Neuro-hierarchical sliding mode control for a class of under-actuated systems

Dianwei Qian* and Xiangjie Liu

Department of Automation,
North China Electric Power University,
Beijing, 102206, China
E-mail: Dianwei.qian@ncepu.edu.cn
E-mail: liuxj@ncepu.edyu.cn
*Corresponding author

Jianqiang Yi and Chengdong Li

Institute of Automation,
Chinese Academy of Sciences,
Beijing, 100190, China
E-mail: Jianqiang.yi@ia.ac.cn
E-mail: Chengdong.Li@ia.ac.cn

Abstract: A neuro-hierarchical sliding mode controller is presented for a class of under-actuated systems with a stable equilibrium point. Such controller is combined with the concept of neural networks and the methodology of hierarchical sliding mode control. At first, the hierarchical sliding mode control law is designed for the class as follows. The system is divided into several subsystems and the sliding surface of every subsystem is defined. Then, the sliding surface of one subsystem is selected as the first layer sliding surface. The first layer sliding surface is then to construct the second layer sliding surface with the sliding surface of another subsystem. This process continues until the sliding surfaces of all the subsystems are included. The control law is derived from Lyapunov theorem. By aiming at unknown factors and uncertainties, the neural networks are designed to approximate the terms of the hierarchical sliding mode control law. The asymptotic stability of the entire sliding surfaces and the convergence of the network weights are proven theoretically. Simulation and physical experiment results show the controller's validity and robustness.

Keywords: sliding mode control; SMC; hierarchy; under-actuated systems; uncertainties; neural networks; NNs.

Reference to this paper should be made as follows: Qian, D., Liu, X., Yi, J. and Li, C. (2011) 'Neuro-hierarchical sliding mode control for a class of under-actuated systems', *Int. J. Modelling, Identification and Control*, Vol. 13, No. 4, pp.243–250.

Biographical notes: Dianwei Qian received his BE from the Hohai University, Nanjing, Jiangsu, China, in 2003. In 2005 and 2008, he received his ME from the Northeastern University, Shenyang, Liaoning, China, and his PhD from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, respectively. Currently, he is an Assistant Professor at the Department of Automation, North China Electric Power University, Beijing, China. His research interests include theories and applications of intelligent control, under-actuated system control, sliding-mode control, process control, etc.

Xiangjie Liu received his PhD in Electrical and Electronic Engineering from the Research Center of Automation, Northeastern University, Shenyang, China, in 1997. He was then in the postdoctoral programme in the Electric Power Research Institute (EPRI), Beijing, China, until 1999. He was an Associate Professor at EPRI and a Research Associate in the University of Hong Kong. He joined the National University of Mexico, Mexico City, in 2001, where he was a Professor in the Research Center of Instrumentation from 2003 to 2004. Currently, he is a Full Professor in the Department of Automation, North China Electric Power University, Beijing. His research areas include non-linear control, adaptive and predictive control, and intelligent control. He is a Guest Editor for the *Journal Neurocomputing*.

Jianqiang Yi received his BE from the Beijing Institute of Technology, Beijing, China, in 1985 and his ME and PhD from the Kyushu Institute of Technology, Kitakyushu, Japan, in 1989 and 1992, respectively. He worked as a Research Fellow at the Computer Software Design, Inc., Tokyo, Japan, from 1992 to 1994. From 1994 to 2001, he was the Chief Researcher and Chief Engineer at the MYCOM, Inc., Kyoto, Japan. Currently, he is a Full Professor at the Institute of

Copyright © 2011 Inderscience Enterprises Ltd.

Automation, Chinese Academy of Sciences, Beijing. His research interests include theories and applications of intelligent control, under-actuated system control, flight control, etc. He is an Associate Editor for the *IEEE Computational Intelligence Magazine* and *Journal of Innovative Computing, Information and Control.*

Chengdong Li received his BS and MS from the Shandong University, Jinan, China, in 2004 and 2007, respectively. He is currently working toward his PhD in the Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include intelligent control and computational intelligence.

This paper is a revised and expanded version of a paper entitled 'Neuro-hierarchical sliding mode control for a class of under-actuated systems' presented at The Sixth International Symposium on Neural Networks (ISNN 2009), Wuhan, China, 26–29 May 2009.

1 Introduction

Under-actuated systems are characterised by the fact that they have fewer actuators than the degrees of freedom to be controlled. Such systems arise in extensive applications (Spong, 1998). The class of under-actuated systems referred in this paper with a control input and multiple outputs is rather large, including Acrobots, inverted pendulum systems, ball-beam systems, overhead cranes, and single-input horizontal under-actuated manipulators, etc. They are often used for research on non-linear control and education in various concepts (Fang et al., 2003). There are three types of under-actuated systems in the class from the viewpoint of the property of equilibrium points. The first type has a stable equilibrium point such as overhead cranes, the second one has an unstable equilibrium point such as Acrobots, the third one has infinite equilibrium points such as single-input horizontal under-actuated manipulators. Here, we only focus on a part of the class that has a stable equilibrium point.

