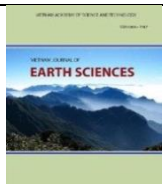
Vietnam Journal of Earth Sciences, 42(3), 276–287, Doi: 10.15625/0866-7187/42/3/15355

(VAST)

Vietnam Academy of Science and Technology

Vietnam Journal of Earth Sciences

http://www.vjs.ac.vn/index.php/jse



Identify some aerodynamic parameters of a airplane using the spiking neural network

Nguyen Duc Thanh[1], Le Tran Thang[1], Vuong Anh Trung[2], Nguyen Quang Vinh[3*]

[1]*Academy of Military Science and Technology, 17 Hoang Sam, Hanoi, Vietnam*
[2]*Air Defence - Air Force Academy, Kim son, Trung Son Tram, Son Tay, Hanoi, Vietnam*
[3]*Department of Control System, Bauman Moscow state Technical University, ul. Baumanskaya 2-ya, 5/1, Moscow*

Received 23 June 2020; Received in revised form 7 August 2020; Accepted 5 September 2020

ABSTRACT

The main objective of this study is to propose a method for identifying aerodynamic coefficient derivatives of aircraft attitude channel using spiking neural network (SNN) and Gauss-Newton algorithm based on data obtained from actual flights. Out of these, the SNN multi-layer network was trained by Normalized Spiking Error Back Propagation, in which, in the forward propagation period, the time of output spikes is calculating by solving quadratic equations instead of detection by traditional methods. The phase of propagation of errors backward uses the step-by-step calculation instead of the conventional gradient calculation method. SNN in combination with Gauss-Newton iterative calculation algorithm proposed in this study enables the identification of aerodynamic coefficient derivatives in a nonlinear model for aerodynamic parameters with higher accuracy and faster calculation time. The identification results are compared with the results when using the Radial Basis Function (RBF) network to prove the algorithm efficiency.

Keywords: aerodynamic identification; nonlinear model; flying vehicle.

1. Introduction

The problem of identifying aerodynamic derivatives based on data obtained from actual flights is quite complex, including planning and conducting flight tests, measuring, analyzing data compatibility, consider the motion model, and aerodynamic coefficient model. When using identification procedures using traditional methods (Klein V. and Morelli E.A., 2006) requirement for an accurate description of the motion state and aerodynamic models of flying vehicles. Document (Thanh N.D. et al., 2019) performed aerodynamic identification derivatives of the aircraft attitude channel by RBF-GN method. In particular, the RBF network serves as an approximation to the nonlinear model of the flight state of the aircraft.

In recent decades, the spiking neural network (SNN) has been actively investigated and developed (Wulfram Gerstner et al., 2002; Filip Popular and Andrzej Kasiński, 2011).

*Corresponding author, Email: vinhquang2808@yahoo.com

Compared to the 2nd generation of the neural network, in which the output of the system is considered as the activation speed in a specific period time (rate firing), for SNN, the magnitude of the spikes contains no information, all information is encoded in the timing of the individual spikes (Filip Popular and Andrzej Kasiński, 2011; Rahib H. Abiyev et al., 2012). 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 (Filip Popular and Andrzej Kasiński, 2011). For multi-layer SNN training, SpikeProp is a method of determining the errors based on the distance between the actual spike time and the desired spike time (target) (Bohte S.M., Kok J.N., La Poutre H., 2002). The Supervised Remote Method (ReSuMe) makes the practice of SNN neurons more efficient (Filip Popular and Andrzej Kasiński, 2011), but requires linear dependence between the input and output transmission speed of each class. The Normalized Spiking Error Back Propagation (NSEBP) algorithm is proposed in (Xiurui Xie et al., 2016). The algorithm is compared to the SpikeProp, and ReSuMe algorithms have 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. The paper performed network training according to the NSEBP algorithm and then combined SNN with the Gauss-Newton algorithm (SNN-GN) to evaluate aerodynamic derivatives of the aircraft attitude channel. The identification results compared with SNN(SpikeProp) - GN and RBF - GN methods showed more advantages.

2. Model and method

2.1. Airplane dynamic model

In the body coordinate system of the aircraft *Oxyz* in Fig. 1, use the following symbols (Thanh N.D. et al., 2019): α, β - the angle of attack, 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.

