Journal of Global Positioning Systems (*2021*)
Vol. 17, No. 1: 81-93
DOI: 10.5081/jgps.17.1.81

Journal of Global Positioning Systems
2021
© CPGPS

Fault detection of integrated navigation system based on genetic algorithm optimization for deep belief network

Weiyu Kong, Ya Zhang* and Jiachong Chang

School of Instrumentation Science and Engineering, Harbin Institute of Technology, Harbin 150001

* Corresponding author, email: yazhang@hit.edu.cn

Abstract: An accurate fault detection method is critical in preventing the integrity of integrated navigation system from the abnormal measurements which may occur any time. Here, a genetic algorithm optimization for deep belief network fault detection method is proposed, where the system measuring residuals sequence is used as the input, and the output is the system operating state, such as normal or fault types, in pointwise. The proposed technique extracts the features with various scales, which contain both the local and the general information of the signal sequence, for making a comprehensive and precise classification. To show the validity of the proposed method, simulations based on INS/GNSS integrated navigation system are carried out. The simulation results show that the proposed fault detection algorithm and method is superior to the existing algorithms on the faults detection rate and false alarm rate, and thus, system reliability and navigation precision have been greatly improved.

Keywords: integrated navigation system; fault detection; deep belief network; genetic algorithm

1 Introduction

In recent years, using other navigation sensors to correct the accumulated navigation solution errors of the inertial navigation system (INS) has been widely implemented in many navigation applications. The increased number of navigation sources can result in the two opposite sides of effects on the stability of the navigation system. Adding navigation sources provides various and accurate measurements as input to the navigation filter, thereby improving the adaptability of the environment. On the other hand, abnormal navigation measurements, i.e. faults, may occur any time in any navigation subsystem. Faults from any of the subsystems can contaminate the data fusion algorithm, which further collapses the entire navigation system [1–3]. Thus, fault detection methods should be designed and implemented to supervise the network for avoiding the serious failures of the navigation and equipment. The fault detection techniques can be generally divided into two families: analytical model-based and data-driven methods.

Analytical model-based methods exploit a relationship between input and output variables for a known physical model. The chi-square method is one of the most popular analytical model-based fault detection methods, which contains hypotheses test for state and residual on the basis of statistical information and relevant probability statistical distributions, respectively [4, 5]. Bedjaoui and Weyer [6] presented a method based on the cumulative sum algorithm to monitor the model-based residuals. Some researchers proposed a modified transformation function to make the residual more sensitive to the faults and more robust to the noise. Thus, faults detection can be more accurate [7]. Analytical model-based methods require accurate prior information of the system. However, due to the non-linear and time-varying characteristics of the navigation system, the construction of the motion model can hardly be completely accurate. Therefore,

the model-based methods have difficulties in distinguishing faults in the navigation sources or imprecision of the physical model.

Instead of constructing a physical model, data-driven methods operate the fault diagnosis by processing the input/output data. In [8], a fault was detected by directly monitoring the signals collected from the sensors. Some other works are based on the residual data from the Kalman filters [9–14]. In these works, pattern recognition techniques are often referred to make a classification of the data. In [9], a belief rule base (BRB) method was used as an automatic non-linear model builder that is trained by the pre-labelled historic data. In [10, 12], neural network was introduced to adjust to the imperfection physical model and diagnose the categories of the faults. However, like most fuzzy algorithms, BRB method greatly relies on the experience of the researchers. Similarly, the performance of the NN algorithm widely depends on the extracted signal features that are the input of the system. Therefore, an inappropriate pre-set feature extraction by researchers can result in a high false alarm rate and a high missed rate. In [14, 15], a deep belief network (DBN) was proposed to get rid of the dependence on human experience for fault detection. DBN can learn the features extracted from the data sequence by itself. While, DBN also has disadvantages as it can only make classifications for a batch of data stream, i.e. determine whether a fault happens or not during a period. Thus, the monitoring timeliness cannot be guaranteed.

To overcome the aforementioned shortcomings, it is needed to apply a more flexible and reasonable approach that is able to autonomously summarize the uncertain information of the signal and output accurate diagnosis for each sample point of the input data sequence. In this paper, a genetic algorithm optimization for deep belief network (GA-DBN) fault detection method is proposed. Several innovations are proposed, such as a new pooling method for obtaining both local and general features of the signal, and a comprehensive classification method using multi-scale feature maps. Moreover, a novel kind of convolutional layer help the method obtain a classification map in ‘pixelwise’. The method proposed in this paper provides a new solution for signal detection with prior information uncertainty.

The paper is structured as follows. Section 2 introduces the proposed fault detection method including the working principle of GA-DBN and the structure of the integrated navigation system. In order to validate the proposed positioning framework, test data were collected; the data acquisition and the test data set are presented in Section 3. Section 4 discusses the results of the performance analysis of the proposed method. Finally, Section 5 summarizes our findings.

2 Descriptions

In this section, the traditional DBN is firstly described and then the structure of the proposed adaptive GA-DBN fault detection with several innovations. It improves the diagnostic accuracy and generalization ability of the original algorithm, and uses the test set to verify, and compares with the improved algorithm.

2.1 Overview of DBN

Deep belief network (DBN) is a deep learning method which combines probability and statistics, machine learning and neural network. Its component is restricted Boltzmann machine（RBM）. The core of DBN is to update and optimize the connection weight of DBN by greedy learning method. When deep belief network is applied to equipment fault diagnosis, firstly, the forward unsupervised layer by layer training is used to extract the fault features from the running state signals of the equipment to be diagnosed, and then the fault recognition ability of deep belief network is optimized by the reverse supervised fine tuning.

