BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES
Vol. 54, No. 3, 2006

Energy flow control system based on neural compensator in the feedback path for autonomous energy source

L.M. GRZESIAK* and J. SOBOLEWSKI

Institute of Control and Industrial Electronics, Department of Electrical Engineering, Warsaw University of Technology, 75 Koszykowa St., 00-662 Warsaw, Poland

Abstract. In this paper an artificial neural network, which realizes a nonlinear adaptive control algorithm, has been applied in a control system of variable speed generating system. The speed is adjusted automatically as a function of load power demand. The controller employs a single layer neural network to estimate the unknown plant nonlinearities online. Optimization of the controller is difficult because the plant is nonlinear and no stationary. Furthermore, it deals with the situation where the plant becomes uncontrollable without any restrictive assumptions. In contrast to previous work [1] on the same subject, the number of neural networks has been reduced to only one network. The number of the neurons in a network structure as well as choosing certain design parameters was specified a priori. The computer test results have been presented to show performance of proposed neural controller.

Key words: neural networks, neurocontrollers, control systems, power electronics.

1. Introduction

Neural network technology has had an enormous influence in development of new approaches to system modelling, estimating and control [2]. Beside fuzzy logic, extended Kalman filters and other competitive methods, it offers good solutions for wide range of problems, where a considered process often involves complicated nonlinear relationships for which classical solutions are either ineffective or unavailable.

Artificial neural networks can be used as a representative framework for modelling nonlinear dynamical systems. It is also possible to incorporate them within nonlinear feedback control structures [3].

Among all types of neural networks, multilayer feedforward networks have primarily been applied in process modelling and control [4].

The ability of the multilayer NNs in mapping nonlinear relationships comes from the nonlinearities within the nodes. It is found that a three layer NN with the backpropagation learning algorithm can model a wide range of nonlinear relationships to a reasonable degree of accuracy [5].

There are typically two steps involved when using neural networks for control: system identification and control design. In the system identification stage, there is designed a NN model of the controlled process. In the control design stage, the model of object is used for training the controller [6]. The performances of NN controller are highly dependant on the accuracy of the object identification.

Our main contribution here is connected with the extension of previous results [1,7–10]. The publication [7] presents an attempt to replace classical PI controller with neural network controller. System was created on the basis of direct modelling where the emulator of the object was implemented as a recursive neural network. Neural networks were trained offline using data acquired by simulation with the fixed object model. That solution did not assure sufficiently good regulating proprieties, especially in case of significant changes of the object.

In publication [1] was presented a similar arrangement to [7] except that the controller was trained online. Those systems all used a process model, which was fixed.

It is possible to develop a neurocontroller in which the emulator of the object is not required at any stage. We propose a regulation structure based on a single neural network which is learning online without any additional neural object emulator. In each of two control architectures described in this paper, the system identification stage is omitted. The controller does not need a process model to predict future performance. In situations where the required controller is less complex than the system model, this type of model-free approach may be one of the most profitable.

Presented solution can be applied to independent electrical energy sources with combustion engine, hybrid electric vehicles, wind turbine power control and other nonlinear systems.

In order to have a high degree of flexibility, the simulation structure was implemented in the Matlab/Simulink software environment.

2. System structure

Presented solution includes system, in which the reference signal is constant however process is deterministically disrupted by load. Figure 1 depicts the load-adaptive variable-speed electricity generating system described in detail in [6]. The system comprises a permanent magnet alternator PMG driven by a combustion engine. A rectifier converts the AC voltage to DC to provide a variable DC voltage to the input of the converter

*e-mail: lmg@isep.pw.edu.pl

$DC/DC-1$, which conditions this voltage to form a DC-link with an appropriately sized energy storage (capacitor C_{dc}). The second converter $DC/DC-2$, supplied by the battery B, is used in case of transient changes of the U_{dc} voltage caused by step load.

Energy flow between the engine, capacitor, battery and load is controlled by means of the converters. The control feature of $DC/DC-1$ converter permits manipulation of the torque acting on the engine, by controlling the current in the alternator. In addition, the voltage in the alternator is allowed to vary freely.

