Adaptive Neural Network Control of Underactuated System

Andrzej Burghardt and Zenon Hendzel
Department of Applied Mechanics and Robotics, Rzeszow University of Technology, W.Pola 2, 35-959 Rzeszow, Poland

Keywords: Neural Network, Underactuated Systems, Adaptive Control.

Abstract: The article presents a synthesis of the control system of an underactuated object of ball-beam type. Based on a mathematical description of the object, we proposed an adaptational control algorithm, ensuring stabilization of the ball position on the beam. The synthesis of the control system was conducted on the basis of Lyapunov's stability theory, using artificial neural networks in the adaptation process. The proposed solution was simulated with Matlab/Simulink software and verified on the real object.

1 INTRODUCTION

Control and modelling of non-linear mechanical systems, where the number of independent control signals is smaller than the number of degrees of freedom (underactuated systems, US) is often analyzed, among others, in these works (Blajer and Kolodziejczyk, 2007), (Leonard and Marsden, 2000), (Spong, 1997). The most popular systems of US type include: a ball rolling along a beam, a ball rolling across a plane, inverted pendulum system (Leonard and Marsden, 2000), two-dimensional gantry cranes and systems of masses connected with springs (Blajer and Kolodziejczyk, 2007), submarines (Leonard, 1997), helicopters, and rotor flying machines.

Analysis of the literature in the field emphasized the fact that mathematical models used in control algorithms are simplified; for example, gravitation and friction phenomena are neglected (Levine and Mullhaupt, 1999), (Lewis and Murray, 1995) which became the impulse for research in control and modelling of US type systems.

The article presents a synthesis of the control system of the underactuated object of the non-linear ball-beam type. The neural control of non-linear systems relays on using neural networks to compensate system nonlinearities and its unknown properties. The neural control systems generally consists of the neural compensator and the classical control element like e.g. PD controller, which generates the control signal at the beginning of the NN's weights adaptation process. In a case of disturbances, weights of NNs are adapted to reduce a change of controlled system dynamics. This approach ensure high control quality in a case of disturbances. In the proposed control systems, based on a mathematical description of the object, a neural control algorithm, ensuring stabilization of the ball position on the beam, was proposed. The synthesis of the control system was conducted on the basis of Lyapunov's stability theory, using artificial neural networks in the adaptation process. The obtained solution was simulated with Matlab/Simulink software and correctness of stabilization of the ball position on the beam was verified using rapid prototyping environment with a dSpace control-measuring card and ControlDesk software.

2 LINEAR IN THE PARAMETER NEURAL NETS

It is commonly known that neural networks have good properties with regard to static mapping. The use of neural networks for real time control may require reproducing the full dynamics of the controlled object, which might result in a large size of dynamic networks. Application of linear neural networks because of their weight, such as, for example, radial networks, B-spline type networks, and networks with functional extensions, prevents the problem of explosion of the solutions. Considering the non-linearity of the controlled object, a linear neural network whose first weight layer is randomly generated was used in this work to compensate for its non-linearity.

The structure of NNs used in the control system is very universal, where can be used many different activation functions. In the presented control systems

Burghardt A. and Hendzel Z..
Adaptive Neural Network Control of Underactuated System.
DOI: 10.5220/0004113505050509
In *Proceedings of the 4th International Joint Conference on Computational Intelligence* (NCTA-2012), pages 505-509
ISBN: 978-989-8565-33-4

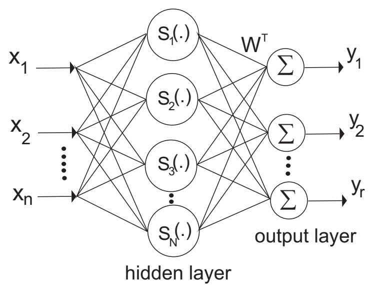

Figure 1: Structure of double-layer neural network.

were applied neural networks with sigmoidal bipolar neurons activation function, which are not local functions, like RBFs. This approach leads to the smaller size of the NNs with many inputs, in the comparison to the RBFNs, what is more adequate in the real time control tasks.The problems of modelling and control of non-linear objects are very complex. Because of the lack, so far, of a systematic approach to analysis and synthesis of dynamic non-linear systems, artificial neural networks have become an attractive tool applied in the theory of non-linear systems because of the possibility of approximation of any non-linear mappings and adaptation. Neural networks are applied, among other things, for modelling and control of complex non-linear systems. Let us analyze the neural network shown in fig. 1.

