Decoupled Neural Network Reference Compensation Technique for a PD Controlled Two Degrees-of-Freedom Inverted Pendulum

Seul Jung and Hyun Taek Cho

Abstract: In this paper, the decoupled neural network reference compensation technique (DRCT) is applied to the control of a two degrees-of-freedom inverted pendulum mounted on an x-y table. Neural networks are used as auxiliary controllers for both the x axis and y axis of the PD controlled inverted pendulum. The DRCT method known to compensate for uncertainties at the trajectory level is used to control both the angle of a pendulum and the position of a cart simultaneously. Implementation of an on-line neural network learning algorithm has been implemented on the DSP board of the dSpace DSP system. Experimental studies have shown successful balancing of a pendulum on an x-y plane and good position control under external disturbances as well.

Keywords: Neural network, 2 DOF inverted pendulum, on-line learning, DSP system.

1. INTRODUCTION

Inverted pendulum application using various control methods has been a typical example for advanced control education, as well as interesting research. Control of an inverted pendulum has been considered as a fascinating, but difficult problem to solve since the system has very challenging characteristics such as nonlinearity and a single-input multi-output structure [1, 2]. Many successful results using the advanced control theories for balancing the inverted pendulum using a cart have been reported throughout various literatures [1, 3]. Those successful results have been mainly focused on balancing the pendulum, rather than on controlling the position of the cart. Recently, successful control of both the angle and the position of the inverted pendulum has been demonstrated by practical experiments [4-6].

The nature of controlling both the angle of the pendulum and the position of the cart with a single input force has remained as an open nonlinear problem to be overcome. The difficulty of controlling both the angle and the position of the inverted pendulum system comes from different dynamic movement patterns of the pendulum and the cart. For example, let us consider the case that a controller for the cart tries to move toward one direction to minimize a positional error while a controller for the pendulum tries to move in the opposite direction to minimize the angle error. When this contradiction occurs, it is difficult to decide the suitable control law. This is one reason that the conventional fixed PD controller cannot control both the angle of the pendulum and the position of the cart concurrently. To tackle this problem, suitable controllers' gains for various cases should be considered.

Fuzzy algorithm is a good candidate for solving this type of problem, but the assessment of fuzzy rules for fuzzification does not simply satisfy both objectives [7-9]. Several trial and error experiments are required to obtain a certain satisfaction. Visual feedback control for the inverted pendulum has also been proposed [9, 10]. Rather than solely depending upon encoder signals, controlling the balance of the pendulum relies on visual feedback. The performance of visual feedback control is dependent upon the accuracy of the vision system.

Neural network based control is another good candidate for this application. In our previous researches, control of both the angle and the position of the inverted pendulum system has been successfully performed on a large x-y table robot [4, 6]. However, due to the large size of our previous system, position control and balancing of the inverted pendulum of both axes on the x-y plane failed. One of the reasons for the failure was that the actuated motor could not generate enough torque for rapid movement of the axis due to the heavy weight of one side of the two axes.

In this paper, as an extension of our previous research [4, 6], control of the inverted pendulum on the x-y table is revisited. Like the balancing of a stick on

Manuscript received February 20, 2003; revised October 16, 2003; accepted December 15, 2003. Recommended by Editorial Board member Jin Young Choi under the direction of Editor Jin Bae Park. This research has been supported by the Korea Research Foundation with the contract of KRF 2002-002-D000765. The authors would like to thank them for their kind support.

Seul Jung and Hyun Taek Cho are with the Department of Mechatronics Engineering, Chungnam National University, 220 Gung-dong, Yuseong-gu, Daejeon 306-764, Korea (e-mail: jungs@cnu.ac.kr, hyuntkc@empal.com).

a palm, a pendulum can freely move on the x-y plane. An x-y table is newly built and its size is relatively reduced to generate faster movements. Neural networks are also used as auxiliary controllers to help the PD controller for the system to minimize the errors of angles and positions of each axis. Differing from the previous research, a decoupled neural network structure is used. Decoupled neural networks structure means that two separate neural networks are used for controlling each axis of the x-y table instead of using a single neural network. Since the structure of the x-y table is a more likely decoupled system, the use of two separate neural networks is suitable for eliminating any coupling effects.

For experiments, the newly designed smaller x-y table with the dSpace DSP system is implemented. Interface between the robot and the DSP system has been implemented to drive the motors of each axis. On-line neural computation algorithm is developed and implemented on the DSP board of the dSpace system to achieve real time control. Successful results of maintaining balance of the pendulum and position control of the cart on the x-y plane have been obtained by the proposed control algorithm.