Restricted Boltzmann machine is a kind of randomly generated neural network, which can complete the learning of probability distribution with the help of input data. Each RBM contains the visual layer and the hidden layer. The neurons in the visual layer and the hidden layer are connected in two directions, but there is no connection relationship between the neurons in the same layer. As shown in Figure 1a, RBM can be regarded as an undirected graph model with binary structure. The bottom layer of RBM is the visible layer unit, which is used to

represent the observed data, and n is the number of neurons in the visible layer; through the continuous abstraction of the input data, the high-level characteristics of the signal are obtained, that is, the hidden layer unit of RBM, where n represents the number of neurons in the hidden layer. In addition, b and a represent the offset of the visible layer and the hidden layer cells respectively, and the weight matrix between the visible layer and the hidden layer is represented by ω.

In Figure 1b, DBN achieves greedy learning layer by layer by stacking multiple RBMs, which is used for signal feature extraction. In this paper, the low-level representation is the original data, and the high-level representation is the feature representation extracted from the original data. The feature extracted by each RBM will be used as the original data of the next RBM for continuous training. Before reaching the maximum number of iterations, the weight and threshold in RBM will be maintained to update. DBN can get a higher level expression by extracting the features of the original data layer by layer, so as to mine the hidden essential features and realize the fault state recognition.

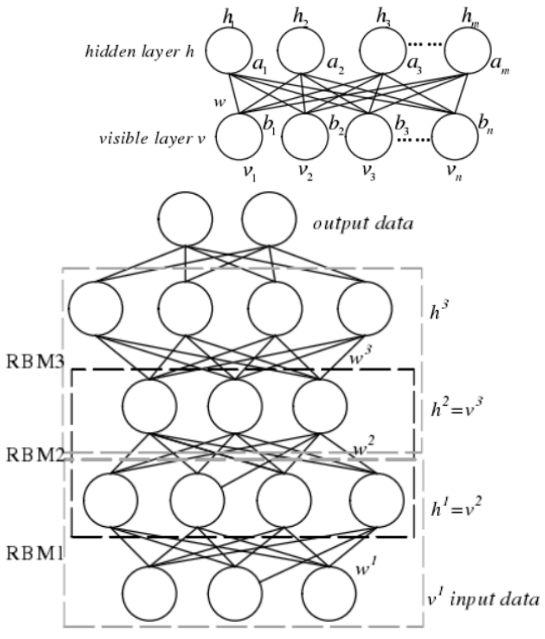

Fig. 1 Structure of RBM and Deep Belief Network

2.2 Fault diagnosis flow based on DBN

In Figure 2, DBN is used to classify and identify the faults, and the specific process of integrated navigation system fault diagnosis through this model mainly includes the following points.

Firstly, the problem of fault diagnosis and the type of fault are defined to determine the number of nodes in the output layer of deep belief network. Secondly, since the input data range of deep belief network is between [0,1], it is necessary to normalize the fault data. Since the sensor fault diagnosis will be carried out in the future, the vibration signal will have positive and negative values alternately, so the linear normalization method is selected.

$$\overline{x_i} = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{1}$$

Thirdly, the normalized fault data is divided into training set and test set for the training and learning of deep belief network structure. Fourthly, the parameters of the deep belief network are initialized, including the number of DBN model layers, the length of input samples, the number of nodes in each layer, the forward stacking learning rate, the backward fine-tuning learning rate, the number of iterations, momentum factor, weight matrix, the bias of hidden layer cells, and the bias of visible layer cells. Fifthly, the training set is used as the input data of deep belief network to train it, which mainly includes forward stacking learning and backward fine-tuning learning. The backward fine-tuning process mainly refers to a small amount of label data, and the weight matrix of DBN and other parameters are refined by gradient descent method. After learning, the diagnosis model is obtained. Sixthly, the test set is input into the trained deep belief network classification model, and the output vectors of each hidden layer are recorded, and finally the fault state recognition results are obtained. To sum up, the process of fault diagnosis based on DBN includes four parts: data preprocessing, forward unsupervised training, reverse fine-tuning and fault state recognition.

There are many key parameters in the learning process of DBN, which are closely related to the performance of the whole DBN model, so the setting of parameters is of great significance to the fault diagnosis based on DBN, so the reasonable setting of parameters can effectively improve the training speed and fault recognition rate of DBN.

