An Application of Speed Gradient Method to Neural Network Control for Underwater Robot

Alexander A. Dyda[1], Dmitry A. Oskin[1,2], Pavel A. Dyda[1]

[1] Maritime State University, 50[a], Verhneportovaya str., Vladivostok, Russia, 690003
Department of Automatic and Information Systems
[2] Far Eastern Federal University, 8, Sukhanova str., Vladivostok, Russia, 690950
Dept. of Information Control Systems
daoskin@mail.ru, adyda@mail.ru

Abstract: in this paper the speed gradient method is applied to design an adjustment algorithm for parameters of neural network controller. Local quadratic criterion expresses generalized error of desired trajectory tracking. Continuous adjustment laws for neural network parameters and their discrete analogies are derived on base of speed gradient method. To illustrate an approach, the mathematical model of underwater robot is taken. Numerical experiments had confirmed.

Keywords: speed gradient method, multilayer neural network, adjustment law, control, underwater robot.

1 Introduction

This paper is devoted to an application of speed gradient method to derive parameter adjustment (adaptation, learning) laws for multilayer neural network (NN) which is used to implement underwater robot (UR) control.

Underwater robots (UR) promise great perspectives and have a widest scope of applications in the area of ocean exploration and exploitation. To provide exact movement along prescribed space trajectory, UR needs a high quality control system. It is well known that UR can be considered as multi-dimensional nonlinear and uncertain controllable object. Hence, the design procedure of UR control laws is difficult and complex problem [4, 10].

Modern control theory has derived a lot of methods and approaches to solve appropriate synthesis problems such as nonlinear feedback linearization, adaptive control, robust control, variable structure systems etc [1, 5, 6]. However, most of mentioned methods of control systems synthesis essentially use information about structure of the UR mathematical model. The nature of interaction of a robot with water environment is so complicated that it is hardly possible to get exact detailed equations of UR movement. Possible way to overcome control laws synthesis problems can be found in the class of artificial intelligence systems, in particular, based on multi-layer neural networks (NN) [1, 2, 7].

In: A. Kononov et al. (eds.): DOOR 2016, Vladivostok, Russia, published at http://ceur-ws.org

Recently a lot of publications were devoted to the problems of NN identification and control, beginning from the basic paper [7]. Many papers are associated, in particular, with applications of NN to the problems of UR control [1, 2, 8].

Conventional applications of multi-layer NN are based on preliminary network learning. As a rule, this process is minimization of criterion that expresses summary deviations of NN outputs from desirable values with given NN inputs. Network learning results in NN weight coefficients adjustment. Such approach supposes the knowledge of teaching input-output pairs [7, 9].

The feature of NN application as a controller consists in the fact that desirable control signal is unknown in advance. Desirable movement trajectory (program signal) can be defined only for the whole control system [1, 2].

So, application of multi-layer NN in control tasks demands a development of approaches, which take into account dynamical nature of controllable objects.

In the paper the intelligent NN based control system for UR is designed. New learning algorithm for intelligent NN controller that uses speed gradient method is proposed. Numerical experiments with control system containing designed NN controller were carried out for cases of varying parameters and expressions for viscous torques and forces. Results of modeling are discussed.

Note that a choice of NN regulator is connected with principal orientation of neural network approach to a priori uncertainty that characterizes UR. In fact, matrices of inertia of UR rigid body are unknown exactly as well, as these of added water masses. Forces and torques of viscous friction are of unknown functional structure and also uncertain. Hence, UR can be considered as controllable object with partial parameter and structure uncertainties.

2 Underwater robot model

UR mathematical model traditionally consists of differential equations of kinematics

$$\dot{q}_1 = J(q_1)q_2 \tag{1}$$

and dynamics

$$D(q_1)\dot{q}_2 + B(q_1, q_2)q_2 + G(q_1, q_2) = U \tag{2}$$

where J the kinematical matrix; q_1, q_2 the vectors of generalized coordinates and body-fixed frame velocities of UR; U the control forces and torques vector; D the inertia matrix taking into account added masses of water; B the Coriolis – centripetal term matrix; G the vector of generalized gravity, buoyancy and nonlinear damping forces/torques [4].