2. SYSTEM STRUCTURE

2.1. Overall system structure

The overall system structure is shown in Fig. 1. The system consists of three parts: a controller, a pendulum on an x-y table, and actuators. The control component includes a computer with the dSpace DSP board and interface. The DSP board is used to calculate the neural network learning algorithm in an on-line fashion. The main body includes an inverted pendulum and an x-y table. The size of the x-y table is 0.8m $\times$0.9m. As shown in Fig. 1, a 2-DOF inverted pendulum is mounted on the x-axis of the robot. The x-axis moves along the y-axis with an LM Guide. Two axes are actuated by two DC motors through timing belts. Belt tension is often considered as an uncertainty.

The neural network control algorithm is implemented on the DSP board and the board generates PWM signals to the motor drivers.

2.2. Inverted pendulum

The dynamics of an inverted pendulum can be modeled in x and y directions separately. The dynamic equation for the x axis is

$$J\ddot{\theta}_x = v\sin\theta_x - hL\cos\theta_x + b_{\theta_x}\theta_x, \quad (1)$$

where J is an inertia of a pendulum, θ_x is the angle of x axis, L is the length of a pendulum, b_{θ_x} is friction constant, and v and h are vertical force and horizontal force vectors, respectively. Vertical

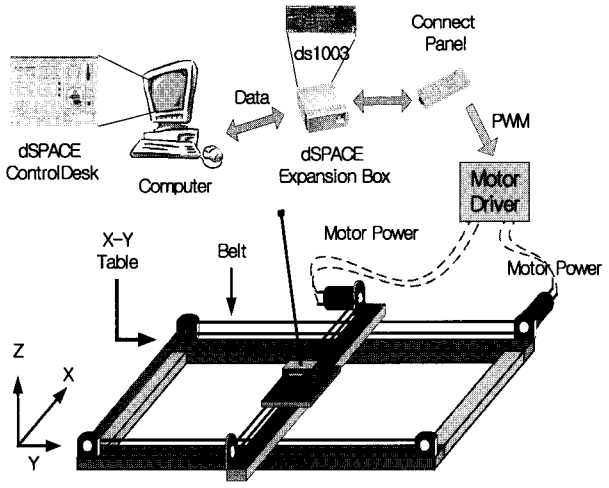


Fig. 1. The overall system.

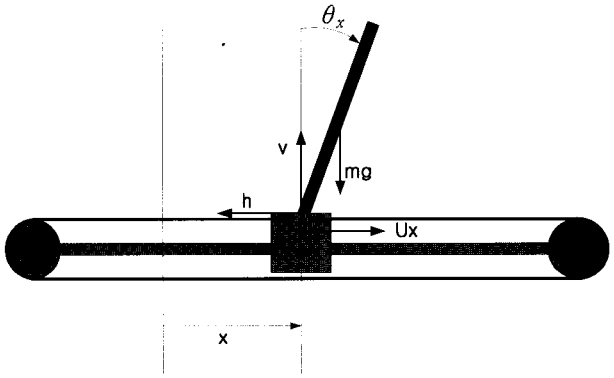


Fig. 2. Inverted pendulum model.

and horizontal forces are given as follows:

$$\begin{aligned} h &= m(\ddot{x} + L\ddot{\theta}_x\cos\theta_x - L\dot{\theta}_x^2\sin\theta_x), \\ v - mg &= -mL\ddot{x}\sin\theta_x - mL\dot{\theta}_x^2\cos\theta_x, \end{aligned} \quad (2)$$

where m is the mass of a pendulum and x is the displacement in the x axis.
For a cart,

$$M_x\ddot{x} = u_x - h, \quad (3)$$

where M_x is the mass of a cart in the x axis.
Solving for $\ddot{\theta}_x$ and $\ddot{x}$ by combining (1), (2) and (3) yields the dynamic model of an inverted pendulum shown as follows:

$$(M_x + m)\ddot{x} + mL\cos\theta_x\ddot{\theta}_x - mL\sin\theta_x\dot{\theta}_x^2 + b_x\dot{x} = u_x, \quad (4)$$

$$(J + mL^2)\ddot{\theta}_x + mL\cos\theta_x\ddot{x} - mgL\sin\theta_x - b_{\theta_x}\dot{\theta}_x = 0. \quad (5)$$

