Biological Motivations for Differential Neurocontrol

Frolov, A. A.
1998

INSTITUTE OF COMPUTER SCIENCE

ACADEMY OF SCIENCES OF THE CZECH REPUBLIC

Biological Motivations for Differential Neurocontrol

Alexander Frolov[1], Stanislav Řízek[2], Michel Dufossé[3]

Technical report No. 757
September 1998

Abstract

A new scheme of neurocontrol based on neurophysiological findings in the field of visual-motor coordinations is proposed. It imitates basic control and learning functions in living creatures. The proposed method uses the principle of differential control and local linearization of the nonlinear working space. All control blocks are realized by layered neural networks. Convergence of learning is investigated both analytically and by computer simulations. The developed mathematical model corresponds to neurophysiological concepts of error learning in biological systems.

Keywords

Differential Control, Neurocontrol, Biological Approch, Cybernetical Approach

[1] Institute of the Higher Nervous Activity and Neurophysiology, Russian Academy of Sciences, Butlerova 5a, 117 865 Moscow, Russia

[2] Institute of Computer Science, Academy of Sciences of the Czech Republic Pod Vodárenskou věží 2, 182 07 Prague 8, Czech Republic

[3] Laboratory CREARE, Univ. P. & M. Curie, case 23, F-75 252 Paris cedex 05, France

1 Introduction

Motor learning in biological systems is performed by modifying the flows of signals transmitted between neural structures, mainly the motor areas of the cerebral cortex, the cerebellum, and the basal ganglia. According to modern control theory, learning requires not only an adaptable system but also the possibility of changing the information processing rules [13].

The cerebral level, which is the first to operate, is able to compensate for any weakness in spinal mechanisms by producing responses which are better adapted to disturbances caused by the environment. When the task is performed repeatedly, the cerebellum is able to deal with some of the repetitive aspects so that the motor response becomes more finely attuned and automatized. The cerebellum can thus free the cerebral cortex during sensori-motor or even mental tasks [15]. The basal ganglia nuclei are more involved in postural stabilization during elementary movements and in optimal de-stabilization between successive elementary movements. Both the cerebellum and basal ganglia are involved in motor control, motor planning and cognitive aspects of action, helping the cerebral cortex.

Artificial neural networks are widely used for control of various complex non-linear systems [2]. They promise a good solution for most of the control problems. Learning ability is one of their main advantages, and special learning algorithms provide rather good convergence. They do not require precise initial mathematical models that can be developed during the adaptation process. Generalization properties may ensure solving such situations in the operation phase that have not been trained in the learning phase. Physical implementations of neural networks offer the advantage of a massive parallelism, that provides a high speed of information processing under the possible low speed function of single processing elements in the operation phase. It may also provide a significant fault tolerance since the damage of several processing elements may not significantly impair the performance of the whole neural network.

The problem of neurocontrol has been addressed for several years, but still new architectures are being investigated [17], [12]. Neurophysiological findings, and the knowledge of functioning of biological neural structures, can aid in the design of new control schemes with improved characterictics. In the case of biomechanical systems, the problem can be solved more efficiently when using control by velocity instead of control by position. This principle has been generalized for control of arbitrary systems, and implemented in the differential neurocontroller proposed in [22], [8].

The paper summerizes recent neurophysiological findings, describes the proposed scheme of differential neurocontrol, analyzes the learning characteristics and presents the results of computer simulations. Its aim is to investigate the correspondence between mathematical models of control and neurobiological concepts stated by neuroscientists in visual-motor control.

2 Neurophysiological Basis

Living creatures have solved the problem of movement control rather successfully. They are able to learn complicated movements and to coordinate their complex muscle systems. This ability is supported by the complex structure of the brain that performs individual control functions in its specialized areas. Utilization of neurophysiological knowledge in the design of controllers supposes a detailed analysis of functioning of visual - motor systems.

Many neurophysiological studies have been devoted to analysing the relationships between neuronal activity and arm movement kinematics. These studies have been performed in the associative, premotor and primary motor areas of the cerebral cortex [5], [6], [10], [11] and in the cerebellum [7]. The authors, using two- and three-dimensional reaching tasks, have shown that when a monkey makes arm movements in various direction towards visual targets in space, the cell activities vary in an orderly fashion with the direction of the movement. The description of the broad directional tuning of cortical cells around their preferred direction (the direction of movement in which they discharge most during the reaction time period) led to directional information being treated like a population code, as opposed to a single cell code. This has provided a useful tool for interpreting directional relations, not only in various regions of the distributed motor system but also in associative areas dealing with visual information processing.

The importance of directional information in determining neuronal activity raises related questions concerning the coordinate system used by the frontal cortex to represent the direction of arm movements. In fact, muscle commands and sensory information are not mapped in the same reference frame and need to be correlated with the information signalling positions of arm segments relative to the body. This information comes from several sources, motor, somesthetic and proprioceptive. When the initial arm position changes, the body-centered information about the target position is not invariant in an arm-centered coordinate system, and has to be combined with information about the initial position of the arm in order to compute the appropriate motor commands.

All the results obtained upon recording motor cortex cells are consistent with the following statements:

1. Motor cortical cells command muscle synergies, which can be represented by a vector, the "cell's preferred direction", corresponding to the overall effect of the motor command on the arm and hand positions.

