ISSN 1060-992X, Optical Memory and Neural Networks (Information Optics), 2015, Vol. 24, No. 3, pp. 201–208. © Allerton Press, Inc., 2015.

Learning of Semi-Empirical Neural Network Model of Aircraft Three-Axis Rotational Motion

M. V. Egorchev and Yu. V. Tiumentsev

Moscow Aviation Institute (National Research University), Moscow, Russia

e-mail: mihail.egorchev@gmail.com

Received April 7, 2015; in final form, June 4, 2015

Abstract—We analyze the problems of mathematical and computer simulation of the controlled motion of an aircraft when the knowledge about the object and its operation condition is insufficient. The goal of this paper is to develop a class of modular semi-empirical dynamic models that combine the capabilities of theoretical and neural network based simulations. We analyze the learning procedure of such models in generating a multi-step forecast.

Keywords: nonlinear dynamic system, semiempirical model, neural network simulation, sequential learning procedure

DOI: 10.3103/S1060992X15030042

INTRODUCTION

During the operation, the properties of aircraft may change in unpredictable ways. This fact ought to be considered when developing models used as the parts of the aircraft onboard systems. One of the possibilities to solve this problem is to develop models possessing the adaptivity.

In particular, we can obtain the required adaptive model using the semi-empirical approach proposed in [1] that allows us to combine the theoretical knowledge of the object with an improving of the model basing on the experimental data. This method provides a substantial increase of the accuracy of the solution comparing with the traditional purely empirical models such as the nonlinear autoregression exogenous model (NARX). At this theoretical knowledge is presented in the form of the differential equation systems, and the methods allow one to improve the model, using the methods of the theory of artificial neural networks (ANN). In [2] we discussed the problems of working out of the control actions providing the adequate mapping of specific features of the simulated system in the training set.

The learning of the semi-empirical neural network model is not a simple problem, because of its specificity as a dynamical neural network. In the following sections, we present the special procedure of the network learning to produce the multi-step forecast. We illustrate the efficiency of the proposed approach by the results of our computer simulations.

FORMATION OF SEMI-EMPIRICAL MODEL OF ANGULAR MOTION

To estimate the efficiency of the proposed approach, let us consider the problem of simulating the aircraft three-axis rotational motion. We describe it by a typical set of equations for aircraft flight dynamics [3]. This set consists of fourteen differential equations, which we do not present here because of it being too complicated. The state variables of the dynamic system are the roll angular rate p, the pitch angular rate q and the yaw angular rate r (degree/second); the roll angle Φ, the yaw angle Ψ and the pitch angle Θ (degree); the angle of attack α, the angle of sideslip β; the angle of the all-moving tailplane deflection δ_e, the angle of the rudder deflection δ_r, the angle of the aileron deflection δ_a (degree); the angular rates of the all-moving tailplane, the rudder and the aileron deflections $\dot{\delta}_e$, $\dot{\delta}_r$, $\dot{\delta}_a$ (degree/second), respectively. The control inputs are the quantities δ_e^{act}, δ_r^{act}, δ_a^{act} that are command driving signals supplied to the all-moving tailplane, the rudder and the aileron (degrees), respectively.

The given theoretical model contains six unknown nonlinear functions of several variables. These functions describe the dependences on the state variables of the coefficients of the axial $C_x(\alpha, \beta, \delta_e, q)$, the