Mapping of entrance-exit of the network from fig. 1 assumes the following form

$$y_i = \sum_{j=1}^{N} \{w_{ij} S[\sum_{k=1}^{n} v_{jk} x_k + v_{vj}] + w_{wi}\} i = 1,..,r. \quad (1)$$

Assuming the element of entrance vector $x \equiv 1$ and threshold value vector of $[v_{v1}, v_{v2}, ..., v_{vn}]^T$ the following was recorded as the first column of, V^T matrix:

$$y(x) = W^T S(V^T x) \quad (2)$$

where $S = [S_1(.), S_2(.), ..., S_n(.)]^T$ constitutes a vector of the functions describing neurons, whose first element equals 1 while $[w_{w1}, w_{w2}, ..., w_{wn}]^T$ vector constitutes the first column of W^T. From the mathematical viewpoint a double-layer network may approximate a continuous function of many variables. Any continuous function $f : D_f \subset R^n \to R^\lambda$, where D_f is a compact R^n, sub-set, can be approximated with any accuracy by a double-layer neural network with properly selected weights. Which means, that for any compact D_f set and a positive value of approximation error ε there exists such double-layer neural network (fig. 1) that $f(x)$ function can be expressed as:

$$f(x) = W^T S(V^T x) + \varepsilon \quad (3)$$

for $||\varepsilon|| < \varepsilon_N$. If the first layer of V^T network weights is randomly designated, then W^T weights of the second layer define its properties, and in this case it is a single-layer network. If we define $\rho(x) = S(V^T x) + \varepsilon$, then we can write down the relationship (2) as:

$$y = W^T \rho(x) \quad (4)$$

where: $x \in R^n, y \in R^r, \rho(.) : R^n \to R^N$ and N is a number of neurons in the hidden layer. Such network is linear because of W^T weights and possesses approximation properties of non-linear functions. Sigmoid bipolar functions were assumed as the vector of basic functions of the network for approximation of non-linearity of the ball-beam system. Then the estimate of non-linear function of $f(x)$ is given by the following equation:

$$\hat{f}(x) = \hat{W}^T S(V^T x) \quad (5)$$

where V constitutes a constant matrix of weights of the entrance layer, randomly generated. The following relationship describes neuron activation functions:

$$S(V^T x) = \frac{2}{1 + exp(-\beta V^T x)} - 1 \quad (6)$$

where coefficient β is responsible for the function slope.

3 MODELLING AND CONTROL OF THE BALL-BEAM SYSTE

Dynamic equations of the ball-beam, shown fig. 2, could be recorded in the following form (Burghardt and Giergiel, 2011b), (Burghardt and Giergiel, 2011a):

$$M(a,q)\ddot{q} + C(a,q,\dot{q})\dot{q} + G(q) + \tau_d(t) = u \quad (7)$$

where: $q = [s_A \varphi]^T$ and matrices $M(a,q)$, $C(a,q,\dot{q})$ as well as vectors $G(q), u$, result from the description of dynamics of the analyzed system using Appell transformation (Blajer, 1998) and from the dynamics of executive systems.

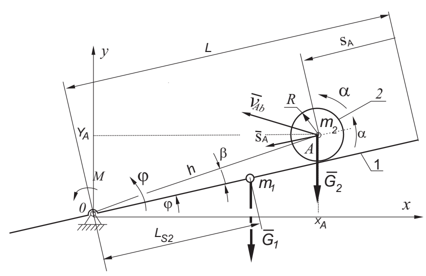

Figure 2: The ball-beam system.

