

Article

A General Framework for Flight Maneuvers Automatic Recognition

Jing Lu ^{1,2,*}, Hongjun Chai ² and Ruchun Jia ³

¹ College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

² College of Computer Science, Civil Aviation Flight University of China, Guanghan 618307, China; chaihongjun@cafuc.edu.cn

³ Wangjiang Campus, Sichuan University, Chengdu 610065, China; ruchunjia@zaibei.org.cn

* Correspondence: lujing_cafuc@nuaa.edu.cn

Abstract: Flight Maneuver Recognition (FMR) refers to the automatic recognition of a series of aircraft flight patterns and is a key technology in many fields. The chaotic nature of its input data and the professional complexity of the identification process make it difficult and expensive to identify, and none of the existing models have general generalization capabilities. A general framework is proposed in this paper, which can be used for all kinds of flight tasks, independent of the aircraft type. We first preprocessed the raw data with unsupervised clustering method, segmented it into maneuver sequences, then reconstructed the sequences in phase space, calculated their approximate entropy, quantitatively characterized the sequence complexity, and distinguished the flight maneuvers. Experiments on a real flight training dataset have shown that the framework can quickly and correctly identify various flight maneuvers for multiple aircraft types with minimal human intervention.

Keywords: flight maneuver recognition (FMR); unsupervised clustering; phase space reconstruction

Citation: Lu, J.; Chai, H.; Jia, R. A General Framework for Flight Maneuvers Automatic Recognition. *Mathematics* **2022**, *10*, 1196. <https://doi.org/10.3390/math10071196>

Academic Editor: Paolo Crippa

Received: 3 March 2022

Accepted: 1 April 2022

Published: 6 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Flight Maneuver, according to the standard definition given by the Federal Aviation Administration (FAA) [1], refers to a series of flight patterns of an aircraft under the control of the pilot. FMR as a key technology for automatic evaluation of flight technology is the focus of research on the application of artificial intelligence in the field of flight training. In the 1970s, for one-on-one, air-to-air combat training, NASA developed an adaptive maneuvering logic computer program (AML) [2,3], which provides a virtual competitor for human pilots at NASA Langley Research Center's (LRC) Differential Maneuvering Simulator (DMS). As AI, AML recognizes the maneuvers and intentions of the opponent and makes the right decisions to drive the next maneuvers. In addition, the study of flight maneuvers' aircraft loads is an important issue in the field of flight safety involving aircraft design, flight certification, and accident investigation, and FMR is the basic technology for this study. Barndt G. [4] examined how the Navy could process raw parameter data generated by HUMS to identify the maneuvers flown so as to support the structural monitoring function in 2007. Many studies in this area have been generated since then.

Although there is also a wide demand for FMR in the field of UAV and air combat research, due to the limitation of data sources and author's concentration, this paper only focuses on manned fixed-wing civil aviation training flight.

The raw data of FMR is multivariate time-series data generated from nonlinear aircraft power system, which has typical chaotic characteristics and cannot be directly

applied to common time series analysis methods. As an artificial mechanical operating system, the data performance of the same flight maneuver of different pilots of different types of aircraft is very different, not to mention the influence of environmental factors such as weather variation.

Essentially, FMR is a multiple nonlinear time-series pattern-recognition problem [5]. Pattern-recognition problems mainly include classification and clustering.

Classification-based FMR

In the time-series classification problem, feature volume construction and classifier design are the core problems. Time-series classification aims to take the whole time series as input to assign a discrete label. In FMR, different maneuvers often have different lengths due to differences in aircraft types, and the same maneuvers have different lengths due to differences in pilot operating habits. It is more difficult than the general classification problem owing to the inequational length of the classified time-series data, which makes it impossible to apply the general classification algorithm directly.

In order to solve these difficulties, there are usually two approaches. First, define the appropriate distance degree using a distance-based pattern-recognition method, such as Dynamic Time Warping (DTW) distance [6–8], Locality Sensitive Hash (LSH) distance [9], and Approximate Entropy [10]. The advantages of these methods are that they conform to the basic principles of pattern recognition; the more similar the patterns are, the smaller their distances are; and the algorithms are simple and easy to implement, do not limit the length of the time series between patterns, and can analyze nonlinear time series. The significant disadvantages are expensive calculation and inability to identify subtle differences between patterns.

Second, using knowledge rules or context-dependent modeling, each sequence is represented by an equal-length and same-dimension feature vector of model parameters and then trained and classified by a conventional classification algorithm, which is a domain-related approach called the model-based method. In general, model-based FMR methods can be divided into four categories: (1) feature extraction-based [11–14], (2) expert knowledge rule-based [15–27], (3) probabilistic graphical model-based [28–31], and (4) neural network-based [32–36].

(1) The main methods for feature extraction are SVD and SVM methods, combined with least squares or hierarchical classification methods, which reduce the computational effort by reducing the number of dimensions and compressing the data. The models are simple and easy to train but are not complete, and they are sensitive to temporal length and require manual prior knowledge. (2) The expert knowledge rule model method needs to establish the artificial rule knowledge database first, then use the pattern-matching query method to achieve recognition. The knowledge rule extraction method includes Natural Language Processing, Genetic Algorithm, and Swarm Optimization. This type of method is very widely used, with high recognition efficiency and correct rate, but the unavoidable disadvantages are high labor cost; the fact that a certain model only corresponds to a certain type of aircraft type or flight task; and the inability of the method to be generalized. (3) The probabilistic graphical model-based mainly uses hidden Markov model (HMM), Kalman filtering, and dynamic Bayesian methods, which can not only identify but also predict and only need a few parameters to form a complete model but cannot handle nonlinear time series. (4) The model based on neural network work uses deep neural network with fully supervised training method to constitute the model, with high recognition rate and good model maturity but also with high cost of integration with labeled data and computational complexity. Different aircraft types correspond to different models and need to be completely retrained.

Naturally, hybrid methods combining multiple methods have also been proposed [37,38]; these methods have better recognition performance but still do not have the ability to generalize.

Clustering-based FMR

In addition, some scholars have also conducted FMR from the perspective of clustering [39–42]. These methods do not require prior knowledge with the ability to generalize. However, the clustering results rely heavily on good temporal segmentation, and most of the papers appearing now use manual segmentation without automatic segmentation capability, and the clustered results still need to be interpreted by human experts and cannot correspond automatically.

In summary, it can be found that the existing literature methods all perform FMR for a certain task of a certain aircraft model and generally have the significant disadvantages of relying on manual expert knowledge, being unable to automatically segment, and being difficult to generalize.

To the best of our knowledge, there is no general framework that can automatically segment sequences and quickly discriminate between maneuvers with minimal human intervention.

This paper proposes a new general framework; the general idea is to integrate the automatic segmentation capability of unsupervised clustering and the ability of information entropy to distinguish sequence complexity.

This paper is organized as follows. Section 2 introduces the automatic segmentation method of flight maneuver sequence. Section 3 introduces the automatic recognition method of maneuver segments. Section 4 completely elaborates the overall framework of automatic FMR processing. Section 5 covers the experimental process and experimental results, and the conclusion is given in Section 6.

2. Sequence Segmentation

2.1. The Trend Fragmentation Algorithm

In this paper, a key parameter is selected for trend identification, and the index of all trend segments is obtained using the slope method combined with a height change threshold, using a sliding model with a double window.

The slope method is based on the least square method, where the sequence to be segmented is fitted to a straight line, and the main trend of the sequence is determined by comparing the slope of the line with a threshold size.

