Yugoslav Journal of Operations Research
2 (1992), Number 2, 235-252

CONNECTIONIST ARCHITECTURES FOR CONTROL OF MANIPULATION ROBOTS BY FEEDBACK-ERROR LEARNING METHOD

Duško M. KATIĆ, Miomir K. VUKOBRATOVIĆ

Robotics Department, Mihailo Pupin Institute Belgrade, 11000 Beograd, Yugoslavia

Abstract. A major objective in this paper is the application of highly efficient connectionist architectures for fast and robust learning of dynamic relations used in robot control at the executive hierarchical level. Two types of neural network control structures are considered: a single-layer neural network and a multilayer perceptron. The proposed connectionist learning models are applied as a form of intelligent feedforward robot control in the frame of decentralized control algorithm with feedback-error learning method. The final result of this approach is a trainable robot controller with excellent learning properties. Efficiency and verification of the proposed algorithms through simulation examples of robot trajectory tracking is shown.

1. INTRODUCTION

The powerful development of flexible manufacturing systems with high and complex demands for all components of the production process, leads toward the design of intelligent manipulation robots [1, 2]. We can define these robots as autonomous machines capable of learning, making decisions, fault tolerance, analysis, etc. Being robust and adaptive to internal and external disturbances, problems as compensating for uncertainties, diagnosing failures, identifying failed components, and recovering from errors can be successfully tackled by using these machines. Such robots also, can perform anthropomorphic tasks in an unfamiliar of familiar working environment. There has been a significant effort in making robot more intelligent by integrating advanced sensor systems such as vision, tactile sensing, etc. But, one of the major and ultimate steps in robotic research is the formulation and development of intelligent control algorithms which can further improve the performance of robotic systems, using control strategies generated by human intelligent functions such as perception, association, reasoning, generalization or learning.

The basic problems for the classic model-based control of manipulation robots are the state variables-dependency of the robot dynamic model, the expressive coupling bet ween robot subsystems, coping with structured and unstructured uncertainties and time-dependency of robot parameters. It is known that none of the classic robot controllers cannot provide desirable solutions to these problems, because traditional control laws are, in most cases, based on models with incomplete information and partially known or inaccurately defined parameters. Also, the classic algorithms are extremely sensitive to the lack of sensor information and unplanned events and unfamiliar situations in the robot working environment. The robot performance is not able to capture and utilize past experience and available human expertise. All previously mentioned facts and examples provide a motivation for robotic intelligent control and emphasize the necessity that efficient robotic intelligent control must be based on learning, generalization and self-organizing capabilities.

The classic adaptive and non-adaptive control algorithms comprise robot control problem during execution of single robot trajectories without considering repetitive motion. Hence, in terms of learning, almost all manipulation robots are memoryless. In this way, the previously acquired experience about dynamic robot model and control algorithms is not applied in robot control synthesis. It is expected that using a training process which repeats a control task and records the results accumulated in the entire process would steadily improve the performance. Also, state variables-dependency of robot dynamics may be solved by learning and storing the solution, while time-dependency of robot parameters requires an on-line learning approach. If learning control algorithm once learned some movement, it could control quite different and faster movement using generalization properties of learning algorithm. hence, one of the primary goals in intelligent control of manipulation robots is the addition of learning and generalization capabilities to the classic non-adaptive and adaptive control algorithms.

The recent research reports and extensive simulation studies carried out on models containing neural networks have demonstrated an ability to identify and control sophisticated manipulation robots [**3–9**]. From a systems theoretic point of view, we can say that multilayer neural networks used in robot control in most cases represent static nonlinear mappings as a special part of the pattern recognition problem. In this case the patterns to be recognized are the signals of "change" that map in "control action" signals aiming at desired control goals. The neural network controller should recognize and isolate signals of "change" in real-time conditions, and using learning by experience and generalization properties, to control efficiently system behavior. Through the training process, the model uncertainties are eliminated, and thus, a neural network serves as compensation tool in control systems.

In this paper, our purpose is presentation of new robot control learning algorithms with fast and robust learning properties using special connectionist (neural network) architectures. The major concern is the application of neural networks in robot control at executive hierarchical level (motion control problem) for learning inverse dynamic model of robot mechanism in the case where exact robot dynamics are generally unknown.

Also, one of the main ideas of this paper is to accomplish coexistence of structures that are developed for non-reccurent single-layer and multilayer networks, using the framework of a decentralized control algorithm (a well-known classic robot control algorithm) [**3**]. Several neural network models and learning schemes were recently applied to learning of robot dynamics. One main distinction between these methods is in the extent of the knowledge about dynamic models which is used in design procedures. Some methods (single-layer neural networks) use in design procedure complete available information about robot model [**7–9**]. At the other side of dynamic connectionist approaches are methods (multilayer perceptrons) that use "black-box" approach in design of neural network algorithms for robot dynamic control [**4–6**]. In this case neural networks can be used as very general computation models.

The other important features of this new proposed control structure is fast convergence properties due to functional decomposition of system dynamics and new learning rules based on recursive estimation methods. Using repetitive execution of the working task and new learning algorithms with feedback driving torque error, redistribution of feedforward and feedback control are accomplished, that results in fast system response and efficient generalization. Training and learning by proposed neural network architectures can be accomplished using off-line and on-line approach. Efficiency of proposed learning algorithms will be shown using data about industrial robot UMS-2.

2. ROBOT CONTROL PROBLEM AND CONNECTIONIST SOLUTION

In contemporary robotic systems, there is a need for more flexible and robust robot controllers in order to take full advantage of the inherent flexibility and versatility of manipulation robots. The difficulties in solution of control problem arise from various aspects.

The one of the most important problems is high nonlinearity with expressive couplings between subsystems. Based on well-known equations of rigid body mechanics, a dynamic model of manipulation robot in the absence of friction and other disturbances (deterministic model) can be written as

$$P = f(q, \dot{q}, \ddot{q}, \Theta) = H(q, \Theta)\ddot{q} + h(q, \dot{q}, \Theta) \tag{1}$$

or

$$P = f(q, \dot{q}, \ddot{q}, \Theta) = H(q, \Theta)\ddot{q} + \dot{q}^{\mathrm{T}} C(q, \Theta)\dot{q} + g(q, \Theta) \tag{2}$$

where $P \in \mathbf{R}^n$ is the vector of driving torques of forces; $H(q, \Theta) : \mathbf{R}^n \times \Theta \to \mathbf{R}^{n \times n}$ is the inertial matrix of the system; $h(q, \dot{q}, \Theta) : \mathbf{R}^n \times \mathbf{R}^n \times \Theta \to \mathbf{R}^n$ is the vector which includes Centrifugal, Coriolis and gravitational effects; $C(q, \Theta) : \mathbf{R}^n \times \Theta \to \mathbf{R}^n \times \mathbf{R}^n \times \mathbf{R}^n$ is the matrix which includes Centrifugal and Coriolis effects; $g(q, \Theta) : \mathbf{R}^n \times \Theta \to \mathbf{R}^n$ is the gravitational vector; $\Theta \in \mathbf{R}^{nt}$ is the system parameter vector; n is the number of degree of freedom; nt is the number of system parameters.

A common and classic way for robot control represents local PID regulators for each degree of freedom of the robotic mechanism [**10**]:

$$u = u_{fb} = -KP\varepsilon - KD\dot{\varepsilon} - KI\int \varepsilon\, dt \qquad (3)$$

where $u \in \mathbf{R}^n$ is the control input; $u_{fb} \in \mathbf{R}^n$ is the feedback control; $KP \in \mathbf{R}^{n\times n}$ is the matrix of local position feedback gains; $KD \in \mathbf{R}^{n\times n}$ is the matrix of local velocity feedback gains; $KI \in \mathbf{R}^{n\times n}$ is the matrix of local integral feedback gains; $\varepsilon = q - q^0$ is the feedback error ($\varepsilon \in \mathbf{R}^n$); q and q^0 are the real and nominal internal coordinates ($q \in \mathbf{R}^n$, $q^0 \in \mathbf{R}^n$).

However, this control law is not adequate for advanced industrial robots with the requirements for high precision and speed in a complex working environment. The influence of couplings between the subsystems is substantial, and we have to include as a solution "dynamic" control [**10**] which takes the dynamic model of robot mechanism in control synthesis as form of feedforward control. On the basis of the above, we can apply the decentralized control algorithm [**10**]:

$$u = u_{ff} - KP\varepsilon - KD\dot{\varepsilon} - KI\int \varepsilon\, dt \qquad (4)$$

where u_{ff} is the nominal centralized feedforward control which is off-line synthesized using the integral robot model (model of mechanism with the model of robot actuators).