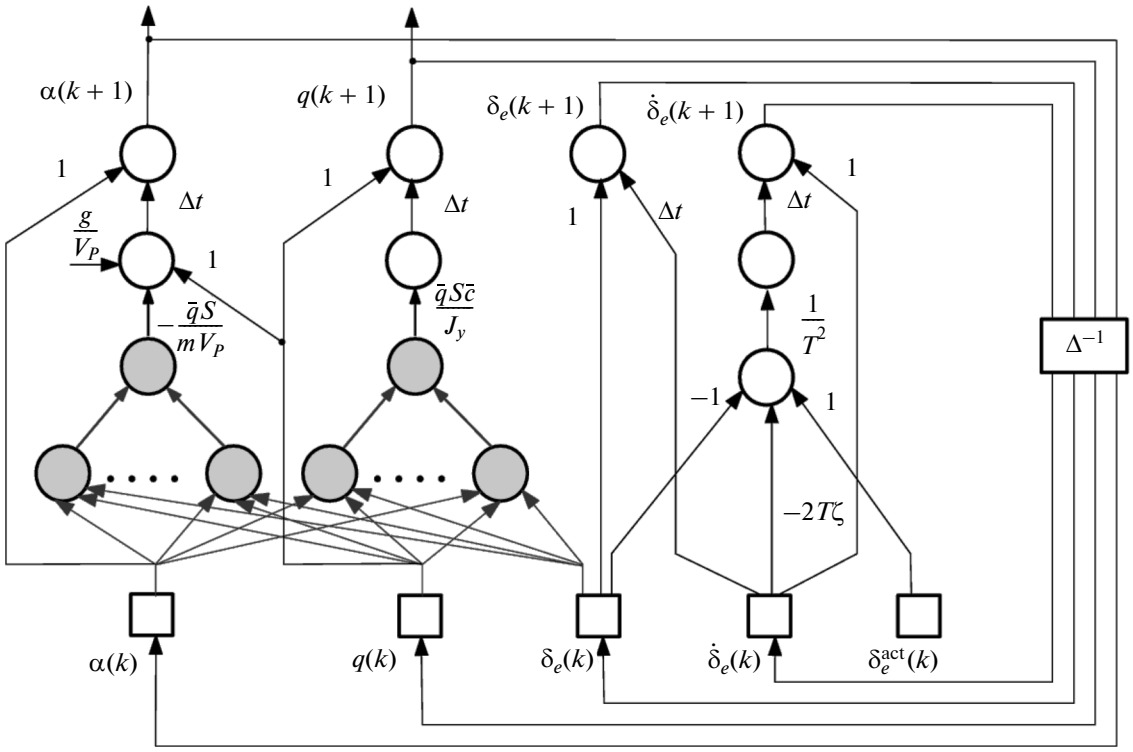

Fig. 1. The semi-empirical neural network model of the aircraft longitudinal rotational motion (based on the Euler difference scheme); the shaded elements of the scheme with the related connections are included into the neural network modules that provide the functions C_z and C_m that have to be reconstructed.

transverse $C_y(\alpha, \beta, \delta_r, \delta_a, p, r)$ and the normal $C_z(\alpha, \beta, \delta_e, q)$ of the aerodynamic forces, respectively. They also define aerodynamic moments $C_l(\alpha, \beta, \delta_e, \delta_r, \delta_a, p, r)$, $C_m(\alpha, \beta, \delta_e, q)$ and $C_n(\alpha, \beta, \delta_e, \delta_r, \delta_a, p, r)$ of the roll, the pitch and the yaw, respectively. To implement these functions we include in the developing model six neural network modules in the form of the sigmoid feedforward neural networks with one hidden layer. The hidden layers include 1, 5, 3, 5, 10 and 5 neurons for the modules C_x, C_y, C_z, C_l, C_m and C_n, respectively. In the concerned case of the three-axis rotational motion, it is not possible to present the block diagram of the model since it is too cumbersome. To give the reader an idea of its specific features in Fig. 1 we show the structure of the semi-empirical neural network model for the special case of the longitudinal rotational motion that includes the modules C_z and C_m only [2]. This is the condensed version of the full model. The analogous scheme for the case of the completely empirical model (NARX) of the longitudinal angular motion is shown in Fig. 2. In Figs. 1 and 2 the darkened circles show neurons with adjustable connection weights.

Note, since in the model there are no controls acting on the acceleration/ braking along the longitudinal axis of the aircraft, it is not possible to obtain the training set for the neural network module representing the drag coefficient C_x. This is the reason why we form the neural network module for C_x independently basing on the data from [4]. We embed it into the generated semi-empirical model. We "freeze" this module, that is, we disable variations of its adjustable parameters.

FORMATION OF REPRESENTATIVE TRAINING SET

When synthesising the neural network models one of the critically important problems is the formation of a representative data set characterizing the behaviour of the simulated dynamic system. Unfortunately, this problem has no simple solution, but it is crucial for obtaining a reliable model of the dynamic system.