Internal control loop (Engine Control System) is used for stabile operation of engine. The Main Voltage Controller adjusts the engine speed and hence the alternator voltage to maintain the DC link voltage at the reference voltage. The Voltage and Current Controller control system regulates the decoupling converter ($DC/DC-1$) current according to the reference current. In a simple control strategy the reference current could be set to constant. The use of the controllable decoupling converter ($DC/DC-1$) between the generator and the DC-link provides the opportunity to control the load on the engine shaft throughout the variable speed range.

3. Object identification system

To assure a fast response and high performance of control, the configuration of the system is based on a typical control system with compensator in the feedback path [11,12]. Figure 2 presents a simplified block diagram of the model-free control design concept.

NN compensator is constructed as a multilayer feedforward network. The output signal produced by the NN controller Ψ, is subtracted from the reference model's output signal $\tilde{r}$. The resulting signal is used as a compensated reference signal u for the object. This signal should then be adjusted in order to produce a ripple-free output signal y. The subsystem Reference Model is usually a linear filter, which can be designed to introduce desirable robustness and tracking response to the closed-loop system.

Figure 3 shows scheme of the neural network model structure and Fig. 4 shows block diagram for the object identification.

Object is represented by the nonlinear I/O model:

$$y(k+1) = F(y(k), \ldots, y(k-n+1), \\ u(k), \ldots, u(k-m)) \tag{1}$$

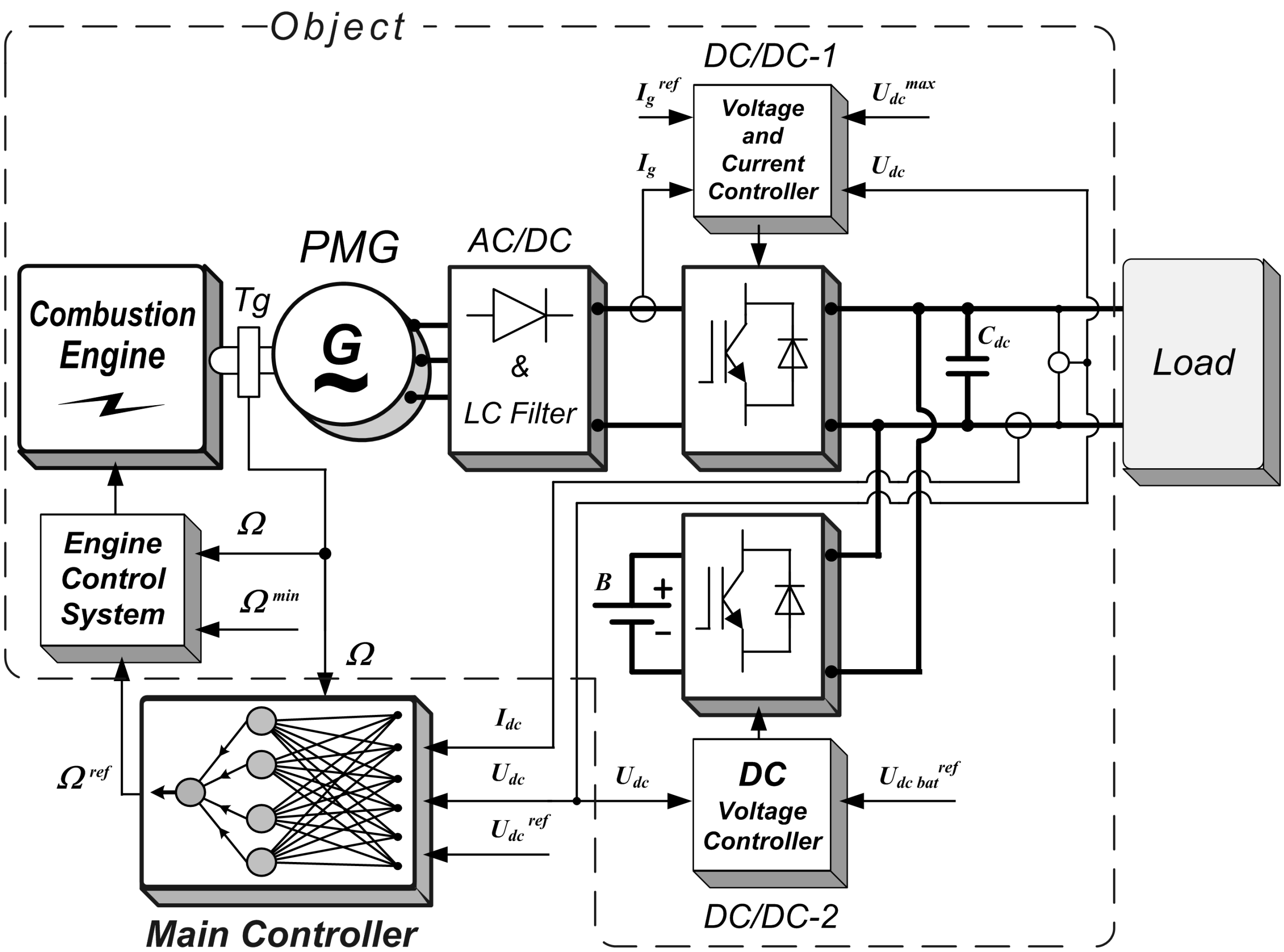

Fig. 1. Scheme of independent electrical energy source with the neural network voltage controller

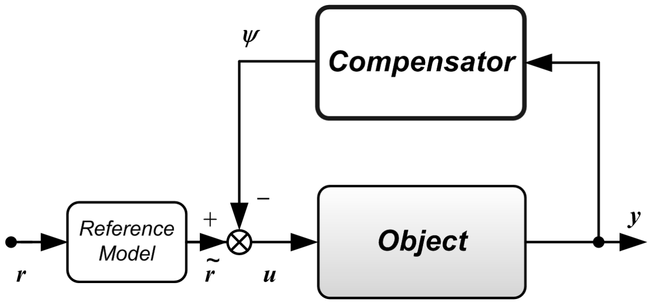

Fig. 2. The model-free control design concept

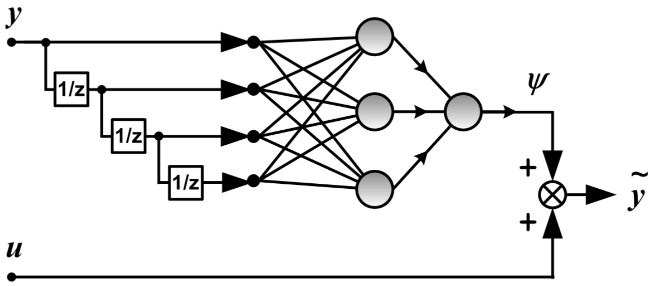

Fig. 3. Neural network model structure

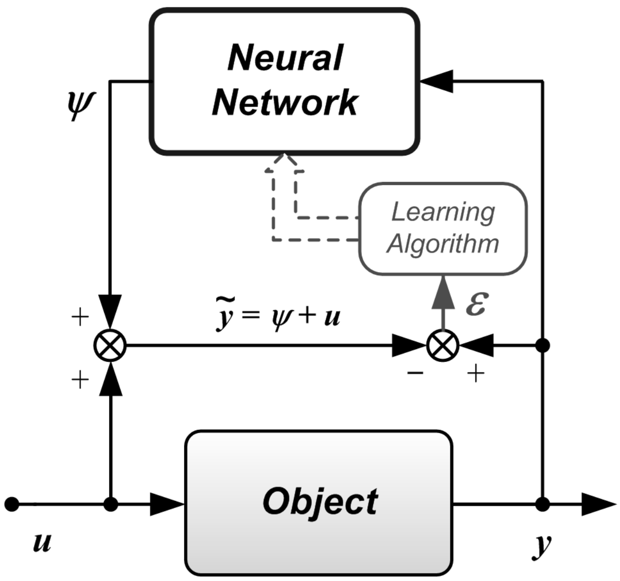

Fig. 4. Block diagram for the object identification

and the math representation of the neural adaptive controller is as follows:

$$\psi(k+1) = NN(\varepsilon(k), W(k), \\ y(k), \ldots, y(k-n+1)) \tag{2}$$

$$u\,(k)\ = \tilde{r}\,(k+1) - \psi\,(k+1) \tag{3}$$

where: $y(k)$ – denotes the output of the object at time index $k, u(k)$ – the input of the object, $\varepsilon(k)$ – the error function, $W(k)$ – computed weight vector, $\Psi(k)$, $\tilde{r}(k)$ – the output of the network and the reference signal, respectively. Substituting (3) to (1) and taking $\tilde{r}(k)$ = const. into consideration, the closed loop dynamics yields:

$$y(k+1) = g(y(k), \ldots, y(k-n+1), \tilde{r}(k), \\ NN(k), \ldots, NN(k-m)). \tag{4}$$

As the object responds, a feedback control uses these measurements to modify the effect of the control. The difference between the object output $y(k)$ and signal $\tilde{y}(k)$ is then used for learning purposes:

$$\tilde{y}\,(k)\ = \psi\,(k) + u\,(k-1) \tag{5}$$

$$\varepsilon\,(k) = y\,(k) - \tilde{y}\,(k) = y\,(k) - \psi\,(k) - u\,(k-1) \tag{6}$$

where: $\tilde{y}(k)$ – denotes the reference signal of the optimization system.

Structure of the control system based on compensator in the feedback path is shown in Fig. 5.

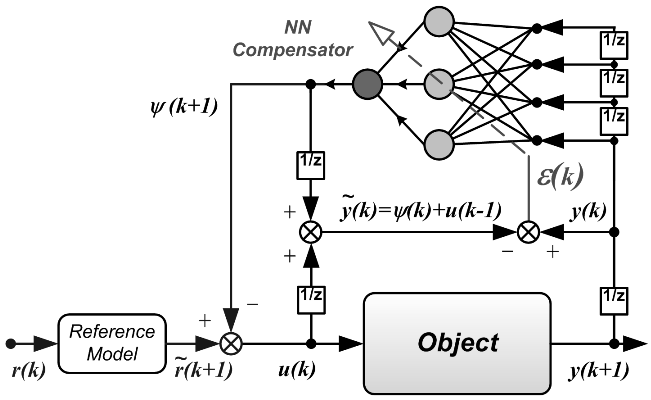

Fig. 5. Structure of the control system based on compensator in the feedback path

The idea is to find a set of weights for the network that maximize the fit to the training data, modified by some sort of weight penalty to prevent overfitting. To increase the stability of the controller we enclosed an integrated error function to the learning vector. In this connection, new output signal of the network could be defined as follows:

$$\psi(k+1) = NN(\varepsilon(k), W(k), y(k), \ldots \\ \ldots, y(k-n+1), \int \{r(k) - y(k)\}\, dk). \tag{7}$$

The goal of the optimization is to find a specific control action $u(k)$ to minimize the desired criterion. We use the squared error cost function, so at any point in time, k, the output error function is:

$$E\,(k) = \frac{1}{2}\varepsilon\,(k)^2\,. \tag{8}$$

It should be noted that system uncertainties of the structure $F(\Delta)$ are compensated by adapting the synaptic connection weights $W(k)$ to minimize the desired control performance [13]. Weights are updated according to the formula:

$$W_{ij}^{(k+1)} = W_{ij}^{(k)} + \Delta W_{ij}^{(k)} \tag{9}$$

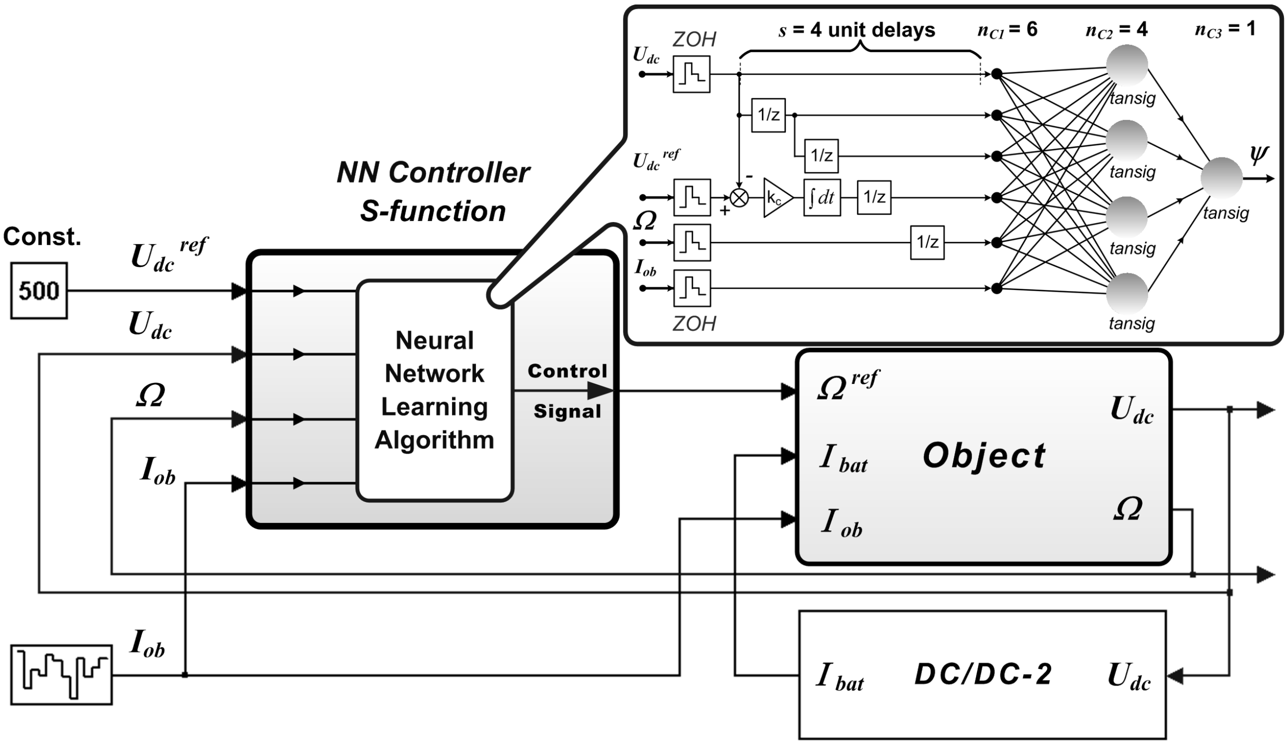

Fig. 6. The NN controller structure

where: $W_{ij}(k+1)$ – new value of the weight from neuron j to i, $W_{ij}(k)$ – previous value of the weight, $\Delta Wij(k)$ – update of the weight.

The function $E(k)$ defined above is used to train the network in the backward path. The error is propagated back through the layers and used to compute the required changes ('deltas') in the weights structure in order to reduce the error level.

4. Optimization algorithm

As a universal function approximator, neural network has a remarkable feature in providing gradient information. Information of its outputs with respect to its weights as well as its inputs can be easily calculated. A major reason for this is the existence of a mathematical framework for selecting the NN weights using proofs based on the notion of energy function, or of algorithms that effectively tune the weights online [14]. Gradient algorithms have played a crucial role in this case, and they are also a popular choice in neuro-control design.

However, it should be kept in mind that these algorithms are useful only when gradients are available, and when cost function is convex [4].

Gradient of the multivariable function gives the direction of the steepest changes therefore even little step in this direction causes a sharp increase of this function. However the same step in the opposite direction causes a sharp decrease of gradient.

The backpropagation algorithm (BP) and its variations offer an effective approach to the computation of the gradient. One such variation, resilient back propagation (RPROP), is one of the best in terms of speed of convergence [15–17]. The two major differences between BP and RPROP are that RPROP modifies the size of the weight step taken adaptively, and the mechanism for adaptation in RPROP does not take into account the magnitude of the gradient, but only the sign of the gradient [2]. This method appears to scale up much better than standard BP as the size and complexity of the learning task grows.