Poor a priori knowledge of mathematical structure and parameters of matrices and vectors of the UR model can be compensated by intensive experimental research. As a rule, this way is expansive and takes a long time. One of perspective alternative approach is connected with usage of intelligent NN control

3 Intelligent NN controller and learning algorithm derivation

Our objective is synthesis of underwater robot NN controller to provide its movement along prescribed trajectory q_{d1}(t), $q_{d2}(t)$.

First we consider the control task with respect to velocities $q_d(t)$. Define error

$$e_2 = q_{d2} - q_2 \tag{3}$$

and introduce the local criterion (performance index) Q as measure of difference between desirable and real trajectories:

$$Q = \frac{1}{2} e_2^T D e_2 \tag{4}$$

Further we use the speed gradient method developed by [5, 6]. The main idea of speed gradient method consist in such adjustment of available controlled parameters that time derivative of chosen local or integral criterion (or their combinations)calculated along a system trajectory tends to negative value. If this is a case, a criterion which expresses an aim of control is minimizing. According to the method, compute time derivative of Q:

$$\dot{Q} = e_2^T D \dot{e}_2 + \frac{1}{2} e_2^T \dot{D} e_2 \tag{5}$$

as

$$q_2 = q_{d2} - e_2 \tag{6}$$

one has

$$D(q_1)\dot{q}_2 = D(q_1)\dot{q}_{d2} - D(q_1)\dot{e}_2 \tag{7}$$

Using expression of first term from dynamics equation, one can get the following:

$$\begin{aligned} D(q_1)\dot{e}_2 = D(q_1)\dot{q}_{d2} + B(q_1, q_2)q_{d2} - \\ - B(q_1, q_2)e_2 + G(q_1, q_2) - U \end{aligned} \tag{8}$$

and time derivative of function Q can be written in the form

$$\begin{aligned} \dot{Q} = e_2^T (D(q_1)\dot{q}_{d2} + B(q_1, q_2)q_{d2} - \\ - B(q_1, q_2)e_2 + G(q_1, q_2) - U) + \frac{1}{2} e_2^T \dot{D} e_2. \end{aligned} \tag{9}$$

After terms reorganization, one get

$$\dot{Q} = e_2^T (D(q_1)\dot{q}_{d2} + (q_1, q_2)q_{d2} + G(q_1, q_2) - U) -$$
$$- e_2^T B(q_1, q_2)e_2 + \frac{1}{2} e_2^T \dot{D}(q_1)e_2 =$$
$$= e_2^T (D(q_1)\dot{q}_{d2} + B(q_1, q_2)q_{d2} + G(q_1, q_2) - U) +$$
$$+ \frac{1}{2} e_2^T (\dot{D}(q_1) - B(q_1, q_2)e_2).$$

As known, the matrix in last term is skew-symmetric, hence, this term is equal to zero and we have simplified expression:

$$\dot{Q} = e_2^T (D(q_1)\dot{q}_{d2} + B(q_1, q_2)q_{d2} + G(q_1, q_2) - U). \tag{10}$$

We plan to implement intelligent UR control [1] based on neural network. Without losing of generality of the approach, choose two-layer NN (Fig. 1). Let hidden and output layers have H and m neurons appropriately (m is equal to dimension of e_2). For the sake of simplicity, one supposes that only summing of weighted signals (without nonlinear transformation) is realized in output layer. Input vector has N coordinates.

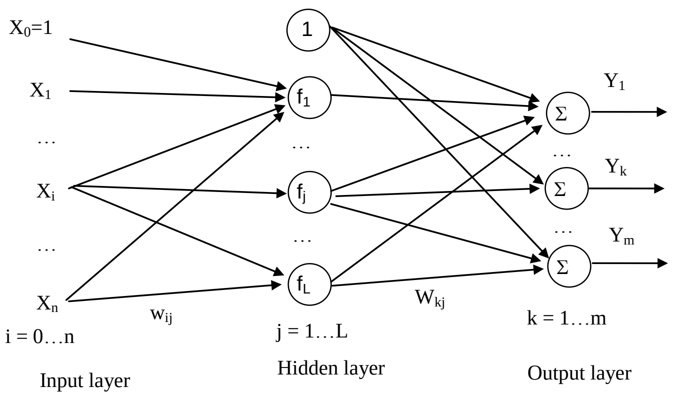

Fig. 1. Neural network structure

Define w_{ij} as weight coefficient for i-th input of j-th neuron of hidden layer. So these coefficients compose matrix

$$w = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1N} \\ w_{21} & w_{22} & \dots & w_{2N} \\ \dots & \dots & \dots & \dots \\ w_{H1} & w_{H2} & \dots & w_{HN} \end{bmatrix} \tag{11}$$