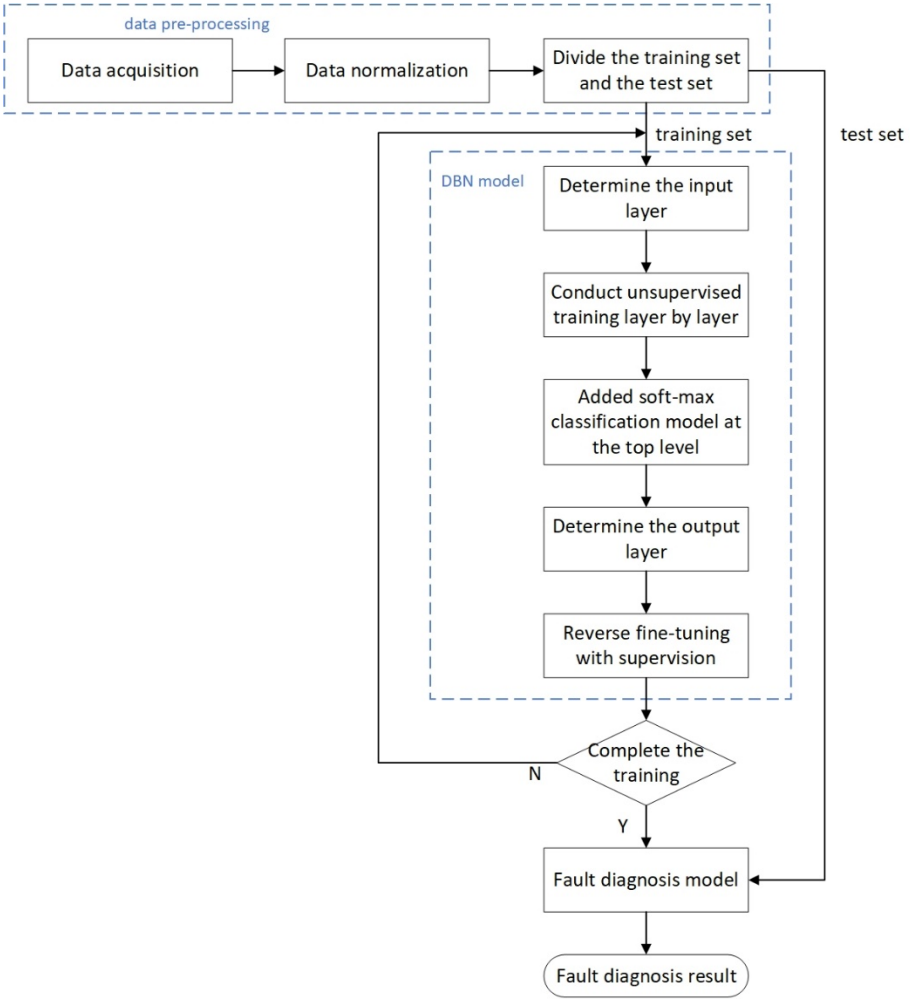

Fig. 2 Fault diagnosis flow based on DBN

First, the setting of connection weight ω: neuron bias (visual layer offset b, hidden layer offset a) is the first. If the initial value of the connection weight is too large, the speed of model training is accelerated, but it is not conducive to the classification results of the fault. According to experience, the initial value of weight is normal distribution in general, which is in accordance with N(0,0.01), while offset b and a of visible layer and implied layer can be initialized to 0. During DBN training, the connection weight and the neuron bias are updated continuously (until the end of training), so the initial value will not have a decisive effect on the fault diagnosis results. Here, we refer to the empirical formula to initialize these three parameters randomly.

$$\begin{cases} \omega = 0.1 \times randn(n,m) \\ b = zeros(1,n) \\ a = zeros(1,m) \end{cases} \tag{2}$$

where n is the number of neural units in the input layer, m is the number of neural units in the output layer, and randn (n, m) generates a pseudo-random number matrix of standard normal distribution (i.e. mean value is 0 and variance is 1) of n rows and m columns, then the connection weight ω is a normal distribution with mean value of 0 and standard deviation of 0.1. zeros (1, n) generates a row of n-column zero matrix (that is, all vectors are 0), and zeros (1, m) generates a row of m-column zero matrix, that is, the initial value of the bias between the view layer and the hidden layer is 0.

Second, the setting of learning rate (forward learning rate ε, reverse fine-tuning learning rate α): Random gradient descent algorithm is the key algorithm to realize DBN network training, and the learning rate is an important parameter of the algorithm. The learning rate determines how far the gradient direction should move in the process of executing the algorithm. If the value of the learning

rate is too low, the step toward the minimum value of the loss function is small, and the learning will be relatively reliable. However, due to the slower update speed of the weight, the model can not converge quickly, resulting in more time-consuming in the whole training process, which seriously affects the learning efficiency. On the contrary, if the set learning rate is too high, the weight and bias will change more, which will easily increase the reconstruction error, and even cause the training not to converge. Similarly, according to the experience, the learning rate of DBN forward stack learning process is generally set to 0.1, while the learning rate of BP reverse fine-tuning process is generally set to 0.01

Finally, the setting of momentum factor m_b : like the learning rate, the momentum factor is also the key parameter in the gradient descent method. Its main function is to take the gradient estimation of the last iteration into account in the algorithm, so as to improve the anti oscillation performance of the whole training process and make the algorithm converge faster and more stable.

$$\begin{cases} \theta_{t+1} = \theta_t + \Delta\theta_t \\ \Delta\theta_t = m_b + \Delta\theta_{t-1} + \varepsilon \times \dfrac{\partial \ln L}{\partial \ln \theta} \end{cases} \tag{3}$$

The above formula is the parameter updating formula after introducing momentum factor, where: $\partial \ln L / \partial \ln \theta$ represents the gradient of the log likelihood function of the current sample, and m_b is momentum factor and learning rate ε . It can be seen from the formula that after adding momentum factor to the parameter updating process, the parameter correction is determined by the gradient value of this iteration and the last correction value. Similarly, according to experience, the momentum factor is generally set to a value in the range of 0.5-0.9, which is set to 0.9 in this paper.

2.3 Implementation of genetic algorithm

Generally, the network parameters of the DBN algorithm are obtained though empirical knowledge or experiments, which have certain limitations in practical applications. So an improved DBN algorithm which can adaptively select parameters is needed to achieve the fault diagnosis better. In this manuscript, the genetic algorithm is used to optimize the hyperparameter of the DBN, which includes the number of the input layer nodes, three numbers of the hidden layer nodes, learning rate, fine-adjustment learning rate and momentum factor.

Genetic algorithm makes the whole population develop in the direction of adapting to the environment through multiple iterative evolution process. Finally, the individual with the highest degree of adaptability to the environment in the population is selected as the optimal solution of the whole algorithm, and the optimization results are output for subsequent use. The algorithm implementation process is shown in Figure 3.

