Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2017) 1–6

Procedia Computer Science

www.elsevier.com/locate/procedia

2017 Annual International Conference on Biologically Inspired Cognitive Architectures

Semi-empirical Neural Network Based Approach to Modelling and Simulation of Controlled Dynamical Systems

Mikhail V. Egorchev, Yury V. Tiumentsev

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

Abstract

A modelling and simulation approach is discussed for nonlinear controlled dynamical systems under multiple and diverse uncertainties. The main goal is to demonstrate capabilities for semi-empirical neural network based models combining theoretical domain-specific knowledge with training tools of artificial neural network field. Training of the dynamical neural network model for multi-step ahead prediction is performed in a sequential fashion. Computational experiments are carried out to confirm efficiency of the proposed approach.

Keywords: nonlinear dynamical system, semi-empirical model, neural network, sequential learning algorithm

1. Introduction

A modelling and simulation problem for multidimensional, highly nonlinear and nonstationary controlled dynamical system such as maneuverable aircraft is considered. Traditional approach to mathematical modelling and computer simulation of dynamical systems relies upon differential equations. However, dynamical system models based on differential equations lack adaptivity, which motivates the search for alternatives. One possibility would be to develop dynamical system models based on artificial neural networks (ANN). This would allow for model adaptivity, but at the same time it would significantly restrict level of complexity for the plant and thus prohibit application to most of the practical problems. The reason is that traditional ANN approach considers plant as a "black box" [1], which leads to significant increase of model dimensionality and, as a consequence, to increase of training dataset size up to values, unattainable in real world problems. Basic idea of suggested approach is to introduce available theoretical knowledge for the plant into the purely empirical model ("black box") in order to decrease both model dimensionality and required training set size. Such semi-empirical ("gray box") models [2, 3, 4] possess the required adaptivity feature and utilize both theoretical knowledge for the plant and experimental data of its behavior. Models of this class attain high accuracy and performance, as evidenced by computational experiments.

Learning recurrent neural networks to perform multistep prediction is a difficult optimization problem. In the following sections, we present sequential learning algorithm designed to circumvent some of the difficulties and illustrate efficiency of the proposed approach by results of computer simulations.

2. Semi-empirical neural network based model development

Development process for semi-empirical neural network based model of dynamical system consists of the following stages:

1. development of continuous-time theoretical model for the considered dynamical system as well as acquisition of experimental data about behavior of the system;
2. accuracy assessment for theoretical model of dynamical system using collected data;
3. conversion of original continuous-time model into a discrete-time model [5];
4. generation of ANN-representation for discrete-time model [6, 7];
5. learning of ANN-model [8];
6. structural adjustment of ANN-model to fit modelling accuracy requirements.

To estimate efficiency of proposed approach, let us consider a problem of modelling and simulation of aircraft three-axis rotational motion. Traditional continuous-time theoretical model for aircraft flight dynamics consists of 14 ordinary differential equations [9], omitted here for the sake of brevity. State variables of corresponding dynamical system include: roll angular rate p, pitch angular rate q and yaw angular rate r (degree/second); roll angle ϕ, yaw angle ψ and pitch angle θ (degree); angle of attack α, angle of sideslip β; angle of all-moving tailplane deflection δ_e, angle of rudder deflection δ_r, angle of aileron deflection δ_a (degree); angular rates of all-moving tailplane, rudder and aileron deflections $\dot{\delta}_e$, $\dot{\delta}_r$, $\dot{\delta}_a$ (degree/second), respectively. Control inputs include command signals supplied to all-moving tailplane, rudder and aileron δ_e^{act}, δ_r^{act}, δ_a^{act} (degrees), respectively. This theoretical model contains 6 unknown nonlinear functions of several variables that correspond to aerodynamic coefficients of axial $C_x(\alpha, \beta, \delta_e, q)$, transverse $C_y(\alpha, \beta, \delta_r, \delta_a, p, r)$ and normal $C_z(\alpha, \beta, \delta_e, q)$ aerodynamic forces, as well as roll $C_l(\alpha, \beta, \delta_e, \delta_r, \delta_a, p, r)$, pitch $C_m(\alpha, \beta, \delta_e, q)$ and yaw $C_n(\alpha, \beta, \delta_e, \delta_r, \delta_a, p, r)$ aerodynamic moments. These unknown functions are replaced by 6 feedforward neural network modules with one hidden layer. Hidden layers include 1, 5, 3, 5, 10 and 5 neurons with sigmoid activation functions for modules C_x, C_y, C_z, C_l, C_m and C_n, respectively. Output layer neurons are linear functions. This semi-empirical neural network based model has quite complex structure, thus we consider a restricted case of longitudinal rotational motion for illustration purposes: structure of this simplified model is given in Fig. 1. For comparison, structure of the completely empirical NARX model (Nonlinear AutoRegressive neural network with eXogenous inputs) is given in Fig. 2.