As result of nonlinear transformation $f(\cdot)$, hidden layer output vector can be written in the form

$$f(w,x) = \begin{bmatrix} f_1(w_1^T x) \\ \dots \\ f_H(w_H^T x) \end{bmatrix} \tag{12}$$

where w_k denotes *k*-th raw of matrix w.

By analogy, introduce matrix W which element W_{li} denotes weight coefficient from *i*-th neuron of hidden *l*-th neuron of output layer.

With defined NN parameters, the underwater robot control signal (NN output) is computed as following:

$$\mathrm{U} = \mathrm{y}(\mathrm{W}, \mathrm{w}, \mathrm{x}) = \mathrm{Wf}(\mathrm{w}, \mathrm{x}) \tag{13}$$

Substitution of this control let us to get

$$\begin{aligned} \dot{Q} = e_2^T (D(q_1)\dot{q}_{d2} + B(q_1, q_2)q_{d2} + \\ + G(q_1, q_2) - Wf(w, x)). \end{aligned} \tag{14}$$

To derive NN learning algorithm, apply the speed gradient method [5, 6]. For this, compute partial derivatives of function Q time derivative with respect to adjustable NN parameters – matrices w and W.

Direct differentiation gives

$$\frac{\partial \dot{Q}}{\partial W} = -e_2 f^T(w, x). \tag{15}$$

It is easy to demonstrate that choosing of all activation functions in the usual form

$$f(x) = 1/(1 + e^{-\tau x}) \tag{16}$$

imply property

$$\frac{\partial}{\partial w_{ij}} f_i(w_i^T x) = f_i(w_i^T x)[1 - f_i(w_i^T x)]x_j \tag{17}$$

Introduce additional functions

$$\varphi_i(w_i^T x) = f_i(w_i^T x)[1 - f_i(w_i^T x)] \tag{18}$$

and matrix

$$\Phi(w, x) = diag(\varphi_1(w_1^T x) \dots \varphi_H(w_H^T x)) \tag{19}$$

Direct calculation gives

$$\frac{\partial \dot{Q}}{\partial w} = -\Phi W^T e_2 x^T \tag{20}$$

As a final stage, we can write the NN learning algorithm in following form:

$$\begin{gathered} W^{(k+1)} = W^{(k)} + \gamma e_2 f^T(w, x) \\ w^{(k+1)} = w^{(k)} + \gamma e_2 \, \Phi \, W^T e_2 x^T \end{gathered} \tag{21}$$

(γ is learning step, k is number of iteration).

Now consider which components should be included in NN input vector. As NN controller is oriented to compensate an influence of appropriate matrix and vector functions, in common case the NN input vector must be composed of q_1, q_2, e_2, q_{d2} and its time derivative.

The NN learning procedure leads to reducing of function Q, consequently in ideal conditions, error e_2 tends to zero and the UR movement follows to desirable trajectory

$$q_2(t) \to q_{d2}(t) \tag{22}$$

If UR trajectory is given by $q_{d1}(t)$, one can choose

$$q_{d2}(t) = J^{-1}(q_1)(\dot{q}_{d1}(t) + k(q_{d1}(t) - q_1(t))) \tag{23}$$

(k is positive constant). As follows from kinematics equation,

$$\dot{q}_1(t) \to \dot{q}_{d1}(t) + k(q_{d1}(t) - q_1(t)) \tag{24}$$

and

$$\dot{e}_1(t) + ke_1(t) \to 0 \tag{25}$$

where

$$e_1(t) = q_{d1}(t) - q_1(t) \tag{26}$$

Hence, UR follows to the planned trajectory $q_{d1}(t)$.

