S & M 4117

Application of AI Deep Learning in Aircraft Shape Analysis and Validation for High-performance Numerical Wind Tunnel

Ching-Huei Huang,[1]* Kun-Lin Tsai,[2] and Shih-Ting Tseng[2]

[1]Department of Automatic Control Engineering, Feng Chia University,
No. 100, Wenhua Rd., Xitun Dist., Taichung City 407102, Taiwan (R.O.C.)
[2]Department of Electrical Engineering, Tunghai University,
No. 1727, Sec. 4, Taiwan Boulevard, Xitun District, Taichung City 407224, Taiwan (R.O.C.)

(Received February 27, 2025; accepted May 8, 2025)

Keywords: aircraft shape design, AI, backpropagation neural network, deep learning, numerical wind tunnel

Aircraft represent one of the most significant technological advancements in human history. Aircraft not only provide convenient and rapid transportation, but also serve as critical tools for exploring the natural world. During the early stages of aircraft design, aerodynamic coefficients must undergo meticulous calculations and analysis to ensure the development of safe, reliable, and efficient aircraft. Generally, aerodynamic coefficients are obtained through numerical simulations using computational fluid dynamics and wind tunnel testing. These methods are effective and accurate; however, they often consume significant amounts of time and resources. With advancements in AI, deep learning techniques have been increasingly applied in aerodynamics research. To reduce the time and cost spent on wind tunnel testing during the aircraft design phase, we apply deep learning techniques to high-performance numerical wind tunnels to analyze aerodynamics in the aviation field. A backpropagation neural network model with an error compensation mechanism is created to enhance the efficiency of analyzing and validating aerodynamic coefficients. This approach minimizes the reliance on physical wind tunnel testing, thereby reducing overall development costs.

1. Introduction

In recent years, with the rapid development of modern information technology, the development of AI has become one of the most revolutionary technologies, with its applications expanding across numerous fields, including smart home appliances(1,2) and autonomous vehicle systems.(3,4) The extensive applications of AI have not only changed people's lifestyles but also enhanced convenience, driving innovation and breakthroughs across various areas.

Among these applications, the integration of AI with aerodynamics in the aviation field has been increasingly utilized by experts and scholars for the design, analysis, and validation of aircraft configurations.(5,6) Aerodynamics, which is the study of the behavior and interaction of airflow, plays a critical role in influencing aircraft performance and design. By merging

*Corresponding author: e-mail: chueihuang@o365.fcu.edu.tw

https://doi.org/10.18494/SAM5598

advancements in AI with aerodynamic research, it is possible to optimize and accelerate the prediction of aircraft designs and performance, further advancing the development of aviation technologies.

Generally, during the preliminary design stage of aircraft, aerodynamic coefficients are primarily obtained through computational fluid dynamics (CFD) by numerical simulations and wind tunnel tests.(7,8) Although these methods are both effective and accurate, they often consume a lot of time and resources. CFD software requires a large number of computational resources and takes a long time to run. Completing a numerical simulation for an aircraft design involves analyzing multiple shape iterations across different stages, which can take weeks to months, depending on the complexity of the design requirements and the computational capacity available. Wind tunnel tests provide high-precision data; however, the process is constrained by the need to first fabricate a physical model for each specific configuration. The hardware of wind tunnels is expensive; moreover, the experiments are also costly and time-consuming. As a result, wind tunnel testing is primarily intended for final design validation rather than for use during early-stage iterative design processes.(9)

Although the Stanford University unstructured (SU2) open-source software provides a new analysis tool for aerodynamics,(10) its interface is not intuitive, and its complex flow field setup increases time costs. To address these challenges, reduce costs, improve efficiency, and maintain high accuracy, in this study, we employed deep learning algorithms to train neural network models. These models leverage the interpretability and real-time prediction capabilities of AI to rapidly learn and estimate aerodynamic coefficients, thereby establishing a valuable numerical wind tunnel. The numerical wind tunnel developed in this study, powered by AI and deep learning, significantly streamlines the aircraft design cycle by reducing reliance on extensive physical testing, improving overall efficiency, and minimizing both labor and cost. Furthermore, applying AI technology to aircraft shape analysis and validation represents a promising interdisciplinary advancement.

2. Related Literature

2.1 Numerical wind tunnel development

The numerical wind tunnel is a wind tunnel testing method developed and studied using AI techniques.(11) It is generally considered superior to CFD technology and SU2 software analysis. The numerical wind tunnel can be used to evaluate and analyze the aerodynamic performance of aircraft under various flight conditions. Compared with traditional wind tunnel testing, the numerical wind tunnel offers detailed data and analysis results with significantly lower cost and time investment. It has gradually become a critical technology in modern aircraft design.

The preliminary design of an aircraft primarily determines its basic configuration based on performance requirements and mission specifications. Figure 1 shows the aircraft shape design process. The dimensions of the aircraft are calculated on the basis of wing area and aerodynamic coefficients, and the obtained results can generate the external mold lines. After obtaining the complete aerodynamic data set (ADS), it will be verified whether the previously established

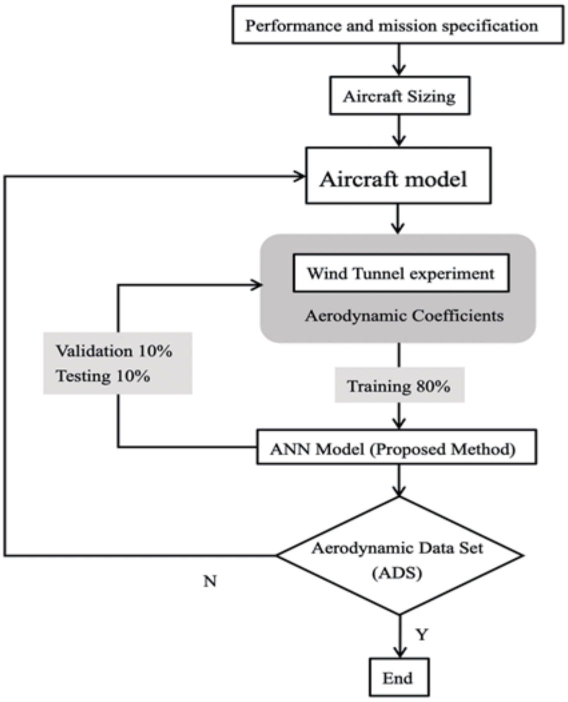


Fig. 1. Flowchart of aircraft shape design used for aerodynamic data analysis with neural network model.

performance requirements and mission specifications are met. If satisfied, the preliminary aircraft design is considered complete, and the next step of developing and validating the flight control laws can proceed. However, aircraft designs typically do not meet all the requirements on the first attempt. If the requirements are not met, adjustments to the aircraft shape will be made, and the design process will continue iteratively until the performance requirements and mission specifications are fully satisfied.

Before developing a numerical wind tunnel, it is essential to understand the definitions of aerodynamic forces and moments acting on the test model. Figure 2 shows an aircraft tested under relative wind conditions. The lines represent the body axes and the stability axes in three-dimensional space (X, Y, Z axes), following the conventional right-hand body coordinate system.(12)

In this system, X represents the horizontal axis, Y represents the axis perpendicular to X, and Z represents the axis perpendicular to both X and Y, whereas V represents the direction of the relative wind. The angle α refers to the angle of attack, defined as the angle between the X body-axis and the X stability-axis. The angle β, known as the sideslip angle, is defined as the angle between the X-axis of the direction of the relative wind and the X stability-axis in the wind tunnel. The transformation equations from body axes to stability axes are shown in Eq. (1).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\mathrm{STAB}} = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\mathrm{BODY}} \tag{1}$$

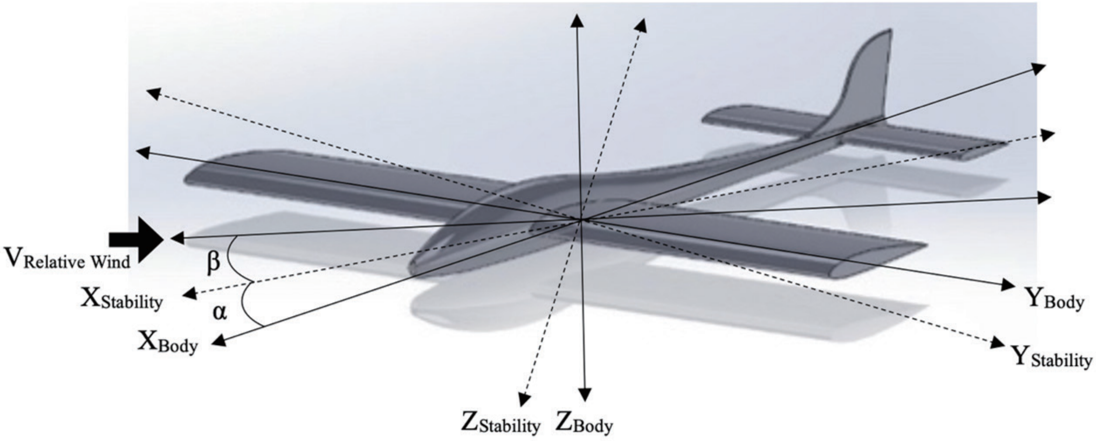


Fig. 2. (Color online) Definitions of aerodynamic coefficients and coordinate system.

