Hindawi
International Journal of Aerospace Engineering
Volume 2021, Article ID 6620368, 10 pages
https://doi.org/10.1155/2021/6620368

Research Article

Research on Feature Extracted Method for Flutter Test Based on EMD and CNN

Hua Zheng, Zhenglong Wu, Shiqiang Duan, and Jiangtao Zhou

School of Power and Energy, Northwestern Polytechnical University Shaanxi, Xi'an 710072, China

Correspondence should be addressed to Hua Zheng; isea_zh@mail.nwpu.edu.cn

Received 1 November 2020; Revised 5 February 2021; Accepted 13 February 2021; Published 28 February 2021

Academic Editor: Zhiguang Song

Due to the inevitable deviations between the results of theoretical calculations and physical experiments, flutter tests and flutter signal analysis often play significant roles in designing the aeroelasticity of a new aircraft. The measured structural response from aeroelastic models in both wind tunnel tests and real fight flutter tests contain an abundance of structural information, but traditional methods tend to have limited ability to extract features of concern. Inspired by deep learning concepts, a novel feature extraction method for flutter signal analysis was established in this study by combining the convolutional neural network (CNN) with empirical mode decomposition (EMD). It is widely hypothesized that when flutter occurs, the measured structural signals are harmonic or divergent in the time domain, and that the flutter modal (1) is singular and (2) its energy increases significantly in the frequency domain. A measured-signal feature extraction and flutter criterion framework was constructed accordingly. The measured signals from a wind tunnel test were manually labeled "flutter" and "no-flutter" as the foundational dataset for the deep learning algorithm. After the normalized preprocessing, the intrinsic mode functions (IMFs) of the flutter test signals are obtained by the EMD method. The IMFs are then reshaped to make them the suitable size to be input to the CNN. The CNN parameters are optimized though the training dataset, and the trained model is validated through the test dataset (i.e., cross-validation). The accuracy rate of the proposed method reached 100% on the test dataset. The training model appears to effectively distinguish whether or not the structural response signal contains flutter. The combination of EMD and CNN provides effective feature extraction of time series signals in flutter test data. This research explores the connection between structural response signals and flutter from the perspective of artificial intelligence. The method allows for real-time, online prediction with low computational complexity.

1. Introduction

Flutter is a destructive aeroelastic instability produced by the coupling of aerodynamic, inertial, and elastic forces in the elastic structure of an aeroelastic system. It is important to clarify the flutter characteristics of the aeroelastic system through the structural response signal to successfully complete a safe and effective aircraft design. Inherent limitations of the test environment and conditions give the actual, measured structural response signal low quality, low signal-to-noise ratio, modal density, non-stationary process characteristics, and a short window of effectiveness. The structure response signal and flutter criterion were investigated in this study from the perspective of signal analysis based on a deep learning algorithm and empirical mode decomposition (EMD) method in an effort to remedy these problems.

The most commonly used flutter signal processing method is damp prediction, where system stability is assessed by considering flutter as an instability phenomenon and an appropriate dynamic data model is established for subcritical test signals. The damping factor of at least one modal becomes zero when flutter occurs, so the point at which flutter occurs can be predicted by extrapolating the damping factor observed at different subcritical velocities [1]. The flutter margin method proposed by Zimmerman and Weissenberg [2] uses Routh's criterion to determine system stability conditions; limit point information is then obtained through sampled data, then a second-order equation describes the

relationship between the flutter margin and dynamic pressure. Price and Lee [3] effectively analyzed the flutter margin of a three-degree-of-freedom system, but the calculation is very complicated and the prediction error increases over the course of the analysis [4]. Similarly, Matsuzaki and Ando [5] established an autoegressive moving average model (ARMA) model using difference equations. After the autoregressive term coefficients are obtained through system identification, the Jury criterion is used to determine the stability of the system. However, for prediction purposes, this method is not suitable when the test data is in the subcritical region [6].

The Piston theory was proposed by Ashley and Zartarian [7] and has since been applied as an aerodynamic model by many researchers [8, 9]. Ganji et al. [10] pointed out the limitation of the traditional piston theory at low supersonic mach numbers and developed an improved piston theory to analyze the single-degree-of-freedom flutter of a panel. Cooper et al. [11] developed an envelope function method which judges system stability according to the shape parameters of the impulse response weakening envelope; their method can be used to predict flutter velocity by extrapolation from shape parameters calculated at different subcritical velocities. Cooper et al. [12] also developed an online envelope function method based on the online difference equation. However, this method takes a lot of time to converge to a stable estimate, giving it low online efficiency.

The aforementioned methods based on system stability are widely used in actual engineering practice, but they have significant drawbacks [13]. For example, the accuracy depends largely on the established model and various factors including low signal-to-noise ratio in the measured signal and a nonstationary structural response process. Traditional methods do not meet the requirements for modern structural (modal) time-varying aircraft design. Advancements in artificial intelligence have yielded artificial neural networks with strong feature extraction capability. Combining artificial intelligence with traditional flutter signal processing is a promising potential approach to a highly effective aircraft design. This paper introduces a data-driven deep learning method for extracting flutter characteristics from the structural response signals of aeroelastic models.

Computing power has improved considerably in recent years, which has led to new innovative technologies in many field. Song et al. [14] proposed a novel multipopulation parallel coevolutionary differential evolution to optimize the parameters of photovoltaic models and enhance the conversion efficiency of solar energy. Deng et al. [15] combined five mutation strategies to exploit their respective advantages in optimizing a differential evolution algorithm. In deep learning, a quantum-inspired differential evolution algorithm [16] with better global search ability was designed to update the parameters of deep belief network (DBN).

