EMBRY-RIDDLE Aeronautical University
SCHOLARLY COMMONS

International Journal of Aviation, Aeronautics, and Aerospace

Volume 7 | Issue 3 Article 3

2020

Identify aerodynamic derivatives of the airplane attitude channel using a spiking neural network

Nguyen Quang VInh
Academy of Military Science and Technology, Ha Noi, Viet Nam, vinhquang2808@mail.ru
Nguyen Duc Thanh
Academy of Military Science and Technology, Ha Noi, Viet Nam, ndthanh74@gmail.com
Hoang Minh Dac
Academy of Military Science and Technology, Ha Noi, Viet Nam, Hoangminhdac@gmail.com
Truong Dang Khoa
Le Quy Don Technical University, Ha Noi, Viet Nam, Khoahn67@gmail.com

Scholarly Commons Citation

VInh, N. Q., Duc Thanh, N., Minh Dac, H., & Dang Khoa, T. (2020). Identify aerodynamic derivatives of the airplane attitude channel using a spiking neural network. *International Journal of Aviation, Aeronautics, and Aerospace, 7*(3). Retrieved from https://commons.erau.edu/ijaaa/vol7/iss3/3

In recent decades, the spiking neural network (SNN) has been actively researched and developed (Abusnaina & Abdullah, 2017; Bohte, Kok, & La Poutre, 2002; Popular, 2010). Compared to the 2nd generation neural network, in which the output of the system is considering as the activation speed in a specific period time (rate firing), for SNN, the magnitude of the spikes contains no information, all information is encoding in the timing of the individual spikes (Popular, 2010; Popular & Kasińksi, 2011). Due to its significant computing performance and real-time response, SNN is used a lot in technical applications such as speech recognition, image processing, robot control, artificial intelligence (Abiyev, Kaynak, & On, 2012). For multi-layer SNN training, SpikeProp (Xu, Zeng, Han, & Yang, 2013) is a method of determining the errors based on the distance between the actual spike time and the desired spike time (target). Compared to the SpikeProp and Multi-ReSuMe algorithms, this algorithm has fundamental differences: The algorithm only focuses on the time interval target spikes and ignores other periods; During the training process, the calculation errors are propagated backward by intermittently changing the time of the previous class spikes, without using the traditional error backpropagation method; In the straight propagation calculation stage, use the analytical formula for the spike response model to determine the time of the spike instead of deciding the post-synaptic voltage.

Identify Aerodynamic Coefficient Derivatives of the Airplane's Attitude Channel Based on the *SNN* Airplane Kinematic Model

In the body coordinate system of the aircraft $Oxyz$ in Fig 1, use the following symbols (Sporea & Grüning, 2013): α, β - attack angle and slip angle; V - aircraft speed; X, Y, Z - aerodynamic force components; V_x, V_y, V_z - speed components; $\omega_x, \omega_y, \omega_z$ - angular speed components; M_x, M_y, M_z - aerodynamic moment components.

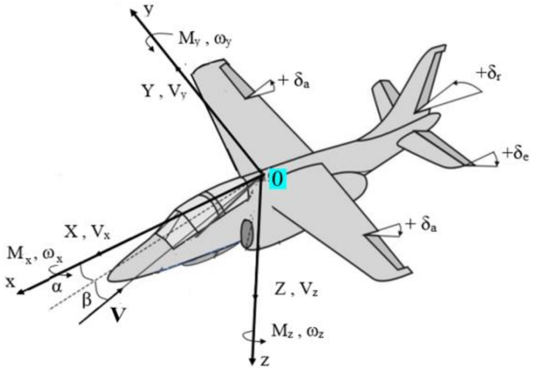


Figure 1. Body-axis of the Airplane and sign conventions.

In the altitude channel, the motion of the plane is describing by the system of nonlinear equations on formula (1) (Gerstner & Kistler, 2002).

$$\begin{cases} \dot{V} = \dfrac{P}{m}\cos\alpha - \dfrac{qS}{m}C_D - g\sin(\vartheta - \alpha) \\ \dot{\alpha} = \omega_y - \dfrac{P}{mV}\sin\alpha - \dfrac{qS}{mV}C_L + \dfrac{g}{V}\cos(\vartheta - \alpha) \\ \dot{\vartheta} = \omega_y \\ \dot{\omega}_y = \dfrac{qSb_A}{I_y}m_y \end{cases} \qquad (1)$$

where: ϑ - pitch angle; C_L, C_D - lift force coefficient, drag force coefficient; m_y - pitch moment coefficient; I_y - the inertia moment axis Oy .

