Neural Model of Underwater Vehicle Dynamics

Andrzej ŻAK

***Abstract*—**Paper presents method of modeling dynamic of underwater vehicle using technique of artificial neural network. Used neural networks including dynamic model of neuron, method of adaptive calculation of its parameters and method of learning multilayer neural network were described. Algorithm of modeling dynamics of underwater vehicle and example results of research realized in experimental conditions were presented.

***Keywords*—**Artificial Neural Networks, Underwater Vehicles, Dynamics Modeling.

I. Introduction

Modeling is a very general notation and has its customary meaning in common speech. In technics it is related with creating model of researched objects. As a model of object it is understood the presentation of interesting, essential peculiarity of real object in convenient form. The model should behave like real object but it may have different internal structure. So, as modeling it is frequently understood the procedure in which result, basing on input and output signals of object, arise the model which is admitted as the best according to the accepted criterions [2].

In many cases modeling starts from definition of basic physics laws which take part in investigated processes. From these laws arise some equations and if there are known the values of all internal and external conditions of modeled object and the knowledge about physics of happen phenomenon is full then it is possible to calculate all coefficients. Unfortunately such case are very rare, because the knowledge which is required for proper design of the system and indefinableness carried in by environment are not available a priori. In other hand, very often only dependencies between input and output signals are interesting. In this case the object can be treated as "black box" with many or single input and many or single output, but with unknown internal structure.

In last year to modeling of multidimensional object very often are using the methods of artificial neural networks [7, 8]. The huge popularity of this method and rising number of its application is related with ability of neural networks to adaptation and self organization. More over neural networks are synchronous systems what gives possibility to speed up the calculations. Another advantage of neural networks is programming by learning. It cause that there is no need to create complicated mathematical equations, but only general definition of parameters, what make possible to realize the task independently of problem's kind.

II. Modeling of dynamics using neural networks

A. Description of modeling process

Problem of modeling and object's dynamic identification are the basic problem in many methods of synthesis of control systems. The evolution of neural networks theory and technological possibilities of its practical realization create in last year new, effective and universal tools used for modeling. The general structure of system which is realizing the task of synthesis of neural model is shown on figure 1. The presented situation is adequate to the process of neural network learning, it means to calculate the values of its coefficients.

Information about error which is the difference between object output and neural model output, is the input information in learning algorithm. Because in control systems processes have dynamic character, neural modeling need to usage special solutions. In last investigations the dynamical character of neural networks is receiving by introducing dynamics of process into neuron in such way that activity of neuron depends of its internal state.

B. Synthesis of neural modeling

Properties of neural networks and its possibilities depend on both architecture of connections between neurons and type of neurons. Putting into the static mathematical model of neuron

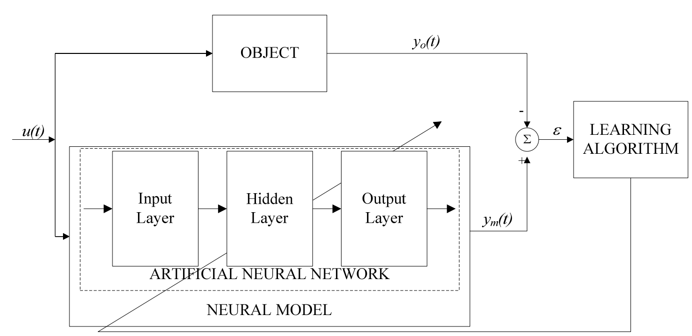

Fig. 1 Synthesis of neural model

various kind of reversible conjugations it is possible to create dynamical model of neuron. The dynamics is putting into neuron in such way that its activity depends on its internal states. It is realized by putting into the structure of linear neuron the dynamic system [1]. Using this system neuron is reconstructing past values of signals from two sets of signals: the inputs signals $x_i(k)$, for $i = 1, 2, \dots, N$ and output signal

Manuscript Received December 21, 2006, Revised Received April 9, 2007

$y(k)$ for current and past times (fig. 2).

