Hybrid Neural Sliding Mode Controller Design for Robotic Manipulator

Lon-Chen Hung* and Hung-Yuan Chung**
Department of Electrical Engineering, National Central University,
Chung-Li, Tao-Yuan, 320, Taiwan, R.O.C
TEL: 886-3-4227151 ext 4475
Fax: 886-3-425-5830

Abstract: - In this paper, we will propose a cooperative control approach that is based on the hybrid grey-neural network with sliding mode control (HGNS) methodology. The main purpose is to eliminate the chattering phenomenon such that the two-link robotic manipulator has a superior tracking response. Moreover, the system performance obtained via the method of the HGNS can be improved. In the present approach, two parallel Neural Networks are utilized to realize a Neuro-Sliding Mode Control. The equivalent control and the corrective control in terms of sliding mode control are the outputs of the Neural Networks. The weights adaptations of Neural Network are determined based on the sliding mode control equations. The proposed method is to investigate tracking control of a two-link robotic manipulator. The results demonstrate that not only the system performances are considerably improved, but also the system exhibits desired stability and robustness.

Key-Words: - neural, sliding mode, grey, robotic

1 Introduction

Generally, physical systems have certain non-linear and time-varying behaviours and various uncertainties. It is difficult to establish an appropriate model for the controller design. The Sliding Mode Control (SMC) theory has been developed and has been applied to closed-loop control systems for the last three decades [1-3] which is commonly known as the sliding mode control (SMC), is a nonlinear control strategy that is well known for its robustness characteristics. The essential characteristic of SMC is that the feedback signal is discontinuous, switching on one or more manifolds in state space. When the state crosses each discontinuity surface, the structure of the feedback system is altered. All motion in the neighborhood of the manifold is directed toward the manifold. Note that a sliding motion occurs in which the system state repeatedly crosses the switching surface. When the system states stay in the sliding surface, the equivalent control is capable of making the system stay in the surface.

The main feature of SMC is that it uses a high-speed switching control law to drive the system states from any initial state onto a user-specified surface in the state space, namely, switching surface, and to keep the states on the surface for all successive time. As a result, we have the system dynamics entirely determined by the parameters that describe the sliding surface. This also results in a system that is insensitive to parametric uncertainties and external disturbances. In the design of the sliding mode control law, it is assumed that the control can be switched from one value to another infinitely fast [4-6]. However, this is impossible to achieve in practical systems because finite time delays are present for control computation, and limitations exist in the physical actuators. This nonperfect switching results in a phenomenon called chattering. The high frequency component of chattering is not only undesirable by itself but also they can excite unmodeled high-frequency plant dynamics which could result in unforeseen instability. The chattering behavior is especially unacceptable in process control and thus it has received considerable notice from the research community [7-11].

Improved generalization performance for error-based neural network learning can be obtained with techniques such as validation, pruning or constructive algorithms [12-13]. The methods are based on the principle that generalization is associated with the number of network parameters (weights), This study presents a HGNS system for the tracking control of a two-link robotic manipulator to achieve high-precision position control.

In the present approach, two parallel Neural Networks are utilized to realize a Neuro-Sliding Mode Control. The equivalent control and the corrective control in terms of Sliding Mode Control are the outputs of the Neural Networks. The weights adaptations of Neural Network are determined based on the Sliding Mode Control equations. Then, the gradient descent method is used to minimize the control force so that chattering phenomenon can be

eliminated. The action of such HGNS are equivalent to that of full-state feedback controllers for second-order systems and, hence, these systems can always be stabilized. This controller guarantees some properties, such as the robust performance and stability properties.

We show that the HGNS has the following advantages: (1) It can well control most of complex systems without knowing their exact mathematical models. (2) The dynamic behavior of the controlled system can be approximately dominated by a hybrid neural sliding surface. (3) HGNS can not only increase the robustness to system uncertainties but also decrease the chattering phenomenon in the conventional sliding mode controller.

The rest of the paper is divided into five sections. In Section 2, the systems are described. In Section 3, the hybrid neural network sliding-mode control is presented. In Section 4, the proposed controller is used to control a Robotic system. Finally, we conclude with Section 5.

