INFORMATICA, 1996, Vol. 7, No. 4, 431–454

CONTROL DISTRIBUTED PARAMETER SYSTEMS WITH ORTHOGONAL NEURAL NETWORK LEARNING

Algis GARLIAUSKAS

Institute of Mathematics and Informatics
Akademijos 4, 2600 Vilnius, Lithuania
E-mail: galgis@ktl.mii.lt

Madan M. GUPTA

University of Saskatchewan
Saskatoon, SK, Canada, S7N 5A9

Abstract. Adaptive Control Distributed Parameter Systems (ACDPS) with adaptive learning algorithms based on orthogonal neural network methodology are presented in this paper. We discuss a modification of orthogonal least squares learning to find appropriate efficient algorithms for solution of ACDPS problems. A two times problem linked with the real time of plant control dynamic processes and the learning time for adjustment of parameters in adaptive control of unknown distributed systems is discussed.

The simulation results demonstrate that the orthogonal learning algorithms on a neural network concept allow to find perfectly tracked output control distributed parameters in ACDPS and have rather a good perspective in the development of generalised ACDSP theory and practice in the future.

Key words: adaptive control, distributed parameter system, orthogonal neural network learning, nonlinear control.

1. Introduction. A number of identification and adaptive control proposals have been presented in journals, books, in the past. Before 1970s the analogous realisations of control systems will more developed. The development of digital computers reactivated the creation of different classes of discrete adaptive control algorithms.

Fundamental studies on identification and adaptive control systems were conducted by Åström (1970), Iserman (1974), Tsypkin (1971), Feldbaum (1965), Landau (1979), Nerendra (1980), Parks (1966) and others.

Regarding to Iserman (1974) or Nerendra (1980), different identification and adaptive control models are known either without neural implementations or with feedforward, feedback, and recurrent single-input single-output (SISO) or multi-input multi-output (MIMO) neural networks (Nerendra and Parthasarathy, 1990; Jin *et al.*, 1994). One frequently meets adaptive control models with the reference model approach.

Although there were published very many works on adaptive control theory and applications with a neural networks paradigm or least mean squares (LMS) method for learning models, but little attention is paid to two times problem appearing in design of dynamic adaptive control systems working in real time. One time is the real time of plant processes in dynamics. If we regard these processes as discrete ones described by some order difference equations, we will have a short sampling time. The sample time is the shorter, the quicker a dynamic process of the plant is in reality. Another time is the learning time for adjustment of parameters for identification or adaptive control processes by LMS or neural network approaches, as a rule, by means of gradient methods which are very slowly. To learn the middle size neural network it takes several hundred thousand epochs (steps) (e.g., 50–100 thousand time steps for neural network SISO $N^3_{1.10.10.1}$ using the static back propagation method by Nerendra *et al.* (1990) were required). It means, either the computing devices must be very fast or learning algorithms requiring a short time of calculations, or the plant processes are sufficiently slow. No one of the investigators assessed the relation between these two times. As a rule, the instances are restricted by simplified schemes of a plant with bounded-input bounded-output and stable in the assumed class of input.

Hereby, we choose a class of control objects with distributed parameters as slow plants and suggest speeding up the Orthogonal Least Squares Learning (OLSL) algorithm to overcome two times difficulties without using superpower computers. In Section 2, the general adaptive control of distributed parameter systems and two time (the first of which is the delay time, the second is the learning time) problems are discussed. Mathematical descriptions of concrete heat and gas dynamics systems considering adaptive control peculiarities are presented in Section 3. Orthogonal least squares learning algorithms with neural network control are proposed for improving the computational methods are presented in the next section. In Section 5, a practical importance of the proposed theoretical and applied methodologies, for example of gas pipeline as a distributed parameter control system, is demonstrated.

2. General adaptive control distributed parameter system

2.1. General mathematical description. Most of physical systems are specially distributed and their nature is described by partial differential equations. If we look at these systems from the point of view of ACDPS, control actions are applied to the boundary of the controlled area or in the interior to govern the system to a desired state. It is sometimes possible to approximate ACDPS by an ordinary differential equation model, if the spatial energy distribution of the system is sufficiently concentrated. However, almost as a rule, in many physical systems the energy is wide dispersed, and it is impossible to foresee the system behavior without dealing directly with a partial differential equation description. Another peculiarity of ACDPS is a long time-delay, i.e., that the system pure delay time is much more greater than its ingredient transition time. In the lumped parameter theory, the delay time flows from the system parameters themselves, but when the delay time is much longer, there are no all-round theoretical results yet. Now the outstanding ACDPS theory considering the feature of ACDPS explains many questions dealing with a time-delay phenomenon. It proposes effective methods to overcome difficulties of the time-delay problem in distributed parameter systems (DPS).

It should be emphasized that there exist very many practical tasks and application areas. There are a lot of applied DPS some of which are: heating systems, oil-gas pipeline and sources objects, optimal control systems of meteorology, nuclear reactors, refinery plants, electron, ion, and laser or piezoceramic affect in the context of mobile distributed parameter systems, and others.

In the general case a nonlinear distributed parameter system (NDPS) is mathematically presented in dynamics by Butkovskiy *et al.*, (1991) with fixed sources

$$\frac{\partial Q}{\partial t} = a(Q)\nabla Q + F(x,y,z,t) - \psi(Q), \quad (x,y,z) \in V,\ t > 0, \tag{1}$$

$$Q(x,y,z,0) = Q_0(x,y,z), \quad (x,y,z) \in V, \tag{2}$$

$$\left[\alpha Q + \lambda \frac{\partial Q}{\partial n}\right]_{(x,y,z)\in\Gamma} = \alpha Q_\Gamma, \tag{3}$$

where $Q = Q(x, y, z, t)$ is an ingredient (heat, temperature, volume of oil or gas, pressure and others; hereby we orient ourselves to heat and gas pipeline cases) of distributed forward spatial coordinates $x,\ y, z$ and the time t;

$a(Q)$ is a thermal conductivity coefficient;
$F(x, y, z)$ is a heat source function;
$\psi(Q)$ is the function for removing heat from the system;
$Q_0(x, y, z)$ is an initial distribution of heat;
$\alpha,\ \lambda$ are constants;
$\partial Q/\partial n$ is a derivative in the external normal direction to a domain of boundary;
Q_Γ is an assigned number;
V is a bounded domain, where the system is determined.

Eq. 1 defines the state parameters of NDPS in dynamics. Eqs. 2 and 3 characterize the initial and boundary conditions, respectively.