The convolutional neural network (CNN) is mainly used to process data in two-dimensional space such as images. The CNN works by deploying convolution and pooling operations to extract the input two-dimensional data features and ultimately to classify differently labeled data through the fully connected layer. Previous researchers have used the short-time Fourier transform (STFT) and CNN to extract features and classify flutter signals [17]. The time series of the flutter signal can be transformed by the STFT to obtain a time-frequency diagram. The time-frequency diagram is then input to the CNN, which performs feature extraction and classification. This combination has proven somewhat effective; however, the STFT has some notable drawbacks [18]:

(i) The Heisenberg-Gabor's uncertainty principle induces $\Delta f \cdot \Delta t \geq 1/4\pi$, so the time-frequency plane is not capable of both accurate time and frequency localization

(ii) This representation is not suitable for short duration/high-frequency and long-duration/low-frequency signals due to the fixed duration of the window function

The EMD directly performs time series decomposition, which avoids the STFT constraints. Xiao [19] combined EMD with CNN for the fault diagnosis of rolling bearings in offset printing machines. Xie and Zhang [20] calculated the characteristics of rolling bearing vibration signals by CNN and EMD and compared the fault classification effects of different classifiers. Yin et al. [21] classified impulse radio ultrawide band (IR-UWB) radar signals based on EMD and CNN for the diagnosis of heart rate abnormalities. Saini et al. [22] applied EMD to ECG signals for support vector machine classification for medical diagnosis. Liu et al. [23] used wavelet packet decomposition, EMD, and artificial neural networks for wind speed prediction. Qiu et al. [24] combined EMD and deep learning to predict the load demand of power companies. Semwal et al. [25] applied human health recovery based on EMD and deep neural networks. The features of time series calculated by the EMD can indeed be input to a CNN. Further, Zheng et al. [26, 27] used the EMD to process flutter signals and successfully extract their features.

The time series intrinsic mode function (IMF) obtained by EMD can be input to the CNN to perform feature extraction and realize signal classification. In this study, flutter test signal features were extracted based on the CNN from a large dataset to develop a trained model. The trained model can be used for the simultaneous monitoring of multichannel measured signals. The CNN and EMD were combined to process time series signals for flutter test signal analysis. The structural response signals markedly differ in the case of flutter versus when no flutter has occurred. We identified a distinct flutter signal characteristic based on the literature and our own analysis [26–28]. At the critical point of flutter, the dynamic structural response signal is harmonic or divergent in the time series as per the flutter test mechanism and signal characteristics. The flutter mode in the frequency domain tends to be singular, and the energy of the flutter signal significantly increases at this point. The CNN extracts features from the input information through a two-dimensional convolution operation, so the IMFs signal obtained by the EMD of the flutter test signal can be input to the CNN as an array of two-dimensional elements to extract features and perform an effective time series signal classification.

Zheng and Zhang [29] conducted a preliminary study on flutter signal analysis based on the three-layer back-propagation network, but no previous researcher has used CNN-based feature extraction to classify flutter signals or predict flutter boundaries. This paper proposes a method for classifying flutter signals using CNN and EMD. It is only necessary to obtain the IMFs of the flutter signal by the EMD; the IMF signals can then be resized so that they can be input to the CNN. The CNN extracts features through a series of convolution and pooling operations, then implements the classification using fully connected layers and loss function calculations.

2. Introduction to Proposed Method

The proposed method is operated in a three-step process. First, in order to minimize any artificial error arising in the acquisition of actual wind tunnel test data, the flutter test signal $x(n)$ is subjected to zero-average preprocessing to obtain the signal $y(n)$; then, $y(n)$ is subjected to EMD to obtain its IMFs. The time series of the flutter test signal is thus decomposed into multiple IMFs. The original information of the time series is saved while the time series is converted into a two-dimensional space. Next, the IMF signals in these two-dimensional spaces are reshaped into a two-dimensional array of suitable size for the CNN input. Finally, the CNN performs further feature extraction and distinguishes flutter test signal.

2.1. Flutter Signal Preprocessing. The flutter test signal $x(n)$ is normalized to preprocess the time series (Equation (1)). This serves to reduce the error in measured flutter wind tunnel test signals which emerges during the artificial acquisition process.

$$y(n) = \frac{x(n) - \text{mean}(x(n))}{\text{std}(x(n))}, \tag{1}$$

where $\text{mean}(x(n))$ is the mean of $x(n)$ and $\text{std}(x(n))$ is the standard deviation of $x(n)$.

2.2. Empirical Mode Decomposition. EMD uses the "screening" algorithm to decompose the signal into several IMF components and a remainder. The remainder is the trend term of the original signal, which is monotonous and smooth. Each IMF must satisfy the following two conditions:

(1) For the entire time series, the number of time series extreme value points is equal to the number of times series crossing the zero point, or at most has a difference of 1

(2) At any time, the average of the upper envelope consisting of the local maximum and the lower envelope consisting of the local minimum of the signal is zero

A flow chart of the EMD screening process is shown in Figure 1.

The algorithm is operated as follows, using the signal shown in Figure 2 as an example:

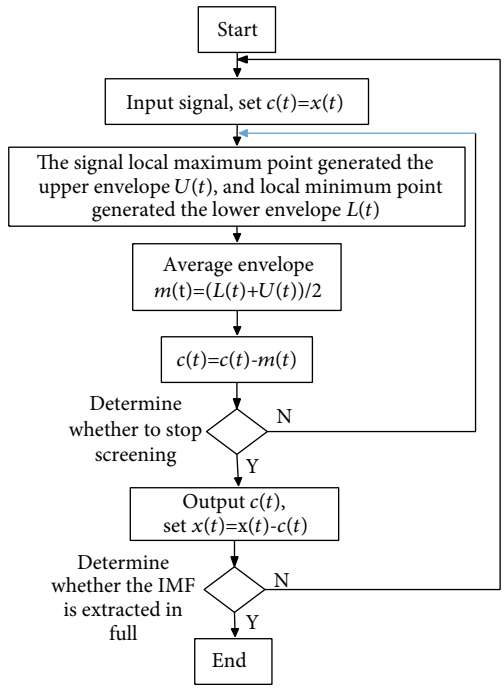

Figure 1: EMD method for screening IMFs.

