ADAPTIVE SLIDING-MODE CONTROL WITH GAUSSIAN NETWORK

Lei Ma [*] **Klaus Schilling** [*] **Christian Schmid** [**]

[*] *Lehrstuhl Informatik VII, Bayerische Julius-Maximilians Universität Würzburg, Germany*
[**] *Lehrstuhl für Automatisierungstechnik und Prozessinformatik, Ruhr-Universität Bochum, Germany*

Abstract: This paper is concerned with the adaptive sliding-mode control of a class of nonlinear systems with model uncertainties. A direct adaptive sliding-mode control scheme is presented. A network of Gaussian radial basis functions with variable weights was used to compensate the model uncertainties. A new growing scheme of this network is proposed. It starts with a loose structure in order to reduce the computational effort. More nodes are added to the network progressively in order to improve the transient behaviour. The adaptive law developed using the Lyapunov synthesis approach guarantee the stability of the overall control scheme, even in the presence of modelling error. The performance of the control scheme is illustrated by simulation studies with convincing results.

Keywords: Adaptive control, Sliding-mode control, Radial base function networks, Gaussian functions, Neural Networks, Nonlinear systems

1. INTRODUCTION

The robustness in the face of model uncertainties in a control system with a sliding-mode controller (SMC) is due to the high-frequency switching term of the control, which in practice equals to a continuous high-gain control (Utkin *et al.*, 1999). The switching gain has to be higher than the known norm of the uncertainties. When the uncertainties grow beyond this bound, the switching controller is no longer capable of maintaining the sliding mode, and the system loses robustness to uncertainties and disturbances. A more conservative estimation of the uncertainties may help to maintain the stability but leads to a higher control gain and more control effort. Furthermore, this may also lead to problems with parasitic dynamics of the system (Young and Kokotovic, 1982). It is then necessary to extend the standard SMC to an adaptive one (Slotine and Coetsee, 1986). However, classical parameter estimation methods and adaptive control schemes require that the system model be linearly parameterised and that the nonlinearities are exactly known. In general, this is not always the case.

In this paper, a direct adaptive SMC scheme is proposed based on a network of Gaussian radial basis functions (GRBF) for control of nonlinear systems with model uncertainties and disturbances. The reason of choosing a GRBF network is that its outputs are linear combinations of the neurons outputs, such that the stability of the overall system can be easier achieved. The properties of a GRBF arranged on a regular lattice for approximation of nonlinear functions are well studied in (Sanner and Slotine, 1992) and (Lewis *et al.*, 1999). Given an estimation of the upper bound of the function to be approximated, the error bound can be calculated analytically.

The main drawback of such networks is that the neurons are located on a fixed lattice. Therefore, the quality of the function approximation depends on the density of the lattice. However, a high density of the lattice as well as a large operating range leads to a network of very large size, and this causes a high computational effort. In order to reduce the network size, some improvements have been proposed by neglecting the neurons, which are located far from the system trajectory and, thus, have little influence on the function approximation (Fabri and Kadirkamanathan, 1996). However this neglect leads to a new source of disturbance that is very difficult to model.

In this paper, a new network growing scheme is proposed, where the network starts with a very loose regular grid, so that the approximation error bound can be estimated. New nodes based on subgrids of higher density are inserted into the network according to the control error. The idea of subgrids was first proposed by (Liu *et al.*, 1999), but no quantitative description of the approximation error was given there. In this paper, a switching control term is applied to compensate the network approximation error. Therefore, an excellent transient performance and a relatively small network size can be expected. The remainder of this paper is organised as follows. In Section 2, the problem statement is presented for a class of uncertain nonlinear systems where the uncertainties fulfil the matching condition. In Section 3, the adaptive SMC scheme based on a growing GRBF network is developed. The stability of the overall system is analysed. The performance of the presented control scheme is demonstrated in section 4 by simulation studies of the tracking control of a two-link robot manipulator.

2. PROBLEM STATEMENT

Consider the following multivariable time-varying system described in nonlinear phase-variable canonical form (NPVCF) as defined by (Sommer, 1980)

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x},t) + \boldsymbol{B}(\boldsymbol{x},t)\,\boldsymbol{u}(t)\ . \qquad (1)$$