In recent years, there are increasing interests in the control problems of under-actuated systems (Yi et al., 2002; Zhang and Tarn, 2002; Reyhanoglu et al., 1999; Hussein and Bloch, 2008). The dynamics of under-actuated systems often contain non-linearities, non-holonomic constraints, couplings and so on. These make their control design difficult. Sliding mode control (SMC) is a powerful non-linear design method, which has been developed and applied for the last three decades (Kaynak et al., 2001). The SMC provides a good candidate for the control problems of under-actuated systems (Wang et al., 2004; Xu and Özgüner, 2008). But designing a conventional sliding surface is not appropriate for under-actuated systems, because the parameters of the sliding surface can't be obtained directly according to Hurwitz condition as linear systems (Wang et al., 2004). On the other hand, the computation of the equivalent control requires the exact knowledge of system dynamics and parameters. For partly known or uncertain systems, there exist some difficulties during the computation of the equivalent control, which may lead to an ineffective sliding mode controller. In this paper, we work at the structure design of the sliding surfaces for the class and how to utilise the control law for the systems with uncertainties.

Neural networks (NNs) are a developing intelligent method. NNs provide a good candidate for the computation of the equivalent control of such partly known or uncertain systems. In Kim and Oh (1995), Jezernik et al. (1997), Ertugrul and Kaynak (2000) and Tsai et al. (2004), neuro-sliding-mode-control methods were addressed. In those methods, NNs were used to calculate the equivalent control. An inherent problem is that such network architecture may lead to calculational burden as the inputs increase. Moreover, none of them gives the convergence analysis about their NNs and sliding surfaces in theory.

The structure characteristic of the class of under-actuated systems with a stable equilibrium point is that they can be divided into several subsystems. Based on this structure, the hierarchical sliding mode control (HSMC) was presented in Wang et al. (2004), Lin and Mon (2005) and Qian et al. (2008). Further, this structure provides an idea to simplify the network structure, which could solve the above problem associating with the computational burden. Therefore, we consider designing a neuro-hierarchical sliding mode controller for the class of under-actuated systems with a stable equilibrium point.

Based on the concept of NNs and the methodology of HSMC, the neuro-hierarchical sliding mode control (NHSMC) is presented in this paper. The reminders are organised as follows. In Section 2, the NHSMC law is designed. The asymptotic stability of the entire sliding surfaces and the convergence of the network weights are proven theoretically in Section 3. Simulation and physical experiment results in Sections 4 and 5 show the validity and the robustness of this method, respectively. Conclusions are derived in Section 6 at last.

2 Design of NHSMC

2.1 Design of HSMC

The state space equation of the class can be depicted by a canonical expression as the following form:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f_1(X) + b_1(X)u \\ \dot{x}_3 = x_4 \\ \dot{x}_4 = f_2(X) + b_2(X)u \\ \quad\vdots \\ \dot{x}_{2n-1} = x_{2n} \\ \dot{x}_{2n} = f_n(X) + b_n(X)u \end{cases} \tag{1}$$

Here, $X = [x_1, x_2, \ldots, x_{2n}]^T$ is the state variable vector; $f_i(X)$ and $b_i(X)$ ($i = 1, 2, \ldots, n$) are the non-linear functions of the state variables X, they are abbreviated as f_i and b_i; and u is the single control input.

Equation (1) is the normal form of the class. It could describe different systems by different n, f_i and b_i. If $n = 2$, it can represent Acrobots, single inverted pendulum systems, Pendubots, and cranes; if $n = 3$, it can express double inverted pendulum systems, double-pendulum type cranes, and double-pendulum systems; if $n = 4$, it can be considered as triple inverted pendulum systems, etc. In this paper, we only focus on a part of the class with a stable equilibrium point, such as cranes and pendulum systems.

Further, (1) can be divided into several subsystems according to the physical structure. For example, a series triple inverted pendulum system consists of four subsystems: upper pendulum, middle pendulum, lower pendulum, and cart. The state variables (x_{2i-1}, x_{2i}) can be treated as the states of the ith subsystem. The sliding surface of the ith subsystem is defined below:

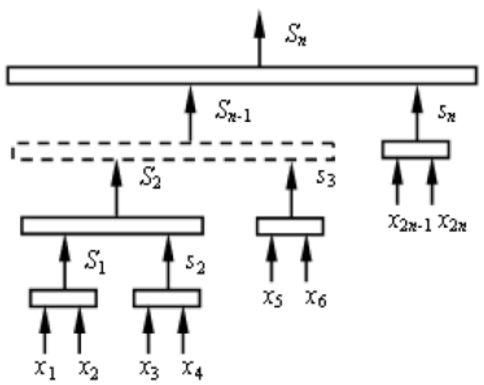
$$s_i = c_i x_{2i-1} + x_{2i} \tag{2}$$

Here, c_i is a positive constant. By differentiating s_i with respect to time t in (2), the equivalent control can be obtained from $\dot{s}_i = 0$ as:

$$u_{eqi} = -\left(c_i x_{2i} + \mathrm{f}_i\right) / \mathrm{b}_i \tag{3}$$

The structure of the hierarchical sliding surfaces is designed as follows. Without loss of generality, the sliding surface of the first subsystem s_1 is chosen as the first layer sliding surface S_1. Then, S_1 is used to construct the second layer sliding surface S_2 with s_2. This process continues until all of the sliding surfaces of the subsystems are included. Such structure is shown in Figure 1.