In paper [2] we showed that as applied to the class of problems we consider, the most effective way is to use the polyharmonic (multisine) actuating signal. Under this approach for each of m aircraft controls the

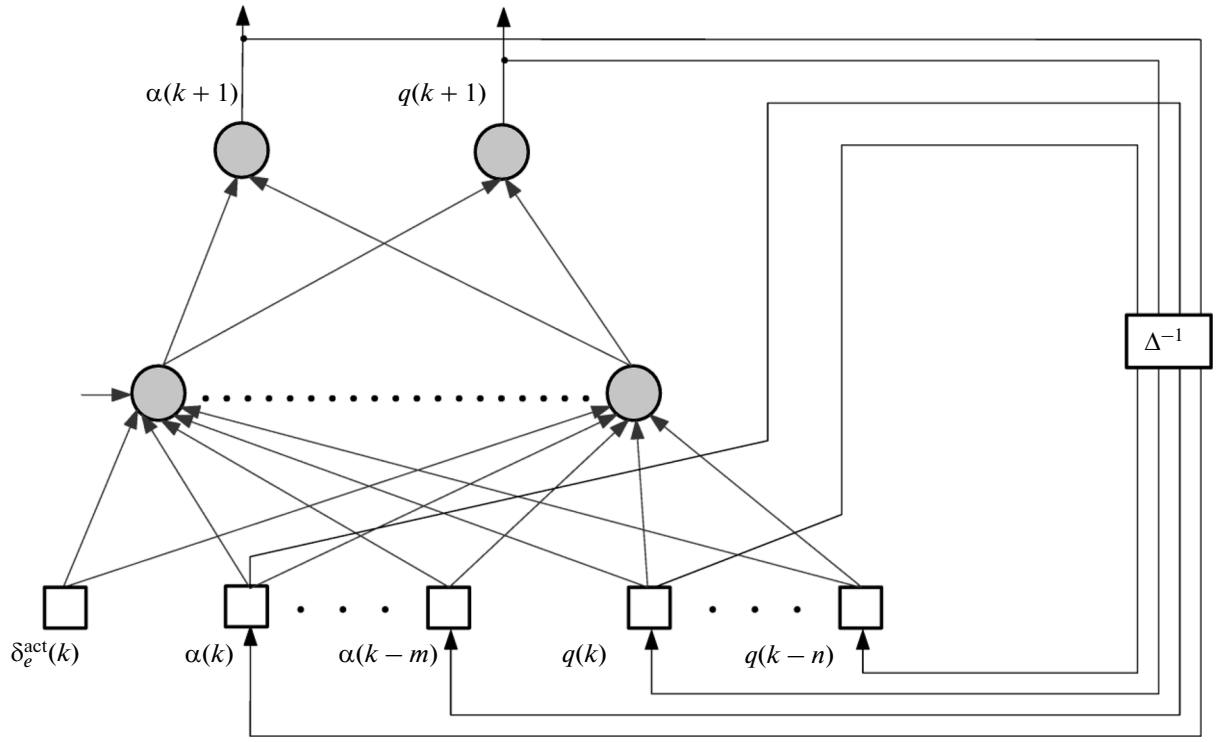

Fig. 2. The empirical model of the neural network for the aircraft longitudinal rotational motion (NARX).

input actions are formed as the sum of harmonic signals each of which has its own phase difference φ_k. The input signal u_j corresponding to the j-th control surface has the form:

$$u_j(t) = \sum_{k \in I_j} A_k \cos\left(\frac{2\pi k t}{T} + \varphi_k\right), \quad j = 1,\ldots,m, \quad I_j \subset K, \quad K = \{1,2,\ldots,M\}, \quad u_j^*(t) = \tilde{u}_j(t) + u_j(t),$$

where M is the total number of harmonically connected frequencies, T is the interval of time during which the test actuating signal acts upon the dynamic system; A_k is the k-th amplitude of the sinusoidal component; $u_j^*(t)$ is the complete controlling action for the j-th control surface; $\tilde{u}_j(t)$ is the controlling action for the j-th control surface realizing the test maneuver.

When generating the training set as well as when testing the obtained semi-empirical neural network model the controlling actions affected the aircraft over all three channels simultaneously and at that the signals δ_e^{act}, δ_r^{act}, δ_a^{act} were formed as the polyharmonic ones when obtaining the training set and as random ones when testing the learned model.

COMPUTER SIMULATIONS

In our computer simulations using the theoretical model, the interval of time was $t \in [0, 20]$ seconds when obtaining data for the ANN-model learning, and it was $t \in [0, 40]$ seconds when testing the obtained neural network model. In both cases simulations were performed with the time step of $\Delta t = 0.02$ s for the partially observable state vector $y(t) = [\alpha(t); \beta(t); p(t); q(t); r(t)]^{\mathrm{T}}$. The additive white noise with the standard deviations $\sigma_\alpha = \sigma_\beta = 0.02$ deg and $\sigma_p = 0.1$ deg/s, $\sigma_q = \sigma_r = 0.05$ deg/s affected the system output $y(t)$. If the neural network model reproduces the original system perfectly well, the noise affecting the system output defines the error of simulation completely. Consequently, comparison of the error of simulation with the standard deviation of the noise allows us to judge how well our simulations were. The standard deviation of the noise can be regarded as the target error of the simulations.