These aerodynamic coefficients depend on many factors: $\alpha, \omega_y, \delta_e, V...$ aerodynamic diagram, Geometric parameters of the aircraft, and the wings (wing arrow angle, profile). With a defined aircraft type and for flights with subsonic speeds and without excellent maneuverability, the aerodynamic coefficient model is usually determining by a linear combination of aerodynamic coefficient derivatives for the control variables, which are stable. and intermediaries (Popular & Kasiński, 2011):

$$\begin{cases} C_D = C_{D_0} + C_D^{\alpha}\alpha + C_D^{\omega_y}\dfrac{b_A}{2V_0}\omega_y + C_D^{\delta}\delta_e & (a) \\ C_L = C_{L_0} + C_L^{\alpha}\alpha + C_L^{\omega_y}\dfrac{b_A}{2V_0}\omega_y + C_L^{\delta}\delta_e & (b) \\ m_y = m_{y0} + m_y^{\alpha}\alpha + m_y^{\omega_y}\dfrac{b_A}{2V_0}\omega_y + m_y^{\delta}\delta_e & (c) \end{cases} \qquad (3)$$

where: C_{D0}, C_{L0}, m_{y0} - drag coefficient, lift force coefficient and torque moment coefficient, when $\alpha = \delta_e = 0$; C_D^{α}, $C_D^{\omega_y}$, $C_D^{\delta_e}$, C_L^{α}, $C_L^{\omega_y}$, $C_L^{\delta_e}$, m_y^{α}, $m_y^{\delta_e}$, $m_y^{\omega_z}$ - the partial derivatives of drag coefficient, lift force coefficient and torque moment coefficient concerning $\alpha, \omega_y, \delta_e$

the identification of aerodynamic coefficient derivatives in equation (3), it is required to solve the system of nonlinear differential equations (Klein & Morelli, 2006). Solving this system of equations with analytic methods is very complicated. In this paper, we propose to use approximately this nonlinear dependency by SNN.

Structure of the Identification Model

The structure of the identification implementation model is showing in Figure 2. Because the SNN network implements the time change mechanism, before and after, the system must perform the time-signal encoding and decoding.

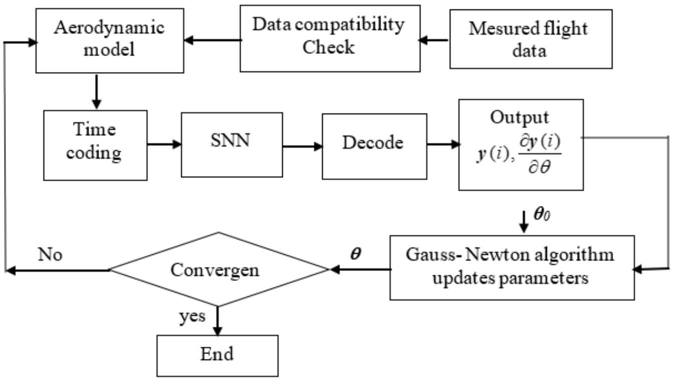


*Figure 1***.** Algorithm identification structure using SNN network.

The structure of the problem of identifying aerodynamic coefficients of the proposed aircraft is shown in Figure 2. The two main contents of the method include: Using SNN to approximate the nonlinear motion model of aircraft altitude channel (equation 2); Using the Gauss-Newton algorithm to identify the aerodynamic coefficients corresponding to (equation 3).

The SNN Network Approximates the Nonlinear Motion Model of Aircraft Altitude Channel

The proposed SNN network structure diagram is showing in Figure 3, which uses: the input layer with seven parameter sets, one hidden layer with 50 neurons, and the output layer consisting of 6 neurons, corresponding to 6 output parameter sets.

Network input is the vector with seven parameters:

$$\boldsymbol{u}(i) = [\alpha(i) \;\; \vartheta(i) \;\; \omega_y(i) \;\; V(i) \;\; C_D(i) \;\; C_L(i) \;\; m_y(i)]^T \quad (4)$$

The output vector of the network $\boldsymbol{z}(i+1)$ is a one-step prediction of the aircraft's movement parameters:

$$\boldsymbol{z}(i+1) = [\alpha(i+1) \;\; \vartheta(i+1) \;\; \omega_y(i+1) \;\; V(i+1) \;\; a_x(i+1) \;\; a_z(i+1)]^T \quad (5)$$

For network training as well as testing the ability to use the system to replace the nonlinear motion model in the aircraft altitude channel, the input-output data of the network in Equations (4), (5) must be determined. Measured values and preliminary treatment of these parameters were implemented (Thanh, Khoa, & Dac, 2020) In particular, the author has prepared two data sets from two flights with relatively similar flight conditions and typical parameters of the aircraft to serve for network training and identification of aerodynamic parameters.