The transformation equations from stability axes to wind axes are shown in Eq. (2).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\mathrm{WIND}} = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\mathrm{STAB}} \tag{2}$$

Thus, the general transformation of any vector between the body and wind axes is shown in Eq. (3).

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\mathrm{WIND}} = \begin{bmatrix} \cos\alpha\ \cos\beta & \sin\beta & \sin\alpha\ \cos\beta \\ -\cos\alpha\ \sin\beta & \cos\beta & -\sin\alpha\ \sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{\mathrm{BODY}} \tag{3}$$

Equations (1) to (3) are appropriately transformed into the corresponding coordinate systems and applied to the numerical wind tunnel. By integrating AI techniques and utilizing machine learning and deep learning methods, the efficiency and accuracy of simulations can be significantly enhanced. The development process of the numerical wind tunnel typically includes the following steps: data collection, model training, model validation and optimization, as well as simulation and analysis. These steps will be discussed in detail in the subsequent sections.

In aircraft design, the numerical wind tunnel can be utilized to enhance aerodynamic shape optimization, reduce design iteration time, and evaluate whether the aerodynamic coefficients meet performance and mission requirements. This approach improves design efficiency and minimizes the costs associated with physical wind tunnel testing. As a highly effective tool for aerodynamic analysis, the numerical wind tunnel accelerates the development of modern aircraft designs. It enables the rapid and precise evaluation and optimization of aircraft shapes, significantly improving overall efficiency. The flowchart of aircraft shape design used for aerodynamic data analysis with a neural network model is shown in Fig. 1.

2.2 AI

AI primarily aims to enable machines to mimic human cognitive processes, such as thinking, learning, reasoning, decision-making, and planning, allowing them to autonomously solve problems and make decisions across various tasks.(13) The core concept of AI is to empower computer systems to learn from experience, adapt to new data, and manage complex tasks.

The development of AI can be traced back to the 1950s when computers began to gain popularity in society. During that time, people became passionate about training computers in various computational methods, aspiring to create AI. Over the following decades, AI experienced several periods of stagnation and resurgence. Early AI was largely based on rules and symbolic logic, such as expert systems. Around the 2000s, with the rise of machine learning and statistical methods, AI technology began to advance.

However, the most significant breakthroughs occurred in the 21st century, driven by advancements in computational power, the emergence of graphics processing units (GPUs), the integration of software and hardware technologies, the advent of big data, and the maturity of deep learning techniques. These factors enabled AI to evolve rapidly. By late 2022, OpenAI launched ChatGPT, a conversational AI model, reigniting interest in AI and drawing widespread attention to the field. ChatGPT itself is a product of large-scale natural language processing models within the domain of AI.

The rapid development of modern AI is primarily driven by three indispensable elements: large-scale databases, advanced algorithms, and high-speed computational capabilities. Together, these factors enable AI to achieve widespread applications across various fields, including speech recognition, image processing, machine translation, autonomous driving, and medical diagnosis. AI has already become one of the most influential and promising technologies in the world today, significantly impacting the development of society and the global economy.

Machine learning(14) is a subfield of AI that focuses on enabling computers to learn and analyze data details autonomously through statistical methods. This technology identifies patterns and rules from data and continuously improves performance as experience grows, thereby accomplishing various tasks and objectives. In machine learning, model training involves providing data and labels, allowing the model to learn how to make accurate predictions or classifications.

Deep learning(15,16) is a subfield of machine learning that mimics the structure and functioning of the human brain's neural networks to accomplish highly complex data processing tasks. The defining characteristic of deep learning lies in its multilayered neural network architecture, which can automatically extract features and make decisions, deriving higher-level insights from input data.

Artificial neural networks in deep learning can be trained using numerical simulation data from CFD and wind tunnel tests as inputs. This enables the development of rapid and accurate models that can optimize efficient aerodynamic shape designs and identify optimal configurations.(17)

Deep learning has been widely used in numerous fields. Common deep learning models include convolutional neural networks (CNNs), recurrent neural networks (RNNs), and backpropagation neural networks (BPNNs), all of which have become core technologies in many applications.

For instance, CNN and other deep learning architectures can be trained to predict complex flow fields around aircraft shapes by learning underlying physical knowledge from simulation or experimental data.(17,18) This approach provides valuable insights into flow patterns, separation, and other aerodynamic phenomena, which can help design analysis and optimization. Through the technology and advantages of AI deep learning, the accuracy of aerodynamic analysis can be improved to effectively reduce the development time of aircraft shape design.

2.3 BPNNs

Backpropagation is a core technique in artificial neural networks, first introduced in the 1986 paper "learning representations by back-propagating errors" by Rumelhart *et al.*(19) This technique effectively addresses a significant challenge in training multilayer neural networks—how to efficiently update network weights to enhance prediction accuracy. By utilizing backpropagation, deep learning neural networks can self-learn from training data and adjust weights on the basis of the error between actual and expected outputs, minimizing errors. This process is the key to the development of modern deep learning in aerodynamic parameter systems.

The backpropagation architecture involves two main stages: forward propagation and backward propagation. The key focus in backward propagation is adjusting weights based on predicted outputs to improve prediction accuracy. The primary goal of backpropagation is to adjust network weights by minimizing the loss function, which evaluates the difference between the prediction results of this neural network model and the actual results. Common loss functions include the mean squared error (MSE) and the mean absolute error (MAE_1).

MSE: The calculation involves subtracting the predicted output value of the model from the actual target value, squaring the difference, summing up the squared differences, and then averaging the total. The formula is shown in Eq. (4) below.

$$MSE=\frac{1}{n}\sum_{i=1}^{n}\left(Y_i-Y_i'\right)^2 \tag{4}$$

The concept of this formula is that for each input data, the error between the actual target value Y_i and the predicted output value of the model Y_i' (i.e., $Y_i - Y_i'$) is computed. This error is then squared, and the squared errors for all data are summed up and divided by the total number of input data (n), yielding the average error.

The reason for squaring the error is to prevent the positive and negative errors from canceling each other out. However, owing to the squaring characteristic, MSE is also particularly sensitive to extreme values. A small MSE indicates that the prediction results are close to the actual values, implying high model accuracy. In contrast, a large MSE suggests considerable errors between the predictions and the actual values, reflecting poorer model performance.

MAE_1: The calculation involves taking the error between the actual target and predicted output values of the model, computing the absolute value of this error, and then averaging it. The formula is shown in Eq. (5) below.

$$MAE_1 = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - Y_i'\right| \tag{5}$$

The concept of the formula is similar to that of *MSE*, with the key difference being that while *MSE* squares the errors, MAE_1 takes the absolute values of the errors. This approach provides a more direct reflection of the actual situation of the errors. Moreover, since MAE_1 involves the absolute value of the errors, it avoids the issue of error magnification caused by squaring, which occurs in *MSE*, making MAE_1 less sensitive to extreme values. Conversely, a large MAE_1 suggests considerable errors between predictions and actual values, implying poor model performance.

Backpropagation relies on the gradient descent method, which iteratively adjusts weight parameters to minimize prediction errors and loss values. The learning rate determines the step size for updating weights. If the learning rate is very high, the model may fail to converge or skip the global minimum. Otherwise, a learning rate that is very low may result in slow convergence or getting trapped in local minimum. Thus, selecting an appropriate learning rate is important for a successful gradient descent method.