2 System description

Consider a nonlinear, non-autonomous, multi-input multi-output system of the form

$$x_i^{(ki)} = f_i(\boldsymbol{X}) + \sum_{j=1}^{m} b_{ij} u_j \tag{1}$$

where $x_i^{(ki)}$ indicates the k_i^{th} derivative of x_i. The vector U of components u_j is the control input vector and the state X is composed of the x_{is} and their first (k_i–1) derivatives. Such systems are called square systems since they have as many control inputs as outputs x_i to be controlled. The system can be written in a more compact form as letting

$$\boldsymbol{x} = [x_1 ... x_m \quad \dot{x}_1 ... \dot{x}_m \quad x_1^{(k_1-1)} ... x_m^{(k_m-1)}]^T \tag{2}$$

$$\boldsymbol{u} = [u_1 ... u_m]^T \tag{3}$$

Assuming that $\boldsymbol{x} \in \boldsymbol{R}^{n\times 1}$, the system equation becomes

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}) + B\boldsymbol{u}(t) \tag{4}$$

where $\boldsymbol{B} \in \boldsymbol{R}^{n\times m}$ input gain matrix.

2.1 Sliding mode controller

For the system given in (4), the sliding surface variable that is represented by $S \in \boldsymbol{R}^{m\times 1}$. $S = 0$ defines a sliding surface. It is selected as

$$S(X,t) = C(X_d(t) - X(t)) = S_d(t) - S_a(X) \tag{5}$$

where

$$S_d(t) = CX_d(t), \quad S_a(X) = CX(t) \tag{6}$$

i.e., the time and the state dependent parts. The vector X_d represents the desired (reference) state and $C \in \boldsymbol{R}^{m\times n}$ is the slope matrix of the sliding surface. Generally, C is selected such that the sliding surface function becomes

$$S_i = (\frac{d}{dt} + \lambda_i)^{k_i - 1} e_i \tag{7}$$

where e_i is the error for x_i ($e_i = x_{d_i} - x_i$) and λ_is are selected as positive constants. Therefore, e_i goes to zero when S_i equals zero. The aim in SMC is to force the system states to the sliding surface. Once the states are on the sliding surface, the system errors converge to zero with error dynamics dictated by the matrix C.

To follow above, the design of an SMC based on the selection of a Lyapunov function is presented. The control should be chosen such that the candidate Lyapunov function satisifies Lyapunov stability criteria. Let the Lyapunov function be selected as below [14]:

$$V(S) = \frac{S^T S}{2} \tag{8}$$

$$\frac{dV(S)}{dt} = -S^T G sign(S) \tag{9}$$

where $G \in \boldsymbol{R}^{m\times m}$ positive definite diagonal gain matrix, and $sign(S)$ means signum function is applied to each element of S, i.e.

$$sign(S) = [sign(S_1) \;\; sign(S_2) ..... sign(S_m)]^T \tag{10}$$

and $sign(S_i)$ is defined as

$$sign(S_i) = \begin{cases} +1 & \text{if } S_i > 0 \\ -1 & \text{otherwise} \end{cases} \tag{11}$$

Taking the derivative of (8) and equating this to (9), the following equation is obtained:

$$S^T \frac{dS}{dt} = -S^T G sign(S) \tag{12}$$

The time derivative of S can be obtained using (5) and the plant equation as given below:

$$\frac{dS}{dt} = \frac{dS_d}{dt} - \frac{\partial S_a}{\partial X}\frac{dX}{dt} = \frac{dS_d}{dt} - C(f(\boldsymbol{x}) - B\boldsymbol{u}) \tag{13}$$

By putting (13) into (12), the control input signal can be written as

$$U(t) = U_{eq}(t) + U_c(t) \tag{14}$$

where $U_{eq}(t)$ is the equivalent control given by

$$U_{eq}(t) = -(CB)^{-1}(Cf(\boldsymbol{x}) - \frac{dS_d}{dt}) \tag{15}$$

and $U_c(t)$ is the corrective control given by

$$U_c(t) = (CB)^{-1} G sign(S) = K sign(S) \tag{16}$$

$U_{eq}(t)$ is used to control the overall behavior of the system and $U_c(t)$ is used to reject disturbances and to suppress parameter uncertainties. The controller of (14) exhibits high frequency oscillations in its output, causing a problem known as the chattering phenomena. Chattering is highly undesirable because

it can excite the high frequency dynamics of the system. For its elimination, it is suggested to use a modified shifted sigmoid function instead of the sign function. In the latter case, the corrective control is computed as

$$U_c(t) = K\phi(S) \tag{17}$$

where $\phi(\cdot)$ is a modified sigmoid function, defined as

$$\phi(S_j) = \frac{2}{1+e^{-S_j}} - 1 \tag{18}$$

3 Design Hybrid Neural Sliding Mode Controller

3.1 The Structure of Neuro-Sliding Mode Controller

Improved generalization performance for error-based neural network learning can be obtained with techniques such as validation, pruning or constructive algorithms. The last two approaches are based on the principle that generalization is associated with the number of network parameters, whereas validation is based on the validation set error to select the model with best generalization without explicit reference to network complexity.

