

# Design of a Deep Learning Based Nonlinear Aerodynamic Surrogate Model for UAVs

Hasan Karali\*

*Istanbul Technical University, Istanbul, 34469, Turkey*

Mustafa Umut Demirezen†

*ITU Aerospace Research Center, Istanbul, 34469, Turkey*

M. Adil Yukselen‡

*Istanbul Technical University, Istanbul, 34469, Turkey*

Gokhan Inalhan §

*Cranfield University, Bedford, MK43 0AL, United Kingdom*

In this paper, we present a deep learning based surrogate model to determine non-linear aerodynamic characteristics of UAVs. The main advantage of this model is that it can predict the aerodynamic properties of the configurations very quickly by using only geometric configuration parameters without the need for any special input data or pre-process phase. This provides a crucial and explicit design and synthesis tool for mini and small UAVs. To achieve this goal, a large data set, which includes thousands of wing-tail configurations geometry parameters and performance coefficients, was generated using the previously developed and computationally very efficient non-linear lifting line method. This data is used for training the artificial neural network model. The preliminary results show that the neural network model has generalization capability. The aerodynamic model predictions show almost 1-1 coincidence with the numerical data even for configurations with different 2D profiles that are not used in model training. Specifically, the results of test cases are found to capture both the linear and non-linear region of the lift curves, by predicting the maximum lift coefficient, the stall angle of attack, and the characteristics of post-stall region correctly. Similarly, total drag and pitching moment coefficients are predicted successfully. The developed methodology provides the basis for bidirectional design optimization and offers insight for an inverse tool that can calculate geometry parameters for a given design condition.

## I. Nomenclature

$A_n$	=	Fourier series coefficients
$a$	=	lift curve slope
$a_\infty$	=	section lift curve slope
$\alpha$	=	angle of attack
$AR$	=	aspect ratio
$\beta_{tail_i}$	=	tail section twist angle
$c$	=	chord length
$C_l$	=	section lift coefficient
$C_L$	=	lift coefficient
$C_D$	=	total drag coefficient
$C_{D_i}$	=	induced drag coefficient
$C_m$	=	pitching moment coefficient

---

\*Research Assistant, Department of Aeronautical Engineering, karalih@itu.edu.tr, AIAA Student Member

†Affiliated Professor, ITU Aerospace Research Center, umut@demirezen.tech

‡Professor, Department of Aeronautical Engineering, yukselen@itu.edu.tr

§Professor, Centre for Autonomous and Cyber-Physical Systems, inalhan@cranfield.ac.uk, AIAA Associate Fellow

$\varepsilon$	=	downwash angle
$N$	=	semi-span number of stations
$s$	=	semi-span
$S$	=	surface area
$V_\infty$	=	free stream velocity

## II. Introduction

IN this work, we design a methodology and a neural network model that can predict the non-linear aerodynamic characteristics of lifting-surface configurations by using deep learning techniques. To train the artificial neural network model, a large data set was produced using the previously developed non-linear lifting line method [1]. Thanks to the speed and reliability of the aerodynamics analysis tool[1], it was possible to produce and analyze tens of thousands of configurations in minutes even on a personal computer. The designed artificial neural network model essentially calculates the aerodynamic performance of various wing-tail configurations including 3D maximum lift coefficient and pre- and post-stall aerodynamic behavior. As such, it provides a crucial and explicit design and synthesis tool for mini and small UAVs and the developed tool is explicitly used at our in-house UAV design and development programs as illustrated in Figure 1.

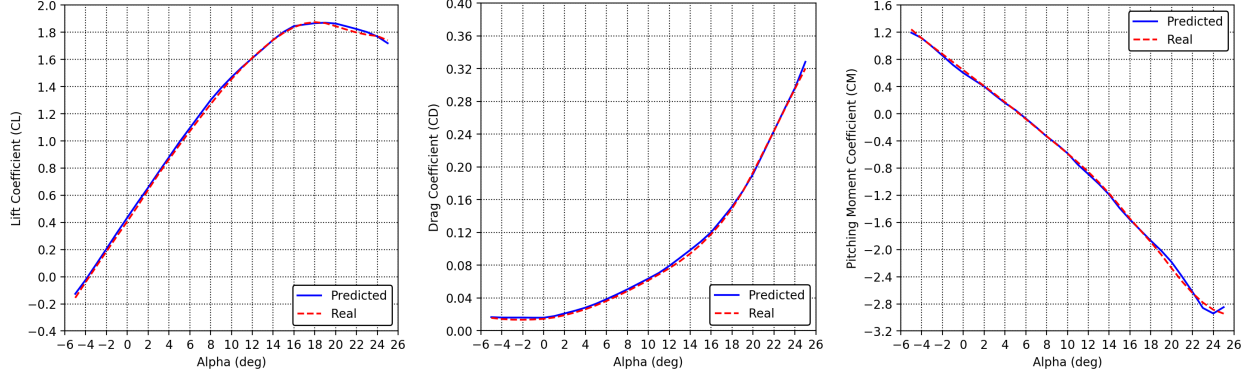


**Fig. 1 ITU ARC UAV geometry and flight tests**

In aerospace applications, there is always a need for a fast and reliable tool for aerodynamic characterization that can be used in both design optimization and mathematical modelling for high performance systems. Traditional engineering tools for conceptual design such as Data Compendium (Datcom) include semi-empirical methods which depends correlations and historical data do not give the flexibility or sufficiently general performance estimate capability to address arbitrary new designs [2]. It has been possible to meet these needs with deep learning techniques which are popular especially with the increase in computational powers. However, for such grand design optimization, it is necessary to train the model with huge amounts of data. In the literature, flight tests, experimental studies and some high-low order computational methods have been used to generate data sets for more specific problem studies [3]. A much larger set of data is required for a wider design optimization model. Creating large data sets with flight tests, experimental studies, and high order computational methods (CFD) is costly and not practically feasible[4]. At this stage, low-order computational aerodynamic methods stand out as a cheap and fast option. However, these methods generally do not take into consideration the viscous effects. Therefore, non-linear behavior of the lift curve at high angle of attack and the stall characteristics of the wing at post stall region cannot be determined. Especially in unmanned aerial vehicle (UAV) applications, it is crucial to determine non-linear performance of configurations which are operating at extensive flight regimes and agile maneuvering [5].