In dynamic model of neuron three blocks can be pointed:

- adder of weighted input signals;
- dynamic linear system;
- nonlinear activation block.

In the block of adder the sum of weighted input signals is calculated according to the following formula [1]:

$$\varphi(k) = \sum_{i=1}^{N} w_i(k) x_i(k) \tag{1}$$

where:

$w_i(k)$ – weight of i -th input;

$x_i(k)$ – i -th input signal;

N – number of components of input signal;

k – index of discreet time.

Calculated weighted sum is processed in the block of dynamic linear system, which can be filter of any order. Structure of such system contains delay elements, feed-back conjugate, feed-forward connections and adequately suited weights.

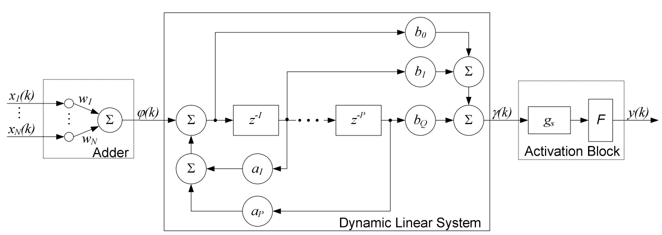

Fig. 2 Structure of dynamic model of neuron with N inputs and one output

This structure can be described by following differentia equation [1]:

$$\begin{aligned} \gamma(k) &= -a_1\gamma(k-1) - \ldots - a_P\gamma(k-P) + \\ &+ b_0\varphi(k) + b_1\varphi(k-1) + \ldots + b_Q\varphi(k-Q) \end{aligned} \tag{2}$$

where:

$\varphi(k)$ – input of filter block in time k;

$\gamma(k)$ – output of filter in time k;

$a = [a_1, \ldots, a_P], \quad b = [b_0, \ldots, b_Q]$ – weight's vector of feed-back and feed-forward connections;

P, Q – constant values.

According to the structure of dynamic neuron model output signal of dynamic linear system is the input signal for activation block. Out-put signal of neuron which is output signal of activation block is calculated using equation:

$$y(k) = F\left(g_s, \gamma(k)\right) \tag{3}$$

where:

$F(\cdot)$ – nonlinear activation function;

g_s – coefficient of slope of activation function.

The aim of the learning algorithm is to calculate the values of parameters of dynamical neuron (values of input weights, values of co-efficient of dynamical linear system, value of activation function slope), basis on the set of pair of input and output patterns. Its calculation can be done by solving the optimization problem in which the output error is described by equation:

$$e(k) = y^d(k) - y(k) \tag{4}$$

where:

$y^d(k)$ – desired output of system;

$y(k)$ – current output of system;

it should be minimalized the criterion J which has the following form:

$$J = \frac{1}{2} E\{e(k)^2\} \tag{5}$$

where:

E – operator of expected value.

To solve such formulated optimization problem and to calculate the optimal values of neuron's parameter, the method of the biggest de-crease gradient can be used [3, 4, 5].

Let $\upsilon = \upsilon(a, b, w, g_s)$ be the generalized parameter of network, $\frac{dJ}{d\upsilon}$ be derivative of J relative to υ and η be the coefficient of learning, so general equation for calculation the value of neural model parameter has following form:

$$\upsilon(k+1) = \upsilon(k) - \eta \nabla_\upsilon J\big|_{\upsilon=\upsilon_k} \tag{6}$$

In results of simple transformations can be written the equation:

$$\frac{dJ}{d\upsilon} = E\{-e(k)F'(g_s\gamma(k))S_\upsilon(k)\} \tag{7}$$

where:

$S_\upsilon(k) = \frac{\partial \gamma(k)}{\partial \upsilon}$ – sensitive vector of signal $\gamma(k)$ for change of parameter υ;

$F'(\cdot)$ – derivative of neuron's activation function.