BPNNs are one of the indispensable technologies in deep learning. By updating weights layer by layer, they transform complex input data into precise output results. This capability allows for learning aerodynamic characteristics from input aircraft shape parameters and aerodynamic conditions. The method has been extensively studied and developed in the integration of AI with aerodynamics, demonstrating significant potential for advancing research in this field.

Several studies have demonstrated the application of BPNN in numerical wind tunnels.(20–22) For example, Balla *et al.*(23) proposed a BPNN-based method to predict aerodynamic characteristics under various flight conditions. Their research showed that BPNN-based predictions not only reduced computation time but also adapted to different aerodynamic conditions, providing accurate results.

In 2020, Huang *et al.*(9) investigated the effects of additional aircraft shape parameters on aerodynamic coefficients. They used parameters such as aspect ratio, angle of attack, Mach number, sideslip angle, Reynolds number, tank size, taper ratio, and sweep angle as input features for BPNN. The target outputs were lift and drag coefficients. However, they did not account for varying aerodynamic phenomena under different Mach numbers. Therefore, in this research, we incorporated Mach number as a critical feature in the experimental stage of the AI deep learning model, conducting segmented testing based on aerodynamic theory to enhance prediction accuracy.

With advancements in deep learning, BPNN exhibits significant potential in numerical wind tunnel applications. This method enables neural networks to effectively learn and predict aerodynamic characteristics in aircraft shape design, simultaneously improving computational efficiency and reducing the costs of wind tunnel testing. In contrast to a previous study,[9] the present deep learning research incorporates the parameters within the ADS framework.

3. Deep Learning BPNN Method

We utilized aerodynamic data to construct a neural network model for deep learning, employing BPNN to establish an error compensation model for improving prediction accuracy. By this method, reliable aerodynamic data can be learned and predicted from historical wind tunnel tests. The expected outcome is to reduce the number of wind tunnel tests required while providing more accurate and reliable aerodynamic data, which can serve as a tool for AI-based aerodynamic analysis and significantly shorten the iterative design cycle.

The comprehensive table of aerodynamic coefficients for flight control laws, known as ADS, is analyzed and evaluated on the basis of validated aerodynamic data corresponding to various flight attitudes. The numerical wind tunnel system incorporates ADS parameters, which define key aerodynamic features. These coefficients are processed using neural network and deep learning techniques for shape prediction and analysis. These attitudes and analysis parameters involving these reliable data are shown in Table 1. The parameters include 10 feature values, namely, Mach number (*Mach*), angle of attack (*AoA*, α), angle of sideslip (*AoS*, β), leading edge flap (*LEF*), trailing edge flap (*TEF*), aileron asymmetry (*dA*), horizontal tail symmetry (*dHS*), horizontal tail asymmetry (*dHA*), rudder symmetry (*dRS*), and rudder asymmetry (*dRA*). The target outputs are the lift coefficient (C_L) and drag coefficient (C_D).

The architecture of BPNN is shown in Fig. 3. This method constructs an error compensation model comprising three layers: input, hidden, and output layers. The input layer contains a total of 10 features, selected on the basis of the parameters adjusted during wind tunnel tests for different aircraft configurations. Therefore, these 10 fundamental test parameters were included as features in the model.

A particular parameter among them is the Mach number. The other nine features represent variations in the attitude angles of aircraft, whereas the Mach number plays a crucial role in both aerodynamics and aircraft design. It directly reflects the relationship between the flight speed and the speed of sound, and aerodynamic phenomena behave differently under different Mach numbers. We ultimately adopted a segmented training approach based on Mach number to better capture variations in lift and drag coefficients, thereby reducing prediction errors, improving accuracy, and providing more reliable data.

Table 1
Summary of ADS parameters used in the numerical wind tunnel system.

Abbreviation	ADS parameter	Abbreviation	ADS parameter
Mach	Mach number	*dHS*	Horizontal tail symmetry
AoA (α)	Angle of attack	*dHA*	Horizontal tail asymmetry
AoS (β)	Angle of sideslip	*dRS*	Rudder symmetry
LEF	Leading edge flap	*dRA*	Rudder asymmetry
TEF	Trailing edge flap	C_L	Lift coefficient
dA	Aileron asymmetry	C_D	Drag coefficient

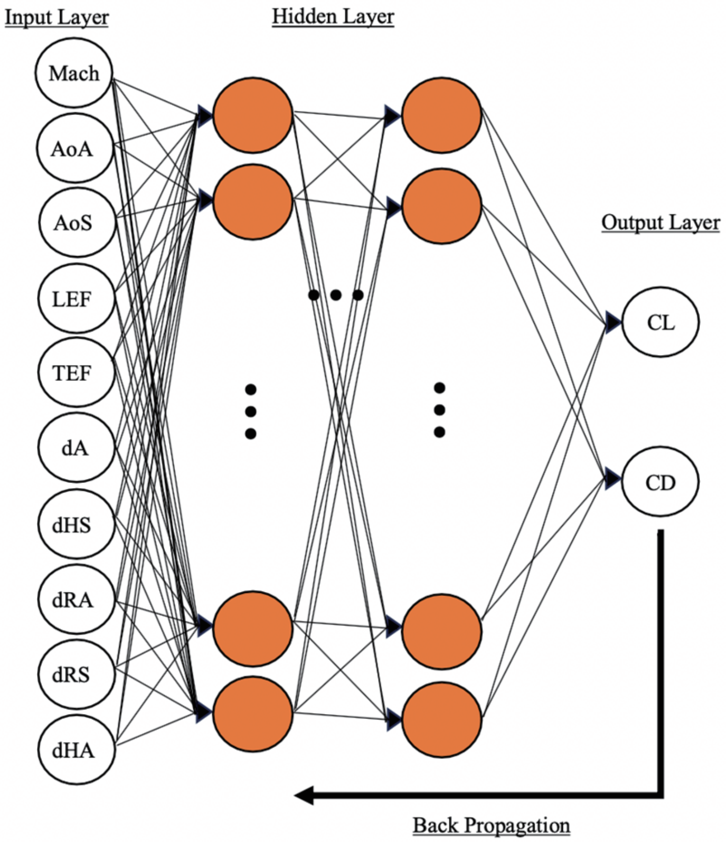


Fig. 3. (Color online) Architecture of ADS applied to BPNNs.

The optimal numbers of hidden layers and neurons will be determined on the basis of experimental results in the next section. The output layer generates two parameters, the lift coefficient and the drag coefficient. The lift coefficient affects flight stability, directly affecting whether the aircraft can maintain stable flight. The drag coefficient, on the other hand, impacts flight efficiency and fuel consumption, making it a critical indicator for minimizing energy losses in aircraft design. The balance between the lift and drag coefficients determines overall aircraft performance, including endurance, fuel efficiency, and stability. These coefficients serve as essential design references to meet flight requirements effectively.

The model's fundamental setup includes a maximum training iteration of 30000, representing the upper limit of training cycles. Once this iteration count is reached, training stops. The learning rate is set to 0.01, which controls the step size for weight updates. Ultimately, the model aims for a target training error of near zero, which means that during the training process, if the error value is reached, the training will stop. However, this goal is idealistic, as it is nearly impossible to achieve a completely error-free state in practice.

The model aims to enhance the accuracy of aerodynamic parameter predictions by minimizing errors during the prediction process. It achieves this by iteratively updating weights through the backpropagation framework. Through iterative training, the neural network learns and predicts reliable aerodynamic parameters, providing a credible source of aerodynamic coefficients for flight vehicle control laws. This approach significantly reduces reliance on wind tunnel tests, which are resource-intensive and time-consuming.

For example, during testing, time and manpower are required to calibrate the balance accuracy to ensure that the measured aerodynamic data fall within the acceptable error range. The obtained data must also undergo data processing procedures before being used for further analysis. Additionally, prior to testing, the model manufacturing for different aircraft configurations must be completed, which typically takes no less than six months. Therefore, the results of this study can save costs and time during the aerodynamic shape design and development process.

