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

Neural network based semi-empirical models for dynamical systems represented by differential-algebraic equations of index 2

Dmitry S. Kozlov[a], Yury V. Tiumentsev[b]

[a]*Federal State Unitary Enterprise "State Research Institute of Aviation Systems", Moscow, Russia*
[b]*Moscow Aviation Institute (National Research University), Moscow, Russia*

Abstract

A simulation problem is discussed for nonlinear controlled dynamical systems represented by differential-algebraic equations of index 2. The problem is proposed to be solved in the framework of a neural network based semi-empirical approach combining theoretical knowledge for the object with training tools of artificial neural network field. Special form neural network based semi-empirical models implementing implicit Runge-Kutta method of numerical integration are proposed. The training of the semi-empirical model allows to elaborate the models of aerodynamic coefficients implemented as a part of it. The results of simulation using this elaboration procedure of lift coefficient for reentry hypersonic vehicle are presented.

Keywords: dynamical system, differential-algebraic equations, semi-empirical model, neural network based simulation

1. Introduction

Usually when modeling nonlinear dynamic systems we deal with incomplete and inaccurate knowledge of their characteristics. In particular, the behavior of an aerospace vehicle is characterized by possible unpredictable changes in its operation, which may be caused, for example, by failures and damages of its systems and structure. This feature should be taken into account when developing a model of the controlled object. One of possible ways to solve this problem is generation of models with adaptability feature. An approach was proposed in [1] to realize such kind of the models. According to this approach, semi-empirical artificial neural network (ANN) based model for controlled nonlinear dynamic systems was developed, which combines theoretical knowledge about the concerned system with empirical model refinement methods. Such models are well-suited for adaptation techniques. The simulation for several problems [2], [3] confirms high efficiency of the ANN based semi-empirical modeling approach in contrast to traditional black-box dynamic ANN models such as widespread NARX (Nonlinear AutoRegressive network with eXogeneous inputs) and NARMAX (Nonlinear AutoRegressive with Moving Average and eXogenous inputs) [4], [5]. The semi-empirical approach, unlike the traditional ANN based black-box one, assumes the generation of

Email addresses: dskozlov@2100.gosniias.ru (Dmitry S. Kozlov), tium@mai.ru (Yury V. Tiumentsev)

the “grey-box” models using the theoretical knowledge about the simulated object in a form of a system of ordinary differential equations (ODE) [1–3]. The original ODE system is transformed into the ANN form so that the methods of neural network learning can be used to improve it. Such approach reduces the number of the adjusting parameters of the model and increases its generalization properties. When using the same training set, the accuracy of the semi-empirical model is much higher comparing with NARX [1], [3]. However, in some cases, in addition to ODE, the model includes algebraic-type constraints, that is, it is a system of differential algebraic equations (DAE). An example of such a task is the control of the space shuttle during descent in the atmosphere.

A characteristic feature of the approach described in [1]–[3] is that here the generation of the neural network semi-empirical model is based on the explicit conditionally stable methods of numerical integration. This is the reason why it is impossible to use this approach directly for modeling the systems described by DAE. Therefore, the approach needs to be modified taking the specific character of DAE into account. For DAE, the concept of the index is introduced [6], characterizing the smallest number of analytic differentiations required for reducing the DAE to the ODE of explicit form. This concept is important when choosing the appropriate scheme of numerical integration for the DAE.

The aim of our paper is to improve the generation methods for ANN based grey-box models related to dynamical systems represented by DAE of index 2. These semi-empirical models are prospective for usage in algorithms of trajectory prognosis for aircraft descending in the upper atmosphere. The most widely used approach here is a trajectory control in accordance with drag acceleration versus relative velocity profile. In accordance with this approach the flight trajectory can be divided into separated parts. The motion along each part can be performed when state variables satisfy specific constraint in the form of algebraic equality. The values of control variable are usually calculated by the formula obtained from simplified system of equations of motion extended by specific algebraic equation [7]. For realization of the trajectory control algorithm it is required to obtain a precise model of space shuttle motion. This is the reason why it is important to develop methods allowing us to elaborate an object model basing on real data.