The disturbance vector fulfils the restriction of $||\tau_d(t)|| < b$, $b = const > 0$. The matrices and vectors assume the following form:

$$M(a,q)=\begin{bmatrix} a_1 & a_1R \\ a_1R & a_1R^2+a_2+\frac{5}{7}a_1(L-s_A)^2 \end{bmatrix},$$
$$u=\begin{bmatrix} 0 & u_2 \end{bmatrix}^T$$

$$C(a,q,\dot{q})=\begin{bmatrix} 0 & \frac{5}{7}a_1(L-s_A)\varphi \\ -\frac{5}{7}a_1(L-s_A)\varphi & -\frac{5}{7}a_1(L-s_A)s_A \end{bmatrix},$$
$$\tau_d(t)=\begin{bmatrix} \tau_{d1} & \tau_{d2} \end{bmatrix}^T \tag{8}$$

$$G(q)=\begin{bmatrix} \frac{5}{7}a_1\sin(\varphi) \\ a_3g\cos(\varphi)+\frac{5}{7}a_1(L-s_A)\cos(\varphi) \end{bmatrix}$$

where: s_A, φ are generalized coordinates of the analyzed system, while u_2 is a moment generated by the engine driving the beam. Parameter vector $a=[a_1,a_2,a_3]^T$ contains parameters resulting from geometry, mass distribution, motion resistances as well as dynamic properties of the executive systems.The objective of synthesis of neural control algorithm is the determination of such a control law and network weights adaptation law that would allow realizing the set trajectory of $q_d=[s_{Ad},\varphi_d]$ form.

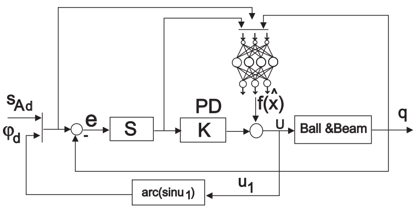

Figure 3: Structure of neural controller.

For this purpose we shall define lag error $e\in R^2$, generalized error $s\in R^2$ as well as auxiliary vector $v\in R^2$ as:

$$e=q_d-q, \tag{9}$$
$$s=e+\Lambda e, \tag{10}$$
$$v=q_d+\Lambda e, \tag{11}$$

where $\Lambda\in R^{2x2}$ is a diagonal positive-definite matrix. In this case the equation (7) can be transformed into the following form:

$$\begin{aligned} M(a,q)\dot{s}=-[u_1^*+u_2]^T+ \\ -C(a,q,\dot{q})s+f(x)+\tau_d(t) \end{aligned} \tag{12}$$

where: $u_1^*=\frac{5}{7}a_1gu_1$, $u_1=\sin(\varphi)$, u_1^* is a function of fictional control, which will be determined later, while vector function $f(x)$ has the following form:

$$f(x)=\begin{bmatrix} a_1[\dot{v}_1+R\dot{v}_2+\frac{5}{7}(L-s_A)\dot{\varphi}v_2] \\ a_1[R\dot{v}_1-\frac{5}{7}(L-s_A)(\dot{\varphi}v_1+\dot{s}_Av_2+ \\ -g\cos(\varphi))]+a_3g\cos(\varphi)+ \\ +\dot{v}_2[a_2+a_1(R^2+\frac{5}{7}(L-s_A))] \end{bmatrix} \tag{13}$$

where $x=[v^T,\dot{v}^T,q^T,\dot{q}^T]^T$. Let us select control signal $u=[u_1^*u_2]^T$ considering compensation of the controlled object's non-linearity:

$$u=\hat{f}(x)+K_Ds, \tag{14}$$

where $K_D=K_D^T>0$ is a design matrix, while the term K_Ds is a PD controller equation:

$$K_Ds=K_D\dot{e}+K_D\Lambda e. \tag{15}$$

In this system, the neural network task is compensation of non-linear vector function $f(x)$ of the controlled object, and the PD controller task is stabilization of the feedback control system. A linear neural network, described in chapter 2, was used for approximation because of weights.

In this case the non-linear function approximated by the network shall be recorded in the following form:

$$f(x)=w^T\varphi(x)+\varepsilon, \tag{16}$$

where ε is approximation error fulfilling $||\varepsilon||\leq\varepsilon_N$, $\varepsilon_N=const>0$ limitation. While the $f(x)$ function estimate shall be recorded as:

$$\hat{f}(x)=\hat{W}^T\varphi(x), \tag{17}$$

where $\hat{W}$ is the weight estimate of the ideal neural network. Using the relationship (17) we shall obtain a control law in the following form:

$$u=\hat{W}^T\varphi(x)+K_Ds. \tag{18}$$

By substituting equations (18) to relationship (12) we obtained:

$$M\dot{s}+C(\dot{q})s+K_Ds=\tilde{f}(x)+\tau_d(t), \tag{19}$$

where $\tilde{f}(x)$ function approximation error, $f(x)$, which is:

$$\begin{aligned} \tilde{f}(x)=f(x)-\hat{f}(x)=W^T\varphi(x)-\hat{W}^T\varphi(x)+ \\ +\varepsilon=\tilde{W}^T\varphi(x)+\varepsilon \end{aligned} \tag{20}$$

where $\tilde{W}=W-\hat{W}$ is the estimation error of neural network weights. Using (20) feedback control system equation (19) was recorded as follows:

$$M\dot{s}+C(\dot{q})s+K_Ds=\tilde{W}^T\varphi(x)+\varepsilon+\tau_d(t), \tag{21}$$

Lyapunov's stability theory was used in order to derive a weight adaptation algorithm $\hat{W}$of the network. If we select a square formula of the following form:

$$L=\frac{1}{2}s^TMs+\frac{1}{2}tr(\tilde{W}^TF^{-1}\tilde{W}), \tag{22}$$

where $F=F^T>0$ is the design matrix, it is possible to demonstrate, that selecting weights' adaptation law of neural network as:

$$\dot{\hat{W}}=F\varphi(x)s^T, \tag{23}$$

a derivative of square for (22) is a negative semi-definite, if the following dependence is fulfilled:

$$\psi = \{s : ||s|| > \frac{\varphi_N + b}{K_{Dmin}} \equiv b_s\}. \quad (24)$$

This results from the formula (24) that generalized lag error s is uniformly end-limited to ψ, set, with the practical boundary of b_s. By increasing K_D matrix coefficients it is possible to reduce lag error s, as well as errors e and $\dot{e}$, which are also limited. Such synthesis of adaptational neural control allows for the correct operation of a control system with PD controller until the neural network begins to learn. The conducted synthesis of neural control allows determination of a fictitious control signal

$$u_1 = \frac{7}{5a_1}[\hat{f}_1(x) + K_{D1}s_1]. \quad (25)$$

from which the set radius of the beam's own rotation φ_d in the following form was determined:

$$\varphi_d = arcsin(u_1). \quad (26)$$

The simulation and verification of the control algorithm was conducted on the basis of the given solutions.

4 COMPUTER SIMULATION

The proposed solution of a control system was simulated with Matlab/Simulink software, using the constructed emulator of the ball-beam system. The assumed values of masses and geometric sizes correspond to the real structure. The following data was assumed for simulation:$a_1 = 0,1329[kg]$, $a_2 = 0,0951[kgm]$, $a_3 = 0,0433[kgm]$, $L = 1[m]$, $R = 0,015[m]$. Mass moment of inertia of the beam and laser sensor was determined during the first approximation through modelling of mass distribution of these elements with a concentrated particle. It was assumed that the ball with the initial condition of $s_A = 0.02[m]$ should reach the set position $s_{Ad} = 0.5[m]$.

Fig. 4b presents errors accompanying realization of the task of reaching the set position by the ball. Figures 4c and 4d present components of the control signal, that is PD control as well as control compensating non-linearity of the object. Figs. 4e, and 4f present weight values of neural networks used in the control system. The conducted simulation tests demonstrated the correctness of theoretical considerations.

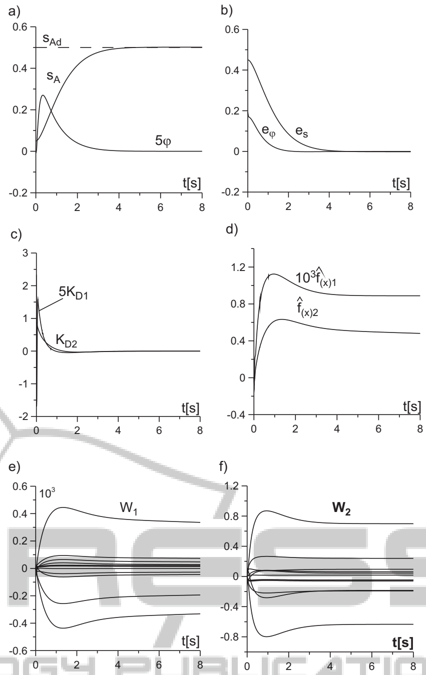

Figure 4: Results of simulation: a) desired (s_{Ad}) and actual position of the ball on the beam, beam angle φ_d, b) error of the ball position (e_s) and the beam orientation (e_φ), c) PD control signals, d) control signals of object's non-linarites compensation, e) weight of the first NN, f)weight of the second NN.