Set $\mathbf{D} = (y_1, y_2, \dots, y_m)^T$ is a sample set, $\mathbf{X} = (x_1, x_2, \dots, x_m)^T$ is the time sequence set, D_i is a subset of the samples, $i = (1, 2, \dots, L)$, and L is the number of trend segments. The model parameters are obtained by fitting the least squares method as in Equation (1).

$$\omega_i = (k_i, b_i)^T = (D_i^T D_i)^{-1} D_i^T \cdot X_i, i = (1, 2, \dots, L) \quad (1)$$

The height-change threshold is used to determine long, slow-climbing, or circling maneuvers in flight, which have small slopes and long durations and can be misjudged based on the slope alone. The algorithm is described in Algorithm 1.

Algorithm 1 The Trend Fragmentation Algorithm.

Input: sample set $\mathbf{D} = (y_1, y_2, \dots, y_m)^T$; time sequence set $\mathbf{X} = (x_1, x_2, \dots, x_m)^T$;

1. Set slope threshold k_s and height-change threshold Δ_s ;
2. Initial value of fitting parameter $\omega = (k, b)^T$;
3. Initialize the fixed window $\mathbf{F} = (x_{F_{start}}, x_2, \dots, x_{F_{end}})^T, F_{start} = 1, F_{end} = h$;
4. Initialize the sliding window $\mathbf{S} = (x_{S_{start}}, x_2, \dots, x_{S_{end}})^T, S_{start} = 1, S_{end} = h$;
5. Initialize the output sequence $O_j = (x_{F_{start}}, x_{F_{end}}, f)$;
6. Read a samples subset $D_i = (y_{S_{start}}, \dots, y_{S_{end}})^T$; time sequence set $X_i = S$;
7. Least squares fitting model to obtain parameters: ω_i ;
8. If $k \geq k_s$, identifies X_i as an upward trend, set $f = 'U'$;
9. Otherwise, if $-k_s < k < k_s$ identifies X_i as a level trend, set $f = 'L'$;
10. Otherwise, identifies X_i as a downward trend, set $f = 'D'$;

11. If X_i is not a level trend, and k 's signs are unchanged, set $F_{start} = F_{start}, F_{end} = F_{end} + 1, S_{start} = S_{start} + 1, S_{end} = S_{end} + 1$;
12. Otherwise, if k 's signs are changed, set $O_j = (x_{F_{start}}, x_{F_{end}}, f)$, $F_{start} = S_{start}, F_{end} = F_{start} + h, S_{start} = S_{start} + 1, S_{end} = S_{end} + 1$;
13. If X_i is a level trend, set $F_{start} = F_{start}, F_{end} = F_{end} + 1, S_{start} = S_{start} + 1, S_{end} = S_{end} + 1$;
14. Least squares fitting model to obtain parameters: $\omega_l = (D_l^T D_l)^{-1} D_l^T \cdot X_l$;
15. Calculate fixed window height change $\Delta = |k_l \cdot (F_{end} - F_{start})|$;
16. If $|k_l| > k_s$, set $f = 'L'$, set $O_j = (x_{F_{start}}, x_{F_{end}}, f)$, $F_{start} = S_{start}, F_{end} = F_{start} + h, S_{start} = S_{start} + 1, S_{end} = S_{end} + 1$;
17. If $|k_l| < k_s$, and $\Delta > \Delta_s$, set $f = 'U'$ ($k_l > 0$) or $'D'$ ($k_l < 0$), set $O_j = (x_{F_{start}}, x_{F_{end}}, f)$, $F_{start} = S_{start}, F_{end} = F_{start} + h, S_{start} = S_{start} + 1, S_{end} = S_{end} + 1$;
18. Otherwise, if $|k_l| < k_s$, and $\Delta < \Delta_s$, set $F_{start} = F_{start}, F_{end} = F_{end} + 1, S_{start} = S_{start} + 1, S_{end} = S_{end} + 1$;
19. If $S_{end} \geq m$, end iterations; otherwise, go back to 7.

Output: O .

The core of Algorithm 1 is to use the sliding double window method to fit the slope to the original data and determine the flight attitude as ascending, leveling, or descending at that time based on the slope and use the change in attitude as the signal for automatic sequence segmentation. The k_s is slope threshold, Δ_s is height-change threshold, ω is fitting matrix, F is the fixed window, S is the sliding window, O_j is the output subsequence, f is the flag bit, and takes values in the range $\{ 'U', 'L', 'D' \}$.

2.2. The Clustering Algorithm

With Algorithm 1, we obtain the trend segments, and this section will use the dynamic clustering method ISODATA (Iterative Self Organizing Data Analysis Techniques Algorithm) to complete the segment classification. ISODATA algorithm automatically selects a number of samples as cluster centers and adjusts the class centers by sample mean iteration in subsequent calculations and realizes the adjustment of cluster center data by merging and splitting of patterns. However, the input data are time series, so the algorithm cannot be used directly; therefore, this paper improves the algorithm to TS-ISODATA, and the algorithm is described as follows Algorithm 2.

For input raw data $X = \begin{bmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \dots & x_{mn} \end{bmatrix}$, n parameters, m data points of the fragment, normalized as

$$x_{ij} = \frac{x_{ij} - x_{j,min}}{x_{j,max} - x_{j,min}} - 0.5 \tag{2}$$

calculate its statistics as

$$\bar{x}_i = \frac{1}{m} \sum_{j=1}^m x_{ij}, s_i = \frac{1}{m} \sqrt{\sum_{j=1}^m (x_{ij} - \bar{x}_i)^2} \quad (i = 1, 2, \dots, n) \tag{3}$$

Algorithm 2 TS-ISODATA Algorithm.

Input: X , trend sequence O ;

1. Normalized processing x_{ij} ;
2. Statistics calculation \bar{x}_i, s_i ;
3. Construct feature vectors $y = (\bar{x}_1, s_1, \bar{x}_2, s_2, \dots, \bar{x}_n, s_n)^T$;
4. Randomly select k_0 samples as initial clustering centers $C = \{c_1, c_2, \dots, c_{k_0}\}$;
5. Calculate the distance from each sample x_i to the cluster center of the k_0 cluster centers and assign it to the class with the min distance;
6. Determine whether the number of elements in each class above is less than N_{min} . If so, discard the class, make $k = k - 1$, and reassign the samples to the class with the min distance;

7. For each category c_i , recalculate the clustering centers $c_i = \frac{1}{|c_i|} \sum_{x \in c_i} x$;
 8. If the current $k \leq \frac{1}{2}k_0$, split operation;
 9. If the current $k \geq 2k_0$, merge operation;
 10. Terminate if the maximum number of iterations is reached; otherwise, go back to 2.
- Output: Clustering results

3. Flight Maneuver Recognition

Algorithm 2 assigns the fragment to a specific class without knowing which flight maneuvers it is. In this section, the algorithm will use phase reconstruction to reconstruct the feature space and identify specific classes of flight maneuvers based on the principle that different maneuvers have different approximate entropy.

3.1. Phase Space Reconstruction

Due to the superiority of PSR (phase space reconstruction) for chaotic time-series computation [43], this paper adopts a multivariate data fusion reconstruction method based on Bayesian estimation theory, and the main calculation steps are as follows.

3.1.1. Reconstruction Parameters

The phase space reconstruction technique has two key parameters: the dimension of the embedding m and the delay time τ , which are determined here using the C-C method.

