat{Z}(k \mid k-1)\right]\left[\hat{Z}^{(i)}(k \mid k-1) - \hat{Z}(k \mid k-1)\right]^{\mathrm{T}} \\ P_{XZ}(k \mid k) = \sum_{i=0}^{2n} W_c^{(i)} \left[X_X^{(i)}(k \mid k-1) - \hat{X}(k \mid k-1)\right]\left[\hat{Z}^{(i)}(k \mid k-1) - \hat{Z}(k \mid k-1)\right]^{\mathrm{T}} \end{cases} \tag{8}$$

6) Calculate the Kalman gain matrix:

$$K(k) = P_{XZ}(k \mid k) P_{ZZ}^{-1}(k \mid k) \tag{9}$$

7) Calculate the status update and covariance update of the system

$$\begin{cases} \hat{X}(k \mid k) = \hat{X}(k \mid k-1) + K(k)[Z(k) - \hat{Z}(k \mid k-1)] \\ P(k \mid k) = P(k \mid k-1) - K(k) P_{ZZ} K^{\mathrm{T}}(k) \end{cases} \tag{10}$$

4 Implementation of Horizontal Tracker

Expand the dimension of the state vector to get the augmented state x^{a}:

$$x^{\mathrm{a}} = \left[x \; w\right]^T \tag{11}$$

Step 1 initialization

1) Perform sigma sampling on the observation vector $\left[r_{\mathrm{slant}} \;\; \theta \; h_{\mathrm{rel}}\right]$ to obtain the sigma point set of z_{p}:

$$\begin{cases} x^0(0 \mid 0) = z_{\mathrm{p}}(0) \\ x^j(0 \mid 0) = z_{\mathrm{p}}(0) + (\sqrt{(n+\lambda)}\sqrt{R})_j, j = 1 \sim n \\ x^j(0 \mid 0) = z_{\mathrm{p}}(0) - (\sqrt{(n+\lambda)}\sqrt{R})_j, j = n+1 \sim 2n \end{cases} \tag{12}$$

2) Perform a nonlinear transformation on the sigma point:

$$z^j = h_2\left(x^j\right), j = 0 \sim 2\mathrm{n} \tag{13}$$

3) Calculate the mean and covariance of z^j:

$$\begin{cases} \hat{z}(0 \mid 0) = \sum_{j=0}^{2n} \omega_m^{(j)} z^j(0 \mid 0) \\ P_z(0 \mid 0) = \sum_{j=0}^{2n} \omega_c^{(j)} \left[z^j - \hat{z}(0 \mid 0)\right]\left[z^j - \hat{z}(0 \mid 0)\right]^T \end{cases} \tag{14}$$

4) Initial state estimation:

$$\begin{cases} x(0 \mid 0) = \left[\hat{z}^{\mathrm{T}}(0 \mid 0) \quad 0 \quad 0\right]^{\mathrm{T}} \\ x^a(0 \mid 0) = \left[x^{\mathrm{T}}(0 \mid 0) \quad 0 \quad 0\right]^{\mathrm{T}} \end{cases} \tag{15}$$

5) Initial covariance estimation:

$$\begin{cases} P_z(0 \mid 0) = P_z(0 \mid 0) \\ P_x(0 \mid 0) = \begin{bmatrix} P_z(0 \mid 0) & 0 \\ 0 & V \end{bmatrix} \end{cases} \tag{16}$$

where: $x = \left[x_{\mathrm{rel}} \;\; y_{\mathrm{rel}} \;\; \dot{x}_{\mathrm{rel}} \;\; \dot{y}_{\mathrm{rel}}\right]^{\mathrm{T}}$, $w = \left[w_r \;\; w_\theta\right]^{\mathrm{T}}$, $z_{\mathrm{p}}(0) = \left[r_{\mathrm{slant}}(0) \;\; \theta(0)\right]^{\mathrm{T}}$.

Step 2 state estimation

1) State prediction:

$$\begin{cases} \hat{x}\left(k \mid k-1\right) = F\hat{x}\left(k-1 \mid k-1\right) \\ P^{\mathrm{xx}}\left(k \mid k-1\right) = FP^{\mathrm{xx}}\left(k-1 \mid k-1\right)F^{\mathrm{T}} + BQB^{\mathrm{T}} \end{cases} \tag{18}$$