As already noted above, the ANN based NARX (Nonlinear AutoRegressive network with eXogeneous inputs) and NARMAX (Nonlinear AutoRegressive with Moving Average and eXogenous inputs) [4], [5] dynamical models were among the first used for creating adaptive models of controlled dynamic systems. The main shortcoming of these neural networks is their inability to provide a long-term forecast of the behavior of the modeling object. As an alternative to the neural networks of the NARX type it was proposed in [8] to use other ANN models where the right-hand sides of equations of original ODE system were approximated using a multilayer perceptron. In this ANN architecture, the forward Euler integration formula of the first order was implemented. The Runge-Kutta networks are an extension of this approach [9]. The integration formulas of the classical Runge-Kutta method are implemented in the structure of the neural network. The disadvantages of this approach include the dependence of the obtained neural network modules on the specific integration formula.

As an alternative to the NARX black-box approach in [1] semi-empirical models were proposed basing on the involvement of theoretical knowledge about the modeled object, which allows to improve generalizing properties of the model. The semi-empirical approach assumes the generation of grey-box models combining the theoretical knowledge about the object with the possibility to improve the model using the experimental data. The distinction of the semi-empirical approach from NARX is that in the first case when generating the model a part of connections between the state and control variables of the initial ODE system is embedded in the model without changing. It allows one to reduce the number of the adjusting parameters of the model and to increase its generalization properties.

In the papers [2] and [3] the problems of mathematical modeling and computer simulation of controlled aircraft motion are analyzed for the case of insufficient knowledge about the object properties and its operation conditions. It is proposed to seek a solution in this situation by means of the semi-empirical approach. In the semi-empirical models obtained, the Adams and the Euler integration formulas were used. As mentioned above the generation of semi-empirical models in [1]–[3] is based on the explicit conditionally stable methods of numerical integration. This approach is not applicable directly to simulate systems described by DAE. It has to be modified taking the specific character of DAE into account. Further, it will be shown how this modification can be carried out.

2. Semi-empirical models for DAE systems

Let us examine the system of the differential-algebraic equations of the semi-explicit form

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}, z, \mathbf{u}), \quad 0 = \mathbf{g}(t, \mathbf{y}), \tag{1}$$

where $\mathbf{y} = \mathbf{y}(t)$ is a vector of state variables of the system, $\mathbf{y}$ and $\mathbf{f}$ have the dimension l; $z = z(t)$ is DAE algebraic variable being the same time the state variable of the system 1, $\mathbf{u} = \mathbf{u}(t)$ are the control variables. The initial values are consistent if $0 = \mathbf{g}(t_0, \mathbf{y}_0)$ and $0 = \mathbf{g}_y(t_0, \mathbf{y}_0) \cdot \mathbf{f}(\mathbf{y}_0, z_0)$. It is proposed to use index reduction by differentiation [6]. At that, the algebraic constraint takes the form

$$0 = 2\dot{\mathbf{g}} + \mathbf{g}. \tag{2}$$

The use of one-step s-stage methods for numerical integration of index 2 DAE systems is promising. We propose to use implicit Runge-Kutta (IRK) method based on Radau IIA quadrature formulas [6]. Using of the implicit scheme suggests the solution of the nonlinear system of equations at each integration step.

The structural scheme of the semi-empirical model is shown in Fig. 1. All the generated semi-empirical models have the unified representation in the form of the neural network with delay lines and feedbacks to the neurons of the first layer. In the initial system of equations, we single out the variables that are the state variables of DAE system. The individual layers (R-layers) realize the right-hand sides of DAE. The connections between the state variables are taken into account in the structure of the neural network: only the values corresponding to the input variables of the initial equation are fed to the R-layers. For this purpose, the layers realizing separation of variables (S-layers and SF-layer) are incorporated into the neural network. The control signal ($U(t)$) is fed to the neural network input. The output layer realizes the observer equation of the dynamic system. In this paper the observer equation corresponds to one of the state variables of the system. Individual neural network modules corresponding to those parts of the initial model that have to be set are built in the R-layers. When generating the semi-empirical model these neural network modules have to be updated structurally and parametrically. The neural network training is done using the RTRL algorithm (Real-Time Recurrent Learning) [10]. All other layers including the ones realizing the integration scheme and separation of variables are “frozen” and they do not change.