We know from equations (4) and (5) that the

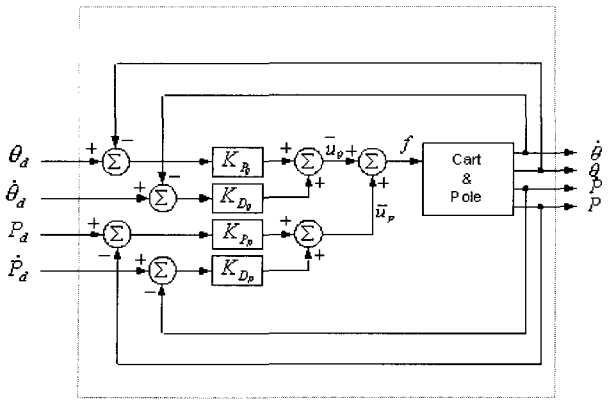


Fig. 3. PD controlled pendulum system.

system has a single input u_x and two outputs x and θ for an x axis.

Combining (4) and (5) yields

$$\ddot{\theta}_x = \frac{M'(mgLs\theta_x + b_{\theta_x}\dot{\theta}_x) - mLc\theta_x(mLs\theta_x\dot{\theta}_x^2 - b_x\dot{x} + u_x)}{M'J' - m^2L^2c^2\theta_x} \tag{6}$$

$$\ddot{x} = \frac{J'(mLs\theta_x\dot{\theta}_x^2 - b_x\dot{x} + u_x) - mLc\theta_x(mLgs\theta_x + b_{\theta_x}\dot{\theta}_x)}{M'J' - m^2L^2c^2\theta_x} \tag{7}$$

where

$M' = M_x + m, J' = J + mL^2, s\theta = \sin\theta, c\theta = \cos\theta$.

For the y axis, we have a similar dynamic equation as

$$\ddot{\theta}_y = \frac{M_y(mgLs\theta_y + b_{\theta_y}\dot{\theta}_y) - mLc\theta_y(mLs\theta_y\dot{\theta}_y^2 - b_y\dot{x} + u_y)}{M_yJ' - m^2L^2c^2\theta_y} \tag{8}$$

$$\ddot{y} = \frac{J'(mLs\theta_y\dot{\theta}_y^2 - b_y\dot{x} + u_y) - mLc\theta_y(mLgs\theta_y + b_{\theta_y}\dot{\theta}_y)}{M_yJ' - m^2L^2c^2\theta_y} \tag{9}$$

Note that M_y includes the mass of an x axis because the y axis carries the x axis at all times.

3. PD CONTROL OF INVERTED PENDULUM

In this section, the non-model based PD control is presented. It is known that a PD control can stabilize the second order system. Even though we have derived the dynamic equations in the previous section, it is very difficult to obtain the exact dynamic equations including uncertain nonlinear terms. Therefore, ignoring the system dynamics, the most uncomplicated method is to use the simple linear PD controllers as the main controllers.

Fig. 3 depicts the PD control structure of controlling the inverted pendulum in one axis. In order to control a two degrees-of-freedom pendulum, the combined two separate control inputs are required.

The control input u_x for an x axis is formed by adding two controllers' outputs from an angle error and a displacement error. The errors are defined as follows:

$$e_\theta = \theta_d - \theta, \quad e_x = x_d - x, \tag{10}$$

where e_θ is the angle error and e_x is the positional error.

The PD controller is now formed as

$$\bar{u}_x = \bar{u}_{\theta x} + \bar{u}_{px}, \tag{11}$$

$$\begin{aligned} \bar{u}_{\theta x} &= k_{p\theta x}\cdot e_\theta + k_{d\theta x}\cdot \dot{e}_\theta, \\ \bar{u}_{px} &= k_{ppx}\cdot e_x + k_{dpx}\cdot \dot{e}_x, \end{aligned} \tag{12}$$

where $\bar{u}_{\theta x}$ and $\bar{u}_{px}$ are nominal control inputs for a pendulum and for a cart of the x axis, respectively.

The control input $\bar{u}_{\theta y}$ and $\bar{u}_{py}$ for y axis can be represented in a similar way. The difficulty of controlling the pendulum angle and the cart position simultaneously comes from different configurations $\bar{u}_\theta$ and $\bar{u}_p$.

The use of diagonal controller gains can decouple two axes, but there still are coupled effects such as the coriolis force and other unknown nonlinear terms. As a result the linear controller cannot cope with system parameter variations well enough, resulting in the failure of position control of the cart [6]. Extensive simulation studies have been done.

In order to improve the control performance, two neural networks are used. Since the neural network is nonlinear it is a good candidate for nonlinear system control. A decoupled neural network structure for controlling each axis separately helps the system to be more decoupled. Fig. 5 indicates the decoupled neural network structure for a single axis.

