Neural Computing and Applications (2020) 32:16857–16874
https://doi.org/10.1007/s00521-018-03975-z

SMART DATA AGGREGATION INSPIRED PARADIGM & APPROACHES IN IOT APPLNS

Intelligent active fault-tolerant system for multi-source integrated navigation system based on deep neural network

Chengjun Guo[1,2] · Feng Li[1] · Zhong Tian[1] · Wei Guo[2] · Shusen Tan[1]

Received: 9 October 2018 / Accepted: 20 December 2018 / Published online: 5 January 2019

Abstract

This paper proposes an intelligent active fault-tolerant system based on deep neural network. That is, an active fault-tolerant integrated navigation system is established by adding neural network to the fault-tolerant integrated navigation system based on one-class support vector machine fault detection algorithm. When there is no fault, the neural network trains each sub-filter; when there is a fault, the neural network which has been in the training state will predict the fault time data and use the neural network prediction data to replace the fault data into the main filter for fusion. It can be seen from the simulation analysis that the system can detect the fault of the navigation sub-filtering system well, and when the fault occurs, the prediction data of the neural network is used for information fusion. Simulation results show that the system can provide stable and reliable navigation under the condition of time-varying system and observation noise and complex environment.

Keywords Active fault-tolerant · One-class SVM · Fault detection · Deep neural network · State estimation

1 Introduction

With the development of technology, the requirements for reliability, accuracy, and robustness of multi-sensor navigation systems are getting higher and higher in the recent years [1], and the navigation systems must be developed toward integration and being fault-tolerant, that is, to develop various integrated navigation systems and fault-tolerant navigation systems based on inertial navigation system [2]. For fault-tolerant integrated navigation, there are many subsystems inside. Once a subsystem fails, the information fusion feedback may affect each subsystem, resulting in inaccurate or unusable navigation and positioning information. Therefore, in order to improve the reliability and accuracy of fault-tolerant navigation systems, fault detection and diagnosis are essential.

The common fault detection methods include Chi-square detection methods, autonomous integrity monitored extrapolation (AIME), optimal fault detection (OFD), multiple solution separation (MSS) and so on. The Chi-square test method is simulated and validated in paper [3], but this method is based on model, and the accurate real-time system model, measurement model, and noise model are always difficult to get, which will affect the detection effect of failure. The AIME and OFD are used for fault detection, respectively, in paper [4] and paper [5]. However, due to the error tracking effect of integrated navigation system filter, these two methods have low sensitivity to faults and large fault delays. In recent years, the rise of artificial intelligence has provided a new method for fault diagnosis. In [6, 7], BP neural network is used for fault detection. However, the convergence speed of neural network is slow and there is a problem of local optimization, and the training samples of neural network needs are large.

In multi-source integrated navigation systems, accurate and effective state assessment methods and theories are

✉ Chengjun Guo
johnsonguo@uestc.edu.cn

[1] Research Institute of Electronic Science and Technology, University of Electronic Science and Technology of China, Chengdu 611731, China

[2] National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China

playing an increasingly important role. When using the INS/GPS integrated navigation system for precise positioning, the traditional classical Kalman filtering method has been very maturely applied in integrated navigation [8, 9] and can achieve good results in its state estimation. The volume filtering proposed in paper [10] solves the nonlinear and non-Gaussian problems of the system. Now, besides Kalman filtering and other traditional methods for state estimation, neural network is also proposed for state estimation. For example, in paper [11], the state estimation of integrated navigation system using Elman neural network is proposed. The paper [12] proposes the state estimation of integrated navigation using BP neural network.

As an intelligent deep learning model, deep neural network (DNN) has been widely used in pattern recognition and accurate classification [13, 14]. According to the navigation characteristics of inertia, terrain and geomagnetism in the environment that multi-source integrated navigation system needs to face, this paper proposes a new navigation state estimation by combining Laplace feature map (LE) with deep neural network model. It is to perform stable navigation state estimation when the model is not accurate enough and the system and observation noise are time-varying and the surrounding environment is complex. Then, a system fault diagnosis method based on one-class support vector machine is proposed. The simulation results show that the proposed method can detect the abrupt and slow change faults of the navigation sub-filtering system with short delay, strong real-time performance and analysis. Compared with the comparison, it shows that the method has better detection performance in small samples. Based on the above, this paper proposes an active fault-tolerant integrated navigation system, in which the neural network is added to the fault-tolerant integrated navigation system based on one-class SVM fault detection algorithm.

2 Problem description and the proposed method

2.1 Intelligent active fault-tolerant strategy and scheme design for multi-source integrated navigation system

Figure 1 shows the block diagram of intelligent active fault-tolerant integrated navigation. This navigation system is based on the Federal Kalman structure and includes sub-filters and main filters. The inertial navigation system (INS) is used as a common reference system, and the sub-filters are composed of GPS, synthetic aperture radar (SAR), and terrain-aided navigation (TAN), respectively [15–17]. Each sub-filter works independently and is input to the main filter after one-class SVM fault diagnosis. At the same time, INS and other three sub-navigation systems will be input into their respective Laplacian eigenmaps deep neural network (LE-DNN) neural networks in addition to the input into the sub-filter for training [18].

In order to improve the adaptive ability of active fault-tolerant integrated navigation, the information distribution coefficient is constructed by extracting the traces of the covariance matrix in real time, as shown in the following formula:

$$\beta_i = \frac{\mathrm{tr}(P_i)}{\mathrm{tr}(P_1) + \mathrm{tr}(P_2) + \cdots + \mathrm{tr}(P_n) + \mathrm{tr}(P_m)}. \qquad (1)$$

2.2 State estimation based on LE-DNN and fault detection based on one-class SVM

2.2.1 System state estimation based on LE-DNN

It is assumed that the system equation of the navigation system can be represented by the following formula:

$$X(k+1) = \ddot{O}(k+1,k)X(k) + G(k)w(k) \qquad (2)$$

$$Z(k) = H(k)X(k) + v(k) \qquad (3)$$

where $w(k)$ is the process noise of the system and $v(k)$ is the measurement noise of the system.

When using the model for simulation, we assume that the statistical properties of process noise A and measurement noise B are known. In the multi-source integrated navigation system mentioned in this paper, in practical applications, especially in complex environments, system noise, especially measurement noise, may be affected by many strong external conditions, such as temperature, humidity, electric waves, geomagnetic changes, etc. The actual statistical characteristics of the noise will be greatly different from the assumed characteristics. In this case, if the fixed statistical model is used to estimate the navigation state, the positioning accuracy will be greatly reduced, the accuracy of the system cannot be met, and even the filter may diverge in serious cases.

Under this application background, a variety of adaptive online modification parameter methods have been proposed, and even some people have proposed Kalman filtering method combined with fuzzy control theory. Although these methods can reduce the influence of the above situation on filtering to some extent, these methods have limited application range and are not suitable for strong disturbance environment. The effect is very limited in complex environments. The method in this paper is designed based on this consideration. The neural network not only has the ability to process information in parallel and store information in a distributed manner, but also has the advantages of self-organization, adaptability, and fault

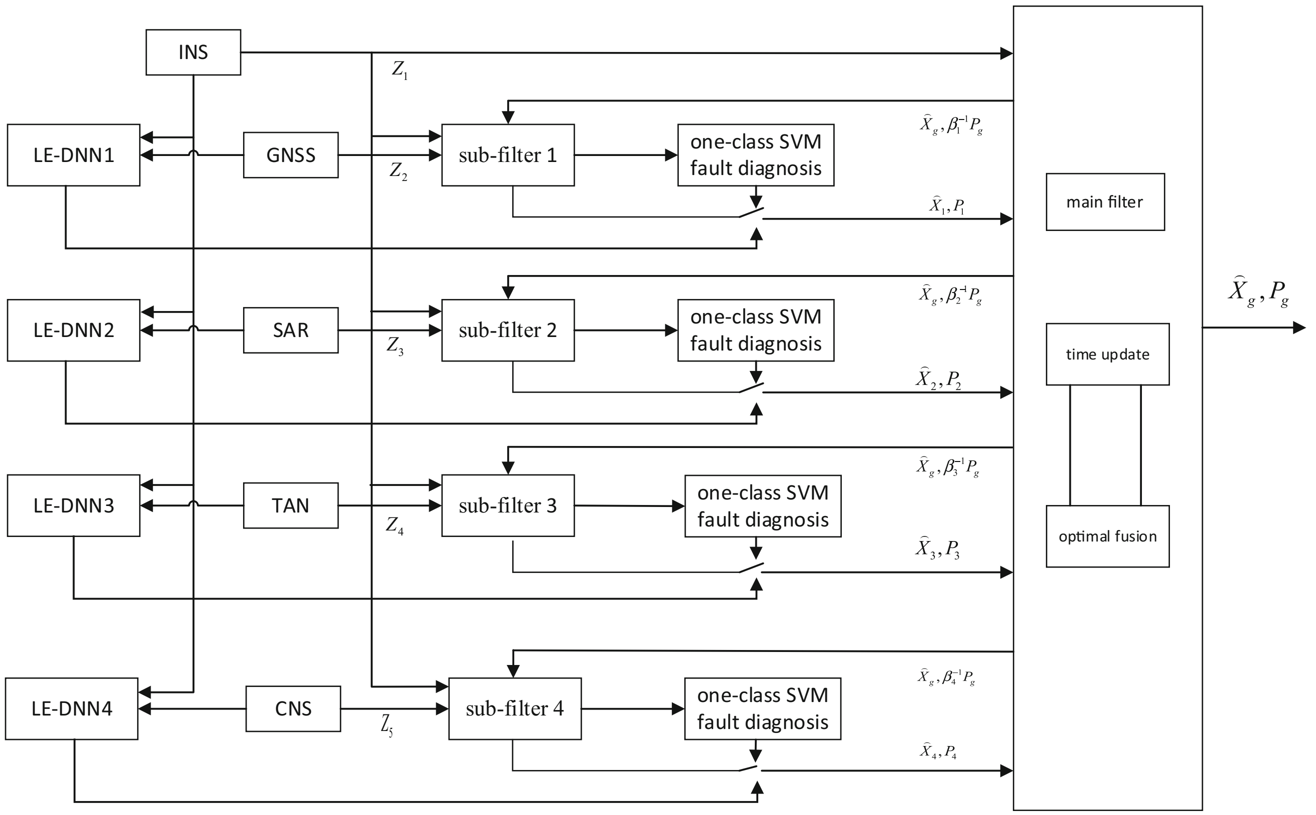


Fig. 1 Block diagram of intelligent active fault-tolerant integrated navigation system

tolerance. It also has strong learning and association functions for different environments. The state estimation method in this paper is based on this, and the deep neural network based on Laplacian feature map is used to estimate the state of the navigation system in a strong disturbance environment.

Step 1: Original features extraction

Data preprocessing, such as moving out the abnormal data form the original signal dataset, is an essential procedure before feature extraction, and more precise outcomes require more meticulous preprocessing. Since the original signal doped with much messy interference information is unsuitable for analyzing directly, it is a good choice to extract the features of the signal data for further analysis. Both classical and contemporary methods for feature extraction were employed in this paper.

1. Time and frequency analysis

Time and frequency domain feature analysis is one of the main methods of system state estimation. The time domain signal has the characteristics of containing large information, intuitive and easy to understand. It is the initial basis of system state estimation. The frequency domain characteristic parameters describe signal through the change in frequency band in signal spectrum and the dispersion of spectrum energy.

In this paper, 11 characteristic parameters in time domain and 13 characteristic parameters in time domain displayed in the literature are adopted [19].