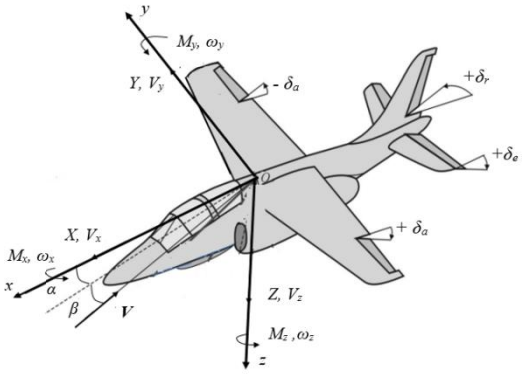


Figure 1. Body-axis of the Airplane and sign conventions

In the altitude channel, the motion of the plane is describing by the system of nonlinear equations on formula (1) (Klein V. et al., 2006).

$$\begin{cases} \dot{J} = \dfrac{P}{m}\cos a - \dfrac{qS}{m}C_D - g\sin(J-a) \\ \dot{a} = w_y - \dfrac{P}{mV}\sin a - \dfrac{qS}{mV}C_L + \dfrac{g}{V}\cos(J-a) \\ \dot{J} = w_y \\ \dot{w}_y = \dfrac{qSb_A}{I_y}m_y \end{cases} \quad (1)$$

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

The aircraft output model, in addition to the state parameters as shown in equation (1), also add linear acceleration:

$$\begin{cases} a_x = \dfrac{1}{m}(qSC_x + P) \\ a_z = \dfrac{qSC_z}{m} \end{cases} \quad (2)$$

with C_x, C_z - component aerodynamic coefficients in in body axes; a_x, a_z - component accelerations in body axes.

These aerodynamic coefficients depend on many factors: aerodynamic diagram, Geometric parameters of the aircraft, and the wings (sweep 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, stable and intermediaries variables (Klein V et al., 2006):

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

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

The identification of aerodynamic coefficient derivatives in equation (3) is required to solve the system of nonlinear differential equations (1). Solving this system of equations with analytic methods is very complicated. In this paper, we propose to use approximately this nonlinear dependency by SNN.

2.2. Structure of the identification model

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

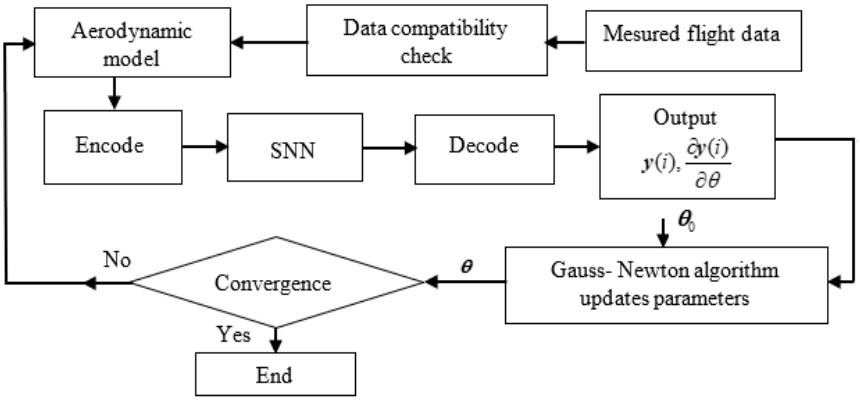


Figure 2. Algorithm identification structure using SNN network

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

2.3. The SNN approximates the nonlinear motion model of aircraft altitude channel

The proposed SNN network structure diagram is shown in Fig. 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.

The input vector of the network has 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 in (Thanh N.D. et al., 2019), In particular, three flight datasets were prepared from three flights with relatively similar flight conditions and typical parameters of the aircraft to serve for network training and identification of aerodynamic parameters.