In publication [1] we described other reasons for using RPROP instead of classical backpropagation algorithm in our system. Finally, we used one of few RPROP algorithm version, i.e. RPROP$^-$ [15,18].

The RPROP$^-$ – algorithm omits the weight-backtracking process that relies on holding previous weights values in memory.

It should be emphasized that adaptation of the weights is relatively difficult in practice. If the controller parameters are poor, then the resulting control system may fail, because sets of the controller (i.e. number of neurons, unit delays, sample time as well as parameters of the algorithm) are strongly dependent on a dynamics, complexity and nonlinearity of the process.

5. Control system design and simulation results

The system of electrical energy source with combustion engine was applied in the simulation studies of the neural adaptive controller. The scheme of the neural network used in electrical energy source is shown in Fig. 6.

The output signal of a one-hidden-layer neural network with one output neuron and the sigmoidal node architecture is defined as follows:

$$\Psi(k+1) = \tanh\left(\sum_j W_j \cdot \tanh\left(\sum_i W_{ij} \cdot y_i(k) + W_{j0}\right) + W_0\right) \quad (10)$$

where: $i, j, 0$ – the indexation of neurons in the network structure.

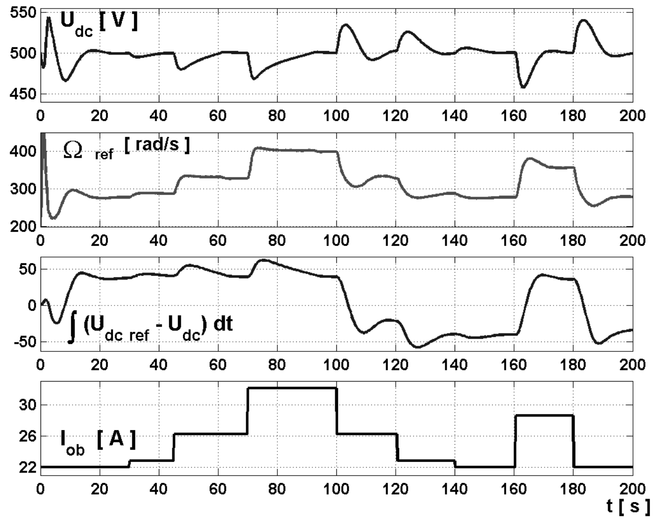

Fig. 7. Output voltage of $DC/DC - 1$ U_{dc}, reference speed Ω_{ref}, integrated error and load current I_{ob} during adaptation process

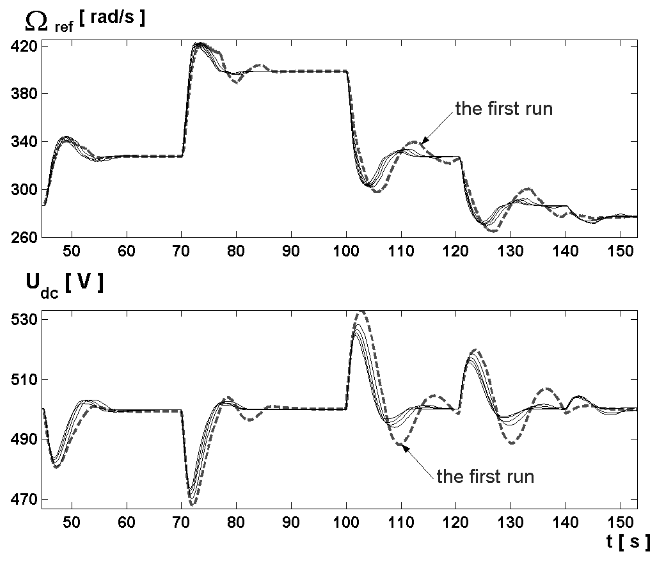

Fig. 8. Reference speed Ω_{ref} and dc-link voltage U_{dc} during adaptation of controller (every 5th run in the midst of 20 runs)

The optimization algorithm is being executed every sampling time and its parameters are updated online. Hidden layer consists of $n_{C2} = 4$ sigmoidal neurons and an output layer contains $n_{C3} = 1$ sigmoidal neuron. The input layer consists of 6 units, however the input vector consists of 2 next samples of voltage U_{dc}, 1 delayed sample of integrated error function, 1 delayed sample of speed Ω and 1 sample of load current I_{ob} (see the NN controller structure in Fig. 6). There are used $s = 4$ unit delays, however sample time of the system is amounted to $T_S = 0.05$ s.