In this study, we use neural network to generate U_{eq} and U_c of SMC. Hence, U_{eq} is generated by NN1 and U_c is generated by NN2. The structure of SNNS for the Robotic system is presented in Fig. 1. We have introduced the structure of SNNS, and then we need to discuss how to adjust weights of NN1 and NN2 such that the system state trajectory reaches the sliding surface as soon as possible. In this work, we update the weights of the NN1 and NN2 to minimize the *S* of SMC.

3.2 Computation of the Equivalent Control

The structure of NN1 is chosen by a two layers feed-forward neural network, which has one hidden layer and one output layer. The structure of inputs and output of the network are established by the equivalent control equation. From Fig. 2, it is found that the equivalent control is computed by using desired and actual states.

In Fig. 2, some symbols are defined as follows descriptions. The input and the output of the hidden layer are designated as $Ynet_j$ and $Yout_j$, respectively. The sub index j means that the j-th hidden-layer neuron. Similarly, the inputs and output of the output layer are designated as U_{net} and U_{out}, respectively. The values can be computed as

$$Yout_j = \varphi\left(Ynet_j\right), \quad i = 1,2,...,n \tag{19}$$

$$Unet = \sum_{j=1}^{m} Wy_j Yout_j, \; Uout = g\left(Unet\right), \; j = 1,2,...,m \tag{20}$$

$$Wz_{i,j} = Wz_{i,j} + \Delta Wz_{i,j}, \; Wy_j = Wy_j + \Delta Wy_j \tag{21}$$

$$g(x) = \frac{2}{1+e^{(-net)}} - 1 \tag{22}$$

where $Wz_{i,j}$ means that the weights between the neurons of input layer and hidden layer. Concretely, the $Wz_{i,j}$ can be regarded as the weight of i-th input layer neuron to the j-th hidden-layer neuron. The activation function $g(\cdot)$ is selected as a sigmoid transfer function, defined in (22). The symbol n represents the total number of the input layer, and m represents the total number of the hidden neurons. U_{eq} is the estimated value of the equivalent control in Fig. 2. In order to avoid the equivalent control exceeds the maximum bound of actuator, reaching to a unreasonably large value, the output of neural network is [-1 , 1].

3.3 Weight adaptation of NN1 for the equivalent control estimation

The weight adaptation is based on a minimization of a cost function that is selected as the difference between the desired and the estimated equivalent control

$$E = \frac{1}{2}\sum_{j=1}^{m=2}\left(\frac{U_{eq_j} - \hat{U}_{eq_j}}{U_{\max_j}}\right)^2 \tag{23}$$

The gradient descent method is used to update the weight of NN1. The update formula are shown as follows:

$$\Delta W_{yj,i} = -\eta\frac{\partial E}{\partial W_{yj,i}} = -\eta\frac{\partial E}{\partial U_{net_j}}\frac{U_{net_j}}{\partial W_{yj,i}} = \eta\delta_{yj}Y_{out_j} \tag{24}$$

where

$$\delta_{yj} = -\frac{\partial E}{\partial U_{net_j}} = -\frac{\partial E}{\partial \hat{U}_{eq_j}}\frac{\partial \hat{U}_{eq_j}}{\partial U_{net_j}} = \frac{(U_{eq_j} - \hat{U}_{eq_j})}{U_{\max_j}}\frac{\partial g(U_{net_j})}{\partial U_{net_j}} \tag{25}$$

$$\delta_{yj} = \frac{U_{c_j}}{U_{\max_j}}\frac{\partial g(U_{net_j})}{\partial U_{net_j}} \tag{26}$$

The derivative of the modify shifted sigmoid function is computed as

$$\frac{\partial g(U_{net_j})}{\partial U_{net_j}} = \frac{1}{2}(1 - g(U_{net_j})^2) \tag{27}$$

Gradient descent for the hidden layer is computed as

$$\Delta W_{z_{j,i}} = -\eta\frac{\partial E}{\partial W_{z_{j,i}}} = -\eta\frac{\partial E}{\partial Y_{net_j}}\frac{\partial Y_{net_j}}{\partial W_{z_{j,i}}} = \eta\delta_{z_j}z_i \tag{28}$$

where

$$\delta_{z_j} = -\frac{\partial E}{\partial Y_{net_j}} = (\sum_{k=1}^{m}\delta_{y_k}W_{y_{k,j}})(1-(Y_{out_j})^2) \tag{29}$$

The most important point in this derivation is that the error between the desired and the estimated

equivalent control is replaced with the corrective control of the sliding mode control, as it is seen from (26).