However, in procedure of controller design, we have to cope with structured uncertainties (inaccuracies of model parameters), unstructured uncertainties (unmodelled high frequency dynamics as structural resonant modes, neglected time-delays, actuator dynamics, sampling effects, etc.) and measurement noise. Also, time-varying nature of robot parameters and variability of robot tasks represent additional difficulties for the control system. In this case, the classic non-adaptive algorithms are not robust enough, because these algorithms compensate only a small part of the mentioned uncertainties. Hence, a more suitable approach would be the one using adaptive control techniques. The adaptive control techniques in robotics can be applied as a form of the well-known Model-Referenced Adaptive Control (MRAC) or Self-Tuning method [**10**], with the possibility of adaptation in feedforward or feedback loops.

In conclusion, the classic adaptive control techniques robotics are effective to compensate structured uncertainties, but in the presence of sensor data overload, heuristic sensor information, limits on real-time applicability and wide interval of unstructured uncertainties, the application of adaptive control is not sufficient for high-quality performance.

Therefore, a solution to the robot control problem will likely need to combine classic approaches with new learning approaches in order to achieve good performance. For the robot control problem and learning we can identify three main paradigms: a) Iterative — analytical methods; b) Tabular methods; c) Connectionist methods.

Connectionist methods as most promising approach for learning control provide the implementation tools for complex input/output relations of robot's dynamics and kinematics. One of the main goals of dynamic learning methods is to find solution for the robot inverse dynamic problem. Let us explain the inverse dynamic problem of robot control in a computational framework. There are causal relation between robot driving torque and the resulting robot movement coordinates. Let $P(t)$ denote the time history of driving torque and $q(t)$ denote the time history of the robot internal coordinates during the trajectory. Also we can denote the causal relation between P and q using the functional F, i.e. $F(P(\cdot)) = q(\cdot)$. If we want the robot to tracks desired trajectory q_d, the problem to generate a desired driving torque P_d which realizes q, is equivalent to finding an inverse mapping of the functional F. The connectionist approach may, in principle, solve the problem of variable-coupling complexity and state-dependency of robot dynamic model, because neural networks through the process of training can approximate input/output mappings. In this way connectionist structure as part of decentralized feedforward control law can compensate wide range of robot uncertainties. Also, learning by neural networks is based on excellent association and generalization properties. Hence, if a neural network has once learned a certain movement, it could control quite different and faster movements. The fast computational capability of neural networks enables real-time applicability of robot control algorithms.

However, there are some problems in application of connectionist approach in robot control. First, there is no guarantee in learning processes for convergence of error to a local minimum. Second, neural networks implements only an approximation of inverse mapping of functional F and in this way accuracy and robustness of this approximation may be questionable. Third, there is no systematic way for determination the optimal topology of connectionist structure (number of layers and neural units, choice of activation functions, convergence parameters, etc.). Also, there are problems in specifying the "good" teacher ("good" basic control algorithm) for supervised learning robot control and specifying the optimal set of input patterns for efficient generalization. The one of main problems is acceleration of learning rules because connectionist structure may require a long training period to converge. All these problems are open field for connectionist and robotic researchers. Hence, it is our intention in this paper to give a solution to some of the above problems in order to achieve high efficient robot learning control.

3. CONNECTIONIST LEARNING STRUCTURES AS PART OF FEEDFORWARD ROBOT CONTROLLERS

3.1. Single-Layer Neural Network Control Structure

In the first case, it is proposed that some, whatever small, a přiori knowledge of the robot dynamics is always available. We expect that single-layer neural network model using the a priori knowledge about robot dynamics, will significantly improve performance of robotic system. hence, the proposed neural network models can be regarded as examples of the autonomous driving torque generator (Fig. 1). This connectionist structure is commonly used as part of feedforward controller in decentralized control algorithm. In this case, the feedback controller serves as a robust controller to achieve low errors and perform high-quality learning, because the feedforward controller alone is not sufficient for accurate tracking.

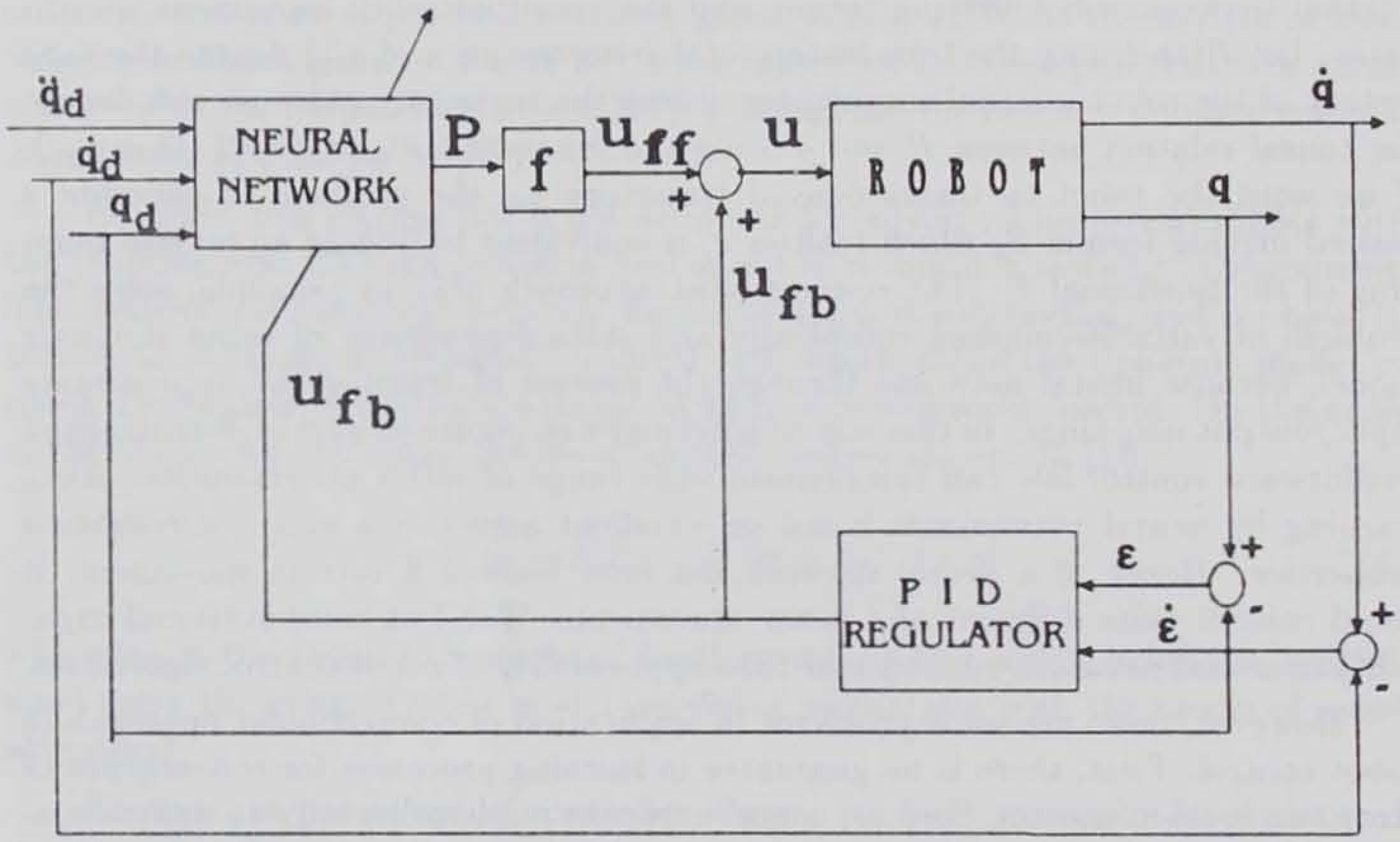

Fig. 1. Decentralized control with connectionist feedback-error learning

Training and learning by proposed connectionist structure is accomplished exclusively in on-line regime by feedback-error learning method [7] (Fig. 1). This method is exclusively on-line method for robot control, but this control structure provides an internal teacher so that the control scheme works in an unsupervised manner, because we have no external teacher in this case. The adjustment of the network weights during the real-time control by feedback-error learning is more convenient than other learning structures as generalized or specialized learning [4].

The neural network model as fixed nonrecurrent single-layer network, generates necessary driving torques in robot joints which is expressed as linear (weighted) sum of a set of specific nonlinear dynamic functions:

$$P_i^{0^*} = \sum_{j=1}^{n} w_{ij} H_{ij}(q^0, \Theta)\ddot{q}_j^0 + w_{i,n+1} h_i(q_i^0, \dot{q}_i^0, \Theta), \qquad i = 1, \dots, n \tag{5}$$

where $P_i^{0^*} \in \mathbf{R}^n$ is joint driving torque generated by neural network; w_{ij} — adaptive weighting factors of neural network ($W \in \mathbf{R}^{n \times n+1} = W(w_{ij})$).