3. Sequential learning algorithm and simulation results

Learning of long input sequences with recurrent neural networks is difficult due to the existence of spurious valleys on the error surface [10], effects of exponential decrease or increase of the gradient norm [11], possible unbounded growth of network outputs. Thus, gradient optimization methods fail to find satisfactory solution except for rare occurrences when initial values for network parameters are very close to such solution. Now we might consider a problem of finding such initial values for network parameters. This problem may be stated as optimization problem for slightly perturbed objective function of original problem. Following this logic, we need to find such sequence of optimization problems that the first one is simple to solve for almost any initial values of network parameters, each subsequent problem is similar to the previous one and the sequence converges to the original difficult problem. Similar approach was previously discussed in [12, 13, 14, 15] and reported to provide a substantial improvement of learning results.

For a problem of learning recurrent neural networks to perform multistep prediction, it is natural to suggest a sequence of optimization problems generated by varying prediction horizon length k:

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

where x_i and u_i represent vector of state variables and vector of control variables at discrete instants of time i; w is the vector of the adjustable parameters of neural network model. We also use notation $\operatorname*{argmin}_w J_k(X, w; w_0)$

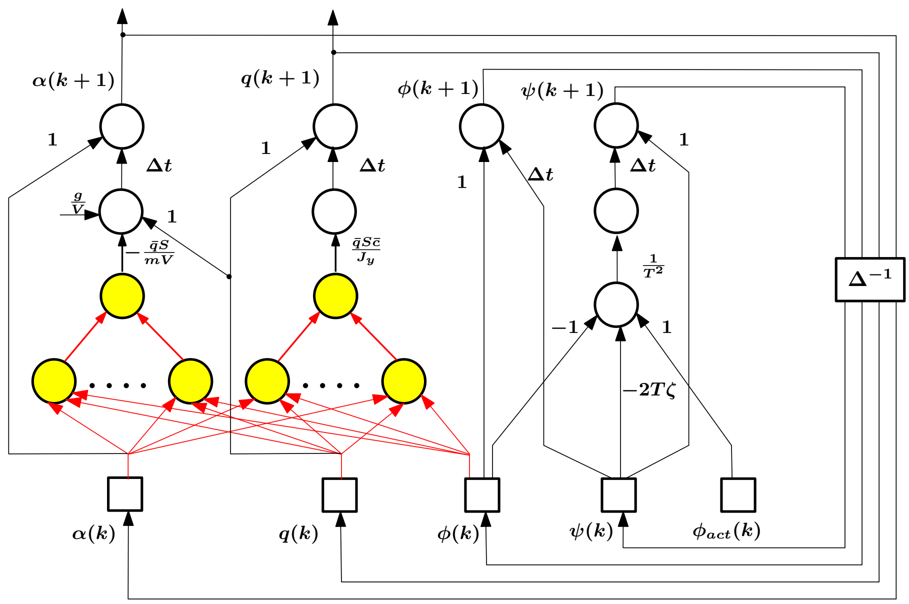

Fig. 1. Semi-empirical neural network model of aircraft longitudinal rotational motion (based on Euler difference scheme); shaded elements and related connections are included into neural network modules that represent functions Cz and C_m.