(1) Calculate all extreme points (including maxima and minima)

(2) Interpolating the sequence of maxima and minima using the cubic square spline function method to obtain the upper envelopes $U(t)$ and lower envelopes $L(t)$ of the signal

(3) Calculate the average envelope, $m(t) = (U(t) - L(t))/2$, and extract the details of the signal, $c(t) = x(t) - m(t)$. As shown in Figure 3, the black line is the original line $x(t)$, the blue line is the upper envelope $U(t)$, the red line is the lower envelope $L(t)$, and the pink line is the average envelope $m(t)$

(4) Determine $c(t)$ to judge whether the two conditions of the IMF are met. If it is satisfied, $c(t)$ is the first IMF recorded as imf_1; if it is not satisfied, it is recorded $x(t) = c(t)$ and Step 1-4 are repeated until the condition is met

(5) Let $r_1 = x(t) - \text{imf}_i, r_1$ be the signal after the high-frequency component is removed from the original signal, which is in essence a new $x(t)$, then repeat the above steps until the remainder is a monotonic signal, and the decomposition ends

(6) Decompose $x(t)$ into I IMFs and a remainder $r_I(t)$, which is recorded as

$$x(t) = \sum_{i=1}^{I} \text{imf}_i + r_I(t), \tag{2}$$

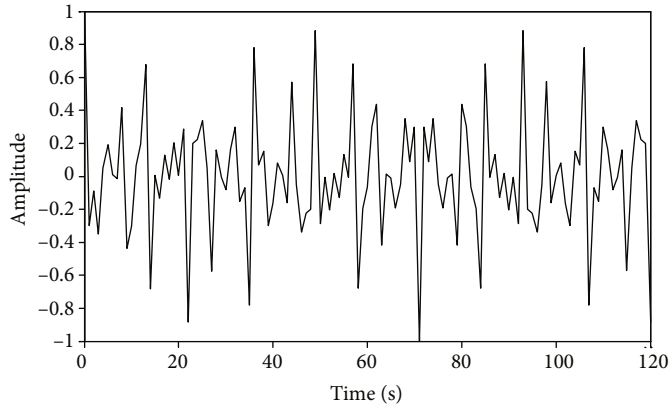

Figure 2: Original signal $x(t)$.

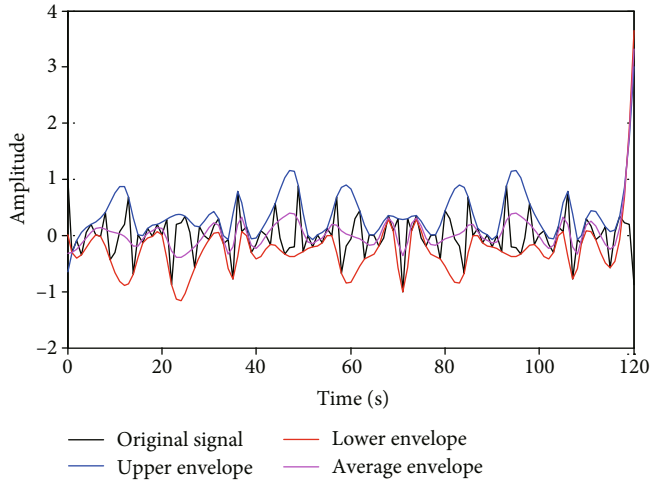

Figure 3: Calculating the average envelope.

where $r_I(t)$ is the remainder term, representing the average trend of the signal. The IMF components imf_1, imf_2, imf_3, $\cdots$, imf_I arranged according to the size of the scale, respectively, contain different components of the signal. The IMF and remainder after EMD are shown in Figure 4.

As shown in Figure 4, that after initial signal is filtered by EMD, several IMF components with gradually decreasing frequency can be obtained. These IMF components contain the local features of the original signal at different time scales.

2.3. Convolutional Neural Network. The CNN is an efficient deep learning feature extraction and information fusion method that has been developed in recent years and garnered widespread research attention. The CNN includes two basic calculations. The first is feature extraction, where the input of each neuron is connected to the local perception field of the previous layer and the local features are extracted. Once the local feature is extracted, its positional relationship with other features is also determined. The second is feature mapping. Each computing layer of the network consists of multiple feature maps. Each feature map is a layer, and the layers have equal weights. The ReLU function is used as the activation function of the convolution network, so the feature map

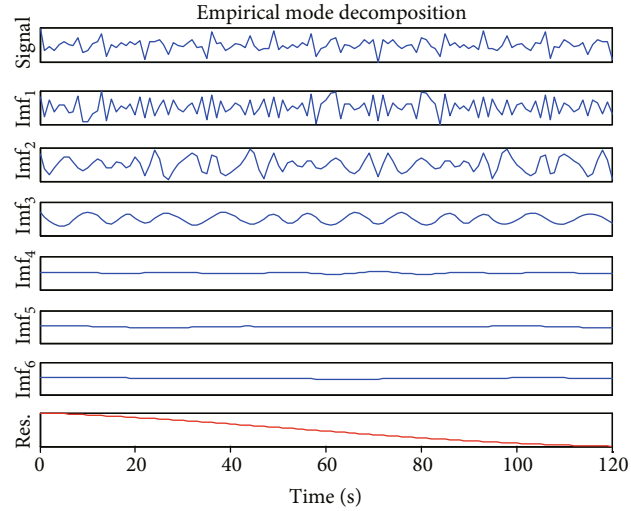

Figure 4: Results after EMD screening.

has displacement invariance. The neurons on any one mapping surface share weights, so there are relatively few free network parameters. These two operations form the convolutional layer of the CNN. This convolutional layer is followed