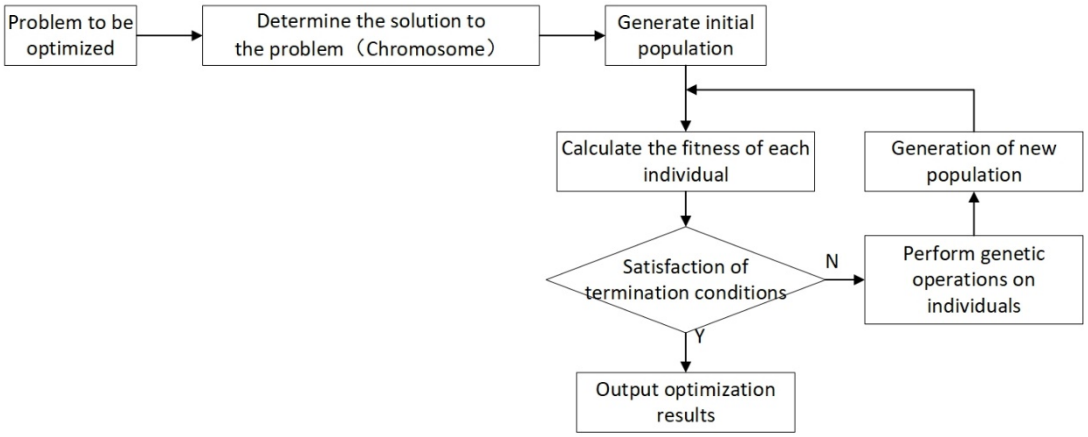

Fig. 3 Implementation process of genetic algorithm

The basic steps of genetic algorithm include the following aspects. Firstly, an initial population is generated, which is composed of M randomly generated individuals. Each individual represents a

solution of the problem, which is called chromosome. Secondly, according to the actual problem to be optimized, the fitness of each individual in the group is calculated. Fitness refers to the individual's adaptability in the environment, that is, the individual can achieve the optimal solution in the actual optimization problem. Thirdly, judge the termination condition of genetic algorithm. If it is satisfied, the optimal solution or the final population will be output. Fourth, crossover and mutation operations are performed on individuals in the population to generate new chromosomes, namely offspring. Fifthly, a certain number of individuals are selected to form a new population. Then, we need to find the appropriate intelligent optimization algorithm to optimize the model parameters of deep belief network, so as to build the structural optimal deep belief network fault diagnosis model, and realize the fault state classification of each sensor in the integrated navigation system. According to the analysis of the optimization problem, this paper mainly chooses group optimization algorithm to complete the parameter optimization of deep belief network. Genetic algorithm has the support of mathematical theory, and has the characteristics of universality, adaptability and global, so it is very suitable for solving discrete problems. Therefore, through comprehensive consideration, the genetic algorithm is finally selected to solve the parameter optimization problem of deep belief network.

2.4 Structure of integrated navigation system

To achieve an optimized fusion and a better fault tolerance for the integrated navigation systems, a non-reset federal Kalman filter (FKF) is utilized for the data fusion. In this paper, a integrated navigation system consisting of INS and GNSS, is used as an example [17–19]. The structure of FKF allows the fault detection network to be trained independently for each local system and to work parallel.

The FKF consists of two local filters and one main filter. INS is the reference navigation system. In the local filters, GNSS assist the reference system with navigation observations. The local estimates $\hat{X}_i$ and the local covariance matrixes P_i are sent to the main filter for obtaining a global optimal estimation $\hat{X}_g$ and update the global covariance matrix P_g. Note that, navigation observations output from GNSS system is position. The detail of implemented FKF is presented in the Appendix.

In a local filter, as described in the Appendix, after the prediction state vector $\hat{x}_{k+1/k}$ is obtained according to (11), the following equation can be used to calculate the observations prediction at time k+1:

$$\hat{y}_{k+1/k} = H_k \hat{x}_{k+1/k} \tag{4}$$

Then, the filter residual can be defined as the difference between the observation y_{k+1} and its prediction $\hat{y}_{k+1/k}$ which can be expressed by

$$r_{k+1} = y_{k+1} - \hat{y}_{k+1/k} \tag{5}$$

The residual is an important measure of how well the estimator is performing, as it is a zero-mean Gaussian white-noise process, and once a fault occurs, the characteristics of the distribution are very likely to change. In this paper, the fault detection is implemented by monitoring the residual. Residual is calculated with every observation update, and is stored consecutively to a predefined length l. Then, the residual data stream is input to the GA-DBN fault detection module. The proposed GA-DBN fault detection function can be defined as (see (6)) where S_{k+i} is the output of the GA-DBN, which is the scores for categories of each residual timing. S_{k+i} is a vector of length equal to the number of categories. The category with the highest score is output as the final diagnosis result at time i, denoted as d_{k+i}, which indicates different system states, such as working properly, observation error surge, INS drift error surge etc.

$$\begin{aligned} & f_{GA-DBN}([r_{k+1}^T, r_{k+2}^T, ..., r_{k+i}^T]) \\ &= [\max(S_{k+1}), \max(S_{k+2}), ..., \max(S_{k+i})] \\ &= [d_{k+1}, d_{k+2}, ..., d_{k+i}] \end{aligned} \tag{6}$$

Since input of GA-DBN is a residual array collected over a period of time, faults can be spread to the main filter by contaminated INS fixing error

during the period between fault occurrence and being detected, and then to the other normal operating local filter. To isolate the fault and eliminate the contamination of the main filter and local filters, data from filters, including states, covariance matrixes, and observation updates, are stored for backup, until fault detection modules assure that no faults have been detected during the time. Once a fault has been detected, all the filters are reset to the moment before the occurrence of it, and conduct re-estimation without the fault navigation source.

3 Simulation test set-up

In order to examine the performance of the proposed fault detection method, a simulation test was conducted. The simulation test was used to illustrate the performance of detecting faults under massive samples. While, simulation test showed the influence of fault detection methods on trajectory solution.