4. REFERENCE COMPENSATION TECHNIQUE FOR NN CONTROL

In this section, one of the on-line learning algorithms for neural network control is presented. The algorithm called the reference compensation technique has been proposed and it has shown good performances in the robot position control [11, 12].

This scheme is identical to the feedback error learning method in that it performs inverse dynamic control, but it is also different in that compensation is done without modifying pre-fixed linear controllers. This control scheme is depicted in Fig. 4. The basic

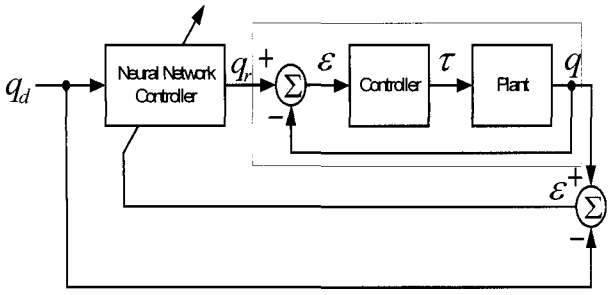


Fig. 4. Reference compensation technique scheme.

concept of this scheme is that the NN controller acts as the inverse of the system under PD control so that the system response q tracks the desired response q_d with minimal distortion. Neural networks are placed in front of the closed loop controlled system as pre-filters as seen in Fig. 4. Neural network outputs are added to reference trajectories. Added terms are subtracted by output signals to generate error signals ε. The errors are multiplied by controller gains.

Therefore, they eventually shape the reference input trajectory q_r in such a way that the output error ε is minimized to zero.

Our proposed control block diagram is shown in Fig. 5. The inverted pendulum system is controlled by PD control and neural network control.

From Fig. 5, the PD controller with compensating signals from a neural network forms the following control inputs:

$$u_{\theta x} = k_{p\theta}(e_\theta + \phi_\theta) + k_{d\theta}(\dot{e}_\theta + \dot{\phi}_\theta), \tag{13}$$

$$u_{px} = k_{px}(e_x + \phi_p) + k_{dx}(\dot{e}_x + \dot{\phi}_p). \tag{14}$$

The derivatives $\dot{\phi}_\theta$ and $\dot{\phi}_p$ of neural network outputs ϕ_θ and ϕ_p are obtained by the finite difference method, respectively.

The total control input for one axis is the sum of (13) and (14).

$$u_x = \bar{u}_{\theta x} + \bar{u}_{px} + \Phi_x. \tag{15}$$

The new control input is actually the addition of a neural network compensating signal to nominal control input $\bar{u}_x$. These compensating terms compensate for any dynamic uncertainties that are not modeled.

Define the neural network output as

$$\Phi_x = \Phi_{\theta x} + \Phi_{px}, \tag{16}$$

where $\Phi_{\theta x} = k_{p\theta}\dot{\phi}_{\theta x} + k_{d\theta}\phi_{\theta x}$ and

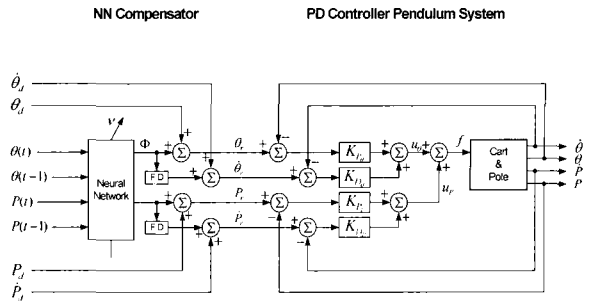


Fig. 5. Neural network control block diagram for a single axis.

$$\Phi_x = k_{px}\phi_{px} + k_{dx}\dot{\phi}_{px}$$

Also note that the compensating signal Φ_x solves the sign problem between $\bar{u}_{\theta x}$ and $\bar{u}_{px}$.

Since compensation is performed at the trajectory level, those compensating signals are amplified through controller gains so that the magnitudes of those pre-filtered signals are small compared with ones in other auxiliary type controllers [5]. In the paper [5], the neural network is used to adjust PID controller gains. The goal of minimizing error is the same with the proposed RCT algorithm, but the difference comes from the controlling structure. The RCT algorithm is known to have the advantage of outer loop control without modifying the internal control structure [11, 12].

5. LEARNING ALGORITHM

The neural network structure for a single axis is shown in Fig. 6. A two layered feed-forward structure is used. Input patterns are the combination of position and angle errors. For experiments, 9 hidden units are used. Selection of the number of hidden units is based on trial and error. Delayed states are used as inputs of a neural network to give dynamics into the neural network. Two neural network outputs are used, one for compensating an angle error and the other for a position error are used for a single axis.