The learning was carried out basing on the sample (training set) $\{y_i\}$, $i = 1, \ldots, N$, obtained for the initial theoretical model with the aid of the Matlab system using the Levenberg–Marquardt algorithm and the

root mean-square error (RMSE) criterion. The Jacobian matrix we calculated according the RTRL (Real-Time Recurrent Learning) algorithm [5].

PROCEDURE OF THE SEMI-EMPIRICAL MODEL LEARNING

In paper [6] it was shown that in the presence of the additive noise affecting the observable outputs of the dynamic system, in theory the recurrent neural network is the optimal model. However, the process of learning of such networks basing on long input sequences causes some difficulties. They are the existence of spurious valleys on the error surface [7], the effects of the exponential decrease or increase of the gradient norm [8], the possible unrestricted increase of the network outputs. That is the reason why using the gradient optimization methods one can obtain the global minimum only for a small set of the initial values for parameters of the network. If we pass on to the problem of finding of the initial network parameters whose values are close enough to the minimum, it may be assumed that this problem is very close to the previous one. That is, we need to find such a sequence of problems where the first one is simple enough and we can solve it for any initial values of the network parameters; each subsequent problem has to be similar to the previous one and their solutions have to be close in the space of the parameters values. The sequence of the problems converges to the initial problem we have to solve.

When learning stepwise the network using a sequence of the abovementioned problems, there is a chance to reach a rather deep minimum. Similar approaches were previously discussed in [9–12]. As a rule, the authors suggested to learn the network on the sequences of problems of increasing complexity. (However, it is not seemed to be a must). Generally, using of these approaches leads to a substantial improvement of the learning results. In the given case of the multistep forecast problems, it is natural to suggest the following sequence of the problems. The first one is the problem of the one-step forecast; next one is the problem of the two-step forecast and so on. The last one is the problem of the N-step forecast.

It is evident that the first problem is the simplest one. Moreover, when solving this problem the recurrent network is learned as an ordinary feedforward neural network. The problem of the N-step forecast is the most complex one since the longest sequence is used when learning the network. So, the objective function for the k-step forecast problem has the form:

$$J_k\left(\left\{x_i,u_i\right\}_{i=1}^{n},w\right)=\frac{1}{k(n-k)}\sum_{i=1}^{n-k}\sum_{j=1}^{k}\left\|x_{i+j}-\mathrm{net}(\ldots\mathrm{net}(x_i,u_i;w),\ldots,u_{i+j-1};w)\right\|,$$

where x_i is the vector of the state variables at discrete instants of time i; u_i is the vector of the control variables at discrete instants of time i; w is the vector of the adjustable parameters of the neural network model. The procedure of learning the neural network model is presented in Table 1.

Let us discuss a demonstration learning problem of the recurrent network with two neurons and one configurable parameter. This is a semi-empirical model of the following dynamic system

$$\begin{cases}x_{1,i}=\tanh(u_{i-1}+x_{1,i-1}+wx_{2,i-1}),\\ x_{2,i}=\tanh(x_{1,i-1}).\end{cases}\tag{1}$$

The vector of the state variables is partially observable: $y_i=x_{2,i}$. The additive white noise with the standard deviation $\sigma=0.001$ affects the system output. The training set is generated by the system (1) for the value of the parameter $w=0.5$ for the steps $i=1$, ..., 200 and 21 input values $x_{2,1}$ uniformly distributed at the interval $[-1; 1]$. We use the normal random sequence as the control signal u_i. In Fig. 3 we show the error surface of this neural network for the 199-step forecast problem. We also denote the initial value of the parameter w and the local minimum achieved with the aid of the gradient descent. In Fig. 4 we in turns show the error surfaces of the sequence of problems of the multistep forecast as well as the relative minimums reached with the aid of the same gradient descent for the same initial value w. We see that our approach ensures the possibility to reach the global minimum of the original problem. It also should be noted that in real-world multidimensional problems the possibility to find the global minimum is even more crucial and the advantages of the sequential learning are more important. We used this algorithm successfully when solving the stated problem of identification of aerodynamic coefficients for the 1000-step forecast. The obtained results are given in Table 2 and in Fig. 5. From the analysis of the results of our simulations, we conclude the following.

