Robotica (1997) volume 15, pp 23–30. © 1997 Cambridge University Press

Neural network sliding mode robot control

Karel Jezernik, Miran Rodič, Riko Šafarič & Boris Curk

Universityo f Maribor, Faculty of Electrical Engineering and Computer Science, Smetanova 17, SI-2000 Maribor (Slovenia) E-mail: karel.jezernik@uni-mb.si

SUMMARY

This paper develops a method for neural network control design with sliding modes in which robustness is inherent. Neural network control is formulated to become a class of variable structure (VSS) control. Sliding modes are used to determine best values for parameters in neural network learning rules, thereby robustness in learning control can be improved. A switching manifold is prescribed and the phase trajectory is demanded to satisfy both, the reaching condition and the sliding condition for sliding modes.

A major objective of the work described has been to develop neural network architectures which will provide fast and robust on-line learning of the dynamic relations required by robot controller at the executive hierarchical level. The approach to proposed robot control involves using a neural network feedforward loop together with a discrete time 'chattering-free' feedback loop. Such a use of the neural network with a sliding mode learning algorithm is considered to be a new approach to adaptive control of a non-linear robot system. The advantage of the proposed control scheme prevails over those conventional model based control scheme since no precise knowledge of mathematical model is necessary. The algorithm was verified by experiments where inverted pendulum with the additional mass-spring-damper load was used.

KEYWORDS: Nonlinear control; Sliding mode control; Neural network; Direct drive robots.

1 INTRODUCTION

Both, sliding mode control and neural network control are direct methods for the control of dynamic systems without the need of a mathematical model, in contrast to the conventional control which is an indirect method with the implementation of the mathematical model. In the control theory and industrial applications there are, however, few systematic procedures available for analysis and design of neural network control. In this paper, a neural network control with sliding mode is proposed to improve control system performance. The sliding mode is used to determine the learning procedure for neural network's weights and thus guarantee a linear input-output behavior of overall motion control system. With the aid of sliding modes it provides an effective design method of neural network control to ensure robustness.

Application of the sliding modes in the electrical machines and robotic manipulator control is widely discussed in many papers.[1–3] It was reported[1] that sliding mode in motion control exhibits unwanted motion imposed by the discontinuity of the control action. In robotic applications a control plant, a robotic manipulator with the torque vector as the control input has continuous control input and thus the discontinuity of control is artificially introduced. As the result the acceleration becomes discontinuous and consequently velocity exhibits high frequency oscillations – so called chattering. On this ground it was concluded that some changes should be introduced in the design methods.

Many different ideas were proposed to deal with that problem. One of them is so called 'boundary layer method' which uses a continuous function instead of the discontinuous one within the boundary layer.[1] This approach is based on the idea of the equivalence of the high gain systems and the systems with sliding mode. The drawback of this method is that there is no systematic method of choosing that function in the analysis of a system dynamics within the boundary layer is very complicated. Another idea is to divide a controller in two parts.[4,5] One part is the equivalent control or linear state feedback and the other part is the traditional VSS control. The application of this method has the problem of the abundant information about controlled plant needed to determine the VSS part.

In this paper a new method of suppressing chattering is presented. A basic idea of this method is that to eliminate the chattering effect with the equivalent control which is estimated by using on-line estimator realized by the neural network is used directly as a control input after reaching sliding mode manifolds. The reaching condition of sliding mode is assured by the application of the Lyapunov design method.

The presented control is composed of the equivalent control which has the meaning of feedforward control and the additional linear feedback control with respect to sliding mode manifolds. Thus the chattering of control is indeed eliminated. The control performance of this method depends heavily on the estimation of the equivalent control, therefore as a learning rule in on-line estimator the revised backpropagation algorithm is used to make fast convergence rate of gradient descent function $\partial E/\partial t$.

Neither the explicit calculation of the equivalent control nor high gain inside boundary layer are used. The parameters of the control depend on the plant gain matrix and the gradients of the sliding mode manifold. This control method is then applied to develop a control strategy for the trajectory tracking control of a robotic manipulator. In this way the system state is forced to

https://doi.org/10.1017/S0263574797000040 Published online by Cambridge University Press

remain on the selected manifold in the state space after reaching it. These manifolds are defined by the selection of the sliding mode functions. The desired dynamics is achieved by the selection of these manifolds. Such formalization of the control enables the selection of the controller having the same structure for common control problems in the motion control systems. To validate the proposed method an experimental inverted-pendulum-like system with an additional mass-spring-damper load to emulate effects of coupling between two robot joints is designed (Figure 1).

2 SLIDING MODE CONTROL

2.1 Control design

The design of the control system will be demonstrated for a nonlinear systems presentable in the form:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{d}(\boldsymbol{x}); \quad \boldsymbol{x} \in \mathbb{R}^n, \quad U \in \mathbb{R}^m, \quad rank(B(\boldsymbol{x})) = m \tag{1}$$

$\boldsymbol{d}(\boldsymbol{x})$ is the disturbance vector. The components of control input and of the vector $(d\boldsymbol{x}/dt)$ are assumed bounded. The goal is to find control $\boldsymbol{u}$ such that the motion of the system (1) is restricted to belong to the manifold $\boldsymbol{S}$,

$$\begin{aligned} \boldsymbol{S} &= \{\boldsymbol{x} : \boldsymbol{\sigma}_a(\boldsymbol{x}) - \boldsymbol{\varphi}(t) = \boldsymbol{\sigma}(\boldsymbol{x}, t) = 0\}, \\ \boldsymbol{\sigma}_a^T &= (\sigma_{a1}, \sigma_{a2}, \ldots, \sigma_{am}) \in \mathbb{R}^m, \\ \boldsymbol{\varphi}(t)^T &= (\varphi_1, \varphi_2, \ldots, \varphi_m) \in \mathbb{R}^m \end{aligned} \tag{2}$$

where $\sigma_{ai}(\boldsymbol{x})$ and $\varphi_i(t)$, $(i = 1, \ldots, m)$ are continuous functions. Functions $\sigma_{ai}(\boldsymbol{x})$ and $\varphi_i(t)$ have bounded first-time derivatives. They can be interpreted as the references to be tracked by the selected combinations $\sigma_{ai}(\boldsymbol{x})$ of the system's states. In the theory of the VSS with sliding modes it is proven that the closed-loop behavior of the system (1) on manifold (2) is determined by the selection of the manifold.[6]