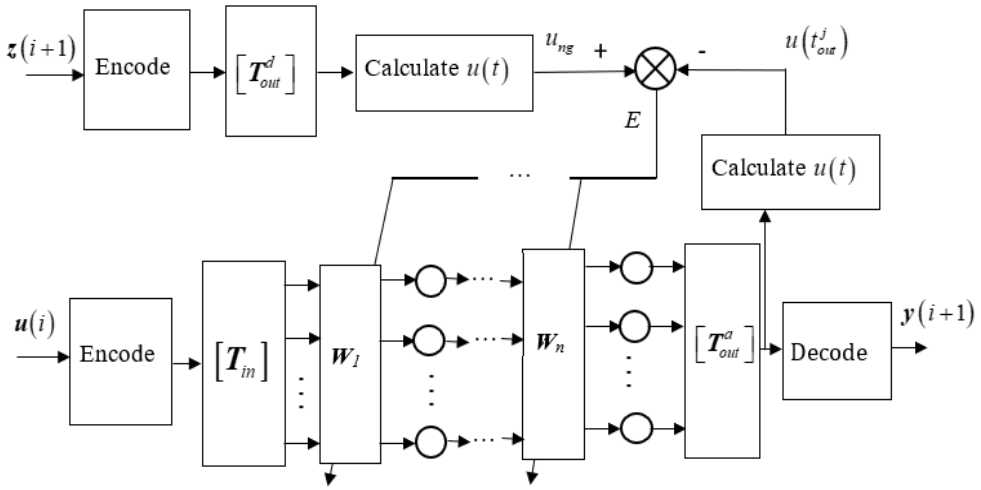


Figure 3. Proposed SNN network structure

2.4. Encode and decode

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_{out}^j. However, in most engineering problems, the computations must be performed on analogue data, which leads to the development of methods for encoding analogue 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].

Table 1. Range of input value vector [x_{min} , x_{max}]

Parameters	x_{min}	x_{max}
α [deg]	1,9519	3,3905
ϑ [deg]	1,6476	6.0143
ω_y [deg/s]	-0.1344	0.1525
$V[m/s]$	124,8805	140,0235
a_x $[m/s^2]$	-0,0086	0,1267
a_z $[m/s^2]$	0,9529	1,2852
δ_e [deg]	2,0238	2,4905

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 receives 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 (Rahib H. Abiyev et al., 2012) :

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

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

Perform coding according to equation (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 Fig. 4.

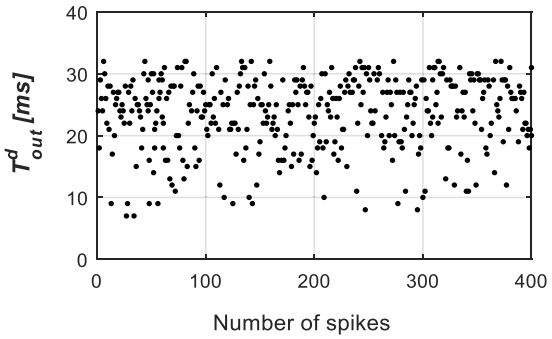


Figure 4. Time coding - parameter value ϑ

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 (Xiurui Xie et al., 2016). To convert backwards from spike time series to a set of output signal values, use the following conversion formula:

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

With t_a^i - output spike times.

2.5. Identification aerodynamic coefficient derivatives by the Gauss-Newton method

The vector of parameters in equation (3) has to be estimated as:

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

The Gauss-Newton algorithm is defined 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 (Vinh N.Q. et al., 2019).

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

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

$$\theta_{k+1} = \theta_k + \Delta\theta_k \; ; \; \Delta\theta_k = - g_k M_k^{-1} \quad (9)$$

$$g_k = \sum_{i=1}^{N} \frac{\partial y_k(i)}{\partial \theta_k} R_k^{-1} \nu_k(i) \; ; \; M_k = \sum_{i=1}^{N} \frac{\partial y_k^N(i)}{\partial \theta_k} R_k^{-1} \frac{\partial y_k(i)}{\partial \theta_k} \; ; \; \nu_k(i) = z(i) - y_k(i) \; R_k = \frac{1}{N} \sum_{i=1}^{N} \nu_k(i) \nu_k^T(i) \quad (10)$$

- $S_k(i)$- the sensitivity matrix at the iteration k is calculated as follows:

$$S_k(i) = \left[\partial y(i) / \partial \theta \right]_{jk} = (y_{pk}(i) - y_k(i)) / \partial \theta_j \quad (11)$$