2. WPT

When wavelet packet transform (WPT) is decomposing the low frequency part of the signal, it can also decompose the high-frequency part more meticulously at the same time, and there is neither redundancy nor omission in the decomposition. Therefore, WPT can provide better time–frequency analysis than wavelet transform for the measured value containing both medium- and high-frequency information. The steps to extract wavelet packet energy features mainly include [20]:

1. Extract signals in each sub-band

Recorded the wavelet function $W(k)$ and scaling function $\varphi(k)$ as $\mu_1 = W(k)$ and $\mu_0 = \varphi(k)$, respectively, then

$$\begin{cases} \mu_{2n}(t) = \sqrt{2} \sum\limits_{k \in Z} h(t) \mu_n(2t - k) \\ \mu_{2n+1}(t) = \sqrt{2} \sum\limits_{k \in Z} g(t) \mu_n(2t - k) \end{cases} \tag{4}$$

where $g_k = (-1)^k h_{1-k}$ is a biorthogonal filter, $n = 2l$ or $n = 2l+1$, $l = 0, 1, 2, \ldots$. The recursively defined function

μ_n denotes the wavelet packet defined by orthonormal scaling function $\mu_0 = \varphi(k)$.

2. Calculate the energy of each sub-band

Set the signal energy corresponding to the regenerated signal c_{jk} of the jth frequency band of the kth layer after the wavelet packet decomposition as E_{jk}, then

$$E_{jk} = \int \left|c_{jk}(t)\right|^2 \mathrm{d}t = \sum_{m=1}^{N} x_{jm}. \tag{5}$$

Here m is the discrete point of the regenerated signal c_{jk} of the jth frequency band of the kth layer, while x_{jm} stands for the amplitude of the discrete points of the regenerated signal c_{jk}.

3. Constructing wavelet packet feature vector

The feature vector of the wavelet packet can be obtained through normalizing the characteristic parameters calculated by the following formula:

$$e = \{E_{j0}, E_{j1}, \ldots, E_{jl}\}/E, \quad l = 2^j - 1 \tag{6}$$

where $E = \sum_{K=1}^{l} E_{jk}$ is the total energy of the whole signal that equals to the sum of the energy of each sub-band.

Step 2: LE feature space transformation

The sample data of high-dimensional spaces is actually in a low-dimensional manifold, of which the structure contains the geometric characteristics and the intrinsic dimensionality information of the original data [21]. The sample data in high-dimensional space (D dimension) actually can be projected into a low-dimensional manifold (L dimension, $L \leq D$) which can accurately reflect the geometric characteristics of the original data. As a nonlinear space dimensionality transformation technique, LE builds a graph from neighborhood information of the data set, and each data point serves as a node on the graph and connectivity between nodes is governed by the proximity of neighboring points [22], which can be generally represented as:

$$M^D \overset{\mathrm{LE}}{\Rightarrow} M^L, \ (L \leq D) \tag{7}$$

where M^D and M^L stand for the original features in D-dimensional space and projected features in L-dimensional space, respectively. The steps can be summarized as follows [23]:

(A) Constructing the graphs

Given k points $x_1, \ldots, x_k$ in M^D, construct a weighted graph with k nodes, one for each point and a set of edges connecting neighboring points to each other. For this purpose, put an edge between nodes i and j that are close. In this work, the n-nearest neighbors algorithm is adopted to find the nodes that are close to each other. In this method, nodes of i and j are connected by an edge if i is among n-nearest neighbors of node j.

(B) Choosing weights

The heat kernel algorithm described in previous section was introduced to calculate the weights of the edges in the constructed graph. If nodes i and j are connected, put

$$W_{i,j} = e^{-\frac{x_i - x_j^2}{4t}}. \tag{8}$$

(C) Eigenmaps

As for a constructed graph G, to obtain the connected components, we should compute the eigenvalues and eigenvectors for the generalized eigenvector problem as:

$$Ay = \delta By \tag{9}$$

where B is the diagonal weight matrix, of which the entries are columns sums of W, $B_{ii} = \sum_j W_{ji}$ and $A = B - W$ is the Laplacian matrix.

Therefore, an $N \times 18$ feature array composed of 18 original features extracted from the measured values is acquired in high-dimensional feature space. And before the high-dimensional feature array is projected to lower dimensional space by LE, maximum likelihood estimation (MLE) was adopted to calculate the intrinsic dimension of the array; then, an $n \times m$ $(m < 38)$ low-dimensional feature array was obtained.

Step 3: DNN construction and training

(A) Deep neural network

Hinton et al. [24] proposed a feasible scheme to construct deep structure neural network. The key point of this method is to use some restricted Boltzmann machines (RBM) to generate the pre-training without supervision and tack up these RBMs layer by layer to construct a DBN.

RBM is a probabilistic model that can be represented by a kind of undirected graph models. The undirected graph model has two layers, of which one is a visible layer used to describe the characteristics of the input data, while another is a hidden layer, and each layer is composed of a plurality of probability units. All the visible layer elements are connected with the random binary hidden layer elements by undirected weights; however, there is no connection between the elements in the same visible or hidden layer.

DBN is built through stacking a number of RBMs from bottom to top layer by layer, of which the rules are available in the literature [25]. Since the input features of this paper are continuous variables, the first two layers are built as Gaussian–Bernoulli RBM model, while other hidden layers are built as Bernoulli–Bernoulli RBM models. The output values of lower layer are used as inputs of the higher one between two binary RBM layers, through repeating which the network structure with desired hidden layer number can be obtained at last.

(B) Construct and fine-tune DNN

In this paper, a linear output layer is added at the top of the DBN to form deep neural network (DNN) that is used to study the mapping relationship between the measured value features and the system state information; the architecture of DNN is shown in Fig. 2. The input of DNN is the characteristic parameters of each navigation sensor measured value, and the output is a state estimation value of the system. Firstly, acquire the parameters of each layer with the help of the pre-training DBN and then initialize these parameters that correspond to each layer of DNN, so that we can get a deep network structure with almost optimal parameters. Finally, the error back propagation (BP) algorithm is adopted to fine-tune the parameters of the DNN network. The details of DNN training can be studied in the literature [26].

As for the DNN models, the quantities of input nodes, hidden nodes, hidden layers and output nodes are the most significant parameters. In this paper, the architectures of DNN models are defined as follows:

$$\mathrm{DNN}\left[\mathrm{param1}, \mathrm{param2}_1, \ldots \mathrm{param2}_j, \mathrm{param3}\right] \tag{10}$$

where param1 represents the number of input nodes, $\mathrm{param2}_i$ denotes the number of hidden nodes of ith hidden layer, while param3 stands for the number of output nodes.

To obtain a DNN network which can be used for system state estimation, set the first 2500 min data of the projected dataset M^L as N^L, of which all the data are collected from the sensors that is under normal conditions, then N^L was adopted for training and fine-tuning to obtain a normal DNN model, that is

$$N^L = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \end{bmatrix} = \begin{bmatrix} p_1^1 & \cdots & p_1^L \\ \vdots & \ddots & \vdots \\ p_N^1 & \cdots & p_N^L \end{bmatrix}. \tag{11}$$

Step 4: Assessment

Deep neural network that contains more neuron nodes and a number of hidden layers has better ability of expression than the shallow network, besides, the most important advantage of DNN is that it can express numerous function sets in a more compact and concise way, which make it very suitable for DNN to obtain the essential characteristics of massive data. Hence the entire dataset M^L composed of feature data of the measured values collected under normal condition as well as abnormal condition is used as the testing data that would be input into DNN model to estimate the system state, that is testing data:

$$M^L = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_M \end{bmatrix} = \begin{bmatrix} p_1^1 & \cdots & p_1^L \\ \vdots & \ddots & \vdots \\ p_M^1 & \cdots & p_M^L \end{bmatrix} \tag{12}$$

where $N^L \in M^L$.

Figure 3 exhibits the main procedures of the proposed method for state estimation of navigation systems by integrating LE into deep neural network.

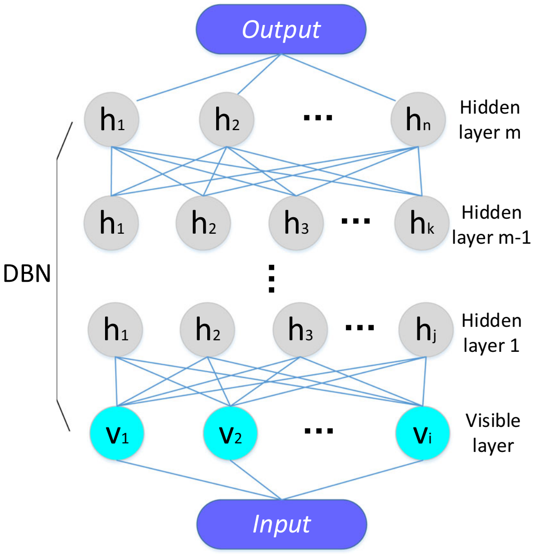


Fig. 2 Structure diagram of DNN

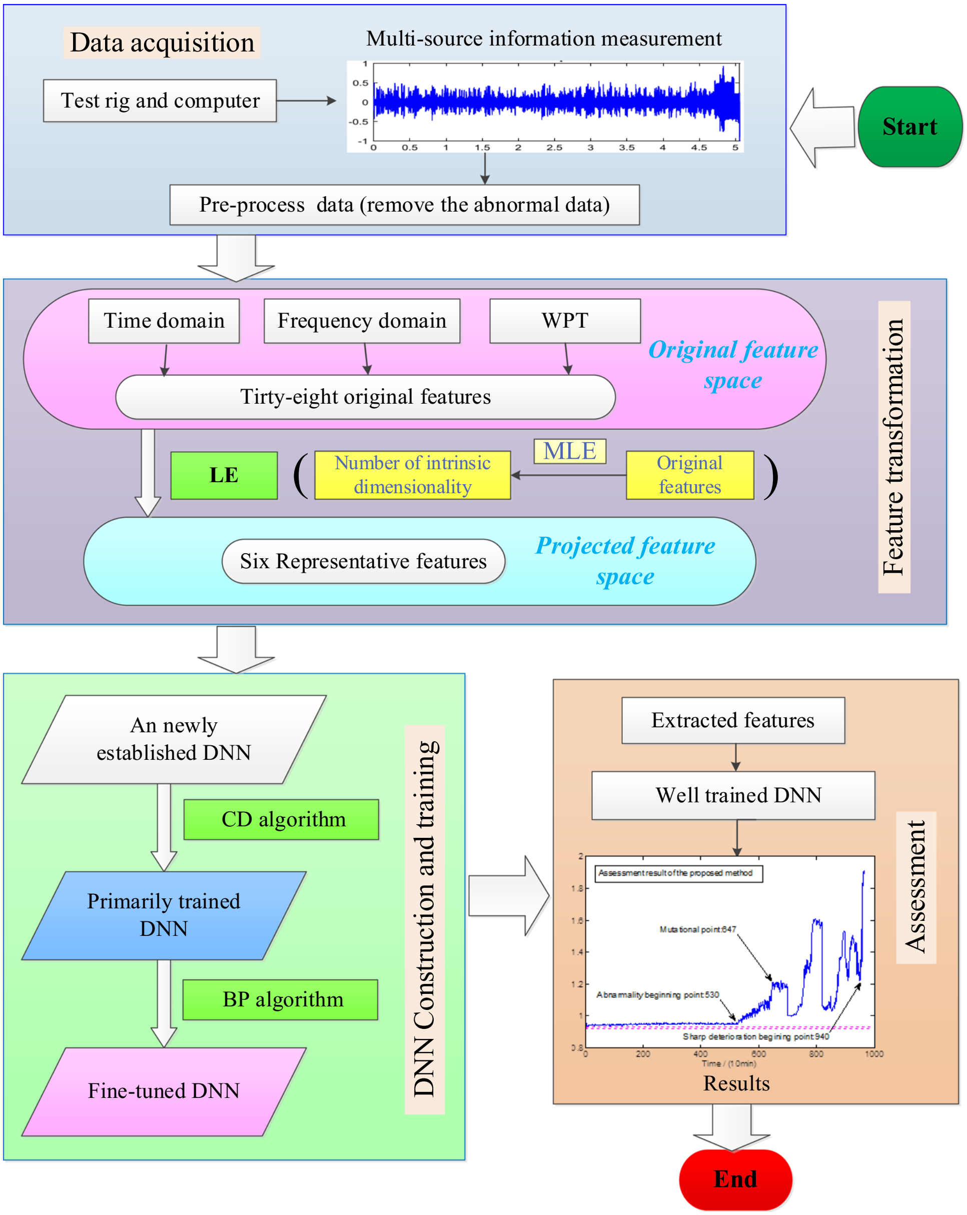
2.2.2 System fault detection based on one-class SVM

It can be known from the block diagram of intelligent active fault-tolerant integrated navigation system that for any sub-filter system, the navigation information should be detected before enter the main filter. While the subsystem is running normally, one-class SVMs are trained on normal data. When the system fails, the already trained SVM will automatically detect the fault data and isolate it to avoid entering the main filter. To improve the accuracy and real-time performance of the fault-tolerant integrated navigation system, aiming at the timing signal of navigation system, the training samples need to be preprocessed: phase space reconstruction and support vector preselection to improve modeling efficiency. The preprocessing steps are as follows: phase space reconstruction and support vector preselection.