1. Define the correlation integral corresponding to each point y of the embedded time series in the reconstructed phase space as in Equation (4).

$$C(m, N, r, t) = \frac{2}{M(M-1)} \sum_{1 \leq i \leq j \leq m} \theta(r - d_{ij}) \tag{4}$$

$$d_{ij} = \|Y_i - Y_j\|_\infty, \theta(z) = \begin{cases} 0, & z < 0 \\ 1, & z > 0 \end{cases} \tag{5}$$

where Y_i is the reconstructed phase space vector, M is the number of vectors $M = N - (m - 1)\tau$, m is the embedding dimension, N is the number of points of the original time series, t is time, and $\theta(z)$ is the associative integral, a cumulative distribution function that expresses the probability that the distance between any two points in the phase space is less than the radius r . Here, the distance between points is expressed as an infinite number of parameters of the difference of vectors.

2. Split the given time series into t equidistant and disjoint subsequences as Equation (6), where t is the reconstruction time delay.

$$x^1 = \{x_1, x_{t+1}, \dots, x_{N-t+1}\}, x^2 = \{x_1, x_{t+2}, \dots, x_{N-t+2}\}, \dots, x^t = \{x_1, x_{2t}, \dots, x_N\} \tag{6}$$

3. Calculate the original sequence's S_1 and each sequence's S_2 :

$$S_1(m, N, r, t) = C(m, N, r, t) - C^m(1, N, r, t) \tag{7}$$

$$S_2(m, r, t) = \frac{1}{t} [\tag{8}$$

4. Select the two radiuses r with the max and min values and define the increments ΔS_2 :

$$\Delta S_2(m, t) = \max\{S_2(m, r_j, t)\} - \min\{S_2(m, r_j, t)\} \tag{9}$$

5. Calculate the statistics:

$$\bar{S}_2(t) = \frac{1}{16} \sum_{m=2}^5 \sum_{j=1}^4 S_2(m, r_j, t) \tag{10}$$

$$\Delta\bar{S}(t) = \frac{1}{4} \sum_{m=2}^5 \Delta S(m, t) \tag{11}$$

$$S_{2cor}(t) = \Delta S_2(m, t) + |S_2(m, r, t)| \tag{12}$$

6. Take the value corresponding to the first zero point of $\bar{S}_2(t)$ or the first minimal value of $\Delta\bar{S}(t)$ as the optimal time delay τ .
7. Let the t corresponding to the global minimum of $S_{2cor}(t)$ be the length of the time series window and the embedding dimension m .

3.1.2. Fusion Phase

As previously stated, the single variable delay time is τ , and the embedding dimension is m . To ensure that the multivariate is fully expanded in the same phase space without distortion, each variable's $\tau = \min(\tau_i)$, and $m = \max(m_i)$, ($i = 1, 2, \dots, r$). Each reconstructed sequence expression X_i as in Equation (13).

$$X_i = \begin{bmatrix} x_{i,1} & x_{i,1+\tau} & \dots & x_{i,1+(m-1)\tau} \\ x_{i,2} & x_{i,2+\tau} & \dots & x_{i,2+(m-1)\tau} \\ \vdots & \vdots & \vdots & \vdots \\ x_{i,M} & x_{i,M+\tau} & \dots & x_{i,M+(m-1)\tau} \end{bmatrix}, i = 1, 2, \dots, r \tag{13}$$

Extract the r reconstructed sequences of the same position k out of phase points in Equation (13) to form the fusion set $D_k = [x_1, x_2, \dots, x_r]$. The specific expression is given in Equation (14).

$$\begin{cases} x_1 = [x_{1,k} & x_{1,k+\tau} & \dots & x_{1,k+(m-1)\tau}] \\ x_2 = [x_{2,k} & x_{2,k+\tau} & \dots & x_{2,k+(m-1)\tau}] \\ \vdots & \vdots & \vdots & \vdots \\ x_r = [x_{r,k} & x_{r,k+\tau} & \dots & x_{r,k+(m-1)\tau}] \end{cases} \tag{14}$$

Let the expression of the phase point at position k after fusion be z_k , and the optimal fusion phase point at k is obtained according to Equation (15).

$$p(z_k|x_1, x_2, \dots, x_r) = \frac{p(x_1, x_2, \dots, x_r|z_k)}{p(x_1, x_2, \dots, x_r)} \cdot p(z_k) \tag{15}$$

Let $p(z_k|x_1, x_2, \dots, x_r)$ obey a normal distribution with mean z and variance δ^2 . According to (16) and (17), the calculation gives (18).

$$\begin{cases} \frac{1}{\sigma^2} = \sum_{i=1}^r \frac{1}{\sigma_i^2} + \frac{1}{\sigma_0^2} \\ \frac{z}{\sigma^2} = \sum_{i=1}^r \frac{x_i}{\sigma_i^2} + \frac{z_0}{\sigma_0^2} \end{cases} \tag{16}$$

$$\gamma \exp\left[-\frac{1}{2} \sum_{i=1}^r \left(\frac{x_i - z_k}{\sigma_i}\right)^2 - \frac{1}{2} \left(\frac{z_k - z_0}{\sigma_0}\right)^2\right] = \frac{1}{\sqrt{2\pi\sigma}} \exp\left[-\frac{1}{2} \left(\frac{z_k - z}{\sigma}\right)^2\right] \tag{17}$$

$$z = \frac{\sum_{i=1}^r \frac{x_i}{\sigma_i^2} + \frac{z_0}{\sigma_0^2}}{\sum_{i=1}^r \frac{1}{\sigma_i^2} + \frac{1}{\sigma_0^2}} \tag{18}$$

The final Bayesian estimate of the optimal fusion phase point at position k is obtained as in Equation (19), where the upper and lower limits of ω are the maximum and

minimum values of the phase point, and the PSR can be completed after finding all M position phase points.

$$\hat{z}_k = \int_{\omega} z_k \frac{1}{\sqrt{2\pi\sigma}} \exp[-\frac{1}{2}(\frac{z_k - z}{\sigma})^2] dz_k, \forall k = 1, 2, \dots, M \tag{19}$$

3.2. Recursion Graphs and Approximate Entropy

Recursion graphs (RP) is an effective method for qualitative analysis of nonlinear dynamical systems, which can reveal the internal state evolution process of the system by using the image-change pattern. It is generally implemented using the Heaviside function. The black dots in the RP diagram indicate that the attractor trajectories reach the same region of the orbit at different moments and vice versa for the white dots.

Approximate entropy can quantitatively analyze the structural complexity of nonlinear systems [44] as defined in Equation (20). Different flight maneuvers generally have different complexity and have different approximate entropy. By calculating the approximate entropy and combining with the maneuvers entropy library, we can know which maneuver it is.

$$ApEn(m, r) = \lim_{N \rightarrow \infty} [\Phi^m(r) - \Phi^{m+1}(r)], \Phi^m(r) = \frac{1}{N - m + 1} \sum_{i=1}^{N-m+1} \ln C_i^m(r) \tag{20}$$

$$C_i^m(r) = \frac{1}{N - m} \{d[X_i, X_j] < r\} \tag{21}$$

$$d[X_i, X_j] = \max_{k=0,1,\dots,m-1} |x_{i+k} - x_{j+k}| \tag{22}$$

where $i = (1, 2, \dots, N - m + 1), j = (1, 2, \dots, N - m + 1),$ and $i \neq j.$

4. The FMR General Framework

The general idea of the generic framework proposed in this paper is to integrate the automatic segmentation capability of unsupervised clustering and the information entropy capability of distinguishing sequence complexity.