The most important characteristic of the generated model is its ability to generalize. For the neural network models that usually means the ability of the model to ensure the desired accuracy not only for the data used for the model learning, but also for any values and combinations of the control and state vari-

Table 1. Learning algorithm of the neural network model

Step	Operation
1	To prepare the training set $X^{\text{train}} \leftarrow \left\{x_i^{\text{train}}, u_i^{\text{train}}\right\}_{i=1}^{n}$ and the validation set $X^{\text{val}} \leftarrow \left\{x_i^{\text{val}}, u_i^{\text{val}}\right\}_{i=1}^{n}$, to choose target accuracy $\varepsilon^{\text{goal}}$
2	To choose the maximal permissible value of the error increase $\Delta^{\max}$
3	To choose at the training set the maximal permissible value of the number of stages with the error increase $s^{\max}$
4	To choose initial values w_0 of the network parameters (for example, the random ones)
5	At the validation set to define the running value of epochs with the error increase $s \leftarrow 0$ and the running value of the forecast steps $k \leftarrow 1$
6	To solve the optimization problem $w_1 \leftarrow \underset{w}{\operatorname{argmin}} J_1\left(X^{\text{train}}, w\right)$, $\varepsilon_1^{\text{train}} \leftarrow J_1\left(X^{\text{train}}, w_1\right)$
7	To return to the step 4 if $\varepsilon_1^{\text{train}} > \varepsilon^{\text{goal}}$
8	To calculate the error of the $(n-1)$-step forecast at the validation set $\varepsilon_1^{\text{val}} \leftarrow J_{n-1}\left(X^{\text{val}}, w_1\right)$
9	To set a new value of the forecast's steps $k^+ \leftarrow k$
10	To continue setting the new values of the forecast's steps $k^+ \leftarrow k^+ + 1$ till $k^+ \le n-1$ and $\varepsilon_{k^+}^{\text{train}} < \varepsilon_k^{\text{train}} + \Delta^{\max}$
11	To return to the step 4 if $k^+ = k$
12	To solve the optimization problem $w_{k^+} \leftarrow \underset{w}{\operatorname{argmin}} J_{k^+}\left(X^{\text{train}}, w\right)$, $\varepsilon_{k^+}^{\text{train}} \leftarrow J_{k^+}\left(X^{\text{train}}, w_{k^+}\right)$
13	To set $k^+ \leftarrow k^+ - 1$ and to return to the step 11 if $\varepsilon_{k^+}^{\text{train}} > \varepsilon^{\text{goal}}$
14	To calculate the error of the $(n-1)$-step forecast $\varepsilon_{k^+}^{\text{val}} \leftarrow J_{n-1}\left(X^{\text{val}}, w_{k^+}\right)$ at the validation set
15	To set $s \leftarrow s+1$ if $\varepsilon_{k^+}^{\text{val}} > \varepsilon_k^{\text{val}}$
16	To return to the step 4 if $s \ge s^{\max}$
17	If $k^+ < n-1$, to set $k \leftarrow k^+$ and to return to the step 9 or finish otherwise; w_{n-1} are the neural network parameters we are looking for

Table 2. Simulation errors at the test set for semi-empirical model at different learning stages

Number of steps of the forecast	RMSE_α	RMSE_β	RMSE_p	RMSE_q	RMSE_r
2	0.1376	0.2100	1.5238	0.4517	0.4523
4	0.1550	0.0870	0.5673	0.4069	0.2738
6	0.1647	0.0663	0.4270	0.3973	0.2021
9	0.1316	0.0183	0.1751	0.2931	0.0530
14	0.0533	0.0109	0.1366	0.1116	0.0300
21	0.0171	0.0080	0.0972	0.0399	0.0193
1000	0.0171	0.0080	0.0972	0.0399	0.0193