$\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, defined as

$$\boldsymbol{x}(t) = \left[x_1\ \dots\ x_1^{(n_1-1)}\ \dots\ x_m\ \dots\ x_m^{(n_m-1)}\right]^{\mathrm{T}},$$
$$n = n_1 + \dots + n_m \qquad (2)$$

and $\boldsymbol{u}(\boldsymbol{x},t) \in \mathbb{R}^m$ is the control vector. The vector $\boldsymbol{f}(\boldsymbol{x},t) \in \mathbb{R}^n$ and the input matrix $\boldsymbol{B}(\boldsymbol{x},t) \in \mathbb{R}^{n\times m}$ are nonlinear functions and are assumed to be differentiable with respect to $\boldsymbol{x}$ and t. Both functions

$$\boldsymbol{f}(\boldsymbol{x},t) = \begin{bmatrix} \boldsymbol{f}_1(\boldsymbol{x},t) \\ \boldsymbol{f}_2(\boldsymbol{x},t) \\ \vdots \\ \boldsymbol{f}_m(\boldsymbol{x},t) \end{bmatrix}, \quad \boldsymbol{B}(\boldsymbol{x},t) = \begin{bmatrix} \boldsymbol{B}_1(\boldsymbol{x},t) \\ \boldsymbol{B}_2(\boldsymbol{x},t) \\ \vdots \\ \boldsymbol{B}_m(\boldsymbol{x},t) \end{bmatrix}$$

are composed of m subvectors and submatrices, respectively,

$$\boldsymbol{f}_j(\boldsymbol{x},t) = \begin{bmatrix} \dot{x}_j(t) \\ \ddot{x}_j(t) \\ \vdots \\ x_j^{(n_j-1)}(t) \\ f_j(\boldsymbol{x},t) \end{bmatrix}, \quad \boldsymbol{B}_j(\boldsymbol{x},t) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{b}_j^{\mathrm{T}}(\boldsymbol{x},t) \end{bmatrix}$$

for $j = 1, 2, \dots, m$.

$\boldsymbol{b}_j^{\mathrm{T}}(\boldsymbol{x},t) = \left[0\ \dots\ 0\, b_{j,j}(\boldsymbol{x},t)\ \dots\ b_{j,m}(\boldsymbol{x},t)\right]$ is the last row of matrix $\boldsymbol{B}_j(\boldsymbol{x},t)$, where $b_{j,j}(\boldsymbol{x},t) \neq 0$ must be satisfied to be controllable. If the system is not given in NPVCF, one has to find a transformation to transform it into the canonical form. However, this is not always possible. Conditions for that are given in (Sommer, 1980) and a more detailed analysis with respect to the controllability can be found in (Isidori, 1995).

If measurements of the state vector $\boldsymbol{x}$ are available, the control objective is, to find a control $\boldsymbol{u}(\boldsymbol{x},t)$ such that the state $\boldsymbol{x}(t)$ tracks the desired trajectories

$$\boldsymbol{x}_{\mathrm{d}}(t) = \left[x_{\mathrm{d}_1}\ \dots\ x_{\mathrm{d}_1}^{(n_1-1)}\ \dots\ x_{\mathrm{d}_m}\ \dots\ x_{\mathrm{d}_m}^{(n_m-1)}\right]^{\mathrm{T}},$$

which are known functions of time and have bounded derivatives. In terms of the tracking error

$$\boldsymbol{e}(t) = \boldsymbol{x}(t) - \boldsymbol{x}_{\mathrm{d}}(t) \qquad (3)$$

the control task is, to find a proper $\boldsymbol{u}(\boldsymbol{x},t)$ such that $\boldsymbol{e} \to 0$ as $t \to \infty$.

Consider model uncertainties in the system and rewrite equation (1) by dropping all arguments as

$$\dot{\boldsymbol{x}} = (\boldsymbol{f}_0 + \Delta\boldsymbol{f}) + (\boldsymbol{B}_0 + \Delta\boldsymbol{B})\boldsymbol{u}\ , \qquad (4)$$

where $\boldsymbol{f}_0 \in \mathbb{R}^n$ and $\boldsymbol{B}_0 \in \mathbb{R}^{n\times m}$ are the nominal parts of $\boldsymbol{f}$ and $\boldsymbol{B}$, respectively. $\Delta\boldsymbol{f} \in \mathbb{R}^n$ and $\Delta\boldsymbol{B} \in \mathbb{R}^{n\times m}$ represent the uncertainties in the system model. $\boldsymbol{B}_0$ must have full rank. Assume that the system uncertainties fulfil the matching conditions (Drazenović, 1969), namely the uncertain dynamics can be lumped together into one vector, the system can be described as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}_0 + \boldsymbol{B}_0\boldsymbol{u} + \boldsymbol{B}_0\boldsymbol{h}\ , \qquad (5)$$

where $\boldsymbol{h} \in \mathbb{R}^m$ represents the unknown dynamics of the system. In order to have a metric for describing the tracking error dynamics, a time-varying sliding surface

$$\boldsymbol{s}(t) = \boldsymbol{G}\,\boldsymbol{e}(t) \qquad (6)$$

with $\boldsymbol{s} \in \mathbb{R}^m$ and $\boldsymbol{G} \in \mathbb{R}^{(m\times n)}$ is introduced. The constant gain matrix $\boldsymbol{G}$ must have full rank and be chosen such that when $\boldsymbol{s} \to 0$, the tracking error $\boldsymbol{e}$ also tends to zero. The motion of the system on the sliding surface depends only on the design of $\boldsymbol{G}$, which can be chosen by classical methods (Slotine and Coetsee, 1986).