1. Phase space reconstruction

One-class SVM has a faster speed for system modeling, which can be directly used for system data samples with

Fig. 3 The main steps of the proposed method



several dimensional features, but for time series navigation signals, the need for phase space reconstruction [16] and then establish fault forecast model using one-class SVM model [17]. The phase space reconstruction method is a method that reconstructs the attractor based on the limited measured data to study the dynamic behavior of the system. The evolution of any component in the system is determined by other components interacting with it. Therefore, the information of these related components is hidden in the development of any component, so that it can be extracted from a set of time series data of a component and restore the original rules of the system; this kind of law is a kind of trajectory of high-dimensional space, and this rule is a kind of track in high-dimensional space. The equivalent state space can be reconstructed by taking the time delay values of some fixed points as the new dimensional coordinates in a certain multidimensional state space. Repeating this process and measuring the amount of delay relative to different time, we can generate many such points. It can keep many properties of attractors, that is, using a system's observation, we can reconstruct the motivity system model. The time sequence from a navigation subsystem is $x(t)(t = 1, 2, \ldots, N)$, the dimension of the phase space is m, and the fixed time delay is called the embedding delay τ. The point of the phase space is

$$X(t) = (x(t), x(t-\tau), \ldots, x(t-(m-1)\tau))^T \tag{13}$$

where $t = 1, 2, \ldots, N_P, N_P = N - (m-1)\tau$.

2. Support vector preselection

One-class SVM has the sparsity, that is, the final form is only determined by the support vector, so the complexity of the algorithm can be reduced by reducing the training sample set, which can effectively improve the efficiency of fault detection and improve the real-time performance of the fault-tolerant integrated navigation system. Sample set $D = \left\{x_i \in R^d | i = 1, 2, \ldots, l\right\}$, the sample set in the center of the high-dimensional feature space

$$m = \frac{1}{l} \sum_{i=1}^{l} \varphi(x_i) \tag{14}$$

$$\cos(m, x) = \frac{\left(\varphi(x) \cdot \frac{1}{l} \sum_{i=1}^{l} \varphi(x_i)\right)}{\|\varphi(x)\| \left\| \varphi\left(\frac{1}{l} \sum_{i=1}^{l} \varphi(x_i)\right) \right\|}. \tag{15}$$

In the high-dimensional space, the angle between the vector x and the vector m formed by the center point is $\theta(m, x)$. For the sample point x, the larger the $\theta(m, x)$ is, the smaller the $\cos(m, x)$ and the smaller the $\|\varphi(x)\|$, the more likely x becomes the support vector. Calculate s_i for each sample point x in the sample set D, and select a smaller sample point s_i from the sample D to form a new training sample set.

$$s_i = \left(\varphi(x) \cdot \sum_{j=1}^{l} \varphi(x_i)\right) = \frac{1}{l} \sum_{j=1}^{l} k(x_i, x_j). \tag{16}$$

3. Fault detection algorithm

In order to improve the success rate of fault detection and ensure the real-time performance of the fault-tolerant integrated navigation system, the one-class SVM model uses fourfold cross-validation. The kernel function uses radial basis function RBF

$$k(x_i, x_j) = e^{-\left\|x_i - x_j\right\|^2 / 2\sigma^2}. \tag{17}$$

Select the statistical detection amount is

$$F(x) = \sum_{i \in I_{SV}} \alpha_i k(x_i, x) - \rho. \tag{18}$$

Ideally, the fault can be judged according to whether $F(x)$ is less than 0 or not. In the actual situation, however, a threshold J need to be set, and when the $F(x)$ is greater than J, the test point is normal; when the $F(x)$ is less than J, the test point is a fault point.

3 Simulation and analysis

3.1 Simulation and analysis based on LE-DNN state estimation

Figures 4, 5 and 6 show the position, velocity and attitude errors of the state estimation of the single LE-DNN neural network, respectively. Figures 7, 8 and 9 show the comparison of LE-DNN neural network estimation error and particle filter estimation error, including the mean error of the multi-group particle filter.

The comparison of the root mean square error (RMSE) of the system state estimation is shown in Table 1.

As can be seen from the figures below, the estimation accuracy of the LE-DNN network filter is higher than that of the single filter and higher than the average estimation of all the filters. However, when selecting LE-DNN network for training, only one noise value can be obtained if the object is a static object with zero speed; therefore, this paper takes the average of the estimated values of the multi-filter, so in this LE-DNN network training, it is not simply to use the true value of zero as the prior data. At this time, the velocity and position of the output sample contrast deviate from the true value. It can be found that the former will have a much greater relativity, because the latitude and precision values are much larger than the measurement noise. Of course, the neural network of training will have some influence on the accuracy of estimation.

3.2 Simulation and analysis of fault detection algorithm based on one-class SVM

In order to verify the validity of the proposed fault detection algorithm, the proposed fault detection algorithm is verified by MATLAB simulation. For the fault-tolerant integrated navigation system, the redundancy is large and the fault detection methods of each subsystem are similar. Therefore, this paper verifies the validity of the one-class SVM algorithm by detecting faults in GNSS and INS subsystems.

The accuracy of the inertial navigation device is as follows: the standard deviation of the gyroscope is 0.0058 rad/s, the static and dynamic drifts are 0.0524 rad/s and 0.00012 rad/s, respectively; the standard deviation of the accelerometer is 0.333 rad/s, and the static and dynamic drifts are 0.49050 m/s^2 and 0.002 m/s^2, respectively. The error of GNSS level measurement is 5 m, and the height measurement error is 10 m [27]. The total simulated flight time is 480 s, and the output frequency of GNSS and INS is 5 Hz. Figure 10 is a simulation of the motion trajectory.

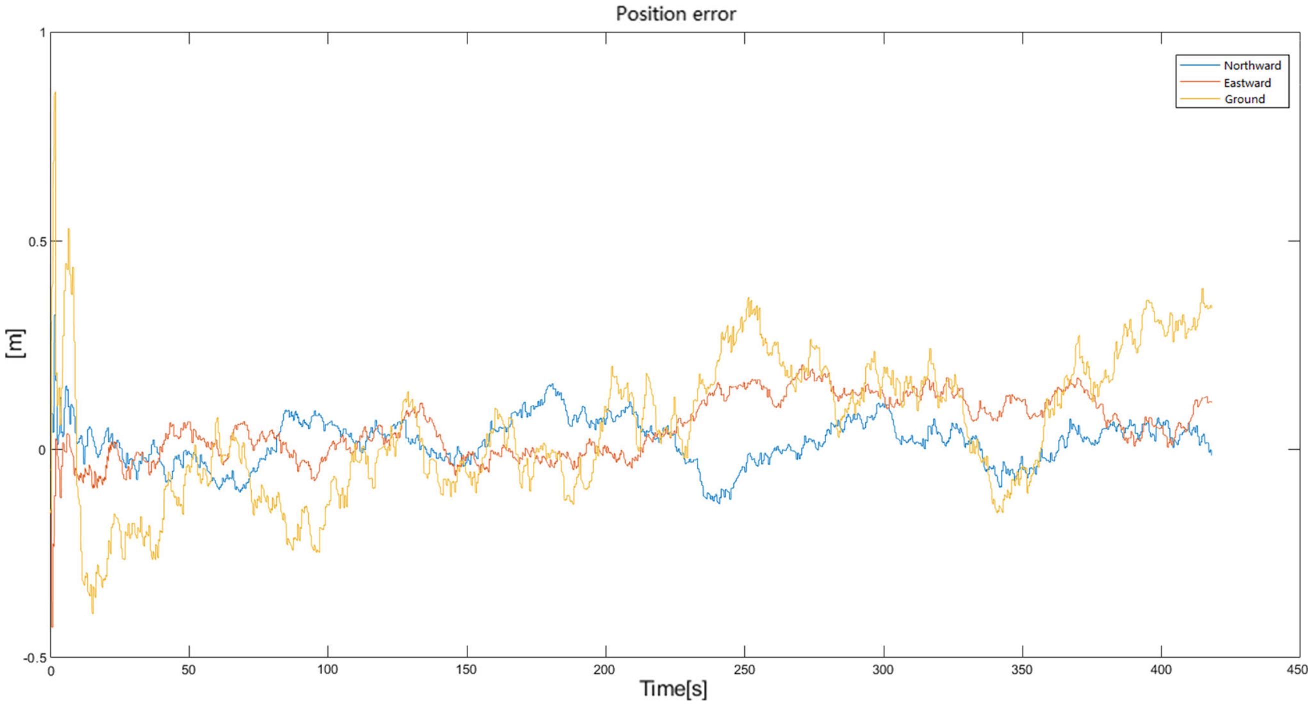


Fig. 4 LE-DNN position estimation error of multi-source integrated navigation system

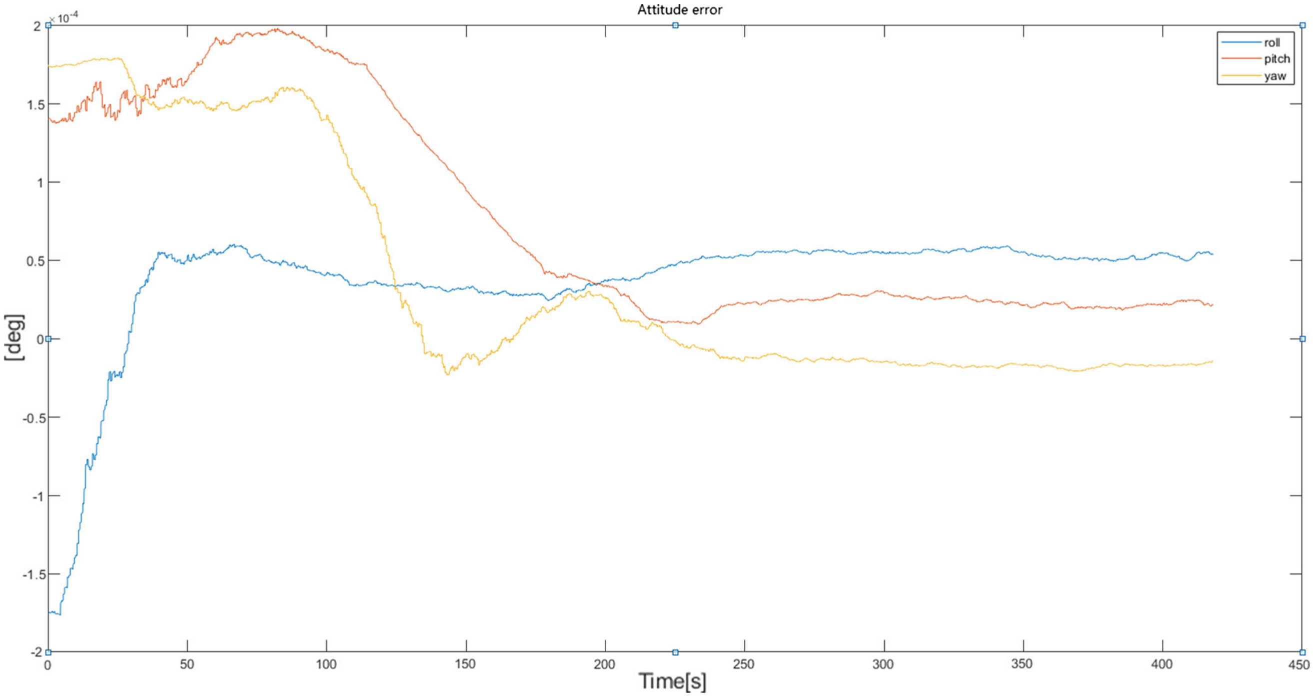


Fig. 5 LE-DNN attitude estimation error of multi-source integrated navigation system

In this simulation, the real-time fault detection is carried out for the abrupt and slow change fault. In order to reduce the impact of initial alignment on fault detection, the data in the middle stage of the trajectory are selected as training samples and test samples. The *Y*-axis error of gyroscope with 2 radians amplitude is added to twentieth sample points. The *X*-axis error of accelerometer with 2 $\mathrm{m/s^2}$ amplitude is added to sixtieth sample points, and the threshold J is set to -0.05 based on empirical value. The simulation results are shown in Fig. 11.