Nonlinear function of hidden unit is tangent hyperbolic function.

$$f(x) = \frac{1-\exp(-x)}{1+\exp(-x)}. \tag{17}$$

From (4), (5), and (15), we have the following closed loop error equation:

$$\bar{u}_{\theta x} + \bar{u}_{px} = f(\ddot{\theta}, \dot{\theta}, \theta, \ddot{x}, \dot{x}, x) - \Phi_x. \tag{18}$$

Define the training signal v with nominal control inputs as

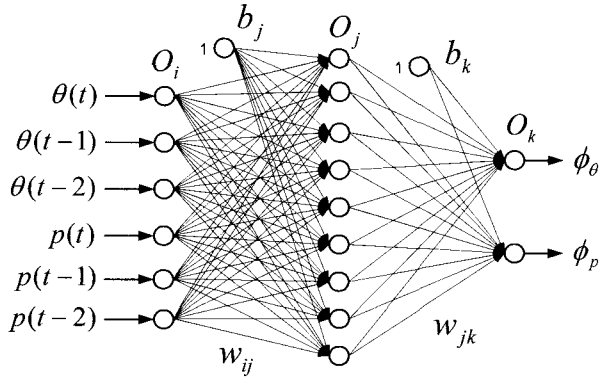


Fig. 6. The structure of neural network for a single axis.

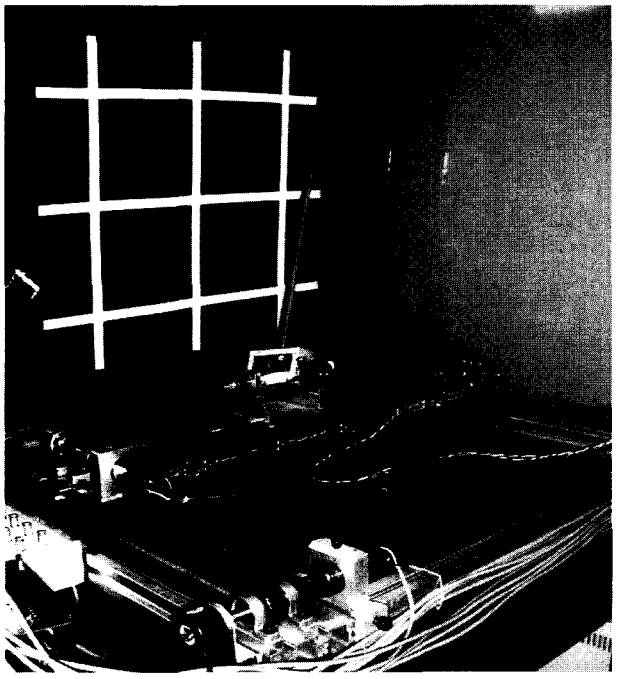


Fig. 7. Real inverted pendulum system.

$$v = \bar{u}_{\theta x} + \bar{u}_{px} . \tag{19}$$

Then, at the convergence $v = 0$, from equation (18) an ideal neural network output becomes the inverse dynamics of the system.

$$\Phi \cong f(\ddot{\theta}, \dot{\theta}, \theta, \ddot{x}, \dot{x}, x) . \tag{20}$$

So, ultimately, the inverse dynamic control can be achieved. This is the difference in the control structure from those of other PID tuning methods [5].

The objective function is defined to minimize the error v as

$$E = \frac{1}{2} v^2 . \tag{21}$$

In order to use the steepest decent algorithm, the gradient should be obtained. Differentiating (21) yields the gradient as

$$\begin{aligned} \frac{\partial E}{\partial w} &= \frac{\partial E}{\partial v}\frac{\partial v}{\partial w} = v\frac{\partial v}{\partial w} \\ &= -v\frac{\partial \Phi}{\partial w} = -v\left(\frac{\partial \Phi_{\theta x}}{\partial w} + \frac{\partial \Phi_{px}}{\partial w}\right), \end{aligned} \tag{22}$$

where $\frac{\partial \Phi_{\theta x}}{\partial w} = k_{p\theta}\frac{\partial \phi_{\theta x}}{\partial w}$ and $\frac{\partial \Phi_{px}}{\partial w} = k_{px}\frac{\partial \phi_{px}}{\partial w}$.

By using the gradient function in (22) the back-propagation algorithm can be used. The weight change is formed as

$$\Delta w(t) = \eta \frac{\partial E}{\partial w} v + \alpha \Delta w(t-1) . \tag{23}$$

Weights are updated as

$$w(t+1) = w(t) + \Delta w(t) , \tag{24}$$

where η is a learning rate and α is a momentum coefficient.