3. ADAPTIVE SLIDING-MODE CONTROL

3.1 The controller

For the unknown $\boldsymbol{h}$ from equation (5) an estimation

$$\hat{\boldsymbol{h}} = \boldsymbol{W}\,\boldsymbol{\Phi}(\boldsymbol{x}) \tag{7}$$

is used, which is approximated by a GRBF network. The network has n inputs, m outputs and L hidden-layer neurons

$$\boldsymbol{\Phi}(\boldsymbol{x}) = \left[\phi_1(\|\boldsymbol{x}-\boldsymbol{\xi}_1\|)\ \dots\ \phi_L(\|\boldsymbol{x}-\boldsymbol{\xi}_L\|)\right]^{\mathrm{T}} . \tag{8}$$

$\boldsymbol{W} \in \mathbb{R}^{m\times L}$ is the matrix of the output weights and

$$\phi_i(\|\boldsymbol{x}-\boldsymbol{\xi}_i\|) = \mathrm{e}^{-\frac{1}{2\sigma_i^2}(\boldsymbol{x}-\boldsymbol{\xi}_i)^{\mathrm{T}}(\boldsymbol{x}-\boldsymbol{\xi}_i)} \tag{9}$$

are the Gaussian functions with the node positions $\boldsymbol{\xi}_i$ in the lattice. Using the network arranged on a regular lattice (Sanner and Slotine, 1992) at the beginning, there exists an optimal output-weight matrix $\boldsymbol{W}^*$ and a positive scalar ε_0 such that

$$\boldsymbol{h} = \hat{\boldsymbol{h}}^* + \boldsymbol{\varepsilon}, \qquad \|\boldsymbol{\varepsilon}\| \le \varepsilon_0 \tag{10}$$

with

$$\hat{\boldsymbol{h}}^* = \boldsymbol{W}^*\,\boldsymbol{\Phi}(\boldsymbol{x}) . \tag{11}$$

By increasing the number of neurons in the network, the upper bound of the approximation error ε_0 can be reduced to be arbitrarily small. Define $\boldsymbol{W}$ as the estimation of $\boldsymbol{W}^*$, and $\tilde{\boldsymbol{W}} = \boldsymbol{W}^* - \boldsymbol{W}$, the adaptive SMC with a GRBF network is described by the following theorem:

Theorem 1. For a dynamical system described by equation (5), the tracking error converges asymptotically to zero with the control

$$\boldsymbol{u} = \boldsymbol{u}_{\mathrm{eq}} + \boldsymbol{u}_{\mathrm{smc}}, \tag{12}$$

with

$$\boldsymbol{u}_{\mathrm{eq}} = -\left[\boldsymbol{G}\boldsymbol{B}_0\right]^{-1}\boldsymbol{G}(\boldsymbol{f}_0 - \dot{\boldsymbol{x}}_{\mathrm{d}}) - \hat{\boldsymbol{h}} \tag{13}$$

and

$$\boldsymbol{u}_{\mathrm{smc}} = -\rho\frac{(\boldsymbol{G}\boldsymbol{B}_0)^{\mathrm{T}}\boldsymbol{s}}{\|(\boldsymbol{G}\boldsymbol{B}_0)^{\mathrm{T}}\boldsymbol{s}\|}, \qquad \rho > \varepsilon_0 . \tag{14}$$

Proof: Consider the Lyapunov function candidate

$$V = \frac{1}{2}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{s} + \frac{1}{2}\gamma\,\mathrm{tr}(\tilde{\boldsymbol{W}}\tilde{\boldsymbol{W}}^{\mathrm{T}}) , \tag{15}$$

it follows that

$$\dot{V} = \boldsymbol{s}^{\mathrm{T}}\dot{\boldsymbol{s}} + \gamma\,\mathrm{tr}(\tilde{\boldsymbol{W}}\dot{\tilde{\boldsymbol{W}}}^{\mathrm{T}}) . \tag{16}$$

