Synthesize Neural Sliding Mode Controller for Nonlinear System

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 Synthesize of self-tuning neural network with sliding mode control methodology. The main purpose is to eliminate the chattering phenomenon such that the seesaw system has a superior fixed position and tracking response. Moreover, the system performance obtained via the method of synthesize neural network with SMC (SNNS) 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 simulation and experiment seesaw system result show which the proposed controller shown that the proposed method is feasible and effective.

Key-Words: - Neural Network, Sliding mode control, Seesaw

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 [12-14]. The methods are based on the principle that generalization is associated with the number of network parameters (weights), This study presents a SNNS system for the tracking control of a seesaw system to achieve high-precision position control. One of the methods to cope with the problem is to utilize a feedforward compensator to offset unpredictable affect of system uncertainties.

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 SNNS 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 SNNS 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 Synthesize neural sliding surface. (3) SNNS 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 Synthesize neural network sliding-mode control is presented. In Section 4, the proposed controller is used to control the seesaw system. Finally, we conclude with Section 5.

2 Review of sliding mode control design

In general, the Sliding Mode Control (SMC) can be separated into the reaching mode and sliding mode. The sliding surface is the desirable path in state space, which is given by the designer. These points constitute a special trajectory along the surface, called a sliding mode. Thus, a phase trajectory of this system generally consists of two parts, representing two modes of the system. The first part is the reaching mode in which the trajectory starts from anywhere on the phase plane moving toward a switching surface and reaches the surface in finite time. The second part is the sliding mode in which the trajectory asymptotically tends to the origin of the phase plane.

Consider the n-th order linear system, which can be represented by the following state-space model in a companion form.

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\vdots \\ \dot{x}_{n-1} &= x_n \\ \dot{x}_n &= a_1 x_1 + a_2 x_2 + \cdots + a_n x_n + u(t) + d(t) \end{aligned} \tag{1}$$

Where and is the n-dimensional state vector and control input of the system, respectively, and represents the external disturbance.

If the disturbance is known and has the upper bound .

$$|d(t)| < D \tag{2}$$

And if the desired state is a step function, then the above dynamic equations can be transformed into the error equation.

The error is defined and its derivatives are the state variables

$$\begin{aligned} \dot{e}_1 &= e_2 \\ \dot{e}_2 &= e_3 \\ &\vdots \\ \dot{e}_{n-1} &= e_n \\ \dot{e}_n &= a_1 e_1 + a_2 e_2 + \cdots + a_n e_n + u(t) + d(t) \end{aligned} \tag{3}$$

In general, the SMC design has a two-step process. First, choose a switching surface, which is chosen by desired behavior. Restricted to the intersection of the switching surface (Sliding mode), it results in the desired behavior. Next, define a switching control such that the trajectory of the system converges to the sliding surface (Reaching mode), and then stay on the sliding surface.

Most, sliding surfaces are defined of follows:

$$S(e) = \mathbf{c}^T \mathbf{e} \tag{4}$$

where $\boldsymbol{c}=[c_1, c_2, ..., c_n]^T$, $\boldsymbol{c} \in \boldsymbol{R}^n$, $\boldsymbol{e}=[e_1, e_2, ..., e_n]^T$. The vector is given for $S(e) = \mathbf{c}^T \mathbf{e} = 0$. It means that the resultant system is stable. Hence, the control can input drive the system (1) converges to the stable sliding surface.

Expanding (4), we have

$$S(e) = c_1 e_1 + c_2 e_2 + \cdots + c_{n-1} e_{n-1} + e_n = \boldsymbol{c}^T \boldsymbol{e} = 0 \tag{5}$$

where $c_n = 1$. Taking the time derivative for $S(e)$, we can obtain the following equation,

$$\dot{S}(e) = c_1 \dot{e}_1 + c_2 \dot{e}_2 + \cdots + c_{n-1} \dot{e}_{n-1} + \dot{e}_n = \boldsymbol{c}^T \dot{\boldsymbol{e}} = 0 \tag{6}$$

Substituting (1), (3) into (6), we have

$$\dot{S}(e) = a_1 e_1 + (a_2 + c_1) e_2 + \cdots + (a_n + c_{n-1}) e_n + u_{eq} + d(t) = 0 \tag{7}$$