Neural network weights are updated at every sampling time. Since rapid sampling time can be achieved with the help of DSP hardware technology, real time control of a neural network becomes possible. Even on-line learning and control of the system can be performed. This means that no *a priori* learning before control action is required.

In the next section, experimental studies of neural network control are presented.

6. EXPERIMENTAL SETUPS

The overall system structure is shown in Fig. 7. The whole system consists of three parts: an x-y table robot, the dSpace DSP system, and a PC. The DSP board is used to implement the neural network algorithm in real time, which requires a huge calculation time. Numerical values of neural network outputs calculated by the DSP board are converted to PWM signals for motor drivers.

The DC motors driven by its own drivers commanded from the DSP board mounted actually actuate each axis of the x-y table. Encoders mounted on each axis sense rotations and those sensed data are sent to the encoder board. An encoder board collects data and converts them into digital values that can be used as a feedback in the control loop. Table 1 shows the specifications of actuators.

Table 1. Specification of actuators.

Devices	Specifications
DC motor	24V/70W
Encoder	2000 counts/turn
Gear ratio	5:1

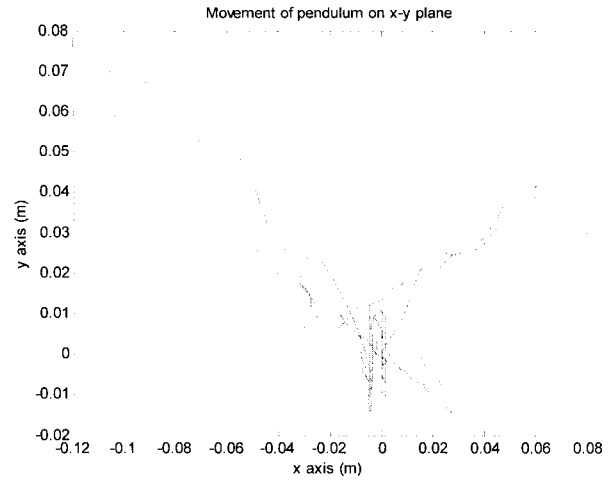


Fig. 8. Movement of pendulum on x-y plane.

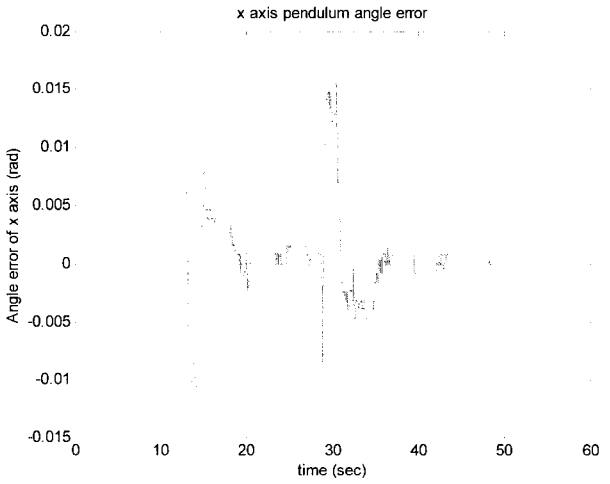


Fig. 9. X axis angle error.

The weight of the pendulum is 0.35kg without a weight and 0.5Kg with a weight at the top.

There are two encoders mounted on the cart to measure the rotational angles of both axes. The inverted pendulum can fall in any direction while a cart can move on the x-y plane. The movement of the cart is measured by encoders attached to the motor joint. The x-y table is actuated by DC motors through timing belts. Since the x axis cart moves on the top of y axis, there are coupling effects, and this leads to the reason that control of the y axis is more difficult.

For the controller gains, $k_{d\theta_x}$ = 0.8, $k_{p\theta_x}$ = 6, k_{dpx} = -0.6, k_{ppx} = -0.5 for x axis and $k_{d\theta_y}$ = 1.4, $k_{p\theta_y}$ = 8.5, k_{dpy} = -0.95, k_{ppy} = -0.8 for y axis are used. The PD gains are optimized by trial errors basis. However, gain values are small enough to maintain stability so that neural networks are allowed to perform the most of control. We found from experiments that if the PD gains are set too large, performance is even worse. As such, it is better to leave the