In this work, the artificial neural networks were trained with the data produced by non-linear lifting line tool (N-LLT) to calculate aerodynamic performance parameters including non-linear effects. This tool uses two-dimensional airfoil performance and geometry parameters as an input to calculate three-dimensional configuration performances. In the training of the neural network model, the NACA 4-digit series airfoils are defined by their geometric parameters: camber location, camber and thickness ratio [6]. This definition allows the model to use only dimension parameters as inputs without the need for two-dimensional airfoil performance data. The preliminary results show that model prediction show

almost 1-1 coincidence with the numerical data even configurations with different 2D profiles that are not used in model training. In Fig. 2, the results of an example application of this model to the wing-tail configuration is presented to show its capability. In the example  $AR = 15$  wing with NACA 3412 section and  $AR = 5$  tail with NACA 0012 section used for lifting surfaces. The tail was placed with the half-span away from the wing. Further information and test cases can be found in Section IV.



**Fig. 2 Neural network predictions for lift, drag and pitching moment coefficients**

The rest of the paper is organized as follows: In Section III, the developed methodology is presented in the subsections of aerodynamic model, data set and neural network model. In Section IV, the application of methodology is provided with several test case studies. In Section V, the conclusions are provided and the objectives that are considered to be completed in the future studies are explained.

### III. Methodology

In this work, an artificial neural networks model was developed using a large data set generated by Non-Linear Lifting Line Tool (N-LLT). The main advantage of this model is that it can predict the aerodynamic properties of the configurations very quickly with only geometry parameters without the need for any special input data or pre-process phases. The aerodynamic tool used in the training of the model comes to the forefront with its ability to calculate pre- and post-stall aerodynamic behavior at flow regimes marked with viscous effects at low Reynolds numbers. This is especially crucial on both the design and also flight control synthesis of mini and small UAVs. Thanks to the fast computational time of the method, a data set containing the aerodynamic performance of thousands of different geometry combinations was created in minutes.

In the subsections, firstly aerodynamic model will be described with its mathematical and physical expressions. In the second subsection, the data set properties created by this method will be summarized. In the third one, artificial neural network architecture will be explained.

#### A. Non-Linear Lifting Line Method

In this study, we utilize our previously developed non-linear lifting line method as the analysis tool [1]. In the method, potential flow based Prandtl's lifting line theory is modified for calculation of non-linear characteristics at high angles of attack. Main features of the method are a partial-linear approximation to the two-dimensional lift curve and an iteration process to correct error due to linear approximation. Thanks to this method; lift, total drag and pitching moment coefficient of a single lifting surface can be calculated for large angle of attack in a very fast and accurate fashion. In addition to these coefficients, the method also provides lift distribution, effective angle of attack and downwash for each station on the wing. This mathematical model was extended for the complete configuration analysis of an aerial vehicle which has multiple lifting surfaces.

In Prandtl's lifting line theory the flow around a finite wing is simulated by an infinite number of horseshoe vortices in a uniform parallel free flow. The leading parts of the horseshoe vortex are all coincide on the wing as the bound vortex and represent the effect of the wing while side parts represent effects of the trailing vortices. The detailed information about this model can be found in any aerodynamics textbook [7].

In the general applications of this model streamwise variation of strength of the bound vortex is represented by a sinus Fourier series,

$$\Gamma(\theta) = 4sV_\infty \sum_{j=1}^N A_j \sin(j\theta) \quad (1)$$

and the aerodynamic characteristics of the wing are obtained as follows:

$$C_L = \pi A R A_1 \quad (2a)$$

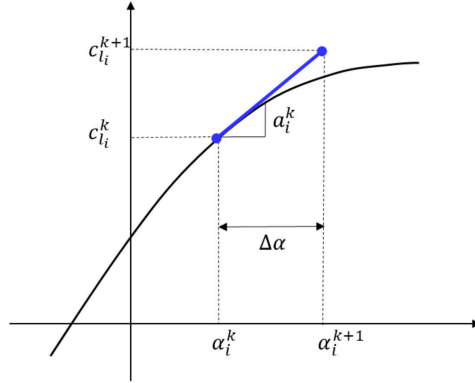
$$C_{D_i} = \frac{C_L^2}{\pi A R} (1 + \delta); \quad \delta = \sum_{j=2}^N j \left( \frac{A_j}{A_1} \right)^2 \quad (2b)$$

$$\varepsilon = \sum_{j=1}^N \frac{j A_j \sin(j\theta)}{\sin(\theta)} \quad (2c)$$

Fourier coefficients,  $A_j$  depends on the geometry of the wing and angle of attack. In non-linear lifting line method a partial linear approach is used to two-dimensional lift curve.