The output of artificial neuron is bounded with linear threshold function (activation function of neuron) due to real physical constraints. Using equation (5) and according to integral model of robotic systems. decentralized control with learning has the next form:

$$u_i = A_{ii}\ddot{q}_i - B_{ii}\dot{q}_i - C_{ii}P_i^{0^*} - KP_{ii}\varepsilon_i - KD_{ii}\dot{\varepsilon}_i - KI_{ii}\int \varepsilon_i \, dt, \quad i = 1, \dots, n, \tag{6}$$

where A_{ii}, B_{ii}, C_{ii} are constant parameters of robot actuators model.

Training and learning of single-layer network is accomplished using error-correcting method based on well-known *DELTA* rule **[11]** (Widrow-Hoff least mean square algorithm):

$$\tau\frac{dw_{ij}}{dt} = D_{ij}E_i = D_{ij}\left(P_i - P_i^{0^*}\right) = D_{ij}\left(P_i - \sum_{j=1}^{n} w_{ij}H_{ij}(q^0, \Theta)\ddot{q}_j{}^0 - w_{i,n+1}h_i(q^0, \dot{q}^0, \Theta)\right) \tag{7}$$

where τ is the learning time constant; E_i — driving torque feedback error; D_{ij} — selected part of dynamic robot model ($H_{ij}(q, \Theta)$ or $h_i(q, \dot{q}, \Theta)$).

Approximately, we can calculate driving torque error in implicit way utilizing robot actuators model and position, velocity and acceleration errors in form:

$$\tau\frac{dw_{ij}}{dt} = D_{ij}\left[\frac{A_{ii}}{C_{ii}}\ddot{\varepsilon}_i + \frac{KD_{ii} + B_{ii}}{C_{ii}}\dot{\varepsilon}_i + \frac{KP_{ii}}{C_{ii}}\varepsilon_i + \frac{KI_{ii}}{C_{ii}}\int \varepsilon_i \, dt\right], \quad i = 1, \dots, n; \quad j = 1, \dots, n+1. \tag{8}$$

The topology of neural network in this case is specified as deterministic model, because the internal dynamic model of robot mechanism using symmetry feature of inertial matrix is divided in $n(n+1)/2$ separate units with $n(n+1)/2$ tuning weighting factors. In this case, for choice of neural units only general knowledge about robot dynamics is used. But this choice is not limited, and we can divide dynamic model in greater number of separate units but only for particular robot mechanism. As it was mentioned, this is a deterministic approach, because we use any a priori knowledge about the dynamic structure of the controlled object and in this way we have better scale-up properties. This approach by single-layer neural networks however, has several drawbacks related primarily to the inherent complexity of the implementation of a complete model of robot dynamics, which demands large computational power in real-time. As second, the way to find out an appropriate set of nonlinear functions is not-trivial and not-unique task. The limitation of this approach is evident in the case when the exact form of the dynamic equation is not a priori known, i.e., the system is not capable to learn how to handle a completely unknown manipulator. Besides, one of the main drawbacks is the problem of generality, because the formulation complexity of nonlinear subsystems is directly dependent on features of each specific model.

3.2. Multilayer Perceptron Control Structure

In this case, basic principles of training and learning according to Fig. 1 are the same as in the previous case. Now, the topology of proposed connectionist structure for robot control is defined by four layer perceptron with symmetric sigmoid function as activation functions in both of hidden layers. The network has input layer with $3n$ neurons and output layer with n neurons. This number of neurons in appropriate layers is determinated according to number of degree of freedom for standard robot configuration. The activation function for input and output layer is identity function. The number of neurons in hidden layers is determinated by simulation experiments and experience ($12n$ neurons in first hidden layer; $6n$ neurons in second hidden layer). The neural network with proposed topology of fixed nonrecurrent multilayer network generates necessary driving torques in robot joints as nonlinear mapping of robot desired internal coordinates, velocities and accelerations:

$$P_i = g(w_{jk}^{ab}, q_d, \dot{q}_d, \ddot{q}_d) \qquad i = 1, \dots, n. \tag{9}$$

where $P_i \in \mathbf{R}^n$ is joint driving torque generated by neural network; w_{jk}^{ab} adaptive weighting factors between neuron j in a-th layer and neuron k in b-th layer; g — nonlinear mapping.

According to integral model of robotic systems, decentralized control algorithm with learning has the next form:

$$u_i = f_i(q_d, \dot{q}_i, \ddot{q}_i, P) - KP_{ii}\varepsilon_i - KD_{ii}\dot{\varepsilon}_i - KI_{ii}\int \varepsilon_i \, dt, \quad i = 1, \dots, n \tag{10}$$

where f_i is the nonlinear mapping which describes nature of robot actuator model.

Training and learning of proposed connectionist structure can be accomplished using well-known *back propagation algorithm* **[11]**. In the process of training we can use two type of output error for back propagation algorithm. The first type of error is feedback control signal

$$e_i^{bp} = u_{\text{fb}}^i \qquad i = 1, \dots, n \tag{11}$$

where $e_i^{bp} \in \mathbf{R}^n$ is the output error for back propagation algorithm.

But, in fact when we consider integral modelling of robot mechanism with model of robot actuators, feedback control signal is not output error for neural network. Thus, we have to calculate driving torque error signal

$$e_i^{bp} = P_i^r - P_i = a^i \ddot{\varepsilon}_i + b^i \dot{\varepsilon}_i + c^i u_{\text{fb}}^i \qquad i = 1, \dots, n, \tag{12}$$

where $P_i^r \in \mathbf{R}^n$ is the real robot driving torque; $a^i \in \mathbf{R}^n$, $b^i \in \mathbf{R}^n$, $c^i \in \mathbf{R}^n$, are constant parameters of robot integral model (this model is valid for robot DC-actuators).

Although the proposed pure or naive neural network approach without knowledge about robot dynamics may be promising, it is important to notice that this approach will not be very practical because of high dimensionality of input-output spaces and long learning time. For example, with most manipulation robots having 6 d.o.f., we have 18 input variables and 6 output variables. Also, the number of trajectory patterns np (input and output variables) may be very large (10–100) if we want to examine the whole working space. Hence, for the robot training it would be necessary to present np^{18} samples, i.e. the training by pure connectionist models would require a neural network of impractical size and unreasonable number of repetition cycles. Therefore, we can conclude that the naive connectionist approaches are only applicable for the low-dimension robotic systems.

The aim of proposed approach for design of connectionist robot controller is the leaving of control synthesis without a priori information about robot dynamics. hence, it is very important to use all the available information about robot dynamic but only in general and specific form. The general knowledge for that purpose is conveniently incorporated into the structure of network. The way of attaining above goals is a decomposition of robot dynamics on simpler robot dynamic relations. In this way, instead of using single neural network, training and learning is accomplished by several neural subnetworks which have simpler input/output relations that enables significant reduction of learning time.

In this paper, **"3F-2SF" decomposition** (decomposition of three-vector function into two two-vector subfunctions) is proposed. Namely, we can see that in robot dynamics several terms can be identified which have a distinctive functional dependency. Exactly, basic robot model (1) can be decomposed into two terms:

first term	$H(q,\Theta)\ddot{q}$	or	$F_1(q,\ddot{q},\Theta)$
second term	$h(q,\dot{q},\Theta)$	or	$F_2(q,\dot{q},\Theta)$.

With this type of decomposition, instead of multilayer perceptron with $3n$ input values, we have two multilayer perceptrons with $2n$ inputs and n outputs for approximation of mapping F_1 and F_2:

$$P_i^{NN1} = F_1(w_{jk}^{NN1ab}, q_d, \ddot{q}_d) \qquad i = 1, \ldots, n \tag{13}$$

$$P_i^{NN2} = F_2(w_{jk}^{NN2ab}, q_d, \dot{q}_d) \qquad i = 1, \ldots, n \tag{14}$$

$$P_i = P_i^{NN1} + P_i^{NN2} \qquad i = 1, \ldots, n \tag{15}$$

where F_1 is a nonlinear mapping for first perceptron $NN1$; F_2 is a nonlinear mapping for second perceptron $NN2$; P_i^{NN1} and P_i^{NN2} are parts of robot dynamic model generated by perceptrons $NN1$ and $NN2$; w_{jk}^{NN1ab} and w_{jk}^{NN2ab} are weighting factors for perceptrons $NN1$ and $NN2$; P_i is driving torque at the output of connectionist structure.

The topology of perceptrons $NN1$ and $NN2$ are determinated using similar activation functions and principles as in previous case (now we have: input layer — $2n$ neural units; first hidden layer — $8n$ neural units; second hidden layer — $4n$ neural units; output layer — n neural units). Training of both perceptrons is accomplished synchronously by feedback-error learning method (Fig. 2) The feedback error signal or driving torque error signal are transferred as output backpropagation error to both of perceptron outputs.