The first term $\boldsymbol{s}^{\mathrm{T}}\dot{\boldsymbol{s}}$ of the above equation with the definition of the control $\boldsymbol{u}$ from equation (12) and the system equation (5) can be rewritten as

$$\begin{aligned}
\boldsymbol{s}^{\mathrm{T}}\dot{\boldsymbol{s}} &= \boldsymbol{s}^{\mathrm{T}}\boldsymbol{G}(\dot{\boldsymbol{x}} - \dot{\boldsymbol{x}}_{\mathrm{d}})\\
&= \boldsymbol{s}^{\mathrm{T}}\boldsymbol{G}(\boldsymbol{f}_0 + \boldsymbol{B}_0\boldsymbol{u} + \boldsymbol{B}_0\boldsymbol{h}) - \boldsymbol{s}^{\mathrm{T}}\boldsymbol{G}\dot{\boldsymbol{x}}_{\mathrm{d}}\\
&= \boldsymbol{s}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{B}_0(\boldsymbol{h} - \hat{\boldsymbol{h}} - \rho\frac{(\boldsymbol{G}\boldsymbol{B}_0)^{\mathrm{T}}\boldsymbol{s}}{\|(\boldsymbol{G}\boldsymbol{B}_0)^{\mathrm{T}}\boldsymbol{s}\|})\\
&= \boldsymbol{s}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{B}_0(\boldsymbol{W}^*\boldsymbol{\Phi} + \boldsymbol{\varepsilon} - \boldsymbol{W}\boldsymbol{\Phi} - \rho\frac{(\boldsymbol{G}\boldsymbol{B}_0)^{\mathrm{T}}\boldsymbol{s}}{\|(\boldsymbol{G}\boldsymbol{B}_0)^{\mathrm{T}}\boldsymbol{s}\|})\\
&= \boldsymbol{s}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{B}_0(\tilde{\boldsymbol{W}}\boldsymbol{\Phi} + \boldsymbol{\varepsilon} - \rho\frac{(\boldsymbol{G}\boldsymbol{B}_0)^{\mathrm{T}}\boldsymbol{s}}{\|(\boldsymbol{G}\boldsymbol{B}_0)^{\mathrm{T}}\boldsymbol{s}\|}) .
\end{aligned} \tag{17}$$

Choosing the adaptation law for the network weights as

$$\dot{\boldsymbol{W}} = -\dot{\tilde{\boldsymbol{W}}} = \frac{1}{\gamma}\left(\boldsymbol{\Phi}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{B}_0\right)^{\mathrm{T}} , \tag{18}$$

it follows that

$$\begin{aligned}
\gamma\,\mathrm{tr}(\tilde{\boldsymbol{W}}\dot{\tilde{\boldsymbol{W}}}^{\mathrm{T}}) &= -\mathrm{tr}(\tilde{\boldsymbol{W}}\boldsymbol{\Phi}\boldsymbol{s}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{B}_0)\\
&= -\boldsymbol{s}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{B}_0\tilde{\boldsymbol{W}}\boldsymbol{\Phi},
\end{aligned} \tag{19}$$

and equation (16) will be

$$\begin{aligned}
\dot{V} &= \boldsymbol{s}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{B}_0\boldsymbol{\varepsilon} - \rho\boldsymbol{s}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{B}_0\frac{(\boldsymbol{G}\boldsymbol{B}_0)^{\mathrm{T}}\boldsymbol{s}}{\|(\boldsymbol{G}\boldsymbol{B}_0)^{\mathrm{T}}\boldsymbol{s}\|}\\
&= \boldsymbol{s}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{B}_0\boldsymbol{\varepsilon} - \rho\left\|\boldsymbol{s}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{B}_0\right\|\\
&\le \left\|\boldsymbol{s}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{B}_0\right\|\left(\|\boldsymbol{\varepsilon}\| - \rho\right) .
\end{aligned} \tag{20}$$

If $\|\boldsymbol{\varepsilon}\| \le \varepsilon_0$ and $\rho > \varepsilon_0$, then $\dot{V} < 0$ for $\|\boldsymbol{s}\| \ne 0$. It follows that $\boldsymbol{s}$ and $\tilde{\boldsymbol{W}}$ converge to zero asymptotically and, thus, the tracking error $\boldsymbol{e}$ converges to zero. ■

The overall control scheme is shown in Figure 1. $\boldsymbol{u}_{\mathrm{eq}}$ is the component of the manipulated variable that is responsible to preserve the sliding mode with the condition $\dot{\boldsymbol{s}} = \boldsymbol{0}$. As it depends on the unknown $\boldsymbol{h}$ its estimate $\hat{\boldsymbol{h}}$ is used. $\boldsymbol{u}_{\mathrm{smc}}$ is the switching component of the SMC.