For a spanwise station,  $i$ , of a wing at any incidence, let  $\alpha_i^k$  and  $c_{l_i}^k$  be the local geometric angle of attack and the local lift coefficient, respectively, in the non-linear region of a lift curve (see Figure 3). Assume that the lift curve slope remains constant for a small increase of  $\Delta\alpha$  in angle of attack. If the new local geometric angle of attack is defined with  $\alpha_i^{k+1}$ , then the new lift coefficient is obtained as

$$c_{l_i}^{k+1} = c_{l_i}^k + a_i^k (\alpha_i^{k+1} - \alpha_i^k) \quad (3)$$



**Fig. 3 Partial-linear approach to the lift curve**

If this equation modifies with two-dimensional approach, effective angle of attack is used for local geometric angle. After implementation of Kutta-Joukowski lift theorem equation become as,

$$\frac{2\Gamma^{k+1}}{V_\infty c_i} = \frac{2\Gamma^k}{V_\infty c_i} + a_{\infty_i}^k [\Delta\alpha - (\varepsilon_i^{k+1} - \varepsilon_i^k)] \quad (4)$$

Introducing Eqs. (1) and (2c) representing the circulation and the downwash angle in terms of Fourier coefficients into Eq. (4) and rearranging, one obtains

$$\sum_{j=1}^N \left( \frac{1}{\mu_i^k} + \frac{j}{\sin(\theta_i)} \right) \sin(j\theta_i) A_j^{k+1} = \sum_{j=1}^N \left( \frac{1}{\mu_i^k} + \frac{j}{\sin(\theta_i)} \right) \sin(j\theta_i) A_j^k + \Delta\alpha \quad (5)$$

where,

$$\mu_i^k = \frac{8s}{a_{\infty_i}^k c_i} \quad (6)$$

This equation gives the relation between the wing geometry and the spanwise circulation distribution. Applying this equation for properly distributed  $N$  sections along wingspan following linear equation system is obtained:

$$\sum_{j=1}^N D_{ij}^k A_j^{k+1} = \sum_{j=1}^N D_{ij}^k A_j^k + \Delta\alpha; \quad i = 1, 2, 3, \dots, N \quad (7)$$

where,

$$D_{ij}^k = \left( \frac{1}{\mu_i^k} + \frac{j}{\sin(\theta_i)} \right) \sin(j\theta_i) \quad (8)$$

This system of equations can be written in matrix form as follows;

$$[D_{ij}^k] \{A_j^{k+1}\} = [D_{ij}^k] \{A_j^k\} + \{\Delta\alpha\} \quad (9)$$

where  $A_j^k$  are the Fourier series coefficients of the previous angle of attack and they are assumed to be known previously. The solution of this system of equation is

$$\{A_j^{k+1}\} = \{A_j^k\} + [D_{ij}^k]^{-1} \{\Delta\alpha\} \quad (10)$$

By using the obtained  $A_j^{k+1}$  Fourier coefficients, aerodynamic properties of the wing such as effective angle of attack, lift coefficient, induced drag coefficient etc. can be calculated for the new angle of attack. Note that, at each new angle of attack, the effective angles of attack and 2D lift curve slope,  $a_i^k$ , and therefore, the  $D_{ij}^k$  coefficients will change. However, before aerodynamic coefficient calculation these Fourier series coefficients are corrected via two-dimensional data. The detailed iteration process for circulation distribution will not be explained in this summary but it can be found in Ref. [1].

The non-linear lifting line method gives lift and induced drag coefficients directly. The viscous drag and the pitching moment coefficients can be obtained by using 2D experimental (or numerical) data of the wing section. For this purpose, first the 2D data is interpolated to obtain corresponding values for the effective angles of attack given by the method at each span wise section. Then these values are integrated numerically along the span. In the multi-surface application of the method, the downwash in the tail region is calculated with the developed mathematical model [8].

If the downwash angle at the tail location is known, the angle of attack for each tail station is obtained as;

$$\alpha_{tail_i} = \alpha_{root} + \beta_{tail_i} - \varepsilon_i \quad (11)$$

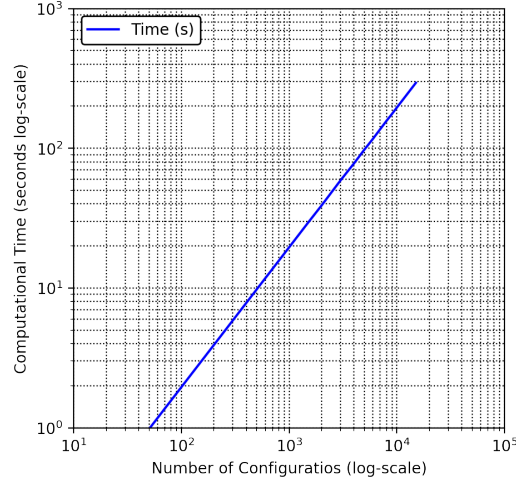
With this information, Non-Linear Lifting Line method can be performed on tail geometry. Thus, non-linear analysis of the wing-tail configurations is possible with this extended method.

$$C_{L_{total}} = C_{L_{wing}} + C_{L_{tail}} \frac{S_{tail}}{S_{ref}} \quad (12a)$$

$$C_{D_{total}} = C_{D_{wing}} + C_{D_{tail}} \frac{S_{tail}}{S_{ref}} \quad (12b)$$

## B. Data Generation

As previously explained, it is necessary to train the model with huge amounts of data for grand design optimization. This data will be generated via Non-linear Lifting Line tool. As a solution to the challenges in data production, the method offers fast and reliable results. The geometry definition part of the code was modified and multiple XFOIL analysis outputs were used as the two-dimensional performance input [9]. Each configuration was analyzed for approximately 30 different angles of attack. The relationship between calculation time and configuration numbers is shown in Fig. 4 with log-log scale. It is important to note that in this study personal computer is used as the computational environment.