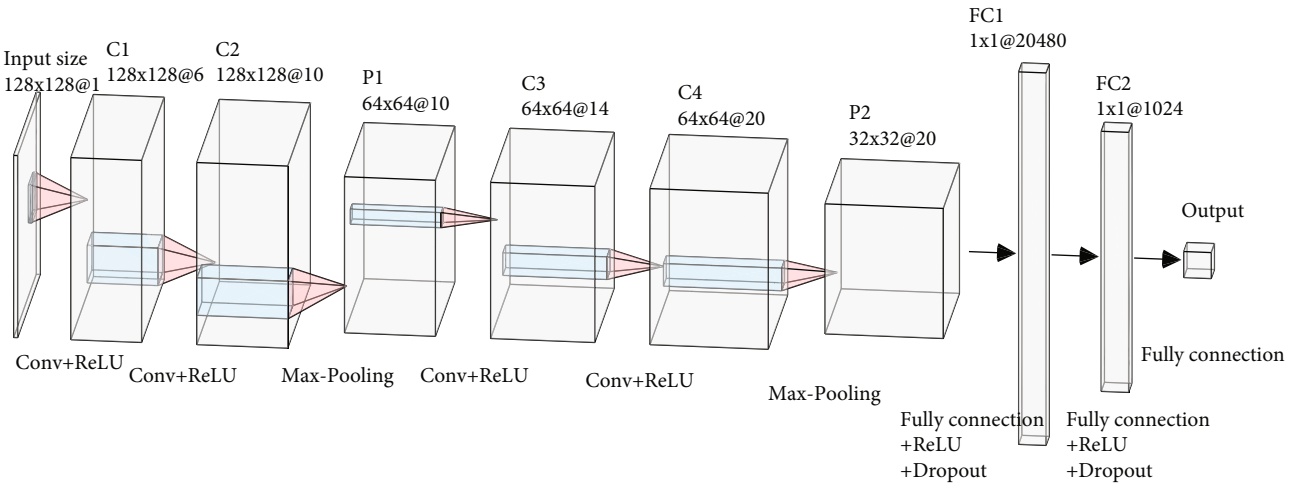

Figure 5: CNN architecture.

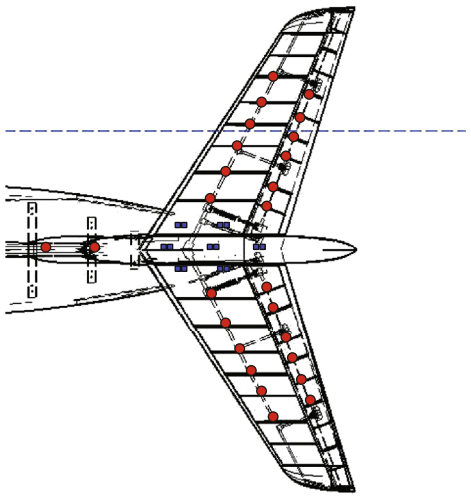

Figure 6: Sensor positions in aeroelastic model.

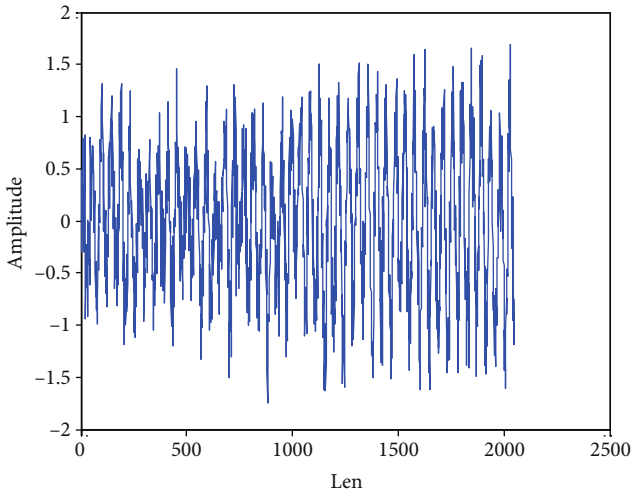

Figure 7: Signal time series when flutter occurs.

by a computational layer for local maximum and secondary extraction, namely, the pooling layer. This unique feature extraction structure minimizes the feature resolution.

Two sets of four convolutional layers and two pooling layers were designed for the proposed feature extraction method. The flutter test signals are classified using a three-layer fully connected network. The network uses the Adam optimizer to optimize parameters. Wu and Gu [30] proved that a dropout layer can improve neural network optimization effects and prevent overfitting, so we added a dropout layer behind the first two fully connected layers. The network structure is shown in Figure 5.

The CNN system input is the IMFs obtained by the EMD for the flutter test signals. The two-dimensional data of size $128 \times 128@1$ is first convolved through the first set of convolutional layers. The filter size of layer C1 is 5×5, the output channel is 6, the stride is 1, and the padding is 2. The feature layer of size $128 \times 128@6$ is obtained through the convolution layer C1. The convolution layer C2 has a filter size of 5×5, the output channel is 10, the stride is 1, and the padding is 2. A feature layer of size $128 \times 128@10$ is obtained through the convolution layer C2. A feature layer of $64 \times 64@10$ is obtained by max-pooling layer P1 with a filter size of 2×2.

The second set of convolutional layers is entered next. The filter size of the convolutional layer C3 is 3×3, the input channel is 10, the output channel is 14, the stride is 1, and the padding is 1. After the convolution layer C3 is a $64 \times 64@14$ feature layer. The convolution layer C4 has a filter size of 3×3,the input channel is 14, the output channel is 20, the stride is 1, and the padding is 1. A feature layer of size $64 \times 64@20$ is obtained through the convolution layer C4. The max-pooling layer P2 with filter size of 2×2 then provides the feature layer of $32 \times 32@20$. The following three fully connected layers are 20480×1024, 1024×512, and 512×2, respectively, which ultimately provide the flutter test signal result.

3. Flutter Test Datasets

3.1. Data Source. The training and test datasets used in this study were sourced from a set of aeroelastic model flutter wind tunnel test data. In the wind tunnel test, the data

acquisition system sensor of the aeroelastic model was positioned as shown in Figure 6: Seven strain measurements were taken from locations marked by blue rectangles (each including bending and torsional data) and 26 acceleration sensors were placed in the locations marked by red dots. Three strain measuring points were placed at the root of the vertical tail with another four points at the left and right horizontal tail, respectively. Two acceleration sensors were arranged on the rear fuselage. The remaining acceleration sensors were divided into four groups and arranged across the left horizontal tail, right horizontal tail, left elevator, and right elevator spanwise.

Original wind tunnel test data were recorded in full digital form on PXIe series data acquisition cards (American Virtual Instrument Company). Two sets of PXIe-4331 strain acquisition cards collected strain signals. Strain signals in one set of acquisition cards were sampled in eight channels, leaving six channels for the other set of cards. The strain signals thus had a total of 14 channels. Acceleration signals were sampled by two sets of PXIe-4499 acceleration acquisition cards with 16 channels in the first set and 10 channels in the second set of cards making for a total of 26 acceleration signals. All experimental data was acquired in real-time on a software driver in the LabVIEW environment. We set the basic sampling frequency of the flutter wind tunnel test to 200 Hz as per the relevant processing requirements and other parameter settings. The acceleration sensor is a 352C03 model (American PCB Company), and the strain sensor is a KFH prewire strain gauge (Omega).