2. The orientation of this synergy vector is invariant with respect to an arm centered reference system. Consequently, it does not remain constant with respect to an extrapersonal coordinate system, but rotates with the initial position of the arm in space.

3. Cortical commands sent to the shoulder and elbow joint muscles will change the arm position in a constant way within this arm-centered coordinate system.

4. The computation of the appropriate motor command, taking the cell's preferred directions into account is performed in the early phase of the reaction time, before the onset of the movement. This therefore results from a combination performed by cortical areas before the movement is initiated, without requiring a feedback loop when the movement is actually performed.

Neurons in the visual system are selectively sensitive to motion in specific directions called the preference directions. It can be supposed that the three-dimensional external coordinate system is transformed by some neural network into an n-dimensional internal system of preference directions. In general, the visual system is more sensitive to motion velocity than to object location.

By contrast to the cerebral cortex, the cerebellar cortex is a mosaic structure, the whole surface of which consists of thousands of microzones, working idependently without any global attentional supervisor. Cerebellar microzones are the basic units of the cortex of the cerebellum. Together with their output nuclear projecting zones, they form the so-called "cortio-nuclear microcomplex" ([13]). Climbing inputs from specific parts of the inferior olive longitudinally organize the cerebellar cortex ([18]). A second input is provided by the mossy fiber system, which is characterized by its considerable divergence onto wide regions on the cerebellar cortex. The set of mossy fiber inputs constitutes a general context about the present sensorimotor actions and intents for future movements, that is a large set of signals providing information about the states of activity in various nervous structure, from command structures to more sensory structures.

Each microzone can be viewed as a three layer neural network [1], [16]: an input layer formed by cells which originate from the mossy fibers, an intermediate layer of granular cells, and an output layer of Purkinje cells that project to cerebellar output nuclei. Two pathways, one cortical, the other via subcortcal nuclei transmit this information to the spinal cord. Since plastic synapses only exist between the second and third layers, it is similar to the original perceptron described by Rosenblatt [21]. During the adaptive phase, the output layer utilizes an error signal conveyed by the climbing fibers, originating from a teacher, the inferior olive nucleus. Each output Purkinje cell receives one and only one climbing fiber. The long term effect of this error signal is a decrease in the synaptic efficiency between parallel fibers, axons of granular cells, and the output Purkinje cells, whenever the parallel fiber activity is correlated with the error signal [1], [9], [13], [16].

An important feature of the cerebellar design is the great number (10^{11}) of granular cells in the intermediate layer, which is of the same order of magnitude as the total number of cells in the nervous system. The role of this architecture is to provide an extended set of new combinations of inputs which are needed for bypassing the mathematical limitations of the classical two-layer perceptron for learning any arbitrary input/output function [16]. The functional role of each microzone is then defined by the climbing fiber error signals which originate from a restricted part of the inferior olive.

A baby typically learns to coordinate its muscles first by generating small random motor signals and by observing the results. Then, the baby learns more complicated movements in order to reach the selected target while minimizing the visually controlled errors [19]. Some individuals learn by fast and chaotic trials with serious errors; it looks like the feedback, performed by cortical neurons, is weak and the control is rather unsmooth. Others learn by slow anxious movements, when the feedback is strong and keeps current errors low [23].

These neurophysiological findings enable us to design a simple model of the visual-motor system, especially for arm movement control. Moveing the working point (usually the hand) from the initial position to the target point in the external working space. The whole system performs successive transformations of input visual signals into output motor signals.

The procedure starts with the specification of the target. The current state of the system and the initial position of the working point are supposed to be known. Both initial and target points in the external space are sensed by the visual system and transformed into the internal visual space of the brain, where the reference trajectory is generated. The reference trajectory is permanently compared with the current trajectory of the working point and the discrepancy is used for corrections and modifications of succeeding transformations. Since neurons of the visual-motor system are selectively sensitive to the movement direction, we suppose that the brain processes the velocity rather than the position.

The reference trajectory in the internal visual space of the brain is transformed into the internal motor space related to the body geometry. Then it is transformed into muscle forces and into torque moments that affect control angles in joints of the multilink arm. Changes of the control angles result in real kinematics of the working point and realize the required movement. All these transformations are in principle nonlinear. The preference direction neurons are activated only in local areas of the visual space relatively to the body geometry [4]. Only a limited set of active neurons contributes to forming the transformation function in every local area of the working space. This causes the fundamental nonlinearity of the visual-motor system.

It is generally accepted that the transformations from the internal visual space into the internal motor space, and then into the muscle forces space, are performed in the cerebellum, while corrections of transformations are performed in the motor areas of the neurocortex [13]. Thus, unfamiliar movements, for which the transformations have not yet been formed precisely, are first of all controlled by the neurocortex. On the other hand, the skilled movements are mainly controlled by the cerebellum and the motor area of the neurocortex is made free from current movement corrections [3], [4].

There are several types of redundancy in the visual-motor system. First, the reference trajectory of the working point from the initial position to the target can be generated in many ways with various velocity profiles. Further, the same trajectory in the internal visual space of the brain can be related to the variety of trajectories in the internal motor space of the brain because of many degrees of freedom in the system (body, arm). Finally, every actual movement can be performed with the variety of the muscle forces, as the number of muscles is higher than the number of control angles in joints. The inherited redundancies in the visual-motor system are overcome in living creatures by learning.