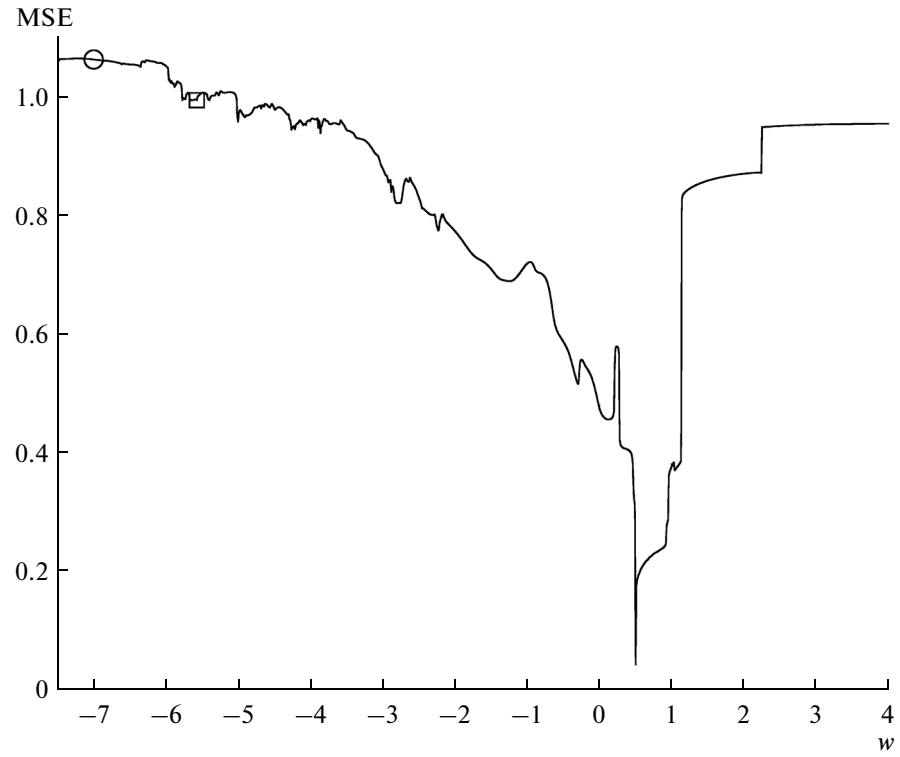

Fig. 3. The plot is the error surface for the demonstration problem, the circle is the initial value of the parameter, the square is the achieved local minimum.

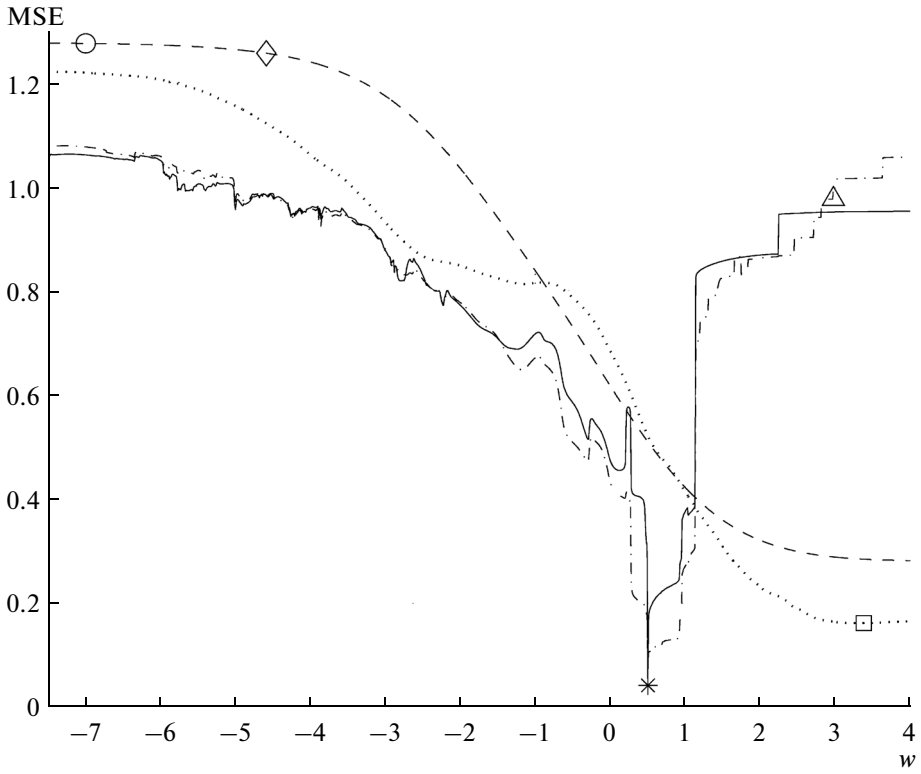

Fig. 4. The dashed, dotted, dash-and-dot and solid lines are the error surfaces for the forecast problem with 2, 7, 148 and 199 steps respectively. The circle is the initial parameter value, and the rhombus, square, triangle and star are the achieved minimums.