The deep learning method helps to obtain the comprehensive aerodynamic database for subsequent flight control law development and validation. These aerodynamic parameters can be evaluated for future flight conditions. By incorporating the actual aerodynamic parameters from flight control into a neural network architecture comprising input, hidden, and output layers, each layer is interconnected with subsequent layers through weighted connections. During the training process, the varied weight distributions between layers enable the model to learn complex, nonlinear relationships among aerodynamic parameters. This approach achieves data modeling and predictive capabilities, ultimately aiming to construct a complete ADS.

4. Experimental Results

4.1 Experimental environment

In these experiments, a computer with 64 GB of memory was employed to handle the primary computational tasks. This memory capacity is sufficient to meet the data processing requirements of most academic studies, particularly those involving complex numerical simulations and deep learning model training. Although this hardware configuration meets the demands of our research, ensuring system stability and efficiency becomes crucial when dealing with large-scale datasets or concurrent multi-experiment operations. The specific hardware specifications used in this study are detailed in Table 2.

The MATLAB toolbox is used to construct and train neural network models, and its powerful matrix computation capabilities are employed for large-scale data processing and analysis. During the experiments, the parallel computing function of MATLAB effectively took the advantages of multicore processors, accelerating the training process and producing results within a reasonable time. The choice of software and hardware environments in this study ensures the feasibility of the experiments and the reliability of the results.

Table 2
Computer hardware specifications.

Item	Specification
Operating system	Windows 10
CPU	Intel i9-14900KF
GPU	RTX-4070TI-O12G
RAM	DDR4 64 GB

4.2 Effect of different activation functions on prediction accuracy

In the first experiment, the prediction accuracies of lift coefficients using different activation functions were compared, including linear functions, sigmoid functions, and hyperbolic tangent (tanh) functions. Their respective equations are shown in Eqs. (6) to (8).

$$f\left(x\right)=x \tag{6}$$

$$\sigma\left(x\right)=\frac{1}{1+e^{-x}} \tag{7}$$

$$\tanh\left(x\right)=\frac{2}{\left(1+e^{-2x}\right)-1} \tag{8}$$

The number of hidden layers was fixed at four, the number of neurons was set to 30, and the learning rate was maintained at 0.01. Each activation function was applied individually to evaluate its impact on prediction accuracy. The red line represents the predicted values generated by the neural network model, whereas the blue line indicates the actual values measured in wind tunnel tests. The *X*-axis denotes the angle of attack, tested in the range from −8 to 18°, whereas the *Y*-axis represents the lift coefficient. The Mach numbers covered in the experiments span from subsonic to transonic to supersonic.

To evaluate the performance more effectively, we adopted *MSE*, MAE_1, and maximum absolute error (MAE_2) as evaluation criteria, with their equations provided in Eqs. (9) to (11).

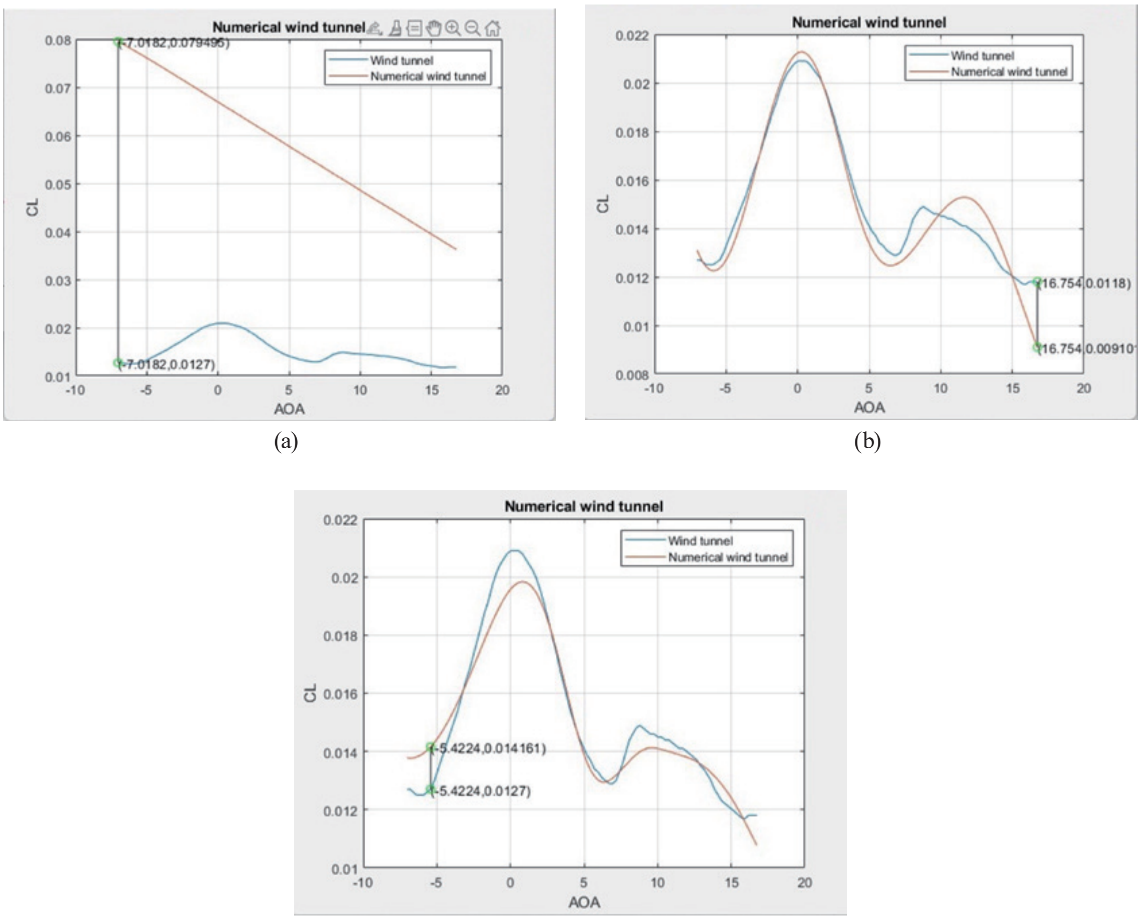
$$MSE=\frac{1}{n}\sum_{i=1}^{n}\left(Y_i-Y_i'\right)^2 \tag{9}$$

$$MAE_1=\frac{1}{n}\sum_{i=1}^{n}\left|Y_i-Y_i'\right| \tag{10}$$

$$MAE_2=\max\left(\left|Y_i-Y_i'\right|\right) \tag{11}$$

For error analysis, 100 data points were extracted for comparison between the trained model and the actual results. In Fig. 4, the green dots indicate the angle of attack corresponding to the largest error between the predicted and actual values, connected by a black line for clarity. For instance, in Fig. 4(a), using the linear function, the largest error occurred at an angle of attack of approximately −7.0°, with a predicted value of 0.0795 and an actual value of 0.0127.

In Fig. 4(b), using the sigmoid function, the largest error occurred at an angle of attack of approximately 16.8°, with a predicted value of 0.0091 and an actual value of 0.0118. In Fig. 4(c),



(a)

(b)

(c)

Fig. 4. (Color online) Using different activation functions in lift coefficient prediction case study. (a) Linear function, (b) sigmoid function, and (c) tanh function.

the largest error for the hyperbolic tangent (tanh) function occurred at an angle of attack of approximately −5.4°, with a predicted value of 0.0142 and an actual value of 0.0127.