There are 27 different statuses of aeroelastic models in the flutter wind tunnel test and a 180-steep wind (tunnel) speed. The excitation can be seen as atmospheric turbulence with various wind speeds. The data we used for deep learning was generated from the aeroelastic model described above. The flutter test data of different states and different speeds was selected with a data point length is 2048. The flutter test signals were decomposed into IMFs by EMD algorithm. The first 8 IMFs reshaped the two-dimensional data into 128 ∗ 128 size for input to the CNN. The parameters were trained; then, the effectiveness and accuracy of the trained CNN model were evaluated as discussed below.

3.2. Dataset. The acquired structural responded signal from an aeroelastic model is a time series of strain and acceleration signals. The time series data needs to be converted into a two-dimensional space to be classified by the CNN. As explained above, we decomposed the time series into multiple IMFs by the EMD so that the original information could be saved and to make the original time series suitable for inputting to the CNN.

A known flutter wind tunnel test signal was first labeled as one of two states, flutter or no-flutter (Figures 7 and 8), as an artificial dataset. The first 8 IMFs obtained by the EMD method (Section 2.2), as shown in Figures 9 and 10, were selected as the CNN dataset. There are 2048 data points for each flutter test signal. The 8 IMFs have 2048×8 data points that were reshaped into a two-dimensional array of 128×128 for training and test datasets to suit the CNN.

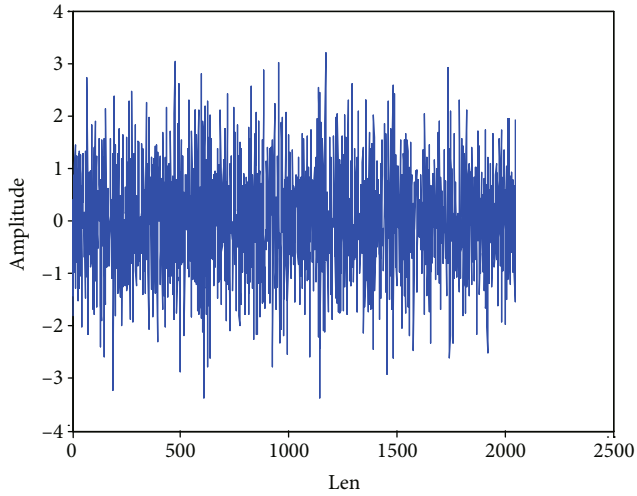

Figure 8: Signal time series with no flutter.

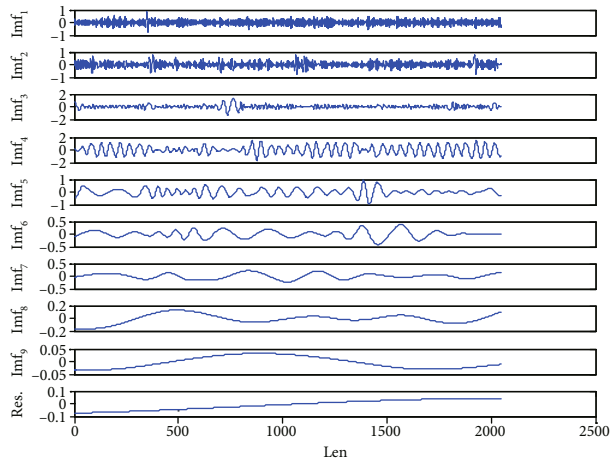

Figure 9: IMFs of signal when flutter occurs.

The label "1" indicates flutter and "0" indicates no-flutter. The total dataset includes 1000 sets of flutter and no-flutter data (80% for training and 20% for testing).

As shown in Figures 7 and 8, the signal time series when flutter occurs is markedly different from the no-flutter signal time series. Figures 9 and 10 show that the EMD results of the two types of signal are also very different, which meets the precondition of the proposed method for classifying flutter test signals.

4. Data Training and Test Results

4.1. Training Model Evaluation Criteria. The CNN model was trained in the Pytorch deep learning framework. We used the loss function and accuracy rate as the training and testing criteria. When the loss function value is minimal and the accuracy rate is maximal, the parameters have been effectively optimized and the results of the model are valid.

4.1.1. Training Loss. We used the cross-entropy function (Equation (3)) to calculate the error between the forward feedback result of the CNN and the label of the flutter test

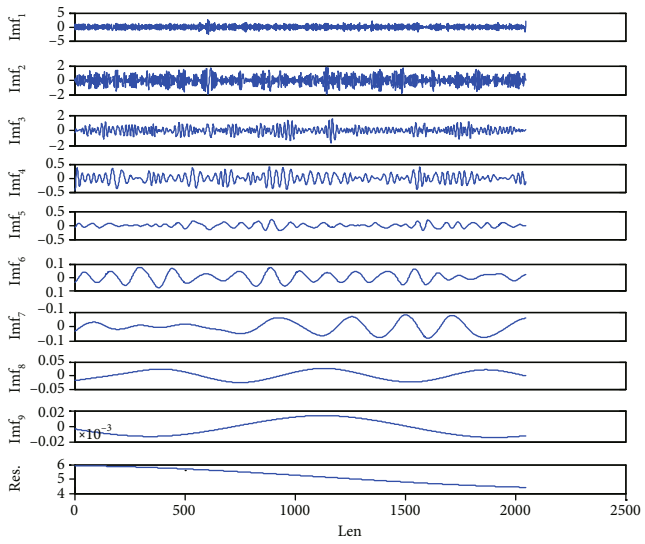

Figure 10: IMFs of no-flutter signal.