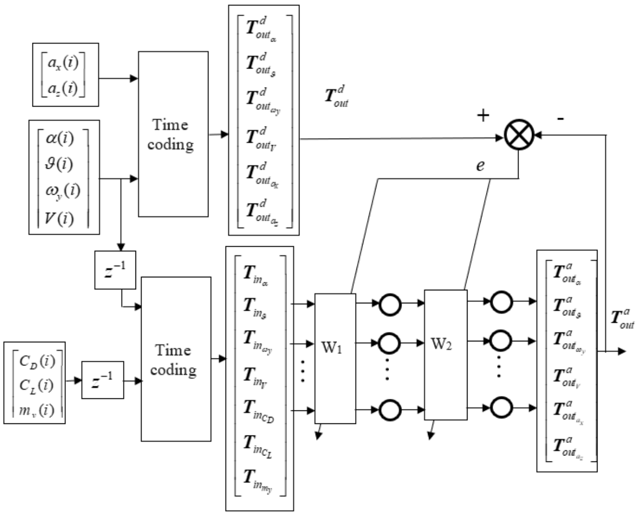


Figure 2. Proposed SNN network structure.

Flight Data of Aircraft Altitude Channel

Prepare data for the identification of aerodynamic coefficients in the aircraft's altitude channel, and the paper will perform the parameter identification of aircraft Cy–30.

Characteristic parameters. The thrust force of engine: $P = 74600$ *[N];* Mass: $m_0 = 24900$ (kg); wing reference area: $S = 65$ *[m^2];* mean aerodynamic chord: $b_A = 4,6$ *[m];* wingspan: $l = 14,1$*[m];* wingspan: $l_e = 9,8$ *[m];* Moment of inertia: $I_y = 62010$ *[kg.m];* dynamic pressure: $q = \rho V^2 / 2$ *[N/m$^{2]}$*.

Flight data of aircraft altitude channel. In the paper, to perform aerodynamic parameter identification using the data set received from the data recorded during the flight. The parameter set here is taking from the actual plane of the Cy - 30 aircraft (via the system of parameter writing itself). The parameters of aircraft altitude channel measured for identification include The angle of attack α*;* Pitch angle J ; pitch angle rate ω_y ; Translational velocity *V*; elevator landing-flap deflection δ_e ; The two flight data sets in the aircraft altitude channel received from two flights with similar flight conditions (average aircraft height $H = 4000[m]$, data recording cycle $T = 0,02[s]$ are showing in Figure 4.

Coding and Decoding

As mentioned earlier, the state of spiking neuron *j* described the voltage *u(t)* crosses a particular constant threshold value u_{ng} ; the neuron fires a spike, which is described by its spike time t^j_{out}. However, in most engineering problems, the computations must be performed on analog data, which leads to the development of methods for encoding analog signals into spike trains. The approach followed in this study associates weaker input signals with a ‘’late’’ firing time, whereas higher signals correspond to an a ‘’early’’ firing time. The values varying from x_{min} to x_{max} can be coded by choosing an interval *[* t_{min} *,* t_{max} *] [ms].*

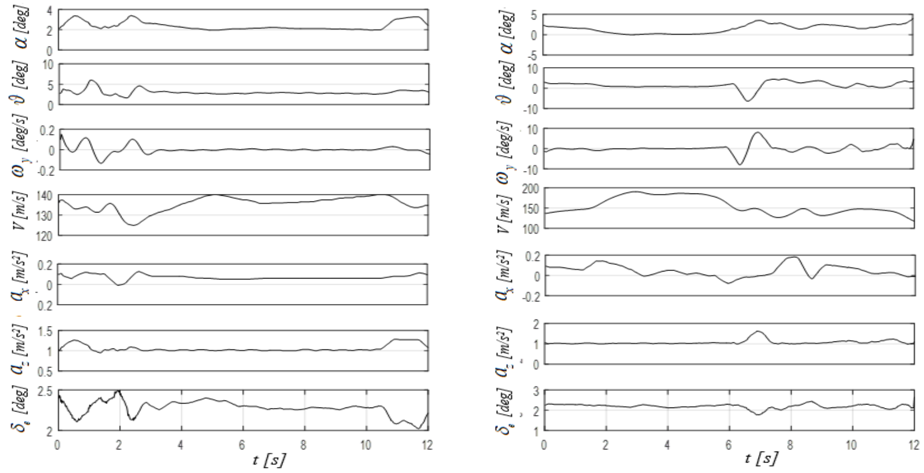
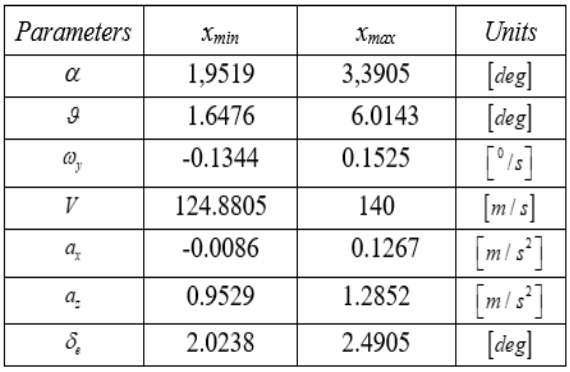


Figure 4. Two sets of measurement data for input-output of channel height.