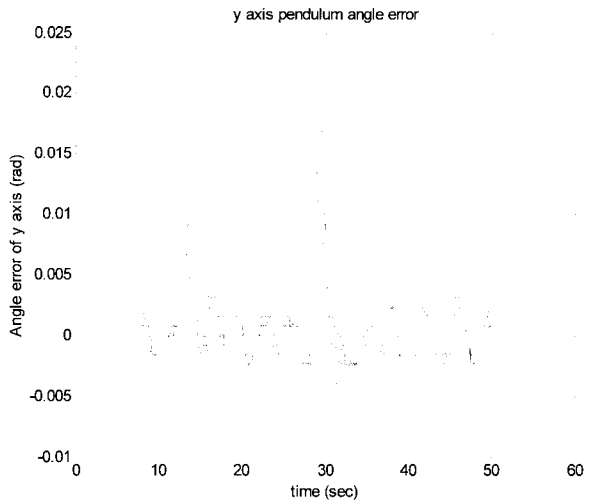


Fig. 10. Y axis angle error.

control to neural networks after stabilizing the system by PD controllers.

For the neural network structure, 9 hidden units for each neural network are used. Learning rate η_x = 0.0011, η_y = 0.001 and momentum α_x = 0.15, α_y = 0.05 are optimized. These constant values are optimized by trial and error basis. We found that the learning rate is the most sensitive variable to performance. Selecting a larger learning rate gives faster convergence of errors as well as occasional instabilities. The overall sampling time is 1 KHz.

7. EXPERIMENTAL RESULTS

Initially the pendulum is located at (0, 0) on the x-y plane. The pendulum is well maintained at the initial position until there is an external hit. Then the movement of the pendulum is affected by an external disturbance by the hand. The pendulum is required to move back to the initial position.

Fig. 8 shows the movement of the pendulum on the x-y plane after following hits.

We can see that there have been two disturbances by hitting the pendulum in diagonal directions. The pendulum has kept returning to the initial position while maintaining the balance of the angle of the pendulum shown in Fig. 9. Traveling distances are from - 10.5 cm to 6 cm in the x axis and from -1.5 cm to 4 cm in the y axis. The pendulum angle errors for both axes are shown in Figs. 9 and 10 for each direction. Large overshoots by disturbances are observed at approximately both 13 seconds and 29 seconds and they settled down quickly.

The positional errors are shown in Figs. 11 and 12. Comparing performances of two axes' movements, the x axis position error is less than 2 mm at steady state, but that of the y axis is about 2 cm. The y axis keeps oscillating with a small bound.

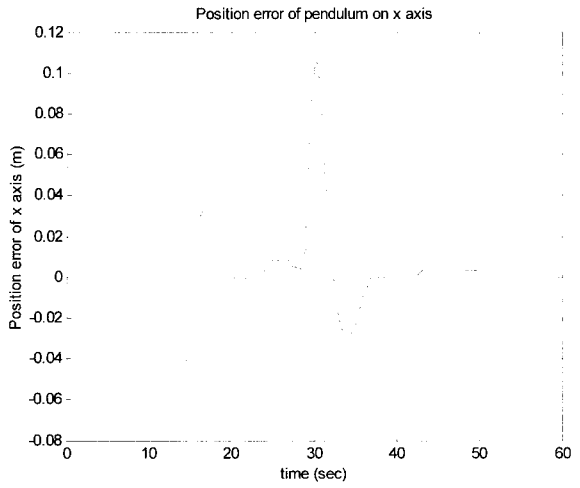


Fig. 11. X axis position error.

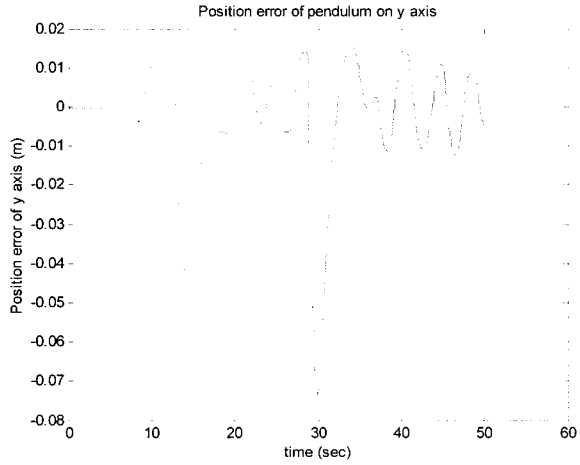


Fig. 12. Y axis position error.

The error eventually converges to zero, but it takes time. One reason for this is that the y axis control is more difficult than the x axis control; because the y axis continually carries the x axis and coupling effects are well aware of this factor, requiring additional torques to generate faster movements.