The heat source can be presented by such a function

$$F(x, y, z, t) = u(t)\Phi[x, y, z], \tag{4}$$

where $u(t)$ is the power of a heat source at time t, $\Phi[x, y, z] \geqslant 0$ is a heat source power distribution usually, the Gaussian distribution

$$\iiint_{-\infty}^{\infty} \Phi[x, y, z]\, dxdydz = 1. \tag{5}$$

A control problem of ACDPS will be as follows. Let a desired system state $Q_d(x, y, z)$, $(x, y, z) \in D$ be given. One needs to find the trajectory $Q(x, y, z, t)$, provided by the heat source power $u(t)$ as a control function contenting the limitations $0 \leqslant u(t) \leqslant V_m$, ($V_m$ is the maximum value of the control parameter function), which in the steady-state $(t \to \infty)$ provides a minimal deviation from the desired state not exceeding the value δ, that is,

$$I = \min_{t \to \infty} \int_V [Q_d(x, y, z) - Q(x, y, z, t)]^2 dxdydz \leqslant \delta. \tag{6}$$

If we accept the sampled period θ, then the steady-state decision can be found approximately replacing Eqs. $1-3$ by the average stationary equations

$$a(\overline{Q})\nabla\overline{Q} + \overline{F}(x, y, z) - \psi(\overline{Q}) = 0, \quad (x, y, z) \in V \tag{7}$$

with boundary conditions (3). Here $\overline{Q}(x,y,z)$ is average system state

$$\overline{Q}(x,y,z) = \frac{1}{\theta}\int\limits_{t}^{t+\theta} Q(x,y,z,\tau)\,d\tau, \tag{8}$$

and $\overline{F}(x,y,z)$ is as follows

$$\overline{F}(x,y,z) = \frac{1}{\theta}\int\limits_{t}^{t+\theta} u(\tau)\Phi[x,y,z]\,d\tau. \tag{9}$$

The functional (6) will be changed to the following

$$I_{\mathrm{st}} = \int\limits_{V} \left[Q_d(x,y,z) - \overline{Q}(x,y,z)\right]^2 dxdydz \leqslant \delta, \tag{10}$$

which is transformed to the minimum varying $u(t)$ in the rate $0 \leqslant u(t) \leqslant V_m$.

2.2. ACDPS problem with dynamic neural network. In the general case, an ACDPS problem can be formulated and solved on the ground of Dynamic Recurrent Neural Networks (DRNN's). (Nerendra *et al.*, 1990; Jin *et al.*, 1994). A continuous representation with nonlinear (1) – (3) mathematical description and DRNN'S learning is very hard for realisation, though it is possible in principle. A more convenient case is an approximation as continuous in time and discrete in space of a linear/nonlinear dynamic control system. Adaptive learning and control of such a system may be represented by linear/nonlinear differential (at the time of state variables) and difference (in space distributed parameters and control values) equations. Taking into account a pure discrete representation of equations (1) – (3) a discrete approximation of ACDPS and an application tor adaptive control of discrete DRNN'S are possible. But due to a great complexity in the SISO or SIMO case, we insist on refusing DRNN's. It was sought for other simpler and more efficient neural network presentations using the ideas of orthogonality.

Since of the considered systems and their processes are nonlinear with unknown parameters in nature and the control parameter functions are not separable or additive with respect to state functions; we must use a general form of an adaptive control model. We assume that for an unknown stable time-invariant

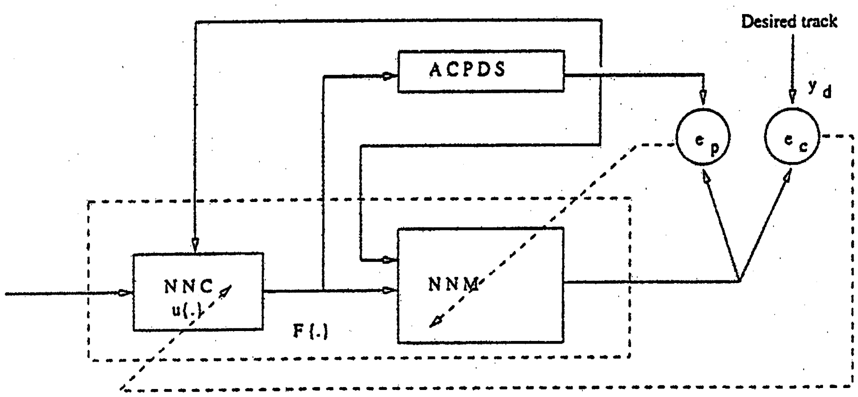

Fig. 1. Control system structure.

system, the input and output signals of discrete-time ACDPS are observable and measurable. The general form is as follows

$$\hat{\mathbf{y}}(k) = \mathbf{F}\big[\mathbf{y}(k-1),\ldots,\mathbf{y}(k-n_a),\mathbf{u}(k-1),\ldots,\mathbf{u}(k-n_a)\big], \qquad (11)$$

where $\mathbf{u}(k)$ and $\mathbf{y}(k)$ are multidimensional system input and output vectors, respectively, $\mathbf{F}(\cdot)$ is the system vector-function with unknown parameters, and $\hat{\mathbf{y}}(k)$ is the estimated vector of $\mathbf{y}(k)$. There is an assumption that the system is a continuous functional of the input, i.e., even an insignificant change in the input results in a slight change in the output of the system.

The models based on Eq. 10 are presented by Nerendra *et al.* (1990). We can construct an appropriate neural network model that produces an output vector $\hat{\mathbf{y}}(k)$ as an estimate of $\mathbf{y}(k)$ by a specified cost function of errors

$$\mathbf{e}(k) = \mathbf{y}(k) - \hat{\mathbf{y}}(k), \qquad (12)$$

equals to the minimum. This error function can also be defined by the L_2-norm criterion

$$\mathbf{E} = \frac{1}{2}\sum_k \left|\mathbf{e}(k)\right|^2. \qquad (13)$$

In order to concretise the unknown plant-control system, we propose a direct parallel control method without the reference model. The scheme is represented in Fig. 1. There are two loops: the control loop with a Neural Network Controller (NNC) and the parameter adjustment loop with a Neural Network Mapper (NNM) compensating the error (13).

3. One-dimensional spatial parameter DPS

3.1. Heat-transfer DPS. Some DPS may be represented as a one-dimensional spatial distributed parameter system. There are, for example, heat, oil or gas transfer systems. Let us consider the a heat exchange system when hot water is transferring heat for heating cold water: two flows are flowing in opposite directions.

Thus, we have one spatial and two state parameter systems whose mathematical model for the heating process without loss of heat (Guangyuan *et al.*, 1994) is the following

$$\begin{cases} c\rho\dfrac{\partial Q}{\partial t} + vc\rho\dfrac{\partial Q}{\partial x} = a(P - Q) \\ dr\dfrac{\partial P}{\partial t} - udr\dfrac{\partial P}{\partial x} = a(Q - P), \end{cases} \tag{14}$$

where Q and P are the temperature of cold and hot water, respectively; c, d are specific heats; ρ, r are densities; a is a heat-transfer coefficient; v, u are cold and hot water velocities.

The initial and boundary conditions are as follows

$$Q(x,0) = Q_0(x), \quad P(x,0) = P_0(x), \tag{15}$$

$$\begin{cases} Q(0,t) = Q_0(t), \quad P(0,t) = P_0(t) \\ Q(L,t) = Q_L(t), \quad P(L,t) = P_L(t), \end{cases} \tag{16}$$

where expression (16) characterizes a couple of boundary conditions.