First, the original input data are processed using a dynamic clustering method such as ISODATA, and the algorithm outputs the segmented, unknown kinds of maneuver sequences. Second, the multivariate phase space reconstruction calculation is applied to establish the complete phase space of the dynamical system. Then, the recurrence map and approximate entropy are calculated in the new phase space to analyze the complexity of the sequences qualitatively and quantitatively. Finally, according to the principle that the complexity of different kinds of maneuver sequences is different, the specific kind of the sequence is determined based on the calculation results so as to complete the FMR. The specific flow chart is shown in Figure 1.

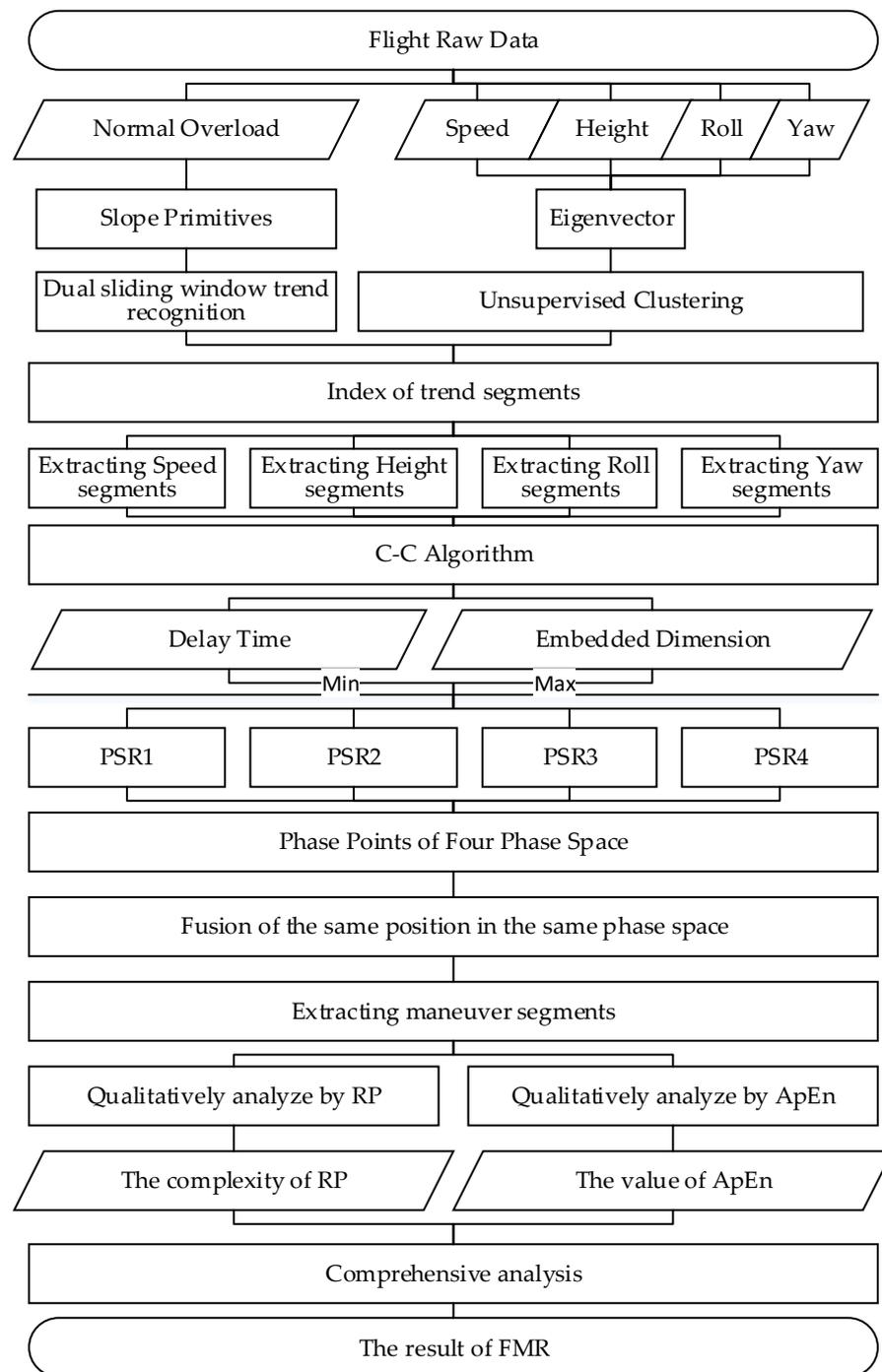


Figure 1. A general framework for FMR.

In the flow chart, the raw flight data are first preprocessed to extract some of the parameter columns. The speed, altitude, roll angle, and pitch angle form a parameter matrix, which is involved in the unsupervised clustering calculation. Using the double-window algorithm, the trend identification is completed by using the normal load as the slope primitive, and the trend is used to segment the whole raw sequence into subseries and output the index values. Based on the index values of subsequences in the previous step, parameter fragments are extracted for each of the four parameter sequences. The extracted four parameter fragments are fed into the C-C algorithm, and the phase space reconstruction is performed according to the calculated minimum delay time and maximum embedding dimension, respectively, and the phase points at the same position in these four spaces are fused to extract the action fragments. After a comprehensive

analysis of the qualitative values of the recurrence map and the quantitative values of the approximate entropy, the action recognition results are finally obtained.

5. Experiments

The experimental environment is Windows 10 operating system, Anaconda development environment, python language, and Matlab7.1 simulation platform. The visualization tool is the three-dimensional flight path recovery system (3D-FPRS) developed by the author's team. The system is based on the open-source CesiumJS platform and implemented using WebGL, HTML5 technology, which can reduce the input flight raw data into 3D dynamic visualization of flight trajectory.

The experimental raw data were obtained from CAFUC real flight training records: the aircraft type is C172R, file name 1 log_210721ZUUCK, 5724 lines; file name 2 log_210316ZUUU, 6445 lines. Due to the problem of data accuracy, 1104 rows of data in log_210721ZUUCK and 4626 rows of data in log_210316ZUUU were used during the experiment, with six columns of data in each row, totaling 34,380 pieces of data. The total length of the experimental sequence was 5730. The whole raw flight data are visualized as Figure 2.

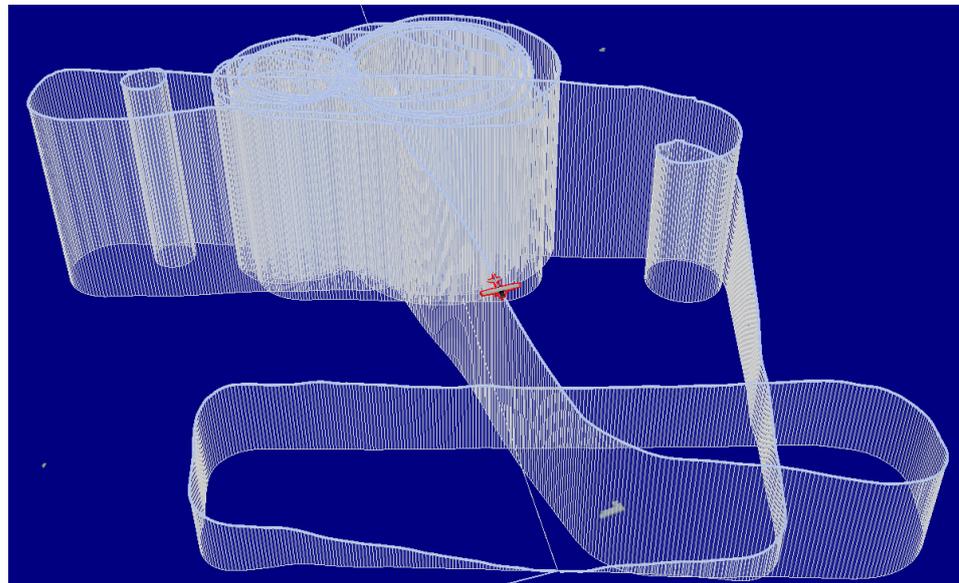


Figure 2. The visualization of whole raw flight data (file 1 and file 2).