Table 1

Input Value Vector Range [x_{min} *,* x_{max} *]*



Parameters	x_{min}	x_{max}	*Units*
α	1,9519	3,3905	$[deg]$
ϑ	1.6476	6.0143	$[deg]$
ω_y	-0.1344	0.1525	$[{}^0/s]$
V	124.8805	140	$[m/s]$
a_x	-0.0086	0.1267	$[m/s^2]$
a_z	0.9529	1.2852	$[m/s^2]$
δ_e	2.0238	2.4905	$[deg]$

The change range corresponds to the value range *[* x_{min} *,* x_{max} *]* selected in the range $0 \div 32$ [ms]. The activation mechanism at the rising edge of the output voltage, The spiking neuron is receiving a higher signal fire sooner than when the same neuron receives a weaker signal. The same idea is supported and called *delay coding*. The following formula is employed for encoding input variables into spike times (Sporea & Grüning, 2013):

$$t_i(x) = t_{max} - round\left(t_{min} + \frac{(x_i - x_{min}).(t_{max} - t_{min})}{(x_{max} - x_{min})} \right) \qquad (6)$$

where: $t_i(x)$ t_{min} t_{max} - the current, the minimum and maximum spike times, respectively; x_i , x_{min} , x_{max} - the present, minimum, and the maximum values of the input variables.; *round* - integer round operation.

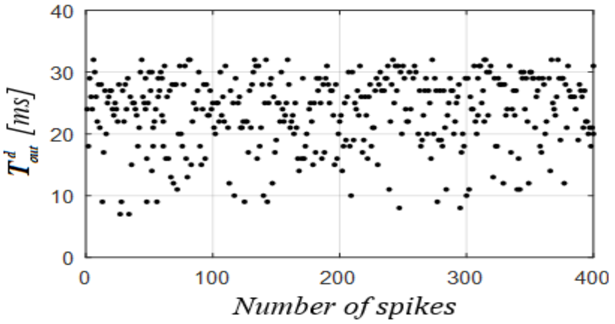


Figure 3. Time coding - parameter value ϑ.

Perform coding according to formula (6) for all network input and output parameter values. For example, with the parameter ϑ, The relationship between parameter value - time with the activation period from $0 \div 32$ [ms] is shown in Figure 5.

To train SNN, change the link weight value between neurons $\Delta w_{ij} = \gamma_i^n E_w^n / \varepsilon_j(s_i)$ to ensure that the spike time $A = \sum_{m=m_0}^{i} w_m \exp(t_S^m / \tau_2)$ becomes the desired spike time [13]. To convert backward from spike time series to a set of output signal values, use the following conversion formula:

$$x^i = x_{max} - t_a^i \left(\frac{(x_{max} - x_{min})}{(t_{max} - t_{min})} \right) \qquad (7)$$

With t_a^i - output spike times.

SNN Training and Testing Results

The SNN training consists of two steps: the weight modification to complete preparation of the current layer and the presynaptic spike jitter to backpropagate the error. The calculation is made using the MATLAB software tool.

The coding follows Eq (6). The network test and test data set consist of 600 data points encoded in time $0 \div 32$ [ms], of which 400 marks are for training, and 200 points are for network testing. The input vector $\boldsymbol{u}(i)$ is encoding into the input spikes sequence vector $\boldsymbol{T}_{in}^i$. The predicted outputs vector $\boldsymbol{z}(i+1)$ is codified into the desired sequence of spike output chains $\boldsymbol{T}_{out}^d = \left[\boldsymbol{T}_{out_\alpha}^d, \boldsymbol{T}_{out_\vartheta}^d, \boldsymbol{T}_{out_{\omega_y}}^d, \boldsymbol{T}_{out_V}^d, \boldsymbol{T}_{out_{a_x}}^d, \boldsymbol{T}_{out_{a_z}}^d \right]$.

Feedforward calculation.

The feedforward calculation is performed before conducting network training; SNN will calculate the spike output chains $\boldsymbol{T}_{out}^{a}=\left[\boldsymbol{T}_{out_{\alpha}}^{a},\boldsymbol{T}_{out_{\vartheta}}^{a},\boldsymbol{T}_{out_{\omega_y}}^{a},\boldsymbol{T}_{out_{V}}^{a},\boldsymbol{T}_{out_{a_x}}^{a},\boldsymbol{T}_{out_{a_z}}^{a}\right]$ from the input-output chains $\boldsymbol{T}_{in}^{i}=\left[\boldsymbol{T}_{in_{\alpha}},\boldsymbol{T}_{in_{\vartheta}},\boldsymbol{T}_{in_{\omega_y}},\boldsymbol{T}_{in_{V}},\boldsymbol{T}_{in_{C_D}},\boldsymbol{T}_{in_{C_L}},\boldsymbol{T}_{in_{m_y}}\right]$.