**Fig. 4 Relationship between number of configurations and calculation time**

As can be seen from Fig. 4, there is a linear relationship between the two parameters. Table 1 shows the number of configurations and the total number of rows in the test sets with calculation times.

**Table 1 Data sets dimensions and generation times**

Set	Number of Configurations	Total Row Number	Parameters (Columns)	Computation Time (s)
Data Set 1	5400	108,000	23	60
Data Set 2	16,200	405,000	23	180
Final Data Set	22,000	668,250	23	245

In this study, the final data set with 22,000 configurations was used, but in the future studies this number can be increased with various combinations.

Table 2 summarizes all 23 parameters in the data set. 19 of them are related to geometry, while one parameter is the flow condition and 3 parameters are for performance outputs.

**Table 2 All configuration parameters in Data set**

Wing and Tail Parameters	Location	Flow Condition	Performance Coefficients
Span	x - Distance	$\alpha$	$C_L$
Root Chord Length	y - Distance		$C_D$
Tip Chord Length	z - Distance		$C_m$
Incidence Angle			
Airfoil Thickness Ratio			
Airfoil Camber Ratio			
Airfoil Max. Camber Location			
2D Performance Input File Name			

In the first stage of the study, certain geometry limits were determined to prevent the data size from increasing too much. However, as mentioned in the previous sections, it will not be difficult to increase these combinations since the program can solve 5400 configurations for 30 different flow conditions in 60 seconds. The parameters determined for the combinations are listed in Table 3.

**Table 3 Geometry parameters and ranges in the first data set**

Surface	Airfoil	Chord	AR	Incidence	x-Location	Reynolds
Wing	NACA 2412	0.2 to 0.6	6 to 18	-2 to 2	-	$0.4 \times 10^6$ to $1.2 \times 10^6$
	NACA 4412					
	NACA 6412					
Tail	NACA 0012	$(0.5 \text{ to } 1) * c_{wing}$	4 to 8	-2 to 2	$(0.4 \text{ to } 0.8) * b_{wing}$	$0.2 \times 10^6$ to $1.2 \times 10^6$

### C. Artificial Neural Networks Model

The motivation of this study is to create a black box function which can predict the aerodynamic properties of lifting surface configurations including viscous effects by using deep learning techniques. For this purpose, an artificial neural network architecture was generated and trained with the data set described in the previous section.

Before the training of the model, the data set was examined and a few bad data (0.02%) detected via anomalies in aerodynamic performance coefficients were cleared. In addition, the data set was shuffled and divided into three groups as test, train and validation. First, 90% of the data set was reserved for train and 10% for testing. In the next step, 20% of the train set was used for validation. The percentages and data numbers of these groups are given in Table 4.

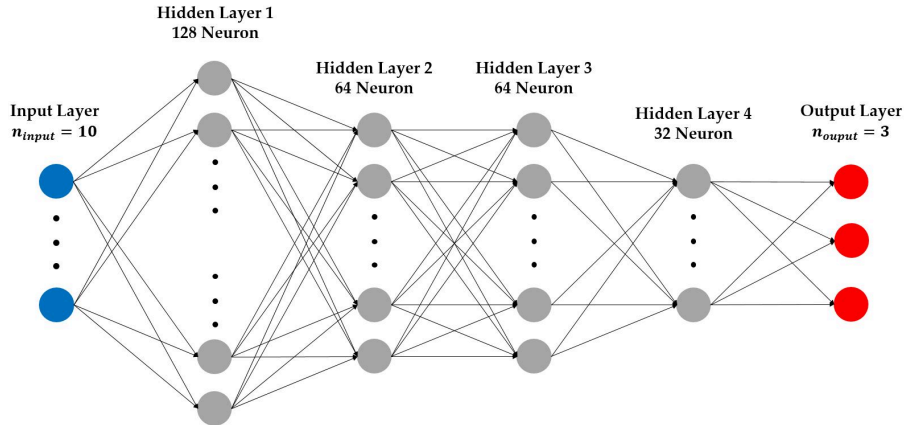
**Table 4 Train, Validation and Test sets**

Set	Data %	Number of Data
Train	72%	~ 481,000
Validation	18%	~ 120,260
Test	10%	~ 66,800

At the last stage before establishing the network structure the train and test sets are individually scaled to prevent convergence problem which is caused by the magnitude difference in the inputs. As shown in Eq. (13), scaling is applied by subtracting the minimum value of the set from each data and dividing the result by the range of the set. In the equation,  $x_i^{sc}$  represents the scaled sample.

$$x_i^{sc} = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (13)$$

The scaled data are given into the network from the input layer as features set. The model consists of input, output and 4 hidden layers. Figure 5 shows the structure and layers of the neural network.

**Fig. 5 Structure of the neural network**

The output of layers can be formulated as follows,

$$z_i^l = f \left( W^l x_i^{l,sc} + b^l \right) \quad (14)$$

where  $f$  is the activation function,  $W^l$  is the weight matrix and  $b^l$  is the bias vector at the  $l^{th}$  layer. Activation function decides which neurons will be activated, in other words what information would be passed to further layers. This function is selected as Rectified Linear Unit (ReLU) which is shown in Eq. (15) [10].

$$f(z) = \max(0, z) \quad (15)$$

"He normal initializer" is selected to initialize the weight matrices [11].

$$W_0^l = \mathcal{N} \left( 0, \sqrt{\frac{2}{n^{l-1}}} \right) \quad (16)$$

Mean absolute error is used as loss function to find the error between real and predicted value of the data. In Eq. (14) while  $y_i$  is the actual output,  $\hat{y}_i$  is the predicted output of the neural network.