Figure 1 Hierarchical structure of the sliding surfaces



In the presented hierarchical structure, we know that the ith layer sliding surface includes the information of the ith subsystem sliding surface and the other i – 1 lower layer sliding surfaces. As a result, the ith layer sliding surface S_i, its control law u_i, and its Lyapunov function V_i can be defined as follows:

$$S_i = a_{i-1}S_{i-1} + s_i \tag{4}$$

$$u_i = u_{i-1} + u_{eqi} + u_{swi} \tag{5}$$

$$V_i(t) = S_i^2 / 2 \tag{6}$$

Here, a_{i-1} is a constant and $a_0 = u_0 = 0$.

Differentiating V_i with respect to time t in (6), and letting $\dot{S}_i = -k_i S_i - \eta_i \operatorname{sgn} S_i$ ($k_i > 0$; $\eta_i > 0$), in light of Lyapunov theorem, we could have the following HSMC law at the ith layer sliding surface. The detailed deduction could be found in Qian et al. (2008):

$$u_i = \frac{\sum_{r=1}^{i}\left(\prod_{j=r}^{i} a_j\right) b_r u_{eqr}}{\sum_{r=1}^{i}\left(\prod_{j=r}^{i} a_j\right) b_r} - \frac{k_i S_i + \eta_i \operatorname{sgn} S_i}{\sum_{r=1}^{i}\left(\prod_{j=r}^{i} a_j\right) b_r} \tag{7}$$

2.2 Approximation by NNs

In (7), the final control u_n could be derived from $n = i$. But we have to know the exact f_i and b_i to calculate u_n. But u_n can not be calculated directly for partly known or uncertain systems. This weak point restricts the applications of the HSMC. Aiming at such weak point, we attempt to make use of NNs to handle it. According to the mentioned physical characteristic of the class, the n + 1 NNs are designed and every network is to approximate a term of (7) in this subsection. From (7), the NHSMC law can be defined as:

$$u_n^{nn} = \sum_{i=1}^{n} u_{eqi}^{nn} + u_{\mathrm{sgn}}^{nn} \tag{8}$$

Here, u_n^{nn} is the NHSMC law at the nth layer sliding surface; u_{eqi}^{nn} is the ith network output to approximate the $[(\Pi_{j=r}^{i} a_j) \bullet b_r u_{eqr}] / [\Sigma_{r=1}^{i}(\Pi_{j=r}^{i} a_j) b_r]$ term; u_{sgn}^{nn} is used to approximate the $[-k_i S_i - \eta_i \operatorname{sgn} S_i] / [\Sigma_{r=1}^{i}(\Pi_{j=r}^{i} a_j) b_r]$ term.

Figure 2 Structure of the neuron to calculate u_{sgn}^{nn}

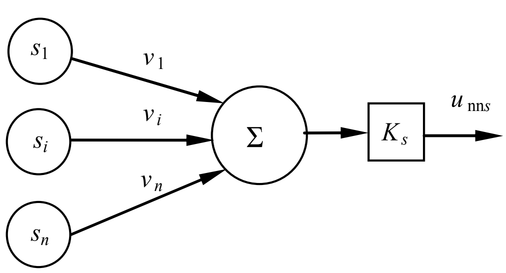
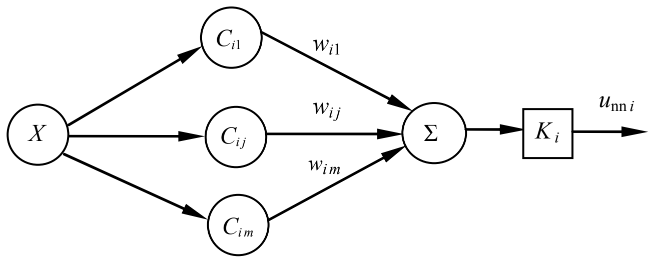


Figure 3 Structure of the RBF NN to calculate u_{eqi}^{nn}



The structures of the NNs used to calculate u_{sgn}^{nn} and u_{eqi}^{nn} are shown in Figure 2 and Figure 3, respectively. As Figure 2 has shown, the neuron is designed to calculate u_{sgn}^{nn}. The symbols in Figure 2 are defined as follows. s_i is the sliding surface of the ith subsystem; v_i is the connecting weight between the ith input unit and the neuron; K_s is the gain; u_{sgn}^{nn} is the neuron output. The network adopted in Figure 3 is RBF network, because RBF network owns the ability to approximate complex non-linear mapping directly from input-output data with a simple topological structure (Huang et al., 2005). The symbols in Figure 3 are defined as follows. $X = [x_1; x_2; \ldots; x_{2n}]^T$ is the input vector of the ith RBF network; m is the number of the hidden layer unit; u_{eqi}^{nn} is the output of the ith RBF network; w_{ij} is the connecting weight between the jth hidden layer unit and the output layer unit; K_i is the gain; the non-linear function of the hidden layer is the radial basis function, which is shown below:

$$\varphi_{ij}(X) = \exp\left(-\| X - C_{ij} \|^2 / \sigma_{ij}^2\right) \tag{9}$$

Here, C_{ij} and σ_{ij} are the centre and width of the jth hidden layer unit of the ith RBF network ($i = 1; \ldots; n; j = 1; \ldots; m$), respectively.