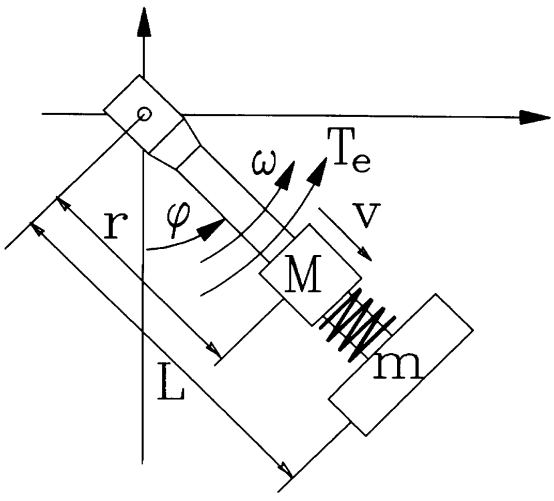


Fig. 1. Nonlinear load.

The condition under which the state will move toward and reach manifold $\boldsymbol{S}$ is known as reaching condition. By choosing the Lyapunov function candidate

$$V(\boldsymbol{x}, t) = \tfrac{1}{2}\boldsymbol{\sigma}^T\boldsymbol{\sigma} \tag{3}$$

a global reaching condition is given by: $\dot{V}(\boldsymbol{x}, t) < 0$ when $\boldsymbol{\sigma} \neq 0$ or equivalently for (3): $\boldsymbol{\sigma}^T\dot{\boldsymbol{\sigma}} < 0$. Various choices of $\dot{V}$ lead to different rates for $\boldsymbol{\sigma}$ and control law can be calculated directly from $\dot{\boldsymbol{\sigma}}$. For example, $\dot{\boldsymbol{\sigma}} = -\boldsymbol{K}\,\boldsymbol{sig}(\boldsymbol{\sigma})$, $\boldsymbol{sig}(\boldsymbol{\sigma}) = (sig\sigma_1, sig\sigma_2, \ldots, sig\sigma_m)$ leads to constant rate with discontinuous control and $\dot{\boldsymbol{\sigma}} = -\boldsymbol{D}\boldsymbol{\sigma}$ to proportional rate with continuous control, if $\boldsymbol{K}$ and $\boldsymbol{D}$ are chosen as positive definite matrix.

In the following sections the sliding mode design assuming that control is discontinuous will be outlined following results in references 6 and 8. Then a new design of a continuous control which will ensure the motion of the system (1) on the manifold (2) like the discontinuous control does will be discussed in details.

2.2 Discontinuous control law

It has been proven in the theory of variable structure systems;[6] that a discontinuous control

$$u_i = \begin{cases} u_i^+(\boldsymbol{x}, t) & \text{if} \quad \sigma_i(\boldsymbol{x}) > 0 \\ u_i^-(\boldsymbol{x}, t) & \text{if} \quad \sigma_i(\boldsymbol{x}) < 0 \end{cases} \quad i = 1, \ldots, m \tag{4}$$

provides the existence of the sliding mode motion for the system (1) on the manifold (2). According to the so-called equivalent control method, a fictitious continuous control is calculated as a solution of the system of the algebraic equation $d(\boldsymbol{\sigma}(\boldsymbol{x}, \ \boldsymbol{u} = \boldsymbol{u}_{eq}))/dt = 0$ and denoted as the equivalent control. Then it should be substituted into system's description (1) and the equations of motion can be determined as $d\boldsymbol{x}/dt = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{B}(\boldsymbol{x})\boldsymbol{u}_{eq} + \boldsymbol{d}(\boldsymbol{x})$. For the system (1) with control (4) on the manifold (2) with

$$\boldsymbol{\sigma} = G\boldsymbol{x}; \quad \boldsymbol{G} = \partial\boldsymbol{\sigma}/\partial\boldsymbol{x}, \quad rank\boldsymbol{G} = m \tag{5}$$

application of the equivalent control methods gives

$$\boldsymbol{u}_{eq} = -(\boldsymbol{GB})^{-1}\boldsymbol{G}(\boldsymbol{f} + \boldsymbol{d}) \tag{6}$$

and consequently the equation of motion are forced to satisfy

$$\boldsymbol{\sigma}(\boldsymbol{x}) = 0 \tag{7}$$

The sliding mode exists if $\boldsymbol{u}_{eq}$ is unique and if reaching conditions $u_i^+ < u_{ieq} < u_i^-$, $(i = 1, \ldots, m)$ are satisfied. If control (4) can be applied to the system's motion, after initial transient reached manifold $\boldsymbol{S}$, it will be restricted to belong to the manifold $\boldsymbol{S}$ and the system's states will satisfy equation (7). The selection of the function $\boldsymbol{\sigma}(\boldsymbol{x})$ plays important role in the closed-loop dynamics. Equations of motion (7) describe a $(n - m)$ order dynamical system. They do not depend on control and are determined merely by the properties of the control

https://doi.org/10.1017/S0263574797000040 Published online by Cambridge University Press

plant and the equations of the surfaces $\sigma_i = 0$, $(i = 1, \ldots, m)$.[6,8]

2.3 Continuous control law

The alternative well known approach to handle dynamic discrepancies is a continuous approximation of discontinuous control.[1] It should be noted that a high slope in the middle part of the linear approximation function used instead of the hysteresis function may result in excitation of the unmodeled dynamics as well, and the trade-off between accuracy an robustness must be achieved. Similar to the definition adopted in references 9 and 10, any motion of the system (1) that occurs in the ε-vicinity of the manifold (2) on which the sliding mode motion can exist with control (4), will be referred to as the sliding mode motion. This reflects the essence of the sliding mode design method. The control is selected to ensure the stability of the solution $\boldsymbol{\sigma}(\boldsymbol{x}, t) = 0$. Another essential feature of systems with sliding modes is that the reaching time from any initial condition to the ε-vicinity of the manifold $\boldsymbol{S}$ is finite. In this framework the design of the control system should ensure the stability on the sliding mode manifold $\boldsymbol{S}$, while the selection of the manifold will provide the means to determine the equations of motion.