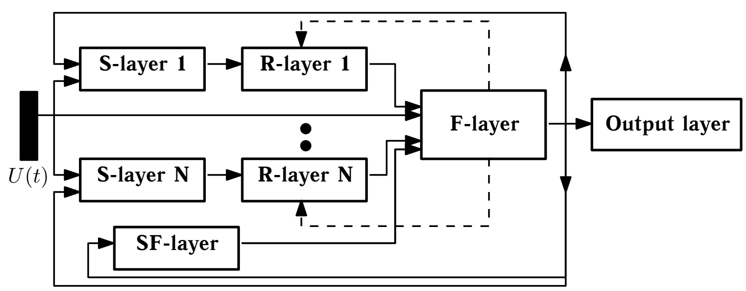

Fig. 1. The structural scheme of the semi-empirical model: the input to the semi-empirical model is the control signal $U(t)$; R-layer i is the neural network layer implementing the right-hand side of the i-th equation of the initial DAE system; S-layer i, SF-layer are the layers realizing the separation of the variables for the appropriate R-layer and F-layer; F-layer is the layer implementing the difference scheme; the output of the semi-empirical model is the values of the one of the state variables

The training set necessary for updating is generated as a sequence of observable outputs for the given controlling and initial conditions. At that, one uses a stochastic control input signal of a specific form defining the behavior of the simulated object along with the initial conditions [1].

In this paper, we propose an approach where the procedure containing the difference scheme is given in the form of an activation function of the F-layer. To realize the proposed approach we need to form the model of the neural network around the F-layer so that it would ensure the supply of the input values into the F-layer as well as the transfer of the training error to the layers implementing the right-hand sides of DAE system. Using the procedure implementing the differential scheme inside the activation function can be considered as a realization of the multivariable function in the layer. The realization of the F-layer has been described in detail by the authors in Refs [11], [12].

Inside F-layer, at each integration step, the transformation (3a) is performed, where $\mathbf{y}$ is the state variables vector of the DAE system, $\mathbf{F}$ is the right-hand sides of the DAE system. This transformation can be represented in the form of a functional $\Phi(\mathbf{F})$. To train the neural network it is necessary to calculate the Jacobi matrix for the activation function of the F-layer (3c). The right-hand sides of the DAE are independent of each other, so the Jacobi matrix elements which are not on the main diagonal are equal to zeros. Applying the variation technique to (3b), we obtain the expression for the element on the main diagonal of the Jacobi matrix (3d). The Jacobi matrix for activation function of the F-layer is equal to the matrix whose values on the main diagonal are the values of the integration step.

$$\mathbf{y}(t_{n+1}) = \mathbf{y}(t_n) + \int_{t_n}^{t_{n+1}} \mathbf{F}(\xi, \mathbf{y}(\xi))d\xi, \tag{3a}$$

$$\Phi(\mathbf{F}) = \mathbf{y}(t_n) + \int_{t_n}^{t_{n+1}} \mathbf{F}(\xi, \mathbf{y}(\xi))d\xi, \tag{3b}$$

$$J = \left[\frac{\partial \mathbf{y}(t_{n+1})}{\partial \mathbf{F}}\right], \tag{3c}$$

$$\frac{\delta\Phi(F_i)}{\delta F_i} = \frac{\int_{t_n}^{t_{n+1}} [F_i + \delta F_i]\, d\xi - \int_{t_n}^{t_{n+1}} F_i d\xi}{\delta F_i} = \frac{\int_{t_n}^{t_{n+1}} \delta F_i d\xi}{\delta F_i} = (t_{n+1} - t_n)\,. \tag{3d}$$