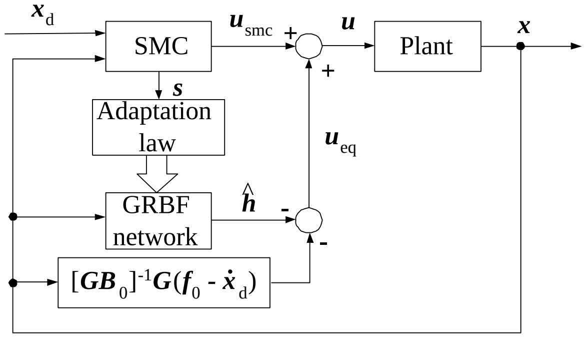

Fig. 1. Block diagram of the adaptive sliding-mode tracking control with a GRBF network

3.2 The GRBF network

The network starts with very few neurons. The approximation error $\boldsymbol{\varepsilon}$ from equation (10) is upper bounded by ε_0, which can be calculated given a conservative upper estimation of the uncertainties (Sanner and Slotine, 1992). New neurons are progressively added according to the novelty of the system states. Considering the computational effort, the idea of a growing network with several subgrids (Liu *et al.*, 1999) was adopted. The centres of the neurons are arranged on regular lattices and the widths are determined by heuristic methods. The adaptation of the network is performed only by the determination of the output weights. Therefore, it remains linear in the parameters. Then one can expect a fast convergence of the adaptation. The crossings of the subgrids provide only potential positions for the new neurons. Here, a popular idea has been adopted, where the neurons whose centres are included in a hypersphere of the actual inputs will be activated. In practice, for approximation of a nonlinear function defined on a compact set, as shown in Figure 2 for a two-dimensional case, the network starts with a very loose 2×2 base grid, where only 4 neurons are arranged on the edges. The arabian numbers denote the grids. Hyperspheres in the two-dimensional case are circles of different radii. Figure 2 also shows that for the current system state "$*$" only those neurons from different subgrids are activated which lie within the corresponding hyperspheres.

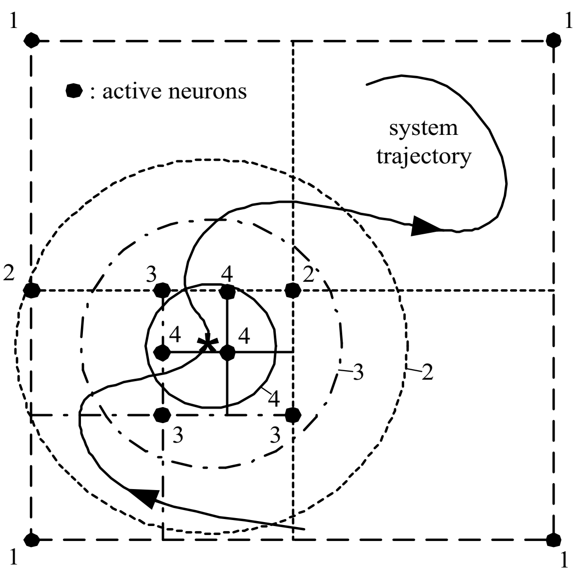

Fig. 2. Phase-plane portrait with an example of active neurons at the current system state for a lattice with three subgrids

It is very difficult to determine analytically a proper amount of necessary neurons. A popular idea is to add new neurons according to the tracking error (Fabri and Kadirkamanathan, 1996). This leads to the problem that unnecessary neurons might be included. When the initial condition of the system lies far from the desired trajectory and the transient period is relatively long, one would get a very large network. Though the network size can be reduced by including a pruning strategy (Liu *et al.*, 1999) (Li *et al.*, 2001) to delete superfluous neurons, the computational effort might be intermittently very large.

In this paper, a time-varying measure of the reasonable error bound is defined as

$$\Delta(t) = \begin{cases} \eta_1 e^{-\eta_2 t} & \text{for} \quad \|\boldsymbol{s}\| > \epsilon \\ \epsilon & \text{for} \quad \|\boldsymbol{s}\| \leq \epsilon \end{cases} , \tag{21}$$

where ϵ is the required accuracy, η_1 and η_2 are design parameters to be chosen. $\Delta(t)$ seeks to represent the available tracking accuracy during the transient period. This is based on the requirement that the tracking error should converge faster than some exponential function. New neurons are only inserted into the network when the sliding variable $\boldsymbol{s}(t)$ is larger than the current error bound $\Delta(t)$. For the design, the factor η_1 can be chosen as $\|\boldsymbol{s}(0)\|$. The exponent η_2 decides about the amount of neurons added to the network.