Feedback modification.

From the voltage $u\left(t_{out}^{j}\right)$and the error E for each target spike required in $\boldsymbol{T}_{out}^{d}$. We will calculate the time of the peak Δt_{S}^{j} with the *jth* input. Finally, calculate the weight adjustment $\Delta w_{ij}=\gamma_{i}^{n}E_{w}^{n}/\varepsilon_{j}(s_{i})$. Continue to calculate for all spikes in the set. Update all mutant moment variations and associated weights to continue counting for straight propagation in the next iteration.

Network training results will give time series of mutations for six network output parameters, corresponding to 6 desired spike ranges. For example, the parameter with the most significant variation, the pitch angle ϑ, the spike ranges $\boldsymbol{T}_{out_{\vartheta}}^{a}$ after four training epochs is as shown in Figure 6.

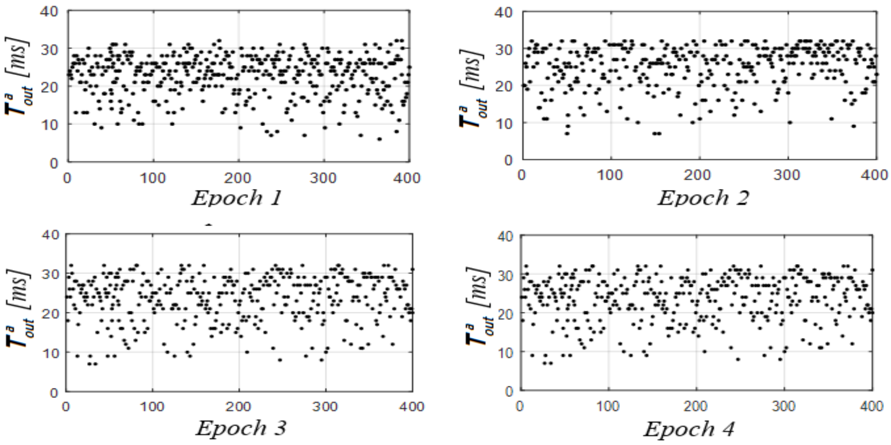


Figure 4. Network training result after four epochs with respect to pitch angle.

The difference in the time of the output spike compared to the time of the input spike for the pitch angle parameter through training epochs is shown in Figure 7 and Table 2. On the horizontal axis, the unit is [ms], on the vertical axis can show the desired number of spikes (black column) and the actual number of peaks (blue column).

Table 2

Results of Training and Network Testing

Epoch number	Standard error for training data set	Standard errors for test data set
1	0.1557	0.1713
2	0.1055	0.1262
3	0.0635	0.0781
4	0,0352	0,0417

After training the SNN with four epochs, the SNN is tested on the test data set with 200 data points. The difference in the time of the output spike with the desired spike time is shown in Figure 7 and Table 2.

To evaluate the quality of SNN training according to the difference in the time of network output spike with the required target spike time, use the comparison chart as shown in Figure 8 for all six network output parameters after four epoch network training.

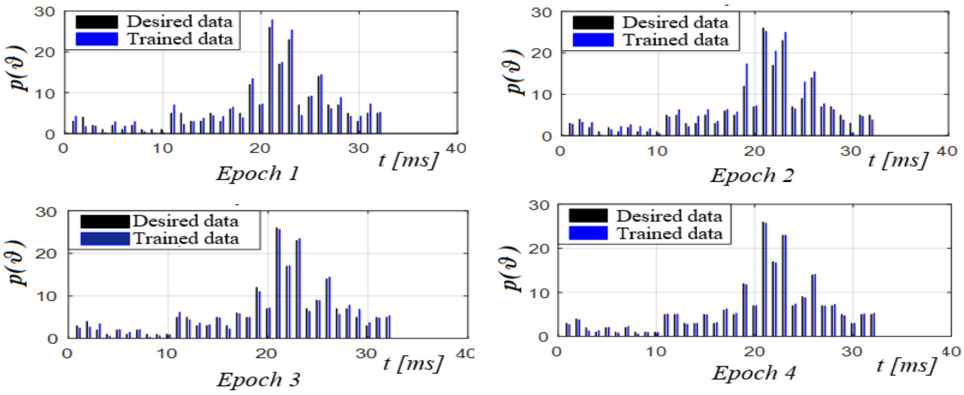


Figure 7. Network training results for pitch angle through 4 epochs.