We represent below a discrete description of differential Eqs. 14 and their initial and boundary conditions (15), (16). Continuous-time-space variables $Q(x,t)$ and $P(x,t)$ are sampled to obtain discrete-time-space variables $Q_j(k)$ and $P_j(k)$, respectively, where k is a discrete-time instant and j is a discrete-space index. (11) – (13) are transformed to a system of linear difference equations

$$\begin{cases} \rho Q_j(k+1) = c\rho Q_j(k) + a\big(P_j(k) - Q_j(k)\big) \\ \qquad\qquad - \dfrac{vc\rho}{x_{j,j-1}}\big(Q_j(k) - Q_{j-1}(k)\big) \\ drP_j(k+1) = drP_j(k) + a\big(Q_j(k) - P_j(k)\big) \\ \qquad\qquad - \dfrac{udr}{x_{j,j-1}}\big(P_j(k) - P_{j-1}(k)\big), \end{cases} \tag{17}$$

$$k = 0, 1, \ldots, N; \; j = 1, 2, \ldots, L-1,$$

where $x_{j,j-1}$ is the discrete length of the jth compartment of water pipeline.

The appropriate conditions become

$$Q_j(0); \; Q_j(0) \quad \text{(initial)}; \quad j = 1, 2, \ldots, L, \tag{18}$$

$$\begin{cases} Q_0(k), \; P_0(k) & \text{(boundary at the begining)} \\ Q_L(k), \; P_L(k) & \text{(boundary in the end).} \end{cases} \tag{19}$$

Sometimes a steady-state situation of control is useful, therefore we present stationary equations

$$\begin{cases} vc\rho \dfrac{\partial Q}{\partial x} = a(P - a) \\ udr \dfrac{\partial v}{\partial x} = a(Q - P), \end{cases} \tag{20}$$

and the difference ones

$$\begin{cases} \dfrac{vc\rho}{x_{j,j-1}}(Q_j - Q_{j-1}) = a(P_j - Q_j) \\ \dfrac{udr}{x_{j,j-1}}(P_j - P_{j-1}) = a(Q_j - P_j) \quad j = 1, 2, \ldots, L-1. \end{cases} \tag{21}$$

The boundary conditions remain the same as (19). Analytic solutions of Eqs. 20 are presented by Guangyuan *et al.* (1994).

3.2. Adaptive control. An adaptive control task is to find the unknown parameters of Eq. 14, if we bear in mind that the control parameter is the temperature $P_1(k)$ at the initial part of a hot water pipeline and a desired output is the cool water temperature $Q_L(k)$ of the end of a cool water pipeline. This control task deals with a design of a heat pipeline system. Another task deals with the exploitation of a really existing system with dynamics under consideration. Now the parameters of the system are known. The problem of control is to design a controller that generates the desired control $P_1(k)$ based on the information at the discrete moment k.

From the general control theory consideration, a control distributed parameter problem, according to Eqs. 17 – 19, is formulated as a discrete-time, nonlinear nonautonomous system (because of conditions (15)), and in the case of Eqs. 20, 21 as a linear autonomous system. The first system is nonlinear due to the initial conditions verified on the space. Sometimes it can be constant, that is, the set point is not varying in time. In this case, it becomes a linear autonomous system too. The stability of a linear system with one equilibrium point, as the time is tending to infinity, is investigated and it is obvious (Nerendra *et al.*, 1990).

If the adjustable parameters of a heat system are unknown, that is a design problem, it is possible to use for modeling multi-input single-output (MISO) neural network models. That means the output variable at time $k+1$ is a linear combination of the past (at time k) values of both the input and the output ones. There are two structional models: a parallel model and a series-parallel model (Nerendra *et al.*, 1990).

3.3. Gas pipeline mathematical description. In the general case, problems based on the expansion of gas supply systems, deal with both the existing and newly designed pipelines with compressor plants. A conventional gas pipeline plant consists of pipeline sections and compressor plants (CP) situated among sections, as shown in Fig. 2 (Garliauskas and Feigin, 1989).

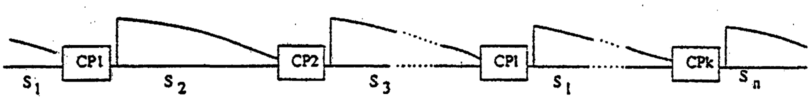

Fig. 2. Pipeline with compressor plants (PCl).

A mathematical description of a gas pipeline section with distributed pressures $P_l(x,t)$ and flows $G_l(x,t)$ is as follows

$$\frac{\partial P_l(x,t)}{\partial t} = \frac{2DF}{\lambda G_l(x,t)} \frac{\partial^2 P_l^2(x,t)}{\partial x^2}, \quad l = 0,1,\ldots,M+1, \tag{22}$$

where D, F are the diameter and the square of pipeline unchanging along the line, respectively, λ is the coefficient of hydraulic resistance, M is the number of sections.

Allowing that the velocity of gas flow is average for section l, we obtain a slightly simplified partial differential equations

$$\frac{\partial P_l(x,t)}{\partial t} = a_l \frac{\partial^2 P_l^2(x,t)}{\partial x^2}, \quad l = 0,1,\ldots,M+1, \tag{23}$$

where $a_l = \dfrac{2D}{\lambda \overline{\rho}_l \overline{w}_l}$, $\overline{w}_l$ is the mean of velocity in section l, $\overline{\rho}_l$ is the mean of density in section l.

The initial and boundary conditions are

$$\begin{aligned} &P_l(x,0) = P_l^{(0)}(x), \\ &P_0(0,t) = f_0(t), \\ &P_{M+1}(M+1,t) = f_{M+1}(t). \end{aligned}$$

The compressor plant work may be characterised by the following relation, that is the degree of compression

$$\varepsilon_l = \frac{P_l(0,t)}{P_{l-1}(m,t)} = c_l u_l^2 \varepsilon_l(q_v) + 1, \tag{24}$$

where $\varepsilon_l(q_v) = a_{l0} + a_{l1} q_{lv} + a_{l2} q_{lv}^2$ is the polynomial of volume flow q_{lv}, a_{l0}, a_{l1}, a_{l3}, c_l are coefficients, u_l is the ratio of compressor revolutions as a control parameter for section and compressor plant l.

Continuous-time-space variables $P_l(x,t)$ are sampled to obtain discrete variables $P_{lj}(k)$, where j is the index of the jth compartment of pipeline section l, k is the time index or cycle number (positive integer) a unit of the sampling interval. Since partial differential Eqs. 23 are determined at discrete-time-space instants, they can be formulated as system of nonlinear equations

$$\begin{aligned} P_{l_j}(k+1) = &P_{l_j}(k) + a_l \left\{ \frac{[P_{l_{j-1}}^2(k) - P_{l_j}^2(k)]}{x_{j-1,j}} \right. \\ &\left. - \frac{[P_{l_j}^2(k) - P_{l_{j+1}}^2(k)]}{x_{j,j+1}} \right\}, \quad \begin{array}{l} l = 1,2,\ldots,M+1, \\ j = 0,1,\ldots,m-1. \end{array} \end{aligned} \tag{25}$$

The appropriate initial and boundary conditions are

$$P_{l_j}(0) = P_{l_j}^{(0)}; \quad P_{10}(k) = f_1(k); \quad P_{M+1\,m}(k) = f_{M+1}(k). \tag{26}$$

The main relation of a compressor plant now will be

$$\varepsilon_l = \frac{P_{l+1\,0}(k)}{P_{lm}(k)}, \tag{27}$$

which is a discrete presentation of Eq. 24.