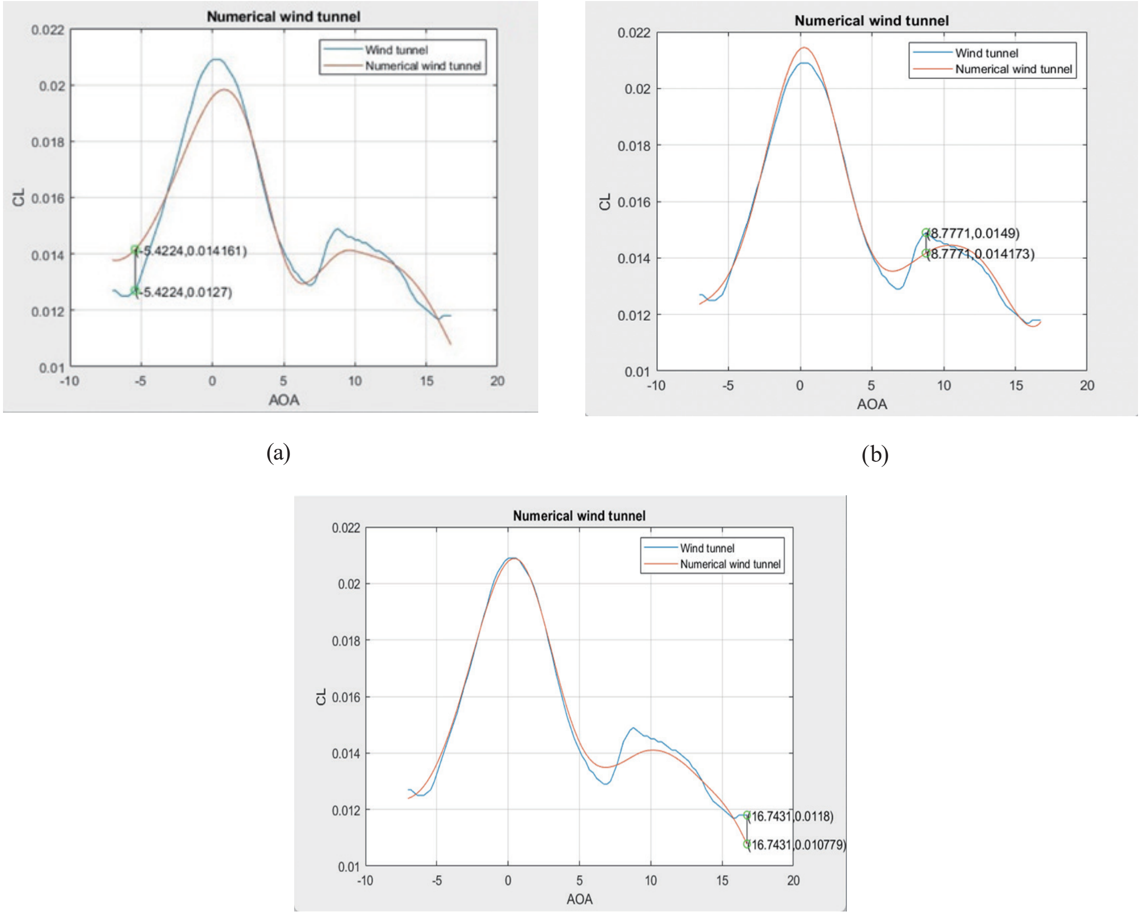
Here, Y_i represents the actual aerodynamic data from wind tunnel tests, whereas Y_i' represents the predicted output of the model. When $Y_i = Y_i'$, the model achieves perfect prediction accuracy. Small *MSE* and MAE_1 values indicate high model accuracy. Additionally, MAE_2 as an evaluation index for capturing special flow field characteristics was used in this study. The smaller the MAE_2 value, the greater the capability to capture special flow field characteristics.

The results showed significant differences between the lift coefficient predictions of the linear function and the wind tunnel testing data. This is attributed to the inherently nonlinear characteristic of lift coefficient prediction, making the linear activation function unsuitable for aerodynamic coefficient predictions. The *MSE* for the linear function reached approximately 1.90×10^{-3}, MAE_1 was about 4.21×10^{-2}, and MAE_2 was about 6.68×10^{-2}.

It was also observed that the errors for the sigmoid function are larger than those for the tanh function in *MSE*, MAE_1, and MAE_2. This is because the output range of the sigmoid function is restricted between 0 and 1, limiting its ability to represent the data characteristics accurately. *MSE* for the sigmoid function was approximately 7.66 × 10^{-7}, MAE_1 was about 6.63 × 10^{-4}, and MAE_2 was about 2.70 × 10^{-3}. Consequently, the hyperbolic tangent (tanh) function was selected as the optimal activation function for the next stage. Its *MSE* was approximately 4.95 × 10^{-7}, its MAE_1 was about 5.54 × 10^{-4}, and its MAE_2 was about 1.46 × 10^{-3}, all of which were lower than the results obtained using the linear and sigmoid functions. It can clearly be seen that this type of nonlinear parameter analysis is not applicable to the linear activation function.

4.3 Effect of different numbers of neurons in hidden layer on prediction accuracy

In the second experiment, the prediction accuracies of lift coefficients using different numbers of neurons in the hidden layers were compared, as shown in Fig. 5. Neural networks with 30, 40, and 50 neurons per hidden layer (with the number of layers fixed) were used to predict the lift coefficient at various angles of attack. The number of hidden layers was fixed at four, the learning rate was set to 0.01, and the activation function was selected as the hyperbolic



(a)

(b)

(c)

Fig. 5. (Color online) Using different numbers of neurons in lift coefficient prediction case study. (a) 30, (b) 40, and (c) 50 neurons.

tangent (tanh) on the basis of the optimal result in the previous experiment, to compare prediction accuracy.

The predicted values obtained using the proposed method were compared with the aerodynamic coefficients from wind tunnel tests. As in the previous experiment, the red line represents the predicted values from the numerical wind tunnel, whereas the blue line represents the actual values from physical wind tunnel tests. The X-axis denotes the angle of attack, tested in the range from −8 to 18°, and the Y-axis represents the lift coefficient. The Mach numbers covered in the experiments span from subsonic to transonic to supersonic. For error evaluation, 100 data points were extracted for comparison between the model predictions and the actual results, and the differences in prediction accuracy were found by adjusting the number of neurons.

Figure 5(a) shows the results obtained using 30 neurons. A notable difference exists between the predicted and testing results, indicating that the model has not yet converged. The MSE for using 30 neurons was approximately 4.95×10^{-7}, MAE_1 was about 5.54×10^{-4}, and MAE_2 was approximately 1.46×10^{-3}, with the largest error occurring at an angle of attack of −5.4°.

In contrast, the graphs and trends shown in Figs. 5(b) and 5(c) appeared similar to the experimental results. On the basis of Table 3, for 40 neurons, MSE was approximately 8.62×10^{-8}, MAE_1 was approximately 2.32×10^{-4}, and MAE_2 was approximately 7.27×10^{-4}, with the largest error occurring at an angle of attack of 8.8°. For 50 neurons, MSE was approximately 1.45×10^{-7}, MAE_1 was approximately 2.86×10^{-4}, and MAE_2 was approximately 1.02×10^{-3}, with the largest error occurring at an angle of attack of 16.7°.

These results indicate that using 40 neurons was more accurate than using 50 neurons. This demonstrates that increasing the number of neurons does not necessarily improve accuracy and may lead to overfitting. Therefore, the optimal setting of 40 neurons was selected for the next experiment.

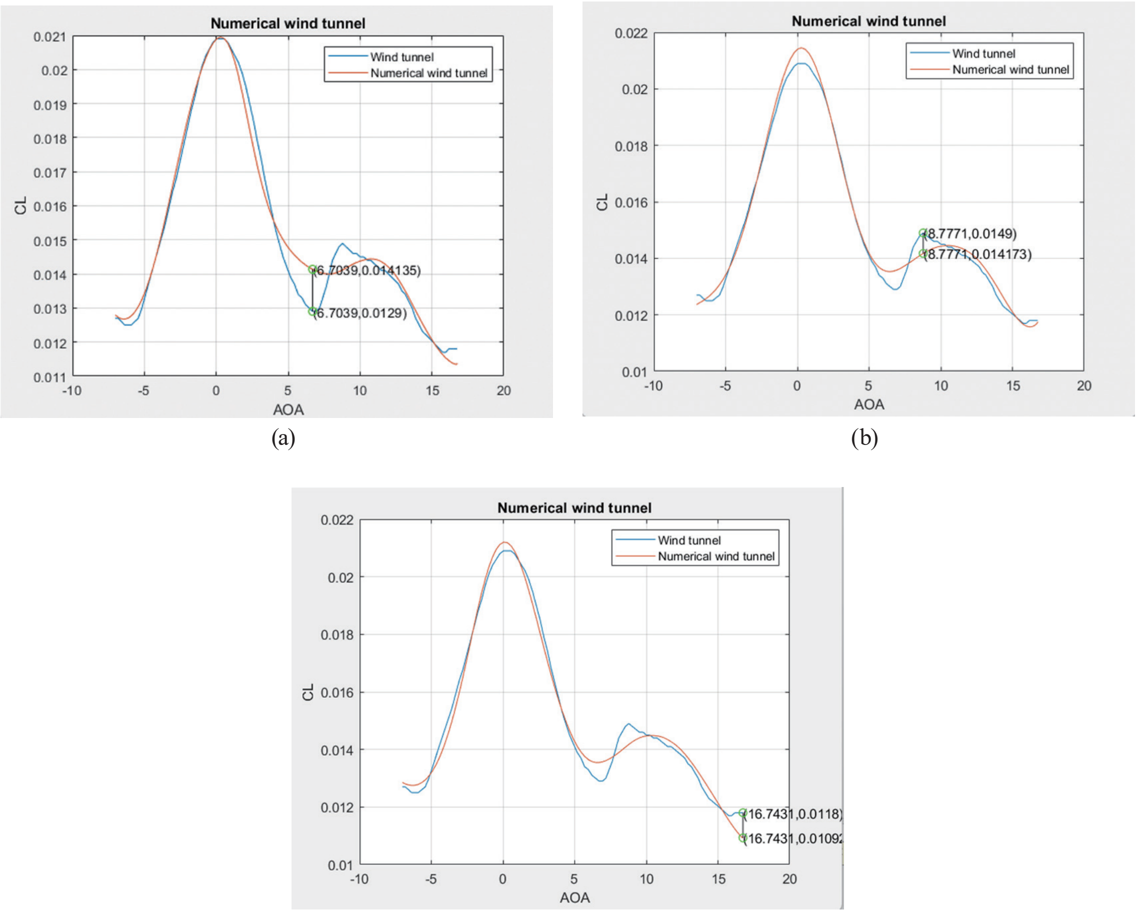
4.4 Effect of different numbers of hidden layers on prediction accuracy