3.4 Computation of the Corrective Control

In this GNNM(1,1) [13] of NN2 structure obtained for the corrective control U_c. Thus the structure of GNNM(1,1) of NN2 is easy to determine via designing SMC. The structure of GNNM(1,1) of NN2 is also a feed-forward network, which has one input layer and one output layer. The structure of NN2 for the manipulator is presented in Fig. 3.

From Fig 3, it can be found that the input neuron are state error. The output neuron is a full connection structure. The advantage is that the controller possesses the self-turning characteristic. As a general neural network, the output of the output neuron also passes through a sign function or sign-like continuous functions. Hence, we use activation transfer function as the expression (22). The output of the output layer is the corrective control.

In order to eliminate the high-frequency control and chattering around the sliding surface caused by the SMC, we now introduce a grey-neural network to be the direct adaptive neural controller to replace the SMC, when the state trajectory of system goes into the boundary layer.

Grey system theory reckons that system of original sequence $x^{(0)}(k)$ (k=0,1,2,…,N-1)is added to a new series $x^{(1)}(k)$ that is characterized by exponential increase. So it is possible to use continuous function or differential equation to fit discrete data and the more is the number, the grey indefinite problems can be improved to solve in methods. On the point of theory view, discrete data from above method seem to be some discrete points extracted from certain function or differential equation but not independent mutually and disorder data.

Some symbols can be changed as follows: original series $x_k^{(0)}(t)$ are represented by $x_k(t)$, the result series $x_k^{(1)}(t)$ after once adding are denoted by $y_k(t)$, the predictive result value $\hat{x}_k^{(1)}(t)$ is showed by $z_k(t)$, the range of t is $t\in(0,N\text{-}1)$.

The grey differential equation GM(1,1) is denoted as follows:

$$\frac{dx^{(1)}(k)}{dt}+ax^{(1)}(k)=u \tag{30}$$

In this paper, they are represented as:

$$\frac{dy(k)}{dt}+ay(k)=u \tag{31}$$

then we get (32)

$$y(t)=(y(0)-\frac{u}{a})e^{-at}+\frac{u}{a} \tag{32}$$

Use the transform method of discrete response model, the time response model can be processed as follows $(e^{-at}>0)$:

$$z(t)=((y(0)-\frac{u}{a})-y(0)\times\frac{1}{1+e^{-at}}+2\times\frac{u}{a}\times\frac{1}{1+e^{-at}})(1+e^{-at}) \tag{33}$$

The grey-neural network such as figure 3, corresponding weights of network are evaluated as follows $(p=\frac{2u}{a})$: $W_{11}=a$, $W_{21}=-y(0)$, $W_{21}=p$, $W_{31}=W_{32}=1+e^{-at}$.

The threshold y_1 is :

$$\theta_{y1}=(1+e^{-at})(\frac{u}{a}-y(0)) \tag{34}$$

According to this grey-neural network in the training, weights are modified continuously is just as whitening of grey parameters extracted.

The neural-grey predictor conform to is represented as the following:

The sliding surface with a boundary width Ω designed in the phase plane. The S; $\hat{S}$; x_1 and x_2 are the sliding function, prediction sliding function, motor angle and motor velocity, respectively. The grey-neural predictor forecasts the movement of the state in the following conditions:

(a) The positive step prediction: $|x_1S|>\Omega$ or $|x_2S|>\Omega$ and one of the following four conditions is satisfied

1. $\hat{S}(k+1)-S(k)>0$ if $S(k)>0$ and $\hat{S}(k+1)>0$,
2. $\hat{S}(k+1)-S(k)<0$ if $S(k)<0$ and $\hat{S}(k+1)<0$,
3. $S(k)>0$ and $\hat{S}(k+1)<0$,
4. $S(k)<0$ and $\hat{S}(k+1)>0$.