5 VERIFICATION

A mechanical system of the ball-beam type was constructed in order to verify the proposed control algorithm.

A mechanical system of the ball-beam type was constructed in order to verify the proposed control algorithm. A direct current engine integrated with a transmission and rotary-impulse transducer was used as a drive system, while a laser distance sensor was used to measure the ball location. ControlDesk, Matlab/Simulink software and dSpace 1104 card were used as the control-software environment. The results of verification obtained during the ball stabilization process (fig. 5)are analogical to simulation results (fig. 4). Small differences with regard to values and shapes of the variables result from simplifying assumptions adopted during the modelling process (not modeled dynamics of the motors with gears, not modeled friction in joints) as well as non-modelled disturbances occurring in the real object.

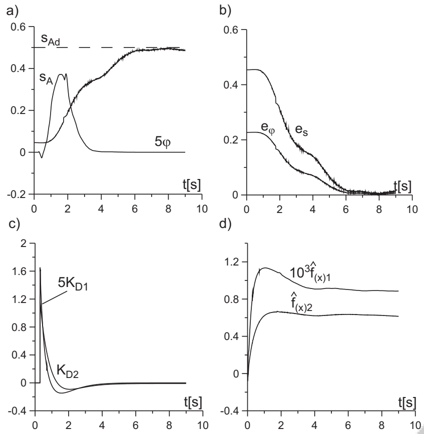

Figure 5: Results of verification: a) desired (s_{Ad}) and actual position of the ball on the beam, beam angle φ_d, b) error of the ball position (e_s) and the beam orientation (e_φ), c) PD control signals, d) control signals of object's non-linarites compensation.

6 CONCLUSIONS

A synthesis of a neural control algorithm allowing for stabilization of the ball location on the beam was conducted on the basis of mathematical model of the ball-beam system. Correctness of the adopted simplifying assumptions as well as correctness of the control system description was confirmed by simulation tests and verification conducted with the use of the object built by the authors.

ACKNOWLEDGEMENTS

This paper is supported by Polish Government under Cont. N N501 068838

REFERENCES

Blajer, W. (1998). *Metody dynamiki ukladow wieloczlonowych*. Wydawnictwo Politechniki Radomskiej, Radom.

Blajer, W. and Kolodziejczyk, K. (2007). Control of underactuated mechanical systems with servo-constraints. In *Nonlinear Dyn., N.4, Vol.50, pp.781-791*. Springer Science.

Burghardt, A. and Giergiel, J. (2011a). Mechatronic modelling and control of a underactuated system. In *10th Conf. on Dyn. Sys. Theory and Applications, pp. 717–722*. Lodz.

Burghardt, A. and Giergiel, J. (2011b). Modelling and control of a underactuated sphere and beam system. In *Communications in Nonlinear Science and Numerical Simulation, Vol. 16, N.5, pp. 2350–2354*. Elsevier.

Leonard, M. B. N. and Marsden, J. (2000). Controlled lagrangians and the stabilization of mechanical systems. In *The First Matching Theorem, IEEE Transactions on Automatic Control, N. 12, Vol. 45, pp. 2253–2270*. IEEE Press.

Leonard, N. (1997). Stability of a bottom-heavy underwater vehicle. In *Automatica, N.3, Vol.33, pp. 331–346*. Elsevier.

Levine, B. K. J. and Mullhaupt, P. (1999). Modelling flatness and simulation of a class of cranes. In *Ser. El. Eng. Vol.43 No.3 ,pp. 215–225*. Periodica Polytechnica.

Lewis, A. D. and Murray, R. M. (1995). Configuration controllability of simple mechanical control systems. In *SIAM Journal on Control and Optimization, N.3, Vol.35, pp.766–790*. SIAM Review.

Spong, M. (1997). Underactuated mechanical systems. In *Lecture Notes in Control and Information,Vol. 230*. Springer.