The afore mentioned neurophysiological knowledge has been used in the design of the differential neurocontroller. Its basic blocks resemble individual parts of the brain and the proposed scheme simulates both learning and operation procedures in living creatures.

The neurocontrol model described below takes into account the above presented findings on the cooperation between different brain structures during the learning and execution of goal directed movements.

3 Scheme of the Differential Neurocontroller

The proposed scheme of the differential neurocontroller is shown in Figure 1.
It involves: F - feedforward control block, B - feedback control block, R - movement planner, I - integrator, P - plant. These blocks process the following vector signals: $x^a, x^c(t)$ and $x^d(t)$ are the target, current and desired outputs of the plant; $x^\Delta(t) = x^d(t) - x^c(t)$ is the discrepancy between the desired and current plant output; $\dot{x}^d(t)$ is the time derivative of the desired output, i.e. the desired output velocity; $c(t)$ is the control vector and $\dot{c}(t)$ is its time derivative; $u_B(t)$ and $u_F(t)$ are the partial control vectors generated by the control blocks B and F; ee is the estimated control error; t is the time parameter. The learning coefficient α and the feedback coefficient γ are the global free parameters of the control scheme; they influence the learning process. It should be mentioned that the control signal c defines uniquely the state of the plant and determines its output coordinates.

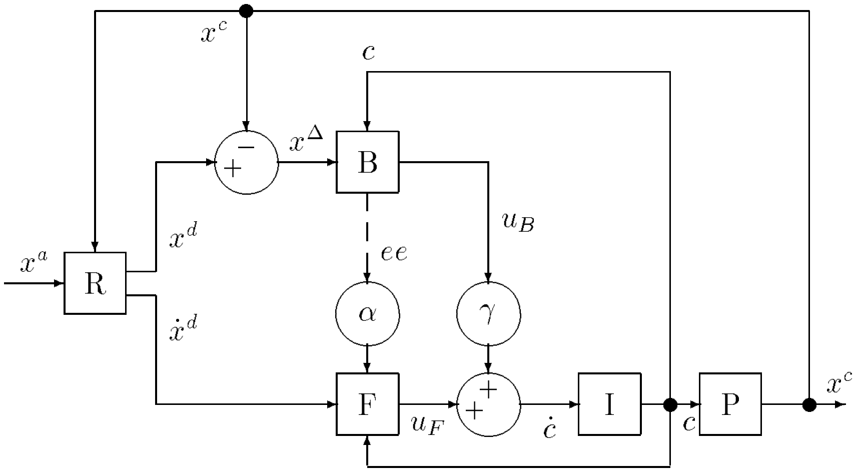

Figure 1. *Scheme of the differential neurocontroller*

The signals are processed according to the relations:

$$x^c = P(c) \tag{3.1}$$

$$\dot{c} = u_F + \gamma \cdot u_B \tag{3.2}$$

$$c = \int \dot{c} \cdot dt \tag{3.3}$$

$$u_B = B(c) \cdot x^{\Delta} \tag{3.4}$$

$$u_F = F(c) \cdot \dot{x}^d \tag{3.5}$$

The integration block I might be involved in the plant. When the neurocontroller works in discrete time, the integration is easily realized by summing the successive additions.

The movement planner R provides the desired movement parameters through time, i.e. the plant output and its time derivative. It generates the desired trajectory e.g. according to the relation

$$\dot{x}^d = \frac{1}{\tau} \cdot (x^a - x^d) \tag{3.6}$$

where τ is the time constant of the desired changes of the plant output. Then the desired movement is a straight line in the plant output space with the exponential decrease of the velocity. It can be realized by a neural network containing one linear neuron and one itegrator for each coordinate [22].

The plant P is essentially nonlinear, so that it cannot be described by a linear model within the whole working space. The control blocks B and F must also be nonlinear, but they can be linearized with respect to local changes of their input signals x^{Δ} and $\dot{x}^d$. The proposed scheme with the differential approach performs it by using layered neural networks with multiplicative neurons.

The simple scheme of the differntial neurocontroller in Figure 1 is sufficient if the time constants of all transient processes in the plant are much less than the time constant of the movement planner τ. Otherwise, the inercity of the plant must be taken into account and some inertial neurocontrol blocks must be involved in the scheme [20].

The time derivative of the plant output (3.1) is given by

$$\dot{x}^c = J(c) \cdot \dot{c} \tag{3.7}$$

where $J(c)$ is the Jacobian of the plant. Hence, by using (3.2) to (3.7), the system can be described by the equation

$$\dot{x}^c + \gamma \cdot J(c) \cdot B(c) \cdot x^c = \gamma \cdot J(c) \cdot B(c) \cdot x^d + J(c) \cdot F(c) \cdot \dot{x}^d \tag{3.8}$$

Its solution is stable if the product $J(c) \cdot B(c)$ is a positive definite matrix. It can be reached by the proper design of the block B so that $B(c) \sim J^T(c)$. It is evident that the high performance of the neurocontroller (i.e. $x^c \doteq x^d$) may be achieved if the product $J(c) \cdot F(c)$ tends to the unit matrix.