The speed, sideslip angle, altitude, pitch angle, roll angle, and normal overload are considered as key data during the experiment, with other multi-column flight parameter data participating in the feature dataprocessing.

TS-ISODATA model has six clustering parameters, and $K = 7$, $L = 1$, and $I = 100$ were selected in the experiment, and the genetic algorithm was used to find $\theta_n, \theta_s, \theta_c$. The final optimal parameter setting values obtained are $\theta_n = 1$, $\theta_s = 0.0373$, and $\theta_c = 0.0043$, and the evaluation result using this set of parameter values is 6.3823. The input raw sequence is segmented into 96 maneuver segments. The segmentation calculation process takes an average of 76 s.

The index of the extracted motorized fragment for a particular experiment was (0, 54, 108, 162, 216, 270, 378, 432, 486, 540, 594, 648, 702, 756, 810, 864, 918, 972, 1026, 1080, 1134, 1188, 1242, 1296, 1350, 1404, 1458, 1512, 1566, 1674, 1728, 1782, 1836, 1890, 1944, 1998, 2052, 2160, 2214, 2268, 2322, 2376, 2430, 2484, 2538, 2592, 2646, 2700, 2754, 2808, 2862, 2916, 3024, 3078, 3132, 3186, 3240, 3294, 3348, 3402, 3456, 3510, 3564, 3618, 3672, 3726, 3780, 3834, 3888, 3942, 3996, 4050, 4104, 4158, 4212, 4266, 4320, 4374, 4428, 4482, 4536, 4590, 4644, 4698, 4752,

4806, 4860, 4914, 4968, 5022, 5076, 5130, 5184, 5238, 5292, 5346, 5368), where each index number represents the specific moment when the file was imported,

The feature vector extracted according to this was (1:{-0.131, 0.108, 0.203, 0.090, -0.039, 0.057, 0.176, 0.099, 0.203, 0.101, 0.472, 0.017},2:{0.018, 0.096, -0.042, 0.091, 0.034, 0.029, 0.074, 0.050, -0.094, 0.109, 0.164, 0.055},...96:{0.018, 0.096, -0.042, 0.091, 0.034, 0.029, 0.074, 0.050, -0.094, 0.109, 0.164, 0.055}). The result of clustering is shown in Table 1. The clustering calculation process takes an average of 121 s.

Table 1. TS-ISODATA clustering results.

Categories	Corresponding Maneuver Segments
1	8,9,16,18,23,24,31,34,35,42,45,48,52,55,60,63,67,74,77,82,19,28,37,40,49,58,69,79
2	1,87,92,94
3	3,5,12,14,15,20,27,33,36,39,41,51,54,57,62,81,84,88,95,96,11,25,38,43,65,68,71,72,
4	4,6,22,53,50,66,73,89,91,2,7,10,13,17,21,26,29,30,44,47,32,46,56,59,61,64,70,75,83,85,76,78,80,86,90,93

After preliminary expert analysis, category 4 are all transitional-level flights between complex maneuvers, which are not significant, so this paper uses PSR method to study the recurrence graph and approximate entropy of category 1, 2, and 3. The ApEn results are given in Table 2.

Table 2. The ApEn value for categories 1–3.

Categories	1	2	3	4	5	6	7	8	9	10	Average
1	0.3937	0.3148	0.4985	0.3512	0.2881	0.4112	0.3309	0.2490	0.3989	0.4117	0.3648
2	0.3166	0.3594	0.4252	0.0870	0.2491	0.0408	0.0870	0.0741	0.1295	0.1922	0.1961
3	0.1346	0.0941	0.2007	0.1178	0.0953	0.1457	0.1178	0.0953	0.1419	0.0344	0.1177

Experimental results show that the dataset as a multivariate time series does fit the chaotic nonlinear dynamical system characteristics. Similar maneuvers show similar characteristics on the recurrence graph, with close values of approximate entropy (ApEn), while different maneuvers vary widely. Thus, the phase space reconstruction recognition method based on approximate entropy can distinguish the recognition of flight maneuvers, especially complex maneuvers.

Three samples of the trace recovery visualization and recurrence map experiment are given in Figures 3–5.

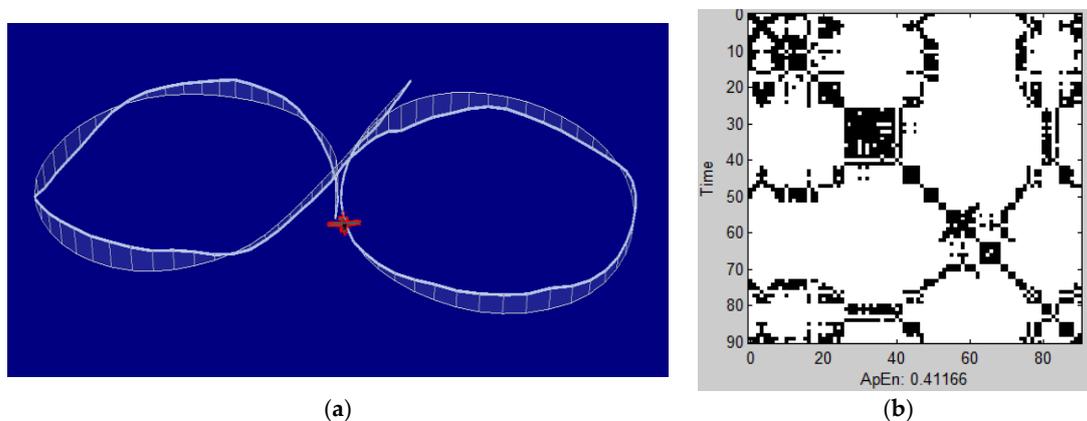


Figure 3. (a) Category 1 maneuver visualization reduction; (b) the maneuver’s RP and ApEn value.

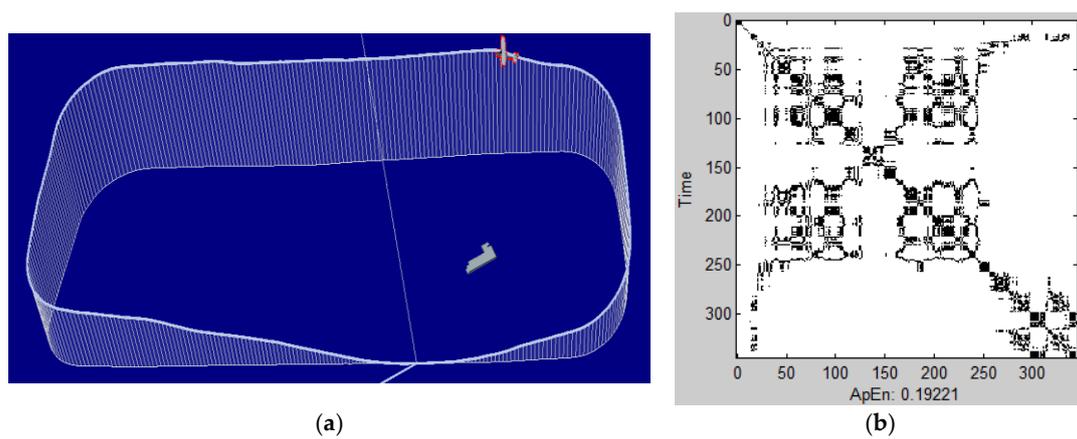


Figure 4. (a) Category 2 maneuver visualization reduction; (b) the maneuver's RP and ApEn value.