In the third experiment, the prediction accuracies of lift coefficients using different numbers of hidden layers were compared. In this experiment, we used three, four, and five hidden layers, as shown in Fig. 6, to obtain the lift coefficients at different angles of attack. The activation function and the number of neurons were fixed as the optimal results from the previous experiments, namely, the hyperbolic tangent (tanh) function and 40 neurons, to evaluate the prediction accuracy.

The predicted values obtained using the proposed method were compared with the aerodynamic coefficients derived from wind tunnel tests. The red line represents the predicted

Table 3
Error comparison of different numbers of neurons.

Number of neurons	MSE	MAE_1	MAE_2
30	4.95×10^{-7}	5.54×10^{-4}	1.46×10^{-3}
40	8.62×10^{-8}	2.32×10^{-4}	7.27×10^{-4}
50	1.45×10^{-7}	2.86×10^{-4}	1.02×10^{-3}



(a)

(b)

(c)

Fig. 6. (Color online) Using different numbers of hidden layers in lift coefficient prediction case study. (a) Three, (b) four, and (c) five hidden layers.

values from the numerical wind tunnel, whereas the blue line corresponds to the actual values from physical wind tunnel tests. The *X*-axis indicates the angle of attack, ranging from -8 to 18°, and the *Y*-axis denotes the lift coefficient. The Mach numbers covered in the experiments span from subsonic to transonic to supersonic.

Upon comparing the graphs and trends produced by the three different numbers of hidden layers, we observed that the results are similar. A detailed comparison of the errors shown in Table 4 reveals that, for Fig. 6(a) with three hidden layers, *MSE* was approximately 2.02×10^{-7}, MAE_1 was about 3.21×10^{-4}, and MAE_2 was about 1.24×10^{-3}, with the largest error occurring at an angle of attack of 6.7°. In Fig. 6(b), using four hidden layers, *MSE* was approximately 8.62×10^{-8}, MAE_1 was about 2.32×10^{-4}, and MAE_2 was around 7.27×10^{-4}, with the maximum error occurring at an angle of attack of 8.8°. For Fig. 6(c), using five hidden layers, *MSE* was approximately 1.29×10^{-7}, MAE_1 was about 2.95×10^{-4}, and MAE_2 was approximately 8.71×10^{-4}, with the largest error at an angle of attack of 16.7°. From these results, the errors were smaller for the four hidden layers than for the three and five hidden layers. Therefore, for the final experiment, the optimal setting of four hidden layers was selected for training.

Table 4
Error comparison of different numbers of hidden layers.

Number of hidden layers	*MSE*	MAE_1	MAE_2
Three	2.02×10^{-7}	3.21×10^{-4}	1.24×10^{-3}
Four	8.62×10^{-8}	2.32×10^{-4}	7.27×10^{-4}
Five	1.29×10^{-7}	2.95×10^{-4}	8.71×10^{-4}

4.5 Mach number segmentation for training on prediction accuracy

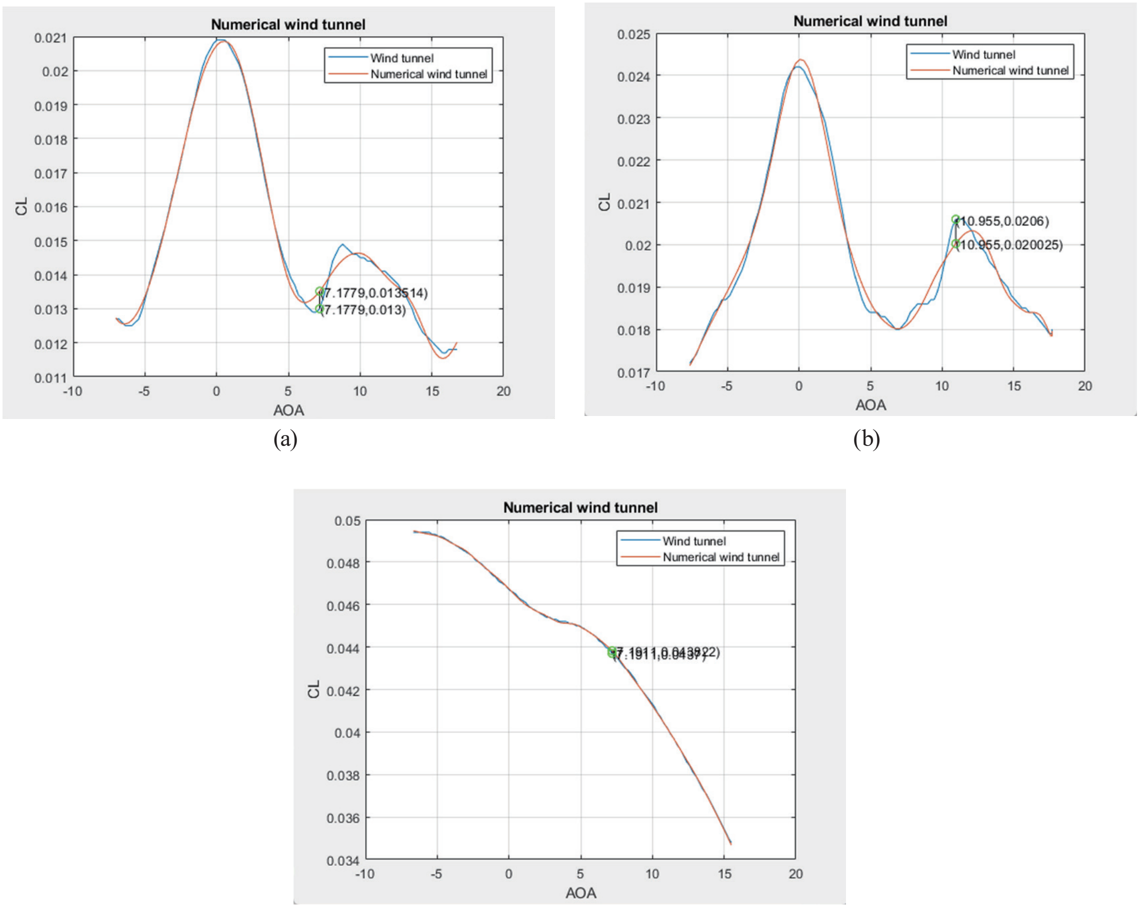
In the fourth experiment, on the basis of the results above, the optimal settings were selected: the hyperbolic tangent (tanh) function as the activation function, 40 neurons, and four hidden layers. *MSE* was reduced to less than 1×10^{-7}, and MAE_1 was reduced to less than 3×10^{-4}. Under the existing feature values, it was found that training the model in segments based on different Mach numbers would further enhance the prediction accuracy and bring the trends closer to the actual values.