In the simulation test, to obtain the training and validation data for the neural network, INS, GNSS output data within several kinds of faults is generated in accordance with a total of 1000 pre-established trajectories, including various manoeuvring conditions, such as uniform motion, acceleration, deacceleration, climbing, descending, S curve, and Eight curve path. Detailed specifications of simulated navigation source are shown in Table 1.

Table 2 presents detailed information of generated various kinds of faults. Different kinds of faults are set artificially into observations from different navigation sources according to the error value and lasting time in the table. Backpropagation method is used in this paper to train the network, which adjusts the weight of each neuron by calculating the gradient of the loss function [20]. In the training and validation data set, residuals of the local filters are labelled in pointwise with normal or specific fault category, and truncated to sequences with a fixed length. GA-DBN are trained for INS/GNSS, respectively. Consequently, the training set of each GA-DBN is consisted of 4700 sequences and another 300 sequences are generated as a validation set. Note, as the performance reflecting on residuals of gradually increasing position error and INS constant drift surge is similar, the above two kinds of faults are both labelled as gradual faults. In practice, during the detection process, if gradual fault has been detected from both local filters at the same time, it is determined as INS fault. In the simulation test, the DBN method and CNN method [15] is utilized as comparisons. And the maximal number of iterations is set as 20.

Table 1 Simulation specifications for INS/GNSS integrated navigation system

	Accelero-meter	Gyro	GNSS
output frequency	100 Hz	100 Hz	1 Hz
error	100μgal (bias)	0.5°/h (bias)	3 m (position)

Table 2 Details of faults

	Outliers	Positioning error surge	Positioning error Increasing gradually	Gyro, accelerometer drift error surge
source	GNSS	GNSS	GNSS	INS
mode	sudden	sudden	gradual	gradual
error	10–20 m	10–20 m	0.5–10 m/s	gyro bias: 3–10°/h; accel. bias: 500–2000 μgal
lasting time	one or two samples	>10 s	>10 s	>20 s
Fault cause	Signal blocking, Weak satellites nodes signal etc.	Ionospheric scintillation in GNSS, Deterioration of the geometry, Signal interference etc.	Multipath signal, Deception jamming etc	Abnormal vibration, temperature, electromagnetic interference etc.

In the simulation test, the car is used to carry GNSS antenna and a MEMS IMU. All sensors are synchronized to the GPS time via PPS (pulse per second) signals. The GNSS output frequency is 1 Hz. The specifications of the IMU are shown in Table 3.

Table 3 Specification for the MEMS IMU system

	Accelerometer	Gyro
output frequency	100 Hz	100 Hz
bias error (in-run)	40 μgal	0.5 °/h
bias error (initial)	2 mgal	0.25 °/s
scale factor stability	0.05%	0.05%

4 Results and discussion

In this section, the training performance of GA-DBN using simulation data is firstly introduced. Then, the performances of several fault detection methods are presented and compared. Finally, we compare and analyze the results.

In general, the validation set is studied by the NN during the training process, and it is employed to evaluate the performance of training. Fig. 4 demonstrates four typical fault detection results in the validation set from GNSS/INS system. The upper side of every subfigure presents one of the three position residual vectors that input to the GA-DBN algorithm. The lower side of subfigure present different categories.

In Fig. 4a, position residuals in X direction fluctuate within 3 m, and then, outliers with 5 m error occur around the 60 s. In this case, scores indicating outliers are output with values close to 1 during the fault, while scores for normal state drop deeply correspondingly. Fig. 4b shows another example, when position solution accuracy from GNSS degraded and lasted for over 15 s. As the fault appearances of outliers and error surge are similar, scores from both categories increase; however, because of a long fault duration time, scores for error surge maintain higher. In the case of Fig. 4c, a gradual fault from GNSS occurs at the 22 s and lasts for >20 s. Although scores of gradual faults increase at the beginning of the fault, however, since the residual error is not obvious during the first a few seconds, scores of normal state maintains even higher values. The fault has not been detected until 7 s after it occurred. The above delayed fault detection results are found in some other gradual faults cases. Fig. 4d shows another example of the gradual fault, which is caused by a sudden increased accelerometer drift error, in this case, the fault has been detected at the beginning.

Table 4 presents the fault detection performance of the GA-DBN, DBN method, and CNN for the simulated validation set. The table distinguishes four cases. The ‘Success detection rate’ row presents those results when the output of GA-DBN is same as the ground truth. The missed faults are presented in the second row titled as ‘Misdetection rate’. In these cases, faults are all classified into normal class. The ‘Misdiagnosed rate’ row indicates those faults that have been detected but classified as incorrect fault class. In some other cases, normal residuals are identified as faults. These cases are presented in the ‘False alarm rate’ row. As the chi-square method cannot distinguish types of the fault, the ‘Misdiagnosed rate’ and ‘False alarm rate’ except for the ‘total’ have been missing.

The table clearly shows the proposed GA-DBN approach achieves better detection rates. Especially for the fault type of gradual, the detection rate is significantly higher. The result shows that 92.6% of the artificially generated faults can be detected with correct fault type by the proposed GA-DBN, while for the DBN method only 74.5% can be detected, yet, without the specific fault type. CNN can detect 81.6% of the faults. However, the method can only classify a whole piece of data; therefore, an outlier will cause the whole time to be judged as faults, which resulting in a higher false alarm rate of 2.7%.

In additional, 95.6% of the outliers and 94.6% of the positioning error surge fault can be detected correctly by GA-DBN, which is higher than the gradual faults. The reason leading to a higher detection rate of the sudden faults is that the impact of the residuals brought by gradual fault at the beginning stage is very weak, and thus, is hard to detect and will do minor contamination to the navigation filters.