The gradient descent method is adopted to update the parameters of the neuron and the RBF NNs. In general, the cost function is the error between the actual output and the desired output. But there is no supervisory signal before the neuro-controller works (Tsai et al., 2004). Thus, other substitution should be found. In (7), the last term works when any system state deviates from the nth layer sliding surface. Moreover, the other n terms make the states of the subsystems slide on their own sliding surfaces. Based on the above viewpoints, the cost functions and the update formulas can be gotten in the following description.

For the neuron, the cost function J_{sgn} can be defined as:

$$J_{\mathrm{sgn}} = S_n^2 / 2 \tag{10}$$

(10) means minimising the distance between the entire state variables and the last layer sliding surface. Further, the update formula of the neuron can be deduced as:

$$\Delta v_i = -\mu \frac{\partial J_{\mathrm{sgn}}}{\partial v_i} = -\mu \bullet S_n \bullet s_i \tag{11}$$

Here, positive constant μ is learning rate.

As we have expressed, b_i and f_i are the non-linear functions of the state variables. Thus, the state vector X is selected as the input. For the ith RBF network, its cost function J_{eqi} is defined as:

$$J_{eqi} = s_i^2 / 2 \tag{12}$$

(12) means minimising the distance between the state variables of the ith subsystem and its own sliding surface. By the gradient descent method, the update formulas of the parameters of the ith RBF network can be deduced as:

$$\Delta w_{ij} = -\lambda_i \frac{\partial J_{eqi}}{\partial w_{ij}} = -\lambda_i \bullet s_i \bullet K_i \bullet \varphi_{ij}(X) \tag{13}$$

$$\Delta C_{ij} = -\lambda_i \frac{\partial J_{eqi}}{\partial C_{ij}} = -\lambda_i \bullet s_i \bullet K_i \bullet w_{ij} \bullet \varphi_{ij}(X) \bullet \left(X - C_{ij}\right) \bullet \sigma_{ij}^{-2} \tag{14}$$

$$\Delta \sigma_{ij} = -\lambda_i \frac{\partial J_{eqi}}{\partial \sigma_{ij}} = -\lambda_i \bullet s_i \bullet K_i \bullet w_{ij} \bullet \varphi_{ij}(X) \bullet \left\| X - C_{ij} \right\|^2 \bullet \sigma_{ij}^{-3} \tag{15}$$

Here, positive constant λ_i is learning rate of the ith RBF network.

3 Convergence analysis

The designed controller combines the concept of the HSMC and the methodology of the NNs. In this section, the stability of the sliding surfaces and the weight convergence of the neuron and the parallel RBF networks are detected.

Theorem 1: Consider a class of under-actuated systems with a stable equilibrium point as (1), design the entire sliding surfaces as (2) and (4), and adopt the control law as the NHSMC (8). If s_i and $\dot{s}_i$ are bounded, then the entire sliding surfaces are asymptotically stable, and the weights of the neuron and the parallel RBF NNs are bounded.

Proof: In the ith RBF network, for minimising the cost function (12), there exist:

$$\int_0^\infty s_i^2 < \infty \tag{16}$$

Thus, we have $s_i \in L_2$ (s_i is square integral).

According to the known condition that s_i and $\dot{s}_i$ are bounded, we can obtain:

$$s_i \in L_\infty \tag{17}$$

and

$$\dot{s}_i \in L_\infty \tag{18}$$

From (16), (17) and (18), we have $\lim_{t \to \infty} s_i = 0$ according to Barbalat's lemma.

From $\lim_{t\to\infty} s_i = 0$, we know that Δw_{ij}, ΔC_{ij}, $\Delta\sigma_{ij}$ are convergent to zero as $t \to\infty$ in (13), (14) and (15). Thus, the weights of the ith RBF network are bounded.

In the neuron, the following recursive formula could be gotten from (4):

$$S_i = \sum_{r=1}^{i}\left(\prod_{j=r}^{i} a_j\right)s_r \tag{19}$$

Here, $a_i = 1$. In light of $\lim_{t\to\infty} s_i = 0$, we could obtain:

$$\begin{aligned}\lim_{t\to\infty} S_i &= \lim_{t\to\infty}\sum_{r=1}^{i}\prod_{j=r}^{i} a_j s_r \\ &= \sum_{r=1}^{i}\prod_{j=r}^{i} a_j \lim_{t\to\infty} s_r \\ &= 0\end{aligned} \tag{20}$$

Equation (20) means that the ith layer sliding surface is asymptotically stable. From it, Δv_i is convergent to zero as $t \to \infty$ in (11). Thus, the weights of the neuron are bounded.