Obviously, the equivalent control u_{eq} of (7) are define as

$$u_{eq} = -a_1 e_1 - (a_2 + c_1) e_2 - \cdots - (a_n + c_{n-1}) e_n - d(t) \tag{8}$$

The equivalent control is to keep the system states at the sliding surface S=0 for all t ≥ 0. Hence, if the states is outside the sliding surface, to drive the state to the sliding surface, we choose the control law such that

$$\boldsymbol{S} \dot{\boldsymbol{S}} < -\sigma |\boldsymbol{S}| \tag{9}$$

where σ is a positive constant, and (9) is called reaching condition. We have to guarantee the state trajectory converge to the sliding surface. Next, we define the corrective control, u_c, which are shown as follows.

$$u_c = K \cdot \mathrm{sgn}(S) \tag{10}$$

Where K is constant. The sign function is a discontinuous function.

$$\mathrm{sgn}(S) = \begin{cases} 1 \ , \ S>0 \\ 0 \ , \ S=0 \\ -1, \ S<0 \end{cases} \tag{11}$$

Hence, the whole control input is a combination of and

$$u = u_{eq} + u_c = -a_1 e_1 - (a_2 + c_1)e_2 - \cdots - (a_n + c_{n-1})e_n \\ - d(t) + K \cdot \mathrm{sgn}(S) \tag{12}$$

Notice that the (10) exhibits high frequency oscillations, which is defined as chattering. Chattering is undesired because it may excite the high frequency response of the system. Hence, most of approaches use the saturation or the sigmoid function to replace the sign function.

3 Design Synthesize 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 [15].

The values can be computed as

$$Yout_j = \varphi\left(Ynet_j\right), \quad i = 1, 2, ..., n \tag{13}$$

$$Unet = \sum_{j=1}^{m} Wy_j Yout_j, \ \ Uout = g\left(Unet\right), \ j = 1, 2, ..., m \tag{14}$$

$$Wz_{i,j} = Wz_{i,j} + \Delta Wz_{i,j}, \ Wy_j = Wy_j + \Delta Wy_j \tag{15}$$

$$g(x) \ = \frac{2}{1 + e^{(-net)}} - 1 \tag{16}$$

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 (16). 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

$$E = \frac{1}{2}\sum_{j=1}^{m=2}\left(\frac{U_{eq_j} - \tilde{U}_{eq_j}}{U_{\max_j}}\right)^2 \tag{17}$$

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

$$\Delta Wy_j = -\lambda \cdot \frac{\partial E}{\partial Wy_j} = \lambda \cdot \left(u_{eq} - \hat{u}_{eq}\right) \cdot \frac{1}{2}\left[1 - g\left(Unet\right)^2\right] \cdot Yout_j \tag{18}$$

where λ is the learning-rate parameter of the back-propagation algorithm and it is a constant. From (18), we find that the actual equivalent control u_{eq} is unknown. Hence, ΔWy_j of (18) cannot be calculated. In order to overcome this problem, we use   to replace the $u_{eq} - \tilde{u}_{eq}$. The reason is that is given by the designer, and the characteristics of $u_{eq} - \tilde{u}_{eq}$ and S are similar. Thus, (18) can be rewritten as follows:

$$\Delta Wy_j = \lambda \cdot (S) \cdot \frac{1}{2}\left[1 - g(Unet)^2\right] \cdot Yout_j \quad (19)$$

The weights between input neurons and hidden neurons are updated by the following:

$$\Delta Wz_{i,j} = -\lambda \cdot \frac{\partial E}{\partial Wz_{i,j}} = \frac{1}{4}\lambda \bullet (S) \bullet \left[1\text{-}g(Unet)^2\right] \bullet W_j \bullet \left[1\text{-}g(Ynet_j)^2\right] \bullet Z_i \quad (20)$$

where λ is the learning-rate parameter of the back-propagation algorithm and is a constant. The most important point in this derivation is that the error between desired and estimated equivalent control is replaced with the corrective control of the sliding mode control.