The stability conditions for selected control must be examined first. This selection should ensure the stability of the system's motion in the origin of the subspace, whose coordinates are distances from the sliding mode manifold. For system (1) and the selected manifold (2) the first choice is the Lyapunov function in a quadratic form of control error (3). The solution $\boldsymbol{\sigma}(\boldsymbol{x}, t) = 0$ will be stable if the time derivative of the Lyapunov function can be expressed as a negative definite function. It can be obtained:

$$\dot{V} = -\boldsymbol{\sigma}^T \boldsymbol{D} \boldsymbol{\sigma};\ \boldsymbol{D} - \textit{positive definite matrix} \tag{8}$$

From (3, 8) the equation of desired system motion in reaching mode, when system states are not in sliding-mode, can be obtained as:

$$\dot{\boldsymbol{\sigma}} = -\boldsymbol{D}\boldsymbol{\sigma} \tag{9}$$

From (5) and (1) follows

$$\dot{\boldsymbol{\sigma}} = \boldsymbol{G}\boldsymbol{f} + \boldsymbol{G}\boldsymbol{B}\boldsymbol{u} + \boldsymbol{G}\boldsymbol{d} \tag{10}$$

With the combination of (9) and (10) the *desired continuous control* $\boldsymbol{u}^d$, which satisfies (9), can be calculated:

$$\boldsymbol{u}^d = \boldsymbol{u}_{eq} + \boldsymbol{\Delta u} \tag{11}$$

where $\boldsymbol{\Delta u} = -(\boldsymbol{G}\boldsymbol{B})^{-1}\boldsymbol{D}\boldsymbol{\sigma}$ and $\boldsymbol{u}_{eq}$ is given by (6).

Control (11) is continuous and chattering can not occur in this system. For calculation of control (11) information about equivalent control is needed. If system parameters are not known exactly, this solution must be modified.

Using simple manipulations, (11) can be rewritten in another form more practical for implementation, especially in the discrete-time systems. From (10) the equivalent control in implicit form can be expressed as: $\boldsymbol{u}_{eq} = (\boldsymbol{G}\boldsymbol{B})^{-1}(\boldsymbol{G}\boldsymbol{f} + \boldsymbol{G}\boldsymbol{d}) = (\boldsymbol{G}\boldsymbol{B})^{-1}\dot{\boldsymbol{\sigma}} - \boldsymbol{u}$. The results of short algebra can be written as[11]

$$\boldsymbol{u}^d = \boldsymbol{u} - (\boldsymbol{G}\boldsymbol{B})^{-1}(\boldsymbol{D}\boldsymbol{\sigma} + \dot{\boldsymbol{\sigma}}) \tag{12}$$

Expression (12) determines $\boldsymbol{u}^d$, which satisfies (9), as a function of implemented control signal $\boldsymbol{u}$. Control (12) is a continuous function everywhere except in the points of discontinuity of the function $\boldsymbol{\sigma}(\boldsymbol{x}, t)$.

In the discretized form, the continuous first derivative of the switching function can be replaced by it's first step approximation in the difference: $d\boldsymbol{\sigma}/dt \approx (\boldsymbol{\sigma}(k) - \boldsymbol{\sigma}(k-1))/T$. Desired continuous control $\boldsymbol{u}^i(k)$ is calculated as in the discrete time expression:

$$\boldsymbol{u}^d(k) = \boldsymbol{u}(k) - (\boldsymbol{G}\boldsymbol{B}\boldsymbol{T})^{-1}((\boldsymbol{E} + T\boldsymbol{D})(\boldsymbol{\sigma}(k) - \boldsymbol{\sigma}(k-1))) \tag{13}$$

The stability condition of the algorithm is fulfilled if $\boldsymbol{D}$ is selected so, that all eigenvalues of $(\boldsymbol{E} - T\boldsymbol{D})$ are within the unity circle of z-plane. Our system will, generally speaking, reach the sliding mode manifold $\boldsymbol{S}$ in infinite time, but we can define the ε vicinity of the manifold which is reached in finite time.

3 NEURAL NETWORK FEEDFORWARD CONTROL

Neural networks are used for the approximation of non-linear functions. It was shown,[12] that a neural network having enough neurons in at least one hidden layer with appropriate values of initial weights is able to approximate the input-output behavior of any function of practical interest. It is important to note, that neural networks can be regarded as tools for pattern recognition, since they actually perform and learn the mapping from input data to output data.

The main backdraft to use of neural networks in the control applications is the use of classical backpropagation algorithm that enables only a slow and inefficient learning. With the use of sliding mode based learning function it is possible to improve learning in order to increase its speed and robustness.[5]

Control (12) for system (1) ensures the stability of motion on the manifold $\boldsymbol{S}$ and after reaching manifold control (12) is equal to equivalent control: $\boldsymbol{u}^d = \boldsymbol{u}_{eq}$. For the calculation of control (12) information about the equivalent control in it's implicit form is needed $(\boldsymbol{u}_{eq} = (\boldsymbol{G}\boldsymbol{B})^{-1}\dot{\boldsymbol{\sigma}} - u)$, so this solution is not practical for the implementation. In this study we present a design method of on-line estimator which estimates a part of equivalent control containing a nonlinear system matrix $\boldsymbol{f}$, input matrix uncertainty $\boldsymbol{\Delta B}(\boldsymbol{x})\boldsymbol{u}$ and load disturbance $\boldsymbol{d}$ as a lineary combined nonlinear function by the use of a neural network's most powerful ability, that is, function of approximation.[13–16]

$$\boldsymbol{G}\boldsymbol{N}(\boldsymbol{x}) \approx \boldsymbol{G}(\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\Delta B}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{d}(\boldsymbol{x})) \tag{14}$$

In order to realize the control (12), unknown function in this control will have to be estimated with two-layer neural network (Figure 3). The linearly combined

https://doi.org/10.1017/S0263574797000040 Published online by Cambridge University Press

nonlinear function $\boldsymbol{f}(\boldsymbol{x}) + \Delta\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{d}(\boldsymbol{x})$ is replaced by the mapping of the neural network $\boldsymbol{N}$.