The introduction of new, denser "higher-order" subgrids has to satisfy the condition that the tracking error is larger than the current error bound $\Delta(t)$. Furthermore, the time period between adding of subgrids must be long enough. In (Liu *et al.*, 1999), 11 subgrids were used to control a SISO system. In this paper, much less subgrids are required. This is due to the use of a sliding-mode control term for the compensation of the approximation error. The neurons of "lower-order" subgrids, especially those of the base grid, can be treated as "global" approximators, which try to provide general information about the unknown function on the entire compact set. The neurons of the "higher-order" subgrids can then be treated as local approximators, which provide more details about the unknown function in a certain region.

4. SIMULATION RESULTS

A two-link manipulator model (Utkin *et al.*, 1999) is used to demonstrate the performance of the proposed control scheme. The dynamics of the planar manipulator can be expressed as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{r}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau} , \tag{22}$$

where $\boldsymbol{q} = \begin{bmatrix} q_1 & q_2 \end{bmatrix}^{\mathrm{T}}$ denotes the joint angles of the links. $\boldsymbol{M}(\boldsymbol{q}) \in \mathbb{R}^{2\times2}$ stands for the inertial mass matrix, $\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) \in \mathbb{R}^{2\times2}$ comprises Coriolis and centripetal forces, vector $\boldsymbol{r}(\dot{\boldsymbol{q}}) \in \mathbb{R}^2$ describes viscous friction, and $\boldsymbol{\tau} \in \mathbb{R}^2$ is the vector of torques applied to the joints. Equation (22) can be rewritten in the state-space form as

$$\begin{bmatrix} \dot{\boldsymbol{q}} \\ \ddot{\boldsymbol{q}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & -\boldsymbol{M}^{-1}\boldsymbol{C} \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \\ \dot{\boldsymbol{q}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{M}^{-1}\boldsymbol{r} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{M}^{-1} \end{bmatrix} \boldsymbol{\tau} . \tag{23}$$

Redefine the state vector as $\boldsymbol{x} = \begin{bmatrix} q_1 & \dot{q}_1 & q_2 & \dot{q}_2 \end{bmatrix}^{\mathrm{T}}$ and $\boldsymbol{u} = \boldsymbol{\tau}$, equation (23) can be easily transformed by permutations into the form of equation (5). The model of the manipulator has the following parameters:

$$\boldsymbol{M} = \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} , \quad \boldsymbol{r} = \begin{bmatrix} 2.5 \operatorname{sgn}(\dot{q}_1) + 1.5\, \dot{q}_1 \\ 1.5 \operatorname{sgn}(\dot{q}_2) + 0.7\, \dot{q}_2 \end{bmatrix} ,$$

$$\boldsymbol{C} = \begin{bmatrix} -l_1 l_2 m_2 \dot{q}_2 \sin(q_2) & -l_1 l_2 m_2 (\dot{q}_1 + \dot{q}_2) \sin(q_2) \\ l_1 l_2 m_2 \dot{q}_1 \sin(q_2) & 0 \end{bmatrix}$$

with

$$\begin{aligned} m_{11} &= l_1^2 (m_1 + m_2) + 2 m_{12} - m_{22} \ , \\ m_{12} &= m_{21} = m_{22} + l_1 l_2 m_2 \cos(q_2) \quad \text{and} \\ m_{22} &= l_2^2 m_2 \ . \end{aligned}$$

The length of the links is $l_1 = l_2 = 1$ and the masses $m_1 = 10$ and $m_2 = 1 + 0.4 \sin(t)$. The latter represents the variation of the payload. The elements of $\boldsymbol{M}$ vary as

$$\begin{aligned} 10 \leq m_{11} &\leq 15.6 \ , \\ 0 \leq m_{12} = m_{21} &\leq 2.8 \quad \text{and} \\ 0.6 \leq m_{22} &\leq 1.4 \ . \end{aligned}$$

The nominal values of the manipulator parameters used for the SMC design are $\boldsymbol{M}_0 = \begin{bmatrix} 12 & 1.5 \\ 1.5 & 1 \end{bmatrix}$ and $\boldsymbol{C}_0$, $\boldsymbol{r}_0$ both set to zero and from these follows

$$\boldsymbol{f}_0 = \begin{bmatrix} x_2 \\ 0 \\ x_4 \\ 0 \end{bmatrix} , \quad \boldsymbol{B}_0 = \begin{bmatrix} 0 & 0 \\ 0.1026 & -0.1538 \\ 0 & 0 \\ -0.1538 & 1.2308 \end{bmatrix} .$$