(b) The negative step prediction: $|x_1S|>\Omega$ or $|x_2S|>\Omega$ and one of the following two conditions is satisfied

1. $\hat{S}(k+1)-S(k)>0$ if $S(k)>0$ and $\hat{S}(k+1)>0$,
2. $\hat{S}(k+1)-S(k)<0$ if $S(k)<0$ and $\hat{S}(k+1)<0$,

We add the positive step prediction control to prompt the state into the boundary layer when the estimation state far away. Similarly, we add the negative step prediction control when the state outside the boundary layer and the estimation state go toward the boundary.

To satisfy the equivalent control concept are substituted into ; we get

$$U_c=-\gamma\hat{S}, \tag{35}$$

where

$$\gamma=\begin{cases}\sigma>0 & \text{for positive or negative step prediction}\\ 0 & \text{otherwise,}\end{cases} \quad (36)$$

where γ represents the gain of the positive or negative step prediction and $\hat{S}$ represents the forecast value of the sliding function S by the GNNM(1,1) predictor. The grey-neural network sliding controller satisfies the hitting condition at the sliding line and no reaching time when t = 0, and the error of e in the boundary layer W when $t \to \infty$. Moreover, the boundary concept should be considered when designing γ . The overall algorithm described above is summarized in the Application Algorithm.

Remark 1: Based on the Lyapunov theorem, the sliding surface reaching condition is $s\dot{s}$ <0. If a control input *u* can be chosen to satisfy this reaching condition, the control system will converge to the origin of the phase plane. Since a HGNS is employed to approximate the non-linear mapping between the sliding input variable and the control law, the weightings of the HGNS should be regulated based on the reaching condition, $s\dot{s} < 0$.

Remark 2: We combine the neural-grey predictor and sliding mode technique to reduce the sensitivity for the disturbances and the parameter variations. We need not know the uncertain bounded disturbances and parameter variations. The sliding function is estimated to reject the disturbances and the parameter variations.

Remark 3: The adaptive rule is derived from the steep descent rule to minimize the value of $s\dot{s}$ with respect to W_j of HGNS. The weightings between hidden and output layers neurons can be on-line adjusted to achieve the learning ability of NN1, and updating the grey-neural controller parameter γ of NN2. Theoretically, the HGNS can be used to model and approximate any non-linear function with a reasonable accuracy. From the above analysis, it can be concluded that the proposed HGNS controller is stable and the system output error at least converges into a small error bound.

4. Computer Simulations

In this section, a two rigid-link robot manipulator shown in Fig. 4 is utilized in this study to verify the effectiveness of the proposed control scheme. The dynamic model of the adopted robot system can be described [16] in the Eq. (37)

$$H(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q)=\tau \quad (37)$$

i.e.

$$\begin{bmatrix}H_{11} & H_{12}\\ H_{21} & H_{22}\end{bmatrix}\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix}+\begin{bmatrix}-h\dot{q}_2 & -h(\dot{q}_1+\dot{q}_2)\\ h\dot{q}_1 & 0\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix}=\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} \quad (38)$$

with $\boldsymbol{q}=[q_1\ q_2]^T$ being the two joint angles, $\boldsymbol{\tau}=[\tau_1\ \tau_2]^T$ being the joint inputs, and $\tilde{q}=q_d-q$ being the tracking error.

$$H_{11}=p_1+2a_3\cos q_2+2a_4\sin q_2$$
$$H_{22}=p_2$$
$$H_{12}=H_{21}=p_2+p_2\cos q_2+a_4\sin q_2$$
$$h=p_3\sin q_2-p_4\cos q_2$$
$$p_1=I_1+m_1l_{c1}^2+I_e+m_el_{ce}^2+m_el_1^2$$
$$p_2=I_e+mel_{ce}^2$$
$$p_3=m_el_1l_{ce}^2\cos\delta_e$$
$$p_4=m_el_1l_{ce}^2\sin\delta_e$$

Equation (38) can be rewritten as

$$\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix}=-\begin{bmatrix}H_{11} & H_{21}\\ H_2 & H_{22}\end{bmatrix}^{-1}\begin{bmatrix}-h\dot{q}_2 & -h(\dot{q}_1+\dot{q}_2)\\ -h\dot{q}_1 & 0\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix}+\begin{bmatrix}H_{11} & H_{21}\\ H_2 & H_{22}\end{bmatrix}^{-1}\begin{bmatrix}\tau_1\\ \tau_2\end{bmatrix} \quad (39)$$

where $\begin{bmatrix}H_{11} & H_{21}\\ H_2 & H_{22}\end{bmatrix}^{-1}$ is assumed to be existent.