The actual input-matrix $\boldsymbol{B} = \bar{\boldsymbol{B}} + \Delta\boldsymbol{B}$ is replaced by estimate on $\bar{\boldsymbol{B}}$ and unknown part $\Delta\boldsymbol{B}$. The control (12) then has a form:

$$\boldsymbol{u}^d = (\boldsymbol{G}\bar{\boldsymbol{B}})^{-1}\left(\boldsymbol{G}\boldsymbol{N}(\boldsymbol{x}) - \frac{d\boldsymbol{\varphi}}{dt}\right) - (\boldsymbol{G}\bar{\boldsymbol{B}})^{-1}(\boldsymbol{D}\boldsymbol{\sigma} + \dot{\boldsymbol{\sigma}}) \quad (15)$$

Using (15) as a control input, the time derivative of Lyapunov function (3) along the system trajectory is evaluated as:

$$\frac{dV}{dt} = \boldsymbol{\sigma}^T(\boldsymbol{G}(\boldsymbol{f}(\boldsymbol{x}) + \Delta\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{d}(\boldsymbol{x})) - \boldsymbol{G}\boldsymbol{N}(\boldsymbol{x})) - \boldsymbol{\sigma}^T\boldsymbol{D}\boldsymbol{\sigma} < 0 \quad (16)$$

From above equation we suppose that the neural network can learn to satisfy next condition:

$$\|\boldsymbol{G}(\boldsymbol{f}(\boldsymbol{x}) + \Delta\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{d}(\boldsymbol{x})) - \boldsymbol{G}\boldsymbol{N}(\boldsymbol{x})\| < \|\boldsymbol{D}\boldsymbol{\sigma}\| \quad (17)$$

The difference among estimated dynamics $\boldsymbol{N}(\boldsymbol{x})$ and actual dynamics $(\boldsymbol{f}(\boldsymbol{x}) + \Delta\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{d}(\boldsymbol{x}))$ will become zero in sliding mode. In reaching mode we will use $\boldsymbol{D}\boldsymbol{\sigma} + \dot{\boldsymbol{\sigma}}$ as the criteria function for learning of the neural network.

$$\lim_{t\to\infty}(\boldsymbol{G}\boldsymbol{N}(\boldsymbol{x}) - \boldsymbol{G}(\boldsymbol{f}(\boldsymbol{x}) + \Delta\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{d}(\boldsymbol{x}))) = 0 \Rightarrow \boldsymbol{D}\boldsymbol{\sigma} + \dot{\boldsymbol{\sigma}} \to 0 \quad (18)$$

The function:

$$\boldsymbol{E} = \tfrac{1}{2}(\boldsymbol{D}\boldsymbol{\sigma} + \dot{\boldsymbol{\sigma}})^2 \quad (19)$$

has to be minimized by the neural network.

Results can be used in the learning function of the neural network. The improved Backpropagation learning algorithm contains a design parameter $\boldsymbol{G}$ that determines manifold $\boldsymbol{S}$ as described in (2). The control system is shown in Figure 2.

In our work a two-layer feedforward neural network, as shown in Figure 3 was used. It's model can be described with the use of following equations:

$$y_i^l = g_i^l\left(\sum_{j=1}^{k} w_{ij}^l a_j^l + b_i^l\right) \quad (20)$$

$$g_i^l(net_i^l) = \tanh(net_i^l) = \frac{2}{1 + e^{-2net_i^l}} - 1 \quad (21)$$

where a_j^l is the j-th input of the l-th layer, y_i^l is the output

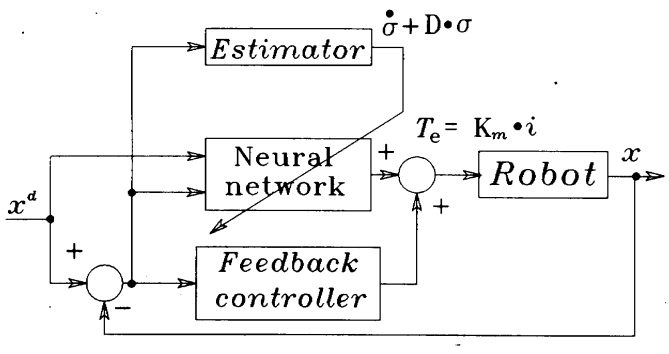


Fig. 2. Sliding-mode neural network controller.

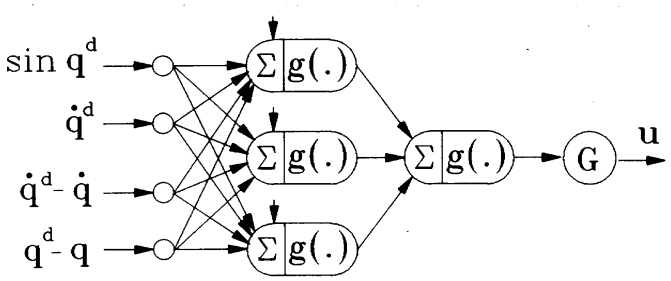


Fig. 3. Neural network.

of the i-th neuron in the l-th layer, w_{ij}^l is the weight connecting j-th input and the i-th neuron in the l-th layer and b_i^l is the bias of the i-th neuron of the l-th layer of the neural network. Learning proceeds using gradient descent method as described in (22) where η denotes the learning rate.

$$\Delta w_{ij}^l = -\eta\frac{\partial E}{\partial w_{ij}^l} \quad (22)$$

where:

$$\frac{\partial E}{\partial w_{ij}^l} = \frac{\partial E}{\partial net_i^l}\frac{\partial net_i^l}{\partial w_{ij}^l} = \frac{\partial E}{\partial net_i^l}a_j^l = \frac{\partial E}{\partial y_i^l}\frac{\partial y_i^l}{\partial net_i^l}a_j^l = \frac{\partial E}{\partial y_i^l}g_i^{l\prime}(net_i^l)a_j^l \quad (23)$$

For the output layer, denoted by L following equations can be used:

$$\frac{\partial E}{\partial y_i^l} = \frac{\partial E}{\partial N} = \frac{1}{2}\frac{\partial}{\partial N}(\boldsymbol{G}(\boldsymbol{f}(\boldsymbol{x}) + \Delta\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{d}(\boldsymbol{x})) - \boldsymbol{G}\boldsymbol{N}(\boldsymbol{x}))^2 = \boldsymbol{G}(\boldsymbol{f}(\boldsymbol{x}) + \Delta\boldsymbol{B}(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{d}(\boldsymbol{x})) - \boldsymbol{G}\boldsymbol{N}(\boldsymbol{x})(-g_m) = \boldsymbol{G}(\boldsymbol{D}\boldsymbol{\sigma} + \dot{\boldsymbol{\sigma}})g_m \quad (24)$$

$$\Delta w_{ij}^l = \eta g_m(\boldsymbol{D}\boldsymbol{\sigma} + \dot{\boldsymbol{\sigma}})g'(net^L)a_j^l \quad (25)$$

For the hidden layer learning proceeds using classical Backpropagation learning rule as described in references 16 and 17.