At last equation (6) can be written as follow:

$$\upsilon(k+1) = \upsilon(k) + \eta E\{e(k)F'(g_s\gamma(k))S_\upsilon(k)\} \qquad (8)$$

Because for ergodic process of discrete time the expected value is average value so equation (8) can be presented in form:

$$\upsilon(k+1) = \upsilon(k) + \eta\sum_{i=1}^{N} e(i)F'(g_s\gamma(i))S_\upsilon(i) \qquad (9)$$

According to the assumption of neural network's learning algorithm [4] modification of values of neural model are made after presentation of all learning patterns. In practice its modification is made after presentation of every learning vector, so equation (9) can be simplify to the following form:

$$\upsilon(k+1) = \upsilon(k) + \eta e(k)F'(g_s\gamma(k))S_\upsilon(k) \qquad (10)$$

The equations which are described the components of sensitive vector of generalized parameter υ can be written as follows:

- sensitive coefficient for values of neurons input weights:

$$S_{w_i}(k) = g_s\frac{\partial\gamma(k)}{\partial w_i} = g_s\left(\sum_{q=0}^{Q} b_q x_i(k-q) - \sum_{p=1}^{P} a_p S_{w_i}(k-p)\right) \qquad (11)$$

- sensitive coefficient for parameters of dynamic linear system:

$$S_{a_p}(k) = g_s\frac{\partial\gamma(k)}{\partial a_p} = -g_s\gamma(k-p) \quad \textit{for} \quad p = 1,\ldots,P \qquad (12)$$

$$S_{b_q}(k) = g_s\frac{\partial\gamma(k)}{\partial b_q} = -g_s\varphi(k-q) \quad \textit{for} \quad q = 0,1,\ldots,Q \qquad (13)$$

- sensitive coefficient for slope of neuron's activation function:

$$S_{g_s}(k) = g_s\frac{\partial\gamma(k)}{\partial g_s} = \gamma(k) \qquad (14)$$

In practice the elementary operations executed by single dynamic neuron isn't very interesting because the neural calculation power arise form connections of many neurons into one network. Often the structure of dynamic neural network is similar to the structure of static feed-forward multilayer network. Such structure doesn't include any feed-back conjugations, which are complicating both the architecture of neural network and equations of adaptive learning algorithm.

In case of multilayer dynamical neural network the equations of single neuron can be expanded for whole network. The error calculated on output of network is propagated back through hidden layers onto the input layer, similarly how it is made in algorithm of back-propagation [3, 4, 5].

Let M means number of layers, s_m means number of neurons in m-th layer, $y_i^m(k)$ means output of i-th neuron which is positioned in m-th layer in time step k ($m = 0,1,\ldots,M$, $i = 0,1,\ldots,s_m$). The function which is described i-th neuron in m-th layer is defined as follows [6]:

$$\begin{aligned} y_i^m(k) &= F(g_{si}^{\ m}\gamma_i^m(k)) = \\ &= F(g_{si}^{\ m}[b_{0i}^{\ m}\varphi_i^m(k) + b_{1i}^{\ m}\varphi_i^m(k-1) + \ldots + b_{ni}^{\ m}\varphi_i^m(k-n) + \\ &\quad - a_{1i}^{\ m}\gamma_i^m(k-1) - \ldots - a_{ni}^{\ m}\gamma_i^m(k-n)]) \end{aligned} \qquad (15)$$

and the general error generated by this neuron is described by equation [6]:

$$\begin{aligned} \delta_i^m(k) &= -\frac{J(k)}{\partial\gamma_i^m(k)} = -\frac{J(k)}{\partial x_i^m(k)}\frac{\partial x_i^m(k)}{\partial g_{si}^{\ m}\gamma_i^m(k)} = \\ &= -\frac{J(k)}{\partial x_i^m(k)}F'(g_{si}^{\ m}\gamma_i^m(k)) \end{aligned} \qquad (16)$$