The sensitivity matrix $S_k(i)$ at each iteration (k) is computed using the approximate relation given by equation (11). The numerical values of y_{pk} (perturbed response) are obtained by replacing the parameter vector θ with the $\theta + \partial\theta_j e^j$ (where 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 stopping condition of the algorithm

+ Calculate the cost function:

$$J(\theta_k, R_k) = \frac{1}{2} \sum_{i=1}^{N} \nu_k(i) R_k^{-1} \nu_k^T(i)^T \quad (12)$$

+ Stop condition of the algorithm

$$\left\| \frac{J(\theta_{k+1}, R_{k+1}) - J(\theta_k, R_k)}{J(\theta_k, R_k)} \right\| \le \Delta J_{cp} \quad (13)$$

The allowable value ΔJ_{cp} is usually chosen 10^{-3} (Xiurui Xie et al., 2016).

If the condition (13) is satisfied, the end of the Gauss-Newton iteration algorithm, the value of aerodynamic coefficient derivatives θ at this iteration is the identified parameters.

3. Materials

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

where: g_k - the gradient of the cost function; M_k - the Fisher information matrix; $\nu_k(i)$ - error vector between data and SNN output; R_k - correlation error matrix. These parameters are determined as follows:

3.1. 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: $b_A = 4,6[m]$; wing span: $l = 14,1[m]$; moment of inertia: $I_y = 62010[kg.m]$; dynamic pressure: $q = \rho V^2 / 2 [N/m^2]$.

3.2. Flight data of aircraft altitude channel

In the paper, to perform aerodynamic parameter identification used the dataset which recorded during the flight. The parameter set here have been taken 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 $\alpha[\text{deg}]$; pitch angle $J[\text{deg}]$; pitch angle rate $\omega_y[\text{deg/s}]$; translational velocity $V[m/s]$; elevator deflection $\delta_e[\text{deg}]$; acceleration component along *0z* in body axes $a_z[m/s^2]$; acceleration component along *0x* in body axes $a_x[m/s^2]$.

The three flight datasets in the aircraft altitude channel are used, including: a dataset for network training and test is shown in Fig. 5a); a dataset for identification of aerodynamic coefficient derivatives is shown in Fig. 5b); a dataset for cross - validation and comparing is shown in Fig. 5c), interval recording data $T = 0,02[s]$.

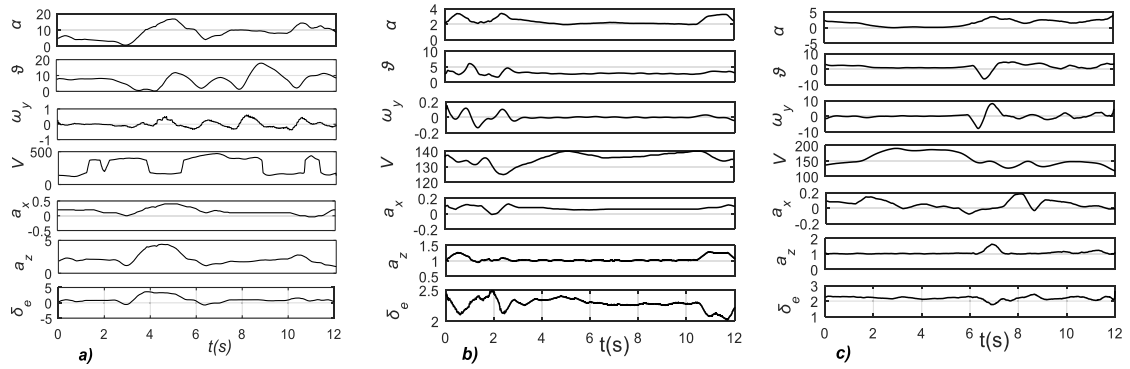


Figure 5. Three datasets of aircraft altitude channel

4. Results and discussions

4.1. 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 equation (6). The network training dataset and test dataset consist of 600 points encoded in time $0 \div 32$ [ms], of which 400 marks are for training, and 200 points are for network test. The inputs vector $\boldsymbol{u}(i)$ is encoded into the input spikes sequence vector $\boldsymbol{T}_{in}^{i}$. The predicted outputs vector $z(i+1)$ 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 and 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 sequence output. 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 Fig. 6.