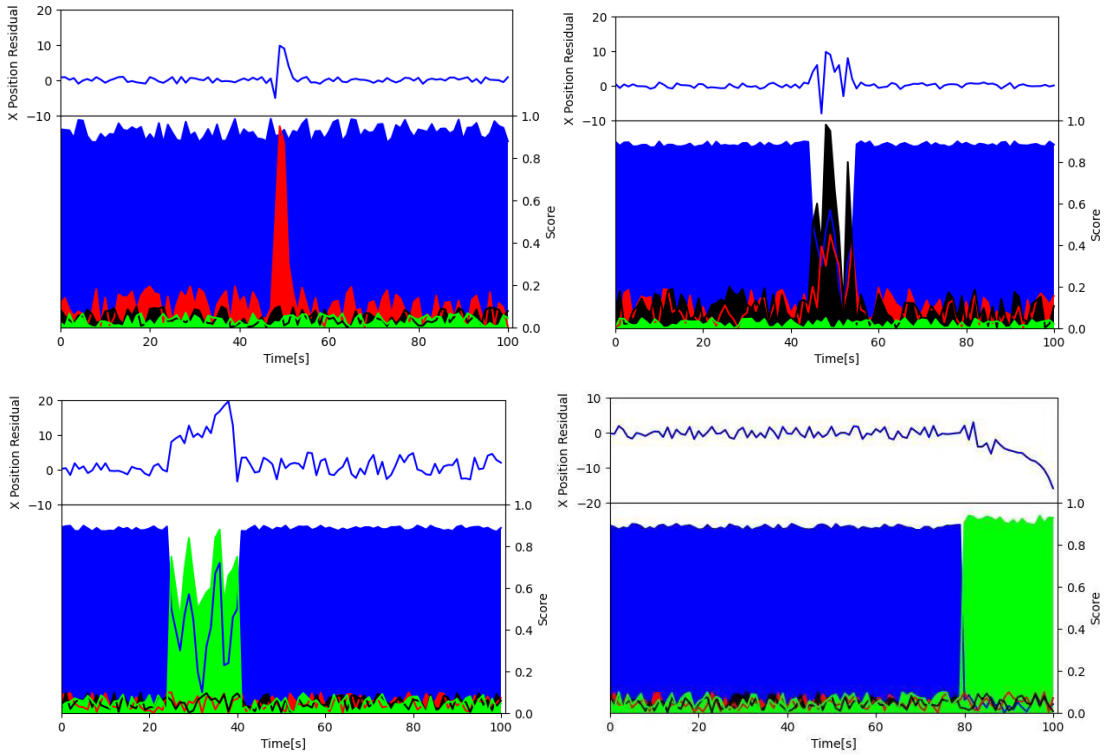

Fig. 4 Trained GA-DBN detecting faults as (a) Outliers, (b) Positioning error surge, (c) Positioning error increasing gradually, and (d)Accelerometer constant drift error surge in GNSS/INS system

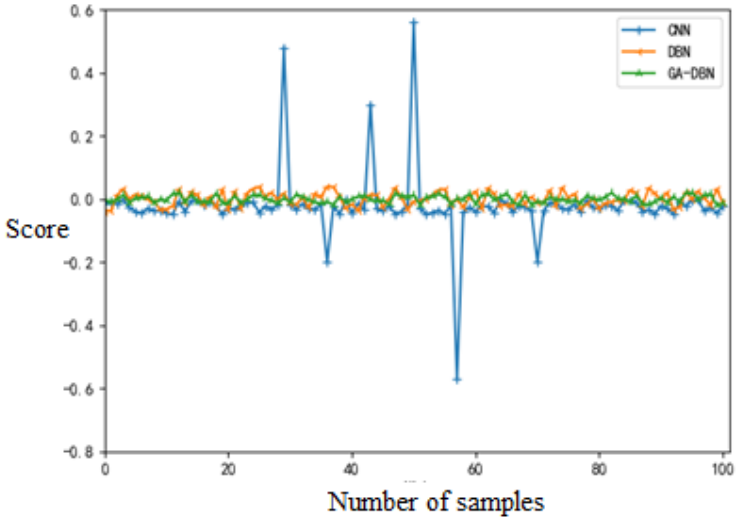

Fig. 5 Test set detection error of three models

Table 4 Fault detection results for simulation dataset

rate	Outliers			Positioning error surge			Gradual faults		
	GA-DBN %	DBN %	CNN %	GA-DBN %	DBN %	CNN %	GA-DBN %	DBN %	CNN %
Success detection	95.6	93	82	94.6	75.2	**87.4**	**92.6**	74.5	81.6
misdetection	4.4	7	9	1.6	24.8	**8.4**	**5.5**	25.8	15.9
misdiagnosed	0.0	/	9	3.8	/	**4.2**	**2.3**	/	2.5
false alarm	0.210	/	2.7	0.13	/	1.3	0	/	1.1

In order to verify the efficiency of the proposed algorithm, we also give the training time of these three algorithms. We have analyzed the relationship between the sample length and the fault identification rate of this training set and we know that: when the sample length is less than 400, the fault identification rate increases with the sample length's increase; while when the sample length is large than 400, the fault identification rate no longer increases but stabilizes to a certain position with the continuously increasing training time. Therefore, in order to facilitate the code running, we only used 400 samples when comparing the training time. And the comparison of the training time is shown in Table 5.

Table 5 Comparison of the training time

	GA-DBN	**DBN**	CNN
Training Time (s)	0.76	**0.55**	**1.32**

From Table 5, we can see that the training time of the DBN algorithm is the least while the one of the CNN algorithm is the most, and the one of the proposed algorithm is the middle, 0.76s. That's because the convolution calculations is involved in the CNN algorithm, its training time is the longest. And the optimization method based on GA is added in GA-DBN algorithm, improving the accuracy at the cost of a certain amount of the computing time. So the training time of GA-DBN algorithm is more than the one of the traditional DBN algorithm.