Remark: Theorem 1 is presented for the under-actuated systems with a stable equilibrium. Therefore, the premise that s_i and $\dot{s}_i$ are bounded can easily be satisfied. For instance, the upper and lower pendulums of a double pendulum system could satisfy this bounded assumption at its downward stable equilibrium, respectively. But for the under-actuated systems with an unstable equilibrium, the premise that s_i and $\dot{s}_i$ are bounded are hardly satisfied. Thus, how to use this NHSMC method solves control problems of the under-actuated systems with an unstable equilibrium is still an interesting field.

4 Simulation results

In this section, we shall demonstrate that this control strategy is applicable to an overhead crane system. Overhead crane is a typical under-actuated system with a stale equilibrium (Wang et al., 2004). It works in many places such as workshops and harbours to transport massive goods. The control objective of overhead crane systems is to transform the loads to the required position as fast and as accurately as possible without collision with other equipments, meanwhile, the swing angle of the loads should be kept as small as possible (Liu et al., 2005). The structure of the overhead crane system is shown in Figure 4. Obviously, it is made up of two subsystems: the trolley and the load. The symbols in Figure 4 are defined as follows. M is the mass of the trolley which moves on x direction rail; m is the mass of the load which is suspended from the trolley by a rigid rope; x is the distance of the trolley from the rail origin, L is the length of the suspension rope; θ is the swing angle of the load; f is the control force; $x_m = x + L\sin\theta$, $y_m = -L\cos\theta$.

Considering the following standard assumptions:

1 the trolley and the load are regarded as point masses

2 friction force is neglected

3 the rope is rigid

4 the trolley moves in the x direction

5 the load moves on the x–y surface.

By using Lagrange's method, the system dynamics can be obtained as:

$$\begin{bmatrix} M+m \\ \cos\theta \end{bmatrix}\ddot{x} + \begin{bmatrix} mL\cos\theta \\ L \end{bmatrix}\ddot{\theta} + \begin{bmatrix} mL\dot{\theta}^2\sin\theta \\ g\sin\theta \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} \tag{21}$$

Here, g is the gravitational acceleration. Let $x_1 = x$, it is the trolley position with respect to the origin; $x_3 = \theta$, it is the swing angle of the load with respect to the vertical line; x_2 is the velocity of the trolley; x_4 is the angular velocity of the load. From $n = 2$, the non-linear functions f_i and b_i ($i = 1$; 2) in (1) are gotten as:

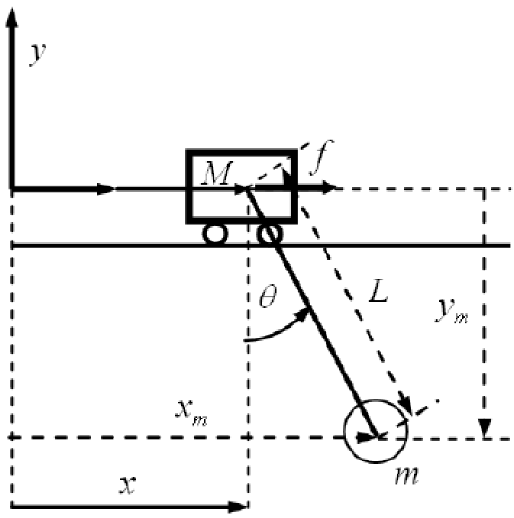
$$f_1 = \frac{mLx_4^2\sin x_3 + mg\sin x_3\cos x_3}{M + m\sin^2 x_3}$$

$$f_2 = -\frac{(m+M)g\sin x_3 + mLx_4^2\sin x_3\cos x_3}{\left(M + m\sin^2 x_3\right)L}$$

$$b_1 = \frac{1}{M + m\sin^2 x_3}$$

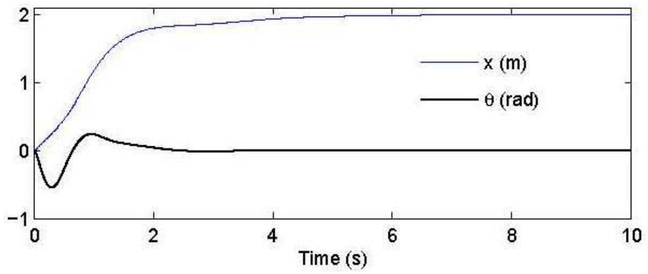
$$b_2 = \frac{\cos x_3}{\left(M + m\sin^2 x_3\right)L}$$