The backpropagation algorithm caused a tremendous breakthrough in the control application of multilayer perceptrons. One of the major drawbacks of this method is its slow convergence. Starting from a random initial state, the path

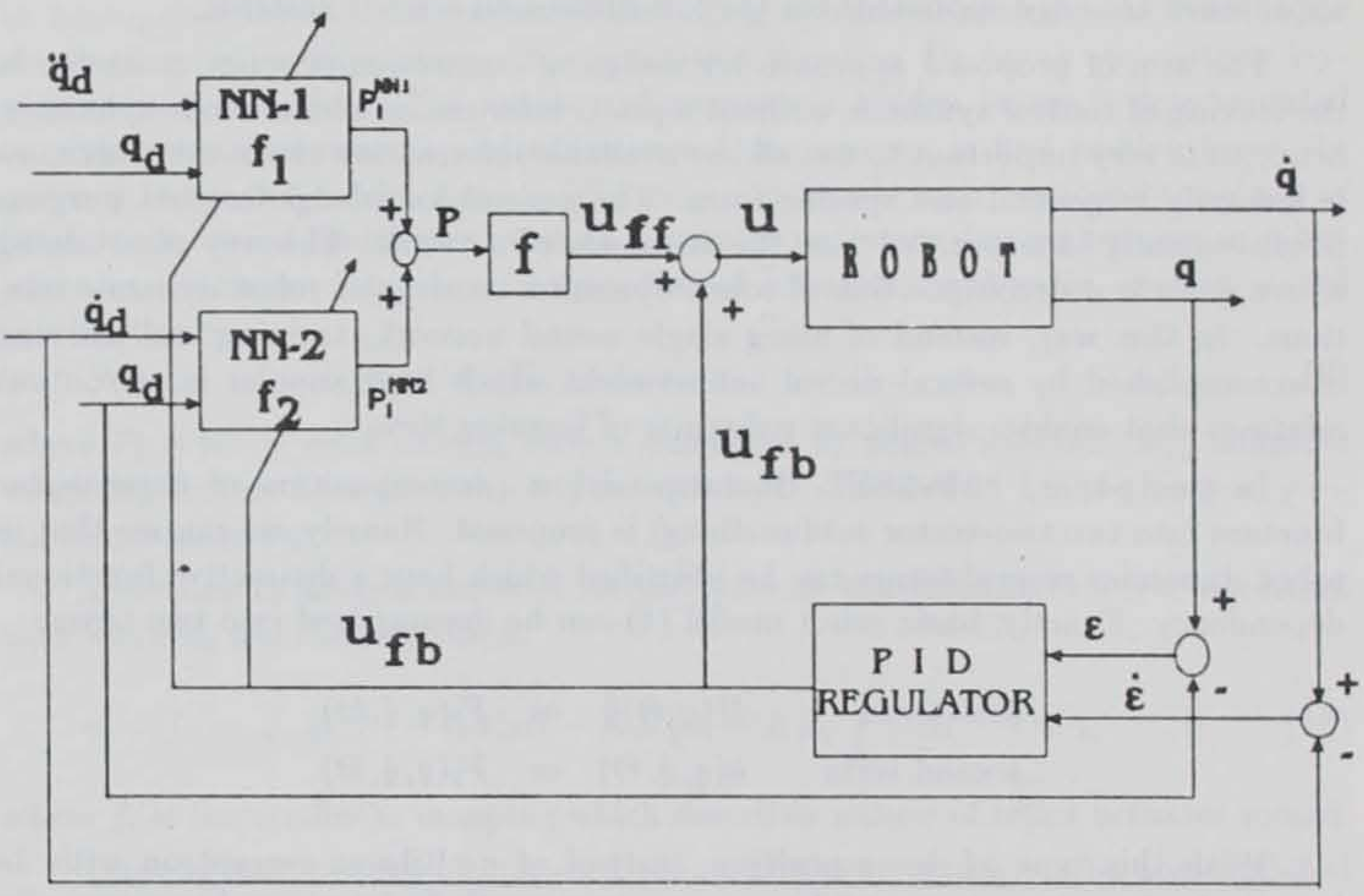

Fig. 2. "3F-2SF" Decomposition connectionist structure with feedback error learning

to the global minimum is often strewn with local minimum, causing oscillations around ravines in the weight space.

In this paper, our intention is that instead acceleration of standard back propagation algorithm using standard methods from numerical analysis, consider the problem of adjusting the weights of internal hidden units as a problem of estimating parameters by well-known identification method — Recursive Least Square (RLS) method [12]. Using these methods with time-varying learning rate yield to benefits for learning speed and generalization in comparison with standard back propagation algorithm.

The proposed new algorithms are based on previously defined 4-layer decomposed connectionist structure ("3F-2SF") with appropriately defined parameters of network.

The main forward network relations in process of training are described according to next expressions:

Forward Relations — RLS Method

$$s^2(k) = W^{12} i^1(k) \tag{16}$$

$$o_a^2(k) = \frac{1}{1+\exp(-s_a^2(k))} - 0.5, \quad a = 1, \ldots, L_1, \quad o_0^2(k) = 1 \tag{17}$$

$$s^3(k) = W^{23}o^2(k) \tag{18}$$

$$o_b^3(k) = \frac{1}{1+\exp(-s_b^3(k))} - 0.5 \quad a = 1, \dots, L_2, \quad o_0^3(k) = 1 \tag{19}$$

$$s^4(k) = W^{34}o^3(k) \tag{20}$$

$$y(k) = s^4(k) = P^{NNp} \quad c = 1, \dots, n, \; l = 1, \dots, n, \; p = 1 \text{ or } 2 \tag{21}$$

where $s^2(k)$, $s^3(k)$, $s^4(k)$ are the output vectors for linear partsof layers; $o^2(k)$, $o^3(k)$ are the output vectors of the hidden layers; $W^{12} = [w^{12}_{m\times L}]$, $W^{23} = [w^{23}_{L1+1\times L2}]$, $W^{34} = [w^{34}_{L2+1\times n}]$ are the weighting factors of the layers; $W^{12} = [w^{12}_{m\times L}]$ is the inputs in the network (robot internal positions, velocities and accelerations — $m = 3n+1$); $y(k)$ is output of network.

The aim of estimation is to define optimal values for matrices W^{12}, W^{23} and W^{23} using models of linear systems according to equations (16), (18) and (20). In application of this method, problem of specification of desired states and errors in hidden layers is arose. Using new solution by taking account of the relationship between the standard backpropagation algorithm and present method at the last layer, high-efficient algorithm that propagates the learning errors at the last layer to the hidden layers is proposed.

The basic equations which describe new learning rules base on RLS method are given according to the following formulas

Learning Rules — RLS Method

$$s_c^{4d}(k) = y_c^d(k) = P_c^r(k) \qquad c = 1, \dots, n \tag{22}$$

$$P^3(k) = \frac{1}{\lambda^3}\left(P^3(k-1) - \frac{P^3(k-1)o^3(k)o^3(k)^{\mathrm{T}}P^3(k-1)}{\lambda^3 + o^3(k)^{\mathrm{T}}P^3(k-1)o^3(k)}\right) \tag{23}$$

$$W^{34}(k) = W^{34}(k-1) + P^3(k)o^3(k)[s^{4d}(k) - W^{34}(k-1)^{\mathrm{T}}o^3(k)]^{\mathrm{T}} \tag{24}$$

$$o^{3d}(k) = o^3(k) + W^{34}(k)\delta^4(k) \tag{25}$$

$$\delta_c^4(k) = S_c^{4d}(k) - \sum_j w_{jc}^{34}(k)o_j^3(k), \quad c = 1, \dots, n, \; j = 1, \dots, L_2+1 \tag{26}$$

$$o^{3d}_{\max}(k) = \max|o_b^{3d}(k)| \qquad b = 1, \dots, L_2 \tag{27}$$

$$s_b^{3d}(k) = f^{-1}\left[\left(\frac{0.49}{o^{3d}_{\max}(k)}\right)o_b^{3d}(k)\right] \qquad b = 1, \dots, L_2 \tag{28}$$

$$P^2(k) = \frac{1}{\lambda^2}\left(P^2(k-1) - \frac{P^2(k-1)o^2(k)o^2(k)^{\mathrm{T}}P^2(k-1)}{\lambda^2 + o^2(k)^{\mathrm{T}}P^2(k-1)o^2(k)}\right) \tag{29}$$

$$W^{23}(k) = W^{23}(k-1) + P^2(k)o^2(k)[s^{3d}(k) - W^{23}(k-1)o^2(k)]^{\mathrm{T}} \tag{30}$$

$$o^{2d}(k) = o^2(k) + W^{23}(k)\delta^3(k) \tag{31}$$

$$\delta_b^3(k) = S_b^{3d}(k) - \sum_j w_{jb}^{23}(k)o_j^2(k), \quad b = 1, \dots, L_2, \; j = 1, \dots, L_1+1 \tag{32}$$

$$o_{\max}^{2d}(k) = \max |o_a^{2d}(k)| \qquad a = 1, \ldots, L_1 \tag{33}$$

$$s_a^{2d}(k) = f^{-1}\left[\left(\frac{0.49}{o_{\max}^{2d}(k)}\right) o_a^{2d}(k)\right] \qquad a = 1, \ldots, L_1 \tag{34}$$

$$P^1(k) = \frac{1}{\lambda^1}\left(P^1(k-1) - \frac{P^1(k-1)i^1(k)i^1(k)^{\mathrm{T}} P^1(k-1)}{\lambda^1 + i^1(k)^{\mathrm{T}} P^1(k-1) i^1(k)}\right) \tag{35}$$

$$W^{12}(k) = W^{12}(k-1) + P^1(k) i^1(k)[s^{2d}(k) - W^{12}(k-1) i^1(k)]^{\mathrm{T}} \tag{36}$$

where λ^1, λ^2, λ^3 are the appropriate forgetting factors.

