The Use of Neural Networks in Systems with Complementary Correction of the Control Action

Alexander Kobzev, Anastasia Lekareva & Oksana Sidorova

Abstract

The paper deals with systems with complementary correction of the control action with non-deterministic disturbing action using a neural network controller. The concept of complementary control is given. Two structures of control systems with complementary correction of the control action are presented. The neural network controller generates an additional component in the control action in two versions: 1) an autonomous additional component to the main control; 2) correction of the control action at the stage of its generation. An algorithm for training a neural network is provided. The input signals of the neural controller are shown, as the error function from the off-design disturbance. An error signal is used between the output coordinate of the reference model and the control object in order to adjust the weights of the neural network. The main parameters that determine the structure of the neural network are shown: the number of layers, neurons inside, activation functions, etc. When simulating, we used the Simulink package of the MatLab program with S-function procedures. The results of the study of neural controller characteristics for driving a robotic system are presented. The tasks of further research are formulated.

Keywords: complementary correction; neural network; compensation of external off-design disturbances; robotic system.

1. Introduction

The problem of constructing a control system that provides the required quality of a dynamic object operation, characterized by non-stationary internal parameters and operating in conditions of nondeterministic external disturbances, is one of the main problems in the contemporary control theory. Practical implementation of the designed control algorithms of dynamic systems is provided by simplicity of their implementation and understanding, as well as flexibility, stability, adaptive properties, the ability to learn.

The traditional approach to the design of adaptive automatic control systems (ACS) based on gradient methods, systems with a model, random search methods, etc., does not always solve the problem of parrying this type of disturbances effectively. This is due to a number of factors, including the difficulty of selecting a simple and reliable adaptation algorithm that works in the case of a wide range of changes in the parameters of the object and external disturbances. If the control object belongs to the category of complex dynamic objects, i.e. it is multidimensional, described by high-order differential equations, has essentially nonlinear characteristics, etc., the choice of the adaptation
algorithm is much more complicated. In addition, the practical acceptability of mathematical models underlying the construction of algorithms should be provided by low sensitivity to changes in parameters, which is quite difficult to implement for nonlinear systems. In recent years, artificial intelligence methods are increasingly being applied for control purposes [1-3].

Methods of neural network control are intensively being designed by Russian (A. I. Galushkin, V. A. Terekhov, A. N. Gorban, I. V. Komashinskii, D. A. Smirnov, T. A. Bondar, A. S. Logovskii, etc.) and foreign scientists (K. Narendra, O. Sigeru, S. Haykin, M. Gupta, S. Osowski, etc.). The literature describes numerous examples of practical application of neural networks to solve problems by various dynamic objects: aircraft [3], automobile [3], mining process [4], rotation speed of the engine shaft [5], electric furnace [6], turbo generator [7], welding machine [8], pneumatic cylinder [9], etc.

Literature analysis reveals that the existing variants of neural network controllers as part of the subordinate control systems can be summarized in three options: 1) creation of a neural network that imitates a linear regulator; 2) synthesis of the optimal nonlinear neural network controller; 3) creation of a nonlinear neural network controller, learning and working in real time in accordance with a given dynamics.

In general, the analysis of the available results in the considered subject points at complexity of designs of almost all schemes of neurocontrol because of several neural networks and nontrivial sequence of their training procedures. Besides, the need for pre-training neural network (NN) limits the application of the designed controller by a specific dynamic object, and also limits the range of variation of the object parameters and disturbances, in which the controller provides the required quality indicators. However, it should be noted that many issues, primarily related to the practical use of NN in ACS, have not been fully investigated yet. Most of the works have either abstractive character, when the main attention is paid to the proof of the principal possibility of using a neural network of some kind in a certain subject area, or they are devoted to solving relatively narrow applied problems and do not contain generalizations that allow to use the results to solve similar problems. In particular, there are no works devoted to the problem of replacing classical regulators with neural network ones, generalizing the actual experience. The list of works on neural network control of a non-stationary object is rather limited. As a rule, only the case of a sufficiently smooth continuous change in properties of the control object is considered there. Summarizing the above, we can conclude of the prospects of research areas for the construction of a single control system operating in online-training mode.