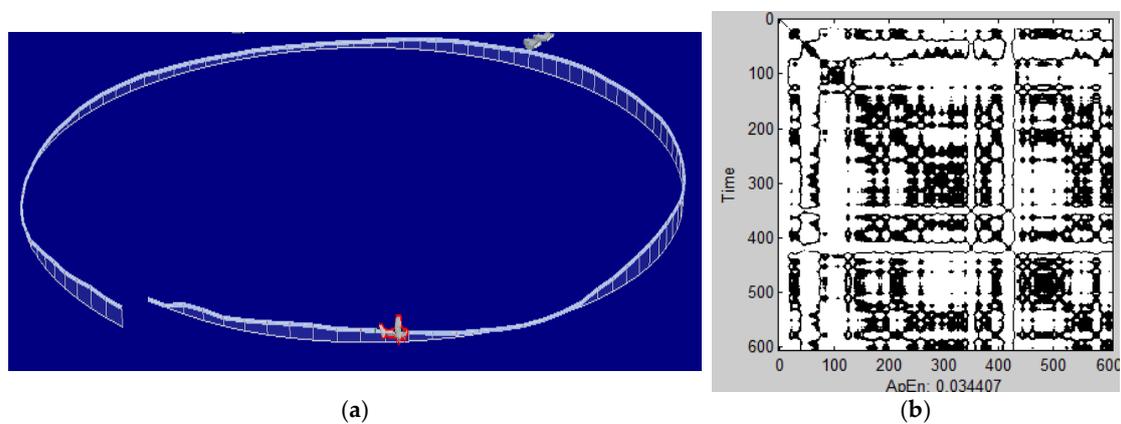


Figure 5. (a) Category 3 maneuver visualization reduction; (b) the maneuver's RP and ApEn value.

In order to study what exactly these three categories of maneuvers are, this paper uses the recurrence diagrams of these three categories of maneuvers in conjunction with the visual flight path recovery system to be able to clearly distinguish the categories of maneuvers. As shown in Figures 3a, 4a, and 5a, category 1 is Eight maneuver, category 2 is RectangularCourse maneuver, and category 3 is Spin maneuver. Not only do these three categories of maneuvers have different ApEn value, but their RPs also have significant differences, which perfectly match the complexity level given by flight experts as shown in Figures 3b, 4b, and 5b. The RP and ApEn calculation process takes an average of 88 s.

The overall average time of the whole framework automatic FMR calculation process is 285 s with 5730 raw input data.

In order to verify that the framework can be applied to multiple aircraft types, we selected two other datasets to complete the validation experiments, which are also from CAFUC real flight training records: aircraft types SR20 and DA42, file 3 name log_210521_ZHCC (13,750 lines), and file 4 name log_210531_ZUUU (13,018 lines); the raw flight data are visualized as Figure 6.

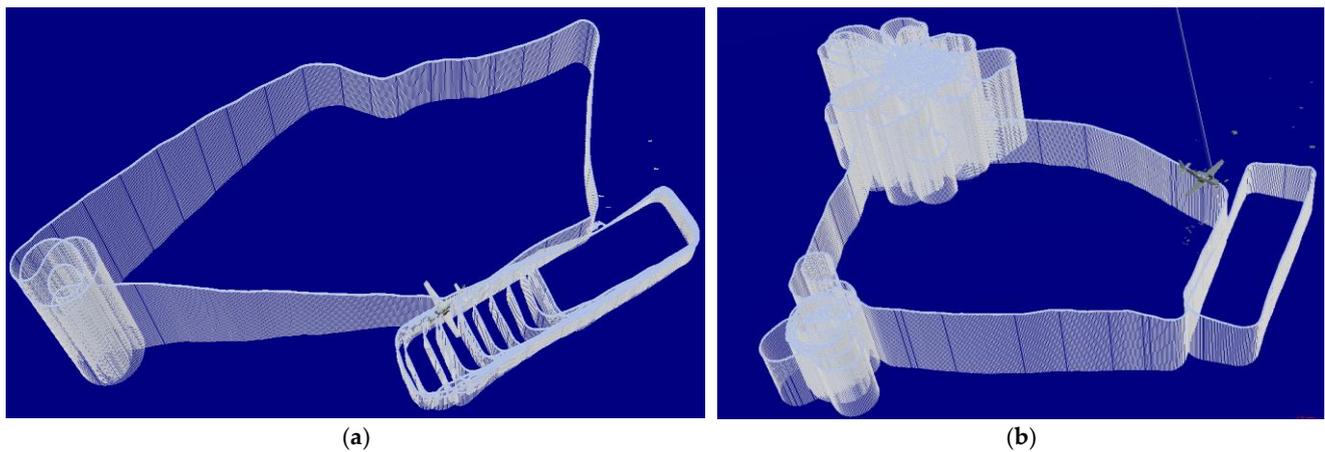


Figure 6. (a) File 1 visualization; (b) file 2 visualization.

In addition, in order to do comparison experiments, the project team developed an expert validation aid tool (EVAT) as shown in Figure 7.

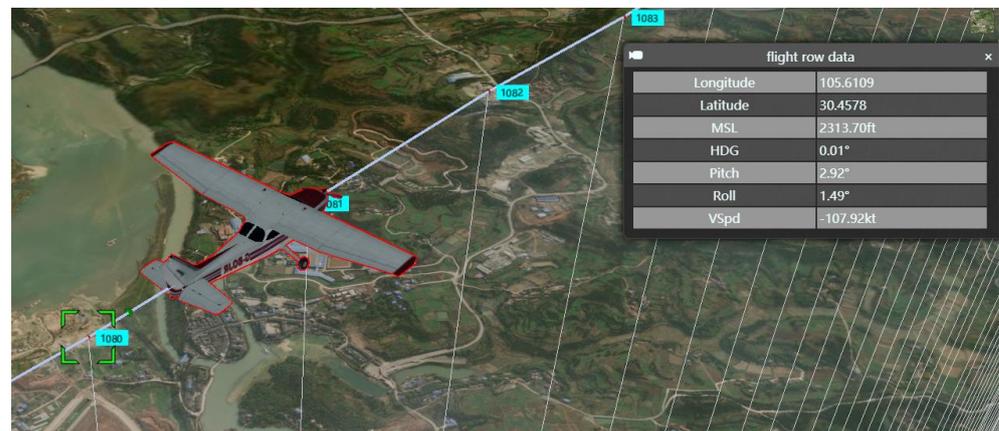


Figure 7. Expert validation aid tool (EVAT).

The system is also based on the CesiumJS platform, which can not only be reduced to 3D dynamic visualization of flight trajectory but also can display each second of temporal parameters and mark them in sequence, helping flight experts to judge flight movements with the naked eye.

With this tool, three flight experts made flight maneuvers judgments on the above two experimental files and two validation files, frame by frame, respectively, and the complete comparison results are shown in Table 3 below.