$$E(W, b) = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (17)$$

Adam method is used as optimizer for minimizing the error function by updating weight and bias values [12]. Adam optimizer calculates partial derivatives of the cost function to update the moment vectors in the each iteration step.

$$\nabla(E) = \left[ \frac{\partial E}{\partial W}(W, b), \frac{\partial E}{\partial b}(W, b) \right] \quad (18)$$

Weight matrices and bias vectors are updated as follows

$$m_{t\partial W} = \beta_1 m_{t-1} + (1 - \beta_1) \frac{\partial E}{\partial W} \quad (19a)$$

$$m_{t\partial b} = \beta_1 m_{t-1} + (1 - \beta_1) \frac{\partial E}{\partial b} \quad (19b)$$

where,  $\beta_1$  is the exponential decay rate and  $t$  is the time step.  $m_{t\partial W}$  and  $m_{t\partial b}$  are the biased first moment vector to the partial derivatives of  $W$  and  $b$ . Each of these parameters update itself via gradient of the error function. In addition, second derivatives are used for updating the moment vector. In the following equations,  $v_{t\partial W}$  and  $v_{t\partial b}$  are the biased second moment vectors.

$$v_{t\partial W} = \beta_2 v_{t-1} + (1 - \beta_2) \frac{\partial^2 E}{\partial W^2} \quad (20a)$$

$$v_{t\partial b} = \beta_2 v_{t-1} + (1 - \beta_2) \frac{\partial^2 E}{\partial b^2} \quad (20b)$$

Bias-corrected first and second moments are calculated as follows

$$\hat{m}_{t\partial W} = \frac{m_{t\partial W}}{1 - \beta_1^t} \quad (21a)$$

$$\hat{m}_{t\partial b} = \frac{m_{t\partial b}}{1 - \beta_1^t} \quad (21b)$$

$$\hat{v}_{t\partial W} = \frac{v_{t\partial W}}{1 - \beta_2^t} \quad (21c)$$

$$\hat{v}_{t\partial b} = \frac{v_{t\partial b}}{1 - \beta_2^t} \quad (21d)$$



Obtained values are used to update weight and bias matrices as shown in Eq. (22). In these equations  $\alpha$  is the learning rate and  $\epsilon$  is the very small number that prevent division by zero number in the implementation.

$$W_t = W_{t-1} - \alpha \frac{\hat{m}_{t\partial W}}{\sqrt{\hat{v}_{t\partial W} + \epsilon}} \quad (22a)$$

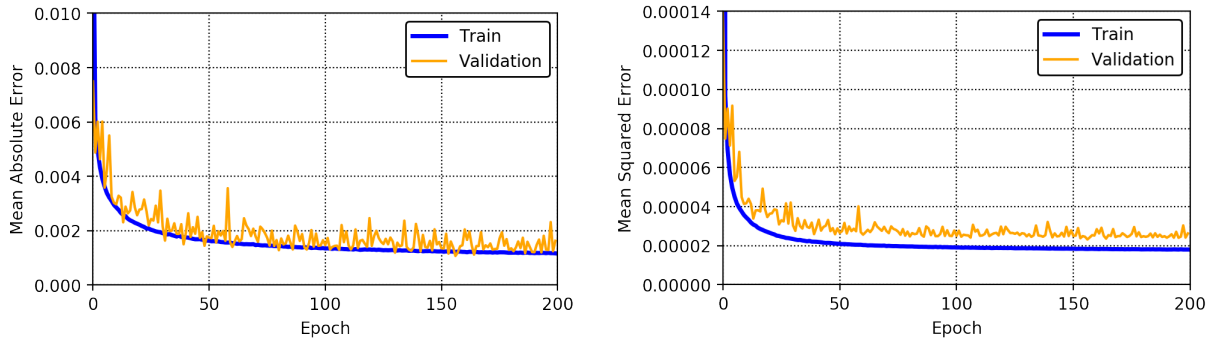
$$b_t = b_{t-1} - \alpha \frac{\hat{m}_{t\partial b}}{\sqrt{\hat{v}_{t\partial b} + \epsilon}} \quad (22b)$$

Finally, neural network model can be built with these parameters summarized in Table 5.

**Table 5 Hyperparameters for the Neural Network Architecture**

Parameters	Values
Number of Hidden Layers	4
<i>Neurons in the 1<sup>st</sup></i>	128
<i>Neurons in the 2<sup>nd</sup></i>	64
<i>Neurons in the 3<sup>rd</sup></i>	64
<i>Neurons in the 4<sup>th</sup></i>	32
Activation Function	ReLU
Optimizer	Adam
Initializer	HE Normal
Loss Function	MAE
Batch Size	128
Epoch Number	200

In the next step, the model was trained using the network structure and data sets. Learning curves of the training process are given in Fig. 6. As can be seen from figures, loss is converged to minimum value after 200 epoch. The behaviour of curves and error rate values show that learning processes are completed successfully without over/under fitting.