At this point it is important to note that the initial weights of the neural network have to enable the system to satisfy the demand for the negative definite Lyapunov function derivative. We propose the use of small initial weights thus making the feedforward control torque contribution of the neural network insignificant until it has obtained enough "knowledge" to ensure the stable operation.

The stability of the control system is assured by the feedback controller. The neural network does not solve the stability problems by itself, it only improves the quality of operation – the position tracking.

As the criteria function for the quality of learning SSE (Sum Square Error) described in was used, since it is the

https://doi.org/10.1017/S0263574797000040 Published online by Cambridge University Press

most usual criteria used in other author's work and can thus serve as a comparison to it.

$$\mathrm{SSE} = \sum (q^d - q)^2 \tag{26}$$

4 TRAJECTORY TRACKING ROBOT CONTROL

4.1 Control strategy

The manipulator dynamic model is represented by a collection of n second-order nonlinear scalar differential equations

$$\bar{J}_i(\boldsymbol{q})\ddot{q}_i + h_i(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) = \tau_i \tag{27}$$

$$h_i = (J_{ii}(\boldsymbol{q}) - \bar{J}_i(\boldsymbol{q}))\ddot{q}_i + \sum_{j=1, j\neq i}^{n} J_{ij}(\boldsymbol{q})\ddot{q}_j + C_i(\boldsymbol{q}, \dot{\boldsymbol{q}}) + G_i(\boldsymbol{q}) + F_i(\dot{\boldsymbol{q}}) \tag{28}$$

where $\boldsymbol{J}(\boldsymbol{q})$ is $n \times n$ inertia matrix; $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, $\boldsymbol{G}(\boldsymbol{q})$ and $\boldsymbol{F}(\dot{\boldsymbol{q}})$ are n dimensional vectors representing Coriolis and centrifugal forces, the gravity loading, and the friction; $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ are $1 \times n$ vectors of joint angular position, velocity and acceleration; and $\boldsymbol{\tau}$ is the n joint torque vector. The subscript i refers to the i-th joint. $\bar{J}_i$ is the known varying effective inertia at the i-th joint and is always positive due to the positive-definiteness of $\boldsymbol{J}$. So $\bar{\boldsymbol{J}}(\boldsymbol{q})$ can be choosen as a constant diagonal matrix. Equation (27) is the *reduce order* dynamic model of the i-th joint (subsystem) with the joint torque $\tau_i(t)$ as the input and the joint angle $q_i(t)$ as the output. The term h_i, given by (28), is treated as a "disturbance torque" by the i-th joint controller ($i = 1, \ldots, n$) and contains unknown parts of the inertial, gravity, friction, Coriolis and centrifugal torques for the i-th joint, as well as the inertial coupling effects from the other joints. The dynamics of a DC motor or a DC equivalent of an AC motor with resolved commutation and field generation, can be represented by a first order differential equation

$$L_i\dot{i}_i = u_i - e_i \tag{29}$$

where L_i is the motor inductance and e_i represents all voltage drops from resistance, back EMF, and for AC motors also equivalent voltages due to inexact commutation, etc. Let us assume a linear connection between the measurable actuator current i_i and the torque τ_i ($\tau_i = K_{mi}i_i$). The *full order* controlled plant will be represent by

$$\begin{bmatrix} \dot{\boldsymbol{i}} \\ \ddot{\boldsymbol{q}} \\ \dot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ \bar{\boldsymbol{J}}(\boldsymbol{q})^{-1}\boldsymbol{K}_m & 0 & 0 \\ 0 & \boldsymbol{I}_{n\times n} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{i} \\ \dot{\boldsymbol{q}} \\ \boldsymbol{q} \end{bmatrix} + \begin{bmatrix} 0 & \boldsymbol{L}^{-1} \\ \bar{\boldsymbol{J}}(\boldsymbol{q})^{-1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{h} \\ \boldsymbol{e} \end{bmatrix} + \begin{bmatrix} \boldsymbol{L}^{-1} \\ 0 \\ 0 \end{bmatrix} \boldsymbol{u} \tag{30}$$

In order to obtain smooth mechanical motion of the robot mechanism a continuous trajectory with values q_i^d, $\dot{q}_i^d$ and $\ddot{q}_i^d$ is prescribed. The switching function which determines the mechanical motion (2) is chosen to be second order and a function of angular position, velocity and acceleration errors for each joint is

$$\sigma_i = (\ddot{q}_i^d - \ddot{q}_i) + K_{vi}(\dot{q}_i^d - \dot{q}_i) + K_{pi}(q_i^d - q_i) \tag{31}$$

where K_{vi} and K_{pi} are constants that determine, in the sliding mode ($\sigma_i = 0$), the damping and the maximum frequency of the decentralized prescribed dynamics of second order. For the practical control implementation the measured quantities are state variables q_i and $\dot{q}_i$. The acceleration signal $\ddot{q}_i$ is not measurable and can be obtained by double differentiation of the angular position q_i; but is contaminated by the measurement noise to such a degree that it can no longer be used.

4.2 Experimental results

A robotic manipulator must have at least three-degrees of freedom in order to move to an arbitrary point in the space. Two-degrees of freedom, however, are sufficient for examining the validity of the control method, since two-linkage manipulator dynamics has interactions such as Coriolis and centrifugal force. In this study an inverted pendulum as shown in Figure 1 is used. To introduce the effects of interaction forces and the effects of the unmodeled dynamics, an additional mass-spring-damper load was used.