Its first part for output layer can be written as follow:

$$\frac{J(k)}{\partial x_i^M(k)} = \frac{\partial J(k)}{\partial y_i(k)} = -(y_i^d(k) - y_i(k)) = -e(k) \qquad (17)$$

But for hidden layers it can be determined as follow:

$$\begin{aligned} \frac{J(k)}{\partial x_i^m(k)} &= \sum_{j=1}^{s_{m+1}}\frac{\partial J(k)}{\partial g_{sj}^{\ m+1}\gamma_j^{m+1}(k)}\frac{\partial g_{sj}^{\ m+1}\gamma_j^{m+1}(k)}{\partial x_j^m(k)} = \\ &= \sum_{j=1}^{s_{m+1}}\frac{\partial J(k)}{\partial g_{sj}^{\ m+1}\gamma_j^{m+1}(k)}g_{sj}^{\ m+1}b_{0j}^{\ m+1}w_{ij}^{\ m+1} = \\ &= \sum_{j=1}^{s_{m+1}} -\delta_j^{m+1}(k)g_{sj}^{\ m+1}b_{0j}^{\ m+1}w_{ij}^{\ m+1} \end{aligned} \qquad (18)$$

From the above follows that general error generated by neuron can be written as follow:

- for output layer:

$$\delta_i^M(k) = e(k)F'(\gamma_i^M(k)) \qquad (19)$$

- for hidden layers:

$$\delta_i^m(k) = \sum_{j=1}^{s_{m+1}}(\delta_j^{m+1}(k)g_{sj}^{\ m+1}b_{0j}^{\ m+1}w_{ij}^{m+1})F'(\gamma_j^m(k)) \qquad (20)$$

Hence change of parameters of i-th neuron in m-th layer in general notation can be defined as follow [6]:

$$\upsilon_i^m(k+1) = \upsilon_i^m(k) + \eta\delta_i^m(k)S_{\upsilon i}^{\ m}(k) \qquad (21)$$

where:

$S_{\upsilon i}^{m}(k)$ – sensitive vector for change of parameter υ which can be determined using equations (11) – (14).

III. Experimental investigations with real object

A. Conditions of research

The experimental investigations were made for remotely operated underwater vehicle „Ukwial" in the Navy Harbour in Gdynia [6]. Mine destroyer was the base for underwater vehicle. During researches the hydrological and meteorological conditions were good.

The vector of input signals (control vector) has the following form:

$$u_s = [\tau_X, \tau_Z, \tau_N] \tag{22}$$

where:

τ_X – the force of thrust along longitudinal axis of vehicle;

τ_Z – the force of thrust along normal axis of vehicle;

τ_N – the rotation moment around the normal axis of vehicle.

The vector of output signals (state vector) has the form:

$$x_e = [d, \psi] \tag{23}$$

where:

d – the depth of immersion of underwater vehicle measured with precision of 0.1 m;

ψ – the angle of course of underwater vehicle measured with precision of 0.5^{o}.

B. The neural algorithm of modeling dynamics of underwater vehicle

Basis on equations presented in previous chapters, the unit which is using dynamical neural network for modeling dynamics of underwater vehicle was worked out. The algorithm of its working was presented on figure 3 [6].

In its first phase become the initialization of artificial neural network which consist of following actions:

- generating of dynamic neurons which structure is described by equation (4);
- setup of beginning values of every neuron's parameters, it means: values of input weights, coefficients of dynamic linear system and angle of slope of activation function;
- creating connections between neurons by defining for every neurons its successors.

As an activation function for every neurons the tangensoidal function was accepted.

The number of neurons in input layer depends on size of control vector, and in examined case this layer consists of three neurons. Similarly the number of neurons in output layer depends on size of state vector and during the investigation with real object it consists of two neurons. The number of neurons in hidden layers was choose basing on Vapnik-Chervonenkis theorem [4].