Initial conditions fro weighting factors are generated by normal distribution with different random numbers:

$$W^{12}(0) = N^{12}(0,1); \quad W^{23}(0) = N^{23}(0,1); \quad W^{34}(0) = N^{34}(0,1). \tag{37}$$

Also, initial conditions for covariance matrices P are given in the following form:

$$P^1(0) = c^1 I_m; \qquad P^2(0) = c^2 I_{L1+1}; \qquad P^3(0) = c^3 I_{L2+1} \tag{38}$$

where c^1, c^2, c^3 are big positive numbers.

4. SIMULATION EXAMPLE

In this section, simulation examples are given to verify the proposed connectionist algorithms compensating the system uncertainties. The manipulation robot used for the simulation is a cylindrical type industrial robot UMS-2 [**10**] (Fig. 3) with 6.d.o.f.

In the learning phase, robot training is accomplished using movement from point A with internal coordinates $q \in \{0.6; 0.05; 0.05; 0.; 0.; 0.\}$ to point B ($q \in \{1.2; 0.12; 0.2; 0.1; 0.1; 0.1\}$). Time duration of movement is $t = 1$s with triangular velocity profile. The PID feedback control was chosen with following values for local feedback gains: $KP \in \{222.; 698.; 426.; 41.; 152.; 51.\}$, $KD \in \{47.; 33.; 21.; 2.; 8.; 2.\}$. We know that exact measuring of the link inertia and position of mass centre is very difficult. Hence in simulation experiments, the model uncertainties are defined by parametric disturbances with approximately 20% variation from nominal values for link mass and moment of inertia).

On Figures 4 and 5 position errors (trial 10 and trial 150) for the first and second degree of freedom in the case of single-layer neural network are given. As we see from these figures, with repetitive trials tracking errors are decreased and learning is accomplished.

In simulation experiments we have applied pure connectionist and decomposed "3F-2SF" network structures. In all simulation experiments, convergence criterion (exactly total epoch square error) was $J_c = 10$ Nm. The learning of robot dynamics is accomplished through trial-and-error approach, when we in successive epochs of training present same trajectory patterns.

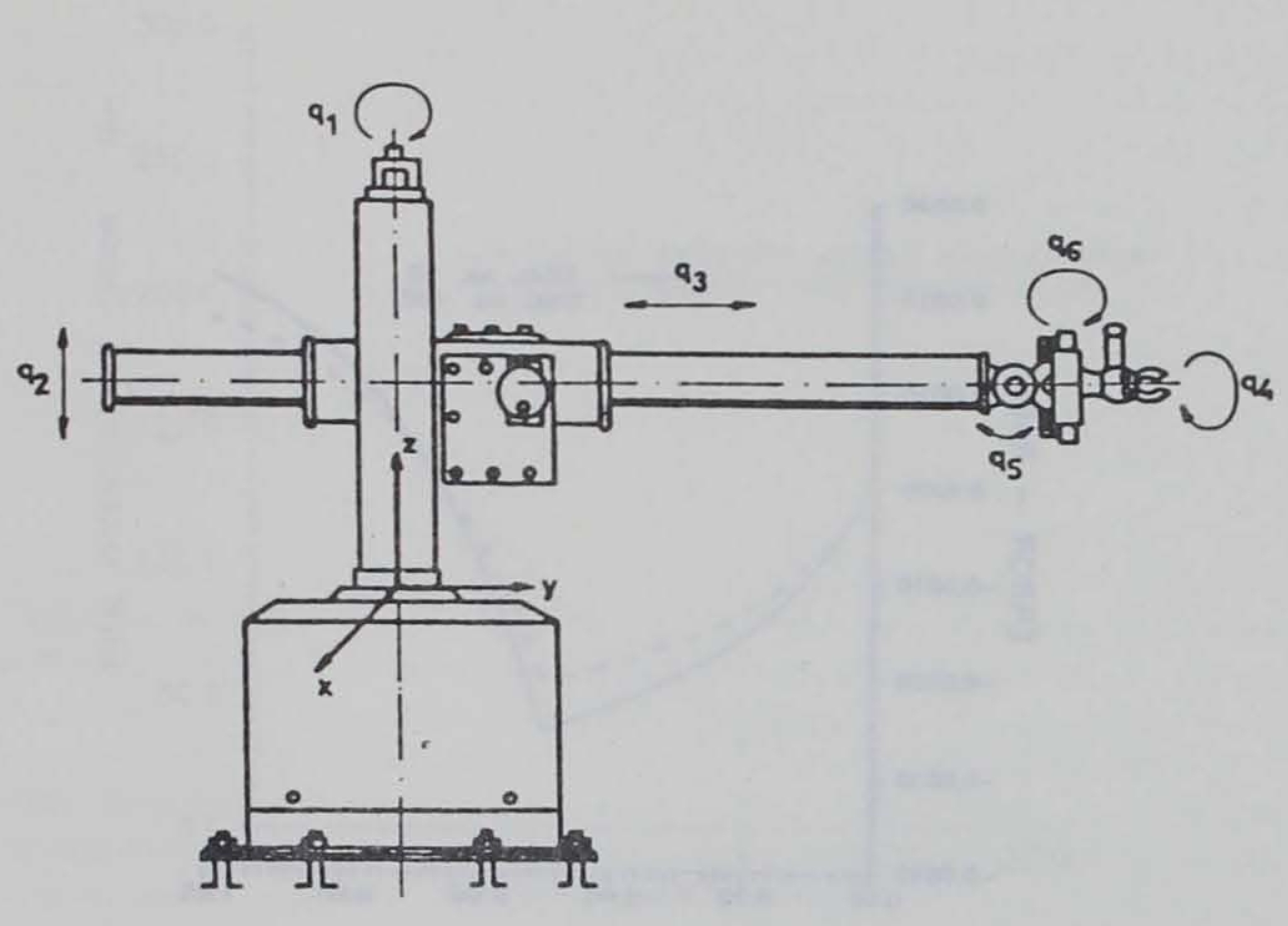

Fig. 3. Industrial robot UMS-2

In simulation experiment with backpropagation algorithm, the learning rate for all layers is $\eta = 0.01$. In the case of RLS connectionist algorithm, covariance matrices have folowing initial values:

$$P^1(0) = 1000; \qquad P^2(0) = 1000; \qquad P^3(0) = 1000;$$

To get a comparison of pure connectionist structure and "3F-2SF" decomposed connectionist structure, some simulations experiments with standard back propagation algorithm under same conditions were performed. Fig. 6 shows the convergence results (total epoch square error during time) for both connectionist structure. The result shows better learning of dynamic robot model and significant reduction of learning time for decomposed connectionist structure.

The convergence results of back propagation algorithm with RLS algorithm (Fig. 7) in the case of on-line learning.

We can conclude from results, that specially with new RLS connectionist method we can achieve fast learning of robot dynamics.

In Figure 8 position error for first d.o.f. in the case of trajectory tracking with back propagation algorithm and RLS method is shown. The figure shows the first and 10th training trial. It took about 1s for one trial, while sampling period was 1 ms.