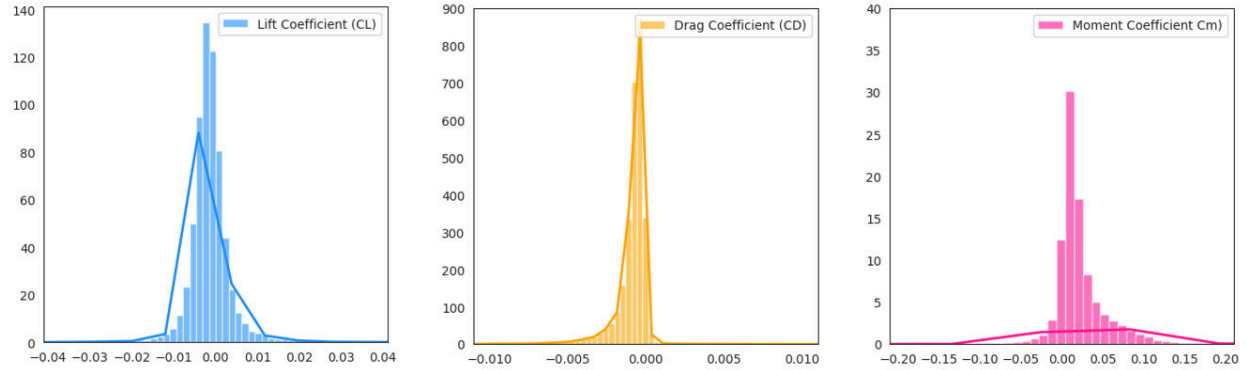
**Fig. 6 Learning curves of model: Mean Absolute and Mean Squared Error**

Furthermore, the error values are shown numerically in Table 6.

**Table 6 Training and Validation error values**

Data Set	Mean Absolute Error	Mean Squared Error
Training	0.00116	0.000018
Validation	0.00124	0.000024

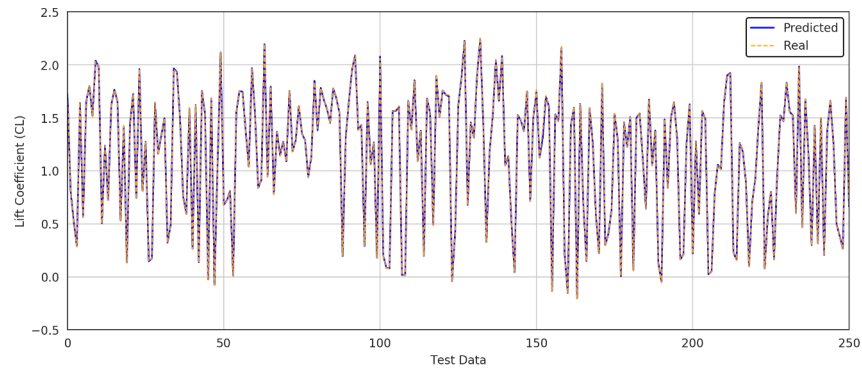
After finishing the training phase, the proposed model is verified with a test set which is not used in the training. In Fig. 7, the distribution of the error is examined by subtracting the predicted results from the real values.



**Fig. 7 Histogram of the difference between real and predict data**

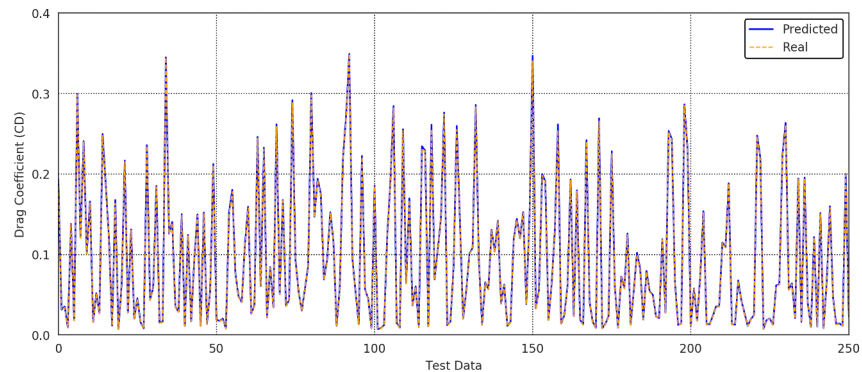
Predicted results are compared with non-linear lifting line tool output in Figures 8, 9 and 10. Since there are more than 66,000 test data, comparisons were made by taking 250 points randomly from the data set.

As can be seen from Fig. 8 the test sets have very different configurations and different flow conditions. However, the model was able to capture even the most extreme values. The predicted lift coefficients are almost 1-1 coincident with the true data.



**Fig. 8 Predicted lift coefficients**

The drag coefficient results obtained from the model and the real values are compared in Fig. 9. As in the previous graph, the predicted results are compatible with the real values.



**Fig. 9 Predicted drag coefficients**

Pitching moment results are compared in Fig. 10. It must be noted that pitching moment calculations are completed according to reference CG point at 50% of chord for all configurations. Except for a few points, the results appear to be equal to the real values.



**Fig. 10 Predicted pitching moment coefficients**

#### IV. Application of the Model

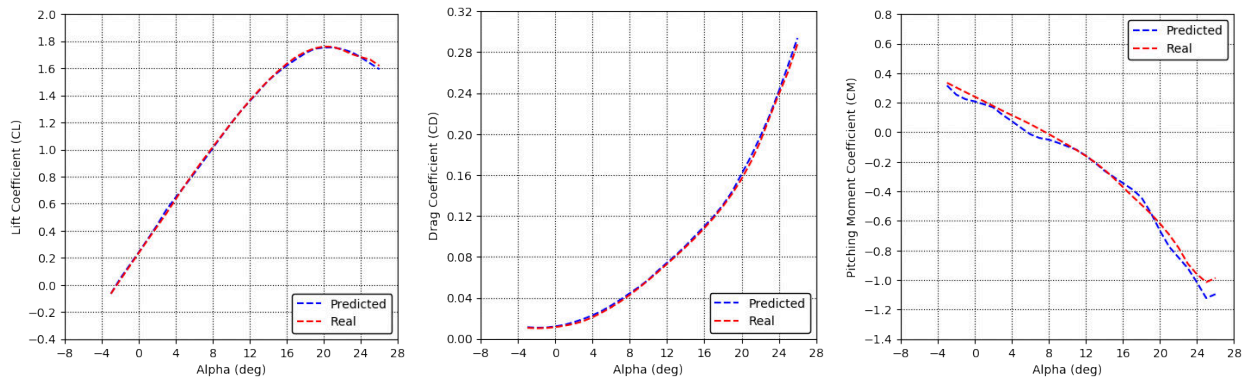
In this section, the results of several test applications of the current neural network model are presented to show its applicability, limits, advantages and disadvantages. In the first and second test case, wing-tail configurations with generic wing airfoils are used to test model prediction capability. These profiles are not used in the training phase. However, the model was able to calculate the impact on aerodynamic performance thanks to the ability to interpolate between profiles.