2. Systems with complementary correction and neural network controllers

With the development of new technologies and production, requirements for ACS increase in accuracy and speed, especially in conditions of uncertain external disturbance and drift of its own parameters. This is true for mobile technologic robotics robots (MTRR) performing manufacturing operations to facilities located generally in an indeterminate space. The MTRR consists of a transport system on which a technologic robot is installed, performing welding operations, material handling when clearing of territories, etc. Let us consider the characteristic structure by the example of MTRR hydraulic cutting of oil pipelines [10]. The MTRR flowchart is shown in Figure 1. The transport system is assigned coordinates of the exit point G to the cutting position, and the technologic robot is assigned coordinates of the control program Xp, which provides reproduction of a given profile Φ of the cut pipe surface by the means of a high pressure jet with an abrasive according to the technological parameters T. There is a discrepancy between the systems of actual coordinates and the axes direction of the cut object X1 and the technologic robot X0 as a result of imprecision of exit to the starting point of cutting, ground profile and orientation TR to the cut object. Besides, there are deviations of the ideal G0 (programmed) and the actual G1 cut surfaces due to deformation, dents, welds, dirt and other factors. It is necessary to compensate the deviation of coordinate systems and the direction of axes and shape of the actual and programmed cut surfaces

\[ ΔX = X_1 - X_0; \ ΔG = G_1 - G_0. \]  

(1)

The equivalent disturbances are shown in the diagram: \( F_1 \) – from the side of the ground profile when moving the transport system; \( F_2, F_3 \) – generating deviations \( ΔX, ΔG, Xg^* \) – total value of coordinates; \( T \) – technological modes.

To parry disturbances to the transport system and the technologic robot, the methods of optimal and adaptive control with different interpretations of quality functions and control and correction algorithms are used. To solve these problems, we consider adaptive systems with complementary correction, providing compensation of additional, off-design disturbance by introducing an additional component into the controlling action. This control is based on the modified fourth form of invariance [11], [12]. Let the equation for the system error \( δ(p) \), which is attached with control input \( g(p) \) and disturbance \( f(p) \), has the form:

\[ F(p)δ(p) = M(p)g(p) + B(p)f(p). \]  

(2)

\( F(p), M(p), B(p) \) are corresponding operator. Then the classical treatment of the fourth form of invariance is:

\[ M(p) ≠ 0, g(p) ≠ 0, M(p)g(p) – M_i(p)g_i(p) = 0; \]

\[ B(p) ≠ 0, f(p) ≠ 0, B(p)f(p) – B_i(p)f_i(p) = 0. \]  

(3)
As the equation (1), (2), (3) shows, the fourth form is associated with artificial introduction of a special source of compensation signal into the system and generation of its channel, while generation of its compensation channel is implemented for each action. Modification of the fourth form of invariance is based on the relationship between the controlling and disturbing actions in robotic systems, as well as errors caused by disturbing actions, on the controlling action of the system. It is assumed to compensate for the external undefined disturbance by introducing an additional component only in the controlling action. Invariance condition of the fourth modified form with additional immeasurable disturbance in ACS has the form:

$$B(p)\Delta f(p) - M_2(p)\Delta g(p) = 0.\quad (4)$$

Invariance condition of the modified form (4) can be implemented using the methods of adaptive control [13]. Generation of corrective amendments for the control law is performed in the error function proportional to the additional undefined disturbance. There are two ways to implement additional control. The first method involves introduction of an autonomous additional component $\Delta g(p)$ to the generated control $g(p)$. The second method provides correction of the polynomial $M(p)$ at the stage of generating the control for the actuators, transforming it into $M^*(p)$. This approach is used in ACS with predictive control [13].

The analysis of ACS structures with complementary correction allows us to propose two diagrams for inclusion of a neural network controller in ACS loop (Fig. 2) [14], [15]. The diagrams indicate: CSSD – control signal synthesis device; CD – control device; CO – control object; RM – reference model with given dynamic parameters; COM – control object model regarding the error by control action; COMP – control object model regarding the error by disturbance. COM and COMP are formed by static dependencies of the first and second error coefficients from the corresponding effect; NN – neural network; TA – training algorithm, $g_0$, $f$ – control and disturbance actions respectively; $X$, $X_M$ – the control coordinate and the output coordinate of the reference model, respectively; $\delta$ – total system error; $\Delta f$ – additional off-design disturbance; $\delta_0$, $\delta g$ – components of the system error caused by disturbance and driving effects; $\delta_0$, $\delta g$ – error component proportional to the additional disturbance; $\Delta g$ – additional component of the control action; $\varepsilon$ – error between the output coordinate of RM and CO.