The figures show that with repetitive trials tracking errors using RLS method are considerably decreased and in this way highly efficient robot dynamic learning

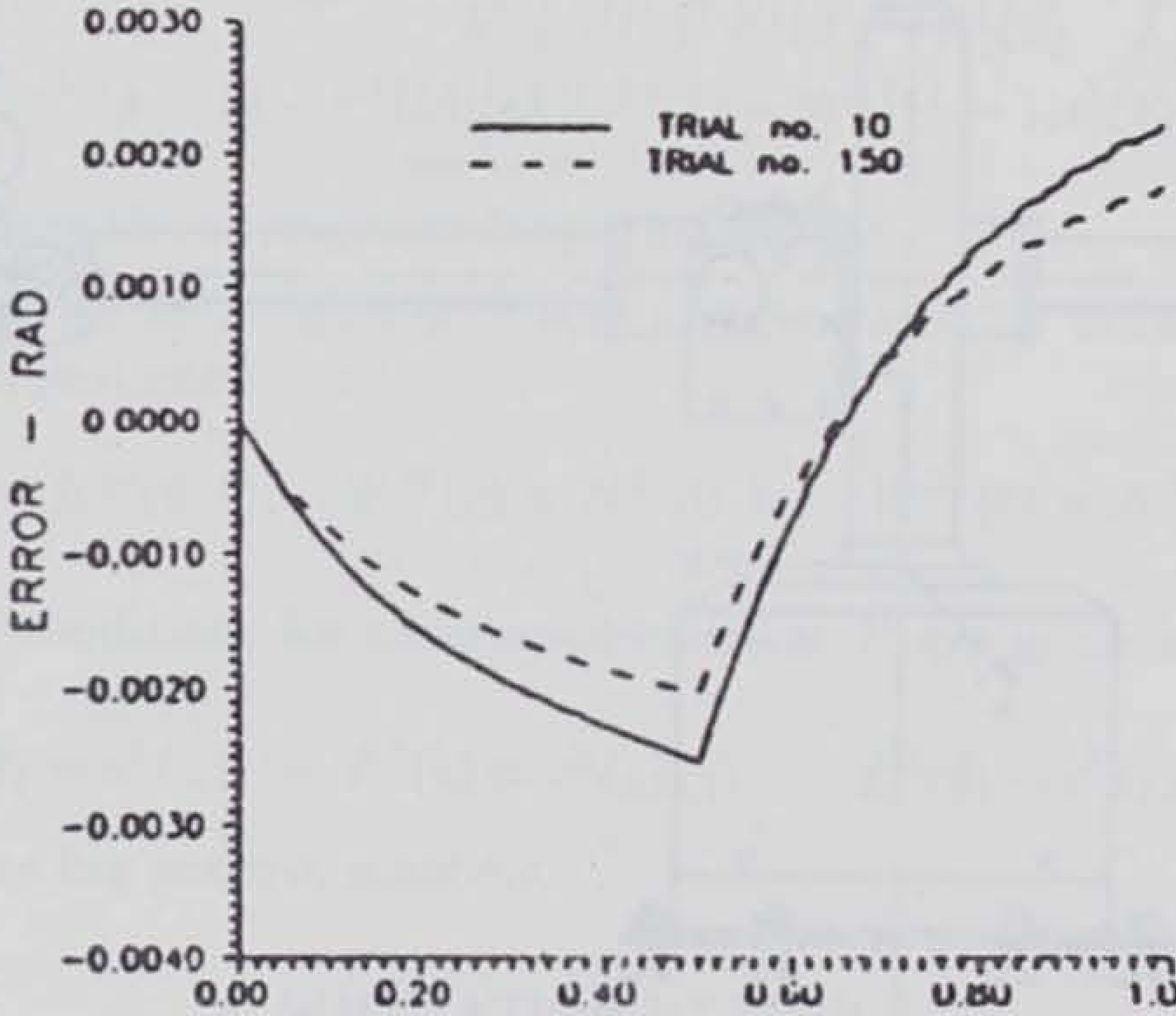

Fig. 4. Position error for the first degree of freedom

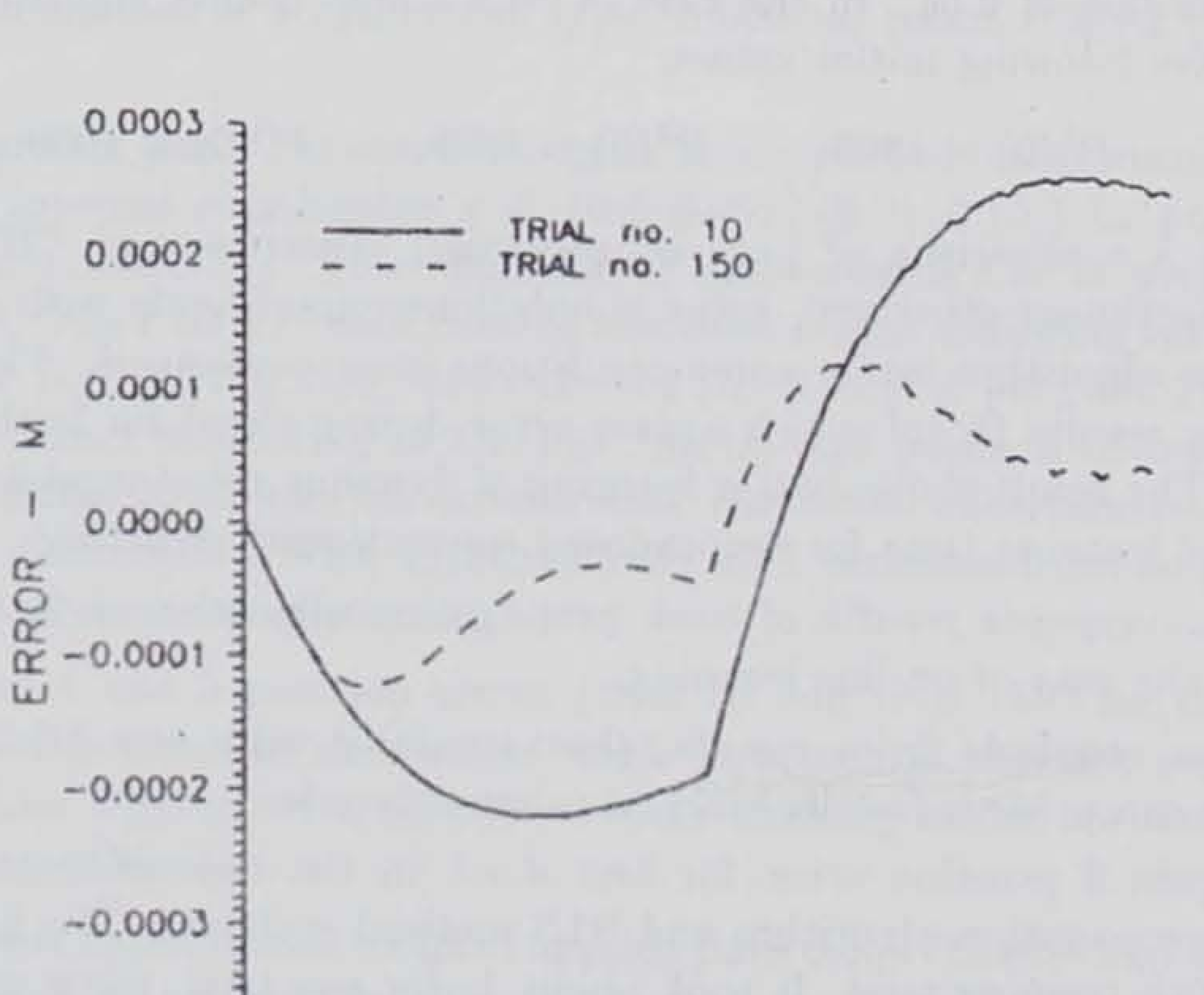

Fig. 5. Position error for the second degree of freedom