3.4. Pipeline control problem on neural network. Consider a concrete pipeline system with compressor plants dynamic processes of which are described by difference Eqs. 25 and conditions 26 – 27. Basing on the control system presented in Fig. 1, we try to concretise the presentation of neural network control and mapper blocks joining them into one general block with a function $\mathbf{F}\{\cdot\}$.

A changed mathematical description (22) is presented for the first pipeline section (pipeline and CP) as follows:

$$\begin{cases} P_{11}(k+1) = P_{11}(k) + a_{11} f\big(P_{10}(k), P_{11}(k), P_{12}(k)\big) \\ P_{12}(k+1) = P_{12}(k) + a_{12} f\big(P_{11}(k), P_{12}(k), P_{13}(k)\big) \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ P_{1j}(k+1) = P_{1j}(k) + a_{1j} f\big(P_{1j-1}(k), P_{1j}(k), P_{1j+1}(k)\big) \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ P_{1m-1}(k+1) = P_{1m}(k) + a_{1m-1} f\big(P_{1m-2}(k) \\ \qquad\qquad P_{1m-1}(k), P_{1m}(k)\big), \end{cases} \tag{28}$$

where $\mathbf{f}\{\cdot\}$ is an unknown function in the general case.

Further, the initial conditions are expressed by the harmonic function

$$P_{10} = f(k), \tag{29}$$

and the right side boundary condition by a setpoint at the beginning of the next pipeline section or a desired value of compression in the output of CP.

$$P_{20}(k) = P_{20}^{(d)} = P_{1m}(k)\varepsilon_1. \tag{30}$$

Now the error function will be the scalar

$$e(k) = P_{20}^{(d)} - P_{20}(k). \tag{31}$$

Eq. 30 allows us to include $P_{20}^{(d)}$ and the control parameter ε_1 into the last expression of equation system (28) substituting $P_{1m}(k)$ of Eq. (30).

Bearing in mind the realisation of a pipeline adaptive control system by neural networks, it will correspond to the multi-variable version of Nerendra's model. We try to generalise this model by so called cascade subnetworks.

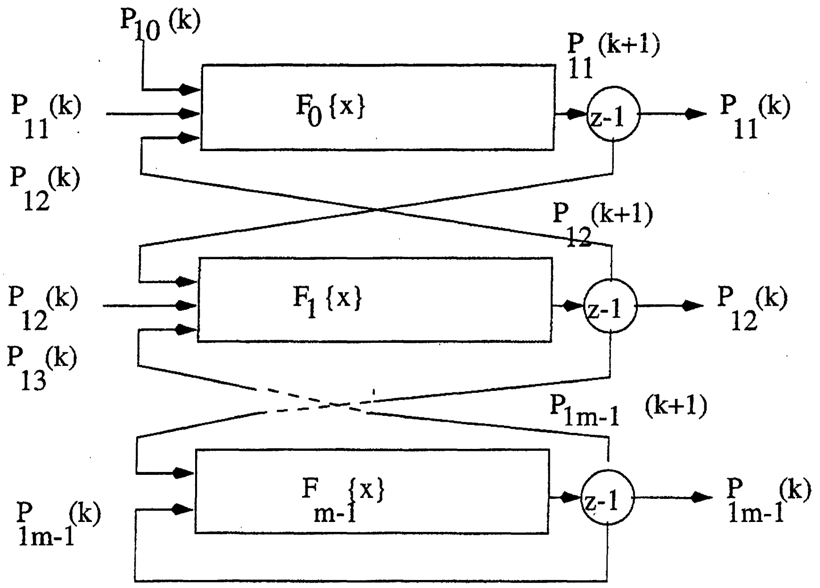

Fig. 3. Cascade of a subnetwork.

The presentation of subnetworks of the first pipeline section in detail are shown in Fig. 3, 4. Here the MISO neural network structure is proposed with crossed junctions between neighbour subnetworks. A specific feature of distributed control systems, in general, and the pipeline system, in part is that the output state variables $P_{11}(k), P_{12}(k), \ldots, P_{1j}(k), \ldots, P_{1m-1}(k)$ are not controllable, they are used for modeling next subnetworks. Only the last compartment output $P_{1m}(k)$ as a CP input is controllable because it and the CP control parameter provide the achievement of a possible desirable state variable value of the next section, for example, $P_{20}^{(d)}$.

Representing the subnetwork system in Fig. 4 by a common mapper function $\mathbf{F}_l\{\cdot\}$ for section l, we construct complete structure by the cascade principle (Fig. 5). The modeling values of state variables at the end of sections are expressed by the vector function $\widehat{\mathbf{P}}(k) = \{\widehat{P}_1(k), \widehat{P}_2(k), \ldots, \widehat{P}_l(k), \ldots, \widehat{P}_{M+1}(k)\}$ and the control vector $\mathbf{u}(k) = \{u_1(k), u_2(k), u_l(k), \ldots, u_M(k)\}$. At time the vector $\widehat{\mathbf{P}}(k)$ converges to the desirable value. Under the assumption that the vector $\mathbf{u}(k)$ is bilaterally bounded and state functions are continuous, the stability of such a system is undoubted.

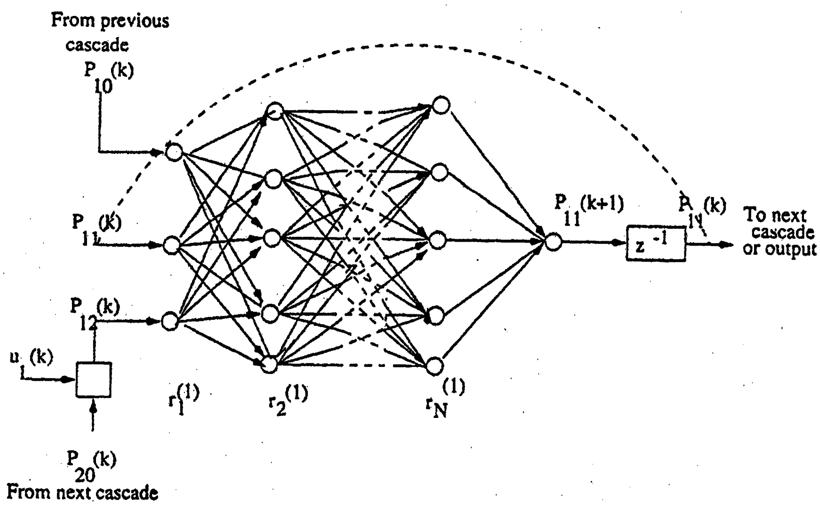

Fig. 4. The recurrent network of the cascade with a local control sub-network.