3.4 Computation of the Corrective Control

In this NN2 structure, the corrective control u_c is obtained by NN2. Thus the structure of NN2 is easy to determine via designing SMC. The structure 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 self-tuning neuron to be the direct adaptive neural controller to replace the SMC, when the state trajectory of system goes into the boundary layer. The self-tuning neuron of SNNS can be schematically shown in Fig. 3 and mathematically expressed as [16]

$$net = I - \varepsilon \quad (21)$$

where is external input of neuron, e threshold or bias, net is internal state of neuron. The output u_c of self-tuning neuron used as the neural controller is given by

$$u_c = g(net) = \frac{\alpha[1 - e^{(-\beta - net)}]}{1 + e^{(-\beta - net)}} \quad (22)$$

where the activation function $g(\bullet)$:$R \to R$ is a modified hyperbolic tangent function; α is the saturated level; and β is the slope value. Note, that these two adjustable parameters, α and β, influence mainly the output range and the curve shape of this activation function. In this case, since the output range of u_c can be automatically tuned according to certain adaptation mechanism, it is unnecessary to consider the scaling problem of the controller. For convenience, let $\theta=[\varepsilon, \alpha, \beta]\in \mathbf{R}^3$, represent the vector of adjustable parameters. We wish to adjust θ, such that, the control objective can be achieved. The NN2 algorithm and the weight adaptation are described in the statements above.

$$J = \frac{1}{2}(S^T S) \quad (23)$$

The aim of control strategy J is to minimize successively by updating the neural controller parameter θ. In order to achieve this goal, the following theorem provides a simple and stable algorithm for parameters updating.

Differentiating in (23), and using the chain rule, we have

$$\dot{J} = \frac{dJ}{dt} = \frac{\partial J}{\partial x_i}\frac{\partial x_i}{\partial u_c}[\frac{\partial u_c}{\partial \theta}]^T \dot{\theta} \quad (24)$$

we have

$$\frac{\partial J}{\partial x_i} = e \quad (25)$$

$$\frac{\partial x_i}{\partial u_c} = \text{sgn}(\frac{\partial x_i}{\partial u_c})\,|\frac{\partial x_i}{\partial u_c}| = \text{sgn}(\frac{\partial x_i}{\partial u_c})\,|\frac{\partial x_i}{\partial u_c}| \quad (26)$$

$$\dot{\theta} = -\eta e \frac{\partial u_c}{\partial \theta}\text{sgn}(\frac{\partial x_i}{\partial u_c}) \quad (27)$$

Substituting (25), (26), and (27) into (24), we have

$$\dot{J} = -e\,\text{sgn}(\frac{\partial x_i}{\partial u_c})\,|\frac{\partial x_i}{\partial u_c}|\,\|\frac{\partial u_c}{\partial \theta}|\,\eta e \frac{\partial u_c}{\partial \theta}\text{sgn}(\frac{\partial x_i}{\partial u_c}) \quad (28)$$

$$= -\eta e^2\,|\frac{\partial x_i}{\partial u_c}|\,[\text{sgn}(\frac{\partial x_i}{\partial u_c})]^2\,\|\frac{\partial u_c}{\partial \theta}\|^2 \leq 0$$

According to the Lyapunov stability theory, this implies that the function will be monotonically decreasing for time t ≥ 0. The tuning algorithm for θ in (27), enables the neural controller to generate a suitable control force. Moreover, using (22) and (23), the partial derivative $\dot{u}_c$ with $\theta=[\varepsilon, \alpha, \beta]\in \mathbf{R}^3$ in (27) can be derived as follows:

$$\frac{\partial u_c}{\partial \varepsilon} = \frac{\beta(u_c^2 - \alpha^2)}{2\alpha} \quad (29)$$

$$\frac{\partial u_c}{\partial \alpha} = \frac{u_c}{\alpha} \quad (30)$$

$$\frac{\partial u_c}{\partial \beta} = -\frac{\alpha \bullet net(u_c^2 - \alpha^2)}{2\alpha} \quad (31)$$

Noted that the adaptation process of ε、 α and β should be stopped when the state error is acceptable. They may be sensitive to system perturbations. Additionally, the limitary concept should be considered when designing ε、 α and β. 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 SNNS is employed to approximate the non-linear mapping between the sliding input variable and the control law, the weightings of the SNNS should be regulated based on the reaching condition, $s\dot{s}$ <0.

Remark 2: The adaptive rule is derived from the steep descent rule to minimize the value of with respect to W_j and θ of SNNS. The weightings between hidden and output layers neurons can be on-line adjusted to achieve the learning ability of NN1, and updating the neural controller parameter θ of NN2. Theoretically, the SNNS 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 SNNS controller is stable and the system output error at least converges into a small error bound.