2) Construct the expanded sigma point:

$$\begin{cases} x^{a,0}\left(n \mid n-1\right) = \hat{x}^a\left(k \mid k-1\right) = \left[\hat{x}\left(k \mid k-1\right) \;\; 0 \;\; 0\right]^{\mathrm{T}} \\ P^a\left(k \mid k-1\right) = \begin{bmatrix} P^{xx}\left(k \mid k-1\right) & 0 \\ 0 & R \end{bmatrix} \\ x^{a,j}(k \mid k-1) = \hat{x}^a(k \mid k-1) + (\sqrt{n+\lambda}\sqrt{P^a\left(k \mid k-1\right)}, j = 1 \sim n \\ x^{a,j}\left(k \mid k-1\right) = \hat{x}^a\left(k \mid k-1\right) - \left(\sqrt{n+\lambda}\sqrt{P^a\left(k \mid k-1\right)}\right), j = n+1 \sim 2n \\ \left[x_x^{a,j} \;\; x_v^{a,j}\right]^{\mathrm{T}} = x^{a,j}\left(k \mid k-1\right) \end{cases} \tag{19}$$

3) Obtain the sigma point, and then perform nonlinear transformation to obtain the sigma point set of the observation value in the polar coordinate system:

$$\begin{cases} z_{pol}^{j} = h_1\left(x_x^{a,j}\right) + x_v^{a,j} \\ y^j\left(k \mid k-1\right) = h_2\left(z_{pol}^{j}\right) \end{cases} \tag{20}$$

4) Observation prediction:

$$\begin{cases} P^{zz}\left(k \mid k-1\right) = \sum_{j=0}^{2n} \omega_m^{(j)} \left[y^j\left(k \mid k-1\right) - \hat{z}\left(k \mid k-1\right)\right]\left[y^j\left(k \mid k-1\right) - \hat{z}\left(k \mid k-1\right)\right]^{\mathrm{T}} \\ P^{xz}\left(k \mid k-1\right) = \sum_{j=0}^{2n} \omega_c^{(j)} \left[x_x^j\left(k \mid k-1\right) - \hat{x}\left(k \mid k-1\right)\right]\left[y^j\left(k \mid k-1\right) - \hat{z}\left(k \mid k-1\right)\right]^{\mathrm{T}} \end{cases} \tag{21}$$

5) Convert the observation value in the polar coordinate system to the Cartesian coordinate system, and finally update:

$$\begin{cases} K\left(k\right) = P^{xz}\left(k \mid k-1\right) P^{zz}\left(k \mid k-1\right)^{-1} \\ \hat{x}\left(k \mid k\right) = \hat{x}\left(k \mid k-1\right) + K\left(z^0\left(k\right) - \hat{z}\left(k \mid k-1\right)\right) \\ P^{xx}\left(k \mid k\right) = P^{xx}\left(k \mid k-1\right) - KP^{zz}\left(k \mid k-1\right)K^{\mathrm{T}} \end{cases} \tag{22}$$

The post-state distribution $\mathsf{N}\left(x\left(k\right); \hat{x}\left(k \mid k\right), P\left(k \mid k\right)\right)$ at time k is a continuous distribution, because the computer cannot handle continuous mathematical equations. When the computer is used to deal with the problem, the logic accepts the discrete distribution, so the post-state distribution must be sampled to generate a sample set with relevant probabilities. Draw samples of sigma points from the post-state distribution, and convert each sample into a horizontal tracking bivariate state representation.

5 Simulation Examples and Comparison

The simulation was performed on MATLAB R2019a, RAM 8GB, and processor 1.6GHz. Assuming that the aircraft and the intruder are flying at a constant speed in the air, the initial horizontal position of the aircraft is in the east-north coordinate system (304.8, 152.4) m, the eastward speed is 304.8m/s, the northward speed is 152.4m/s, and the intruder starts The horizontal position is in the east-north coordinate system (76200, 182.88) m, the eastward speed is -304.8m/s, and the northward speed is 182.88m/s. The trajectory diagram of this aircraft and the intruder is shown in Figure 3. The initial value of the relative height of the two planes is set to 30.48m, and the speed in the height direction is 30.48m/s.

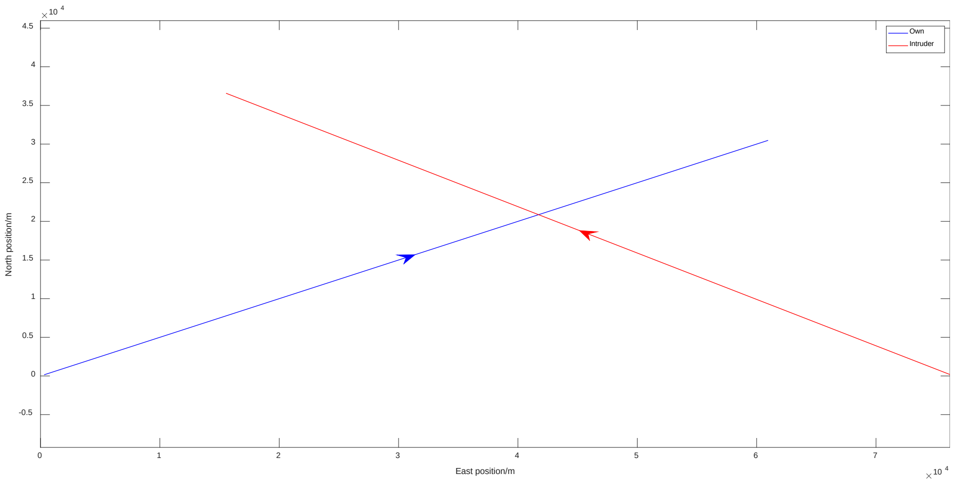

Figure 3. Own and intruder track.