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 Fig. 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).

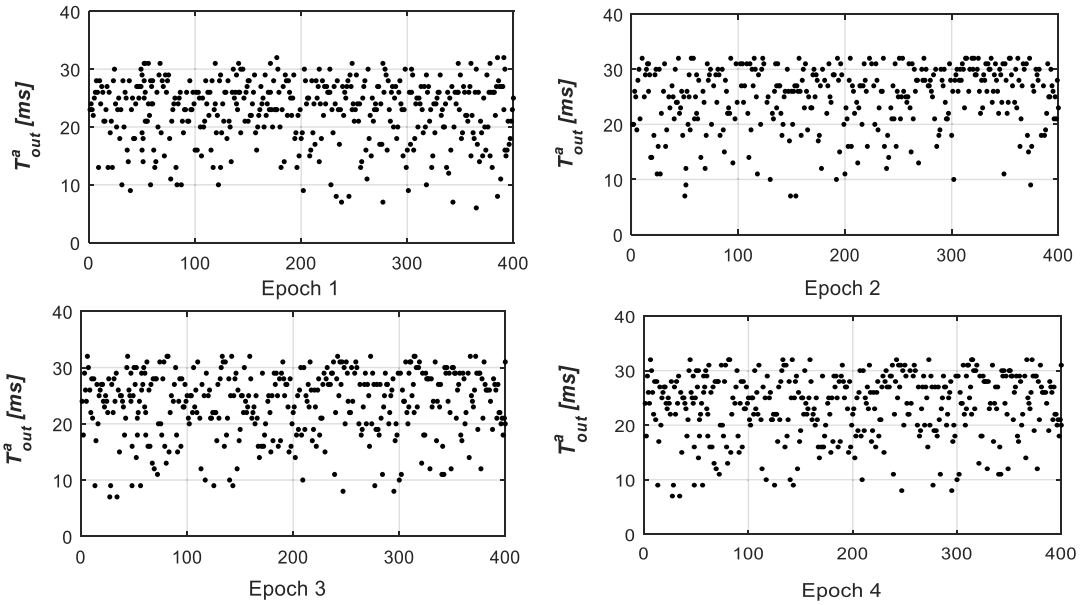


Figure 6. Network training result after four epochs with respect to pitch angle

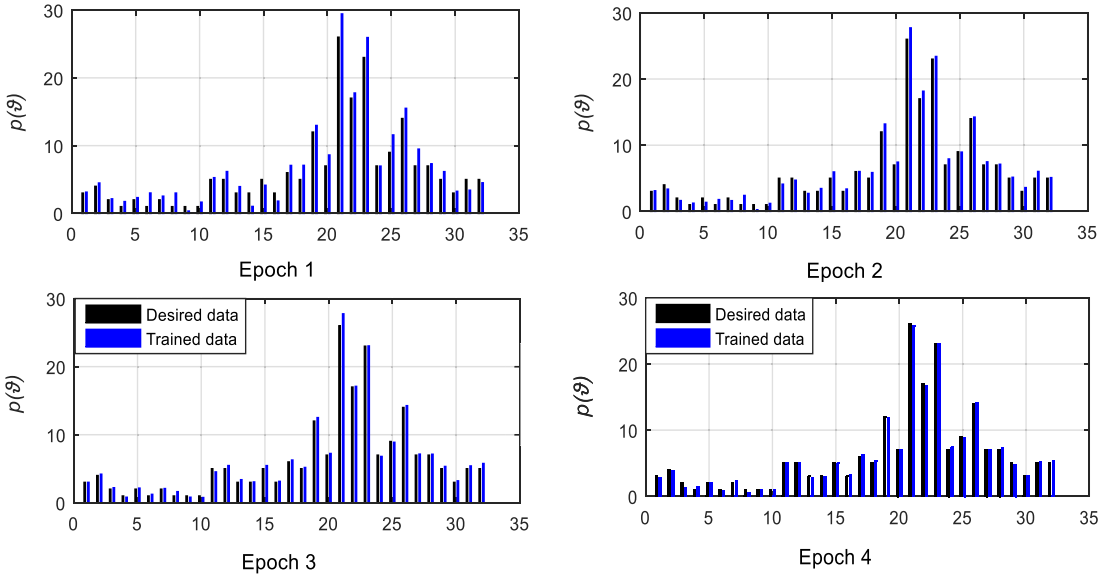


Figure 7. Network training results for pitch angle through 4 epochs

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

The SNN, which trained with four epochs, is tested on the test dataset. 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 Fig. 8 for all six network output parameters after four epoch network training.