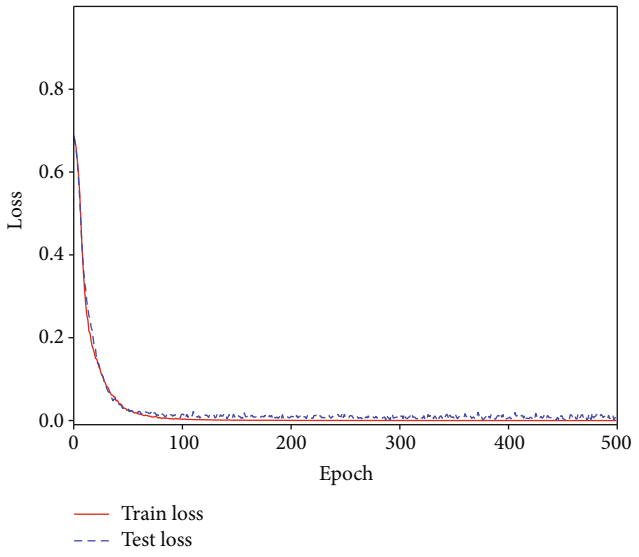

Figure 11: Loss curve.

signal. The cross-entropy function value was updated through the gradient descent and optimization function.

$$\text{loss}\left(y_{\text{pred}}, y_{\text{label}}\right) = -\frac{1}{n}\sum_{\text{class}}\left[y_{\text{pred}} \ln y_{\text{label}} + \left(1 - y_{\text{pred}}\right) \ln\left(1 - y_{\text{label}}\right)\right], \tag{3}$$

where $\text{loss}(y_{\text{pred}}, y_{\text{label}})$ is the loss function, y_{label} represents the labeled results of the IMFs of flutter signals, y_{pred} is the CNN output, and class indicates the class to which the label needs to be divided; in our case, class = [0, 1]. n denotes the dimension of the training data.

4.1.2. Accuracy Rate. In the latter stage of the training process, the accuracy rates of training datasets are calculated

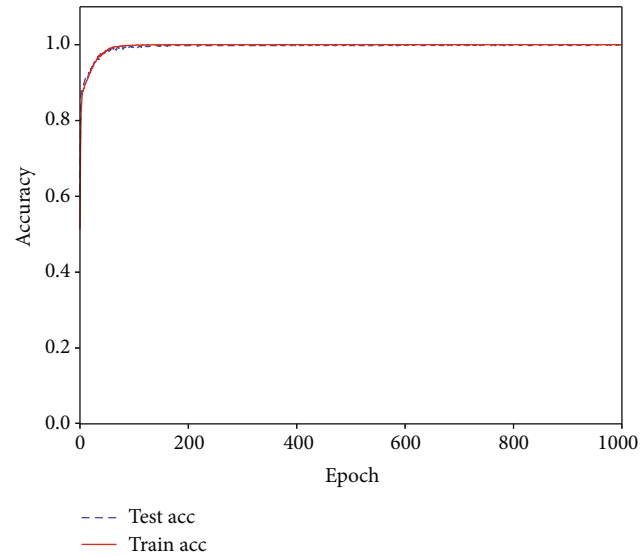

Figure 12: Accuracy curve.

after each epoch. Flutter versus no-flutter signals are distinguished in the IMFs under the current weights and bias of the CNN. When the loss function is minimal, the weights and biases can be expended as the values of IMF feature extractions in the CNN.

Here, the models constituting the CNN architecture and its parameters were saved for reliability verification. The accuracy rate was calculated as follows:

$$\text{acc} = \frac{\text{sum}\left(y_{\text{pred}} == y_{\text{label}}\right)}{N}, \tag{4}$$

where acc is the accuracy rate, y_{pred} is the output of the flutter test signal of the CNN, y_{label} is the labeled flutter test signal, the sum function calculates the same number between y_{pred} of the flutter results output by the CNN and y_{label} of the labeled flutter signals, and N is the dimension of y_{pred}.

4.2. Training and Test Results. According to the requirements of deep learning algorithms, we used two criteria for the CNN: the loss function and accuracy rate. The CNN was trained on the dataset described above. After comparing various hyperparameters, the hyperparameter was set to 1000 iterations and the learning rate to 0.00001. The loss function and accuracy rate were optimal when the two dropout layer parameters were 0.5. The batch size value was set to 100 according to the computer memory and model size. The loss function and accuracy rate curves were calculated by taking the average value of each batch size as shown in Figures 11 and 12, respectively.

The loss curve and the accuracy rate curve show that the loss function drops to a satisfactory value within 100 epochs and that accuracy ultimately reaches 100%. The two datasets of four convolution layer convolutions and two pooling layers appear to have effectively extracted the features of the flutter test signal IMFs.

Table 1: Model validation results.

	Train	Test
Loss value	1.0490417423625331*e*-07	0.00019828170479740947
Accuracy rate	100%	100%

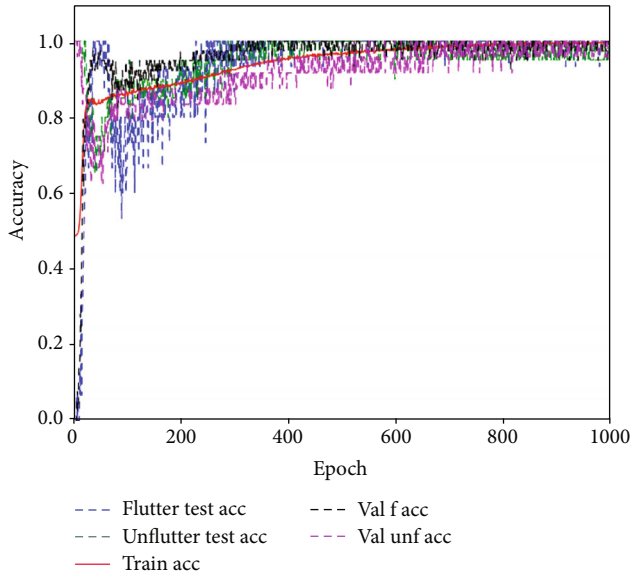

Figure 13: STFT-CNN accuracy curve.