4. Simulation and Experiment of the Seesaw System

4.1. Description of System

The balancing mechanism of the seesaw is shown in Fig. 4. The previous study [17] gives the system model by using Lagrange's formulations based on principle of balance of force and torque below.

$$\begin{aligned} & m(r_1\ddot{\theta}+\ddot{x}) - mx\dot{\theta}^2 - mg\sin\theta = u \\ & I\ddot{\theta} + m[r_1(r_1\ddot{\theta}+\ddot{x}) + x^2\ddot{\theta} + 2x\dot{x}\theta] - Mgr_2\sin\theta \\ & -mg(r_1\sin\theta + x\cos\theta) = 0 \end{aligned} \quad (32)$$

The dynamical equation of the seesaw mechanism is given as follows:

$$\begin{aligned} & u + mg\sin\theta - B\dot{x} = m\ddot{x} \\ & \left(Mg\sin\theta\right)r_2 + mg\sin\left(\theta+\phi\right)\cdot\sqrt{\left(x^2+r_1^2\right)} \\ & +ur_1 - \mu\dot{\theta} = I\ddot{\theta} \end{aligned} \quad (33)$$

where I is the wedge inertia given by (34)

$$I = \frac{1}{2}\rho abc\left(\frac{a}{24}+\frac{b^2}{2}\right) \quad (34)$$

From Fig. 4-3, it can be derived below.

4.2 The Application Algorithm

For the seesaw system given in (32), the seesaw system has three variables of states. The variables are the angle that the wedge makes with the vertical line (θ), change of the angle that the wedge makes with vertical line $(\dot{\theta})$, and the position of the cart from the origin (x).

4.3 The Result of Computer Simulation

In order to prove the practicability of SNNS, simulation is done before it is applied to the practical system. The initial states of these two simulations are different. The value of parameters in Table 1 of the seesaw system in the simulation:

Simulation : The SNNS controller is applied to the seesaw system with initial state is shown as follows: the cart position is 34 centimeters and angle is 11.0 degrees. Fig. 5 and Fig. 6 shows the response of angle and the state trajectory.

4.4 The Result of Experiment

In the experiment, three variables are chosen as the inputs SNNS controller. They are angle error θ, change of angle error dθ and cart position x, respectively.

In Fig. 7, the variable are defined as follows:

$$\theta = k_\theta\left(Y_\theta - R_\theta\right) \quad (35)$$

$$x = k_x\left(Y_x - R_x\right) \quad (36)$$

$$\Delta\theta = k_{\Delta\theta}\left(\theta_{n+1} - \theta_n\right) \quad (37)$$

where R_θ and R_x are the equilibrium point of seesaw angle and cart position. Y_θ and Y_x are the outputs of the potentiometers for measuring the tilt angle of seesaw and position of the cart. θ_{n+1} and θ_n are the angle errors at the $n+1$-th and n-th sampling instants. k_θ, $k_{\Delta\theta}$ and k_x are the scaling factors. In result of Experiment 1, we add the extra force (as disturbance) to the seesaw system when it balances. The seesaw system recovers to balance quickly.

Experiment:

The initial states of the seesaw:

The position of the cart is 34 centimeters.

The angle of inverted wedge is 11.0 degrees.

We have to add a disturbance at t = 4.1s.

Fig. 8 and Fig. 10 shows the response of angle and the state trajectory.

5 Conclusion

In this paper, the SNNS control system has been proposed to solve the output tracking problem for highly nonlinear and coupling, and the complete dynamic model us difficult to obtain precisely. To verify the effectiveness of the proposed control scheme, the SNNS control system was implemented to control a seesaw system. Form the simulation results show the joint-position will be arrived at purpose, and the seesaw system can be stabilized to the equilibrium.

References:

[1] J. Y. Hung, W. Gao, and J. C. Hung, Variable structure control : A survey, *IEEE Trans. Industr. Electron.*, Vol. 40, 1993, pp. 2–22.

[2] W. Gao, Y. Wang, and A. Homaifa, Discrete-time variable structure control systems, *IEEE Trans. Industr. Electron*., Vol. 42, 1995, pp. 117–122.