Number of neurons in the network structure, number of unit delays as well as the sample time was selected experimentally. As a teaching method, we used modified backpropagation method on the basis of RPROP$^-$ optimization.

The control of the process starts from random initial values of weights. At the beginning, not trained network generates incorrect values of signal for object that cause the existence of significant output error. During this time, weights of the controller were automatically adapted to the process, and after about 8 seconds controller achieved proper level of precision. To ensure the correct control of the process, weights were adjusted individually. Output voltage U_{dc}, reference speed Ω_{ref} as well as load current I_{ob} during the adaptation process are shown in Fig. 7.

The process was being disturbed by sudden changes of the load current Iob that caused transitory changes of the output voltage U_{dc}. They were continuously and aperiodically bridged by the controller.

To have a representative picture of the system dynamic responses, it is required to simulate the system with at least 20 runs over a 200-second period for various levels of load. Figure 8 shows reference speed Ω_{ref} and output voltage U_{dc} during the adaptation of the controller. Obviously the higher the load level, the more significant the uncertainty of the controlled plant.

Parameter variations of the real electrical energy source are caused by changing in time: fuel quality, fuel/air mixture composition or incorrect control of a fuel injection system. There are possible changes of DC-link as a result of ageing capacitors.

Reference speed curves Ω_{ref} for changed parameters: α (fuel/air mixture coefficient), β (torque coefficient) and C_{dc} (capacity of dc-link) are shown in Fig. 9. Despite changes in the controlled process, neural voltage controller achieved proper dynamics and stability.

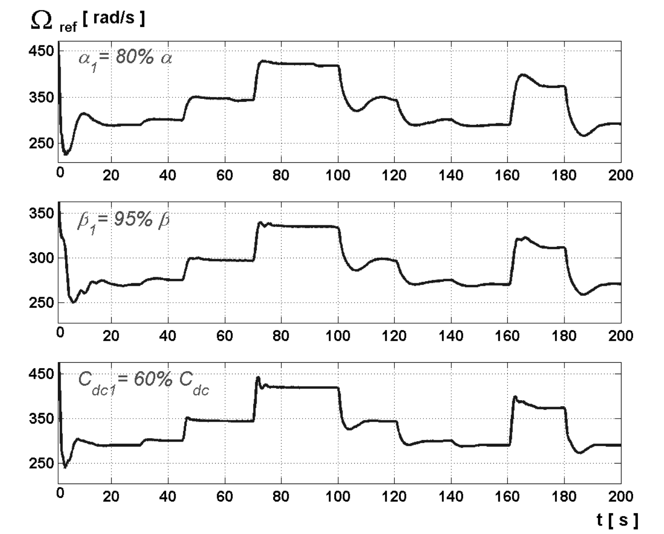

Fig. 9. Reference speed curves Ω_{ref} for changed parameters: α (fuel/air mixture coefficient), β (torque coefficient) and C_{dc} (capacity of dc-link)

6. Conclusions and future work

In this paper, we considered the control problem of a nonlinear load adaptive autonomous energy source system. Optimization of the controller is difficult because the object is nonlinear and no stationary. The presence of a neural network interface made possible to change speed of the generator and hold simultaneously fixed voltage on the output of the system aside from changes of the process.

In creating a control system for the electrical energy source with combustion engine, our aim was for the object to be able to achieve and maintain a goal state regardless of the complexities of its own dynamics or the disturbances it experiences. Configuration of the system is based on control system with compensator in the feedback path.