The dynamic behavior of the manipulator is described with two nonlinear second order differential equations:

$$\dot{\varphi} = \omega \tag{32}$$

$$\ddot{\varphi} = \frac{1}{J_m + m \cdot L^2 + M \cdot r^2}(T_e - B_m \cdot \omega - 2 \cdot M \cdot v \cdot \omega - m \cdot g \cdot L \cdot \sin\varphi - M \cdot g \cdot r \cdot \sin\varphi) \tag{33}$$

$$\dot{r} = v \tag{34}$$

$$\dot{r} = \frac{1}{M}(M \cdot r \cdot \omega + M \cdot g \cdot \cos\varphi - B_r \cdot v - K_r \cdot (L - r)) \tag{35}$$

where φ is the link's angular position, ω is the angular velocity and T_e represents the applied torque. The mass m is concentrated at the top of the manipulator length L. The constant B_m is the viscous damping coefficient, J_m is the induction motor rotor inertia, whereas g is the gravity constant. The fundamental manipulator control problem is to determine the algorithm for generating the joint torque T_e that drives the joint position $\varphi(t)$ to follow closely the reference position trajectory $\varphi_d(t)$. r is the unknown position of the additional link, v is the velocity of the second link's movement, M is the second link's mass, B_r is the translatorial damping coefficient and K_r describes the stiffness coefficient of the used spring. Those parameters of the system were unknown. T_e represents applied torque.

An AC induction motor (7.5 kW), mounted directly on the manipulator link (Figure 1) is used as actuator. Induction motor is fed by a three phase voltage controlled PWM inverter operating at 10 kHz. A 2500 pulse/rev. shaft encoder is used to sense the output

https://doi.org/10.1017/S0263574797000040 Published online by Cambridge University Press

position. The velocity ω is calculated by the differentiation of the position signal. The microcomputer used is a Motorola 68000, supported by transporter T805 thus allowing parallel programming in high level program language C.[18,19]

4.3 Continuous path control

In order to examine the validity of proposed control method, the whole system is forced to move along the desired trajectories, namely, it is under the continuous path control. The desired trajectoreis are illustrated on the Figure 4.

The desired position ($\sin \varphi_d$), the desired velocity (ω_d), the position error ($\varphi_d - \varphi$) and the velocity error ($\omega_d - \omega$) were used as inputs of the neural network. Compared to the work described in reference 20, where the desired position φ_d, the actual position φ, the desired velocity ω_d and the position error ($\varphi_d - \varphi$) were used as the neural network inputs, better information about gravity forces and system dynamics was gained. The desired values of velocity and position were used because of the noise present in actual values.

The system's task was to accelerate the rotor from the position $\varphi = 0[rad]$ (hanging hammer) with a $\sin^2$ shaped acceleration with the amplitude of $\dot{\omega} = 50[rad/s^2]$ in order to reach the velocity of $\omega = 8[rad/s]$, to stop it for 1[s] at the position of $\varphi = 4.7[rad]$, $(3 \cdot \pi/2)$, and then to return it to the position of $\varphi = 0[rad]$ (with the same acceleration and velocity).

The use of sliding mode network controller yielded satisfying results. Figure 5 shows the position error, and Figure 6 the velocity error of the sliding mode neural network controller. Actuator current i_q^R, which is proportional to the torque, is shown in Figure 7. The output of the neural network is featured on the Figure 8.

The steady state position error is within one pulse of the incremental encoder. Also in the phase of movement the position tracking is well performed. The velocity error signal includes a high rate of noise, caused by the operation of differentiation which was used for the calculation of actual velocity. As we can see in the Figure 8, the contribution of the neural network to the control torque is significant. It actually provides all the torque necessary to compensate for gravity forces, whereas it can also contribute in the compensation of inertial forces.

Criteria function $\boldsymbol{SSE} = \sum (\varphi^d - \varphi)^2$ is shown on the Figure 9. The convergence is fast – it can be observed that the most important part of learning is performed within the first cycle of learning, but also improving its operation in some later cycles. Due to the fast convergence we can argue that the learning procedure is robust to parameter variation even in very time-dependent applications like robot control.

5 CONCLUSION

This study proposes a method for neural network control design with sliding mode. Neural network control has been formulated to become a class of VSS control. Sliding modes facilitate determining the best values for parameters in neural learning rules. The Lyapunov function and the boundary layer have been employed to satisfy the reaching condition and to avoid chattering which has been the main obstacle for use of sliding mode control in practical applications. Instead of computer simulations, experiments have been conducted to validate the proposed method. The success of the present method lies in the fact that neural network control inherently resembles VSS control, since both control actions vary with divided regions on phase planes.

Sliding mode control function ensures the stability of the motion on the manifold. After reaching the manifold the control is equal to equivalent control and thus it is easy to analyze. A design of on-line estimator, which estimates a part of previous control by the use of a neural network's powerful ability – function approximation – is described.

The dynamic error is under control because asymptotic stability of motion and learning is guaranteed. The steady state position error is within one encoder pulse.

The learning proceeds smoothly and the vibrations at the initialization of learning are decreased. Random

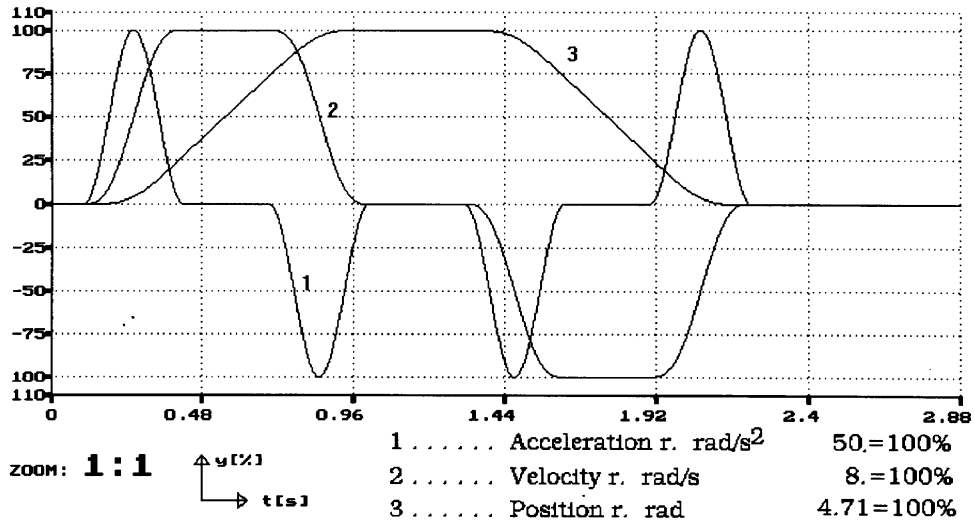


Fig. 4. Desired trajectories of acceleration, velocity, and position.

https://doi.org/10.1017/S0263574797000040 Published online by Cambridge University Press