4 Multiplicative Units

Transfer functions (3.4) and (3.5) include an elementwise multiplication of vectors. This means that only the corresponding elements having the same index are multiplied. It can be realized by the neural network with higher order neurons according to Figure 2. It consists of a linear subnetwork L processing the plant output signal x, of a nonlinear subnetwork N processing the control signal c and of a layer of multiplicative units H. Every unit involves three neural elements, a linear input neuron h_L, a nonlinear input neuron h_N and a multiplicative output neuron h_M. The linear and nonlinear neurons are activated in the standard way

$$h_L = W_L \cdot x \tag{4.1}$$

$$h_N = \sigma(\eta) = \sigma(W_N \cdot c + \theta_N) \tag{4.2}$$

where W_L and W_N are the matrices of synaptical weights of subnetworks L and N, θ_N is the threshold vector and σ is a nonlinear function, e.g. radial basic function, hard limit function, or the sigmoidal function according to

$$\sigma(\eta) = \frac{1}{(1 + e^{-k\eta})} \tag{4.3}$$

where k is the scale constant.

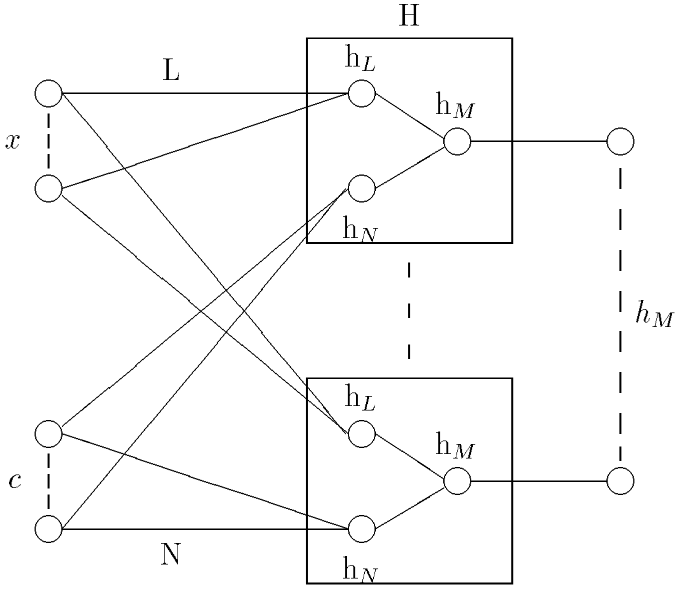

Figure 2. *Multiplicative units*

The set of linear neurons h_L is activated only from the input x and the set of nonlinear neurons h_N is activated only from the input c. The set of higher order multiplicative neurons h_M is activated by the elementwise multiplication

$$h_M = h_L \otimes h_N \tag{4.4}$$

Hence, the output vector h_M is a linear function of the input vector x and a non-linear function of the input vector c. Some examples of neural implementations of the multiplication process can be found in [4].

5 Feedback Block

The feedback block B inputs the discrepancy between the desired and current plant output and realizes the transfer function (3.4). It corrects the plant output errors and ensures the stability of the system, which is especially important in the initial stage of the neurocontroller learning. It also estimates the control errors of the feedforward block F, which are used in the learning procedure. These errors are not expressed explicitely in the scheme and they must be estimated. The feedback block B learns by Hebbian rule. It ensures that the plant output will reach the target position x^a. Its influence gradually decreases as the function of the neurocontroller improves. The feedback block B can be implemented in a three-layer neural network according to Figure 3. It involves three subnetworks B$_O$, B$_L$, and B$_N$ with corresponding matrices of synaptical weights B_O, B_L, and B_N. The hidden layer H consists of a set of multiplicative units that provide the elementwise multiplication defined in (4.1) to (4.4).

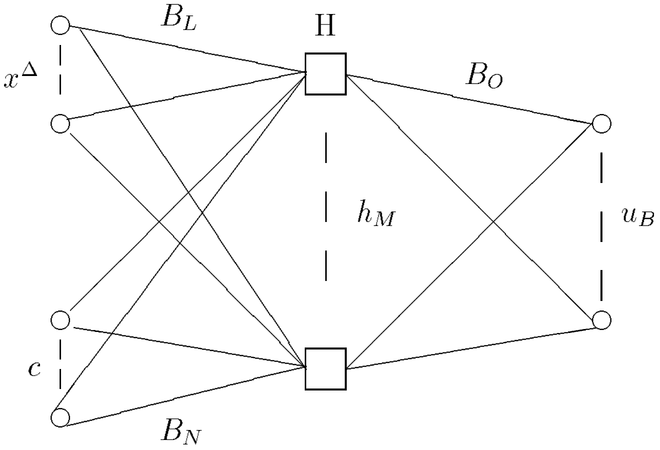

Figure 3. *Implementation of the feedback block*

The linear subnetwork B$_L$ implements the linear transformation of the vector x^Δ to the hidden layer. It transforms the three-dimensional external space into the internal space of higher dimensionality with linearly dependent coordinates. The transformed cordinates are related to the preference directions, i.e. to the movement directions in the plant output space that activate corresponding neurons the most intensively. Preference directions can be uniformly distributed in the external space, and the matrix B_L can be set random.

The nonlinear subnetwork B$_N$ processes the control vector c. It causes that different groups of hidden units are activated in different subspaces of the output plant space, which are uniquely determined by the current control signal c. Such a property of selective activation of motor control neurons has been revealed in the case of arm target movements in a monkey [5]. The matrix B_N may be set random as well.