[3] D. Q. Zhang and S. K. Panda, Chattering-free and fast-response sliding mode controller, *Inst. Elec. Eng. Proc.—Control Theory Appl*., Vol. 146, No. 2, 1999, pp. 171–177.

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

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

[6] S. Hui and S. H. ˙ Zak, "On discrete-time variable structure sliding mode control," Syst. Contr. Lett., Vol. 38, 1999, pp. 283–288.

[7] H. Morioka, K. Wada, A. Sabanovic, and K. Jezernik, Neural network based chattering free sliding mode control, *in Proc. SICE Annual Conference*, 1995, pp. 1303-1308.

[8] Hussain, Mohd Azlan, Ho, Pei Yee, Adaptive sliding mode control with neural network based Synthesize models, *Journal of Process Control* ,Vol. 14, No. 2, 2004, pp. 157-176.

[9] 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.

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

[11] Karakasoglu, Ahmet, Sundareshan, Malur K., A Recurrent Neural Network-based Adaptive Variable Structure Model-following Control of Robotic Manipulators, *Selfmatica*, Vol. 31, No. 10, 1995, pp. 1495-1507.

[12] G.G. Parma, B.R. de Menezes, A.P. Braga, Sliding mode algorithm for training multilayer artificial neural networks, *Electron. Lett,* Vol.34, 1998, pp. 97-98.

[13] G.G. Parma, B.R. de Menezes, A.P. Braga, Neural networks learning with sliding mode control: the sliding mode backpropagation algorithm, *Internat. J. Neural Systems*, Vol. 9, 1999, pp. 187-193.

[14] C.-L. Hwang, Neural-network-based variable structure control of electrohydraulic servo systems subject to huge uncertainties without persistent excitation, *IEEE/ASME Trans. Mechatronics*, Vol.4 ,1999, pp.50-59.

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

[16] W. D. Chang, R. C. Hwang, J. G. Hsieh, Application of an Self-Tunging Neuro to Sliding Mode Control, *IEEE Tran. On Systems, MAN, And Cyber*, Vol. 32, No. 4, 2002, pp. 517-522.

[17] M. L. Lee, H. Y. Chung, Hierarchical Fuzzy Control with Applications to Seesaw Systems, *M. S. thesis, National Central University, Department of Electrical Engineering*, 2000.

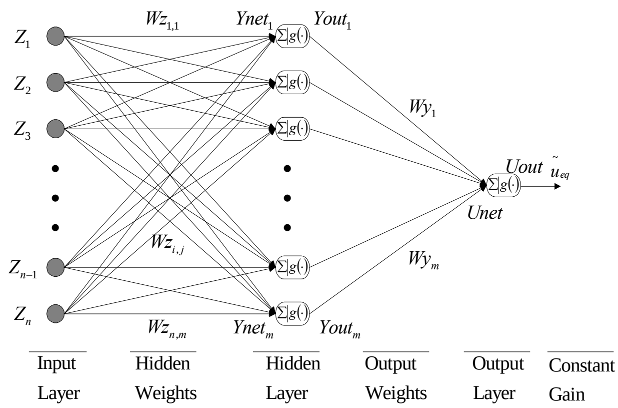

Fig. 2. The structure of NN1 of SNNS to estimate the equivalent control.

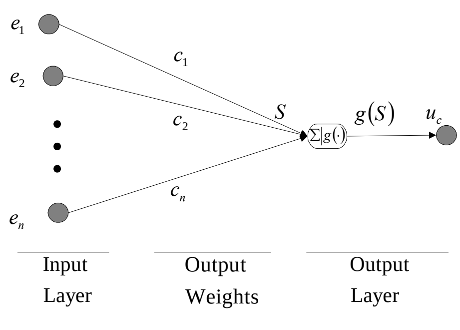

Fig. 3. The structure of NN2 of SNNS to compute the corrective control.

Parameter	Value
The wedge inertia I	0.044
The height of center of wedge r_1	0.148
The height of wedge center of mass r_2	0.123
The mass of the wedge	1.52
The mass of the *cart*	0.46
The damping coefficient of the angle	0.5
The damping coefficient of the cart	15

Table 1 The value of parameters of the seesaw system.