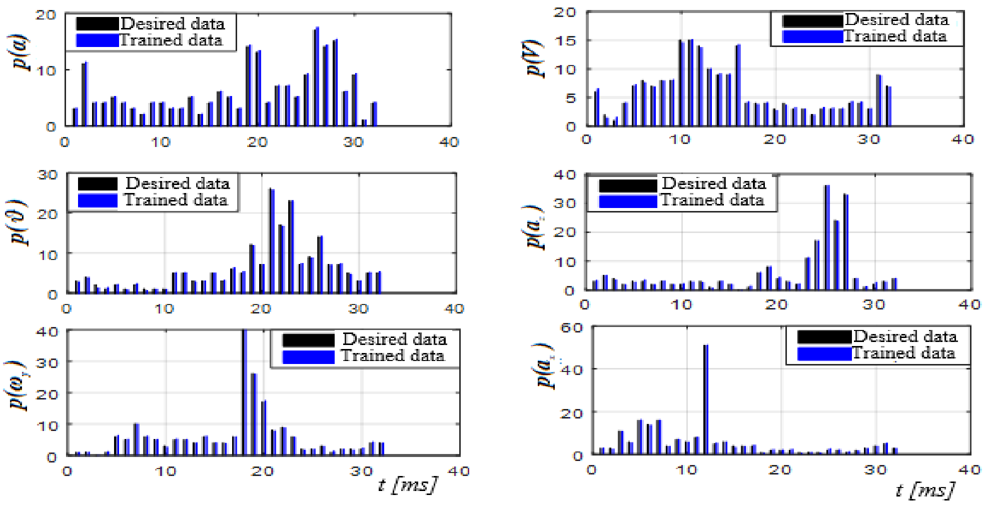


*Figure 8***.** Network training results for the output parameters.

Table 3

Results of Training and Network Testing After Four Epochs

Parameters	Standard error for training data set	Standard errors for test data set
$\sigma_\alpha \left[{}^0\right]$	0,0322	0,0551
$\sigma_\vartheta \left[{}^0\right]$	0.0352	0.679
$\sigma_{\omega_y} \left[{}^0/s\right]$	0.0585	0.0981
$\sigma_V \left[m/s\right]$	0,0253	0,0335
$\sigma_{a_x} \left[m/s^2\right]$	0,0218	0,0356
$\sigma_{a_z} \left[m/s^2\right]$	0,0284	0,0562

Table 3 shows the accuracy of the network output six parameters with network training parameters and network test parameters after four epochs.

From the above results, the following conclusions can be drawn:

- For SNN, the number of different spike times with the desired spike decreases very quickly after a few network training iteration (epoch). With the epoch number greater than 4, the error decreases almost negligible;

- The error between the network output value for the network training data set (400 data points) and the network test data set (200 data points) does not change much, proving that the network after training is generalized for the range of changes of input parameters;

- The number of neurons in the hidden layer of the network is quite small (50 neurons) (compared to the second-generation network, the best approximation is the RBN network is 164 neurons).

Identify Aerodynamic Coefficient Derivatives by the Gauss-Newton Method

The vector of parameters to be estimated was:

$$\boldsymbol{q} = (C_{D_0},\ C_D^{\alpha},\ C_D^{\omega_y},\ C_D^{\delta_e},\ C_{L_0},\ C_L^{\alpha},\ C_L^{\omega_y},\ C_L^{\delta_e},\ m_{y0},\ m_z^{\alpha},\ m_z^{\omega_z}, m_z^{\delta_e}) \qquad (8)$$

The Gauss-Newton algorithm is as follows:

Step 1: Give the original set of parameters $\boldsymbol{\theta}_0$ (these values are selected from the wind-tunnel tests or through documents with previous results).

In the paper, the initial set of parameters $\boldsymbol{\theta}_0$ was taken from Thanh et al. (2020).

$$\boldsymbol{\theta}_0 = \left[0.06,\ -1,\ -1,\ -1.23,\ 0,\ 5.1,\ 1,\ 0.1,\ 0.08,\ -1.26,\ -1,\ -0.76\right]^T$$

Step 2: The Gauss-Newton iteration algorithm updates the settings need to be identified after each loop according to the following formula:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \Delta\boldsymbol{\theta}_k\ ;\ \Delta\boldsymbol{\theta}_k = -\boldsymbol{g}_k\boldsymbol{M}_k^{-1} \qquad (9)$$

where: $\boldsymbol{g}_k$ - The gradient of the cost function; $\boldsymbol{M}_k$ - the Fisher information matrix; $\boldsymbol{v}_k(i)$ - error vector between data and SNN output; $\boldsymbol{R}_k$ - correlation error matrix. These parameters are determined as follows:

$$\boldsymbol{g}_k = \sum_{i=1}^{N}\frac{\partial\boldsymbol{y}_k(i)}{\partial\boldsymbol{\theta}_k}\boldsymbol{R}_k^{-1}\boldsymbol{v}_k(i)\ ;\ \boldsymbol{M}_k = \sum_{i=1}^{N}\frac{\partial\boldsymbol{y}_k^N(i)}{\partial\boldsymbol{\theta}_k}\boldsymbol{R}_k^{-1}\frac{\partial\boldsymbol{y}_k(i)}{\partial\boldsymbol{\theta}_k}\ ;\ \boldsymbol{v}_k(i) = \boldsymbol{z}(i) - \boldsymbol{y}_k(i)$$

$$\boldsymbol{R}_k = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{v}_k(i)\boldsymbol{v}_k^T(i) \qquad (10)$$