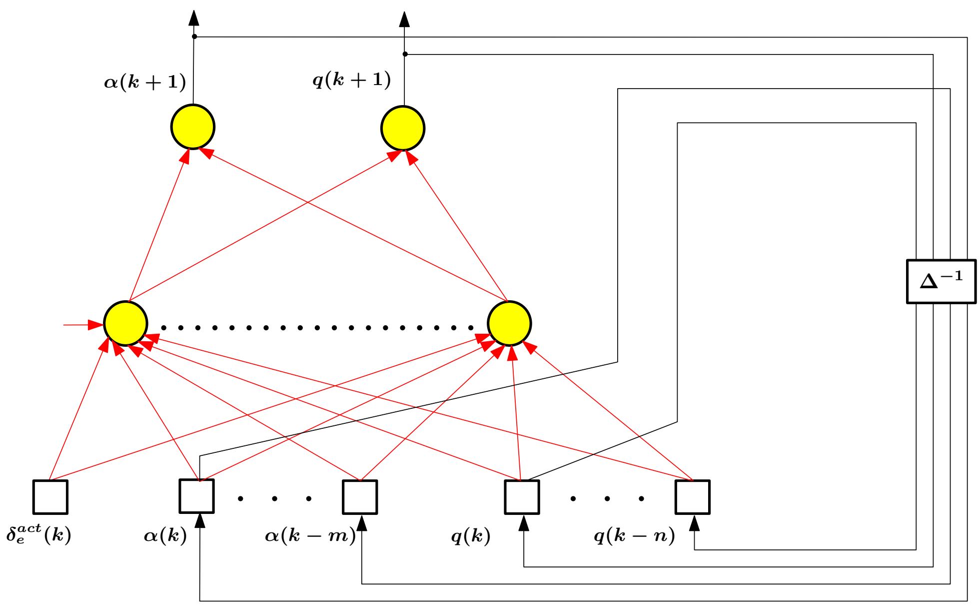

Fig. 2. Empirical NARX model of aircraft longitudinal rotational motion.

for a general iterative minimization algorithm applied to objective function J_k on dataset X using w_0 as initial guess for parameter values w. Proposed sequential learning algorithm basically works as follows: it starts by minimizing 1-step prediction objective function for $n-1$ initial states (all states x_i from a training set trajectory, except for the last one); then it increments prediction horizon, excludes one last initial state and performs minimization using previously found minima as initial guess; it terminates when prediction horizon equals $n-1$ and there is only one initial state x_1. The rest of the algorithm performs some form of steplength adaptation for prediction horizon increment.

Algorithm 1 Sequential learning algorithm for dynamic neural network model

1: Prepare training set $X^{\text{train}} \leftarrow \{x_i^{\text{train}}, u_i^{\text{train}}\}_{i=1}^{n}$
2: Prepare validation set $X^{\text{val}} \leftarrow \{x_i^{\text{val}}, u_i^{\text{val}}\}_{i=1}^{m}$
3: Choose target value $\varepsilon^{\text{goal}}$ for objective function
4: Choose maximum acceptable increase $\Delta^{\max}$ in objective function on training set
5: Choose maximum acceptable number of subsequent learning epochs $s^{\max}$ that increase objective function on validation set and initialize corresponding counter $s \leftarrow 0$
6: Choose initial values w^* for model parameters (for example, random ones)
7: Initialize current number of prediction steps $k^* \leftarrow 1$ and solve optimization problem $w^* \leftarrow \operatorname*{argmin}_w J_{k^*}(X^{\text{train}}, w; w^*)$
8: If $J_{k^*}(X^{\text{train}}, w^*) > \varepsilon^{\text{goal}}$, return to step 6
9: Find minimum number of prediction steps $k^+ \in [k^*, n-1]$ such that $J_{k^+}(X^{\text{train}}, w^*) \leqslant J_{k^*}(X^{\text{train}}, w^*) + \Delta^{\max}$
10: If $k^+ = k^*$, return to step 6
11: Perform backtracking to find maximum number of prediction steps $k^- \in [k^*, k^+]$ such that solution of optimization problem $w^- \leftarrow \operatorname*{argmin}_w J_{k^-}(X^{\text{train}}, w; w^*)$ would satisfy $J_{k^-}(X^{\text{train}}, w^-) \leqslant \varepsilon^{\text{goal}}$
12: If $k^- = k^*$, return to step 6
13: If validation error increases $J_{m-1}(X^{\text{val}}, w^-) > J_{m-1}(X^{\text{val}}, w^*)$, increment $s \leftarrow s + 1$
14: If $s > s^{\max}$, return to step 6
15: Accept $w^* \leftarrow w^-$ and $k^* \leftarrow k^-$
16: If $k^* < n-1$, return to step 9, otherwise terminate: w^* are the desired model parameters.

This algorithm has proven to be successful for learning semi-empirical recurrent neural network to perform 1000-step prediction of aircraft motion. Levenberg-Marquardt algorithm [16] was used to find solutions to intermediate optimization problems on steps 7 and 11 of Algorithm 1. Real-Time Recurrent Learning algorithm [17] was used for computation of Jacobi matrices. Representative training set is obtained using polyharmonic (multisine) excitation signal that has proven to be very effective for considered class of problems [18]. Test set is generated using random steps excitation signal. Simulation results are given in Table 1 and Fig. 3.