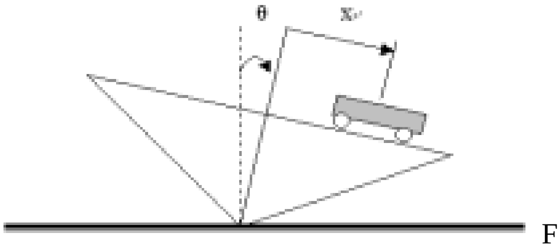

Fig. 4 The initial states of the seesaw of simulation.

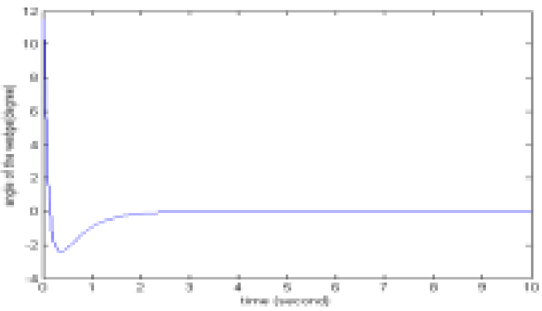

Fig. 5 Time response of angle for the SNNS Control.

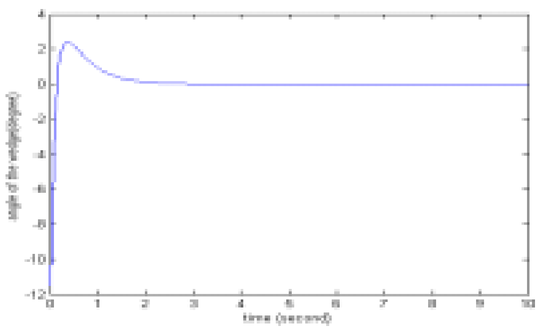

Fig. 6 Time response of angle for the SNNS Control.

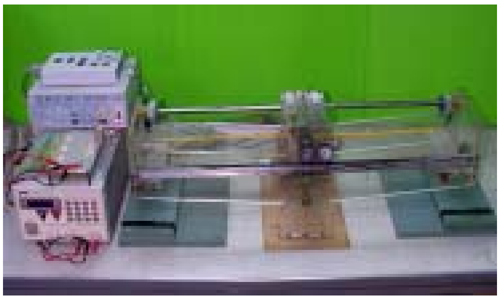

Fig. 7 The practical hardware structure of the seesaw system.

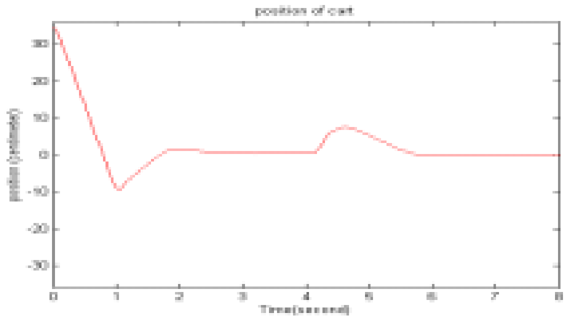

Fig. 8 The position response of the actual seesaw system.

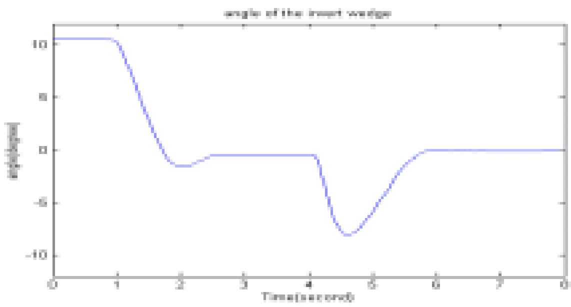

Fig. 9 The angle response of the actual seesaw system.

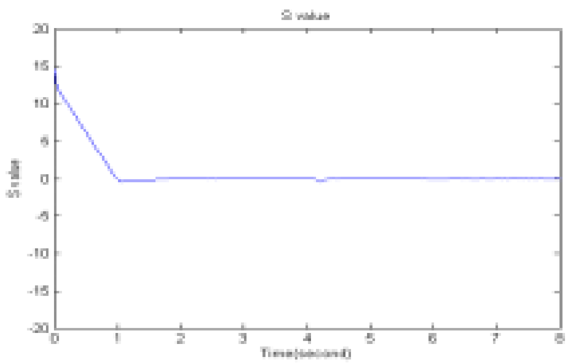

Fig. 10 The *S* response of the actual seesaw system.