A further matter is one side actuation through timing belts. This causes minimal jerk by unbalanced movements to each axis even though an LM guide is used. However, position tracking of the cart as well as upright position of the pendulum is satisfactory.

8. CONCLUSIONS

The reference compensation technique of the neural network for balancing a two degrees-of-freedom PD controlled inverted pendulum has been presented. The DRCT was very effective in balancing the nonlinear inverted pendulum on the x-y plane by decoupling the x and y axis. The neural compensator helps conventional PD controllers to control the angle of the pendulum and the position of the cart simultaneously.

Even though balancing of the pendulum was quite successful, controlling the cart position shows small oscillatory errors in the y axis. One of the reasons might be the location of optical encoders. Shown in Fig. 7, encoders are mounted at the axis of rotating rod, not directly at the motor axis.

Another possibility is the velocity estimation. Simple velocity estimation using the finite difference might not provide good estimation for slow movement of the pendulum. Finally, the y axis is actuated by one side and not the other, which leads to the misalignment that causes oscillation. This yields further nonlinear characteristics such as backlash. These are left as future works to be performed.

REFERENCES

[1] S. Omatu, Y. Kishida, and M. Yoshioka, "Neuro-control for single-input multi-output systems," *Proc. of IEEE Conf. on Knowledge Based Intelligent Electronics Systems*, pp. 202-205, 1998.

[2] I. Fantoni and R. Lozano, *Non-linear Control for Under-actuated Mechanical Systems*, Springer, 2002.

[3] M. W. Spong, "Swing up control of the acrobat," *Proc. of IEEE Conf. on Robotics and Automations*, pp. 2356-2361, 1994.

[4] S. Jung and S. B. Yim, "Reference compensation technique using neural network for controlling a large x-y table robot," *International Symposium on Robotics and Automations*, pp. 461-455, 2000.

[5] S. Omatu, T. Fujinaka, and M. Yoshioka, "Neuro-pid control for inverted single and double pendulums," *IEEE Conference*, pp. 2685-2690, 2000.

[6] S. Jung and S. B. Yim, "Experimental studies of neural network control for nonlinear systems," *ICASE journal of Control, Automation and Systems Engineering*, pp. 918-926, 2001.

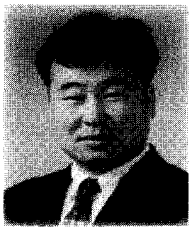
[7] T. Lahdhiri, C. Carnaland, and A. T. Alouani, "Cart-pendulum balancing problem using fuzzy logic control," *Proc. of IEEE Southeastcon'94*, pp. 393-397, 1994.

[8] T. Hung, M. Yeh, and H. C. Lu, "A PI-like fuzzy controller implementation for the inverted pendulum system," *Proc. of IEEE Conference on Intelligent Processing Systems,* pp. 218-222, 1997.

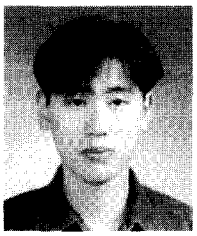
[9] M. E. Magana and F. Holzapfel, "Fuzzy logic control of an inverted pendulum with vision feedback," *IEEE Trans. on Education*, vol. 41, no. 2, pp. 165-170, 1998.

[10] L. Wenzel, N. Vazquez, and ND. N. Jamal, "Computer vision based inverted pendulum," *Proc. of IEEE Conference on Instrumentation and Measurement Technology*, pp. 1319-1323,

vol. 3, 2000.
[11] S. Jung and T. C. Hsia, "On an effective design approach of cartesian space neural network control of robot manipulators," *Robotica*, vol. 15, Part 3, pp. 305-312, 1997.
[12] S. Jung and T. C. Hsia, "Neural network inverse control techniques for PD controlled robot manipulators," *Robotica*, pp. 461-455, vol. 19, no. 3, 2000.



Seul Jung received the B.S. degree in Electrical & Computer Engineering from Wayne State University in 1988, and the M.S and Ph.D. degrees from the University of California, Davis in 1991 and 1996, respectively. After a post doctoral position at the AHMCT (Advanced Highway Maintenance and Construction Technology) of University of California, Davis, he joined the Chungnam National University in 1997. He is currently an Associate Professor at the Department of Mechatronic Engineering, Chungnam National University. His research interests include a hardware implementation of intelligent controllers, intelligent emotional engineering, and intelligent robotic systems.



Hyun-Taek Cho received the B.S. and M.S. degrees in Mechatronics Engineering from Chungnam National University, in 2000 and 2003, respectively. He is now a Researcher in the Department of Intelligent Precision Machines at KIMM. His research interests involve neural network controllers and DSP systems.