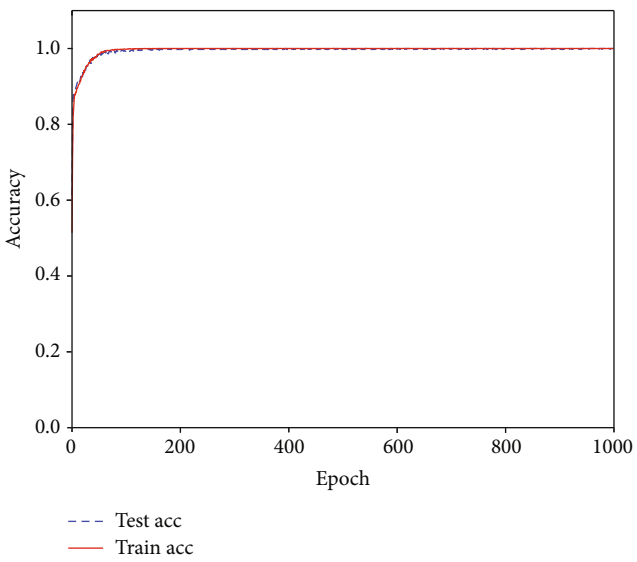

Figure 14: EMD-CNN accuracy curve.

Table 1 lists the calculation results for the 1000th iteration. The reliability and effectiveness of the proposed model were confirmed by the deep learning training dataset and test dataset.

4.3. Comparison of Methods. We tested the superiority of our EMD-CNN method compared to the STFT-CNN [14]. The input of the CNN method is a two-dimensional spatial array.

Table 2: Comparison of two methods.

	STFT-CNN	EMD-CNN
Training dataset loss	0.016996	1.0490417423625331e-07
Test dataset loss	0.091222	0.00019828170479740947
Accuracy rate	95.5%	100%

The STFT-CNN method was operated by first performing an STFT transform on the flutter time series to obtain a spectrogram of the time series signal to satisfy the CNN input. The input size for the STFT-CNN was 32×32 in this case. The flutter data used for training and verification was the same as the wind tunnel test data for the aeroelastic model described in Section 3.1. The STFT-CNN method uses two convolutional layers and two pooling layers to perform feature extraction and then classifies the data through three fully connected layers. We used four convolutional layers and two pooling layers for feature extraction, then completed the classification with three fully connected layers. The CNN input for the EMD-CNN method was 128×128. The accuracy rate curves of the two methods are shown in Figures 13 and 14.

As shown in Figure 13, although the training accuracy curve is relatively stable, the test accuracy curve and validation accuracy curve show significant fluctuations. When the accuracy of the training set reaches 100%, the accuracy of the test set and validation set is only between 90% and 95%. The accuracy curve of the test set reaches its highest level after 400 epochs. The accuracy curve of the validation set reaches its highest level after 800 epochs. The accuracy curve shown in Figure 14 is relatively stable, which indicates that the proposed network structure can effectively and robustly extract the features of IMF signals. Both accuracy curves reach 100% within 200 epochs, suggesting that the proposed method also has a faster convergence rate during training than the STFT-CNN.

Table 2 lists the accuracy rates of STFT-CNN and EMD-CNN after the 1000th iteration. The proposed method shows less loss and better classification effects than the STFT-CNN. Our algorithm has two more convolution layers than the STFT-CNN, which appears to positively impact the feature extraction effectiveness.

5. Conclusion

A feature extraction method for flutter test signals was developed in this study based on EMD and CNN. The accuracy rate of the proposed method was found to reach up to 100%. Compared with the STFT-CNN method, our method has higher accuracy in fewer iterations. It appears that more convolution layers are helpful with regard to feature extraction.

This paper mainly describes our processing method for transforming flutter test signals into a two-dimensional space for input to the CNN for feature extraction. The EMD, which has advantages in nonlinear and nonstationary signal analysis, was chosen as the preliminary feature extraction method of the time series signal. The IMFs can be transformed to satisfy the two-dimensional convolution operation, so that the convolution network can effectively extract the features through the two-dimensional data. This method is feasible and effective. The CNN training process requires numerous calculations (a common drawback of neural networks), but after the training is complete, the network essentially performs only simple matrix operations. Therefore, real-time online prediction can be achieved with low computational complexity. This has great significance for ensuring navigational safety and for accurately monitoring flutter test states in wind tunnel tests.

In this study, we completed only an initial flutter test signal classification. It is still necessary to determine how to calculate the modal parameters (e.g., damping of structural response signals) by the deep learning algorithm. We plan to further predict the flutter speed through these parameters in the future as well. The dataset used to evaluate the deep learning algorithm also had some limitations that may have caused issues with the universality of the model.

Data Availability

The entire flutter test dataset is excessively large, so only a part of signals are uploaded to supplementary information files. The entire dataset used to support the findings of this study can be available from the second author at 2020201938@mail.nwpu.edu.cn.

Additional Points

Highlights. (i) EMD and CNN are combined to extract features from measured flutter test signals and determine whether or not flutter occurs. (ii) The EMD method is an adaptive scheme for nonlinear nonstationary time series analysis. (iii) The data-driven neural network is trained and tested. (iv) Real-time, online prediction can be achieved with low computational complexity.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities (Grant no. 31020190MS702) and National Science and Technology Major Project (Grant no. 2017-V-0011-0062).

Supplementary Materials

The supplementary material file is a portion of the experimental flutter test data used to produce the dataset (Section 3) that we used to verify our method. The data were sourced from strain and acceleration sensors on an aeroelastic model in a wind tunnel test. A part of data with no flutter is given in .doc format. Every .doc file comtains 8 IMFs after the channel signal of the acceleration or strain sensor is decomposed by EMD, where the numerical sequence represents a discrete time sequence. *(Supplementary Materials)*

References

[1] G. Hongtao, *Research on Data Processing Technology of Flutter Test*, China Aerodynamics Research and Development Center, Mianyang, 2005.

[2] N. H. Zimmerman and J. T. Weissenburger, "Prediction of flutter onset speed based on flight testing at subcritical speeds," *Journal of Aircraft*, vol. 1, no. 4, pp. 190–202, 1964.

[3] S. J. Price and B. H. K. Lee, "Evaluation and extension of the flutter-margin method for flight flutter prediction," *Journal of Aircraft*, vol. 30, no. 3, pp. 395–402, 1993.