5 Conclusions

In this paper, we propose a modified DBN method, which is denoted as GA-DBN, to improve the fault detection performance in integrated navigation systems. To test the performance of the proposed method, DBN method and CNN are compared. In the simulation test, 1000 platform trajectories and 4 kinds of faults are generated to train and validate the network. The results indicate that 92.6% of the faults can be detected and diagnosed correctly by using the proposed GA-DBN. The detection rate of GA-DBN is 18.1% and 11% higher than the DBN and CNN methods, respectively. Therefore, the proposed fault detection method is superior to the existing algorithms on the faults detection rate and false alarm rate, and thus, system reliability and navigation precision have been greatly improved. In the future, two research directions will be explored. First, we plan to further test the proposed system under various scenarios and different navigation sources. Second, in order to increase the fault detection rate, instead of relaying the self-learning neural network, feature maps, extracted by formulas or models, can be used to aid the classification process, which may improve the efficiency of training.

6 References

[1] J. Curro and J. Raquet, "Navigation using vlf environmental features," in *Position, Location and Navigation Symposium (PLANS), 2016 IEEE/ION*, pp. 373–379, IEEE, 2016

[2] Li, C., Li, X., Y ang, X.G.: 'A fault detection method for GNSS/INS integrated navigation system based on GARCH model'. Proc. of the ION 2015 Pacific PNT Meeting, Honolulu, Hawaii, April 2015, pp. 713–718

[3] D. T. Venable, "Improving real world performance of vision aided navigation in a flight environment," tech. rep., Air Force Institute of Technology WPAFB, 2016

[4] Rapoport, I., Oshman, Y.: 'A Cramer-Rao-type estimation lower bound for system with measurement fault', IEEE Trans Autom. Control., 2005, 50, (9), pp. 34–45

[5] Wen, X., Ji, L.: 'Fault detection and diagnosis in the INS/GPS navigation system'. World Automation Congress (WAC), HI, 2014, pp. 27–32

[6] Bedjaoui, N., Weyer, E.: 'Algorithms for leak detection, estimation, isolation and localization in open water channels', Control Eng. Pract., 2011, 19, (6), pp. 564–573

[7] Miao, L.J., Shi, J.: 'Model-based robust estimation and fault detection for MEMS-INS/GPS integrated navigation systems', Chin. J. Aeronaut., 2014, 27, (4), pp. 947–954

[8] Hartert, L., Mouchaweh, M.S., Billaudel, P .: 'Monitoring of non stationary systems using dynamic pattern recognition', in Rigatos, G. (Ed.): 'Intelligent Industrial Systems: Modeling, Automation and Adaptive Behavior', (IGI Global, Hershey, PA, 2010),

pp. 417–452
[9] Zhao, X., Wang, S., Zhang, J., et al.: 'Real-time fault detection method based on belief rule base for aircraft navigation system', Chin. J. Aeronaut., 2013, 26, (3), pp. 717–729
[10] Seera, M., Lim, C.P ., Loo, C.K., et al.: 'A modified fuzzy min–max neural network for data clustering and its application to power quality monitoring', Appl. Soft Comput., 2015, 28, pp. 19–29
[11] A. Canciani and J. Raquet, "Absolute positioning using the earth's magnetic anomaly field," *Navigation*, vol. 63, no. 2, pp. 111–126, 2016
[12] Ayoubi, M., Isermann, R.: 'Neuro-fuzzy system for diagnosis', Fuzzy Sets Syst., 1997, 89, (3), pp. 289–307
[13] D. A. Grejner-Brzezinska, C. K. Toth, T. Moore, J. F. Raquet, M. M. Miller, and A. Kealy, "Multisensor navigation systems: A remedy for gnss vulnerabilities." *Proceedings of the IEEE*, vol. 104, no. 6, pp. 1339–1353, 2016
[14] 'Fault detection for airborne multi-source integrated navigation system using fully convolutional neural network'. Available at https://robomow.ion.org/pnt/abstracts.cfm?paperID=4823, accessed 15 September 2017
[15] Ince, T., Kiranyaz, S., Eren, L., et al.: 'Real-time motor fault detection by 1-D convolutional neural networks', IEEE Trans. Ind. Electron., 2016, 63, (11), pp. 7067–7075
[16] Shelhamer, E., Long, J., Darrell, T.: 'Fully convolutional networks for semantic segmentation', IEEE Trans. Pattern Anal. Mach. Intell., 2017, 39, (4), pp. 640–651
[17] C. Tanıl, S. Khanafseh, M. Joerger, and B. Pervan, "An ins monitor to detect gnss spoofers capable of tracking vehicle position," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 1, pp. 131–143, 2017
[18] C. Tanıl, S. Khanafseh, M. Joerger, and B. Pervan, "Sequential integrity monitoring for kalman filter innovations-based detectors," in *31st International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2018*, pp. 2440–2455, Institute of Navigation, 2018
[19] C. Tanıl, A. P. Air, S. Khanafseh, M. Joerger, B. Kujur, B. Kruger, L. de Groot, and B. Pervan, "Optimal ins/gnss coupling for autonomous car positioning integrity," in *32nd International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2019*, pp. 3123–3140, Institute of Navigation, 2019
[20] J. Jurado, J. Raquet, and C. M. Schubert Kabban, "Autonomous and resilient management of all-source sensors for navigation," in *Proceedings of the ION 2019 Pacific PNT Meeting*, (Honolulu, Hawaii), pp. 142–159, April 2019
[21] J. D. Jurado, J. F. Raquet, C. M. Schubert Kabban, and J. Gipson, "Residual-based multi-filter methodology for all-source fault detection, exclusion, and performance monitoring," *Navigation*, 2020
[22] J. D. Jurado and J. F. Raquet, "Towards an online sensor model validation and estimation framework," in *2018 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 1319–1325, April 2018
[23] K. M. Brink, "Partial-update schmidt–kalman filter," *Journal of Guidance, Control, and Dynamics*, pp. 1–15, 2017
[24] K. M. Brink, "Unscented partial-update schmidt–kalman filter," *Journal of Guidance, Control, and Dynamics*, vol. 41, no. 4, pp. 929–935, 2017
[25] Lian Y. F, Li G. H, Wu F. L and Zhao Y. Fault-diagnosis method for INS/GPS integrated navigation system based on PNN and genetic algorithm. *Chinese Journal of Scientific Instrument.* 2012, 33(1), pp. 120-126.
[26] Yang C. Research on Fault Detection and Fault Tolerant Technologies for Integrated Navigation Systems. *PHD Dissertation of Nanjing University of Science and Technology.*
[27] Liu M., Lai J. Z, Liu J Y and Huang K. Fault diagnosis method of integrated GPS/Inertial navigation system based on support vector regression. Control and Decision, 2016, 31(10), pp. 1889-1893.