3. Simulation results

To estimate the workability of the proposed approach let us discuss the problem of the $(\alpha-\beta)$-controlling of aircraft descending in the upper atmosphere. In our computer simulations, we used the vehicle model from the paper [13]. In the equations, the centrifugal force resulting from the Earth rotation was taking into account. At this, the spherical Earth and the spherical geopotential function were supposed. Calculations were performed for an approximate exponential model of the atmosphere [7]. We examined the phase of the flight where the thrust force of the engine was equal to zero. The vehicle possesses a variable lift-to-drag ratio. The control parameters are the bank angle β and the angle of attack α. The aircraft trajectory is described by the equations typical for the dynamics of the reentry vehicles [7]:

$$\begin{gathered}
\dot{H} = V_R \sin\gamma,\ \dot{\xi} = \frac{V_R \cos\gamma \sin A}{r\cos\lambda}, \quad \dot{\lambda} = \frac{V_R}{r}\cos\gamma\cos A, \\
\dot{V}_R = \frac{-D}{m} - g\sin\gamma - \Omega_E^2 r\cos\lambda\,(\sin\lambda\cos A\cos\gamma - \cos\lambda\sin\gamma)\,, \\
\dot{\gamma} = \frac{L\cos\beta}{mV_R} + \frac{\cos\gamma}{V_R}\left(\frac{V_R^2}{r} - g\right) + 2\Omega_E\cos\lambda\sin A + \frac{\Omega_E^2 r\cos\lambda}{V_R}(\sin\lambda\cos A\sin\gamma + \cos\lambda\cos\gamma), \\
\dot{A} = \frac{L\sin\beta}{mV_R\cos\gamma} + \frac{V_R}{r}\cos\gamma\sin A\tan\lambda - 2\Omega_E(\cos\lambda\cos A\tan\gamma - \sin\lambda) + \frac{\Omega_E^2 r\cos\lambda\sin\lambda\sin A}{V_R\cos\gamma},
\end{gathered} \tag{4}$$

where H is the altitude (m); $r = H + a_e$ is the distance from the earth center to the center of mass of the vehicle (m); a_e is the earth radius (m); ξ is the longitude (deg); λ is the geocentric latitude (deg); V_R is the relative velocity (m/sec); γ is the relative flight path angle (deg); A is the relative azimuth (deg); β is the bank angle (deg); α is the angle of attack (deg); Ω_E is the earth rotational rate (1/sec); $g = \mu/r^2 2$ is the geopotential function (m/sec^2); μ is the gravitational constant (m^3/sec^2); m is the mass of the vehicle (kg); $L = 0.5\,C_L S\,V^2$ is the lift force; $C_L = C_{L0}|\sin\alpha|\sin\alpha\cos\alpha$ is the lift coefficient; $D = 0.5\,C_D S\,V^2$ is the drag