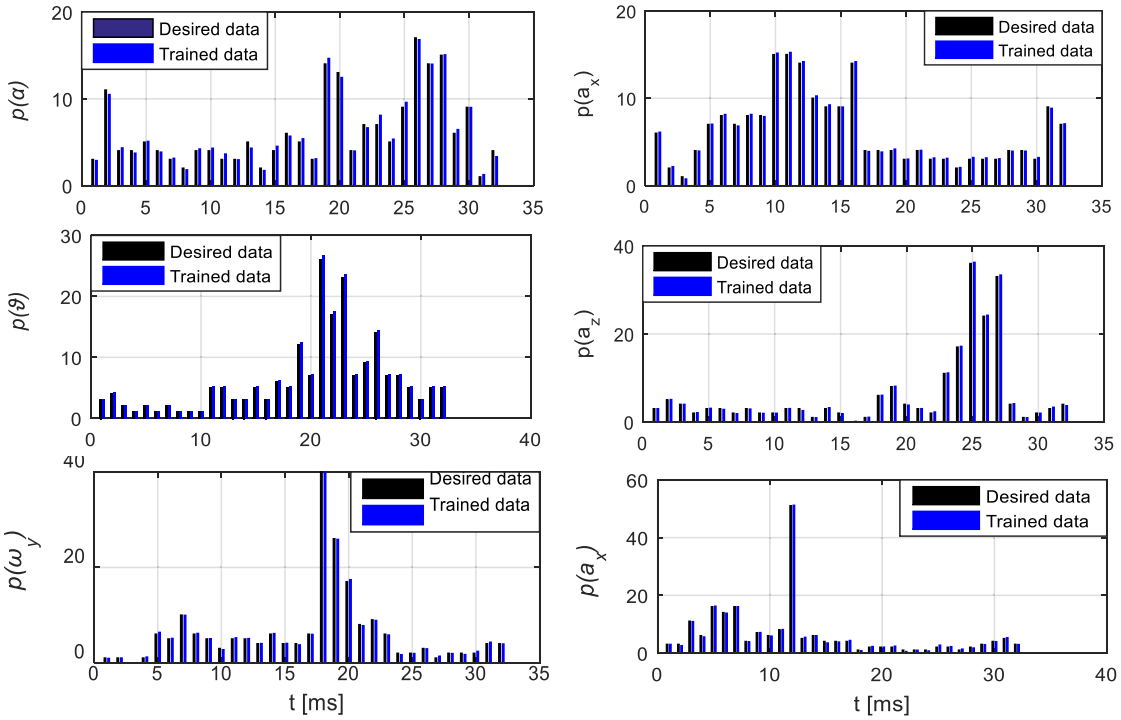


Figure 8. Network training results for the output parameters

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

Table 3. Results of training and network testing after four epochs

Parameters	Standard error for training data set	Standard errors for test data set
σ_α [deg]	0,0322	0,0551
σ_ϑ [deg]	0.0352	0.679
σ_{ω_y} [deg/ s]	0.0585	0.0981
$\sigma_{\dot{V}}$ [m/s]	0,0253	0,0335
σ_{a_x} [m/s^2]	0,0218	0,0356
σ_{a_z} [m/s^2]	0,0284	0,0562

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 epoch. With the epoch number greater than 4, the error decreases almost negligible;

- The error between the network output value for the network training dataset (400 points) and the network test dataset (200 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 RBF network is 164 neurons (Thanh N.D. et al. In this section, two simulations were conducted to evaluate the effectiveness of the proposed SNN network method against the SpikeProp and RBF methods. With the SNN network structure, as shown in Fig. 2 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 (13) is satisfied. The importance of the corresponding aerodynamic coefficient derivatives is given in column 2 of Table 4, 2019).