The output subnetwork B$_O$ performs the linear transformation. The matrix B_O can be gained by learning according to the Hebbian rule. The complete feedback block B should learn the transposed Jacobian $J^T(c)$. Its learning must be performed in the first stage of the design, when the function of the block F does not need to be taken into account.

The learning of the block B can be performed as follows: Randomly generated small differences of the control signal Δc cause the corresponding small changes of the plant output Δx. These values describe the responses of the plant to the disturbances Δc. The Hebbian learning proceeds according to the equation

$$B(c) = \sum \Delta c \cdot \Delta x^T \tag{5.1}$$

and by using a difference form of (3.7)

$$B(c) = \sum \left(\Delta c \cdot \Delta c^T \cdot J^T(c)\right) \tag{5.2}$$

When the disturbances Δc are generated randomly with the uniform distribution, the linearized transfer function $B(c)$ is

$$B(c) \simeq K \cdot J^T(c) \tag{5.3}$$

where the constant K may be normalized to the required value. For a non-linear case and fixed random matrices B_L and B_N, the Hebbian learning of the output matrix B_O gives

$$B_O \sim \sum \Delta c \cdot \Delta h_M^T(c) \tag{5.4}$$

where h_M is given by (4.1) to (4.4). The nonlinear subnetwork B$_N$ must ensure that only a small portion of hidden neurons is activated in every subspace of the control signal c. An increasing number of hidden units may improve the approximation properties of the block, and may model the nonlinear Jacobian more accurately. The approximation can be made precise for linear and quadratical plants [8].

6 Feedforward Block

The feedforward block F inputs the time derivative of the desired plant output $\dot{x}^d$ and realizes the transfer function (3.5). It learns according to the estimated control error ee that is obtained from the feedback block B. The feedforward block F ensures that the plant output will follow the prescribed trajectory with the prescribed time derivative. Its influence increases with the improving function of the neurocontroller. The feedforward block F is implemented in a three-layer neural network according to Figure 4. It contains four subnetworks F$_D$, F$_O$, F$_L$, and F$_N$, with corresponding matrices of synaptical weights F_D, F_O, F_L, and F_N. The hidden layer H consists of a set of multiplicative units introduced above. The subnetworks F_O, F_L, and F_N, have the same structure as the subnetworks B_O, B_L, and B_N, of the feedback block B. However, they process other signals and learn by other algorithms.

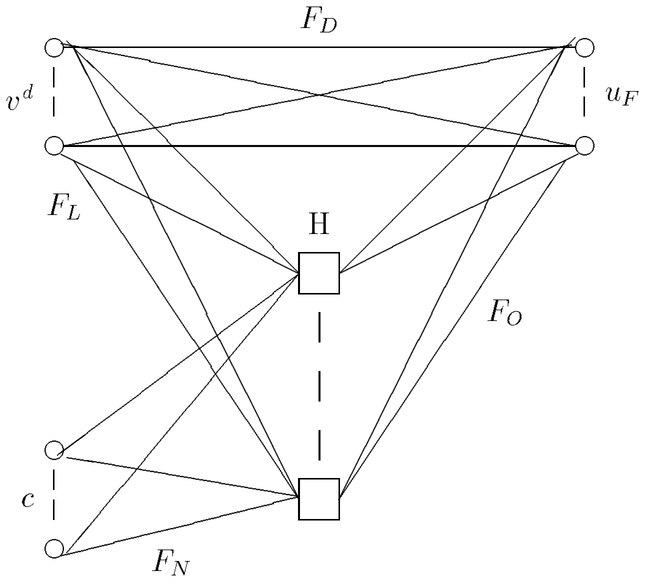

Figure 4. *Implementation of the feedforward block*

The linear subnetwork F_D transfers the input vector $\dot{x}^d$ directly to the output of the block F. It considerably improves the learning convergence of the whole neurocontroller, while the nonlinear part of the block ensures small resulting errors.

Learning of the feedforward block F requires the use of the block B, which must be learned in advance, and which aids in the estimation of the output control errors of the block F. The supervised learning is based on the error back propagation algorithm which minimizes the output errors by the steepest descent method. However, the output control error $err_{\dot{c}}$ is not expressed explicitly in the scheme, and it must be estimated. It is known from the back-propagation learning that the plant output error is transferred backward to the plant input through multiplication by the transposed plant matrix [14]. In the case of the differential control, the error of the plant output derivative $err_{\dot{x}}$ is back propagated through the plant according to the relation $err_{\dot{c}} = J^T(c) \cdot err_{\dot{x}}$, where $err_{\dot{x}} = \dot{x}^d - \dot{x}^c = \dot{x}^\Delta$. Derivative of (3.4) and (5.3) give $\dot{u}_B \simeq \beta \cdot J^T(c) \cdot \dot{x}^\Delta$, where the constant β specifies the gain of the feedback loop. Therefore, the control error can be estimated by $ee \sim \dot{u}_B$.