- $\boldsymbol{S}_k(i)$- The sensitivity matrix at the iteration k is calculated as follows:

$$\boldsymbol{S}_k(i) = \left[\partial\boldsymbol{y}(i)/\partial\boldsymbol{\theta}\right]_{jk} = (\boldsymbol{y}_{pk}(i) - \boldsymbol{y}_k(i))/\partial\boldsymbol{\theta}_j \qquad (11)$$

The sensitivity matrix $\boldsymbol{S}_k(i)$ at each iteration (k) is computed using the approximate relation given by Equation (11). The numerical values of $\boldsymbol{y}_{pk}$ (perturbed response) are obtained by replacing the parameter vector $\boldsymbol{\theta}$ with the $\boldsymbol{\theta} + \partial\boldsymbol{\theta}_j\boldsymbol{e}^j$ (where $\boldsymbol{e}^j$ - column vector with one in the *jth* row and zeros elsewhere) in the input variable vector of the already trained neural model.

Step 3: Determine the stop condition of the algorithm

+ Determining the price function:

$$J(\boldsymbol{\theta}_k, \boldsymbol{R}_k) = \frac{1}{2}\sum_{i=1}^{N}\boldsymbol{v}_k(i)\boldsymbol{R}_k^{-1}\boldsymbol{v}_k^T(i)^T \qquad (12)$$

+ Determine the stop condition of the algorithm

$$\left\|\frac{J(\boldsymbol{\theta}_{k+1},\ \boldsymbol{R}_{k+1}) - J(\boldsymbol{\theta}_k,\ \boldsymbol{R}_k)}{J(\boldsymbol{\theta}_k,\ \boldsymbol{R}_k)}\right\| \le \Delta J_{cp} \qquad (13)$$

The allowable value $\Delta\mathrm{J}_{\mathrm{cp}}$ is usually chosen 10^{-3} 0.

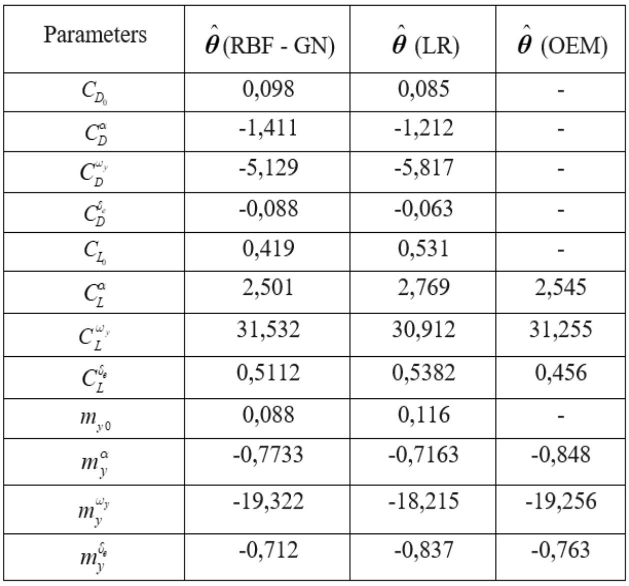
If the condition Eq (14) is satisfied, the end of the Gauss-Newton iteration algorithm, the value of aerodynamic coefficient derivatives θ at this iteration, is the identifiable parameters.

Simulation, evaluation of identification results of the aerodynamic coefficient derivative

In this section, two simulations were conducted to evaluate the effectiveness of the proposed SNN network method against the SpikeProp and RBN methods. With the SNN network structure, as shown in Figure 9 and the Gauss-Newton algorithm has realized the 12 aerodynamic coefficient derivatives of the aircraft attitude channel. After 43 iterations of the Gauss-Newton algorithm, the condition of convergence Eq (12) with the value is satisfied. The importance of the corresponding aerodynamic coefficient derivatives is given in column 2 of Table 4.