Table 4. Identification results using SNN (NSEBP), SNN (SkipeProp) and RBF

Parameters	$\hat{\theta}$(NSEBP)	$\hat{\theta}$(SpikeProp)	$\hat{\theta}$(RBF)
C_{D_0}	0,0815	0,0792	0,098
C_D^{α}	1,4983	1,4592	1,4115
$C_D^{\omega_z}$	5,2055	5,3127	5,129
$C_D^{\delta_e}$	0,0798	0,0827	0,0885
C_{L_0}	0,3911	0,3871	0,419
C_L^{α}	2,9331	2,7151	2,501
$C_L^{\omega_z}$	32,1132	32,1377	31,532
$C_L^{\delta_e}$	0,6011	0,5791	0,5112
m_{y0}	0,0725	0,0685	0,0881
m_y^{α}	-0,7133	-0,7871	-0,7733
$m_y^{\omega_z}$	-20,112	-19,763	-19,322
$m_y^{\delta_e}$	-0,871	-0,755	-0,712

4.2. Simulation, evaluation of identification results of the aerodynamic coefficient derivatives

With the aerodynamic coefficient derivatives values identified in Table 4, calculate the output values of SNN (NSEBP) and compare accuracy with the actual dataset (second dataset shown in Fig. 4b). This result is also compared with the two methods SNN (SpikeProp), RBF (Table 5).

Table 5. Compare standard deviations between three methods

Standard deviation	SNN (NSEBP)	SNN (SkipeProp)	RBF
σ_α [deg]	0,0376	0,0457	0,0573
σ_β [deg]	0.0332	0.0379	0.0317
σ_{ω_z} [deg/ s]	0.0635	0.0681	0.0717
σ_V [m/s]	0,0387	0,0235	0,0519
σ_{a_x} [m/s^2]	0,0257	0,0266	0,0287
σ_{a_y} [m/s^2]	0,0288	0,0302	0,0239

The generality of structured SNN (NSEBP) use for identification aerodynamic coefficient derivatives is considered when using the third dataset (shown in Fig. 4c) to perform cross-validation. The standard deviation between model output data and actual dataset are shown in the second column of Table 6. The third and fourth columns of Table 6 show the standard deviation when using the SNN (SkipeProp) and RBF networks, respectively.

Table 6. Standard deviation for cross validation

Standard deviation	SNN (NSEBP)	SNN (SkipeProp)	RBF
σ_α [deg]	0,0417	0,0595	0,0613
σ_β [deg]	0.0385	0.0432	0.0365
σ_{ω_z} [deg/ s]	0.0751	0.0788	0.0791
σ_V [m/s]	0,0468	0,0519	0,0588
σ_{a_x} [m/s^2]	0,0292	0,0319	0,0305
σ_{a_y} [m/s^2]	0,0314	0,0338	0,0257

From the results received above, we can be compared with the identification results by the method SpikeProp and RBF (Thanh N.D. et al., 2019), with the following remarks:

- Identification the aerodynamic coefficient derivatives use the SNN (NSEBP) and SNN (SpikeProp) give more accurate results than the RBF.

- The accuracy of identification when using SNN (NSEBP) is not much better than SNN (SpikeProp); however, the number of network training epochs is much smaller, resulting in a faster execution time.

5. Conclusions

In this paper, we have demonstrated a plan to identify aerodynamic derivatives for the aircraft's altitude channel based on data received from actual flights when using the SNN network which training by the NSEBP method, approximates the nonlinear motion model of the aircraft's altitude channel and Gaus-Newton algorithm. The simulation results received were reflected the effectiveness of the proposed method when compared to the SNN with the previous network training method (SpikeProp) and the second-generation ANN (RBF). In subsequent work, we will study to improve the

convergence speed of the Gauss-Newton algorithm, as well as the ability to generalize the SNN network in the cases when the aircraft's height channel motion model (1) affected by disturbances that cannot be measured.

Acknowledgements

This research has been supported by a grant for the basic research project (KCT-KT-04) from Academy of Military Science and Technology of Vietnam.