Mach number is the ratio of the speed of an aircraft to the speed of sound in air. Generally, aircraft cruising at high altitudes are controlled by the Mach number, which is a critical factor in aerodynamics and forms the academic foundation for the development of supersonic flight in aerospace technology.

Figure 7 shows the lift coefficient performance at different angles of attack for subsonic (Mach number less than 0.85), transonic (Mach number between 0.85 and 1.25), and supersonic (Mach number greater than 1.25). The red line represents the predicted values from the numerical wind tunnel, whereas the blue line indicates the actual wind tunnel test values. The *X*-axis represents the angle of attack, ranging from −8 to 18°, and the *Y*-axis represents the lift coefficient. In Fig. 7(a), using subsonic data to train the model, *MSE* was approximately 3.11×10^{-8}, MAE_1 was about 1.35×10^{-4}, and MAE_2 was around 5.14×10^{-4}, with the largest error occurring at an angle of attack of 7.2°. In Fig. 7(b), using transonic data for training, *MSE* was about 3.98×10^{-8}, MAE_1 was about 1.54×10^{-4}, and MAE_2 was around 5.75×10^{-4}, and the largest error was at 11.0°. In Fig. 7(c), using supersonic data, *MSE* was approximately 2.38×10^{-9}, MAE_1 was approximately 3.96×10^{-5}, and MAE_2 was around 1.22×10^{-4}, and the largest error was at 7.2°. All *MSE* values were below 1×10^{-8}, all MAE_1 values were below 2×10^{-4}, and the supersonic results almost perfectly matched the actual wind tunnel test values. This result is more accurate than that of training without segmentation, and the detailed error values can be found in Table 5. It can be concluded that training and predicting aerodynamic coefficients based on Mach number segmentation significantly improve accuracy.

After completing the training of the lift coefficient using the above settings and obtaining high accuracy results, segmented training was subsequently performed for the drag coefficient with the above optimal setting. Similar to the lift coefficient, the drag coefficient is an essential parameter in the initial stages of aircraft shape design. Therefore, the drag coefficient was incorporated into the BPNN for training, using the same settings: the hyperbolic tangent (tanh) function, 40 neurons, and four hidden layers.

Figure 8 shows the drag coefficient performance at different angles of attack for subsonic, transonic, and supersonic. As in previous experiments, the red line represents the predicted



(a) (b)

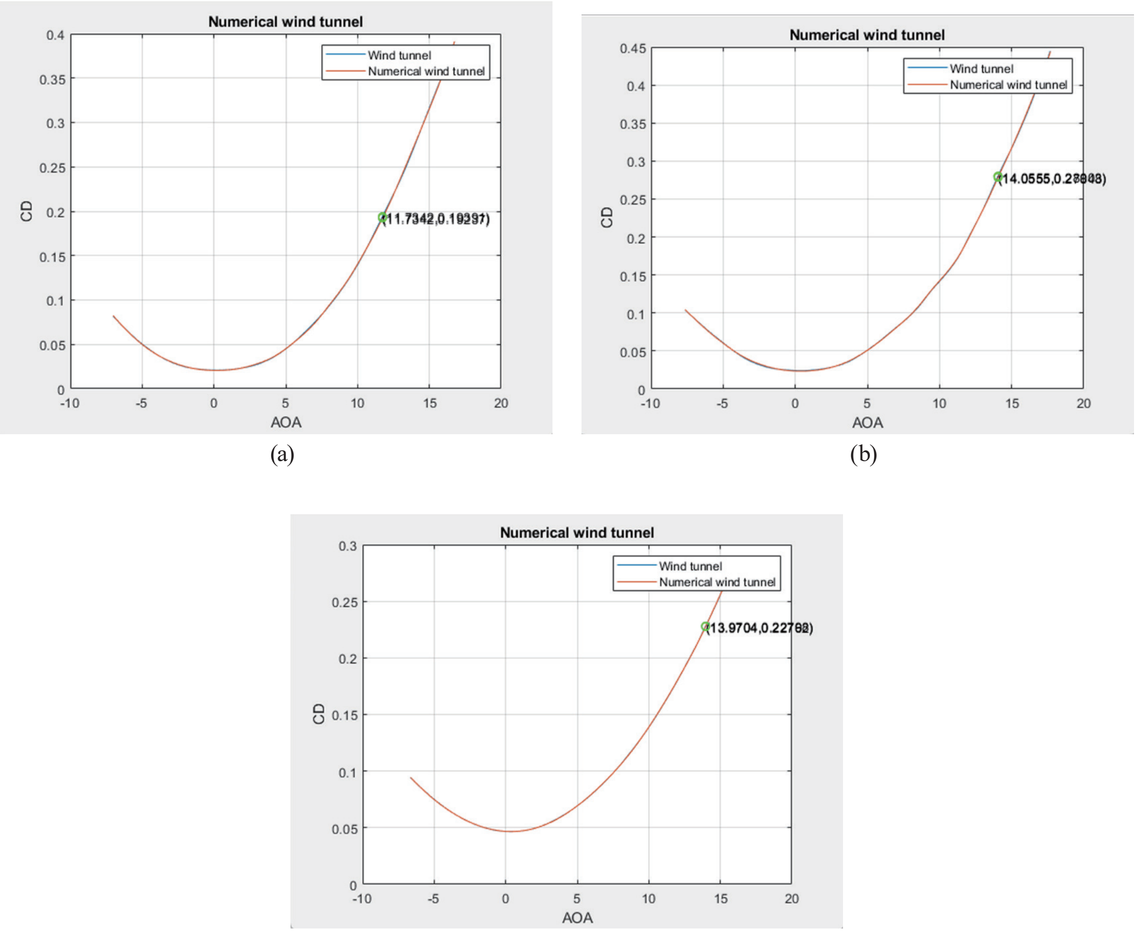
(c)

Fig. 7. (Color online) Lift coefficient performance in case study using different Mach numbers for training. (a) Subsonic, (b) transonic, and (c) supersonic segmentations.

Table 5
Error comparison of lift coefficient prediction at different Mach numbers.

Mach number	*MSE*	MAE_1	MAE_2
Subsonic	3.11×10^{-8}	1.35×10^{-4}	5.14×10^{-4}
Transonic	3.98×10^{-8}	1.54×10^{-4}	5.75×10^{-4}
Supersonic	2.38×10^{-9}	3.96×10^{-5}	1.22×10^{-4}

values from the numerical wind tunnel, whereas the blue line represents the actual wind tunnel test values. The *X*-axis represents the angle of attack, ranging from −8 to 18°, and the *Y*-axis represents the drag coefficient. In Fig. 8(a), using subsonic data to train the model, *MSE* was about 3.26×10^{-7}, MAE_1 was about 4.33×10^{-4}, and MAE_2 was around 1.54×10^{-3}, and the largest error was at 11.7°. In Fig. 8(b), using transonic data for training, *MSE* was approximately 3.61×10^{-7}, MAE_1 was about 4.84×10^{-4}, and MAE_2 was around 1.55×10^{-3}, and the largest error was at 14.1°. In Fig. 8(c), using supersonic data, *MSE* was approximately 1.54×10^{-8}, MAE_1 was about 9.65×10^{-5}, and MAE_2 was around 3.60×10^{-4}, and the largest error was at



(a) (b)

(c)

Fig. 8. (Color online) Drag coefficient performance in case study using different Mach numbers for training. (a) Subsonic, (b) transonic, and (c) supersonic segmentations.

14.0°. The predicted drag coefficient values aligned closely with the actual wind tunnel test values, with the lines nearly coinciding. *MSE* values ranged from 1 × 10^{-8} to 4 × 10^{-7}, MAE_1 values ranged from 9 × 10^{-5} to 5 × 10^{-4}, and the detailed error values can be found in Table 6. This indicates that using the above method for training the drag coefficient still maintains a high level of accuracy.