[4] W. W. Zhang, H. S. Huashou, and H. Xiao, "Review and prospect of flutter boundary prediction methods for flight flutter testing," *Acta Aerodynamica Sinica*, vol. 36, no. 5, pp. 1367–1384, 2015.

[5] Y. Matsuzaki and Y. Ando, "Estimation of flutter boundary from random responses due to turbulence at subcritical speeds," *Journal of Aircraft*, vol. 18, no. 10, pp. 826–868, 1981.

[6] T. Ueda, M. Iio, and T. Ikeda, "Flutter prediction using continuous wavelet transform," *Transactions of the Japan Society for Aeronautical and Space Sciences*, vol. 51, no. 174, pp. 275–281, 2009.

[7] H. Ashley and G. Zartarian, "Piston theory: a new aerodynamic tool for the aeroelastician," *Journal of the Aeronautical Sciences*, vol. 23, no. 12, pp. 1109–1118, 1956.

[8] D. D. Liu, Z. X. Yao, D. Sarhaddi, and F. Chavez, "From piston theory to a unified hypersonic supersonic lifting surface method," *Journal of Aircraft*, vol. 34, no. 3, pp. 304–312, 1997.

[9] D. Xie, M. Xu, and E. H. Dowell, "Proper orthogonal decomposition reduced-order model for nonlinear aeroelastic oscillations," *AIAA Journal*, vol. 52, no. 2, pp. 229–241, 2014.

[10] H. F. Ganji and E. H. Dowell, "Panel flutter prediction in two dimensional flow with enhanced piston theory," *Journal of Fluids and Structures*, vol. 63, pp. 97–102, 2016.

[11] J. E. Cooper, P. R. Emmet, J. R. Wright, and M. J. Schofield, "Envelope function - a tool for analyzing flutter data," *Journal of Aircraft*, vol. 30, no. 5, pp. 785–790, 1993.

[12] J. E. Cooper, M. J. Desforges, and J. R. Wright, "The online envelope function-aguide to aeroelastic stability," in *International Forum on Aeroelasticity and Structural Dynamics*, pp. 981–998, Reston, 1993.

[13] R. Lind, "Flight-test evaluation of flutter prediction methods," *Journal of Aircraft*, vol. 40, no. 5, pp. 964–970, 2015.

[14] Y. Song, D. Wu, W. Deng et al., "MPPCEDE: multi-population parallel co-evolutionary differential evolution for parameter optimization," *Energy Conversion and Management*, vol. 228, article 113661, 2021.

[15] W. Deng, J. Xu, Y. Song, and H. Zhao, "Differential evolution algorithm with wavelet basis function and optimal mutation strategy for complex optimization problem," *Applied Soft Computing*, vol. 100, article 106724, 2020.

[16] W. Deng, H. Liu, J. Xu, H. Zhao, and Y. Song, "An improved quantum-inspired differential evolution algorithm for deep

belief network," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 10, pp. 7319–7327, 2020.

[17] S. Duan, H. Zheng, and J. Liu, "A novel classification method for flutter signals based on the CNN and STFT," *International Journal of Aerospace Engineering*, vol. 2019, Article ID 9375437, 8 pages, 2019.

[18] O. Poisson, P. Rioual, and M. Meunier, "New signal processing tools applied to power quality analysis," *IEEE Transactions on Power Delivery*, vol. 14, no. 2, pp. 561–566, 1999.

[19] X. Liangjun, *Research on fault diagnosis method of rolling bearing rolling bearing based on convolutional neural network*, Xi'an University of Technology, 2018.

[20] Y. Xie and T. Zhang, "Fault diagnosis for rotating machinery based on convolutional neural network and empirical mode decomposition," *Shock and Vibration*, vol. 2017, 12 pages, 2017.

[21] W. Yin, X. Yang, L. Zhang, and E. Oki, "ECG monitoring system integrated with IR-UWB radar based on CNN," *IEEE Access*, vol. 4, pp. 6344–6351, 2016.

[22] I. Saini, D. Singh, and A. Khosla, "Electrocardiogram beat classification using empirical mode decomposition and multiclass directed acyclic graph support vector machine," *Computers & Electrical Engineering*, vol. 40, no. 5, pp. 1774–1787, 2014.

[23] H. Liu, X. Mi, and Y. Li, "Comparison of two new intelligent wind speed forecasting approaches based on wavelet packet decomposition, complete ensemble empirical mode decomposition with adaptive noise and artificial neural networks," *Energy Conversion and Management*, vol. 155, pp. 188–200, 2018.

[24] X. Qiu, Y. Ren, P. N. Suganthan, and G. A. J. Amaratunga, "Empirical mode decomposition based ensemble deep learning for load demand time series forecasting," *Applied Soft Computing*, vol. 54, pp. 246–255, 2017.

[25] V. B. Semwal, K. Mondal, and G. C. Nandi, "Robust and accurate feature selection for humanoid push recovery and classification: deep learning approach," *Neural Computing and Applications*, vol. 28, no. 3, pp. 565–574, 2017.

[26] H. Zheng, J. Liu, and S. Duan, "Online EMD-wavelet method for flutter test excited by atmospheric turbulence," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 33, no. 7, article 1950010, 2019.

[27] H. Zheng, J. Liu, and S. Duan, "Flutter test data processing based on improved Hilbert-Huang transform," *Mathematical Problems in Engineering*, vol. 2018, Article ID 3496870, 8 pages, 2018.

[28] Z. Shanhong, *Method of time frequency syntony for flutter boundary prediction*, Northwestern Polytechnical University, 2002.

[29] H. Zheng and J. Zhang, "A stability criterion method based on neural network and its application on flutter boundary prediction," *Engineering Intelligent Systems*, vol. 24, no. 3-4, pp. 99–104, 2016.

[30] H. Wu and X. Gu, "Max-pooling dropout for regularization of convolutional neural networks," in *Neural Information Processing. ICONIP 2015. Lecture Notes in Computer Science, vol 9489*, S. Arik, T. Huang, W. Lai, and Q. Liu, Eds., pp. 46–54, Springer, Cham, 2015.