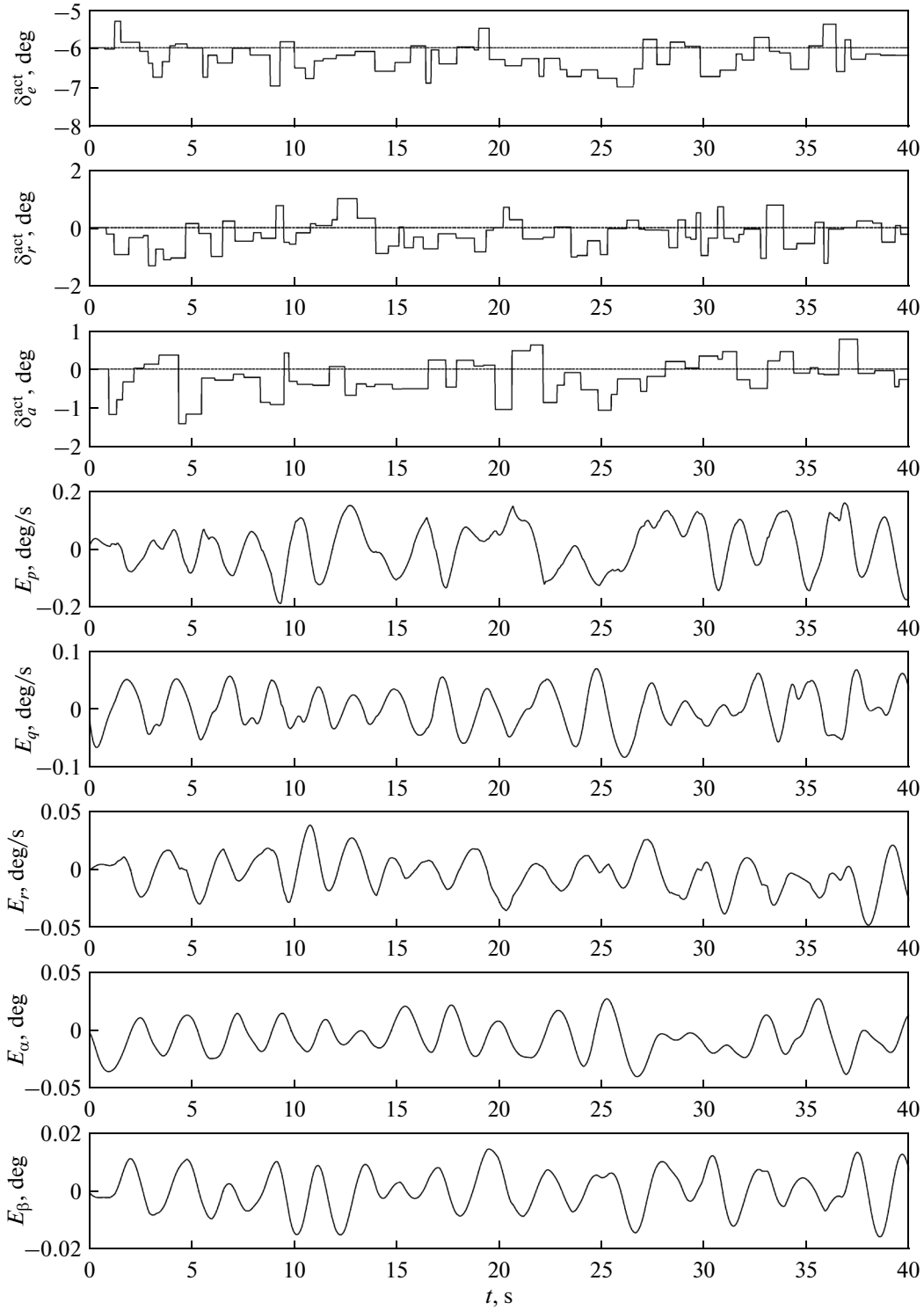

Fig. 5. The estimate of the generalization ability of the neural network model after completing of the 1000-step of the learning process: E_α, E_β, E_p, E_q, E_r are the measures of inaccuracy of the corresponding observed values; the horizontal lines show the values of the controls corresponding to the test maneuver (the absolute altitude is $h = 300$ m, the air speed is $V_P = 148$ m/s).

ables inside their definition domains. One does the relevant verification on the test data covering the abovementioned definition domain and not coinciding with the training data.