Table 4

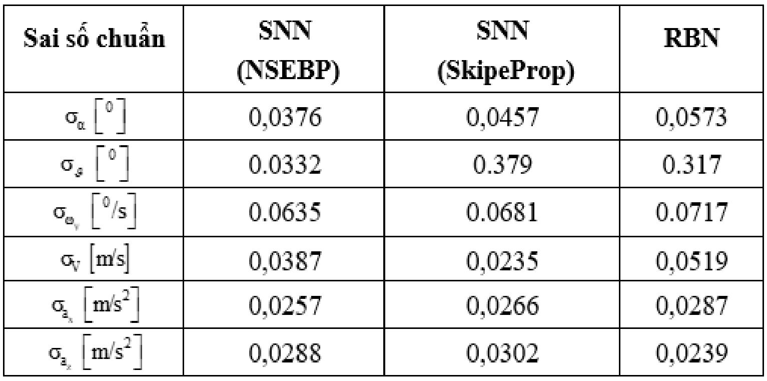
Identification Results Using SNN (NSEBP), SNN (SkipeProp) and RBN



Parameters	$\hat{\theta}$(RBF - GN)	$\hat{\theta}$ (LR)	$\hat{\theta}$ (OEM)
C_{D_0}	0,098	0,085	-
C_D^{α}	-1,411	-1,212	-
$C_D^{\omega_y}$	-5,129	-5,817	-
$C_D^{\delta_e}$	-0,088	-0,063	-
C_{L_0}	0,419	0,531	-
C_L^{α}	2,501	2,769	2,545
$C_L^{\omega_y}$	31,532	30,912	31,255
$C_L^{\delta_e}$	0,5112	0,5382	0,456
m_{y0}	0,088	0,116	-
m_y^{α}	-0,7733	-0,7163	-0,848
$m_y^{\omega_y}$	-19,322	-18,215	-19,256
$m_y^{\delta_e}$	-0,712	-0,837	-0,763

With the aerodynamic coefficient derivative values identified in Table 4, calculate the output values of SNN (NSEBP), SNN (SpikeProp), RBN and compare them with the actual data set received on a different flight from similar flight conditions and situations (Thanh et al., 2020). The fit between these two data sets is assessed by the standard deviation given in Table 5.

Table 5
Results of Training and Network Testing



Sai số chuẩn	**SNN (NSEBP)**	**SNN (SkipeProp)**	**RBN**
$\sigma_\alpha \left[{}^0\right]$	0,0376	0,0457	0,0573
$\sigma_\vartheta \left[{}^0\right]$	0.0332	0.379	0.317
$\sigma_{\omega_y} \left[{}^0/s\right]$	0.0635	0.0681	0.0717
$\sigma_V \left[m/s\right]$	0,0387	0,0235	0,0519
$\sigma_{a_x} \left[m/s^2\right]$	0,0257	0,0266	0,0287
$\sigma_{a_z} \left[m/s^2\right]$	0,0288	0,0302	0,0239

From the results received above, we can be compared with the identification results by the method SpikeProp and RBN (Klein & Morelli, 2006), with the following remarks:

- Identify the aerodynamic coefficient derivatives by the SNN (NSEBP) and SNN (SpikeProp) methods with more accurate results than the RBN method.

- The accuracy of identification by SNN (NSEBP) method is not much better than SNN (SpikeProp) method; however, the number of iteration than SNN (SpikeProp), resulting in shorter execution time.

Conclusions

The paper presented a plan to identify aerodynamic derivatives for the aircraft's altitude channel based on data received from actual flights, using SNN, network training by the NSEBP method, which approximates the nonlinear model of plane and Gaus -Newton algorithm. The results obtained reflect the effectiveness of the proposed method compared to the traditional SNN training method (SpikeProp) and using the second-generation ANN (RBN) when using the same aircraft’s attitude channel movement model in the nonlinear differential system and aerodynamic model.

References

Abiyev, R. H., Kaynak, O., & On, Y. (2012). *Spiking neural networks for identification and control of dynamic plants*. IEEE/ASME International Conference on Advanced Intelligent Mechatronics.

Abusnaina, A. A., & Abdullah, R. (2017). *Spiking neuron models: A review*. School of Computer Sciences, Universiti Sains Malaysia.

Bohte, S. M., Kok, J. N., & La Poutre, H. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing, 48,* 17–37.

Gerstner, W., & Kistler, W. M. (2002). *Spiking neuron models single neurons, populations, plasticity.* Cambridge University Press.

Klein, V., & Morelli, E. A. (2006). Aircraft system identification - theory and practice. *AIAA Education Series*, pp. 181-349.

Popular, F. (2010). Supervised learning in spiking neural networks with ReSuMe: Sequence learning, classification, and spike shifting. *Neural Computation, 22*, 467–510.

Popular, F., & Kasiński, A. (2011). *Introduction to spiking neural networks: information processing, learning and applications*. Institute of Control and Information Engineering, Poznan University of Technology, Poznan, Poland, (2011).

Sporea, I., & Grüning, A. (2013). Supervised learning in multilayer spiking neural networks. *Neural Computation, 25,* 473–509.

Thanh, N. D., Khoa, T. D., & Dac, H. M. (2020). Identify aerodynamic derivatives of airplane attitude channel based on the radial basis network, *Journal of Military Science and Technology*.

Xu, Y., Zeng, X., Han, L., & Yang, J. (2013). A supervised multi-spike learning algorithm based on gradient descent for spiking neural networks. *Neural Networks, 43,* 99–113.