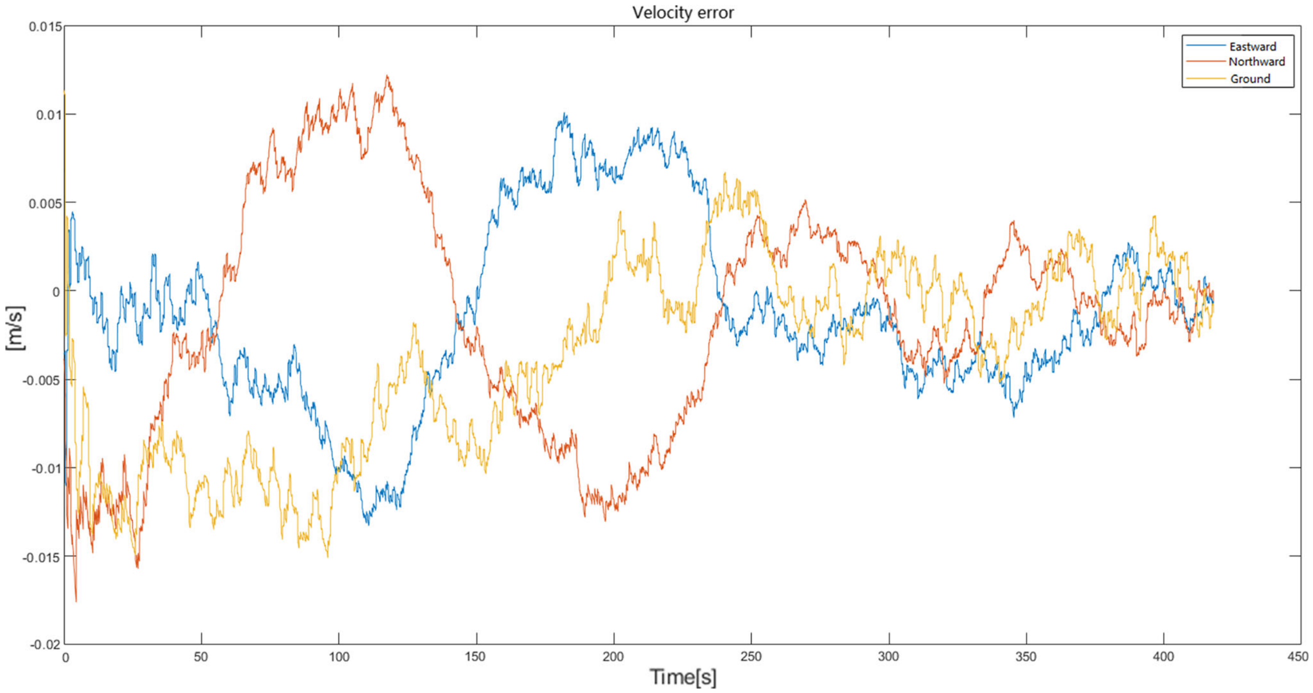


Fig. 6 LE-DNN speed estimation error of multi-source integrated navigation system

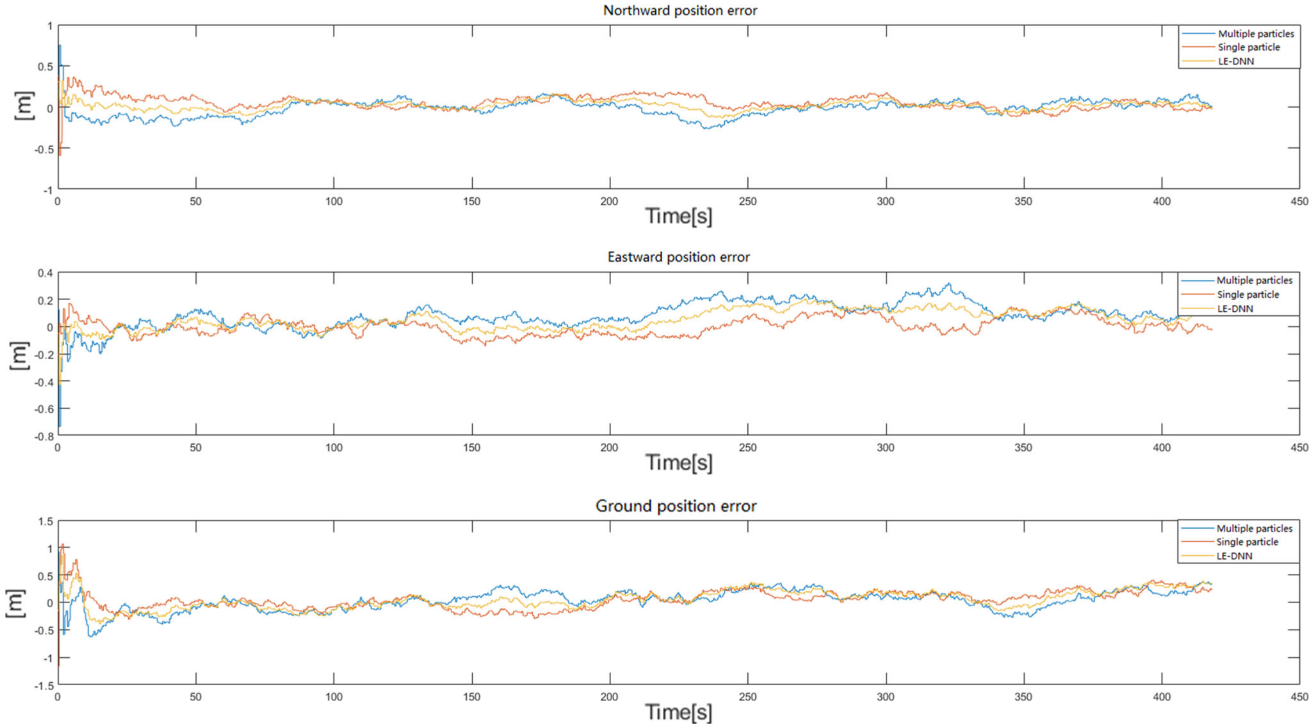


Fig. 7 Comparison of LE-DNN position estimation error of multi-source integrated navigation system

As can be seen from Fig. 11, the algorithm proposed in this paper has high sensitivity to abrupt fault and has little delay. The slow change fault is an accelerometer error that increases the amplitude to the 31st sample points to the 40th sample points. As shown in Fig. 11. It can be seen from the figure that the algorithm can also detect the slow change fault (Fig. 12).

In order to verify that the one-class SVM-based fault detection algorithm is superior to the neural network-based fault detection algorithm in training sample numbers, the

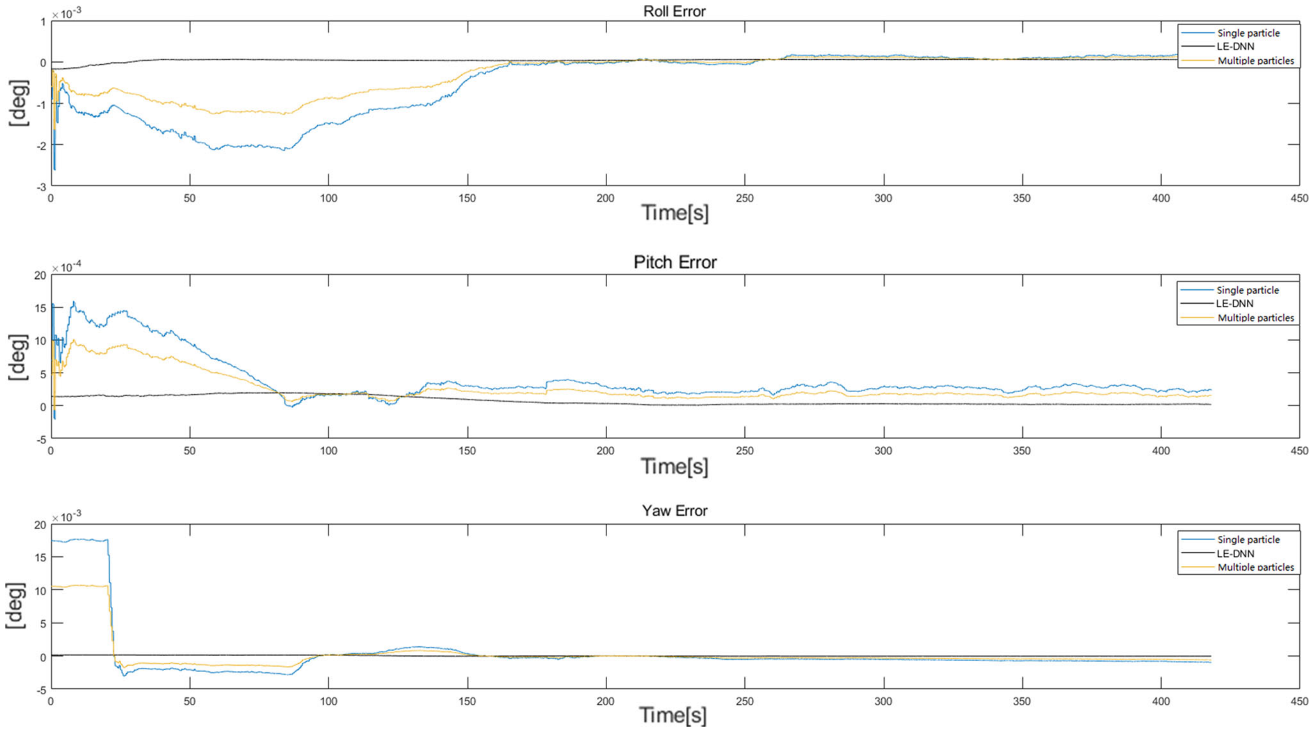


Fig. 8 Comparison of attitude estimation error of multi-source integrated navigation system