A GRBF network with 3 grids with a maximum of 16, 65 and 544 hidden neurons, respectively, was used during the simulation studies. The widths σ_i of the corresponding Gaussian functions of the different grids are π, $\pi/2$ and $\pi/4$, respectively. The hyperspheres have a radius of 1 and 0.5 for the second and the third subgrid, respectively. To avoid infinite-frequency switching of the control torques, a boundary layer (Slotine and Li, 1991) with $\psi = 0.02$ is introduced so that when $\|s\| < \psi$ the control $\boldsymbol{u}_{\mathrm{smc}} = -(1/\psi)\boldsymbol{s}$ is used instead of equation (14). The adaptation of the neural network is also blocked within the boundary layer. Other parameters used in the simulation studies are $\epsilon = \psi$, $\eta_1 = 1$, $\eta_2 = 0.45$, the adaptation rate $\gamma^{-1} = 5$ and $\rho = 1$. The latter is quite conservative according to the approximation error of the GRBF network, yet much lower than the switching gain required for a classical SMC.

The desired trajectories of the joint angles are $q_{\mathrm{d}_1} = \sin(t)$ and $q_{\mathrm{d}_2} = \cos(t)$, respectively. Figures 3 and 4 show details about the signals of the system. The tracking errors converge to 0.05 in less than $t = 4$, though errors at the start of the simulation were quite large. Figure 5 shows that the control effort is constrained in an acceptable range without high-frequency switching. Figure 6 shows the norm of the sliding surface $\boldsymbol{s}(t)$ as well as the error bound $\Delta(t)$. Only when $\|\boldsymbol{s}\| > \Delta$, new neurons are added to the network. Finally, 125 neurons were activated, namely 16 neurons of the base grid, 33 neurons of the 2nd subgrid and 76 of the 3rd one. Figure 7 shows the transient performance of the manipulator. Compared with a network having only the base grid, as shown in Figure 8, the transient performance of a network with subgrids is much better. Subgrids with more neurons reduce obviously the tracking errors during the transient period.