Figure 4 Structure of an overhead crane system

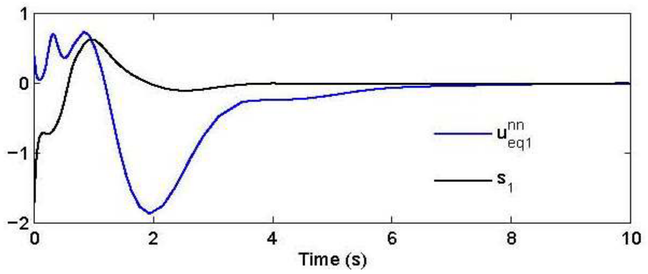


In the simulation, the parameters of the overhead crane system are chosen as trolley mass $M = 1.0$ kg, load mass $m = 0.8$ kg, rope length $L = 0.305$ m and gravitational acceleration $g = 9.81$ m•s^{-2}. The parameters of the sliding surfaces are selected as $c_1 = 0.9$ and $c_2 = 10.2$ after trial and error. The initial weights of the RBF NNs are chosen as the random numbers between –1 and 1; the initial centres and widths of all radial basic functions are chosen as 0 and 2, respectively. The other network parameters are selected as $\lambda_1 = 2$, $\lambda_2 = 15$, $K_1 = 2$, $K_2 = 2$ and $m = 4$. The initial weights of the neuron are chosen as $v_1 = -9.5$ and $v_2 = 0.9$ after trail and error. Learning rate of the neuron is $\mu = 0.1$; the gain K_s is 1.4. The simulation results are shown in Figure 5 and Figure 6 from the initial conditions $x_1 = 0$ m, $x_2 = 0$ m·s^{-1}, $x_3 = 0$ rad and $x_4 = 0$ rad·s^{-1} to the desired states $x_{1d} = 2$ m, $x_{2d} = 0$ m·s^{-1}, $x_{3d} = 0$ rad and $x_{4d} = 0$ rad·s^{-1}. And the sliding mode surfaces of the two subsystems are gotten as $s_1 = c_1(x_1 - x_{1d}) + (x_2 - x_{2d})$ and $s_2 = c_2(x_3 - x_{3d}) + (x_4 - x_{4d})$.

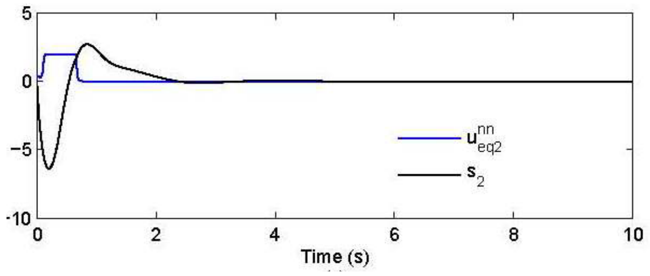
Figure 5 Response curves without a disturbance, (a) is the trolley position and the swing angle of the load (b) is the sliding surface of the first subsystem and the output of the first RBF network (c) is the sliding surface of the second subsystem and the output of the second RBF network (d) is the total sliding surface and the output of the neuron (see online version for colours)



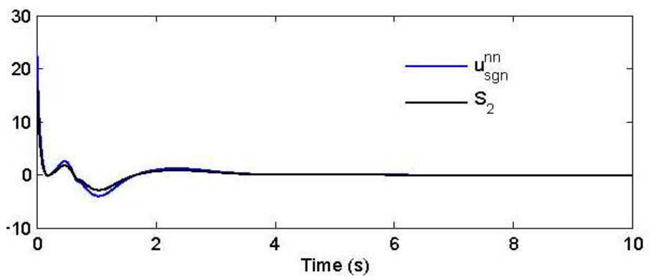
(a)



(b)



(c)

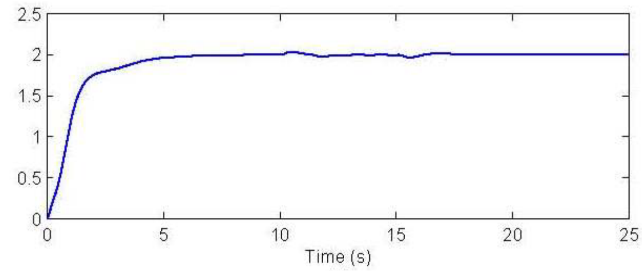


(d)

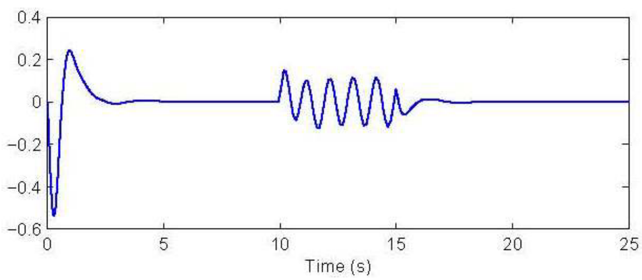
As Figure 5 has shown, the designed NHSMC control law can achieve the control objective. The entire sliding surfaces are asymptotically stable and the network outputs are convergent as has been proven. In Figure 6, a periodic disturbance $y = 0.2 \cdot \sin 2\pi t$ is added between 10 s and 15 s. As Figure 6 has shown, this control method is robust and the controller can resist external disturbances effectively.

Further, we can find that the neuron output is larger than the outputs of the two RBF networks. The reason of this fact is explained as follows. The inputs of the neuron are two subsystem sliding surfaces s_1 and s_2. The parameters of the subsystem sliding surfaces are offered by designers so that the neuron plays an important role in the dynamic process. The two RBF networks contain no information of the system before the dynamic process. They get the system information through the online learning. Thus, a good 'tutor' is necessary to make the two illiterate RBF networks get enough knowledge. In our method, the 'tutor' is the neuron. But this 'tutor' is an online tutor, which is different from the supervised learning. If designers could not offer a group of good parameters of the subsystem sliding surface, the RBF networks would get some bad information. This may lead to an ineffective control process owning to a bad 'tutor' in our control system.