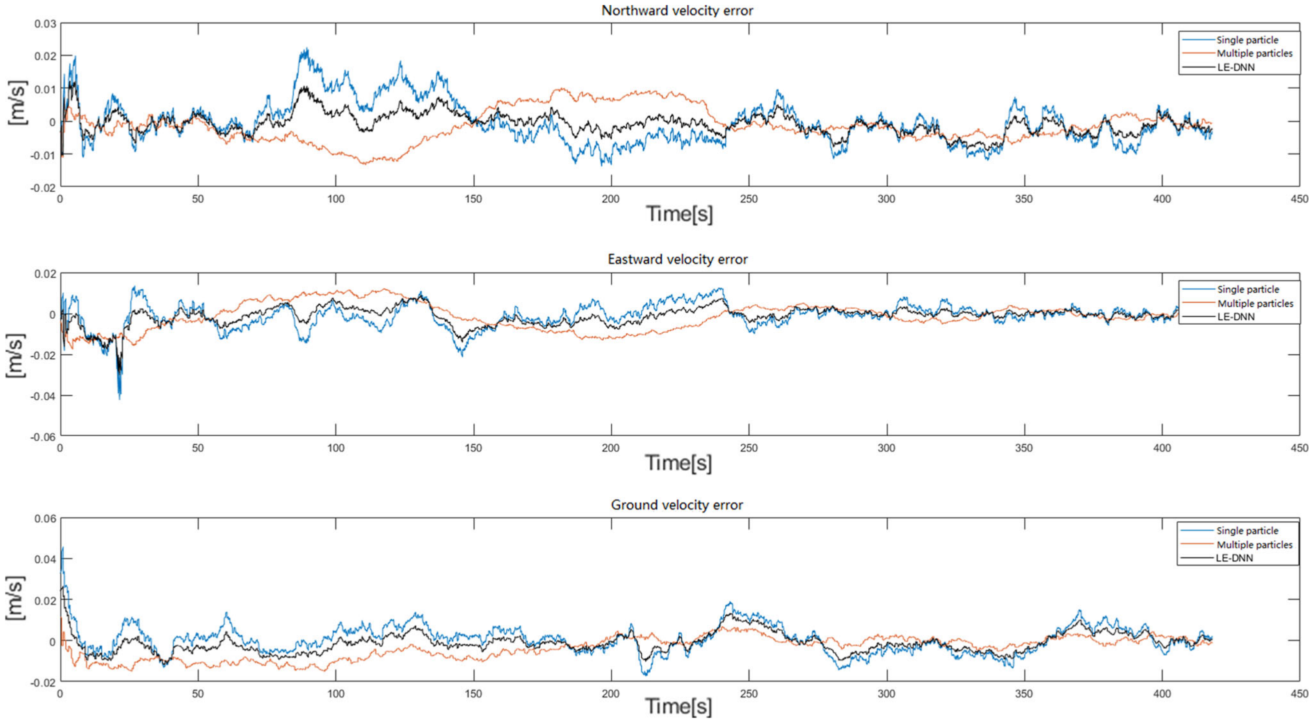


Fig. 9 Comparison of speed estimation error of multi-source integrated navigation system

Table 1 Comparison of the root mean square error (RMSE) of state estimation

	Position error (m)	Speed error (m/s)	Attitude error ($\times 10^{-3}$ deg)
Multiple particle filter average	0.2038	0.0152	0.049
Single particle filter	0.1153	0.0086	0.064
LE-DNN	0.0692	0.0049	0.001

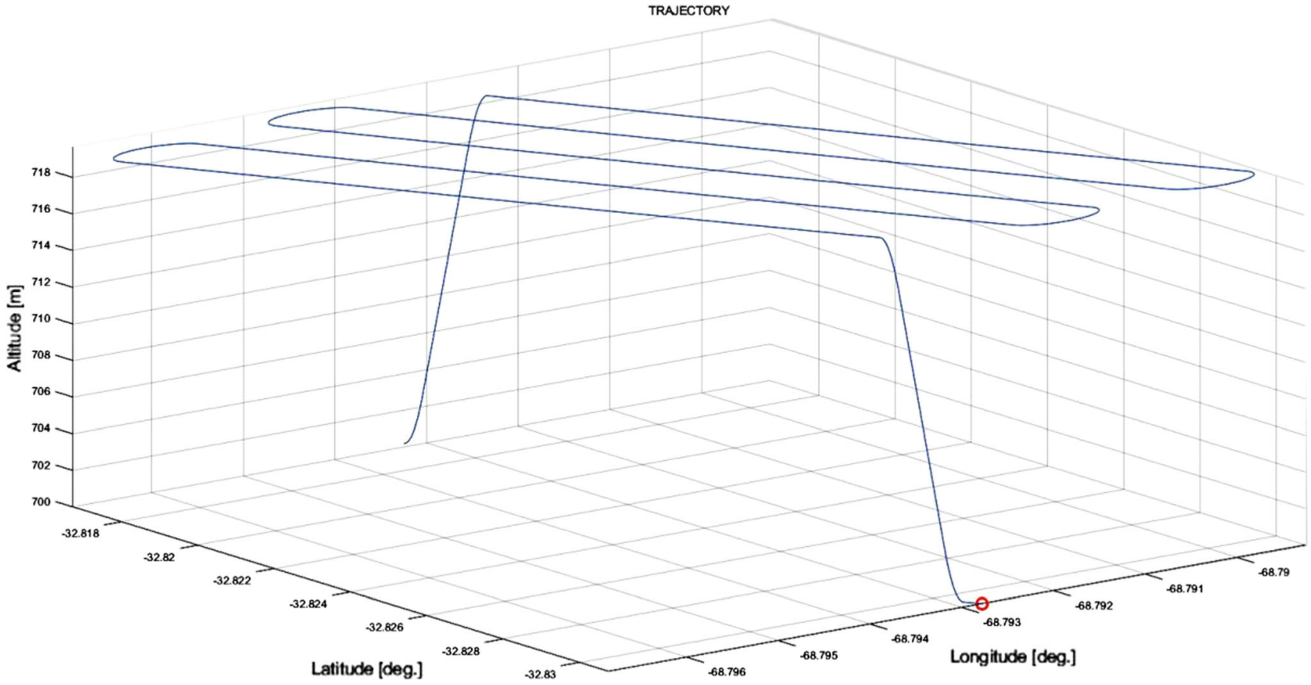


Fig. 10 Motion trajectory diagram

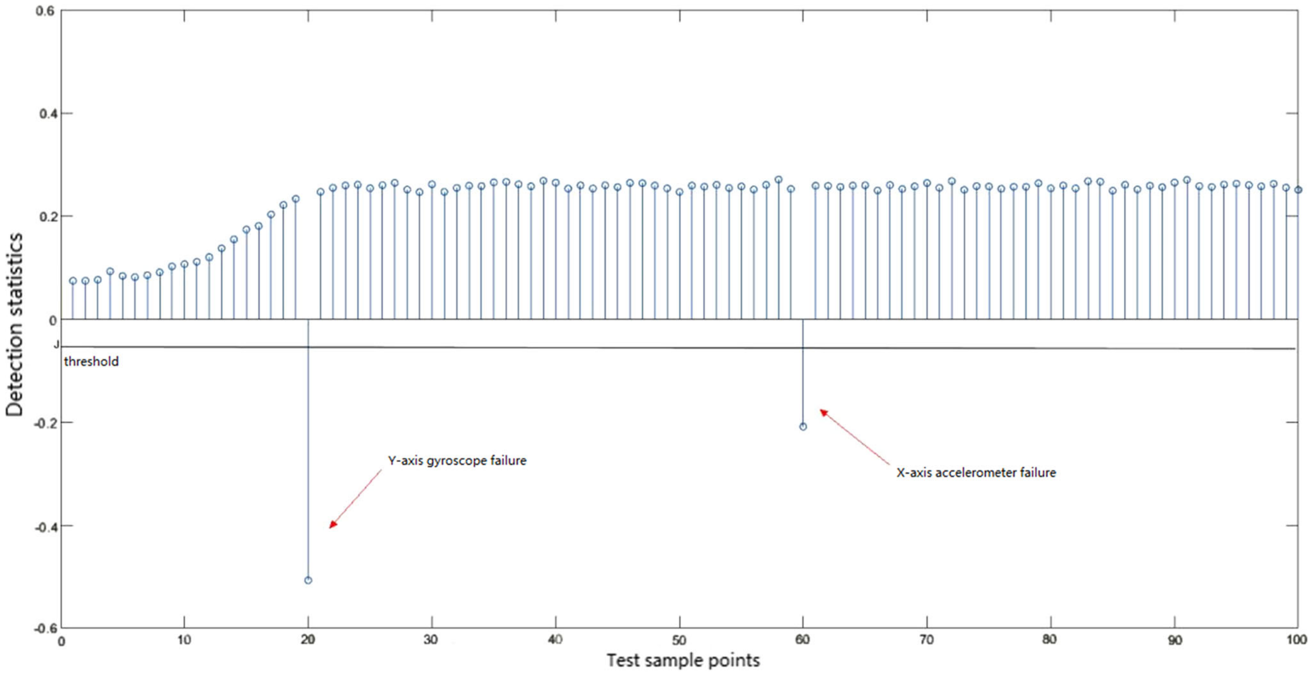


Fig. 11 Abrupt fault detection

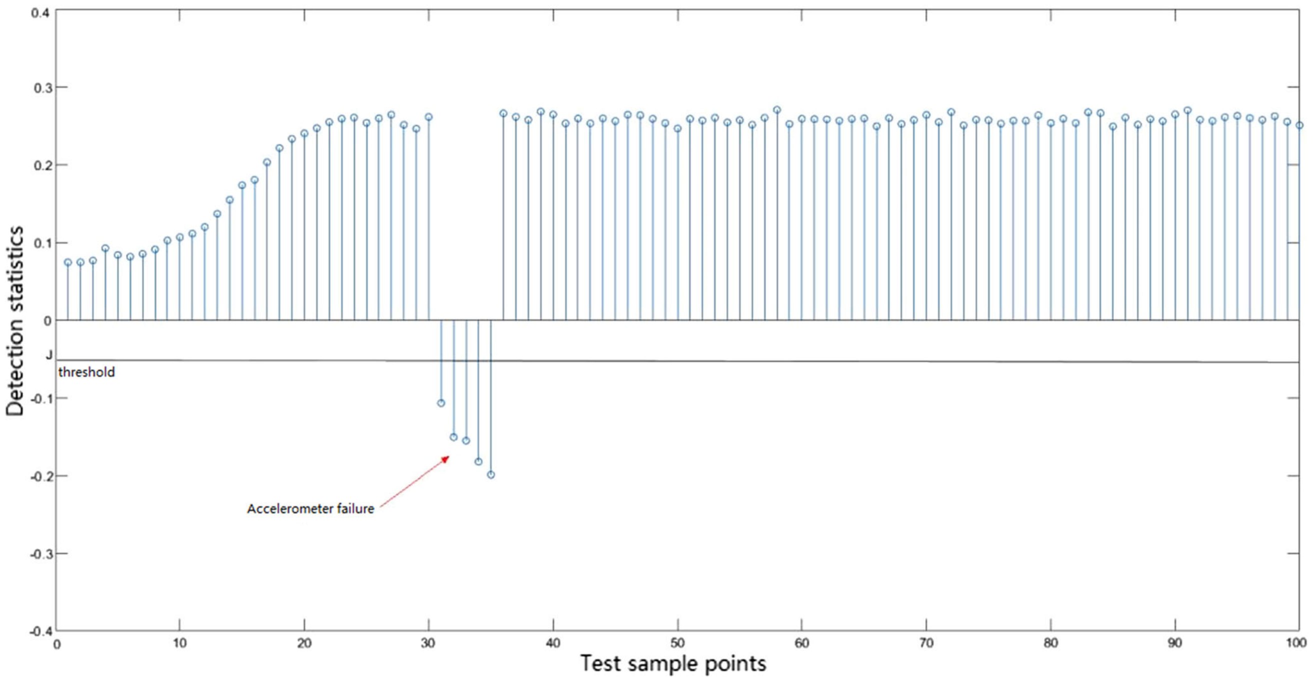


Fig. 12 Slowly changing fault detection

Table 2 Properties of raw materials

Training sample size	50	100	150	200	250	1000
One-class SVM (%)	89	92	93	95	96	98.7
Neural network (%)	70	72	81	85	93	99

neural network and one-class SVM are trained with different numbers of identical training sample sets, and then, the same test sample and error amplitude are used to compare the fault detection rates of the two models. The results are shown in Table 2.

As can be seen from Table 1, when the number of samples is small, the correct detection rate of one-class SVM model is obviously higher than that of the neural network model. When the number of samples is large, both models have good estimation performance. SVM is specific to the finite sample case, and its goal is not only the optimal value when the number of samples tends to infinity, but the optimal solution under the existing information. SVM replaces the empirical risk minimization with structural risk minimization, which solves the learning problem of small samples well. Theoretically, the SVM algorithm will get the global optimum, which solves the local extremum problem that neural network method cannot avoid. Fault is a small probability event with fewer samples, so it can be said that SVM can show better characteristics in the field of fault diagnosis. The simulation of this section also shows that one-class SVM is more suitable for fault detection in integrated navigation.

The simulation results show that the one-class SVM fault detection algorithm can detect the abrupt and slow change faults of the navigation sub-filter system very well, with short delay and strong real-time performance. Compared with the fault detection algorithm based on neural network, it is proved that the detection rate is higher and the performance is better in small sample. These advantages will help the multi-source information fault-tolerant integrated navigation system to improve the adaptability and accuracy.