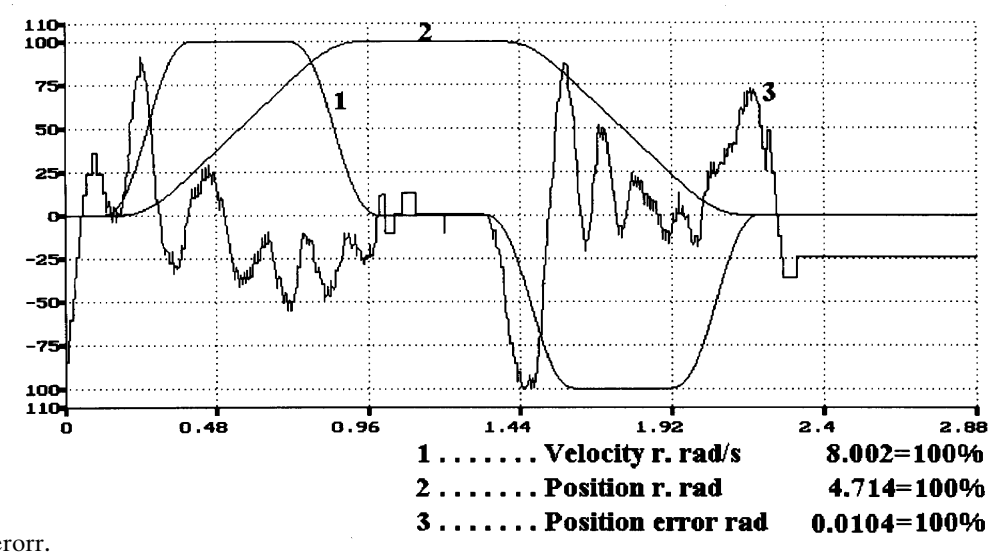


Fig. 5. Position erorr.

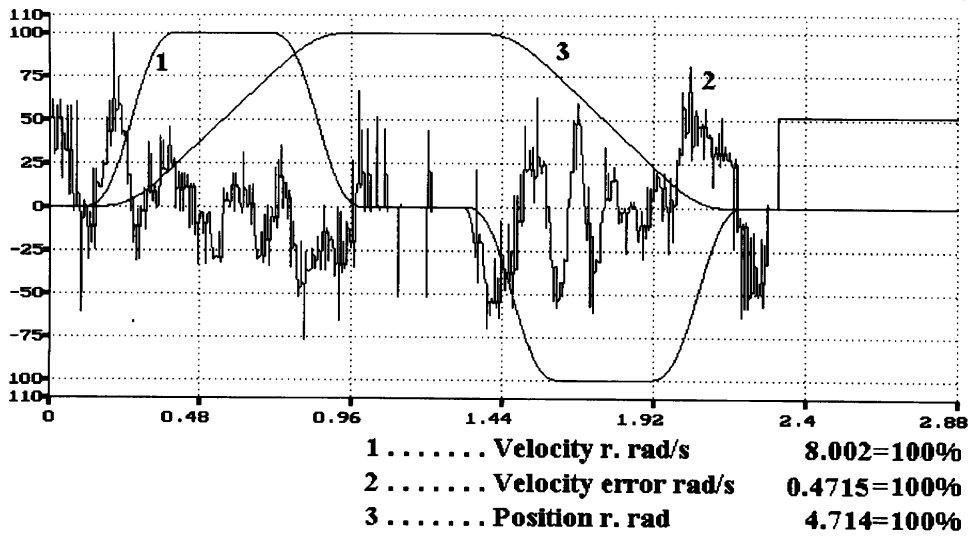


Fig. 6. Velocity error.

initial weights in the range: $-10^3 < w_{ij} < 10^{-3}$ can be used. In the first phase of learning the stability has to be ensured with the feedback controller, if the neural network has not received any previous information concerning the values of weights. Weights were rewritten with their old values in case of mistakes – i.e. completely wrong learning.

The possibility of use on robots with more D.O.F.s was

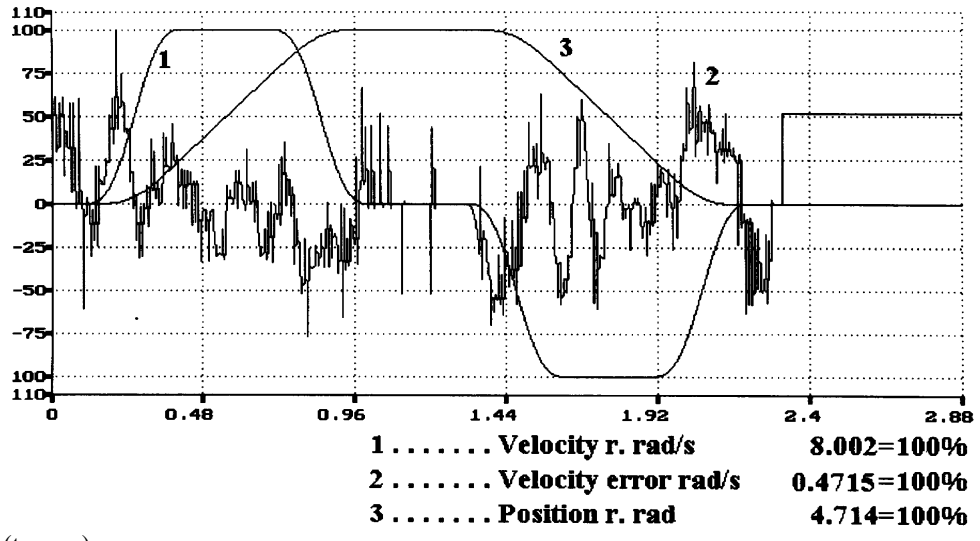


Fig. 7. Current i_q^R (torque).

https://doi.org/10.1017/S0263574797000040 Published online by Cambridge University Press