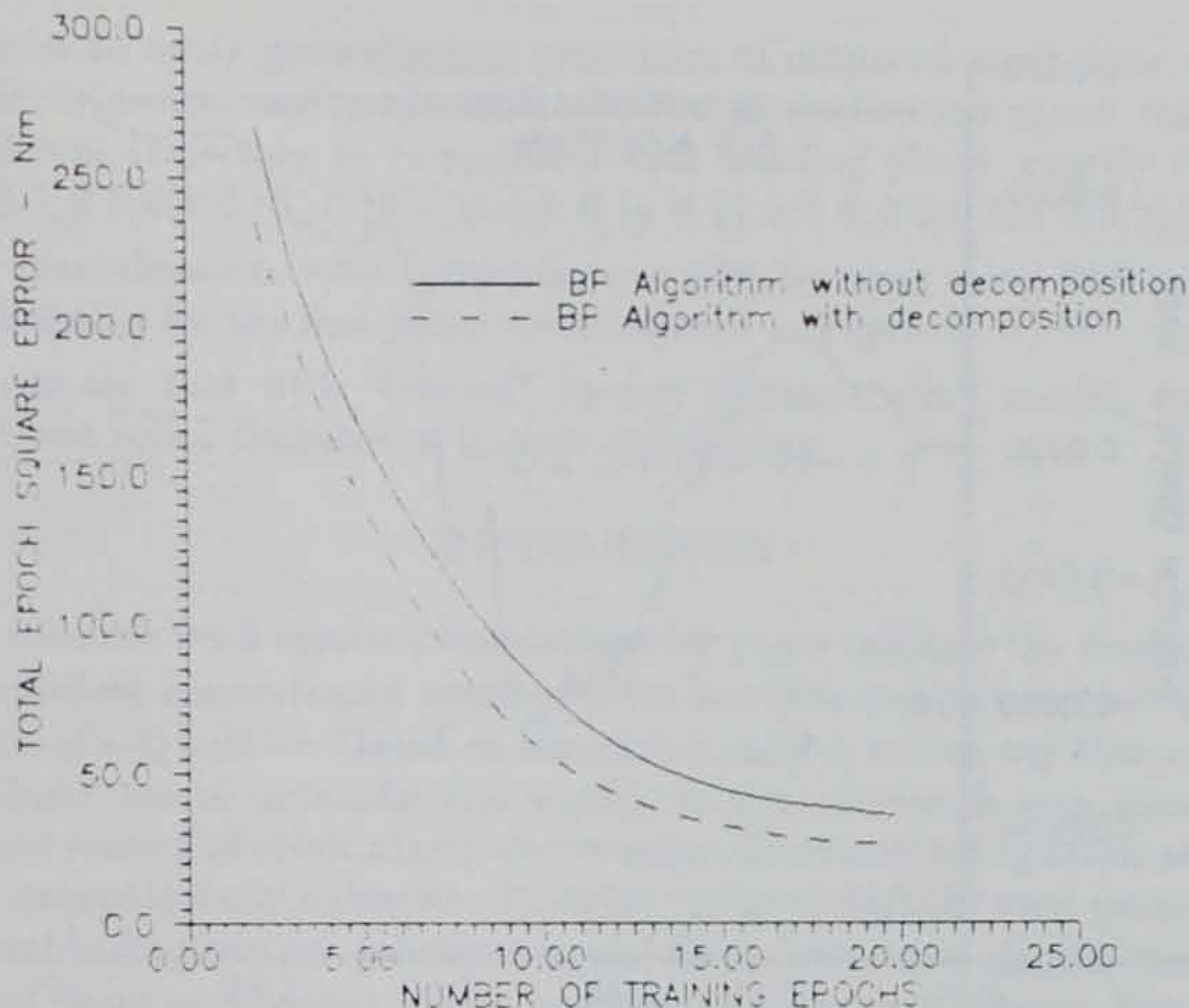

Fig. 6. Convergence results for BP algorithm with and without decomposition

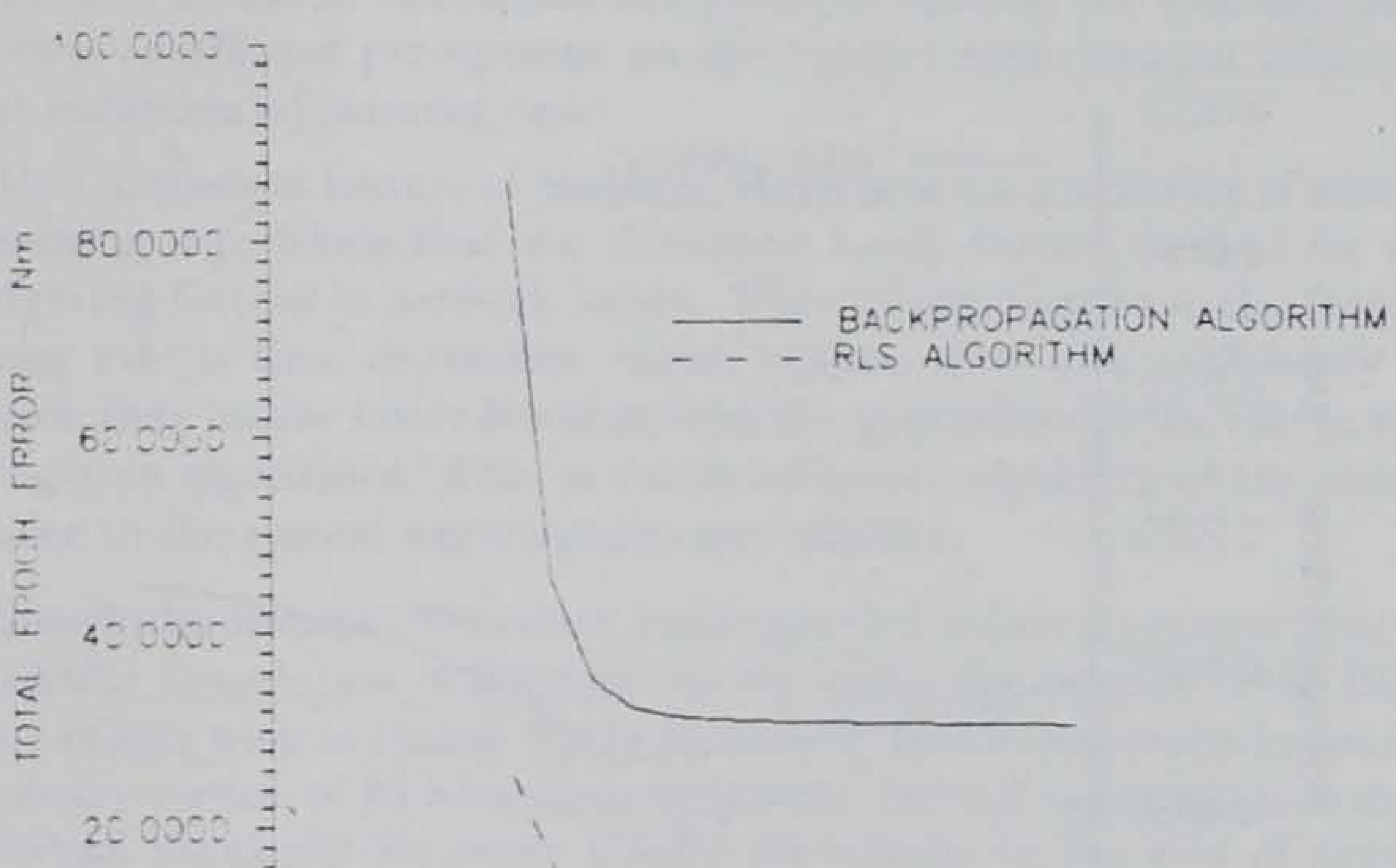

Fig. 7. Total square error during training epochs
for backpropagation algorithm and RLS method

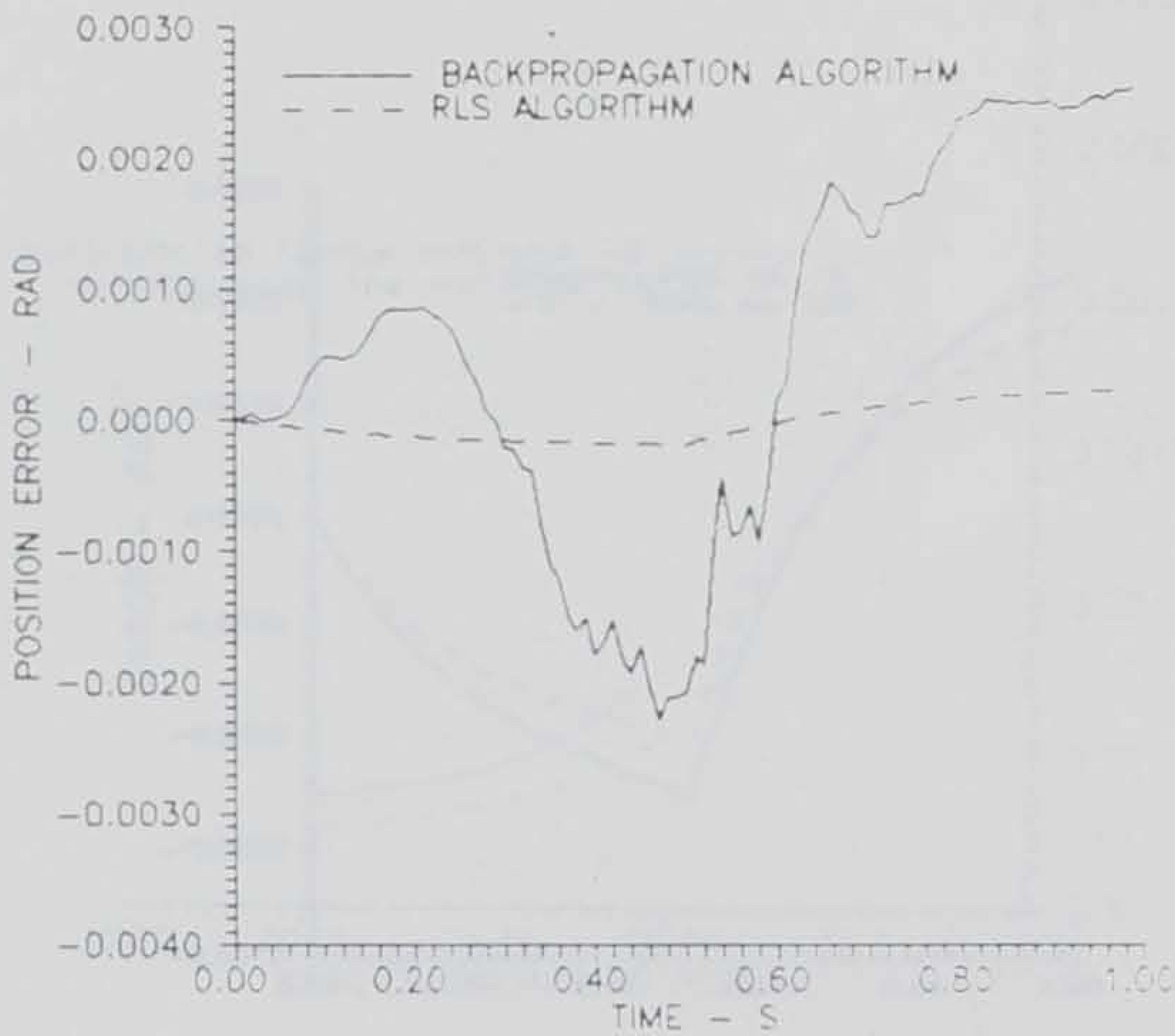

Fig. 8. Comparison of learning algorithms for first d.o.f.