Equation (39) can be expressed as a state equation

$$\ddot{\boldsymbol{q}}=\boldsymbol{f}_q(\boldsymbol{q})+\boldsymbol{G}_q(\boldsymbol{q})\boldsymbol{u} \quad (40)$$

where $\boldsymbol{q}=[q_1\ q_2\ \dot{q}_1\ \dot{q}_2]^T$, $\boldsymbol{u}=[\tau_1\ \tau_2]^T$

$$\boldsymbol{f}_q(\boldsymbol{q})=-\begin{bmatrix}H_{11} & H_{21}\\ H_2 & H_{22}\end{bmatrix}^{-1}\begin{bmatrix}-h\dot{q}_2 & -h(\dot{q}_1+\dot{q}_2)\\ -h\dot{q}_1 & 0\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix},\ \boldsymbol{G}_q(\boldsymbol{q})=-\begin{bmatrix}H_{11} & H_{21}\\ H_2 & H_{22}\end{bmatrix}^{-1}.$$

The joint angle error vector is defined as $\tilde{\boldsymbol{q}}=\boldsymbol{q}_d-\boldsymbol{q}$, and the error state vector is defined as $\tilde{\boldsymbol{q}}=[q_1\ q_2\ \dot{q}_1\ \dot{q}_2]^T$ where $\boldsymbol{q}_d$ is the desired joint angle vector.

The system parameters of the two rigid-link robot manipulator are selected as: m_1=1kg, m_e=2kg, $\delta_e=30^0$, l_1=0.12m, l_{c1}=0.5m, l_{ce}=0.6m, and initial value are $q_1=0^0$, $q_2=0^0$.

The performance of position and tracking control from the simulations are presented and compared with the simulation results of Slotine and Li [16].

(a) position control : The robot, initially at rest at ($q_1=0^o$, $q_2=0^o$), is commanded to move one step to ($q_1=50^o$, $q_2=80^o$). The performance of the proposed HGNS is compared with the SMC control [16]. For the case of 50% mass uncertainties for m_1 and m_e, the

simulation results are shown in Fig. 6 through Fig. 9. The results show that the proposed HGNS control still can cope with the mass uncertainties and model error to achieve performance.

(b) robust tracking control : The robot, initially at rest at (q_1=0^o, q_2=0^o), is commanded to follow a desired trajectory q_{d1}=50+20cos(πt) and q_{d2}=20+40sin(πt). The performance of the HGNS is compared with the SMC control approaches [16]. For the case of 50% mass uncertainties for m_1 and m_e, the simulation results are shown in Fig. 10 through Fig. 11, which shows that the HGNS approach has smaller tracking errors, and the transient tracking performance is also better than those using the sliding control. It is found that the control purpose of the robotic system can be arrived at, and can be shown that q and $\dot{q}$ converge to zero, respectively. This simulation also shows that the proposed HGNS can achieve the best control performance with favorable tracking performance.

5 Conclusion

In this paper, the HGNS control system has been proposed to solve the output tracking problem for highly nonlinear and coupling, and the complete dynamic model was difficult to obtain precisely. To verify the effectiveness of the proposed control scheme, the HGNS control system was implemented to control a two-link manipulator. The simulation results show that the purpose of joint-tracking will be arrived at, and the link of robotic can be stabilized to the equilibrium and robustness.

References:

[1] R.A. Decarlo, Variable structure system with sliding modes, *IEEE Trans*, *Automat. Control* AC-22, 1977, pp. 212-222.

[2] J.A. Burton, A.S.I. Zinober, "Continuous self-adaptive control using a smoothed variable structure controller, *Int. J. Systems Sci*., Vol. 19, 1988, pp. 1515-1528.

[3] W. J. Wang, G. H. Wu, and D. C. Yang, Variable structure control design for uncertain discrete-time systems, *IEEE Trans. Automat. Contr*., 1994, Vol. 39, pp. 99-102.

[4] Kueon Y.S., Bedi, J.S., Fuzzy-neural-sliding mode controller and its applications to the vehicle anti-lock braking systems, *International IEEE/IAS Conference on*, 22-27, 1995, pp. 391-398,.

[5] Hongliu Du, Nair S.S., A neuro-sliding control approach for a class of nonlinear systems, *Knowledge-Based Intelligent Electronic Systems*, *KES '97. Proceedings., International Conference*, Vol. 2, 1997, pp. 331-337.

[6] Ertugrul M., Kaynak O., Neuro-sliding mode control of robotic manipulators, *Advanced Robotics, ICAR '97. Proceedings International Conference on* , 1997, pp. 951-956.