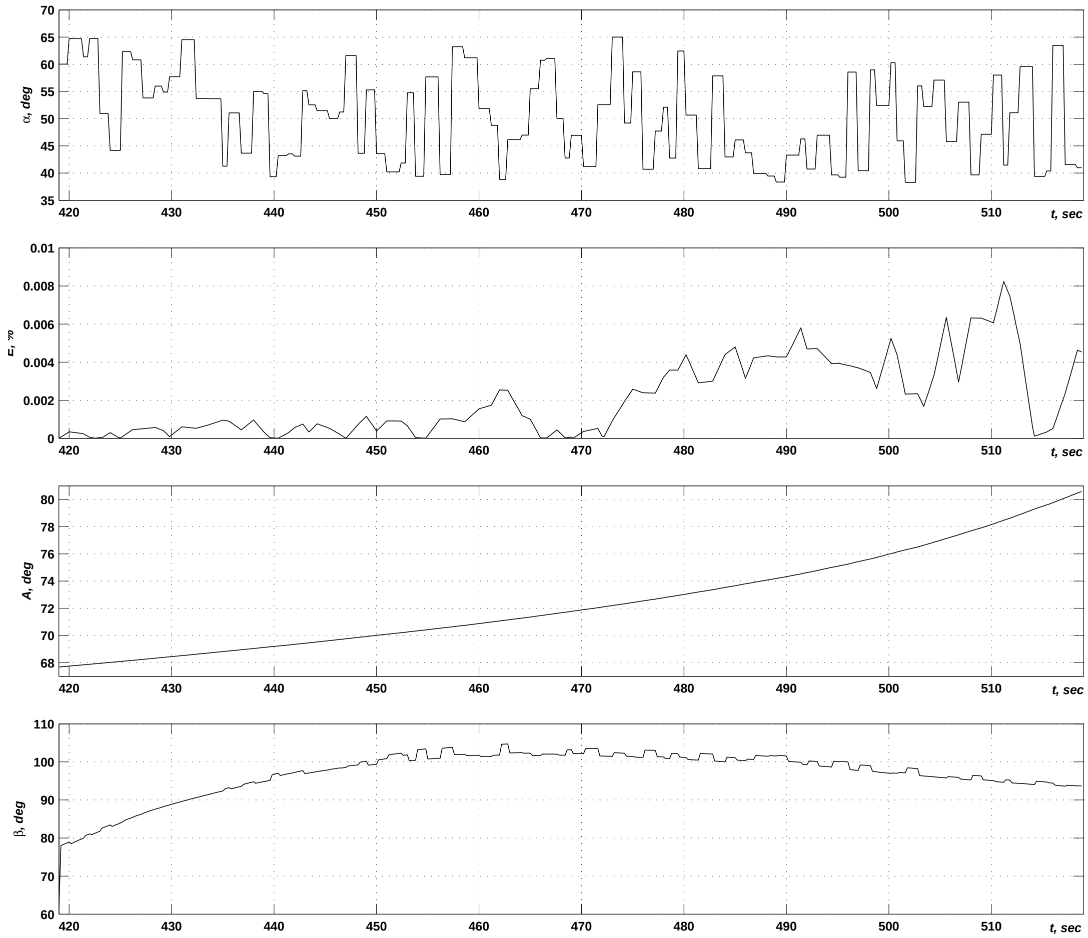

Fig. 2. The neural network output for values from the testing set: α is the control signal; E is the absolute error of the network output A; A is the relative azimuth; β is the bank angle

force; $C_D = C_{D0} + |\sin^3\alpha|$ is the drag coefficient; $\rho = \rho(H)$ is the density of atmosphere; S is the vehicle wing area (m^2). The algebraic constraints in this problem have the form [7]:

$$0 = \gamma + 1 + 9\left(\frac{t}{200}\right)^2, \quad 0 = \dot{\gamma} + \frac{18t}{40000},$$
$$0 = \frac{18t}{40000} + \frac{L\cos\beta}{mV_R} + \frac{\cos\gamma}{V_R}\left(\frac{V_R^2}{r} - g\right) + 2\Omega_E\cos\lambda\sin A + \frac{\Omega_E^2 r\cos\lambda}{V_R}(\sin\lambda\cos A\sin\gamma + \cos\lambda\cos\gamma). \tag{5}$$

In DAE system the variables $[H, \xi, \lambda, V_R, A]$ are the state variables and α is control variable. The control variable β is treated as an algebraic variable. The equations of motion 4 are supplemented with the algebraic equality 5 describing the variation of relative flight path angle γ in the range of $[-1^\circ, 10^\circ]$. In accordance with the form of the system (4)–(5) it can be treated as the index-2 DAE system. Equations (5) contain also differentiated algebraic equation.

For aircraft equations of motion semi-empirical model is generated in the form of the modular neural networks. MATLAB system and Neural Network Toolbox package were used when implementing the semi-empirical models as well as in the course of the computer simulations. The neural network module is trained to reproduce this dependence and it is added to the semi-empirical model. Then it is proposed that due to the occurrence of the failure situation the dependence C_L changes and takes the form:

$$C_L = C_{L0}|\sin\alpha|\sin\alpha\cos\alpha + \tau\sin 80\alpha. \quad (6)$$

In the course of the training, the weight coefficients of the neural network modules are transformed to reproduce this new dependence. In the training set as the input data we use the values sequences of the angle of attack of the special form. The output data are the respective sequences of the values of A.