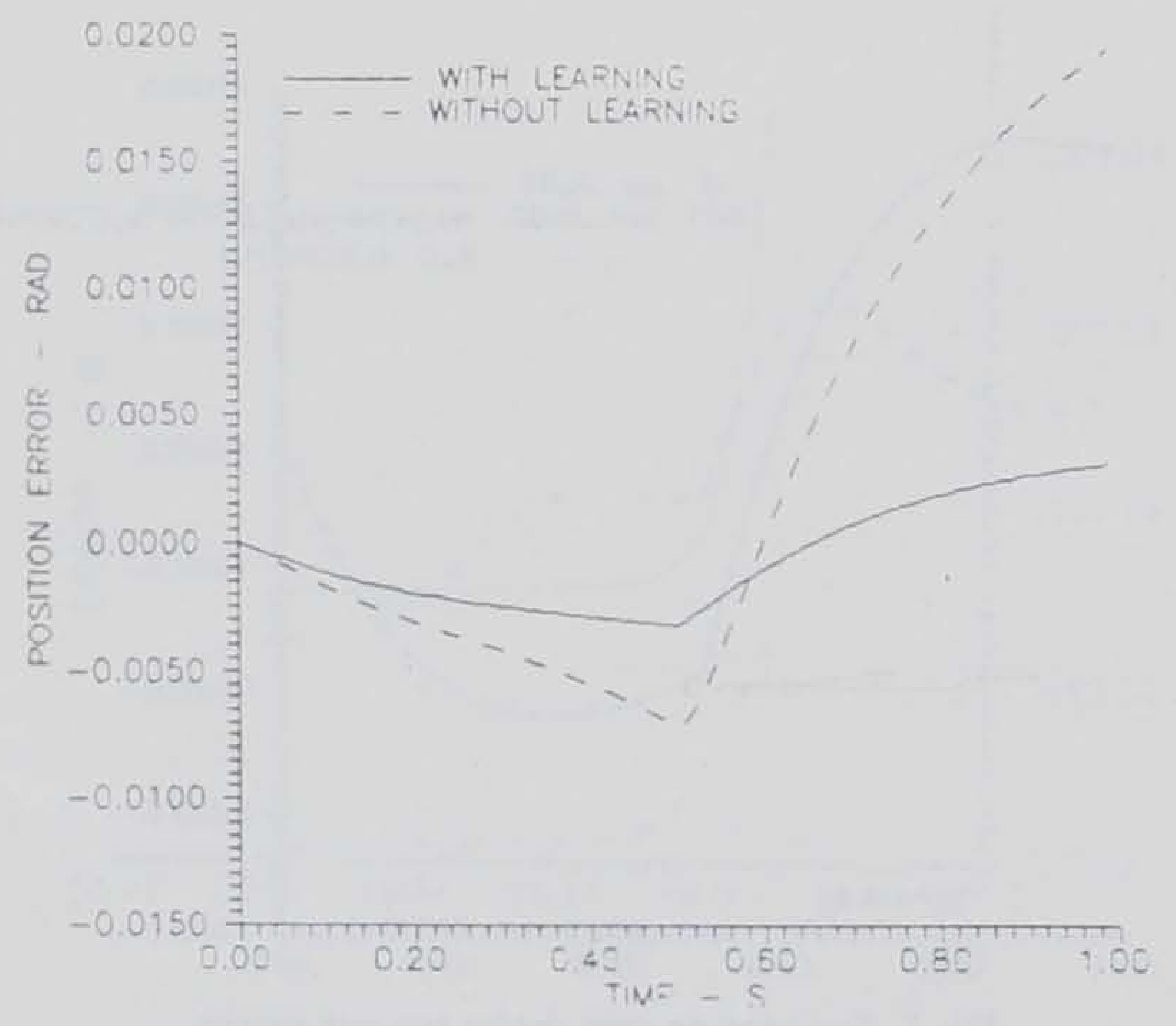

Fig. 9. Position error for the first d.o.f. in generalization phase

is accomplished.

In order to verify generalization properties of proposed algorithms, some simulation experiments were performed. In the generalization phase robot moves along different trajectory in comparison with learning phase, exactly from point A ($q \in \{0.3; 0.1; 0.5; 0.; 0.; 0.\}$) to point B ($q \in \{1.5; 0.6; 0.6; 0.2; 0.2; 0.2\}$).

The generalization result (position error with learning using RLS method and without learning for the first joint) are presented in Figure 9.

We can see that with "learned" inverse dynamic robot model, tracking for quite different robot trajectories is quite satisfactory.

5. CONCLUSION

The research work results presented in this paper indicate the feasibility of using high-efficient connectionist structures for learning complex input/output relations of robot's dynamics. Based on simulation results we can say that application of single-layer neural networks and multilayer perceptrons is very promising for feedforward control of robot manipulators whose fundamental dynamic structure is generally or particularly unknown. However, naive straightforward neural network is not suitable for practical solutions in real-time, because of the increased dimensionality of input and output spaces and long learning time. hence, based on this facts proposed new connectionist control algorithms with decomposed "3F-2SF" structure uses available information about robot dynamic but only in general and specific form. The use of decomposition principle enables the synchronous training of several multilayer perceptrons on the simpler input/output relations with significant reduction of learning time.

Another important feature of proposed structures is a possibility of acceleration of new learning algorithms that use Recursive Least Square method for estimating of weighting factors in network layers. These algorithms have the feature that the learning rate is time dependent, which enables that with sufficiently convergent solution they assure faster learning than the generalized delta rule in standard backpropagation algorithms. Also, as result, adaptive capability of the connectionist controller to the system uncertainties was clarified.

Acknowledgements. This work was supported in part by grants from the National Scientific Foundation of Republic Serbia under the project "1004 ROBOTIKA". The authors wish to thamk Vlada Stefanović for his assistance in the defining the preliminary version of RLS learning algorithm. Special appreciation is expressed to prof. Srdjan Stanković for many helpful discussions in the field of neural Networks.

REFERENCES

[1] G.N. Sardis: *Intelligent robotic control*, IEEE Transactions on Automatic Control **AC-28** (5), May 1983.

[2] A. Meystel: *Intelligent control in robotics*, Journal of Robotic Systems **5** (1988), 269–308.

[3] D. Katić, M. Vukobratović: *Connectionist approaches to control of manipulation robots*, Journal of Intelligent and Robotic Systems, (to appear).

[4] D. Psalitis, A. Sideris, A. Yamamura: *A hierarchical model for voluntary movement and its application to robotics*, Proceedings of 1st IEEE International Conference on Neural Networks, San Diego, June 1987, 551–558.

[5] A. Guez, J. Eilbert, M. Kam: *Neuromorphic architectures for fast adaptive robot control*, Proceedings of the IEEE International Conference on Robotics and Automation, Raleigh, April 1988, 145–149.

[6] D.F. Bassi: *Connectionist Dynamic Control of Robot Manipulators*, Ph.D. dissertation, University of Southern California, Los Angeles, December 1990.

[7] H. Miyamato, M. Kawato, T. Setoyama, R. Suzuki: *Feedback-Error-Learning neural network for trajectory control of a robotic manipulator*, Neural Networks **1** (1988), 251–265.

[8] M. Kawato, Y. Uno, R. Isobe, R. Suzuki: *Hierarchical neural network model for voluntary movement with application to robotics*, IEEE Control Systems Magazine **57** (1987), 169–185.

[9] D. Katić: *Connectionist learning models for intelligent control of manipulation robots*, Proceedings of the IMACS International Symposium on Mathematical and Intelligent Models in System Simulation, Brussels, September 1990.

[10] M. Vukobratović, D. Stokić, N. Kirćanski: *Non-Adaptive and Adaptive Control of Manipulation Robots*, Springer-Verlag, Berlin, 1985.

[11] D.E. Rumelhart, J.L. McClelland: *Parallel Distributed Processing (PDP): Exploration in the Microstructure of Cognition, Vol. 1 and 2*, MIT Press, Cambridge, 1986.

[12] L. Ljung and T. Soderstrom: *Theory and Practice of Recursive Identification*, Cambridge, MIT Press, 1983.