[7] Barambones Oscar, Etxebarria Victor, Robust neural control for robotic manipulators, *Automatica*, Vol. 38, 2002, pp. 235-242.

[8] Huang Shiuh-Jer, Huang Kuo-See, Chiou Kuo-Ching, Development and application of a novel radial basis function sliding mode controller, *Mechatronics*, Vol. 13, 2003, pp. 313-329.

[9] Hussain, Mohd Azlan, Ho, Pei Yee, Adaptive sliding mode control with neural network based hybrid models, *Journal of Process Control*, Vol. 14, 2004, pp. 157-176.

[10] Costa, Marcelo A., Braga, Antonio P., Menezes, Benjamin R., Teixeira, Roselito A., Training neural networks with a multi-objective sliding mode control algorithm, *Neurocomputing*,Vol. 51, 2003 pp. 467-473.

[11] Wai, Rong-Jong, Tracking control based on neural network strategy for robot manipulator, *Neurocomputing*,Vol. 51, 2003, pp. 425-445.

[12] Karakasoglu, Ahmet; Sundareshan, Malur K., A Recurrent Neural Network-based Adaptive Variable Structure Model-following Control of Robotic Manipulators, *Automatica*,Vol. 31, 1995 pp. 1495-1507.

[13] G.G. Parma, B.R. de Menezes, A.P. Braga, Sliding mode algorithm for training multilayer artificial neural networks, *Electron. Lett*., 1998, pp. 97-98.

[14] Ertugrul Meliksah, Kaynak Okyay, Neuro sliding mode control of robotic manipulators, *Mechatronics*, Vol. 10, 2000, pp. 239-263.

[15] L. Zhong, J. L. Yuan, H. X. Xia, C. M. Zou, A study on gray neural network modelling, *Machine Learning and Cybernetics*, *Proceedings. 2002 International Conference*, Vol. 4, 2002, pp. 4-5.

[16] J. J. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice-Hall, 1991.

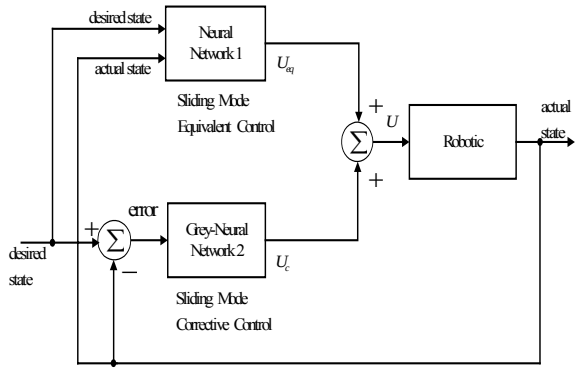

Fig. 1 The Structure of HGNS Controller.

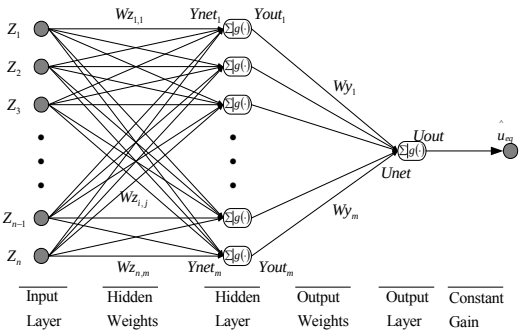

Fig. 2. The structure of NN1 of SNNS to estimate the equivalent control

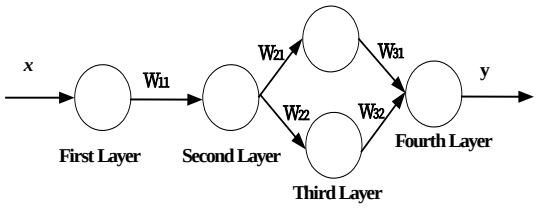

Fig. 3. The structure GNNM(1,1) of NN2 of HGNS to compute the corrective control.

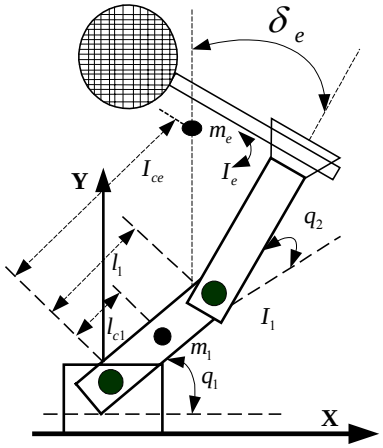

Fig. 5. An two-link robot manipulator.