From the results of the four experiments, the best prediction results for the lift coefficient were *MSE* of approximately 2.38 × 10^{-9}, MAE_1 of approximately 3.96 × 10^{-5}, and MAE_2 of about 1.22 × 10^{-4}. For the drag coefficient, the optimal results were *MSE* of about 1.54 × 10^{-8}, MAE_1 of about 9.65 × 10^{-5}, and MAE_2 of approximately 3.60 × 10^{-4}. The errors for both lift and drag coefficients were randomly distributed and fell within an acceptable error range. In comparison with the study conducted by Huang *et al.*(9) on aircraft shape design using artificial neural networks, the best prediction result for the lift coefficient was MAE_1 of approximately 460.0 × 10^{-5} in their work, while for the drag coefficient, the best result was MAE_1 of approximately 280.0 × 10^{-5}. The results obtained in this study reveal a substantial improvement in prediction accuracy. The overall numerical trends closely matched the actual values from wind tunnel tests,

Table 6
Error comparison of drag coefficient prediction at different Mach numbers.

Mach number	*MSE*	MAE_1	MAE_2
Subsonic	3.26×10^{-7}	4.33×10^{-4}	1.54×10^{-3}
Transonic	3.61×10^{-7}	4.84×10^{-4}	1.55×10^{-3}
Supersonic	1.54×10^{-8}	9.65×10^{-5}	3.60×10^{-4}

indicating that the proposed method has a high prediction accuracy. Lift and drag coefficients are critical parameters for evaluating whether an aircraft meets performance and mission requirements and can be used for initial aircraft shape design analysis and validation. The errors in all cases can be effectively controlled.

5. Conclusions

In this research, BPNN was utilized to establish the numerical wind tunnel so that the process of aircraft shape analysis and validation can be accelerated. The proposed approach, dividing the Mach number into subsonic, transonic, and supersonic ranges, can be used to train and predict the lift and drag coefficients of aircraft currently being designed and developed. Our approach also reduces reliance on physical experimental methods, saving resources and speeding up the design process, demonstrating that Mach number is a critical feature in training the model. Moreover, by adopting the BPNN error compensation model, it is possible to accurately predict aerodynamic parameters, reduce costs, and enhance aircraft design and manufacturing, while also enabling the development of AI-based analysis tools.

The authors advance prior work by leveraging deep learning for the rapid acquisition of preliminary shape parameters to support design decisions, while integrating ADS parameters and an AI-based database to identify key factors affecting aerodynamic design, thereby enhancing the effectiveness of the proposed analysis framework. The current predictive targets of this research are lift and drag coefficients.

Future work will focus on extending the methodology to other critical aerodynamic coefficients, such as side force coefficients and pitching moments, to enhance the value and completeness of the numerical wind tunnel framework. Additionally, efforts will be directed toward improving the prediction accuracy of the BPNN model by incorporating further key aircraft parameters, including Reynolds number, wing area, and other relevant design features.

References

1 X. Guo, Z. Shen, Y. Zhang, and T. Wu: Smart Cities **2** (2019) 402. https://doi.org/10.3390/smartcities2030025

2 S. Sepasgozar, R. Karimi, L. Farahzadi, F. Moezzi, S. Shirowzhan, S. M. Ebrahimzadeh, F. Hui, and L. Aye: Appl. Sci. **10** (2020) 3074. https://doi.org/10.3390/app10093074

3 J. Li, J. Li, H. Cheng, H. Guo, and S. Qiu: Automot. Innov. **1** (2018) 2. https://doi.org/10.1007/s42154-018-0009-9

4 F. Cugurullo: Front. Sustainable Cities **2** (2020) 38. https://doi.org/10.3389/frsc.2020.00038

5 H. Moin, H. Z. I. Khan, S. Mobeen, and J. Riaz: Proc. 19th Int. Bhurban Conf. Applied Sciences and Technology (IEEE, 2022) 175. https://doi.org/10.48550/arXiv.2109.12149

6 J. Li, X. Du, and J. R. R. A. Martins: Prog. Aerosp. Sci. **134** (2022) 100849. https://doi.org/10.1016/j.paerosci.2022.100849

7 A. A. de Paula, F. D. M. Porto, and M. S. Sousa: Proc. AIAA Modeling and Simulation Technologies Conf. (AIAA, 2016). https://doi.org/10.2514/6.2016-4133

8 F. Nicolosi, D. Ciliberti, P. D. Vecchia, and S. Corcione: Proc. Applied Aerodynamics Conf. (AIAA, 2018). https://doi.org/10.2514/6.2018-2855

9 D. C. Huang, Y. F. Lin, L. J. Yang, and W. M. Chen: Sens. Mater. **32** (2020) 3169. https://doi.org/10.18494/SAM.2020.2845

10 T. D. Economon, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso: AIAA J. **54** (2016) 828. https://doi.org/10.2514/1.J053813

11 L. Hu, J. Zhang, Y. Xiang, and W. Wang: IEEE Access **8** (2020) 90805. https://doi.org/10.1109/ACCESS.2020.2993562

12 C. H. Huang: Identification of Flight Vehicle Models Using Eigensystem Realization Algorithm, Master's Thesis, Dept. of Electrical Engineering, National Chung Hsing University, Taichung, Taiwan (2011).

13 S. J. Russell and P. Norvig: Artificial Intelligence: A Modern Approach (Pearson, 2016).

14 M. I. Jordan and T. M. Mitchell: Science **349** (2015) 255. https://doi.org/10.1126/science.aaa8415

15 Y. LeCun, Y. Bengio, and G. Hinton: Nature **521** (2015) 436. https://doi.org/10.1038/nature14539

16 J. Schmidhuber: Neural Netw. **61** (2015) 85. https://doi.org/10.1016/j.neunet.2014.09.003

17 Y. L. Ye, S. T. Tseng, C. H. Huang, and K. L. Tsai: Proc. IEEE 4th Int. Conf. Electronic Communications, Internet of Things and Big Data (ICEIB) (IEEE, 2024) 602–607. https://doi.org/10.1109/ICEIB61477.2024.10602606

18 W. Peng, Y. Zhang, E. Laurendeau, and M. C. Desmarais: Sci. Rep. **12** (2022) 10776. https://doi.org/10.1038/s41598-022-10737-4

19 D. E. Rumelhart, G. E. Hinton, and R. J. Williams: Nature **323** (1986) 533. https://doi.org/10.1038/323533a0

20 R. Malmathanraj and T. F. Tyson: Int. J. Comput. Appl. **35** (2011) 5.

21 M. Rennie and A. Cain: Proc. 50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition (AIAA, 2012). https://doi.org/10.2514/6.2012-321

22 N. R. Secco and B. S. de Mattos: Aircr. Eng. Aerosp. Technol. **89** (2017) 211. https://doi.org/10.1108/AEAT-05-2014-0069

23 K. Balla, R. Sevilla, O. Hassan, and K. Morgan: Appl. Math. Model. **96** (2021) 456. https://doi.org/10.1016/j.apm.2021.03.019

About the Authors

Ching-Huei Huang received his M.S. and Ph.D. degrees in electrical engineering from National Chung Hsing University, Taichung, Taiwan, in 2006 and 2011, respectively. From 2003, he worked in the Aeronautical System Research Division, National Chung Shan Institute of Science & Technology, as an assistant researcher and as an associate researcher in 2017. Since 2023, he has been serving as an assistant professor at the Department of Automatic Control Engineering, Feng Chia University. His research interests are in evolutionary algorithm, system identification, intelligent vehicle control, and the flight control of UAV. (chueihuang@o365.fcu.edu.tw)

Kun-Lin Tsai received his Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 2006. He was a postdoctoral fellow at National Taiwan University of Science and Technology in 2007. He is currently a professor of the Department of Electrical Engineering of Tunghai University, Taichung, Taiwan. He is also the chair of the School of Continuing Education, Tunghai University. He served as the chair of the Department of Electrical Engineering from 2015 to 2018. He received the Outstanding Teaching Award from Tunghai University in 2018, 2021, and 2024. He is a member of IEEE, IICM, CCISA, and CIE. His research interests are in low-power system design, digital integrated circuit design, information security, IoT, and design thinking. (kltsai@thu.edu.tw)

Shih-Ting Tseng received her B.S. degree in foreign languages and literature from Feng Chia University in 2014 and her M.S. degree in electrical engineering from Tung Hai University in 2025. Since 2019, she has made a career transition to an institute of science and technology, where she became involved in aerodynamics and wind tunnel research, leading her to pursue further studies. Her research interests include neural networks in AI and numerical wind tunnels. (g12360009@thu.edu.tw)