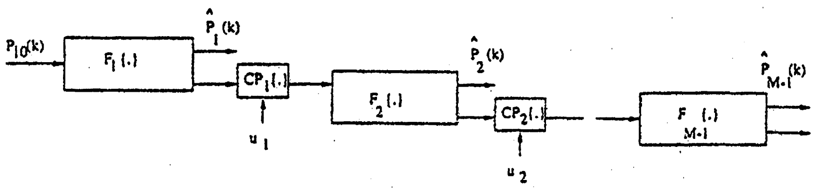

Fig. 5. The cascade neural network of the pipeline.

4. Orthogonal least square learning algorithms. Different orthogonal transforms are widely used in applications of signal and image processing, control problems. We will discuss below a modification of the orthogonal least squares learning (OLSL) algorithm in order to find an appropriate efficient algorithm for solution of ACDPS problems.

It is known that the orthogonality is a generalization of geometric perpendicularity. In the general, case for complex functions, if they exist within the range $-\infty < x < \infty$, the orthogonality can be written as follows

$$S_{\text{orth}} = \int_{-\infty}^{\infty} \Phi_r(x)\Phi_s(x)\,dx = 0, \qquad (32)$$

or for the discrete case and when the argument is finite

$$S_{\text{orth}} = \sum_{k=0}^{N-1} \boldsymbol{\Phi}_r(k)\boldsymbol{\Phi}_s^*(k) = \delta_{r-s} = \begin{cases} 0, & r \neq s, \\ 1, & r = s, \end{cases} \tag{33}$$

where $\boldsymbol{\Phi}_r$, $\boldsymbol{\Phi}_s^*(k)$ are complex conjugate functions in the continuous and the discrete case, respectively. δ_{r-s} is the delta function.

Further, we use a feedforward neural network (FNN) with a multi-input and single-output (MISO) modification for adaptive control systems of unknown distributed parameter plants. FNN can be simple two-layer or radial basis systems (RBS). The objective is to find an algorithm with optimal FNN weights and the amount of the first layer elements. The model based on a linear regression can be formulated as the continuous time case

$$y(t) = \sum_{i=1}^{m} p_i(t)\Omega_i + e(t), \tag{34}$$

where $p_i(t) = p_i(x(t))$, Ω_i is the weight of input independent of time, $e(t)$ is the error function. The discrete-time case is as follows

$$\mathbf{y} = P\boldsymbol{\Omega} + \mathbf{E}, \tag{34$'$}$$

where $\mathbf{y} = [y(0), \dots, y(N-1)]^T$ is a measurable output signal called a the regressed variable column-vector $P = [\mathbf{p}_1, \dots, \mathbf{p}_m]^T$, $(\mathbf{p}_i = [p_i(0), \dots, p_i(N-1)])$, $1 \leqslant i \leqslant m$ is the matrix of known quantities-regressors, $\boldsymbol{\Omega} = [\Omega_1, \Omega_2, \dots, \Omega_m]$ is the row-vector of unknown parameters to be estimated and $\mathbf{E} = [e(0), \dots, e(N-1)]^T$ is the error column-vector of unknown variables, m is the number of the first layer elements, N is the upper limit of the time variable, i.e., $0 \leqslant t \leqslant N-1$ (Widrow *et al.*, 1987; Wang, 1991).

According to the LMS algorithm

$$\boldsymbol{\Omega}_{i+1} = \boldsymbol{\Omega}_i + 2u\mathbf{e}_i\mathbf{x}_i^*, \tag{35}$$

where u is the control parameter of adaptation speed.

It is well known that an arbitrary sequence at j can be transformed to the superposition of orthogonal family functions (Wang, 1991).

Using the transform matrix $\boldsymbol{\Phi}_s^*(k)$ as the Walsh-Hadamard transform (Ahmed and Rao, 1975), having, in a sense, a binary representation of $\mathbf{x}_i^*$ vector, it is possible to do a forward orthogonal transformation, because the orthogonality relation (33) becomes as follows.

$$\mathbf{x}^T\mathbf{x}^* = \delta_{r-s}. \tag{36}$$

Substituting e_i values into (35) and allowing that the initial vector Ω_0 equals zero, the weight vector versus time, using the orthogonality relation (36) at $j = N$, becomes

$$\Omega_N^* = 2u \sum_{k=0}^{N-1} \mathbf{y}_k \mathbf{x}_k^* = 2u \sum_{k=0}^{N-1} \sum_{s=0}^{N-1} \mathbf{y}_k \Phi_s(k). \tag{37}$$

In order to obtain an inverse orthogonality transform of $\mathbf{y}_k$ and appropriate Ω_N it is necessary to transform the matrix $\boldsymbol{\Phi}_s^*(k)$ into $\boldsymbol{\Phi}_k(s)$, i.e., $\boldsymbol{\Phi}_k(s) = \boldsymbol{\Phi}_s^{*T}(k)$. Thus, we have built the orthogonal LMS learning algorithm for finding the dynamic neural network weight vectors. Another generalisation of LMS orthogonality can be found in the paper by Chen *et al.*, 1991.

Another objective is to define a rational subset of the number of neural element in MISO or the number of centers in RBS from the sample set. Such a number is m of (34) and the orthogonal number m_r is considerably smaller than m $(m_r \ll m)$.

The projection $\mathbf{P}\boldsymbol{\Omega}$ is a part of the desired output $\mathbf{y}$ energy defined by regression. The OLSL algorithm involves the mapping of $\mathbf{P}_i$ set onto an orthogonal basis vectors set. Therefore, $\mathbf{P}_i$ vectors allow to find contribution into the desired output energy.

After decompositing the regression matrix $\mathbf{P}$ by

$$\mathbf{P} = \mathbf{V}\mathbf{B}, \tag{38}$$

where $\mathbf{B}$ is a triangular matrix and $\mathbf{V}$ is the matrix with orthogonal column vectors $\mathbf{V}_i$ which are basis vectors spanned in the same space as $\mathbf{P}_i$ and formula (34′) changes into

$$\mathbf{y} = \mathbf{V}\mathbf{q} + \mathbf{E}, \tag{39}$$

where the vectors $\widehat{\mathbf{q}}$ and $\widehat{\boldsymbol{\Omega}}$ as solutions satisfy the condition

$$\widehat{\mathbf{q}} = \mathbf{B}\widehat{\boldsymbol{\Omega}}. \tag{40}$$

Using the well known classical Gram-Schmidt method (Björck, 1967) the energy of $y(t)$ can be written

$$\mathbf{y}^T\mathbf{y} = \sum_{i=1}^{m} q_1^2 \mathbf{V}_i^T \mathbf{V}_i + \mathbf{E}^T\mathbf{E}. \tag{41}$$

The average value of energy will be such

$$(\mathbf{y}^T\mathbf{y}) = \frac{1}{N}\left(\sum_{i=1}^{m} q_i^2 \mathbf{V}_i^T \mathbf{V}_i + \mathbf{E}^T\mathbf{E}\right), \tag{42}$$

where $\frac{1}{N} q_i^2 \mathbf{V}_i^T \mathbf{V}_i$ is an increment.

For definition of the best fitness of $\mathbf{q}_i$ and $\mathbf{V}_i$ to the desired values, we derive a nomination of reliability

$$R_i = 1 - \mathbf{q}_i^2 \mathbf{V}_i^T \mathbf{V}_i / (\mathbf{y}^T\mathbf{y}), \quad i = 1, 2, \ldots, m. \tag{43}$$

Eq. 43 is the criterion for seeking an efficient procedure of a rational regressor subset by to the Gram-Schmidt method.