4 Simulation results of intelligent NN controller

To check the effectiveness of the approach, computer simulations have been carried. The UR nominal model parameters were taken from [8]. Parameters of UR are: $D = D_{RB} + D_A$, where $D_{RB} =$ [1000 0 200; 0 1000 0; 200 0 11000] - system inertia matrix for the rigid body, $D_A =$ [1000 0 100; 0 1100 80; 100 80 9000] - matrix of hydrodynamic added mass, $B =$ [210 20 30; 25 200 70; 15 33 1500], $G =$ [0; 0; 0].

Let consider the nominal model (with added mass) and reduced one.

Vector q_2 consists of following components (linear and angular UR velocities):

$$q_2 = \begin{bmatrix} v_x & v_z & \omega_y \end{bmatrix}^T \tag{27}$$

Dimensions of NN input (q_2 and e_2) and output (control forces and torque) are equal to 6 and 3.

$$U = \begin{bmatrix} F_x & F_z & M_y \end{bmatrix}^T \tag{28}$$

For the NN controller containing 10 neurons in the hidden layer, the simulation results are given on Figs. 2 – 10. In the considered numerical experiments, the desired trajectory was taken as follows:

$$\begin{cases} v_{xd} = 0.75 m/\sec, \\ v_{zd} = 0.5 m/\sec, \quad 0 \le t \le 250 \sec \\ \omega_{yd} = -0.15 rad/\sec, \end{cases}$$

$$\begin{cases} v_{xd} = 0.5 m/\sec, \\ v_{zd} = 0.75 m/\sec, \quad 250 \le t \le 500 \sec \\ \omega_{yd} = 0.15 rad/\sec, \end{cases}$$

Transient processes and control for taken nominal model are shown on Fig. 2 - 4.

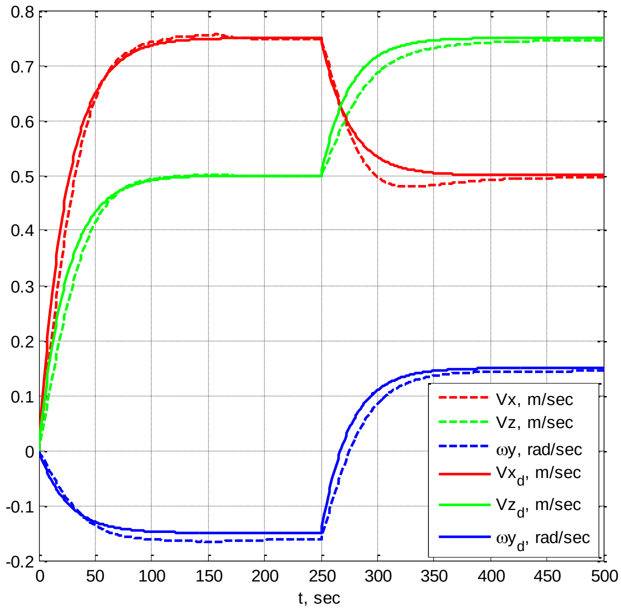

Fig. 2. Transient processes (nominal model)

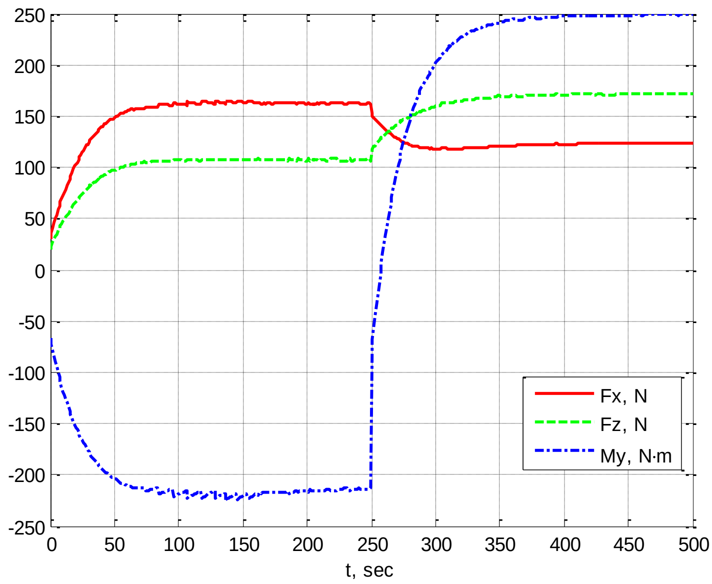

Fig. 3. Control signals (nominal model)