In the Test Case I,  $AR = 15$  wing with NACA 3412 section and  $AR = 5$  tail with NACA 0012 section used for lifting surfaces. The tail was placed with the half-span away from the wing. It should be noted that this profile was not used in model training.

**Table 7 Specifications for Test Case I**

	Span	Chord	Incidence	Airfoil	Reynolds	Location
Wing	9	0.6	0	NACA 3412	$1.2 \times 10^6$	-
Tail	2	0.4	-2	NACA 0012	$0.8 \times 10^6$	4.5

As can be seen from Fig. 11, the predicted results are nearly equal to real values. The neural network model accurately calculated the lift coefficient in both the linear and non-linear regions including post-stall. Similarly, it has calculated the drag curve successfully. Although there are some slight differences in the pitching moment curve, the overall behavior of the curve is correctly estimated.



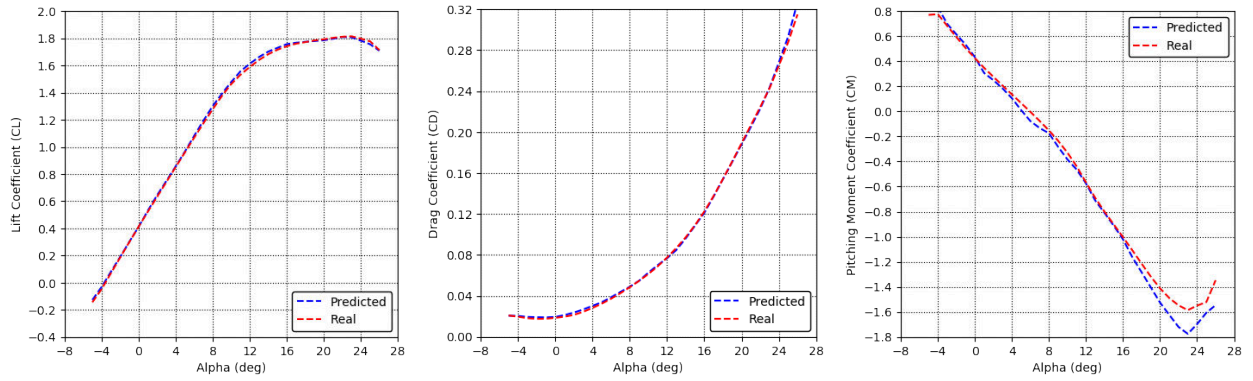
**Fig. 11 Lift, Drag, Pitching Moment coefficient of Test Case I**

In the Test Case II,  $AR = 15$  wing with NACA 5412 section and  $AR = 10$  tail with NACA 0012 section used for lifting surfaces. The tail was placed with the one-third span away from the wing. Again, it should be noted that this profile was not used in model training.

**Table 8 Specifications for Test Case II**

	Span	Chord	Incidence	Airfoil	Reynolds	Location
Wing	3	0.2	0	NACA 5412	$0.4 \times 10^6$	-
Tail	1	0.2	-2	NACA 0012	$0.4 \times 10^6$	1

The results of this analysis are presented in Fig. 12. As the results show, the new developed neural network model is compatible with the real values in the both linear and non-linear region.



**Fig. 12 Lift, Drag, Pitching Moment coefficient of Test Case II**

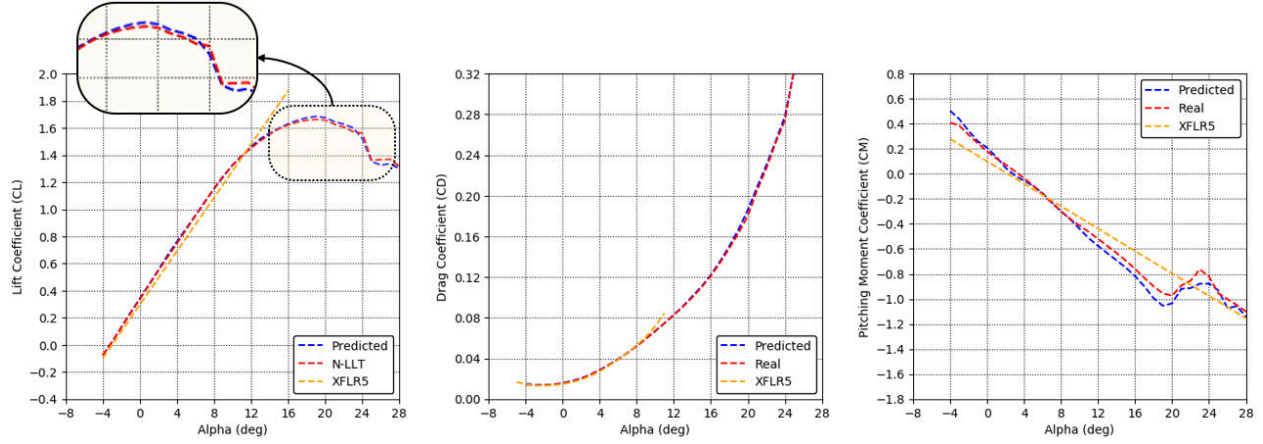
In Test Case 3, unlike the other tests, a geometry in tandem configuration was generated with the profiles used in the training of the model. Specifications for this test case are given in Table 9. The results of the model are compared with the vortex lattice solution of XFLR5 program [13].