In the learning process, targets are chosen in the working output space of the plant, and the control problem is solved through time for sequences of random trials. The error back propagation algorithm leads to the changes of individual synaptical weights according to the negative gradient. For example, in the case of the sigmoidal nonlinear function, it results in the following learning equations determining the time derivatives of individual matrices of synaptical weights

$$\dot{F}_D = \alpha_D \cdot \dot{u}_B \cdot (\dot{x}^d)^T \tag{6.1}$$

$$\dot{F}_O = \alpha_O \cdot \dot{u}_B \cdot h_M^T \tag{6.2}$$

$$\dot{F}_L = \alpha_L \cdot \left(\left(F_O^T \cdot \dot{u}_B\right) \otimes h_N\right) \cdot (\dot{x}^d)^T \tag{6.3}$$

$$\dot{F}_N = \alpha_N \cdot \left(\left(F_O^T \cdot \dot{u}_B\right) \otimes h_A\right) \cdot c^T \tag{6.4}$$

$$\dot{\theta}_M = \alpha_M \cdot \left(F_O^T \cdot \dot{u}_B\right) \otimes h_A \tag{6.5}$$

where $h_A = h_L \otimes h_N \otimes (1 - h_N)$; h_L, h_N, h_M are given by (4.1) to (4.4) for the linear input $\dot{x}^d$ instead of x. The parameters α are the learning coefficients for the corresponding subnetworks. The initial states of the matrices F_D, F_O, F_L, F_N and the vector θ can be arbitrary, but the matrices F_L and F_N must not be zeroes.

7 Convergence of Learning

The plant P is generally a nonlinear system, and the neurocontroller should be a nonlinear network as well. However, the linear analysis can aid in the estimation of the local behaviour of the structure. Let the control scheme be linearized in a certain reference point of the plant output space defined by the chosen control signal c. We can suppose that the nonlinear transfer h_N (4.2) of the control signal c to the hidden layer H of the block F is constant in a limited area around this reference point. It causes that only a small portion of the hidden neurons are activated in the dependence on the selected subspace. Its influence upon the complete transfer function $F(c)$ can be involved in the linear transfer h_L (4.1). Hence, the matrix F_N can be set zero and the transfer functions of all blocks can be described by matrices. The equations (3.1), (3.4), (3.5), (3.7), and (5.3) are simplified to $x = P \cdot c$, $J = P$, $B = \beta \cdot P^T$, $u_B = B \cdot x^\Delta$ and $u_F = F \cdot \dot{x}^d = F_O \cdot F_L \cdot \dot{x}^d$. The direct transfer matrix F_D is zero; the linear input matrix F_L is set random and it is not adapted during the learning. The output matrix F_O must be taught in the learning process.

Let us introduce the matrix variable Φ as a measure of quality of the neurocontroller performance

$$\Phi = P \cdot F - E = P \cdot F_O \cdot F_L - E \tag{7.1}$$

where E is the unit matrix. The learning equation for the output matrix F_O can be derived from (3.8), (6.2), and (7.1); for the removed feedback (i.e. $\gamma = 0$), it is

$$\dot{\Phi} + \alpha \cdot \beta \cdot P \cdot P^T \cdot \Phi \cdot \dot{x}^d \cdot (\dot{x}^d)^T \cdot F_L^T \cdot F_L = 0 \tag{7.2}$$

The detailed analysis of the learning convergence was presented in [23] for $F_L = E$. If the change of the subnetwork F_O is small in every trial of the learning process, then the learning error decreases in average in k succeeding trials exponentialy according to

$$|\Psi_k| \simeq |\Psi_0| \cdot exp(-k \cdot \rho) \tag{7.3}$$

where $|\Psi_0|$ is the starting value of Ψ; $\Psi = U^T \cdot \Phi \cdot U$; U is the matrix of eigenvectors of the matrix product $P \cdot P^T$. The average learning rate ρ can be expressed for small α as

$$\rho = \frac{\alpha' \cdot \lambda \cdot < |V_k|^2 > \cdot \tau}{2 \cdot n_x \cdot (1 + \gamma' \cdot \lambda \cdot \tau)} \tag{7.4}$$

where $\alpha' = \alpha \cdot \beta$; $\gamma' = \gamma \cdot \beta$; λ is the minimal eigenvalue of the matrix product $P \cdot P^T$; n_x is the dimensionality of the plant output space; $< |V_k|^2 >$ is the average squared value of the initial velocity in k trials.

The presented analysis is also valid for $F_L^T \cdot F_L = E$. This can be reached with a good accuracy if the number of the hidden neurons (and therefore the dimensionality of the random matrix F_L) is high enough. The dimensionality can be lowered if the matrix F_L is set orthonormal. This can be made easily for one selected subspace of the control signal c, but it cannot be done simultaneously for all possible subspaces.

It can be seen from (7.4) that under some conditions the learning rate ρ increases with increasing learning coefficient α and decreases with increasing feedback coefficient γ. These relations were verified by computer simulations.

8 Computer Simulations

The dependence of the learning rate on the learning and feedback coefficients was studied in numerical experiments. The computer model simulated the plant with 6 inputs and 3 outputs. The computation was performed for various values of coefficients α' and γ'. The block F was tought in series of trials, every series consisting of 10 movements to random targets. The integral error of each series was normalized to the size of the working space and to the output dimensionality.

The influence of the learning coefficient α' upon learning is presented in Figure 5 for the case $\gamma' = 0$. The figure shows the logarithm of the total output error in dependence on the number of trial series, while α' is the parameter of the graphs. The learning rate increases with increasing α' up to a certain saturation limit. The numerical stability requires that the integration time step must decrease with increasing α'. Hence, the real time of learning increases for larger α' and it is useless to increase α' up to the saturation value.