The input of the neural network is an error proportional to the external indeterminate disturbance. It is distinguished from the total error of the system by eliminating the deterministic components of the error obtained by analysis of COM and COMP outputs. The structures in question pre-suppose the online training of NN in the process of control. As an error signal is used to adjust NN weight matrices, we propose to use the error signal between the output coordinate of RM and CO $\varepsilon$. The purpose of the control is to minimize the error in the system loop. Here, the corrections to the control law, generated in the system correction loop, are assumed to be unknown, i.e. we assume the absence of a reference value for NN output.
The input of the neural network is an error proportional to the external indeterminate disturbance. It is distinguished from the total error of the system by eliminating the deterministic components of the error obtained by analysis of COM and COMP outputs. The structures in question pre-suppose the online training of NN in the process of control. As an error signal is used to adjust NN weight matrices, we propose to use the error signal between the output coordinate of RM and CO $\varepsilon$. The purpose of the control is to minimize the error in the system loop. Here, the corrections to the control law, generated in the system correction loop, are assumed to be unknown, i.e. we assume the absence of a reference value for NN output.

3. Selection of the structure and algorithm for neural network

There are people who are ready and want to help you to present results of your research in the best possible way to the international scientific community. Those nice people are our reviewers. They will review your paper and will suggest improvements. The main target of their work is to help you make your paper better. Their suggestions are following some basic ideas and questions. We are giving here in chaotic order some of their typical ideas, advices and questions. Please take care about these facts, and your ideas, results and paper will be much better presented to the international public. General approaches to neural networks design are shown in the works [16], [17]. The concepts of the works [18], [19] give grounds for the possibility of a sufficiently accurate approximation of any function of many variables by a two-layer structure of NN, provided that a nonlinear activation function is used in the hidden layer. The results of these studies allow us to conclude that it is appropriate to consider two-layer NN for solving the problem: the input layer - the hidden layer - the output layer. The logarithmic function is used as the activation function for the hidden NN layer. The use of this activation function is appropriate due to the assumption of the unknown boundaries of the input and output signals and the complexity of their normalization. A linear activation function is used in the output layer. In general, an overview of the methods of selecting the number of neurons in the hidden layer is given in [17].

In [20] R. Hecht-Nielsen, referring to the Kolmogorov–Arnold theorem, showed that there is a sufficient number of $(2N + 1)$ neurons in the hidden layer, where $N$ is the number of NN inputs. However, in [21] it was proved that the number of neurons of the hidden layer must satisfy the following inequality: $N_{hid} \geq 2N + 1$. This approach can be effectively applied as a lower estimate of a number of neurons in the hidden layer. Studies of the correction circuit with a neural network controller based on two types of NN: a recurrent network and a network of back propagation of error, indicates the feasibility of design a controller using recurrent networks that provide a better rate of convergence of the disturbance compensation process, as well as lower error values. The choice of discreteness $\Delta t$ is essential, it can be considered as an adjustable parameter [16]. This complicates the adjusting algorithm. It would be safe to assume that $\Delta t$ can be computed as the ratio of the transition time of the internal circuit of the system to the number of neurons in the hidden layer of NN [22].

In most of the works devoted to the control of robotic systems and various technological processes, multi-layer neural networks of direct distribution are used to solve control problems. It is explained by the following factors [23]. Such networks are structures with direct connections in which information is transmitted from inputs to outputs, which is especially convenient for working with systems represented as blocks in which the signal is transmitted sequentially in one direction. The main algorithm for training such networks is the algorithm of back propagation of error, which belongs to a wide class of gradient methods often used in the theory of optimal control.