From Fig. 5 we see that the errors for all the observable state variables are sufficiently small. Moreover, these errors do not increase with time and this is the evidence of good generalization ability of the obtained neural network model. It must be emphasized that to ensure a great variety of the modeled system states as well as to ensure the largest possible variety of the differences between states at the adjacent instants of time, when testing the model we simulated very active work of the aircraft controls. An additional complicating factor was in providing the subsequent input disturbance affected the aircraft before the transition processes from one or several preceding disturbances were finished.

Fig. 5 corresponds to the model for which the learning cycle described above has been completed. The changes of the model accuracy in the intermediate points can be judged by the data in Table 2. One is also interested in the accuracy of solution when identifying the aerodynamic characteristics. We can estimate it comparing the values generated by the relevant neural network modules with experimental data in hand [4]. The values of the root mean-square errors (RMSE) of the reconstruction of each function provided by the corresponding neural network modules are $\mathrm{RMSE}_{C_y} = 5.4257 \times 10^{-4}$, $\mathrm{RMSE}_{C_z} = 9.2759 \times 10^{-4}$, $\mathrm{RMSE}_{C_l} = 2.1496 \times 10^{-5}$, $\mathrm{RMSE}_{C_m} = 1.4952 \times 10^{-4}$, $\mathrm{RMSE}_{C_n} = 1.3873 \times 10^{-5}$. At that, the error level does not change significantly with time. We did not find the fluctuations of the error that can influence negatively the adequacy of our semi-empirical neural network model.

CONCLUSIONS

The obtained results show that the methods of the neural network simulations combined with the knowledge and experience in the object domain, as well as with the representative training set are powerful tool for solving complicated problems for controlled dynamic systems of different kinds. The procedure based on using the sequence of the problems of increasing complexity proves to be a good learning tool for such models to provide the multistep forecast.

REFERENCES

1. Egorchev, M.V., Kozlov, D.S., Tiumentsev, Yu.V., and Chernyshev, A.V., Neural network based semi-empirical models for controlled dynamical systems, *Journal of Computer & Information Technology*, 2013, no 9, pp. 3–10 [in Russian].
2. Egorchev, M.V. and Tiumentsev, Yu.V., Training of neural network based semi-empirical models for controlled aircraft motion, *Proc. XVI All-Russian Scientific Engineering and Technical Conference "Neuroinformatics-2014",* 27–31 January, 2014, Moscow. Conference Proceedings, Part 2, Moscow: NRNU MEPhI, 2014, pp. 263–272 [in Russian].
3. Bochkariov, A.F., Andreyevsky, V.V., et al., *Aeromechanics of Airplane: Flight Dynamics*, 2nd ed., Moscow: Mashinostroyeniye, 1985 [in Russian].
4. Nguyen, L.T., Ogburn, M.E., Gilbert, W.P., Kibler, K.S., Brown, P.W., and Deal, P.L., Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability, *NASA TP-1538*, Dec. 1979.
5. Haykin, S., *Neural Networks: A Comprehensive Foundation*, 2nd ed, Prentice Hall, 2006.
6. Rivals, I. and Personnaz, L., Black-box modeling with state-space neural networks, *Neural Adaptive Control Technology, World Scientific*, 1996, pp. 237–264.
7. Horn, J., De Jesus, O., and Hagan, M.T., Spurious valleys in the error surface of recurrent networks—analysis and avoidance, *Proc. IEEE Trans. on Neural Networks,* 2009, vol. 20, no. 4, pp. 686–700.
8. Pascanu, R., Mikolov, T., and Bengio, Y., On the difficulty of training recurrent neural networks, available at: http://arxiv.org/abs/1211.5063, 2013.
9. Elman, J.L., Learning and development in neural networks: The importance of starting small, *Cognition*, 1993, vol. 48, pp. 71–99.
10. Ludik, J. and Cloete, I., Incremental increased complexity training, *Proc. ESANN 1994, 2nd European Sym. on Artif. Neural Netw.*, Brussels, Belgium, 1994, pp. 161–165.
11. Suykens, J.A.K. and Vandewalle, J., Learning a simple recurrent neural state space model to behave like Chua's double scroll, *Proc. IEEE Trans. on Circuits and Systems I: Fundamental Theory and Applications,* 1995, vol. 42, no. 8, pp. 499–502.
12. Bengio, Y., Louradour, J., Collobert, R., and Weston, J., Curriculum learning, *Proc. of the 26th Annual Intern. Conf. on Machine Learning,* ICML 2009, New York, NY, USA, pp. 41–48.