Figure 6 Response curves with a period disturbance, (a) is the trolley position (b) is the swing angle of the load (see online version for colours)



(a)



(b)

In Fang et al. (2003) and Wang et al. (2004), two-dimension and one-dimension overhead cranes were used as controlled plants, respectively. But both of their methods are based on dynamic models, which mean that f_i, b_i (i = 1; 2) are needed to get their control laws. Their control laws may not work well when there exist unknown factors of dynamics models. Under such unknown condition, the NHSMC law could still work and realise the control objective as we have shown by simulations. Tsai et al. (2004) designed a neuro-sliding mode controller for a seesaw system. In his method, only a network was used to calculate the equivalent control. The computational burden may lead a complex network structure as the network inputs increase. For two-order systems (under-actuated systems with two subsystems), our design is more complex than Tsai et al. (2004). But our method could distribute the computational burden for under-actuated systems with more subsystems than two.

5 Experiment results

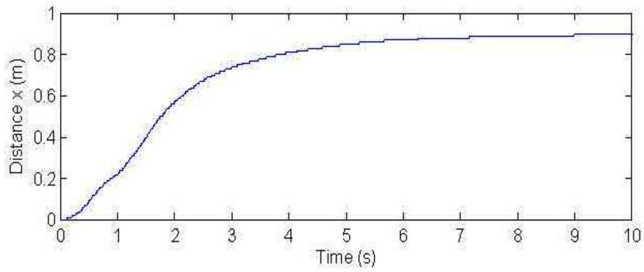
In this section, the control method is implemented on an overhead crane testbed system in Figure 7. Liu et al. (2005) gave the introduction of this testbed in detail. It is a two-dimension crane. Here, we use it as a one-dimension one. The motion control card is PMAC card made by Delta Company. The sample time is 0.025 s offered by the multiple media timer in Microsoft Visual C++. The physical parameters are determined as the trolley mass $M = 37.32$ kg, the load mass $m = 5.00$ kg, the rope length $L = 1.05$ m, the gravitational acceleration $g = 9.81$ m•s^{-2}.

Figure 7 Overhead crane testbed (see online version for colours)

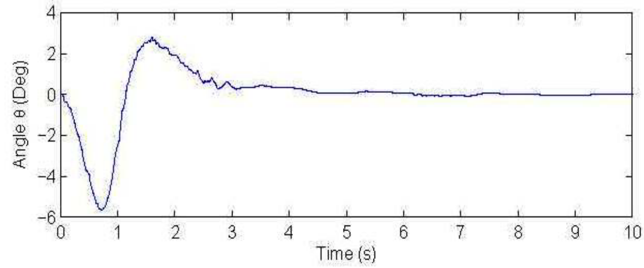
In light of the limitation of the control card in the testbed, it is difficult to calculate the online weights of the neuron and the two RBF NNs. Thus, we train the neuron and the two parallel RBF NNs by simulations. When the weights converge, the network parameters are used for our online control. The experiment results are shown in Figure 8, in which the initial states are $X = [0; 0; 0; 0]^T$ and the desired states are $X = [0.9; 0; 0; 0]^T$. In Figure 9, two random disturbances were added to the load when the crane system arrived at the desired position $[1.2; 0; 0; 0]^T$ from its origin $[0; 0; 0; 0]^T$. The amplitude of the first disturbance was smaller than the second one. The curves with disturbances show the controller is robust and the method could resist external disturbances effectively.

The experiment results show the controller could work well after it gets enough information of the system by simulation learning. As our physical experiment results have shown, the controller could still work although there is no online update of the network weights. Further, we could also deduce that the controller does not work once the new information overs the controller's capability. Under such condition, the NHSMC have to re-learn and update the system information for an effective control process.

Figure 8 Experiment results without a disturbance, (a) shows the trolley position (b) shows the swing angle (see online version for colours)

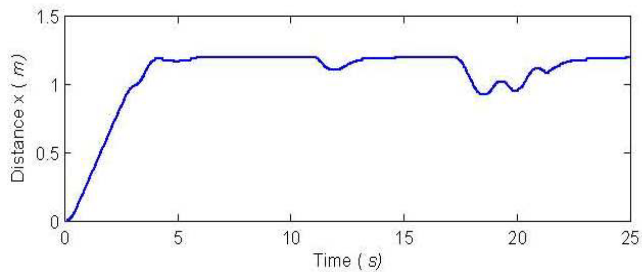


(a)

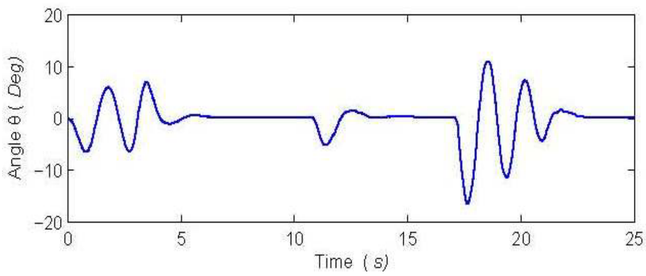


(b)