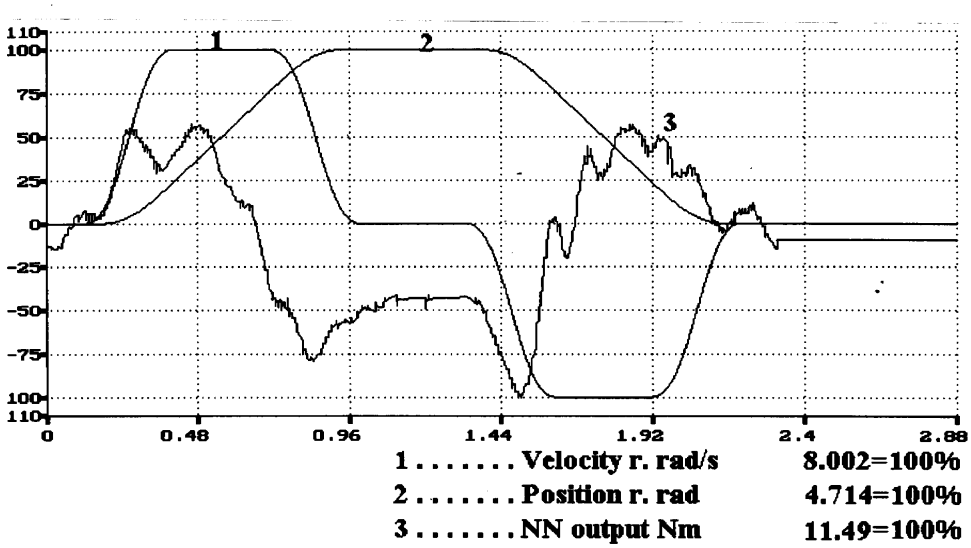


Fig. 8. Neural network output.

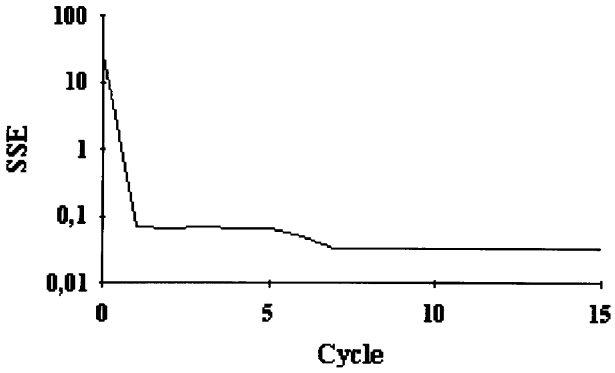


Fig. 9. SSE – Criteria function.

shown by adding the second "link" in the form of mass-spring-damper load. Thus coupling and additional unmodeled dynamics were introduced in the system. Since the position of the second link was unknown, results can also be used to prove the good performance of an independent manipulator controller in each separate joint. In addition, obtained control law is very simple and available for on-line computer control, which is also confirmed through actual experiments.

References

1. J.J. Slotine and S. Shastry, "Tracking control of nonlinear systems using sliding surfaces, with application to robot manipulators" *Int. J. Contr.* **38,** No. 2, 465–492 (1983).
2. W.J. Wang and G.H. Wu, "Variable structure control design on discrete-time systems – another point viewpoint" *Control-theory and Advanced Technology* **8,** No. 1, 1–16 (1992).
3. K. Jezernik, "Robust chattering free sliding mode control of servo drives" *Int. J. Electronics* **80,** No. 2, 169–179 (1996).
4. Z. Fengxi and D.G. Fisher, "Continuous sliding mode control" *Int. J. Control* **55,** No. 2, 313–327 (1992).
5. H. Morioka, K. Wada, A. Šabanović and K. Jezernik, "Neural Network based chattering – free Sliding Mode Control" *Proceedings of SICE Annual Conference,* Japan (1995) pp. 1303–1308.
6. V.U. Utkin, *Sliding Modes in Control and Optimization,* (Springer-Verlag, Berlin, 1992).
7. Y. Itkis, *Control Systems of Variable Structure,* (Wiley, New York, 1976).
8. B. Draženović, "The invariance conditions in variable structure systems" *Automatica* **5,** 287–295 (1969).
9. A. Šabanović, K. Jezernik and K. Wada, "Chattering-free sliding modes in robotic manipulators control" *Robotica* **14,** part 1 17–29 (1995).
10. V.I. Utkin, "Sliding mode control design and chattering problem" *Electrotechnical Review,* Ljubljana (SLO) **60,** Nos. 2–3, 75–81 (1993).
11. K. Jezernik, B. Curk and J. Harnik, "Observer based Sliding Mode Control of a Robotic Manipulator" *Robotica* **12,** Part 5, 443–448 (1994).
12. K. Hornik, M. Stinchombe and H. White, "Multi-layer Feedforward Networks are Universal Approximators" *Neural Networks* **2,** 359–366 (1989).
13. R. Šafarič and K. Jezernik, "Trajectory Tracking Neural Network Controller for Robot Mechanism and Lyapunov Theory of Stability" *IROS'94 proceedings,* Munich, Germany (September 1994) **Vol. 1,** pp. 626–633.
14. R. Šafarič, A. Hace and K. Jezernik, "Transputer Based Trajectory Tracking Neural Network Controller for a Robot Mechanism" *Proceedings of the IECON'94, 20th International Conference on Industrial Electronics, Control and Instrumentation,* Bologna, Italy (September 5–8, 1994) pp. 794–799.
15. S. Khemaissia and A.S. Morris, "Neuro-adaptive control of robotic manipulators" *Robotica* **11,** part 5, 465–473 (1993).
16. T. Ozaki, T. Suzuki, T. Furuhashi, S. Okuma and Y. Uchikawa, "Trajectory Control of Robotic Manipulators Using Neural Networks" *IEE Transaction on Industrial Electronics* 38, No. 3, 195–201 (June, 1991).
17. T. Fukuda and T. Shibata, "Research Trends in Neuromorphic Control" *J. Robotics and Mechatronics* **2,** No. 4, 220–234 (1992).
18. K. Jezernik and V. Volčanjk, "VSC Robust Control of an IM Servodrive" *Proceedings of the IECON'94, 20th International Conference on Industrial Electronics, Control and Instrumentation,* September 5–9 1994, Bologna, Italy (September 5–9, 1994) pp. 627–633.
19. G.M. Asher and M. Summer, "Parallel Processing for Vector Control of Induction Motors" *EPE Journal* **3,** No. 1, 27–38 (March, 1993).
20. J.S. Ko and M.J. Youn, "Robust Digital Position Control of BLDD Motors Using Neural Network with State Feedback" *3rd International Workshop on Advanced Motion Control,* pp. 852–861, Berkeley, California USA (March, 1994) pp. 852–861.

https://doi.org/10.1017/S0263574797000040 Published online by Cambridge University Press