Table 1. Simulation errors on the test set for semi-empirical model at different learning stages

Number of prediction steps	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

From Fig. 3 we see that prediction errors for all observable state variables are sufficiently small. Moreover, these errors do not tend to increase with time which serves as evidence of good generalization ability

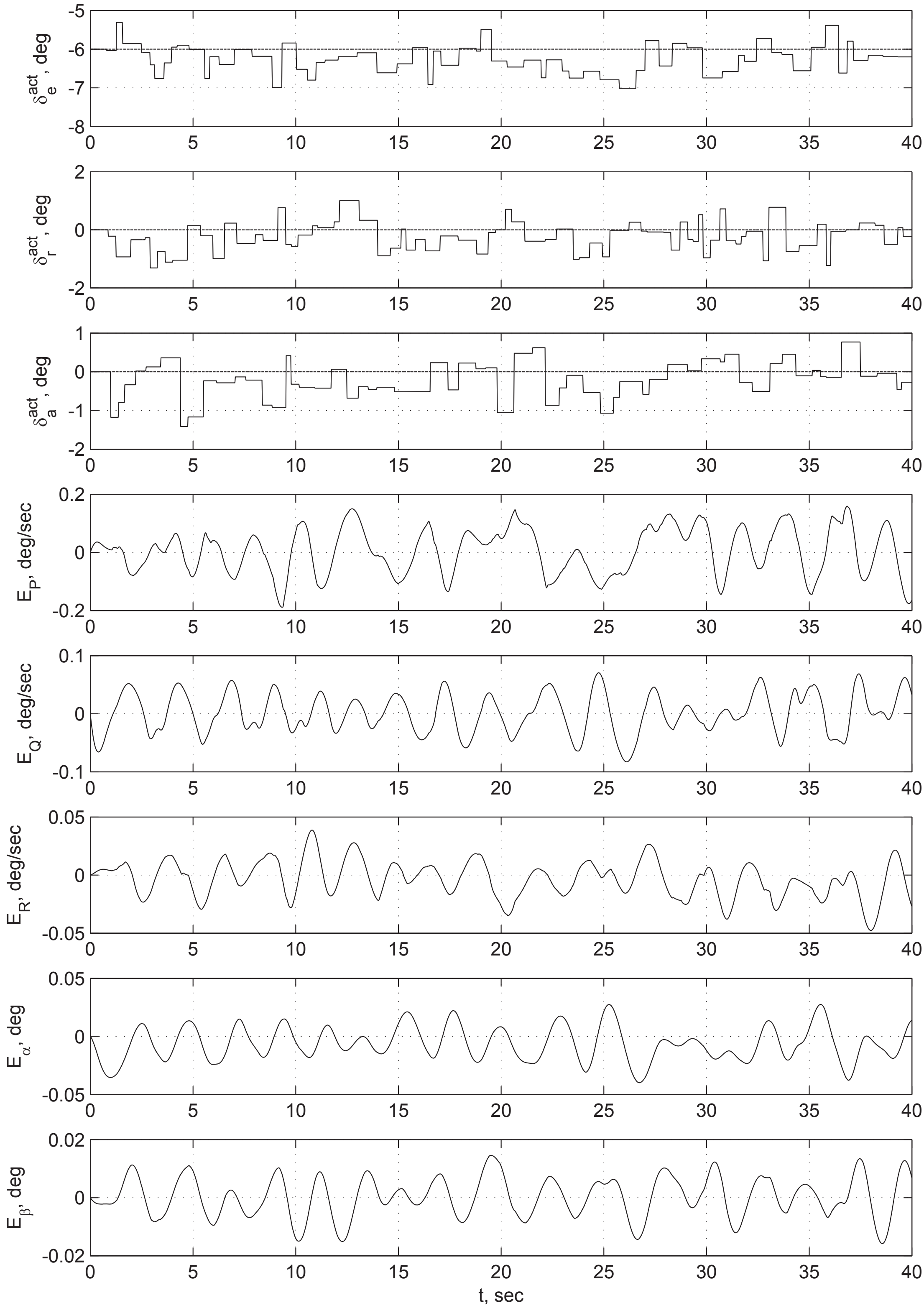

Fig. 3. Generalization error estimate: E_α, E_β, E_p, E_q, E_r represent prediction errors for corresponding observed values; δ_e^{act}, δ_r^{act}, δ_a^{act} represent test excitation signals.

of neural network model. Changes of model prediction error on intermediate stages of learning algorithm are presented in Table 1. We can also estimate error of identified aerodynamic coefficients by comparing outputs of corresponding neural network modules with available experimental data [19]. Root mean-square errors for each module are: $\mathrm{RMSE}_{C_y} = 5.4257 \times 10^{-4}$, $\mathrm{RMSE}_{C_z} = 9.2759 \times 10^{-4}$, $\mathrm{RMSE}_{C_l} = 2.1496 \times 10^{-5}$, $\mathrm{RMSE}_{C_m} = 1.4952 \times 10^{-4}$, $\mathrm{RMSE}_{C_n} = 1.3873 \times 10^{-5}$.