3.3 System simulation and analysis

1. Simulation and analysis of single subsystem without fault

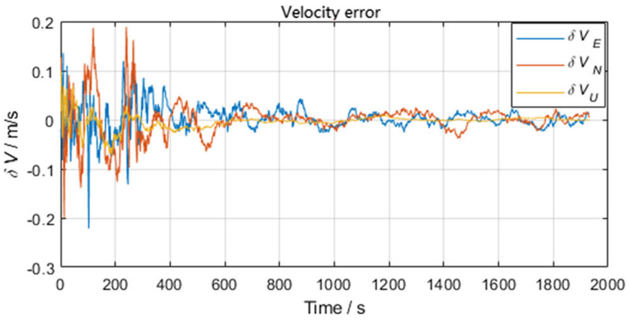


Fig. 13 Speed error of the INS/GPS subsystem

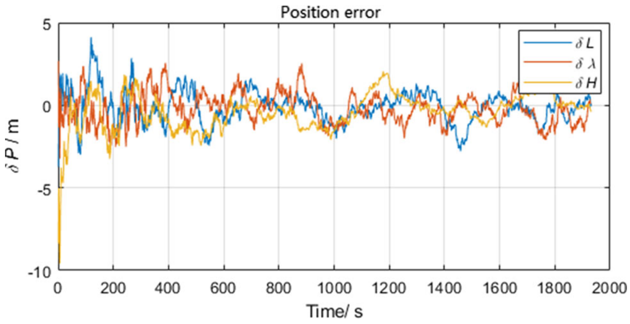


Fig. 14 Position error curve of INS/GPS subsystem

Each sub-filter estimates one or several states of the overall velocity, position, and attitude. The accuracy of the estimated state will be improved, and the estimated state error is large or even divergent. Therefore, the simulation and analysis of a single subsystem will be based on the measurement equation.

First analyze the INS/GPS subsystem. Figures 13 and 14 are plots of speed error and position error for the INS/GPS subsystem.

Analyze the INS/SAR subsystem. Figure 15 is a graph of longitude and latitude error curves.

Analyze the INS/TAN subsystem. Figure 16 is a graph of height error.

Analyze the INS/CNS subsystem. Figure 17 is a roll and pitch error graph and Fig. 18 is a yaw error graph.

1. Analysis of the overall system without failure

After analyzing the individual subsystems, the overall active fault-tolerant integrated navigation is now analyzed. When no fault occurs, the information of the sub-filter will be sent to the main filter for information fusion to obtain an estimate of position, velocity, and attitude.

Figures 19, 20, and 21 are error estimates for the state of active fault-tolerant integrated navigation.

2. Simulation and analysis of active fault-tolerant integrated navigation in case of failure

This section simulates the situation when a sub-navigation system fails. For example, in the case of a TAN fault, INS/TAN is a state estimate of the altitude. In order to avoid the influence of initial alignment, the fault value of TAN is added to the fault at 1500 s. After fault detection, it can be obtained as shown in Fig. 22. It can be seen from the figure that the height error increases sharply at 1500 s and affects the accuracy of the next period of time.

When the one-class SVM detects a fault, it uses the LE-DNN neural network that is always in the training state to predict the data at the start of the fault and replace the fault data. Figure 23 shows the overall prediction of the neural network trained from the start time to 1499 s.

It can be seen that the error curve predicted by the LE-DNN neural network will compensate well for the large error caused by the fault. By replacing the fault data into the main filter, a better estimate of the overall error will be obtained.

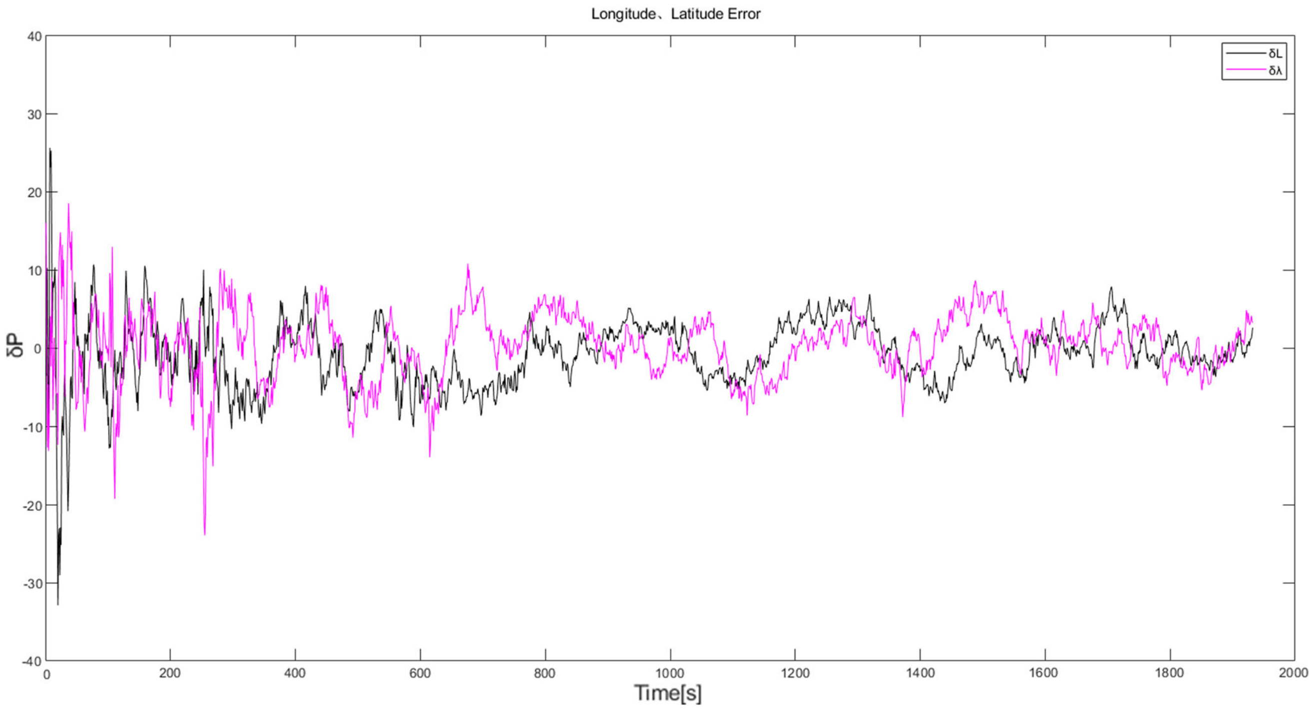


Fig. 15 Longitude and latitude error graph of INS/SAR subsystem

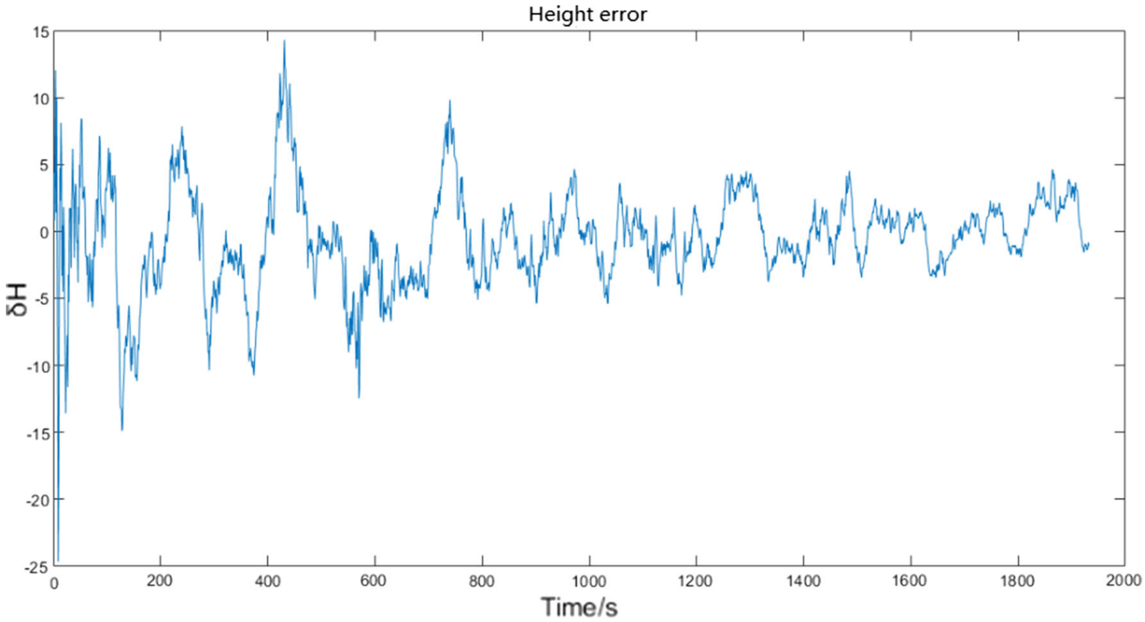


Fig. 16 Height error curve of INS/TAN subsystem

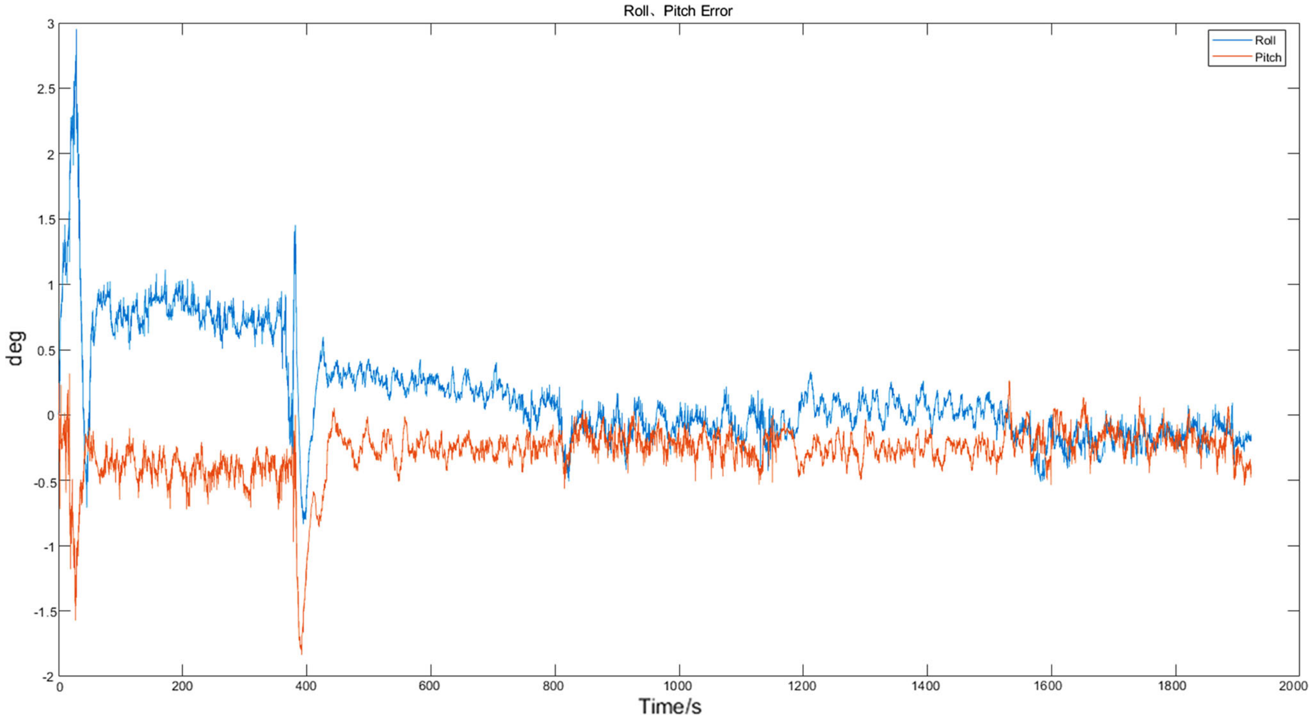


Fig. 17 Roll and pitch error curve of INS/CNS subsystem

4 Conclusion

Accuracy and efficiency are key criteria for evaluating navigation in navigation systems. This paper proposes an active fault-tolerant integrated navigation system, that is, adds neural network to help establish active fault-tolerant integrated navigation in fault-tolerant integrated navigation based on one-class SVM fault detection algorithm. During the running of the active fault-tolerant integrated navigation system, each sub-filter enters the main filter for information fusion, and the one-class SVM fault detection algorithm detects the fault of the system and as the signal for the neural network starts to predict. When there is no fault, the neural network trains each sub-filter; when there