The Monte Carlo simulation method is selected as a means to evaluate the performance of the algorithm. The idea of this method is to add random noise on the basis of the real value of the target movement to simulate the measurement value. Use MATLAB to compare the tracking results of the KFPC and AUKF horizontal trackers, and use the root mean square error (RMSE) to evaluate the performance of the filtering algorithm. The RMSE is defined as equation (23).

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(X_i - \hat{X}_i)^2} \tag{23}$$

The simulation results are shown in Figure 4-6. Figure 4 is a horizontal distance tracking trajectory map. Figures 5 and 6 are the RMSE of r and θ in the horizontal direction in the next 200 seconds. The data is generated by using Monte Carlo method to simulate 1000 encounters. According to the simulation results, it can be seen that AUKF has higher accuracy than KFPC, and the error is relatively stable. Compared with KFPC, AUKF has a more accurate state estimation for the two states of ACAS horizontal collision avoidance. At about 125 seconds, when the two planes are about to converge, KFPC fluctuates and the error increases, while AUKF has a good effect when approaching converge, with small errors and relatively stable. From the comparison of the average running time of KFPC and AUKF in Table 2, it can be seen that under the same hardware platform conditions, the running time of KFPC and AUKF is close, which verifies the effectiveness of AUKF in improving the real-time performance of the system.

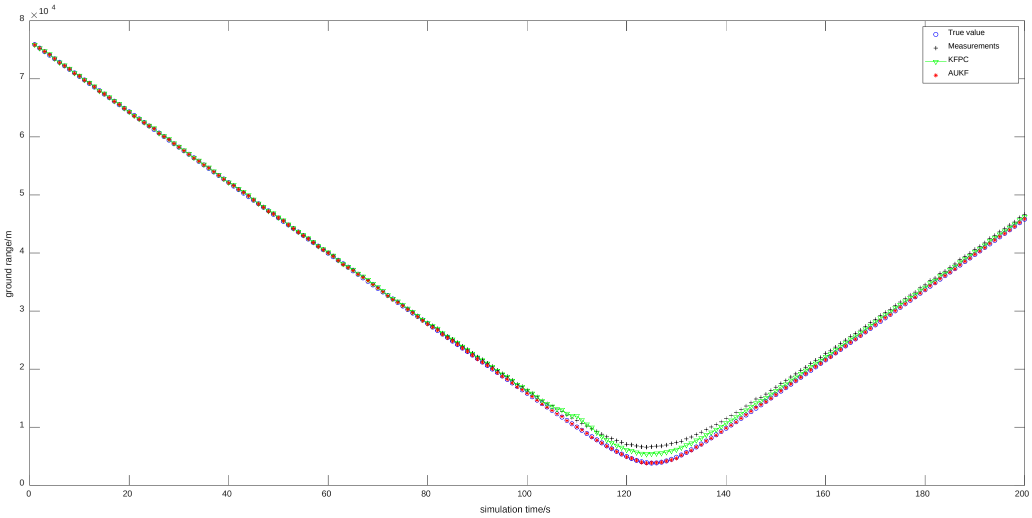

Figure 4. Horizontal collision avoidance model in Cartesian coordinates.

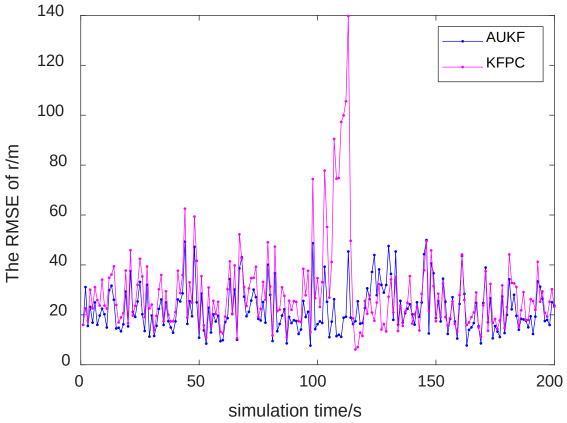

Figure 5. The RMSE of r.