The influence of the feedback coefficient γ' upon learning is demonstrated in Figure 6 for the case $\alpha' = 0.1$. The figure shows again the logarithm of the total output error in dependence on the number of trial series, while γ' is the parameter of the graphs. For all values of α' and γ' the learning rate is quite

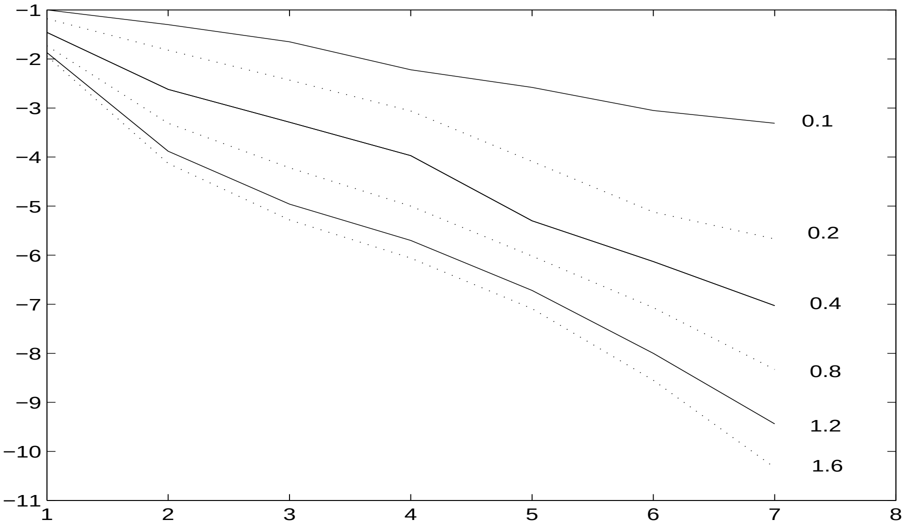

Figure 5. *Influence of the learning coefficient* α'

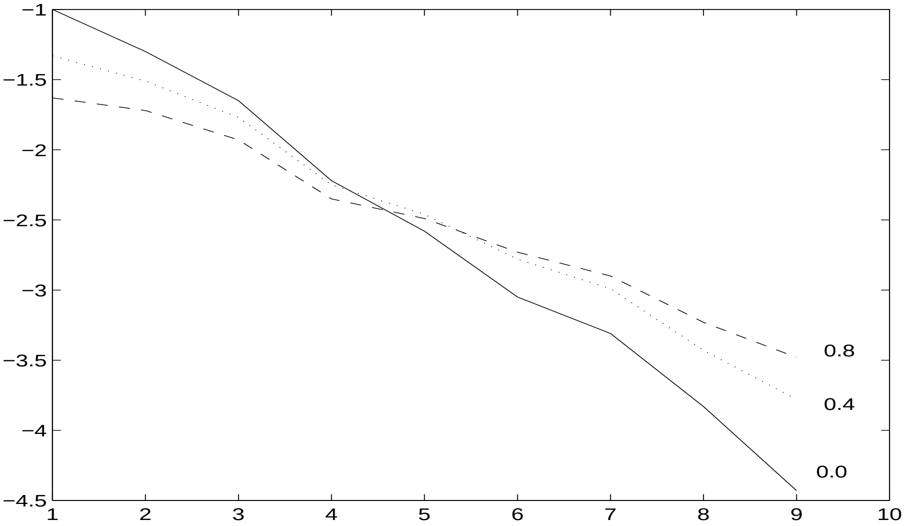

Figure 6. *Influence of the feedback coefficient* γ'

well described by Eqn. (7.4) until $\rho < 0.1$. The saturation of ρ occurs in this case at about 0.17.

Computer simulations show that the initial value of the total output error decreases with increasing γ', but the learning rate decreases as well. The strong feedback in initial trials causes the smooth movement of the plant and small output errors. On the other hand, the weak feedback in latter trials makes it possible to reach the high rate of convergence. The combination of both of these features can ensure small output errors together with a high rate of convergence of learning.

The accuracy and the rate of learning in the linear case do not depend on the dimensionality of the hidden layer of the block F. Practically, 10 hidden neurons are sufficient for the scheme with 6 control signals and 3 plant outputs. However, the nonlinear structure requires rather high number of hidden units. A good performance of the block F may be achieved only when the activities of the hidden neurons are rather sparse. Every hidden neuron can be set active only in a small area of the control signal space, where the structure can be linearized. On the other hand the number of active hidden neurons must be large enough to ensure that the product $F_L^T \cdot F_L$ is almost a diagonal matrix. Otherwise the learning process according to (7.2) could be spoiled. Hence, rather high dimensionality of the hidden layer is required for a good performance of a highly nonlinear structure. For 6 control signals, 3 plant outputs and 100% quadratic nonlinearity, at least 100 hidden neurons were necessary while the number of active neurons was in average 20. These numbers may seem to be rather high for such a small number of the control signals. However, they need not grow so rapidly with growing dimensionality of the control signal space.

9 Conclusions

The described structure of the differential neurocontroller has several advantages. The control by differences causes the architecture to be relatively simple. The control system is nonlinear with local linearization. It results in a good convergency and accuracy of the learning process. The developed procedure of the design does not require any calculation of a matrix inversion. The transfer function of the plant need not be explicitly known. Learning can be performed by using the real plant.