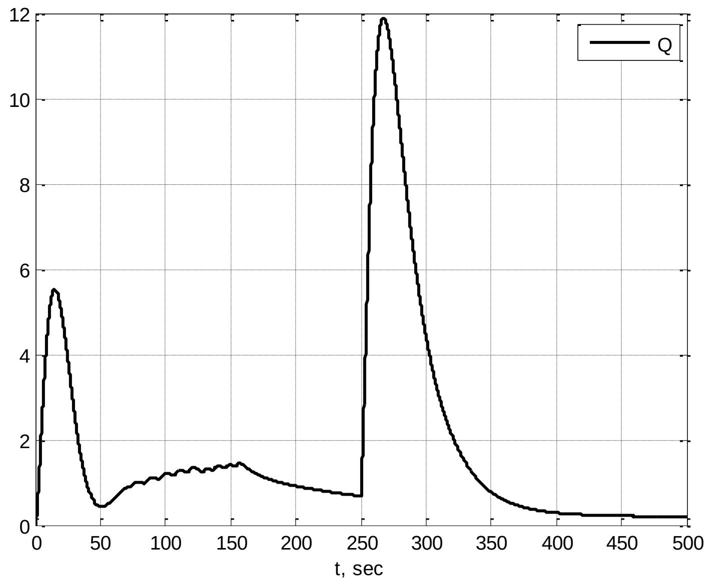

Fig. 4. Performance index (nominal model)

Fig. 5 - 7 present the same processes for the case of reduced UR added masses.

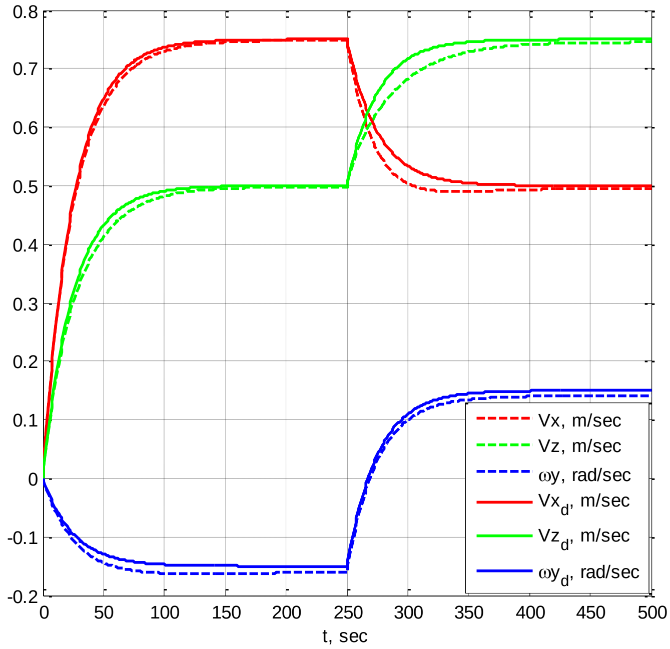

Fig. 5. Transient processes (reduced model)

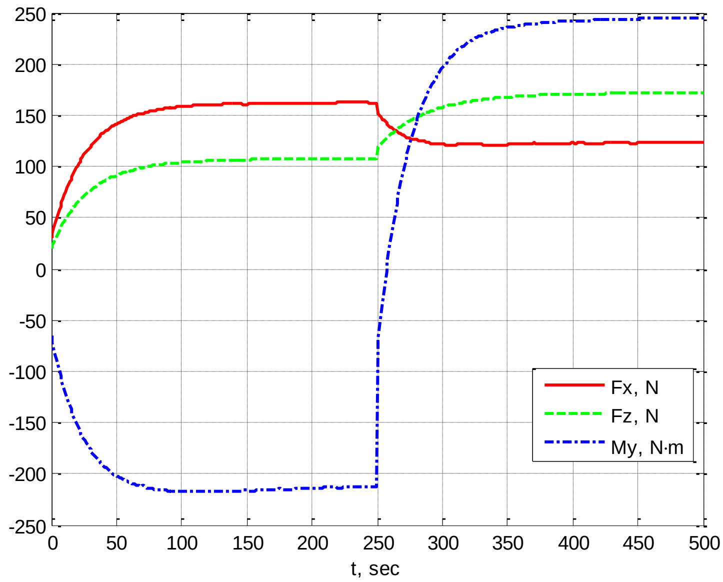

Fig. 6. Control signals (reduced model)