Figure 9 Experiment results with a random disturbance, (a) shows the trolley position (b) shows the swing angle (see online version for colours)



(a)



(b)

6 Conclusions

The NHSMC has been presented for a class of under-actuated systems with a stable equilibrium in terms of their physical structure characteristic. Based on the HSMC law, the neuron and the parallel RBF NNs have been designed to realise this control law for partly known or uncertain systems. The asymptotic stability of all the sliding surfaces and the convergence of the network weights have been proven theoretically. The simulation and experiment results show the feasibility and the robustness of this control method. But this method could only be applied to the under-actuated systems with a stable equilibrium. This fact represents the weak point of the presented approach: the method described here failed when it is applied to under-actuated systems with an unstable equilibrium or infinite equilibrium. This would constitute an important area for future research.

Acknowledgements

This work was supported by the NSFC Projects under grant Nos. 60904008, 60975060, 60974051 and the Fundamental Research Funds for the Central Universities under grant No. 09MG19.

References

Ertugrul, M. and Kaynak, O. (2000) 'Neuro sliding mode control of robotic manipulators', *Mechatronics*, February–March, Vol. 10, Nos. 1–2, pp.243–267.

Fang, Y., Dixon, W.E., Dawson, D.M. and Zergeroglu, E. (2003) 'Nonlinear coupling control laws for an underactuated overhead crane system', *IEEE/ASME Transactions on Mechatronics, September*, Vol. 8, No. 3, pp.418–423.

Huang, G-B., Saratchandran, P. and Sundararajan, N. (2005) 'A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation', *IEEE Transactions on Neural Networks*, Vol. 16, pp.57–67.

Hussein, I.I. and Bloch, A.M. (2008) 'Optimal control of underactuated nonholonomic mechanical systems', *IEEE Transactions on Automatic Control*, April, Vol. 53, No. 3, pp.668–682.

Jezernik, K., Rodic, M., Safaric, R. and Curk, B. (1997) 'Neural network sliding mode robot control', *Robotics*, January, Vol. 15, No. 1, pp.23–30.

Kaynak, O., Erbatur, K. and Ertugrul, M. (2001) 'The fusion of computationally intelligent methodologies and sliding-mode control – a survey', *IEEE Transactions on Industrial Electronics*, February, Vol. 48, No. 1, pp.4–12.

Kim, H-M. and Oh, S.Y. (1995) 'Locally activated neural networks and stable neural controller-design for nonlinear dynamic-systems', *Int. J. Neural Systems*, March, Vol. 6, No. 1, pp.91–106.

Lin, Ch. and Mon, Y. (2005) 'Decoupling control by hierarchical fuzzy sliding-mode controller', *IEEE Transactions on Control Systems Technology*, July, Vol. 13, No. 4, pp.593–598.

Liu, D., Yi, J., Zhao, D. and Wang, W. (2005) 'Adaptive sliding mode fuzzy control for a two-dimensional overhead crane', *Mechatronics*, June, Vol. 15, No. 5, pp.505–522.

Qian, D., Yi, J. and Zhao, D. (2008) 'Hierarchical sliding mode control for a class of SIMO under-actuated systems', *Control and Cybernetics*, Vol. 37, No. 1, pp.159–175.

Reyhanoglu, M., van der Schaft, A., Mcclamroch, N.H. and Kolmanovsky, I. (1999) 'Dynamics and control of a class of underactuated mechanical systems', *IEEE Transactions on Automatic Control*, September, Vol. 44, No. 9, pp.1663–1671.

Spong, M. (1998) 'Underactuated mechanical systems, control problems in robotics and automation', *Lecture Notes in Control and Information Sciences*, December, Vol. 230, pp.135–150.

Tsai, C.H., Chung, H.Y. and Yu, F.M. (2004) 'Neuro-sliding mode control with its applications to seesaw systems', *IEEE Transactions on Neural Networks*, January, Vol. 15, No. 1, pp.124–134.

Wang, W., Yi, J., Zhao, D. and Liu, D. (2004) 'Design of a stable sliding-mode controller for a class of second-order under-actuated systems', *IEE Proceedings of Control Theory and Applications*, November, Vol. 151, No. 6, pp.683–690.

Xu, R. and Özgüner, Ü. (2008) 'Sliding mode control of a class of underactuated systems', *Automatica*, January, Vol. 44, No. 1, pp.233–241.

Yi, J., Yubazaki, N. and Hirota, K. (2002) 'A new fuzzy controller for stabilization of parallel-type double inverted pendulum system', *Fuzzy Sets and Systems*, February, Vol. 126, No. 1, pp.105–119.

Zhang, M. and Tarn, T-J. (2002) 'Hybrid control of the pendubot', *IEEE/ASME Transactions on Mechatronics*, March, Vol. 7, No. 1, pp.79–86.