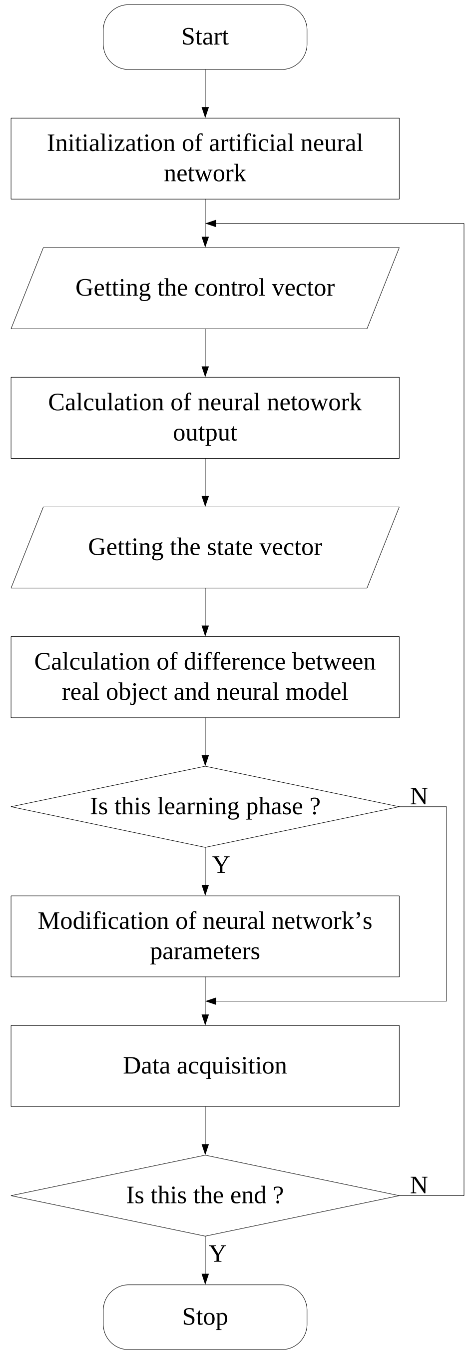

Fig. 3 Block diagram of algorithm of modeling dynamics of underwater vehicle

Neurons in network were connected each other to each other between two neighborhood layer, without connections inside layer, re-current connections and connections which omit any layer. The structure of neural network used during research is presented in figure 4.

The second phase of algorithm starts from getting the control vector, its normalization, and next calculation of output of neural network according to equations (15). Basis on neural network output and measurement of component's values of modeled object's state vector, the error of neural network output is calculated. Next step was modification of values of every neuron's parameters. Due to the errors from output layer are propagated onto previous layers according to equations (19) and (20) a next the new values of neurons parameters are calculated according to equation (21). The output values of neural network were saved for future analysis. The process of modeling was finished after reaching the fixed value of average square error which was the indicator of adjustment of neural model to real object. In case of maladjustment algorithm is going back to the instruction of getting the control vector.

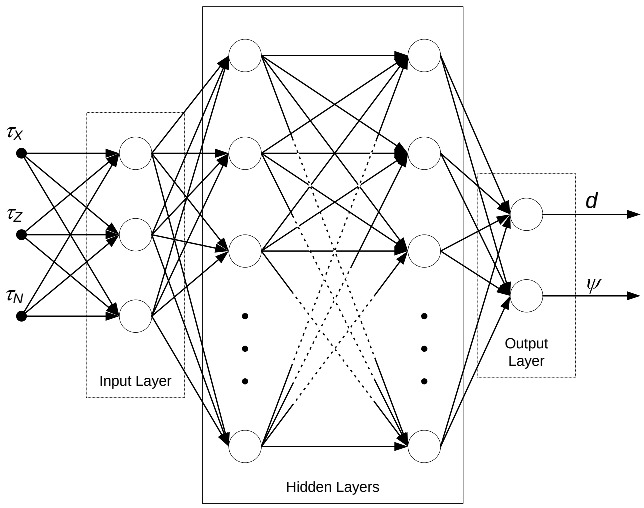

Fig. 4 The structure of neural network used during experimental investigation

C. Results of research

In the first phase of experimental research using real object the neural model was determined. In this phase model works as prediction system because its parameters were changed due to presented above algorithm. This phase was realized during execute the task of moving underwater vehicle into the specified area of operation it means during movement from point to point. The results of this phase were presented on figures 5 to 7.