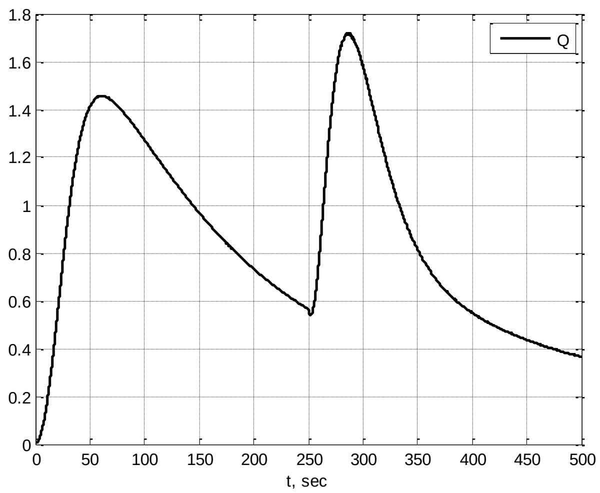

Fig. 7. Performance index (reduced model)

The exact description of hydrodynamic forces and torques is practically impossible. In the nominal model [5] viscous friction was linear with respect to generalized velocities. The effectiveness of the designed NN controller was also proved and confirmed for quadratic (Fig. 8 - 10) function of viscous friction forces (torques).

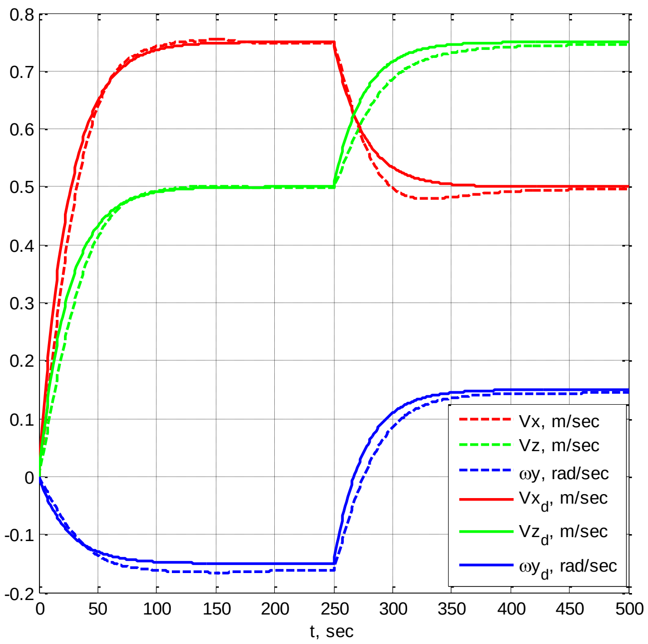

Fig. 8. Transient processes (quadratic viscous friction)

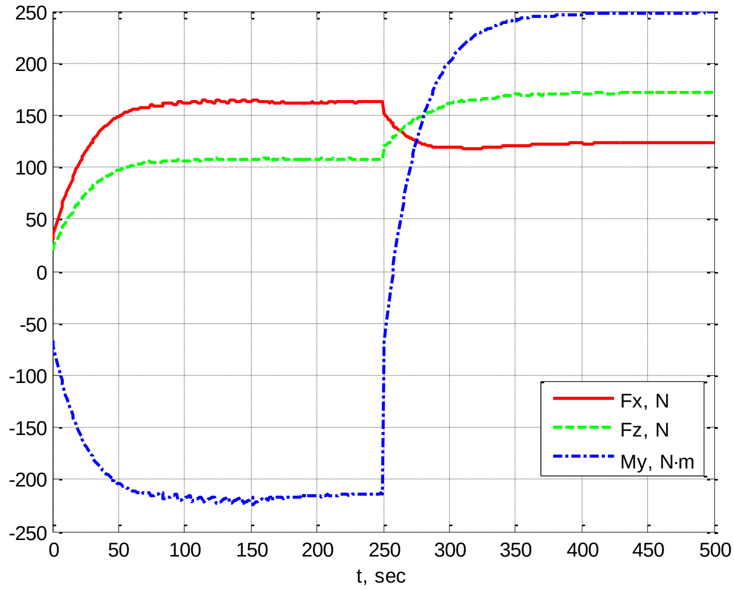

Fig. 9. Control signals (quadratic viscous friction)