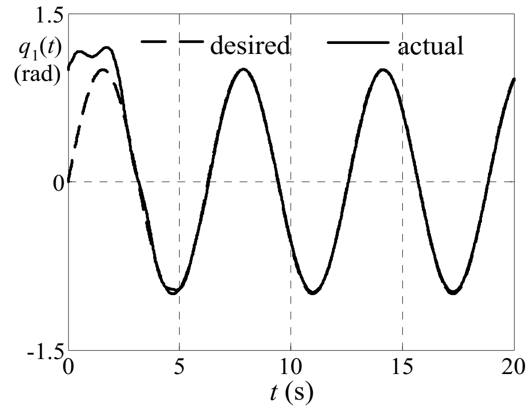

Fig. 3. Desired and actual joint angle of link 1

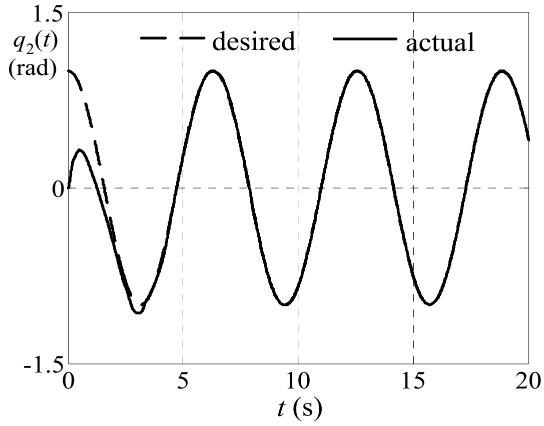

Fig. 4. Desired and actual joint angle of link 2

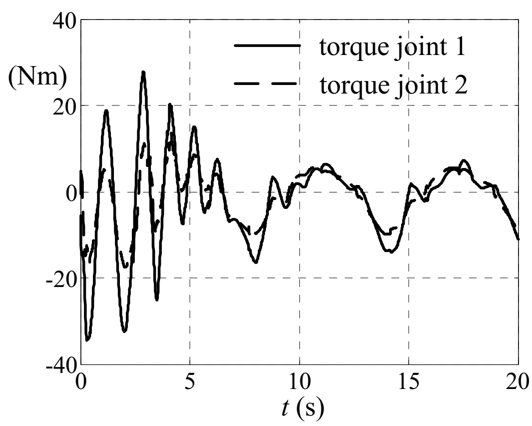

Fig. 5. Torques acting on the joints

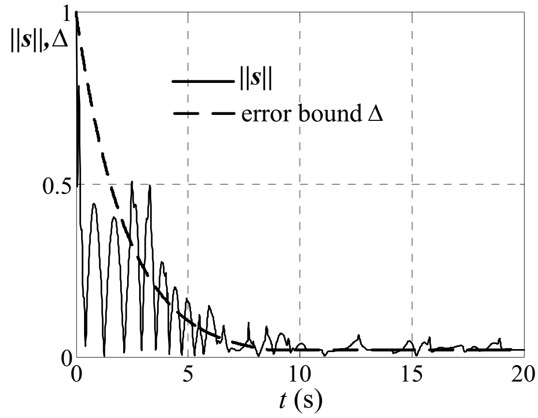

Fig. 6. $\|\boldsymbol{s}(t)\|$ and the error bound $\Delta(t)$

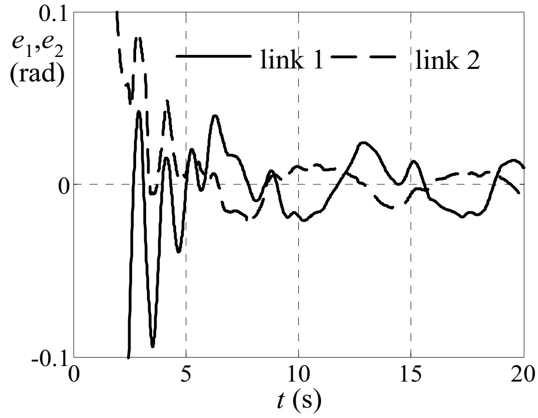

Fig. 7. Tracking errors of the manipulator with a network of 3 subgrids

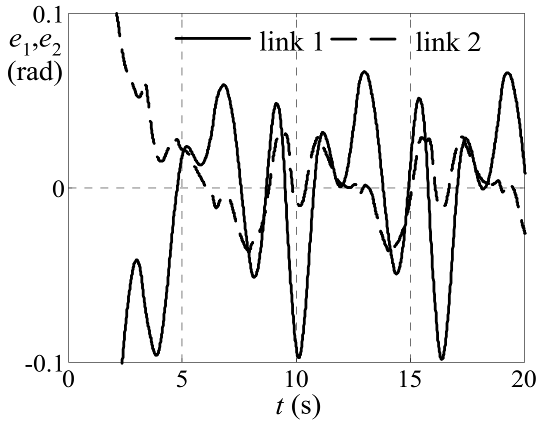

Fig. 8. Tracking errors of the manipulator with a network of the base grid

5. CONCLUSIONS

A direct adaptive sliding-mode control scheme with a network of Gaussian radial basis functions for control of nonlinear systems with matched uncertainties was proposed in this paper. A reasonable compromise between the error performance and the computational effort was achieved by a new subgrid growing strategy. The stability of the overall system is shown by the direct method of Lyapunov. Simulation studies with a two-link manipulator has shown the feasibility of this control scheme.

REFERENCES

Drazenović, B. (1969). The invariance conditions in variable structure systems. *Automatica* **5**(3), 287–295.

Fabri, S. and V. Kadirkamanathan (1996). Dynamic structure neural networks for stable adaptive control of nonlinear systems. *IEEE Transactions on Neural Networks* **7**(5), 1151–1167.

Isidori, A. (1995). *Nonlinear Control Systems.* third ed.. Springer-Verlag. Berlin.

Lewis, F.L., S. Jagannathan and A. Yesildirek (1999). *Neural Network control of Robot Manipulators and Nonlinear Systems.* Taylor & Francis. London.

Li, Y., N. Sundararajan and P. Saratchandran (2001). Neuro-controller design for nonlinear fighter aircraft maneuver using fully tuned RBF networks. *Automatica* **37**(8), 1293–1301.

Liu, G.P., V. Kadirkamanathan and S.A. Billings (1999). Variable neural networks for adaptive control of nonlinear systems. *IEEE Transactions on Systems, Man and Cybertics, Part C* **29**(1), 34–43.

Sanner, R.M. and J-J.E. Slotine (1992). Gaussian networks for direct adaptive control. *IEEE Transactions on Neural Networks* **3**(6), 837–863.

Slotine, J.J.E. and J.A. Coetsee (1986). Adaptive sliding controller synthesis for non-linear systems. *International Journal of Control* **43**(6), 1631–1651.

Slotine, J.J.E. and W.P. Li (1991). *Applied Nonlinear Control.* Prentice-Hall. Englewood Cliffs.

Sommer, R. (1980). Control design for multivariable non-linear time-varying systems. *International Journal of Control* **31**(5), 883–891.

Utkin, V.I., J. Guldner and J. Shi (1999). *Sliding mode control in electromechanical systems.* Taylor & Francis. London.

Young, K.K.D. and P.V. Kokotovic (1982). Analysis of feedback loop interaction with actuator and sensor parasitics. *Automatica* **18**(5), 577–582.