7 Appendix

7.1 Federal Kalman filter

7.1.1 Local filter: A 15-state Kalman filter is implemented in ENU frame as local filter. The state vector of the filter is

$$x = [\delta p^T \quad \delta v^T \quad \delta \theta^T \quad \delta \alpha^T \quad \delta \omega^T] \qquad (7)$$

where the first nine states, denoted with δp , δv , and

$\delta\theta$ are the position, velocity, and attitude errors defined in the ENU frame. The last six states, denoted by $\delta\alpha$ and $\delta\omega$, are the accelerometer and gyroscope bias errors.

Kalman filter consists of two steps; a prediction step and an update step. In the prediction process, the state and variance at time k+1 are estimated dependent on information at time k:

$$\hat{x}_{k+1/k} = F_k \hat{x}_k + G_k u_k \tag{8}$$

$$P_{k+1/k} = F_k P_k F_k^T + Q_k \tag{9}$$

where F_k is the state transition matrix, P_k and $P_{k+1/k}$ are the covariance matrix and the predicted covariance matrix, respectively, and Q_k is the covariance matrix of the process noise.

The update process is also known as measurement update. This updates the state and variance using a combination of the predicted state and the observation y_{k+1} at time k+1, such that

$$K_{k+1} = P_{k+1/k} H_{k+1}^T (H_{k+1} P_{k+1/k} H_{k+1/k}^T + R_{k+1})^{-1} \tag{10}$$

$$\hat{x}_{k+1} = \hat{x}_{k+1/k} + K_{k+1}(y_{k+1} - H_{k+1}\hat{x}_{k+1/k}) \tag{11}$$

$$P_{k+1} = (I - K_{k+1}H_{k+1})P_{k+1/k}(I - K_{k+1}H_{k+1})^T + K_{k+1}R_{k+1}K_{k+1}^T \tag{12}$$

where K_{k+1} is the Kalman gain, H_{k+1} the measurement matrices, R_{k+1} the measurement covariance matrices, and I the identity matrix.

In this paper, the state predictions are applied at each IMU observation, and GNSS and UWB systems provide the updates for two local Kalman filters, respectively. Both coordinate solutions from GNSS and UWB are transformed into the ENU frame, and are denoted as p_k. Thus, the measurement vectors for the local error-state Kalman filters are defined as

$$(y_k)_{GNSS} = (p_k)_{INS} - (p_k)_{GNSS} \tag{13}$$

$$(y_k)_{UWB} = (p_k)_{INS} - (p_k)_{UWB} \tag{14}$$

As position updates are provided in both local filters, the measurement matrices are

$$(H_k)_{GNSS} = (H_k)_{UWB} = [I_{(3\times3)} \quad 0_{(3\times12)}] \tag{15}$$

where 0 is zero matrices. It is worth noting that the $(R_k)_{GNSS}$ and $(R_k)_{UWB}$ measurement covariance matrices are defined based on the sensor specifications.

7.1.2 Global filter: Local optimal estimate $(\hat{x}_{k+1})_l$ and local optimal covariance $(P_{k+1})_l$ from local filters are sent to the global filter. Global optimal estimate $(\hat{x}_{k+1})_g$ and global optimal covariance $(P_{k+1})_g$ are achieved by applying information fusion technique shown as follows:

$$(P_{k+1}^{-1})_g = \sum_{l=1}^{N} (P_{k+1}^{-1})_l \tag{16}$$

$$(\hat{x}_{k+1})_g = (P_{k+1})_g \sum_{l=1}^{N} (P_{k+1}^{-1})_l \cdot (\hat{x}_{k+1})_l \tag{17}$$

where N is the number of local filters, l represents the l th local filter, and g means the global filter.

Authors

Weiyu Kong received his M. Eng from the School of Instrumentation Science and Engineering, Harbin Institute of Technology, Harbin, China, in 2021. He is currently working at CSIC. His current research interests include integrated navigation fault diagnosis, reliability test and strapdown inertial navigation technology.

Ya Zhang received her B.Sc. and Ph.D. from the Department of Automation, Harbin Engineering University, Harbin, China, in 2010 and 2015, respectively. She is currently a post-doctoral fellow at Harbin Institute of Technology. Her current research interests include strapdown inertial navigation technology, initial alignment algorithms, and high-precision inertial measurement technology.

Jiachong Chang received his B.Sc. from the Department of Automation, Harbin Engineering University, Harbin, China, in 2016, and his M.Sc. from the Harbin Institute of Technology, Harbin, in 2018, where he is currently pursuing the Ph.D. degree in instruments science and technology. His current research interests include strapdown inertial navigation systems and fault diagnosis technology