The selection procedure is such:

Step 1. At the first step, define

$$\begin{cases} \mathbf{v}_1^{(i)} = \mathbf{p}_i; \quad q_1^{(i)} = (\mathbf{v}_1^{(i)})^T \mathbf{y} / h_1^{(i)}; \quad h_1^{(i)} = (\mathbf{v}_1^{(i)})^T \mathbf{v}_1^{(i)}, \\ R_i = R_1^{(i_1)} = 1 - \max\limits_{1 \leqslant i \leqslant m} \{(q_1^{(i)})^2 h_1^{(i)} / (\mathbf{y}^T\mathbf{y})\}, \end{cases} \tag{44}$$

and a worse reliability suits

$$\mathbf{v} = \mathbf{v}_1^{(i_1)} = \mathbf{p}_{i_1}.$$

Step 2. At the second step, find

$$\begin{cases} \beta_{12}^{(i)} = \mathbf{w}_1^T \mathbf{p}_i / (\mathbf{v}_1^T \mathbf{v}_1), \\ \mathbf{v}_1^{(i)} = \mathbf{p}_i - \sum\limits_{j=1}^{1} \beta_{j2}^{(i)} \mathbf{v}_j, \\ q_2^{(i)} = (\mathbf{v}_2^{(i)})^T \mathbf{y} / h_2^{(i)}, \end{cases} \tag{45}$$

$$R_2 R_2^{(i_2)} = 1 - \max_{\substack{1 \leqslant i \leqslant m \\ i \neq i_1}} \{(q_2^{(i)})^2 h_2^{(i)} / (\mathbf{y}^T\mathbf{y})\},$$

$$\mathbf{v}_2 = \mathbf{v}_2^{(i_2)} = \mathbf{p}_{i2} - \sum_{j=1}^{1} \beta_{j1}\mathbf{v}_j.$$

....

Step k. At the kth step, where $k \geqslant 2$, compute

$$\begin{cases} \beta_{jk}^{(i)} = \mathbf{v}_j^T \mathbf{p}_i / (\mathbf{v}_j^T \mathbf{v}), \quad 1 \leqslant j < k, \\ \mathbf{v}_k^{(i)} = \mathbf{p}_i - \sum_{j=1}^{k-1} \beta_{jk}^{(i)} \mathbf{v}_j, \\ q_k^{(i)} = (\mathbf{v}_k^{(i)})^T \mathbf{y} / h_k^{(i)}, \end{cases} \tag{46}$$

$$R_k R_k^{(i_k)} = 1 - \max_{\substack{1 \leqslant i \leqslant m \\ i \neq i_1, \ldots, i \neq i_{k-1}}} \left\{ (q_k^{(i)})^2 h_k^{(i)} / (\mathbf{y}^T \mathbf{y}) \right\},$$

and select

$$\mathbf{v}_k = \mathbf{v}_k^{(i_k)} = \mathbf{p}_{ik} - \sum_{j=1}^{k-1} \beta_{jk}\mathbf{v}_j,$$

where $\beta_{jk} = \beta_{jk}^{(i_k)}$, $1 \leqslant j < k$, are the elements of matrix $\mathbf{B}$.

After excluding all $i_1, i_2, \ldots, i_{k-1}$ a definition of m_s has been obtained upon satisfying this condition

$$R_s = 1 - \sum_{j=1}^{m_s} R_j \geqslant r_0, \tag{47}$$

where $0 < r_0 < 1$ is the desired reliability.

Thus, we have an efficient learning algorithm of neural networks unknown parameters and network architecture able to approximate a discrete distributed parameter system through on-line learning processes.

5. Simulation results. In this section simulation results of a nonlinear pipeline plant, using the models suggested above, are presented. One section of pipeline and the compression plant joined in the end of the pipeline were taken.

The difference equations describing the state variables in dynamics for three compartments are presented:

$$\begin{cases} P_{11}(k+1) = P_{11}(k) + a\left[P_{10}^2(k) - P_{11}^2(k)\right] - b\left[P_{11}^2(k) - P_{12}^2(k)\right] \\ P_{12}(k+1) = P_{12}(k) + c\left[P_{11}^2(k) - P_{12}^2(k)\right] - d\left[P_{12}^2(k) - P_{13}^2(k)\right] \\ P_{13}(k+1) = P_{13}(k) + e\left[P_{12}^2(k) - P_{13}^2(k)\right] - g\left[P_{13}^2(k) - P_{14}^2(k)\right], \end{cases} \tag{48}$$

where $P_{ij}(k)$ are the pressure within pipeline the section $i = 1$ and the compartment $j = \overline{0,4}$ at discrete time k, a, b, c, and d are parameters given a priori.

The compression plant has been described by the nonlinear equation

$$P_{20}^{(d)}(k) = P_{14}(k)\big[u^2(1 + h/P_{14}(k) + 1)\big], \tag{49}$$

where $P_{20}^{(d)}(k)$ is the desired pressure at the beginning of the second pipeline section at time k, u is the control parameter as a relative number of compressor revolution, h is the constant.

The initial and boundary conditions were given as follows

$$\begin{aligned} &P_{11}(0),\ P_{12}(0),\ P_{13}(0) \quad \text{(initial)},\\ &r(k) = P_{10}(k) = A\cos\frac{2\pi}{T}k + A_0 \quad \text{(boundary)}, \end{aligned}$$

on the left-hand side, and $P_{20}^{(d)}(k) = P_{20}^{(d)} = \text{const}$ on the right-hand side (to be more exact at the beginning of the second pipeline section) of the pipeline. Parameters A_0, A, π and T are given.

The pressure $P_{14}(k)$ unchanged in time may be expressed from equations (49) and be substituted into equations (48) as follows

$$P_{14}(u) = \frac{P_{20}^{(d)} - hu^2}{1 + hu^2}, \tag{50}$$

or after a linearisation

$$P_{14}(u) \simeq \widehat{P}_{14}(u) = P_{14}|_{u_0} + P'_{14}|_{u_0}(u - u_0), \tag{51}$$

where $P_{14}|_{u_0} = A$, $P'_{14}|u_0 = B$ are valus of the function and its derivative at the point u_0 of linearisation. The comparison of the two functions is presented in Fig. 6.

A reference model has been described by a system of linear difference equations using (51) as follows

$$\begin{cases} P_{11}^{(r)}(k+1) = P_{11}^{(r)}(k)+a_1\left[P_{10}^{(r)}(k) - P_{11}^{(r)}(k)\right] \\ \qquad\qquad\qquad - b_1\left[P_{11}^{(r)}(k) - P_{12}^{(r)}(k)\right] \\ P_{12}^{(r)}(k+1) = P_{12}^{(r)}(k)+c_1\left[P_{11}^{(r)}(k) - P_{12}^{(r)}(k)\right] \\ \qquad\qquad\qquad - d_1\left[P_{12}^{(r)}(k) - P_{13}^{(r)}(k)\right], \\ P_{13}^{(r)}(k+1) = P_{13}^{(r)}(k)+e_1\left[P_{12}^{(r)}(k) - P_{13}^{(r)}(k)\right] \\ \qquad\qquad\qquad - g_1\left[P_{13}^{(r)}(k) - P_{14}^{(r)}(k)\right] \\ P_{14}^{(r)}(k) = A - B(u-u_0), \quad P_{10}^{(r)}(k) = r(k), \end{cases} \tag{52}$$

where the variable $P_{13}^{(r)}(k)$ is the output one.