Table 3. The overall experimental results.

Categories	File 1 + File 2 C172, 5730 Lines			File 3 SR20, 13750 Lines			File 4 DA32, 13018 Lines			Average
	FMR	Expert	Accuracy (%)	FMR	Expert	Accuracy (%)	FMR	Expert	Accuracy (%)	Accuracy
1	28	26	92.3	6	5	80	49	50	98.2	90.2
2	4	4	100	20	24	83.3	7	5	60	81.1
3	28	26	92.3	10	8	75	15	17	88.2	85.2
Time (seconds)			Ratio			Ratio			Ratio	Ratio
	285	15,675	55	382	17,569	45.9	349	21,638	62	54.3

The experimental results in Table 3 show that the method in this paper can perform FMR for three types of aircraft and different file lengths, with the highest accuracy rate for category 1 (Eight maneuver), and the lowest accuracy rate for category 2

(RectangularCourse maneuver), with an overall average accuracy rate of 85.5%. The reason why the Eight maneuver accuracy is the highest is because the maneuver is significantly different from others, and the RectangularCourse maneuver accuracy is the lowest because the maneuver is generally time-consuming, which is accompanied by a half-spin maneuver, and the number of such maneuvers is small, so the recognition is not effective.

In terms of time consumption, the consumption time is related to the document length, and overall, the recognition speed of this paper is 54.3 times faster than human flight experts.

The comparison experiments were difficult to design and implement because none of the other papers disclosed the datasets used, and some of the papers corresponded to aircraft types that were fighter jets or UAVs, which differed greatly from the temporal nature of this paper; neither did any of the other methods cover temporal segmentation and automatic recognition. However, we still completed the recognition experiments using the same datasets provided in this paper, files 1 and 2, and the experimental results are shown in the following Table 4.

Table 4. The comparison experiments results.

Methods	DTW [4]	RF-SVM [10]	Expert System [15]	DBM [29]	CNN-LSTM [33]	Ours
Accuracy (%)	79.6	61	89.6	77	71	85.5
Time (Seconds)	314	276	656	295	489	285

From the results, we can see that under the same flight-type condition, the accuracy of this paper's method is second only to the expert system, and the speed is second only to SVM, which is better than other methods in the comprehensive evaluation. More importantly, if we want to follow the aircraft model, except for this paper, all other methods have to retrain the model or redesign the knowledge rules, which does not have generalization ability in practical application scenarios.

6. Conclusions

In this paper, a general framework was constructed for the first time for automated FMR based on dynamic clustering and phase space reconstruction. The framework decomposes the FMR task into two parts, which are automatic maneuver sequence segmentation and automatic maneuver class identification. The automatic maneuver sequence segmentation was implemented by the improved dynamic clustering method TS-ISODATA, which solves the problem of self-organized iterative clustering of multivariate time series and successfully segments the input data into multiple segments and automatically clusters them into four classes. Due to the chaotic nature of the flight dynamics system, the automatic recognition of maneuver categories partially reconstructs the phase space of multivariate fusion, transforms the representational dimensional change patterns of flight maneuvers that are difficult to organize into attractive subsequences that are easy to identify, and generates recursive graphs from them to calculate ApEn values that can characterize the complexity of maneuvers. With the help of a visual 3D flight-track reduction system, the flight maneuver categories are easily identified. With an input sequence of 5000 s, the entire framework computation process takes an average of 285 s, which is 54 times faster than human expert recognition, with an overall accuracy rate of 85.5%.

In the next step, the entropy corresponding to different flight actions can be solidified so as to form an automatic identification library for fast and automatic classification output. This step requires collecting a large number of samples of a particular flight maneuver and deriving a reasonable range of approximate entropy values through a large number of experiments, and the range of values among the maneuvers should not overlap to avoid duality. According to different entropy value ranges corresponding to different

aircraft maneuvers categories, automatic identification rules were established to realize the final automatic output of flight maneuvers. For maneuvers with close approximate entropy values and little difference, the complexity of the recurrence graph should be considered, and the difference enhancement of information entropy should be designed to further strengthen the difference between maneuvers. In addition, as a pattern-recognition category, although the method in this paper has better generalization ability and does not require pre-training, it is computationally intensive and time-consuming and cannot realize online real-time recognition. At this stage, it can only be used for post-flight analysis to support the next application, such as flight technology scoring based on a specific flight maneuver and post-accident investigation after a flight accident. In the future, the principle of the method can be explored in depth to simplify the computation process.

Author Contributions: Conceptualization, J.L.; methodology, J.L.; software, J.L. and H.C.; formal analysis, J.L.; investigation, J.L. and H.C.; resources, J.L., H.C. and R.J.; data curation, J.L.; writing—original draft preparation, J.L.; writing—review and editing, J.L. and R.J.; visualization, J.L. and H.C.; supervision, J.L.; project administration, J.L.; funding acquisition, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Civil Aviation Flight Technology and Flight Safety Key Laboratory of China research projects, NO: FZ2020ZZ02; The CAFUC Research Project, NO: CJ2021-01; National Natural Science Foundation of China, NO: U2033213; Sichuan Science and Technology Program, NO: 2022YFG0027.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank Xiaodong Liu and graduate students Zhe Cao and Lu Pang for their contributions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. The United States Department of Transportation; Federal Aviation Administration; Airman Testing Standards Branch. *Pilot's Handbook of Aeronautical Knowledge*; The United States Department of Transportation: Oklahoma City, AK, USA, 2016.
2. Burgin, G.H.; Owens, A.J. *An Adaptive Maneuvering Logic Computer Program for the Simulation of One-to-One Air-to-Air Combat*. NASA Contractor Report (CR), Volume 2: Program Description; NASA: San Diego, CA, USA, 1975.
3. Burgin, G.H. Improvements to the Adaptive Maneuvering Logic Program. 1986. Available online: <https://ntrs.nasa.gov/404?original=%2Fcitations%2F19880002266> (accessed on 2 March 2022).
4. Barndt, G.; Sarkar, S.; Miller, C. Maneuver regime recognition development and verification for H-60 structural monitoring. In Proceedings of the Annual Forum Proceedings-American Helicopter Society, Virginia Beach, VA, USA, 1–3 May 2007; American Helicopter Society, Inc.: Fairfax, VA, USA, 2007; Volume 63, p. 317.
5. Zhang, J.Y. *Time Series Analysis Method of Flight Data and Its Application*; National Defense Industry Press: Beijing, China, 2013.
6. Li, Z.-X.; Zhang, F.-M.; Li, K.-W. A multivariate time series indexing structure supporting DTW distance. *J. Softw.* **2014**, *25*, 560–575.
7. Li, H.; Shan, Z.; Guo, H. MDTW-based flight action recognition algorithm. *Comput. Eng. Appl.* **2015**, *51*, 267–270.
8. Shen, Y.; Ni, S.; Zhang, P. A similar subsequence query method for flight data. *J. Air Force Eng. Univ.* **2019**, *20*, 7–12.
9. Tang, C.; Dong, J. LSH-based time subsequence query algorithm. *J. Comput. Sci.* **2012**, *35*, 2228–2236.
10. Qu, J.; Lv, M.; Yang, Y.. Flight Motion Recognition Method Based on Multivariate Phase Space Reconstruction and Approximate Entropy. In Proceedings of the 2021 40th Chinese Control Conference (CCC), Shanghai, China, 26–28 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 7247–7253.
11. Mao, H.; Zhang, F.; Feng, H. Research on flight maneuver evaluation method based on singular value decomposition. *Comput. Eng. Appl.* **2008**, *44*, 240–242.
12. Mao, H.; Zhang, F.; Feng, H. Similar pattern query for multivariate flight data. *Comput. Eng. Appl.* **2011**, *47*, 151–155.
13. Yang, J.; Duan, C.; Xie, S. Fuzzy least squares support vector machine based aircraft flight maneuver recognition. *J. Ballist. Arrow Guid.* **2004**, *S6*, 395–398.
14. Jia, Z.; Fan, X.; Xue, M.. Online identification method for tactical maneuvers of enemy aircraft based on maneuver elements. *J. Beijing Univ. Technol.* **2018**, *38*, 820–827.