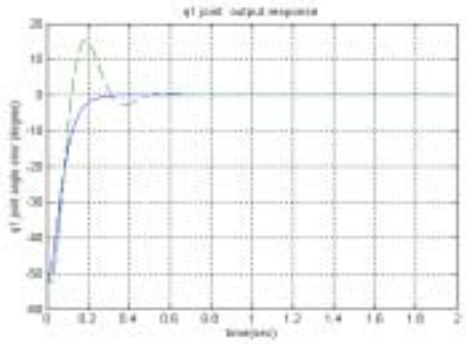

Fig. 6. The joint 1 position response comparison with SMN and HGNS.

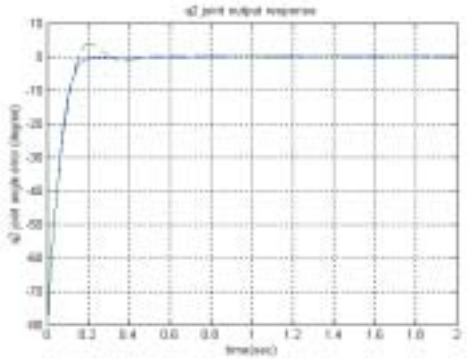

Fig. 7. The joint 2 position response comparison with SMN and HGNS.

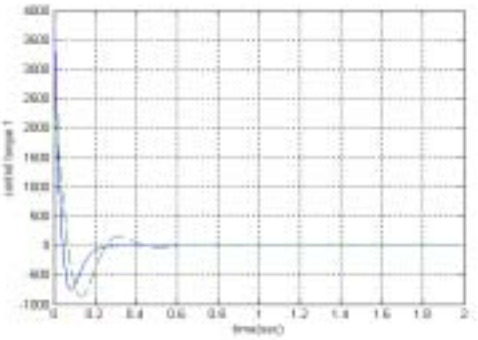

Fig. 8. The joint 1 control force comparison with SMN and HGNS.

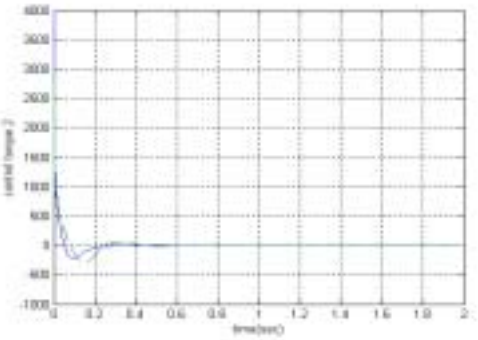

Fig. 9. The joint 2 control force comparison with SMN and HGNS.

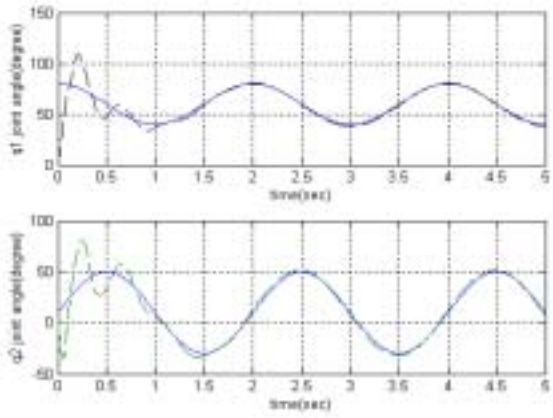

Fig. 10. (a) The joint 1 position response with SMC.
(b) The joint 2 position response with SMC.

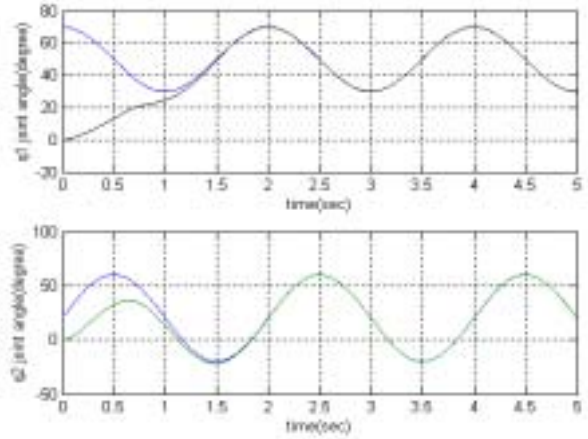

Fig. 11. (a) The joint 1 position response with HGNS.
(b) The joint 2 position response with HGNS.