The algorithm of back propagation of error (method of moments) with regularization [16] is considered as a training algorithm in this paper, due to simplicity of its implementation and low computing cost. It is proposed to use a variable parameter of training rate for each layer of NN in the function of a neuron error of the corresponding layer. The training algorithm is a set of the following actions [14], [15], [24]:

1. Initialize weights $W$ (by small random values), select the initial $\eta_0$ and maximum $\eta_{max}$ values for the training rate, control error $\delta$.
2. Initialize $\Delta W$ by filling it with zeroes.
3. Calculate error $E(n)$.
4. If the result satisfies $(|E(n)| < \delta)$, then training is not implemented, otherwise go to step 5.
5. Calculate the loss function gradient $\nabla E(n)$ on the current iteration.
6. Determine the training rate for $i$-th layer according to the equation

$$\eta[i] = \eta_{\text{max}}(1 + \exp(-|G[i]|)),$$  \hspace{1cm} (5)$$

where $\eta_{\text{max}}$ – maximum value for the training rate; means error value for $i$-th network layer

$$G[i] = \sum_{j=i+1}^{l} x_{(i+1)}^{j} / l;$$  \hspace{1cm} (6)

$l$ – number of neurons in the $j$-th next (regarding $i$-th) layer; $x_{(i+1)}^{j}$ – network error matrix, $\text{dim}(x_{(i+1)}^{j}) = [m_{\text{max}}, d]$, where $m_{\text{max}}$ – maximum number of neurons among all NN layers; $d$ – number of NN layers. In this case, we consider the inputs of the network as the first layer.
7. Calculate the change in parameters:

$$\Delta W(n) = \eta(K E(n) + \rho W(n - 1) + \mu \Delta W(n - 1)).$$  \hspace{1cm} (7)$$
8. Network weight correction in accordance with (5), (6), (7).
9. Go to step 3.

4. Modeling and experimental results

Development and simulating of the algorithm has its own specifics, which has characteristics in the implementation of complementary control. Today, many simulation programs have been developed for neural networks implementation. For example, Artificial Intelligence Recurrent Asymmetric Networks (NARIA), Genesis, Tradecision, University of Hertfordshire, Neural Network Software, Neural Network Toolbox for MatLab, etc. The most popular for design and study of automatic control systems for dynamic objects is MatLab program, but this package has some limitations – it does not have a neural network unit with ability to implement online training. However, MatLab provides the ability to create custom blocks based on Simulink function. With the help of programming languages, the user can create a description of an arbitrarily complex block and connect it to the Simulink-model, while in terms of user interaction with the model, the block based on $S$-function is no different from the standard library block Simulink. The created blocks can be continuous, discrete, or hybrid. $S$-functions created in C, C++, Ada, or Fortran are compiled into executables (*.dll), thereby providing increased speed of implementation of such blocks. $S$-functions also have additional features that include working with different types of data (integers, real and complex numbers of varying degrees of accuracy), the use of matrices as input and output variables, as well as a larger set of internal functions (callback-methods). $S$-function is connected to the Simulink model using Level-1 and Level-2 Matlab $S$-function library blocks. In this case, the Simulink block is described by a set of input variables $u$, state variables $x$ and output variables $y$. In mathematical form the block can be described in general by the following system of equations:

$$\begin{cases} y = f_0(t, x, u); \\
\dot{x}_c = f_1(t, x, u); \\
x_{d_{k+1}} = f_u(t, x, u); 
\end{cases}$$  \hspace{1cm} (8)$$

where $y$ - outputs; $\dot{x}_c$ - derivatives of continuous state variables; $x = x_c + x_d; x_{d_{k+1}}$ - discrete state variables.

The model calculation process is performed by Simulink in several stages. The first phase is the initialization of a model: assembling a model into library blocks, definition signal of different dimensions, data types, parameters of model time, estimation of parameters of blocks, and determining the order of implementation of blocks and memory allocating for calculation. The following steps are combined to form the body of the simulation cycle. At each cycle of simulation (time step) calculation of blocks is performed in the order determined in the process of initialization. For each block, Simulink calls functions that calculate the state variables of block $x$, the derivatives of state variables, and the outputs $y$ for the current model time step. This process continues until the simulation is complete.

Each problem when calling $S$-function in simulation process is solved using a special internal function (callback-method). In MatLab $S$-function the following basic tools are used:
1. mdlInitializesizes – Initialization. Prior the first simulation cycle Simulink initializes $S$-function. During this step, Simulink: 1) initializes a structure named SimStruct containing information about $S$-function; 2) sets the number and dimension of input and output ports; 3) sets the model time step for the block; 4) allocates memory to store variables and set the dimension of arrays.
2. mdlGetTimeOfNextVarHit – Calculation of the next block operation time (for blocks with discrete variable step of calculation).
3. mdlOutputs – Calculation of output values at the external step of simulation.
4. mdlUUpdate – Calculation of discrete state variables at the external step of simulation. The discrete state variables keep their value until the next simulation cycle.
5. mdlDerivatives – Calculation of the derivatives of state variables.
6. mdlTerminate – Termination of S-function.