**Table 9 Specifications for Test Case III**

	Span	Chord	Incidence	Airfoil	Reynolds	Location
Wing	2	0.2	0	NACA 4412	$0.4 \times 10^6$	-
Tail	1	0.1	0	NACA 0012	$0.2 \times 10^6$	1

It can be seen from Fig. 13 that the proposed model can be calculate non-linear effects on the lift curve even post-stall regime. However, computational aerodynamic methods such as vortex lattice methods generally do not take into consideration the viscous effects. Therefore, non-linear behavior of the lift curve at high angle of attack and the stall characteristics of the wing at post stall region cannot be determined.

In the Fig. 13, neural network model results are compared with XFLR5 VLM. In the total drag curve, both results are compatible with each other because of VLM results including add-on viscous drag module terms. In the pitching moment curve, slope is slightly different because these drag forces do not take account at moment calculation in XFLR5.



**Fig. 13 Lift, Drag, Pitching Moment coefficient of Test Case III**

## V. Conclusions and Future Works

The determination of the aerodynamic performance of the configurations is of prime importance for aircraft design and optimization studies. Developed neural network model provides the basis for bidirectional design optimization for UAVs. It can calculate aerodynamic performance of various wing-tail configurations including 3D maximum lift coefficient and pre- and post-stall aerodynamic behavior. One of the major advantages of the model as an analysis tool is that it can calculate the 3D performance of configuration when flow conditions or geometry changed without the need for two-dimensional airfoil performance. In addition to this, the model offers a start for an inverse tool that can calculate geometry parameters including section profiles, incidence angles, distance between surfaces for a given design condition. This provides a practical tool for both design and optimization studies of UAVs.

In this study, the final data set with 22,000 configurations was used, but this number can be increased with various combinations. As mentioned in the previous sections, it will not be difficult to increase these combinations since the program can solve 5400 configurations for 30 different flow conditions in 60 seconds. By increasing the combination of profile diversity and geometry, the model will become more general.

The preliminary test results are found generally quite successful both in the linear and the non-linear region of the lift curves, by giving the maximum lift coefficient, the stall angle of attack, and the characteristics of post-stall region correctly. Our current work focuses on designing a deep learning based inverse model that can calculate geometry parameters for a given performance requirements. In that sense, the proposed methodology offers a start for an inverse bidirectional design optimization tool that can calculate geometry parameters including section profiles, incidence angles, distance between surfaces for a given design condition.

## References

- [1] Karali, H., Yukselen, M. A., and Inalhan, G., "A New Non-Linear Lifting Line Method for 3D Analysis of Wing / Configuration Aerodynamic Characteristics with Application to UAVs," *AIAA SciTech 2019 Forum*, San Diego, CA, 2019, p. 2119. doi:10.2514/6.2019-2119.
- [2] Da Ronch, A., and Marques, P., *Advanced UAV Aerodynamics, Flight Stability and Control: Novel Concepts, Theory and Applications*, John Wiley & Sons, 2017.
- [3] Rajkumar, T., and Bardina, J., "Training data requirement for a neural network to predict aerodynamic coefficients," 2003. doi:10.1117/12.486343.
- [4] Cummings, R. M., Mason, W. H., Morton, S. A., and McDaniel, D. R., *Applied Computational Aerodynamics: A Modern Engineering Approach*, Cambridge Aerospace Series, Cambridge University Press, 2015. doi:10.1017/CBO9781107284166.
- [5] Ure, N., and Inalhan, G., "Design of a multi modal control framework for agile maneuvering UCAV," 2009, pp. 1 – 10. doi:10.1109/AERO.2009.4839598.
- [6] Abbott, I. H., Von Doenhoff, A. E., and Stivers Jr, L., "Summary of airfoil data," 1945.

- [7] Anderson Jr, J. D., *Fundamentals of Aerodynamics*, McGraw-Hill Educatio, Boston, 2010.
- [8] Phillips, W. F., Anderson, E. A., Jenkins, J. C., and Sunouchi, S., "Estimating the Low-Speed Downwash Angle on an Aft Tail," *Journal of Aircraft*, 2002, pp. 600–608.
- [9] Drela, M., "XFOIL: An analysis and design system for low Reynolds number airfoils," *Low Reynolds number aerodynamics*, Springer, 1989, pp. 1–12.
- [10] Nair, V., and Hinton, G. E., "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proceedings of the 27th International Conference on International Conference on Machine Learning*, Omnipress, USA, 2010, pp. 807–814.
- [11] He, K., Zhang, X., Ren, S., and Sun, J., "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification," *CoRR*, Vol. abs/1502.01852, 2015.
- [12] Kingma, D. P., and Ba, J., "Adam: A method for stochastic optimization," 2014.
- [13] Deperrois, A., "XFLR5 Analysis of foils and wings operating at low Reynolds numbers," *Guidelines for XFLR5*, 2009.

# Design of a deep learning based nonlinear aerodynamic surrogate model for UAVs

Karali, Hasan

2020-01-05

Attribution-NonCommercial 4.0 International

---

Karali H, Demirezen MU, Yukselen MA, Inalhan G. (2020) Design of a deep learning based nonlinear aerodynamic surrogate model for UAVs. In: 2020 AIAA SciTech Forum, 6-10 January 2020, Orlando, Florida, USA

<https://doi.org/10.2514/6.2020-1288>

*Downloaded from CERES Research Repository, Cranfield University*