15. Kendrick, J.D.; Maybeck, P.S.; Reid, J.G. Estimation of aircraft target motion using orientation measurements. *IEEE Trans. Aerosp. Electron. Syst.* **1981**, *2*, 254–260.
16. Kou, Y.; Jiang, L.; Wang, D. High-order reconstruction of the decision process of close air combat maneuver. *J. Syst. Simul.* **2019**, *31*, 2085.
17. Molkenthin, J. Determination and Verification of Operational Maneuver Parameters and Time Histories. *Loads Requir. Mil. Aircr.* **1997**, *6*, 3–11.
18. Wang, Y.; Dong, J.; Liu, X.. Identification and standardization of maneuvers based upon operational flight data. *Chin. J. Aeronaut.* **2015**, *28*, 133–140.
19. Xie, C.; Ni, S.; Zhang, Z.. A knowledge-based method for fast recognition of aerobatic maneuvers. *Comput. Eng.* **2004**, *30*, 116–118.
20. Ni, S.; Shi, Z.; Xie, C.. Establishment of a knowledge base for maneuvering flight maneuvers recognition of military warplanes. *Comput. Simul.* **2005**, *22*, 23–26. <https://doi.org/10.3969/j.issn.1006-9348.2005.04.007>.
21. Travert, J.H. Flight Regime and Maneuver Recognition for Complex Maneuvers. Master's Theses, Embry-Riddle Aeronautical University, Daytona Beach, FL, USA, 2009.
22. Chunmei, G.; Lili, Z.; Yu, B.. Recognition of Flight Operation Action Based on Expert System Inference Engine. In Proceedings of the 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 24–25 August 2019; IEEE: Piscataway, NJ, USA, 2019; Volume 1, pp. 17–20.
23. Tian, H.; Xie, S.; Wang, L. Flight trajectory identification based on rough set theory. *Firepower Command Control* **2015**, *40*, 29–33.
24. Zhou, D.Y.; Li, F. Genetic algorithm-based tactical flight maneuver decision for aircraft. *J. Northwestern Polytech. Univ.* **2002**, *20*, 109–112.
25. Wei, Z.; Ding, D.; Zhou, H.. A flight maneuver recognition method based on multi-strategy affine canonical time warping. *Appl. Soft Comput.* **2020**, *95*, 106527.
26. Wang, L.; Huang, C.Q.; Wei, Z.L. Automatic extraction of flight action rules based on SSA algorithm. *Comput. Eng. Appl.* **2019**, *14*, 203–208.
27. Wang, Y.W.; Gao, Y. A flight action recognition rule extraction method based on whale optimization algorithm. *J. Nav. Aviat. Eng. Coll.* **2019**, *33*, 447–451.
28. He, D.; Wu, S.; Bechhoefer, E. Development of Regime Recognition Tools for Usage Monitoring. In Proceedings of the 2007 IEEE Aerospace Conference, Big Sky, MT, USA, 3–10 March 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 1–11.
29. He, D.; Wu, S.; Bechhoefer, E. A regime recognition algorithm for helicopter usage monitoring. *Aerosp. Technol. Adv.* **2010**, 391–404. Available online: https://books.google.co.jp/books?hl=en&lr=&id=bQWQDwAAQBAJ&oi=fnd&pg=PA391&dq=A+regime+recognition+algorithm+for+helicopter+usage+monitoring.+&ots=4Kgn_R61wO&sig=I6atHzxhI1AC8YqD9Tw1cNBI_ZQ&redir_esc=y#v=onepage&q=A%20regime%20recognition%20algorithm%20for%20helicopter%20usage%20monitoring.&f=false (accessed on 2 March 2022).
30. Rajniecek, R.E. Application of Kalman Filtering to Real-Time Flight Regime Recognition Algorithms in a Helicopter Health and Usage Monitoring System. Master's Theses, Embry-Riddle Aeronautical University, Daytona Beach, FL, USA, 2008.
31. Meng, G.-L.; Zhang, H.-M.; Park, H.-Y. Maneuver recognition of warplanes in automated flight training evaluation. *J. Beijing Univ. Aeronaut. Astronaut.* **2020**, *46*, 1267–1274.
32. Li, Y.F.; Ni, S.H.; Zhang, Z.L. A fuzzy Kohonen network-based intelligent processing method for flight data. *Syst. Eng. Electron. Technol.* **2002**, *24*, 53–55.
33. Xu, W. A fuzzy neural network-based approach for shipboard aircraft landing maneuvers recognition. *Appl. Sci. Technol.* **2013**, *2*, 26–29.
34. Hanyang, F.; Hongming, F.; Ruiyuan, G. Research on air target maneuver recognition based on LSTM network. In Proceedings of the 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAL), Shanghai, China, 12–14 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 6–10.
35. Fang, W.; Wang, Y.; Yan, W.J.; Gong, Y. Flight action recognition based on differential ideas and convolutional neural networks. *J. Chin. Acad. Electron. Sci.* **2021**, *16*, 347–353.
36. Fang, W.; Wang, Y.; Yan, W.; Lin, C. Symbolic flight action recognition based on neural networks. *Syst. Eng. Electron.* **2021**, *1*, 13.
37. Zhang, Y.Y.; Wang, Y.Y.; Wang, C.H.X. Analysis of parametric correlation and temporal features for flight action recognition method. *Comput. Eng. Appl.* **2016**, *52*, 246–249.
38. Shen, Y.; Ni, S.; Zhang, P. A Bayesian network-based approach for flight action recognition. *Comput. Eng. Appl.* **2017**, *53*, 161–167.
39. Zhang, X.; Yin, Z.; Liu, F.; Huang, Q. Data mining method for aircraft maneuvering division. *J. Northwestern Polytech. Univ.* **2016**, *34*, 33–40.
40. Zhang, L. A non-supervised automatic method of aircraft maneuver partition. *J. Comput. Methods Sci. Eng.* **2021**, *21*, 383–395.
41. Kang, Z.; Shang, J.; Feng, Y.. A deep sequence-to-sequence method for accurate long landing prediction based on flight data. *IET Intell. Transp. Syst.* **2021**, *15*, 1028–1042.
42. Li, X.; Shang, J.; Zheng, L.. CurveCluster+: Curve Clustering for Hard Landing Pattern Recognition and Risk Evaluation Based on Flight Data. *IEEE Trans. Intell. Transp. Syst.* **2021**. <https://doi.org/10.1109/TITS.2021.3117846>

-
43. Tou, J.T.; Gonzalez, R.C. *Pattern Recognition Principles*; Addison-Wesley Publishing Company: Reading, MA, USA, 1974.
 44. Pincus, S.M. Approximate entropy as a measure of system complexity. *Proc. Natl. Acad. Sci. USA* **1991**, *88*, 2297–2301.