In the next phase the modeling system becomes into simulation mode, so the calculated parameters of neural model didn't change. In this phase the underwater vehicle was executing the task of approach to the underwater construction. The results of this phase were presented on figures 8 to 10. The small value of difference between neural model output and real object output means about good quality of adjustment of neural model to the real object.

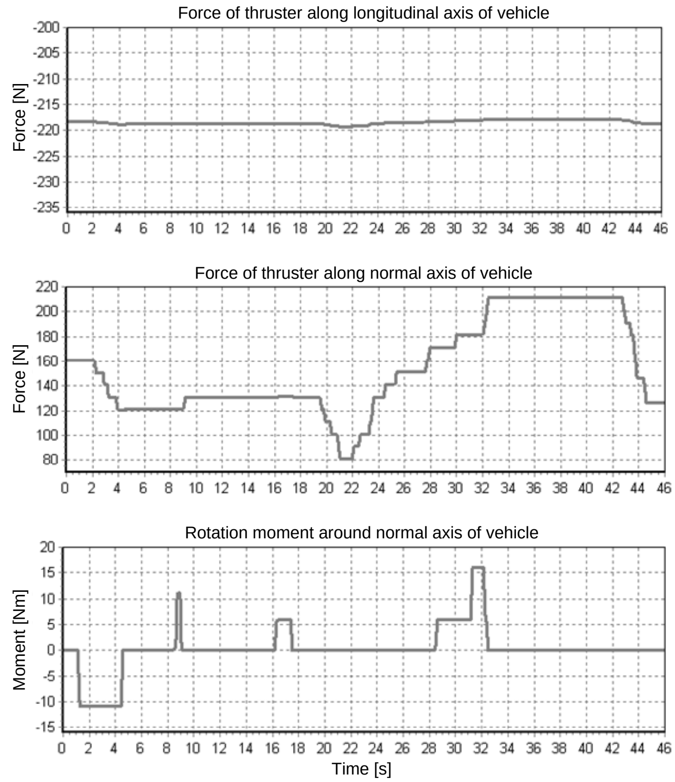

Fig. 5 Plot of components of control vector of underwater vehicle during execute the task of moving into the specified area

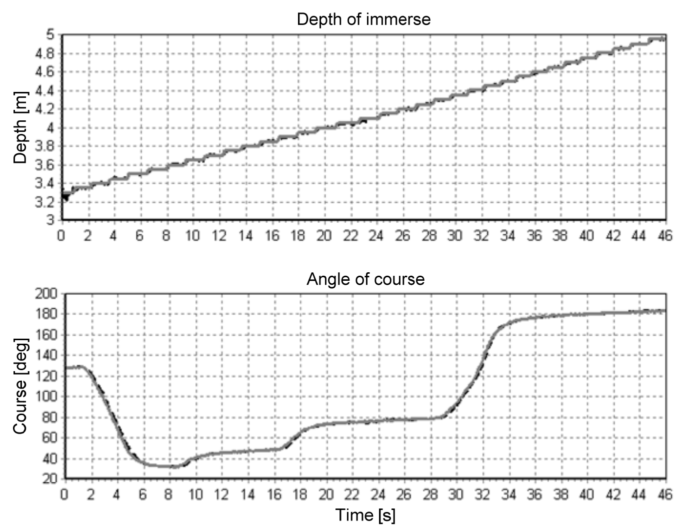

Fig. 6 Plot of components of state vector of underwater vehicle during execute the task of moving into the specified area; black doted line means neural model, gray solid line means object