The main advantage of the system (in comparison with the indirect control system proposed earlier in [1]) is that presented solution does not include a neural network emulator of the object, which emphasizes through the lack of a two-stage control procedure. The lack of an object emulator makes the system adaptable and reduces reliance on knowledge of the object to be controlled.

The resulting performance of presented controller is promising. The system provides an uninterruptible high quality output voltage under the most adverse load conditions: large step/impact loads and nonlinear loads. The ability to learn or at least refine the controller online in real time has been demonstrated. It was shown the ability to cope with changing system parameters too.

The most important task for the immediate future is to apply the new algorithms, perhaps with some additional modifications, to the real-world applications. If the speedup and scaling results hold up in these tests, then we will have achieved something of a breakthrough.

References

[1] L. M. Grzesiakand and J. Sobolewski, "Model reference neural controller for converter based electrical energy source with adjustable speed internal combustion engine", *Electrotechnical Review* 1, 20–25, (2005), (in Polish).

[2] N.K. Treadgold and T. D. Gedeon, "The SARPROP algorithm: a simulated annealing enhancement to resilient back propagation", *Proceedings International Panel Conference on Soft and Intelligent Computing*, Budapest, 293–298 (1996).

[3] K.J. Hunt and D. Sbarbaro, "Studies in neural network based control", *IEEE Control Engineering Series* 46, 94–122 (1992).

[4] D. Rumelhart, G. Hinton, and R. Williams, "Learning internal representations by error propagation", in *Parallel Distributed Processing*, pp. 318–362, Cambridge: MIT Press, 1986.

[5] P.P. Kanjilal, "Adaptive prediction and predictive control", *IEEE Control Engineering Series* 52, 278–282 (1995).

[6] J. Korbicz, A. Obuchowicz, and D. Uciński, *Artificial neural networks, the basics and applications*, Warsaw, 1994, (in Polish).

[7] L.M. Grzesiak and J. Sobolewski, "Converter based electrical energy source with combustion engine, controlled by using the neural voltage regulator", *Electrotechnical Review* 3, 245–251 (2004), (in Polish).

[8] L.M. Grzesiak, W. Koczara, and M. Da Ponte, "Load-adaptive variable-speed electricity generating system - behaviour analyse of dynamic", *8th European Conference on Power Electronics and Applications, EPE'99*, Lausanne, 1–8 (1999).

[9] J. Al-Tayie, R. Seliga, N. Al-Khayat, and W. Koczara, "Steady state and transient performances of new variable speed generating set", *10th European Conference on Power Electronics and Applications EPE'03*, Toulouse, (2003).

[10] L.M. Grzesiak, W. Koczara, and M.Da Ponte, "Novel hybrid load-adaptive variable-speed generating system", *Proceedings IEEE International Symposium on Industrial Electronics ISIE'98*, Pretoria, South Africa, 271–276 (1998).

[11] L.A. Gould, W.R. Markey, K.K. Roberge, and D.L. Trumper, *Control Systems Theory*, Cambridge, 1997.

[12] D. Wyszomierski, *On-Line Trained Neural-Network-Based Speed Controller for AC Motor Drive*, Ph.D. Dissertation, WPW, 2003, (in Polish).

[13] Q. Lu, Y. Sun, and S. Mei, *Nonlinear Control Systems and Power System Dynamics*, Kulwer Academic Publishers, 2001.

[14] O. Omidvar and D. Elliott, *Neural Systems for Control*, Boston, 1997.

[15] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The Rprop algorithm", *Proceedings of the IEEE International Conference on Neural Networks*, 586–591 (1993).

[16] M. Riedmiller, "Advanced supervised learning in multi-layer perceptrons - from backpropagation to adaptive learning algorithms", *Computer Standards and Interfaces* 16, 256–278 (1994).

[17] S.E. Fahlman, *Faster-learning Variations on Backpropagation: An Empirical Study*, San Mateo, 1988.

[18] C. Igel and M. Hüsken, "Improving the rprop learning algorithm", *Proceedings of the Second International Symposium on Neural Computation NC'2000*, 115–121 (2000).

[19] *Using Simulink and Stateflow in Automotive Applications*, The MathWorks Inc., 1998.