Unknown pipeline difference Eqs. 48 – 51 have been modeled according to the OLSL algorithm, where the vector Ω has been defined. The identification error and accuracy versus the number of iterations are shown in Fig. 7. It takes only 60 sec. on the PC-386. It means that for a distributed system such as a pipeline where a dynamic process is very slow, it is possible to control in real time.

The comparison of the reference and the controlled pressure output, control parameter, and error are presented in Fig. 8, 9 and 10, respectively. The gas consumption was taken as a cosine function in time with a maximum at midday (Fig. 8) and minimum at midnight.

The simulation results show that the output of the unknown pipeline tracked well the output of the reference model by orthogonal learning of MISO after some time of adaptation.

6. Conclusions. Scientific literature sources in the control area do not reflect an adaptive control in distributed parameter systems based on a neural network learning paradigm, especially on orthogonal LMS network adaptive learning. Here is the first trial to consider these complex systems from the standpoint of neural network adaptive learning and identification.

An orthogonal least mean squares algorithm based on the linear regression and orthogonal transformations allows to find dynamic neural networks weight vectors and to reduce the learning time maximally. Only in this sense, there is an understanding of a neural network application to solving adaptive control problems in real time.

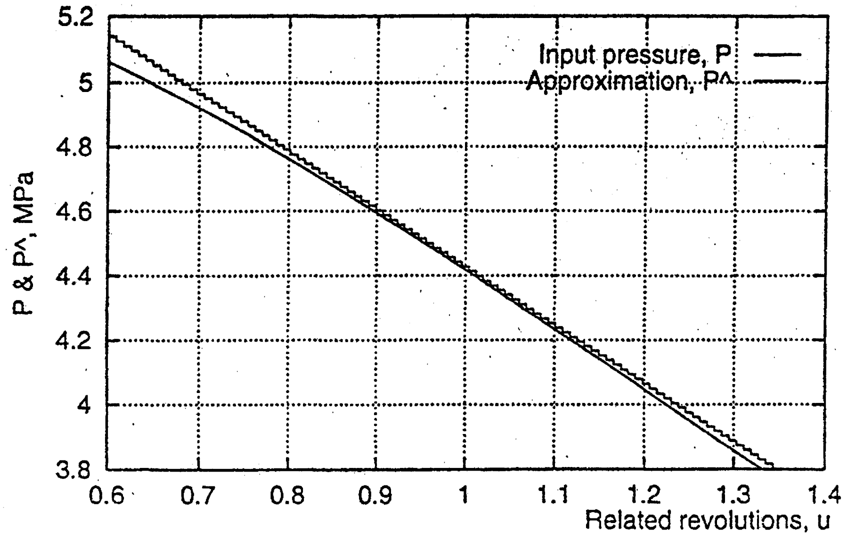

Fig. 6. Linearisation of compression plant relation.

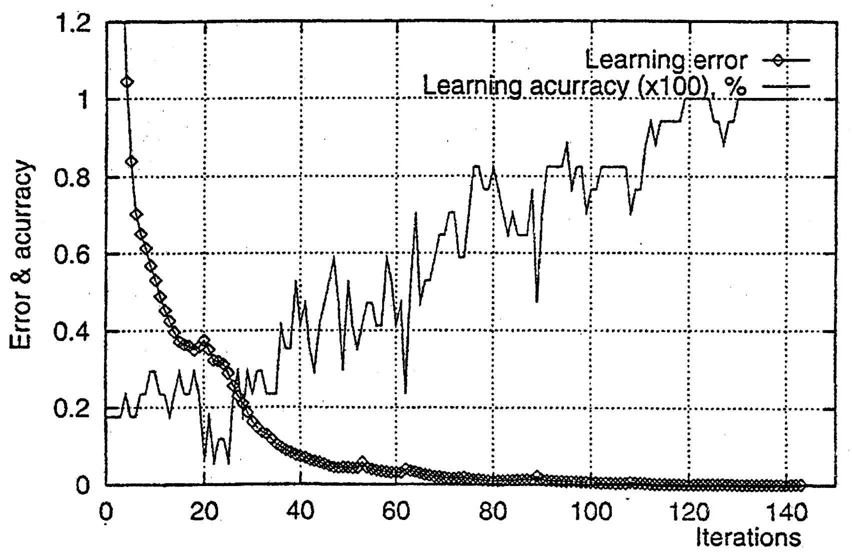

Fig. 7. The identification error and accuracy.

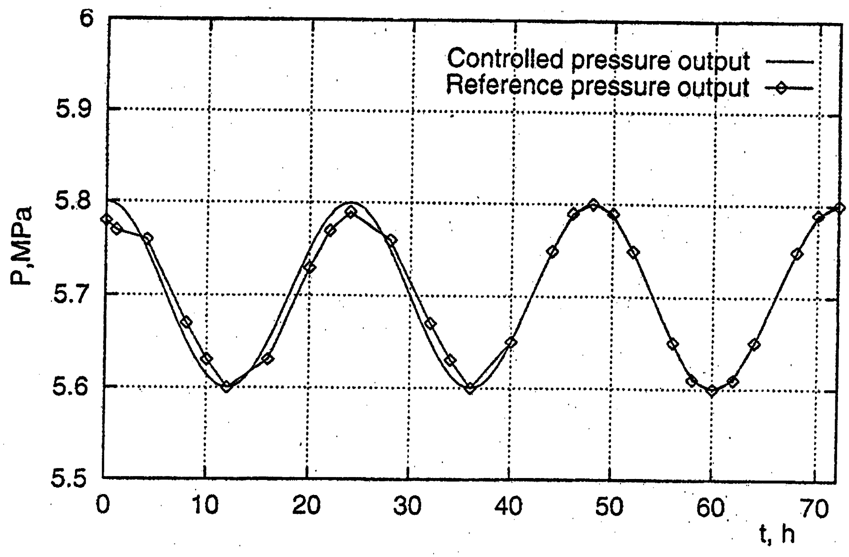

Fig. 8. Controlled and reference pressure outputs.