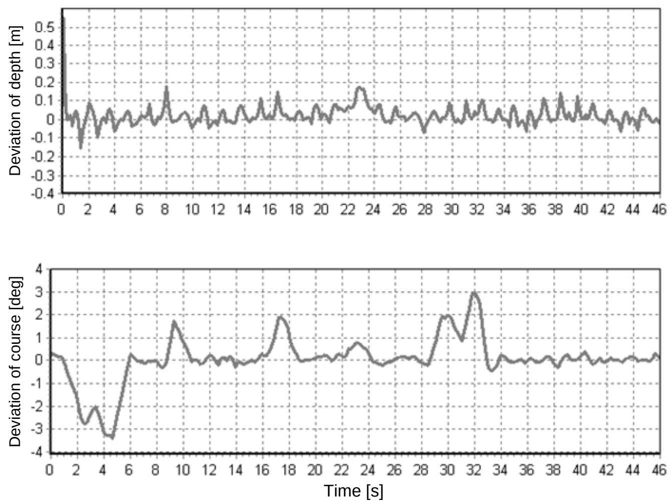

Fig. 7 Plot of difference between output signals of real object and neural model during execute the task of moving into the specified area

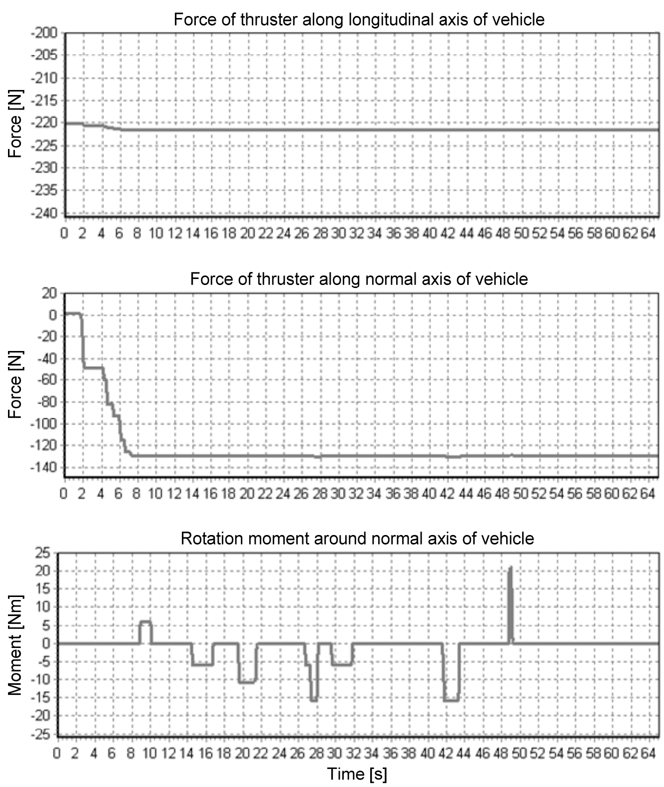

Fig. 8 Plot of components of control vector of underwater vehicle during execute the task of approach to the underwater construction

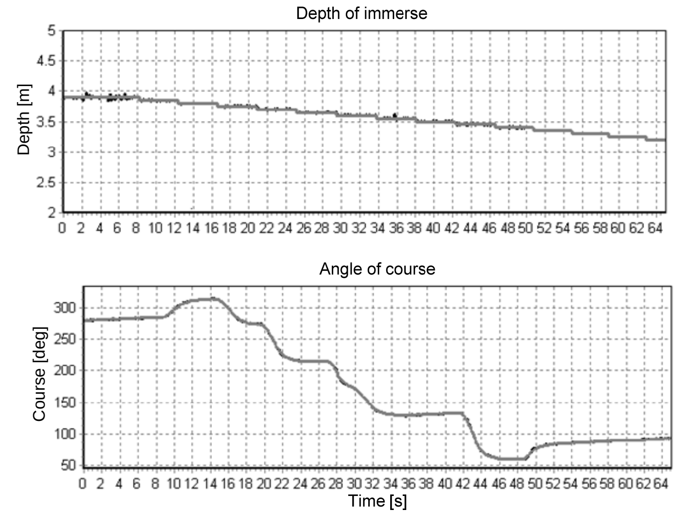

Fig. 9 Plot of components of state vector of underwater vehicle during execute the task of approach to the underwater construction; black doted line means neural model, gray solid line means object