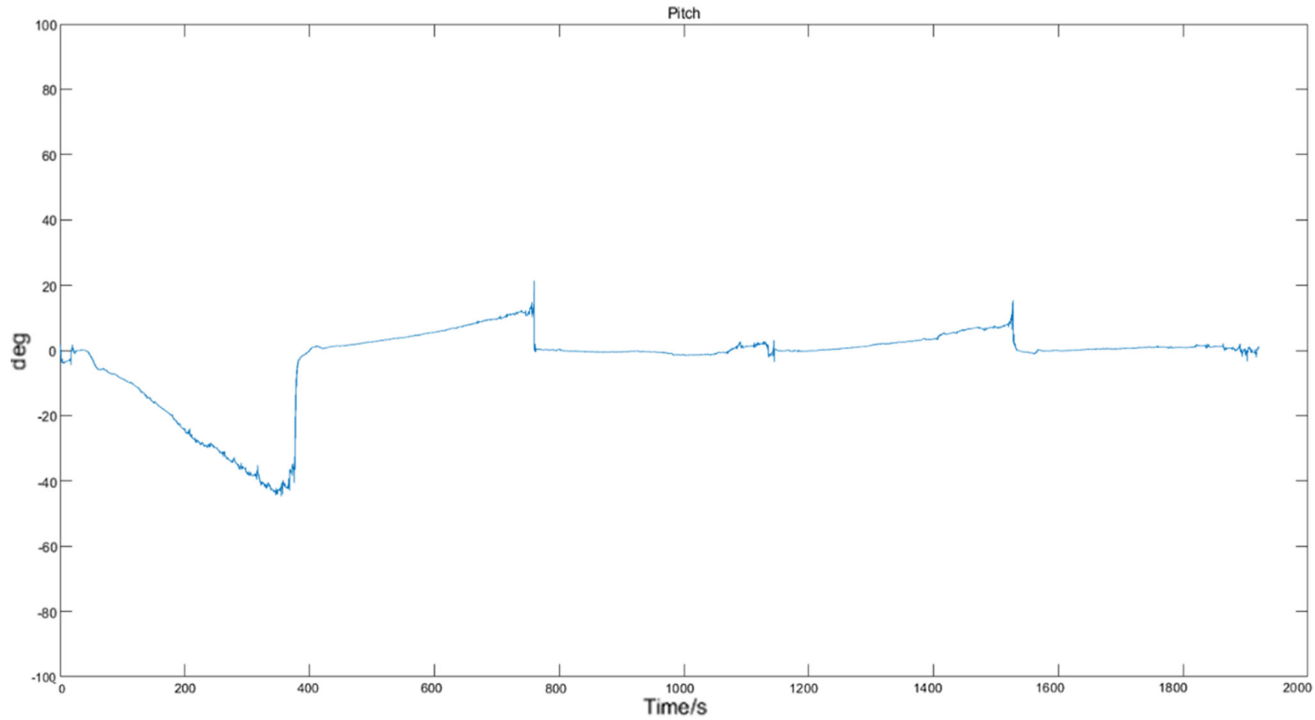


Fig. 18 Yaw error curve of INS/CNS subsystem

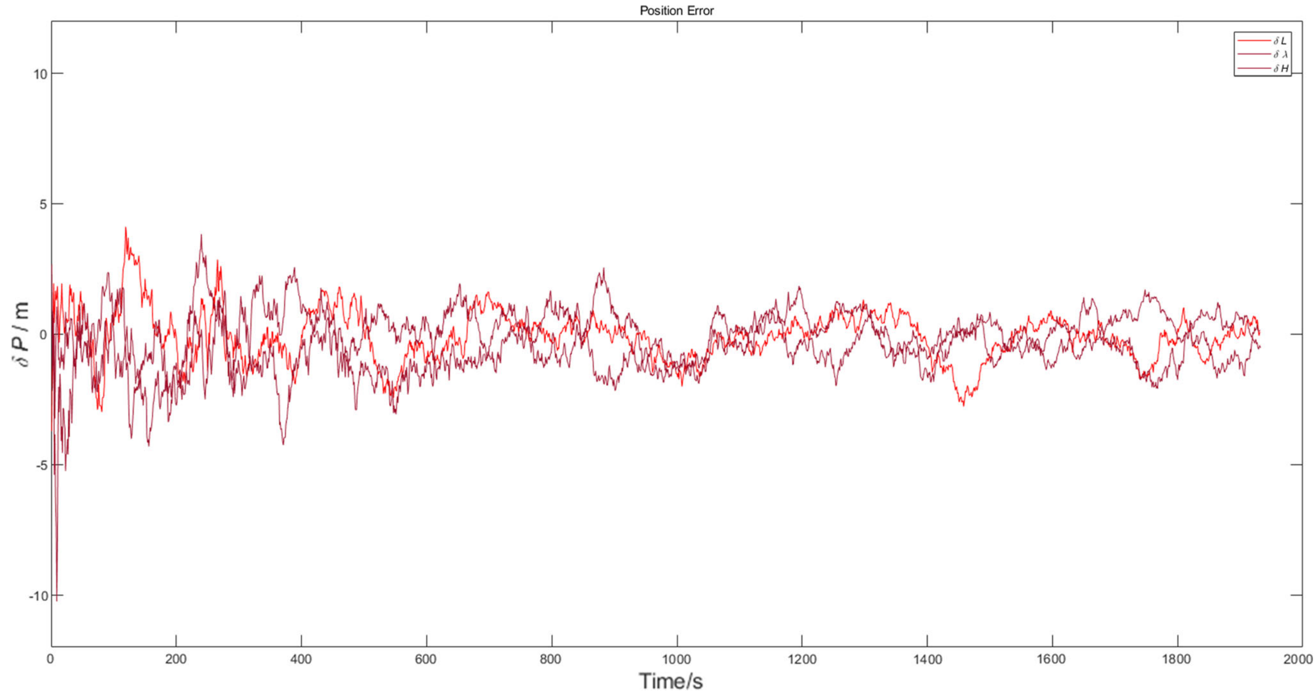


Fig. 19 Position estimation error of active fault-tolerant integrated navigation

is a fault detected, the neural network that has been in the training state will predict the fault time data and replace the fault data with the data predicted by the neural network into the main filter for fusion. Through the simulation of the system failure and the state analysis of the subsystem. It can be known that, compared with the general fault-tolerant integrated navigation, the active fault-tolerant integrated navigation system can make each sub-filter work