References

Albisser M., Berner C., Dobre S., Thomassin M., Garnier H., 2014. Aerodynamic coefficients identification procedure of a finned projectile using magnetometers and videos free flight data. In 28th ISB International Symposium on Ballistics, Atlanta, Georgia.

Audoly S., Bellu G., D'Angio L., Saccomani M., Cobelli C., 2001. Global identifiability of nonlinear models of biological systems. IEEE Transactions on Biomedical Engineering, 48(1), 55–65.

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

Cook M.V., 2012. Flight dynamics principles: a linear systems approach to aircraft stability and control. Butterworth-Heinemann.

Filip popular, Andrzej Kasiński, 2011. Introduction to spiking neural networks: information processing, learning and applications, Institute of Control and Information Engineering, Poznan University of Technology, Poznan, Poland.

Fujimori A., Ljung L., 2006. Model identification of linear parameter varying aircraft systems. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 220(4), 337–346.

Ha L.H., Vinh N.Q., Cuong N.T., 2020. Dynamics of Self-guided Rocket Control with the optimal Angle Coordinate System Combined with Measuring Target Parameters for Frequency Modulated Continuous Wave Radar. Intelligent Computing in Engineering. Advances in Intelligent Systems and Computing, 1125, 951–962.

Klein V., Morelli E,. 2006. Aircraft system identification: Theory and Practice. Chapter Experiment Design, A. I. A. A., 1801 Alexander Bell Drive, Reston, VA, 289–329.

Ljung L., Glad T., 1994. On global identifiability for arbitrary model parametrizations. Automatica, 30(2), 265–276.

McKennoch S., Liu D., Bushnell L.G., 2006. Fast Modifications of the SpikeProp Algorithm. IEEE International Joint Conference on Neural Networks, IEEE, 48, 3970–397.

Nguyen Duc Thanh, et al., 2019. Aerodynamic coefficient identification of of airplane's attitude channel by the output error method, Journal of Military Science and Technology, 6, 28–36.

Popular F. et al., 2010. Supervised learning in spiking neural networks with ReSuMe: sequence learning, classification, and spike shifting, Neural Computation, 22, 467–510.

Rahib H. Abiyev, Okyay Kaynak, Yesim On, 2012. Spiking Neural Networks for Identification and Control of Dynamic Plants, IEEE/ASME International Conference on Advanced Intelligent Mechatronics.

Saccomani M.P., Audoly S., D'Angio L., 2003. Parameter identifiability of nonlinear systems: the role of initial conditions. Automatica, 39, 619–632.

Semenov A.D., Volkov A.V., Schipakina N.I , 2019. Parametric Identification of Nonlinear Systems by Aggregation of Static and Dynamic Neural Networks. International Multi-Conference on Industrial Engineering and Modern Technologies, 76–82.

Sjoberg J., Zhang Q., Ljung L., Beneviste A., Delyon B., Glorennec P., Hjalmarsson H., Juditsky A., 1995. Nonlinear black-box modelling in system identification: a unified overview. Automatica, 31, 1691–1724.

Tsibizova T.Y., 2016. Identification methods for nonlinear control systems, journal of computer technology, automatic control, radio electronics, 6, 11–17.

Verdult V., Lovera M., Verhaegen M., 2004. Identification of linear parameter-varying state-space models with application to helicopter rotor dynamics. Taylor & Francis, International Journal of Control, 77(13), 1149–1159.

Vinh N.Q., Phuong Anh P.T., Vu N., Lai P.T., 2019. Sliding Mode Based Lateral Control of Unmanned Aerial Vehicles. Procedia Computer Science, 150, 78–87.

Wade J.J., McDaid L.J., Santos J., Sayers H.M., 2010. A spiking neural network training algorithm for classification problems. IEEE Transactions on Neural Networks, 21, 1817–1830.

Walter E., Pronzato L., 1997. Identification of parametric models from experimental data. Springer-Verlag.

Xiurui Xie, Hong Qu, Guisong Liu, Malu Zhang, Jürgen Kurths, 2016. An Efficient Supervised Training Algorithm for Multilayer Spiking Neural Networks, Plos one.