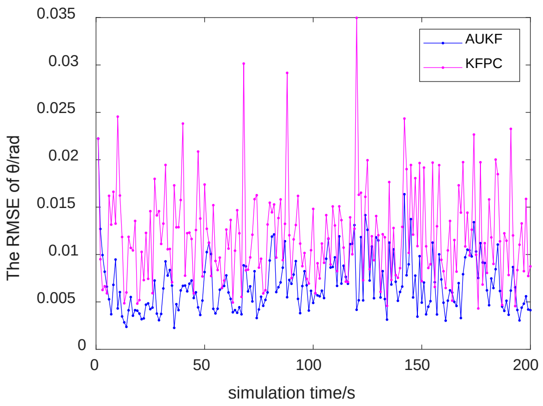

Figure 6. The RMSE of θ.

Table 2. Comparison of the average running time of KFPC and AUKF.

Algorithm	Average run time (s)
KFPC	4.5489×10^{-5}
AUKF	8.5216×10^{-5}

6 Discussion

The modern aviation industry has newer and higher requirements for air traffic control equipment. The use of high technology to improve the advanced nature of my country's civil aviation and military avionics systems is the strategic focus of the development of my country's aviation industry in the future. The improvement of current air traffic equipment, especially the study of air collision avoidance theory and technology, and the manufacture of domestically produced airborne collision avoidance equipment are an important direction for the development of my country's avionics system technology. This article mainly studies the horizontal tracking module of the target in ACAS. For the filtering estimation of nonlinear systems such as target tracking, UKF does not need to linearize the nonlinear state equations and observation equations. It uses the UT method to approximate the posterior probability with a set of determined sampling points, which improves the filtering accuracy. AUKF further improves the filtering accuracy by considering the noise item as a state quantity. The computer simulation results show that the AUKF target tracking algorithm can accurately track the maneuvering flying target, thereby depicting the precise flight trajectory of the air target. Therefore, applying AUKF to the nonlinear filtering problem of target tracking in ACAS has important engineering significance for ensuring flight safety and preventing aircraft collisions.

Acknowledgments. The work of this paper is supported by the Southwest Minzu University Graduate Innovative Research Project (Master Program CX2020SZ98). A special acknowledgement should be given to Southwest Minzu University for its experimental conditions and technical support.

References

1. T.-B. LI, “The design and implementation of hybrid surveillance for integrated traffic collision avoidance system,” University of Electronic Science and Technology of China, 2020.
2. L.-F. PENG, Y. YAN, J.-Q. SHI, et al, “Equivalence study on collision avoidance zone based models for general aviation aircraft collision avoidance,” China Safety Science Journal, 2016, 26(08): 95-99.
3. H.-L. ZHANG, “The Design of General Aviation Airborne Collision Avoidance Logic Based On MDP,” University of Electronic Science and Technology of China, 2019.
4. S-X. HUANG, H. LIU, “Method for radar target tracking based on additive sequential unscented Kalman filter,” Computer Engineering and Applications, 2010, 46(08): 214-216.

5. J.-T. XIA, Z. REN, L. CHEN, et al, “A Kalman filtering algorithm in polar coordinates,” Journal of Northwestern Polytechnical University, 2000, 18(03):396-399.
6. M.S. ARULAMPALAM, S. MASKELL, N. Gordon, et al, “A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking,” IEEE Transactions on Signal Processing, 2002, 50(02): 174-188.
7. R. KARLSSON and F. GUSTAFSSON, “Particle filtering for quantized sensor information,” 2005 13th European Signal Processing Conference, 2005: 1-4.
8. C.-F. GUO, Z.-X. DAI, L. YANG, et al, “Application of the strong tracking UKF in the maneuvering target tracking,” Journal of Physics: Conference Series, 2016, Vol.679(1): 1-5.
9. L. WANG, X. GAO, “Target tracking for traffic alert and collision avoidance system of civil aircraft,” Journal of Terahertz Science and Electronic Information Technology, 2015, 13(03): 415-418+430.
10. S.J. JULIER, J.K. UHLMANN, “Unscented filtering and nonlinear estimation,” Proceedings of the IEEE, 2004, 92(3): 401–422.
11. L.-F. PENG, Y.-S. LIN, “Study on the model for horizontal escape maneuvers in TCAS,” IEEE Transactions on Intelligent Transportation Systems, 2010, 11(02): 392-398.
12. S.J. JULIER, J.K. UHLMANN, “A new extension of the Kalman filter to nonlinear systems,” Proceeding of SPIE: The International Society for Optical Engineering, 1997, 3068: 182-193.
13. J.B. PEARSON, E.B. STEAR, “Kalman filter applications in airborne radar tracking,” IEEE, AES210, 1974: 319-329.