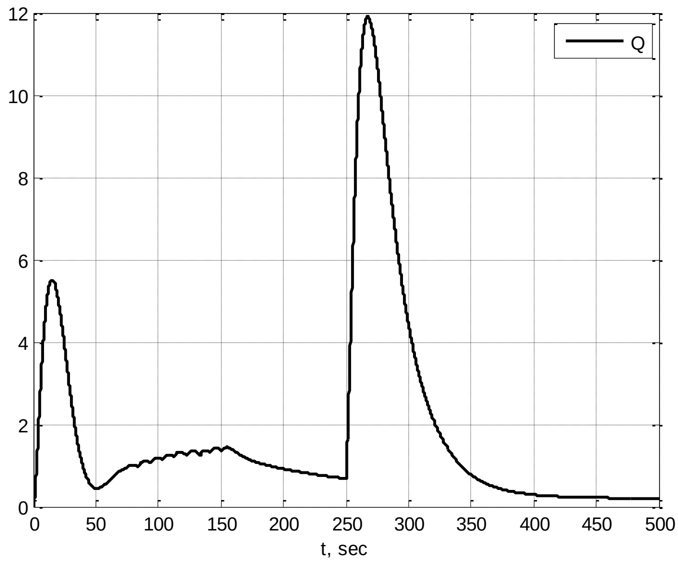

Fig. 10. Performance index (quadratic viscous friction)

Computer experiments had demonstrated control system stability and high quality of transient processes for different situations of parameters and partial structure uncertainties of UR dynamics.

For all considered cases, as seen from simulation results (Figs. 4, 7. 10), performance index (criterion) Q is reducing during transient processes.

5 Conclusion

The approach based on speed gradient method is proposed and applied to design an intelligent NN controller for underwater robot control system and to derive its learning algorithm. The numerical experiments have shown that high quality processes can be achieved with proposed intelligent NN control. The procedure of NN learning makes possible for UR control system to overcome parameter and, partially, structural uncertainties of dynamical object.

References

1. Dyda, A.A. (2007) Adaptive and neural network control for complex dynamical objects. - Vladivostok, Dalnauka. – 149 p. (in Russian).
2. Dyda, A.A., Oskin, D.A. (2004) Neural network control system for underwater robots. // IFAC conference on Control Application in Marine Systems “CAMS-2004”. - Ancona, Italy, 2004., p. 427-432.
3. Dmitry Oskin, Alexander Dyda. Underwater Robot Intelligent Control Based on Multilayer Neural Network. Proceeding of the 7th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS’2013), Berlin, Germany. Page(s): 921 – 924.
4. Fossen, T.I. (2002) Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles. Marine Cybernetics AS, Trodheim, Norway
5. Fradkov A.L. (1990) Adaptive control in large-scale systems.- M.: Nauka., (in Russian).
6. Miroshnik I.V., Nikiforov V.O., Fradkov A.L. Nonlinear and Adaptive Control of Complex Dynamical Systems, St.-Petersburg: Nauka, 2000. – 549p. (Series «Analisis and Design of Nonlinear Systems»), (in Russian).
7. Narendra K.S., Parthasaraty K. (1990) Identification and control of dynamical systems using neural networks // IEEE Identification and Control of Dynamical System, Vol.1. № 1. 20, pp. 1475-1483.
8. Ross A., Fossen T.and Johansen A. (2004) Identification of underwater vehicle hydrodynamic coefficients using free decay tests // Preprints of Int.Conf.CAMS-2004,. Ancona, Italy,2004. - pp.363-368.
9. Sutton,R. and Craven, P.J. (2002) An on-line intelligent multi-input multi-output autopilot design study // Journal of Engineering for the Maritime Environment, vol.216 No.M2, pp.117-131.
10. Yuh Y. (1990) Modelling and control of underwater vehicles IEEE J. of Trans. Syst., Man, Cybern., vol.