The behaviour of the complex nonlinear system differs from the linear case for a larger nonlinearity of the plant working space. Then the control blocks must also be nonlinear and their learning can only be estimated by the linear case. Nonlinear control requires high dimensionalities of the hidden layers of the neural control blocks. The number of hidden neurons might be reduced if the random linear input matrix is made orthonormal. However, this condition can be fulfilled only in several points of the control signal space. Moreover, the orthonormalization is not a biologically plausible operation. The improvement of the organization of activation of hidden neurons might bring better results.

Acknowledgement

This paper was supported by the Grant Agency of the Czech Republic through the Grant No.201/95/0992, project INTAS-RFFR No.95-1327 and project BARRANDE No.970109.

Bibliography

[1] Albus J.: A Theory of Cerebellar Function. Math. Biosci., 1971; 10, 25-61.

[2] Bekey G.A., Goldberg A.Y. (eds.): Neural Networks in Robotics. Kluwer Academic 1992.

[3] Burnod Y., Dufossé M.: A Model for the Co-opearation between Cerebral Cortex and Cerebellar Cortex in Movement Learning. In: Brain and Space, Paillard (Ed.) Oxford Univ. P, 1990.

[4] Burnod Y., Grandguillaume P., Otto I., Ferraina S., Johnson P.B., Caminiti R.: Visuomotor Transformation Underlying Arm Movements toward Visual Target: A Neural Network Model of Cerebral Cortical Operations. Journal of Neuroscience, 1992; 12, 1435-1453.

[5] Caminiti R., Johnson P.B., Galli C., Ferraina S., Burnod Y., Urbano A.: Making Arm Movements within Different Parts of Space: The Premotor and Motor Cortical Representation of a Coordinate System for Reaching to Visual Targets. Journal of Neuroscience, 1991; 10, 2039-2058.

[6] Evarts E. V.: Pyramidal tract activity associated with a conditioned hand movement in the monkey. Journal of Neurophysiol., 1966; 29, 1011-1027.

[7] Fortier P.A., Kalaska J.F., Smith A.M.: Cerebellar Neuronal Activity Related to the Whole-Arm Reaching Movements in the Monkey. J. Neurophysiol., 1989; 62, 198-211.

[8] Frolov A., Řízek S.: Differential Neurocontrol of Multidimensional Systems. In: Dealing with Complexity: A Neural Network Approach. (Ed. Kárný M., Warwick K., Kůrková V.), Springer, London, 1998, 238-251.

[9] Fujita M.: Adaptive Filter Model of the Cerebellum. Biol. Cybern., 1982; 45, 195-206.

[10] Georgopoulos A.P., Schwartz A.B., Kettner R.E.: Neuronal Population Coding of Movement Direction. Science, 1986; 233, 1416-1419.

[11] Georgopoulos A.P., Kettner R.E., Schwartz A.B.: Primate Motor Cortex and Free Arm Movement to Visual Targets in Three-Dimensional Space. II. Coding of the Direction of Movement by a Neuronal Population. J.Neurosci., 1988; 8, 2928-2937.

[12] Henaff P., Delaplace S.: Backpropagation without Output Desired: Implementation of Neural Control in Mobile Robotics. In: Proc. of the Conference on Neural Networks and their Applications, Marseille 1996, 69-76.

[13] Ito M.: The Cerebellum and Neural Control. Raven Press, New York, 1984.

[14] Jordan M.I.: Generic Constrains on Underspecified Target Trajectories. Proc. of IJCNN, New York IEEE press, 1989, 217-225.

[15] Leiner H.C., Leiner A.L., Dow R.S.: Reappraising the Cerebellum: What does the Hindbrain Contribute to the Forebrain. Behav. Neurosci. 103, 1989; 5, 998-1008.

[16] Marr D.: A Theory of Cerebellar Cortex. J.Physiol. (London), 1969; 202, 437-470.

[17] Martinetz T., Ritter H., Schulten K.: Learning of Visuomotor-Coordination of a Robot Arm with Redundant Degrees of Freedom. Parallel Processing in Neural Systems and Computers, Eckmiller R. at all (Eds.), Elsevier Science Publishers, North Holland, 1990, 431-434.

[18] Oscarsson O.: The Sagittal Organization of the Cerebellar Anterior Lobe as Revealed by the Projection patterns of the Climbing Fiber System. In: Neurobilogy of Cerebellar Evolution and Development, R.Llinas (ed.), American Medical Association, 1969, Chicago.

[19] Piaget J.: In: Measurement and Piaget, (Green E. Ed.), The Theory of Stages in Cognitive Development. McGraw Hill, New York, 1971.

[20] Roschin V., Frolov A.: Multidimensional Dynamic Differential Neurocontrol. Neural Network World, 1996; 6, 355-361.

[21] Rosenblatt F.: The Perceptron: A Probabilistic Model for Information Starage and Organization in the Brain. Psychol. Rev., 1958; 65, 386-408.

[22] Řízek S., Frolov A.: Differential Control by Neural Networks. Neural Network World 1994; 4, 493-508.

[23] Řízek S., Frolov A.: Influence of Feedback upon Learning of the Differential Neurocontroller. Neural Network World, 1996; 6, 347-353