If S-function contains continuous state variables, Simulink calls mdlDerivatives and mdlOutputs callback methods to calculate the derivatives of state variables and output values at the internal simulation steps. The generic template msfuntmp1_basic.m is used for generating the block based on Level-2 Matlab S-function. The template contains skeletal implementations of the required callback-methods determined by the MatLab API interface. To initialize the weights, the approach proposed by Glorough and Bengee is used, namely, that the dispersion should be approximately equal to one [25] in both cases for smooth propagation of activation and gradient over the network. Since it is impossible to satisfy both conditions simultaneously for different layer dimensions, it was proposed to initialize the weights of the next layer of the network by a symmetric distribution with dispersion:

\[
Var(w_i) = \frac{2}{n_in + n_out},
\]

where: \(n_in\) – number of neurons in the input layer; \(n_out\) – number of neurons in the output layer.

The input parameters of the block are: the number of layers of the neural network; the input vector of the neural network, reflecting the number of neurons in each layer; activation function in the hidden layers of the neural network; activation function in the output layer of the neural network; sampling the neural network; training rate factor; moment coefficient; regularization coefficient; condition of stopping the training process; training algorithm.

The study of efficiency of the considered ACS with NN was implemented in the package Simulink of MatLab program by the example of a typical electric drive with a DC motor of one of the coordinates of an industrial robot located in the joint of a robotic system [24]. A variant of complementary correction is the introduction of additional control in the control action (Fig. 2, a). NN parameters are: 1) number of NN layers – 3; 2) number of inputs – 5 (the error signal proportional to the external disturbance and its delay, as well as the signal of the setting action and the actual value of the adjustable coordinate were used as inputs); 3) number of neurons in the middle layer – 12; 4) function of neurons activation in the middle layer is logarithmic; 5) activation function of the output layer of NN is linear; 6) training algorithm is the algorithm of back propagation of error; 7) initial parameter of the training rate is 0.01; 8) discreteness of the network is 0.001.

The control object is affected by external disturbances that are not measured. They are of the form of harmonic. In the process of simulation, the accuracy of the neural network controller was evaluated for typical types of control actions – 1) \(g(t) = \text{const}\), Fig. 3; 2) \(g(t) = V_t\), Fig. 4; 3) \(g(t) = \text{Asin}(\omega t + \phi)\), Fig. 5 – and changes in disturbance parameters given below. Fig. 3, 4, 5 display the curves characterizing ACS operation with the use of dynamic correction implemented by the neural network controller, and without it for various types of control action.

2.3.1.3  

\[X_{\text{in}}, X, \delta, \text{rad}\]

\[\delta_1, \text{rad}\]

\[\delta_3, \text{rad}\]

\[\text{a)}\]

\[\text{b)}\]

Fig. 3. Characteristics of ACS operation in stabilization mode:

a) characteristics of ACS in testing mode of a constant input signal without a correction loop: 1 – reference value of the adjustable coordinate; 2 – the curve of the output coordinate of the system on exposure to disturbance \(f(t) = 3\sin(4\pi + 1.57)\); 3 – error; b) error graphs in ACS with a correction loop on exposure to different parameters of disturbance: 1 - \(f_1(t) = 5\sin(2\pi t + 1.57)\); 2 - \(f_2(t) = 3\sin(4\pi t + 1.57)\); 3 - \(f_3(t) = 5\sin(4\pi t + 0.7)\)
Fig. 4. Characteristics of ACS in the constant speed mode:
a) characteristics of ACS in the constant speed mode without a correction loop: 1 – reference value of the adjustable coordinate; 2 – the curve of the output coordinate of the system on exposure to disturbance \( f(t) = 3\sin(4t + 1.57) \); 3 – error;
b) error graphs in ACS with a correction loop.

Fig. 5. Characteristics of ACS in testing of a harmonic input signal of the type \( y(t) = \sin t \):
a) characteristics of ACS in testing of a harmonic input signal: 1 – reference value of the adjustable coordinate; 2 – the curve of the output coordinate of the system without a correction loop on exposure to disturbance \( f(t) = 3\sin(4t + 1.57) \); 3 – error; 4 – curve of the output coordinate with a correction loop; b) error graphs in ACS with a correction loop.