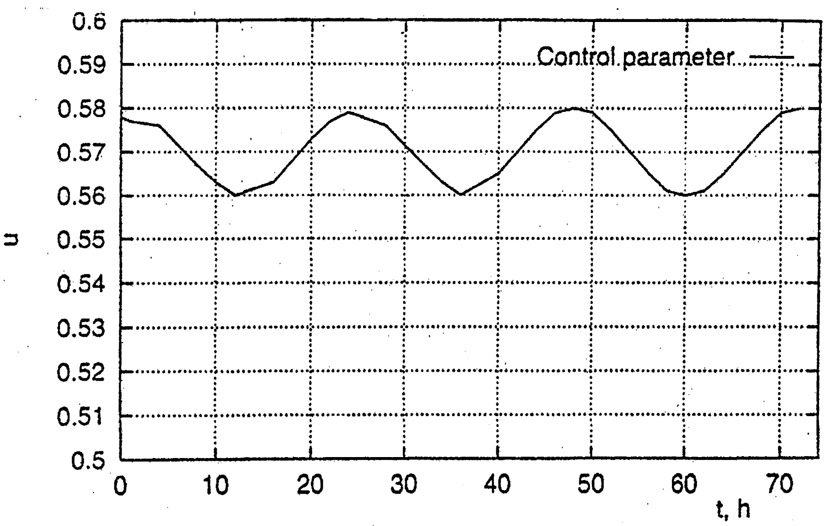

Fig. 9. Control parameter versus time.

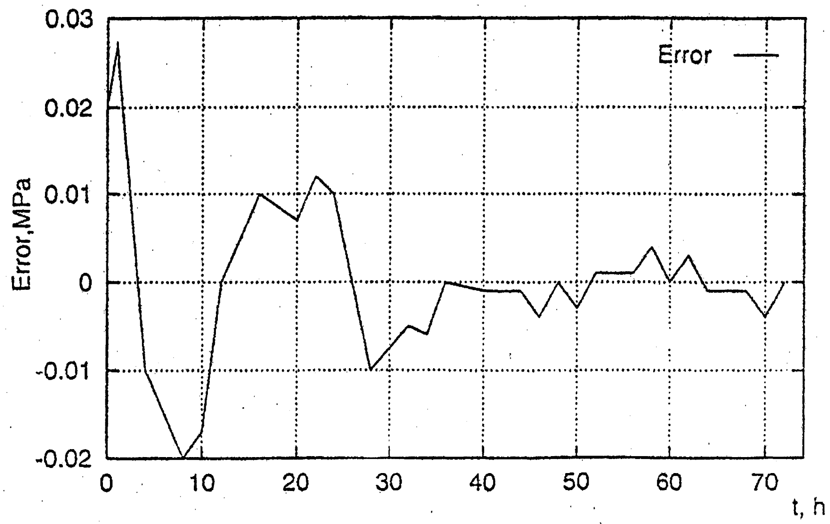

Fig. 10. Error pressure outputs.

The simulation results illustrate that OLSL algorithms with a neural network concept are preferred for perfectly tracked output control parameters during the dynamic control processes and have a good perspective in developing theory and practice of generalised distributed parameter control systems in the future.

REFERENCES

Ahmed, N., and K.R. Rao (1975). *Orthogonal Transform for Digital Signal Processing.* Springer-Verlag, New York.

Åström, K.J. (1970). *Introduction to Control Theory.* AcademicPress, New York.

Björck (1967). Solving linear least squares problems by Gram-Schmidt orthogonalization. *Nordisk Tidskr,* 7, 1–21.

Butkovskiy, A. G., V.A. Kubyshkin and V.I. Finiagina (1990). Methods and models to design mobile controls on surface. In X. Li, J. Yong (Eds.), *Proc. of the IFIP WG 7.2. Working Conf.* pp. 1–13.

Chen, S (1991). Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Trans. on Neural Networks,* 2(2), 302–309.

Feldbaum, A.A. (1965). *Optimal Control Systems.* Academic Press, New York.

Garliauskas, A., and V. Feigin (1989). *Systems Analysis and Optimization of Complex Networks.* Mokslas, Vilnius.

Guangyuan, H., N. Lei and Z. Yaowen (1990). Temperature control system of heat exchangers: an application and theory. In X. Li, J. Yong (Eds.), *Proc. of the IFIP WG 7.2. Working Conf.* pp. 1–13.

Iserman, R. (1974). *Prozeßidentifikation.* Springer, Berlin.

Jin, L., P.N. Nikiforuk and M.M. Gupta (1994). Adaptive control of discrete-time nonlinear systems using recurrent neural networks. *IEE Proc. – Control Theory Appl.,* **141**(3), pp. 169–176.

Landau, I. D. (1980). *Adaptive Control – the Model Reference Approach.* Marcel Dekker, New York.

Nerendra, K.S. (1980). Stable discrete adaptive control. *IEEE Trans. Aut. Control,* **AC–25**.

Nerendra, K. S., and K. Parthasarathy (1990). Identification and control of dynamical systems using neural networks. *IEE Trans. of Neural Networks,* **1**, 4–27.

Parks, P.C. (1966). Lyapunov redesign of model reference adaptive control systems. *IEEE Trans. Aut. Contr.,* **AC–11**, 362–367.

Tsypkin, Ya.Z. (1971). *Adaptation and Learning in Automatic Systems.* Academic Press, New York and London.

Wang, S.S. (1991). LMS algorithm and discrete orthogonal transforms. *IEEE Trans. on Circuits and Systems,* **38**, 949–951.

Widrow, B., and S.D. Stearns (1985). *Adaptive Signal Processing.* Englewood Cliffs, NJ: Prentice Hall.

Received April 1996

A. Garliauskas received the degree of Doctor Technical Sciences from the Computer Center, the Department of the USSR Academy of Sciences, Novosibirsk, USSR, in 1977. He is a head of the Laboratory of Neuroinformatics, Instit. Math. and Inform., The Lith. Republic. His research interest includes neuroinformatics methodology, control problems and development of neural networks learning algorithms, chaos processes.

Madan M. Gupta received the B.Eng and M.Sc. in electronics-communications engineering from the Birla Engineering College (now the BITS), Pilani, India, in 1961 and 1962, respectively. He received the Ph.D. degree from University of Warwick, United Kingdom, in 1967 in adaptive control systems. Dr. Gupta is currently Professor of Engineering and the Director of the Intelligent Systems Research Laboratory and the Centre of Excellence on Neuro-Vision Research at the University of Saskatchewan, Saskatoon, Canada.

VALDYMO SISTEMOS SU PASISKIRSČIUSIAIS PARAMETRAIS IR ORTOGONALIŲ NEUROTINKLŲ MOKYMU

Algis GARLIAUSKAS ir Madan M. GUPTA

Straipsnyje yra pateikti adaptyvių valdymo sistemų su pasiskirsčiusiais parametrais (AVSPP) mokymo algoritmai pagrįsti ortogonalių neurotinklų metodologija. Mes nagrinėjame ortogonalaus mažiausių nuokrypių mokymo algoritmo modifikaciją tam, kad surastume AVSPP problemų sprendimui atitinkamus efektyvius algoritmus. Aptariama dviejų laikų, susietų su objękto valdymo dinaminių procesų realaus laiko ir derinimo parametrų nežinomų pasiskirsčiusių sistemų adaptyviame valdyme mokymo laiko problema.

Modeliavimo rezultatai rodo, kad neurotinklų ortogonalių algoritmų koncepcija suteikia galimybę surasti AVSPP geriausius valdymo išėjimo pasiskirsčiusius parametrus ir turi gerą apibendrintos AVSPP teorijos ir praktikos raidos perspektyvą.