During simulation we used $t_0 = 419$ sec and 400 iterations were performed with the integration step $\triangle t = 0.2$ sec. The starting conditions were == 75883 m, $\xi = 183.83878°$, $\lambda = 34.413°$, $V_R = 7260.512$ m/sec, $\gamma = -1°$, $A = 67.681°$, $\beta = 62.0399°$, $\alpha \in [38°, 65°]$. The simulation parameters $S = 249.9092\,\mathrm{m}^2$, $m = 87045.283\,\mathrm{kg}$, $a_e = 6371203.92\,\mathrm{m}$, $\Omega_E = 7.29211585 \cdot 10^{-5}\,\mathrm{sec}^{-1}$, $\mu = 3.9860319994 \cdot 10^{14}$, $C_D(40°) = 0.8246153846$, $C_{D0} = 0.29344667$, $C_L(40°) = 0.8769230769$, $C_{L0} = 2.7705917645$, $\tau = 0.0128$.

To implement the equations of motion we used the semi-empirical model realizing IRK Radau IIA method [6]. As the neural network modulus for C_L we used the two-layered neural network of the perceptron type with 5 neurons in the hidden layer. In Fig. 2 we show the values of the angle of attack from testing set, the values of A calculated with the aid of the semi-empirical model, the algebraic variable β values, the relevant absolute error of the network output A. The mean square deviations for the training, the validation and the testing sets were 0.0018368, 0.0021458 and 0.0025168 respectively.

4. Conclusion

For the dynamic system represented by differential-algebraic equations of index 2, we suggested the method of generation and tuning of the modular semi-empirical neural network models. We implemented the method of the s-stage implicit numerical integration of differential equation systems within the semi-empirical neural network model.

The obtained results show that the semi-empirical approach is applicable for the neural network modeling of complex dynamic objects whose motion is described theoretically by index 2 systems of differential-algebraic equations.

References

[1] M. V. Egorchev, D. S. Kozlov, Y. V.Tiumentsev, A. V. Chernyshev, Neural network based semi-empirical models for controlled dynamical systems, Journal of Computer and Information Technology (9) (2013) 3–10, (In Russian).

[2] M. V. Egorchev, Y. V. Tiumentsev, Training of neural network based semi-empirical models for controlled aircraft motion, in: Proceedings of XVI International Conference ”Neuroinformatics–2014”, Vol. 2, NRNU MEPhI, Moscow, 2014, pp. 263–272, (In Russian).

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

[4] S. Chen, S. Billings, Representation of non-linear systems: the NARMAX model, International Journal of Control 49 (3) (1989) 1013–1032.

[5] D. S. Kozlov, Y. V. Tiumentsev, Heterogeneous NARX neural networks: application for failure identification, Neurocomputers: Development and Applications (9) (2012) 13–22, (In Russian).

[6] E. Hairer, G. Wanner, Solving ordinary differential equations II: Stiff and differential-algebraic problems, 2nd Edition, Springer Series in Computational Mathematics, Springer, 2002.

[7] V. A. Yaroshevsky, Spacecraft reentry, Nauka, Moscow, 1988, (In Russian).

[8] C. A. L. Bailer-Jones, D. J. C. MacKay, P. J. Withers, A recurrent neural network for modeling dynamical systems, Network: Computation in Neural Systems (9) (1998) 531–547.

[9] Y.-J. Wang, C.-T. Lin, Runge-Kutta neural network for identification of dynamical systems in high accuracy, IEEE Transactions on Neural Networks 9 (2) (1998) 294–307.

[10] 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.

[11] D. S. Kozlov, Y. V. Tiumentsev, Neural network based semi-empirical models for dynamical systems described by differential-algebraic equations, Optical Memory and Neural Networks (Information Optics) 24 (4) (2015) 279–287.

[12] D. S. Kozlov, Y. V. Tiumentsev, Neural network based semi-empirical models for dynamical systems represented by differential-algebraic equations of index-1, in: Proceedings of XVIII International Conference ”Neuroinformatics–2016”, Vol. 3, NRNU MEPhI, Moscow, 2016, pp. 61–71, (In Russian).

[13] K. E. Brenan, Numerical simulation of trajectory prescribed path control problems by the backward differentiation formulas, IEEE Trans. on Automatic Control AC-31 (3) (1986) 266–269.