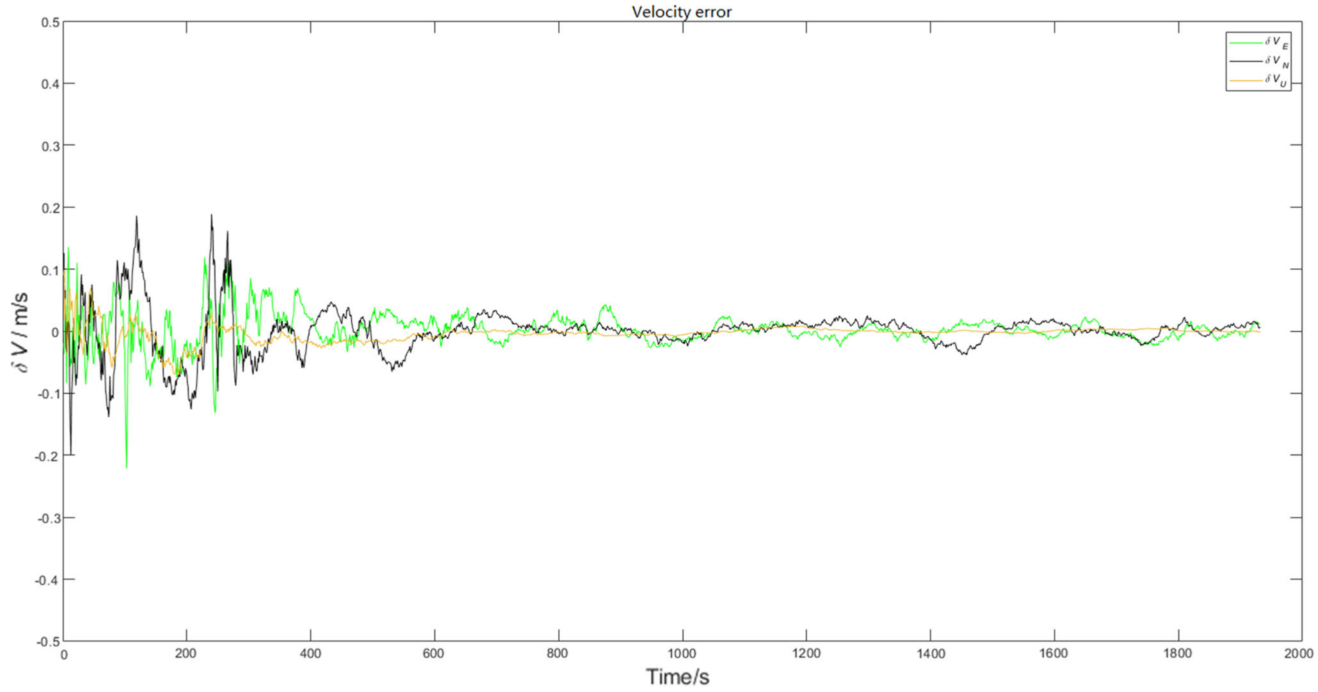


Fig. 20 Speed estimation error of active fault-tolerant integrated navigation

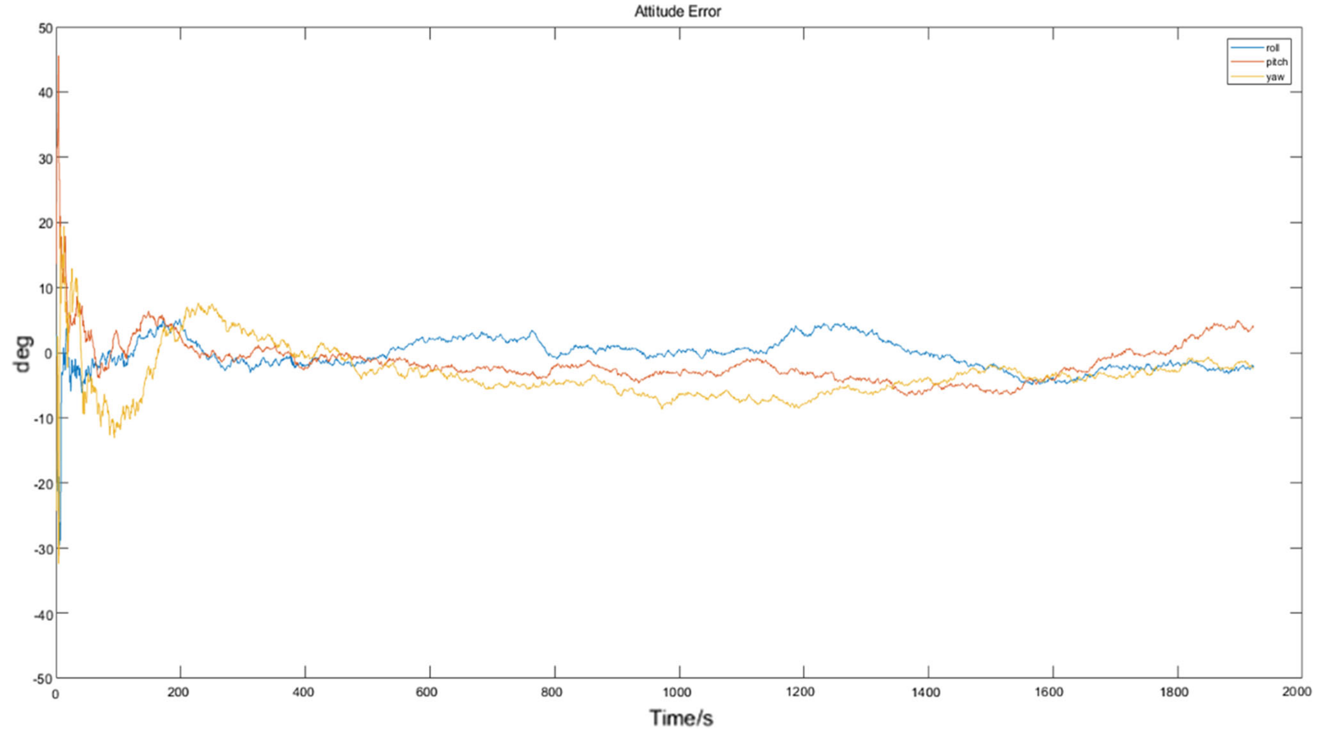


Fig. 21 Attitude estimation error of active fault-tolerant integrated navigation

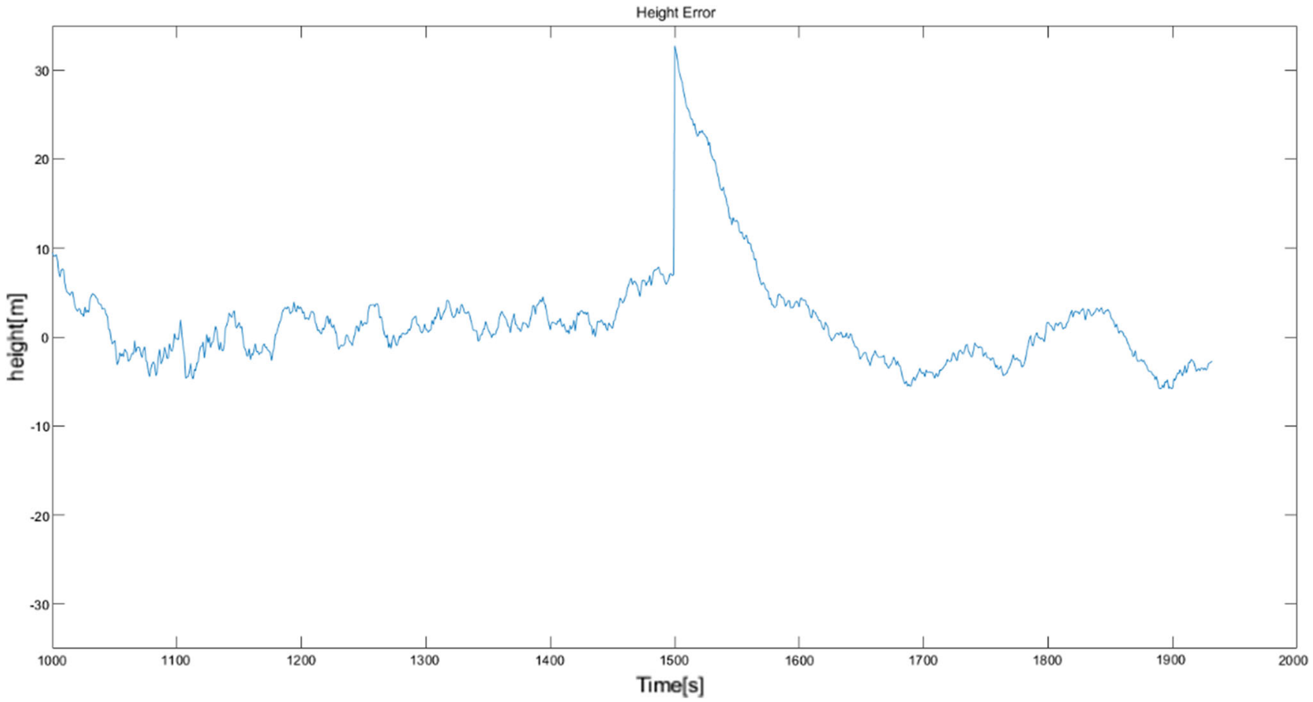


Fig. 22 INS/TAN height estimation error with failure

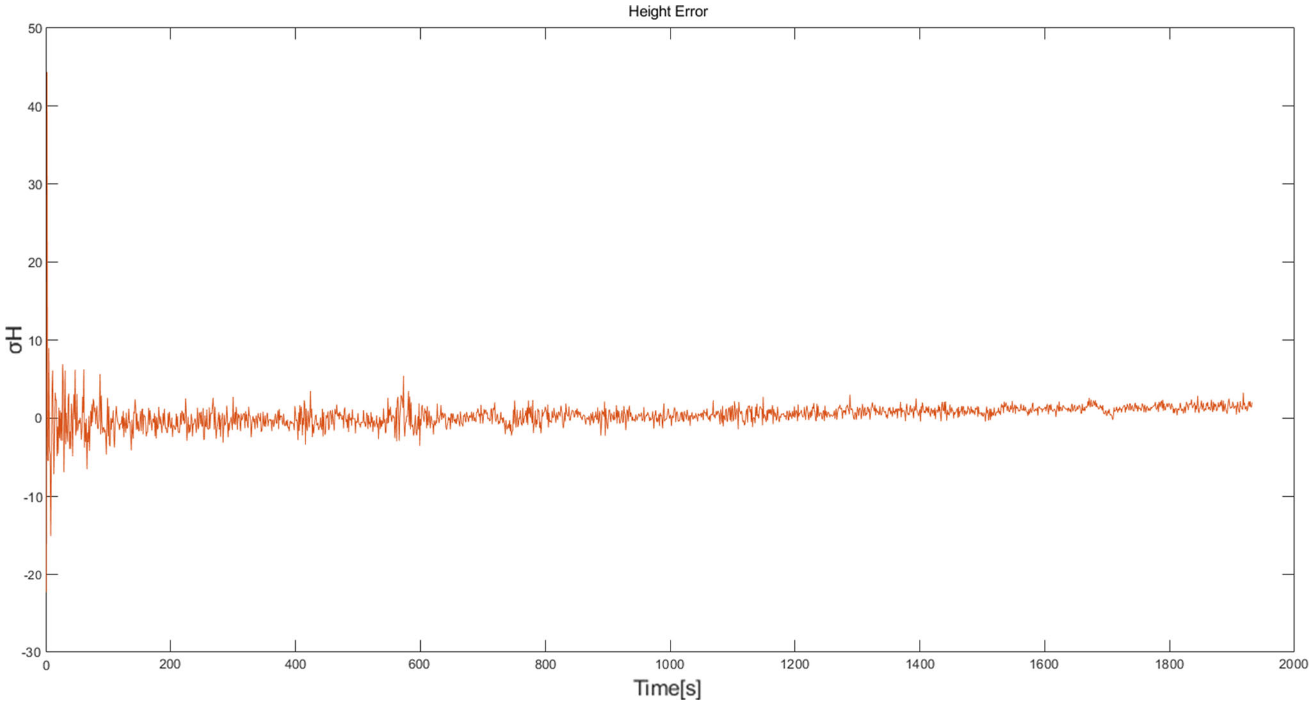


Fig. 23 Error curves predicted by LE-DNN neural network

uninterruptedly and the overall state can be optimally estimated regardless of whether the sub-navigation system fails.

Acknowledgements The authors would like to thank Prof. Jingdong Yu at National Key Laboratory of Science and Technology on Communications of UESTC for help, and Prof. Long Jin and Prof. Yonglun Luo at Research Institute of Electronic Science and Technology of UESTC for the assistance. The author also wants to thank Research Institute of Electronic Science and Technology and Key Laboratory of Integrated Electronic System, Ministry of Education, for their support for this research. However, the opinions expressed in this paper are solely those of the authors.

Compliance with ethical standards

Conflict of interest We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work; there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled.

References

1. Zhou Z, Li Y, Liu J et al (2013) Equality constrained robust measurement fusion for adaptive Kalman-filter-based heterogeneous multi-sensor navigation. IEEE Trans Aerosp Electron Syst 49(4):2146–2157
2. Guo C, Xu B, Tian Z (2018) Research on multi-constellation GNSS compatible acquisition strategy based on GPU high-performance operation. EURASIP J Wirel Commun Netw 1:112
3. Liu L, Fu J (2010) Improved state-χ^2 fault detection of navigation systems based on neural network. In: Control and decision conference (CCDC), 2010 Chinese. IEEE, pp 3932–3937
4. Liu HY, Feng CT, Wang HN (2011) Method of inertial aided satellite navigation and its integrity monitoring. J Astronaut 32(4):775–780
5. Zhenlu S (1999) Optimal fault detection filter design. J Beijing Univ Aeronaut Astronaut
6. Tao H (2008) Application of BP neural network in sensor fault diagnosis for flight control system. Comput Meas Control 16(5):613–615
7. Wei-Guang G (2008) Neural network aided GPS/INS integrated navigation fault detection algorithms. Acta Geod Cartogr Sin 80(2):186–192
8. Zhang J, Zhang T, Jiang X et al (2012) Tightly coupled GPS/INS integrated navigation algorithm based on Kalman filter. In: International conference on business computing & global informatization. IEEE, pp 588–591
9. Zhang Y, Wang H, Wang H (2017) Integrated navigation positioning algorithm based on improved Kalman filter. In: International conference on smart grid and electrical automation. IEEE, pp 255–259
10. Hongxin J, Tao Y, Xiaogang W et al (2017) Unmanned aerial vehicle relative navigation method based on robust high degree cubature filtering. J Natl Univ Def Technol 39(4):139–143
11. Li-Jia XU, Yang-Zhou C, Ping-Yuan C (2004) State estimation of integrated navigation system based on neural network. J Chin Inert Technol 12(2):40–46
12. Wang M, Fu Y (2008) State estimation of ALV integrated navigation system based on BP neural network. In: Eighth international conference on intelligent systems design & applications. IEEE
13. Chen X, Xiang S, Liu CL, Pan CH (2014) Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geosci Remote Sens Lett 11(10):1797–1801
14. Li K, Wu Y, Nan Y et al (2017) Hierarchical multi-class classification in multimodal spacecraft data using DNN and weighted support vector machine. Neurocomputing 259:55–65
15. Kai-Wei C, Chang HW, Li CY et al (2009) An artificial neural network embedded position and orientation determination algorithm for low cost MEMS INS/GPS integrated sensors. Sensors 9(4):2586
16. Li YC, Lu XL, Wang HX, Bao Z (2007) Research on positioning and measuring speed in the high speed sar system based on high precision map matching. Syst Eng Electron
17. Yan X, Ouyang Y, Sun F et al (2013) Kalman filter applied in underwater integrated navigation system. Geod Geodyn 4(1):46–50
18. Guo C, Yan J, Tian Z (2018) Analysis and design of an attitude calculation algorithm based on Elman neural network for SINS. Clust Comput 6:1–6
19. Yin A, Lu J, Dai Z et al (2016) Isomap and deep belief network-based machine health combined assessment model. Stroj Vesn 62(12):740–750
20. Hemmati F, Orfali W, Gadala MS (2016) Roller bearing acoustic signature extraction by wavelet packet transform, applications in fault detection and size estimation. Appl Acoust 104:101–118
21. Hauberg S (2015) Principal curves on Riemannian manifolds. IEEE Trans Pattern Anal Mach Intell 38(9):1915
22. Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. MIT Press, Cambridge
23. Jafari A, Almasganj F (2010) Using Laplacian eigenmaps latent variable model and manifold learning to improve speech recognition accuracy. Speech Commun 52(9):725–735
24. Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
25. Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554
26. AlThobiani F, Ball A (2014) An approach to fault diagnosis of reciprocating compressor valves using Teager–Kaiser energy operator and deep belief networks. Expert Syst Appl 41(9):4113–4122
27. Fu S, Wu J, Wen H, Cai Y, Wu B (2018) Software defined wireline-wireless cross-networks: framework, challenges, and prospects. IEEE Commun Mag 56(8):145–151