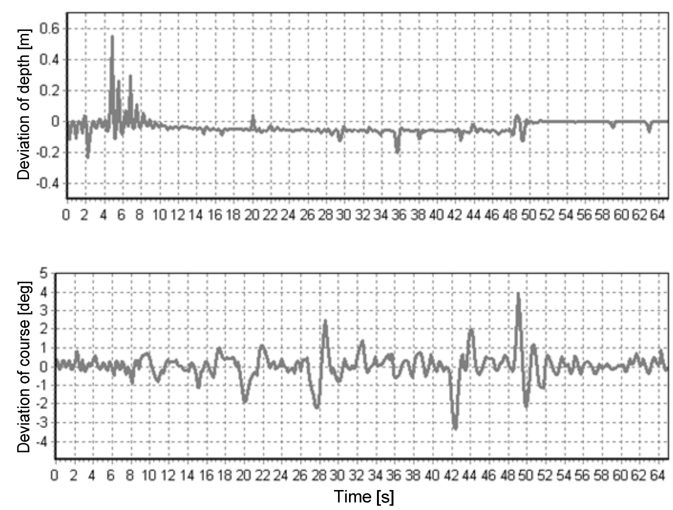

Fig. 10 Plot of difference between output signals of real object and neural model during execute the task of approach to the underwater construction

IV. Summary

The main aim of the research was to create the neural model of underwater vehicle. During investigation the few problems related with selection of architecture of dynamical neural network which was realizing the synthesis of multidimensional object, the limited time of learning neural network, the measurements of control vector and state vector of underwater vehicle were considered. Theirs solutions allows to made the experimental research and determine the neural model of investigated underwater vehicle.

Basis on acquired results of research it can said that modeling using neural network allows for dynamic's identification in real time and using dynamical neural network characterize the small error of modeling.

Described at the paper method of determined the model of remotely operated underwater vehicle can be used both in process of synthesis of control system and for creating the simulators of underwater vehicle movement.

Future research should be focused on method of choosing neural network architecture, another method of optimization in learning algorithm. It will be very interesting to equipping the underwater vehicle with other measurement system (which can measure for example declination, speed etc.) to expand the state vector. In such configuration probably using above described method it is possible to create redundant virtual measurement system.

This method can be used for modeling other multidimensional, dynamical objects.

REFERENCES

[1] Back A. D., Tsoi A. C. "FIR and IIR synapses. A new neural network architecture for time series modeling", *Neural Computation,* vol. 3, pp. 375-385, 1991.

[2] J. Gutenbaum, „Mathematical Modeling of Systems", Publishing House Exit, Warsaw, 2003.

[3] R. A. Koniński, „Artificial Neural Networks. Nonlinear Dynamics and chaos", Scientific and Technical Publishing House, Warsaw, 2002.

[4] S. Ossowski „Neural Networks", Publishing House of Warsaw Technical University, Warsaw, 1996.

[5] R. Tadeusiewicz, W. Duch, J. Korbicz, L. Rutkowski, „Neural Networks", Publishing House Exit, Warsaw, 2004.

[6] A. Żak, „Identyfication of underwater vehicle's dynamic in exploitation conditions", PhD thesis, Polish Naval Academy, Gdynia, 2006.

[7] A. Żak „Dynamic's modeling of underwater vehicles", *Proc. of 15th Domestic Conference on Automation*, Warsaw, pp. 295-298, 2005.

[8] A. Żak, "Neural Algorithm of Underwater Vehicle's Dynamics Identification", *WSEAS Transactions on Computers Research* Vol. 1 No. 2, pp. 89-94, 2006.