It is important to note that conceptually similar approach was suggested in [12]. The algorithm imposed limitations on temporal window within which patterns could be processed by a recurrent neural network. This was performed by periodically restricting access to network prior internal states via the recurrent connections. These limitations were gradually relaxed as learning proceeded. Such approach was found to resemble conditions under which children learn natural language, namely the working memory capacity increase that occurs during maturational changes. It was hypothesized that these early limitations assist efficient learning of natural language.

4. Conclusions

Simulation results clearly demonstrate that proposed ANN-based approach to complex nonlinear dynamical systems modelling is very effective from the standpoint of simulation accuracy, especially if we combine ANN learning techniques with some knowledge about simulated object. This approach can be implemented for systems operating under various uncertainty conditions using adaptation mechanisms based on the ANN training tools. Suggested sequential learning algorithm also proves to be useful for learning such models to perform multistep prediction.

References

[1] I. Rivals, L. Personnaz, Black-Box Modeling with State-Space Neural Networks, Vol. 15 of Robotics and Intelligent Systems, World Scientific Pub Co Inc, 1996, pp. 237–264.

[2] M. V. Egorchev, D. S. Kozlov, Y. V. Tiumentsev, Neural network adaptive semi-empirical models for aircraft controlled motion, in: Proceedings of the 29th Congress of the International Council of the Aeronautical Sciences, Vol. 4, 2014.

[3] G. Dreyfus, Neural networks - methodology and applications, Springer, 2005.

[4] Y. Oussar, G. Dreyfus, How to be a gray box: dynamic semi-physical modeling, Neural Networks 14 (9) (2001) 1161–1172.

[5] L. R. Scott, Numerical Analysis, Princeton University Press, 2011.

[6] G. Dreyfus, Y. Idan, The canonical form of nonlinear discrete-time models, Neural Computation 10 (1) (1998) 133–164.

[7] O. Nerrand, P. Roussel-Ragot, L. Personnaz, G. Dreyfus, S. Marcos, Neural networks and nonlinear adaptive filtering: Unifying concepts and new algorithms, Neural Computation 5 (2) (1993) 165–199.

[8] O. Nerrand, P. Roussel-Ragot, D. Urbani, L. Personnaz, G. Dreyfus, Training recurrent neural networks: why and how? An illustration in dynamical process modeling, IEEE Transactions on Neural Networks 5 (2) (1994) 178–184.

[9] A. F. Bochkariov, V. V. Andreyevsky, V. M. Belokonov, V. I. Klimov, V. M. Turapin, Aeromechanics of Airplane: Flight Dynamics, 2nd Edition, Mashinostroyeniye, Moscow, 1985, [In Russian].

[10] J. Horn, O. D. Jesus, M. T. Hagan, Spurious valleys in the error surface of recurrent networks: Analysis and avoidance, IEEE Transactions on Neural Networks 20 (4) (2009) 686–700.

[11] R. Pascanu, T. Mikolov, Y. Bengio, On the difficulty of training recurrent neural networks, in: Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML'13, JMLR.org, 2013, pp. III–1310–III–1318.

[12] J. L. Elman, Learning and development in neural networks: the importance of starting small, Cognition 48 (1) (1993) 71 – 99.

[13] J. Ludik, I. Cloete, Incremental increased complexity training, in: ESANN 1994, 2nd European Symposium on Artificial Neural Networks, Brussels, Belgium, April 20-22, 1994, Proceedings, 1994.

[14] J. A. K. Suykens, J. Vandewalle, Learning a simple recurrent neural state space model to behave like chua's double scroll, IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 42 (8) (1995) 499–502.

[15] Y. Bengio, J. Louradour, R. Collobert, J. Weston, Curriculum learning, in: Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, ACM, New York, NY, USA, 2009, pp. 41–48.

[16] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

[17] O. D. Jesus, M. T. Hagan, Backpropagation algorithms for a broad class of dynamic networks, IEEE Transactions on Neural Networks 18 (1) (2007) 14–27.

[18] M. V. Egorchev, Y. V. Tiumentsev, Learning of semi-empirical neural network model of aircraft three-axis rotational motion, Optical Memory and Neural Networks 24 (3) (2015) 201–208.

[19] L. T. Nguyen, M. E. Ogburn, W. P. Gilbert, K. S. Kibler, P. W. Brown, P. L. Deal, Simulator study of stall/post-stall characteristics of a fighter airplane with relaxed longitudinal static stability, Tech. Rep. TP-1538, NASA (December 1979).