Analysis of the results of the study using different activation functions in the hidden layer (sigmoid, logarithmic, exponential, sinusoidal) attests that the best convergence of training process is provided by application of sigmoid and exponential activation functions. The advantage of logarithmic activation function is that there is no need to normalize the input signals, however, in the process of testing a harmonic input signal by the system, the oscillatory character of the adjustable variable can be observed, in some cases the loss of stability takes place. The results indicate that the best convergence of training process is provided by the use of SoftPlus activation functions, sigmoid and exponential functions. The obtained error values in the system loop after correction generally meet the requirements for control systems of the considered object. The use of the neurocontroller provides parrying the influence of an external indeterminate disturbance in the range of up to 96.5%.

The study analyzed the accuracy of the correction loop with a neural network controller based on two types of NN with logarithmic activation function in a hidden layer: a recurrent network and a network of direct error propagation. The analysis of the results shows that the use of recurrent networks provides a better rate of convergence of disturbance compensation process, as well as lower error values. However, the lack of accounting for the dynamics of the control object in generation of the training algorithm in testing of a harmonic input signal leads to the need to include an additional component of the gain in the loop. That provides scaling of the output signal of NN, which complicates the correction procedure of the control action. Therefore, it is feasible to modify the training algorithm taking into account the dynamics...
of the control object. The procedure of obtaining derivatives from the control object is complicated under application of
the recurrent structure of NN, thus it is appropriate to use a network of direct error propagation as well.

The analysis of accuracy of the correction loop with a neural network controller with sigmoid activation function in
the hidden layer in testing of a harmonic input signal showed large values of the error ratio in the system loop rather than
using NN with logarithmic activation function. However, in this case there is no need to scale the output signal of NN,
and the compensation process is stable.

Reducing the time discreteness of NN (which actually corresponds to the number of training iterations of the neural
network per one cycle of operation of the system) leads to decrease of the error of the output values of the control object
and reference model at constant parameters of NN.

5. Conclusion

The results of the study of the structures for designing automatic control systems with a neural network controller
based on online training indicate the efficiency of the proposed approach. The study analyzed the accuracy and quality
indicators of the error compensation process caused by external immeasurable disturbance having effect on the object,
when changing the parameters of the neural network controller: the number of layers, neurons, activation function in NN
layers, discreteness of the network, network training algorithm.

The results indicate that the best convergence of the training process is provided by the use of SoftPlus activation
functions, sigmoid and exponential functions. The use of the logarithmic activation function in the output layer precludes
normalizing the output signals. It facilitated the control process.

The obtained error values in the system loop after correction generally meet the requirements for the control systems
of the considered object, the use of the neurocontroller provides parrying the influence of an external indeterminate
disturbance in the range of up to 96.5%. However, the absence of accounting for the dynamics of the control object in
the design of the neural network training algorithm leads to an increase in the error of processing the harmonic input
signal with polyharmonic external influences in the range of up to 15% of the value of the reference-input signal. The
results of the study indicate that the use of the logarithmic activation function in hidden layers in the processing of the
harmonic input signal by the system also requires the inclusion of a link with a transmission coefficient of 0.01 in the
correction loop. In addition, in some cases there was a loss of stability of the compensation process. Therefore, it is
advisable to use the sigmoid activation function, the compensation process in which is stable in all the experiments. In
general, the results of the study indicate the feasibility of the proposed approach to the formation of the control law in
robotic systems affected by external nondeterministic disturbances. However, for full practical implementation it is
necessary to modify the algorithms underlying the proposed approach to improve the accuracy characteristics of the
control system.

The further line of research is the development of modified algorithms for design and operation of neural networks of
systems with complementary control for different types and mathematical models of control objects. Moreover, it is
necessary to consider different types of activation functions and their effectiveness in different frequency ranges of control
and disturbing action, and the optimal discreteness of the network.

6. Acknowledgments

The research is realized with funding from the Russian Fund of Basic Research in the framework of the scientific
project No. 18-08-01126_a.

7. References

for Turbogenerators in a Multimachine Power System. IEEE Transactions on Neural Networks, Vol. 14, Issue 5,