Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Implementation of a Neural Network Controller on a DSP for Controlling an Inverted Pendulum System on an X-Y Plane

Sung S. Kim*, Geun H. Lee, and Seul Jung*****

Agency for Defense Development, Daejeon, Korea, Mechatronics Engineering Department, Chungnam National University, Daejeon, Korea (Tel: 82-42-821-7232; e-mail: sadthink@ paran.com).*
**** Mechatronics Engineering Department, Chungnam National University, Daejeon, Korea, (Tel:82-42-821-6876;e-mail: jungs@cnu.ac.kr)}*

Abstract: This paper presents the hardware implementation of a neural network controller for controlling an inverted pendulum system on an x-y plane robot. The inverted pendulum system can move on an x-y plane while balancing the angle of the pendulum. Neural network algorithm is implemented on a cost effective DSP board in association with an FPGA chip. The reference compensation technique of neural network control scheme is used for on-line learning and control of the inverted pendulum system. Experimental results of tracking the circular trajectory while balancing the pendulum demonstrate to confirm the successful performance of the neural network hardware.

1. INTRODUCTION

Recently, interests in the intelligent system area have enormously been increased. Researches on intelligent systems become a very important subject in a variety of engineering fields such as control systems, robot systems, and signal processing systems. One of the next frontier research topics seems to develop intelligent systems that apply the intelligence to the systems to solve very complicated problems. Although the research history on intelligent systems has not been long compared with that of other control subjects such as adaptive control, optimal control, and robust control, many progresses have been made by enormous attraction by researchers.

After 1980s, the era of intelligent control begins and many successful results have been presented. Neural network and fuzzy logics are two major tools for developing intelligent control systems. Fuzzy logics are more practical and can be relatively implemented as hardware since real time calculation can be easily satisfied although it suffers from the difficulty of determining rules.

Starting from developing the theoretical analysis of how to use neural network in control systems as an online learning and control tool, simulation results have been presented. In recent days, however, an actual hardware implementation of neural network controllers becomes more feasible and it has been implemented to confirm the theoretical analysis. Successful demonstrations of achieving real-time control applications have been presented.

As a test-bed system for confirming neural network control schemes, the inverted pendulum control system has been considered as a prototype example of nonlinear system control applications whose structure is a single-input multiple-output(SIMO) where one single input force has to control both the angle of the pendulum and the position of the cart at the same time(Hung *et al*., 1997, Mogana *et al*., 1998).

Recently, numerous examples of more challenging inverted pendulum systems as extensions of the one dimensional pendulum system have been presented. Control of two-degrees-of-freedom inverted pendulum or a spherical pendulum moving on the x-y plane has been proposed and successfully demonstrated (Kim *et al*., 2004). Further extensions of simple pendulum models include an acrobat(Spong *et al*., 1995), and the Furuta pendulum. Increasing interest in inverted pendulum systems extends the category to a more interesting and challenging 3D inverted pendulum problem(Shen *et al*., 2004) .

In this paper, the hardware implementation of a neural network controller for controlling an inverted pendulum system on an x-y plane robot is presented. The inverted pendulum system has two degrees-of-freedom to move on an x-y plane while balancing the angle of the pendulum. In our previous researches, control of the inverted pendulum system on x-y plane has been successfully demonstrated with expensive DSP systems(Jung *et al*., 2004) . Here, we develop cost effective neural network control hardware. Neural network learning and control algorithm is implemented on a cost effective DSP board and the PID control algorithm is implemented on an FPGA chip to form the reference compensation technique scheme of neural network. Experimental results of commanding to track the circular trajectory while balancing the pendulum show successful demonstration to confirm the performance of the neural network hardware.

2. INVERTED PENDULUM SYSTEM ON AN X-Y PLANE

The inverted pendulum system is shown in Fig. 1. The pendulum can move on the x-y plane controlled by an x-y table robot. It is a more challenging task compared to one axis inverted pendulum system where coupling effects between two axes do not exist. The goal is to maintain balancing the pendulum while tracking the desired trajectory x, y by regulating two control inputs, u_x, u_y.

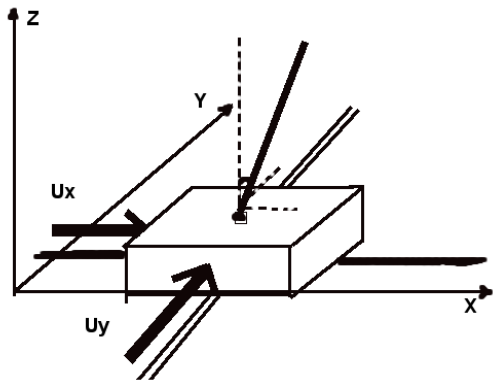


Fig. 1 Inverted pendulum system on the plane

3. NEURAL NETWORK CONTROL

When neural network is used in the control system, neural network is used as an auxiliary controller to help minimizing errors. Without primary controllers, it is very difficult for neural network alone to stabilize the system in an on-line fashion. Therefore, in most of neural network control schemes, the primary controllers are used to stabilize the system and neural network eliminates tracking errors due to nonlinearities or uncertainties in the system. The detailed block diagram of the RCT scheme for controlling the x axis of the inverted pendulum system is shown in Fig. 2.

To regulate the inverted pendulum system on the plane, we need a replica of Fig. 2 for the y axis control. Two separate neural networks are used for controlling the inverted pendulum system on the plane. Each axis is decoupled and controlled separately although axes are coupled. Decoupling effects by neural network is expected along with nonlinear uncertainties compensation.

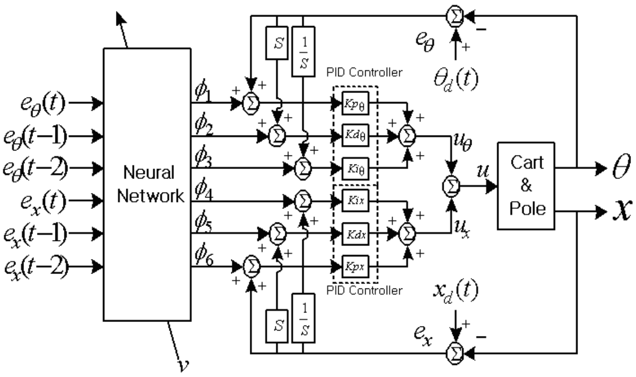


Fig. 2. Neural network control block diagram for x axis

The concept of the RCT is to compensate the system controlled by predetermined controllers by closing another outer loop. The neural network output signal ϕ_i compensate at the desired trajectory q_d to modify the control input u by minimizing the output error, $e = q_d - q$.

The detailed control structure for the pendulum system is shown in Fig. 2. Neural network outputs are added to tracking errors to form PID controller outputs. A control input $u_{x\theta}$ for the pendulum angle and a control input u_{xp} for the cart position are summed together to generate an input force u_x to the system.

The pendulum angle error of the x axis is defined as

$$e_{x\theta} = \theta_{xd} - \theta_x, \qquad (1)$$

where θ_{xd} is the desired angle of the pendulum and θ_x is the actual angle of the pendulum .

Then a PID control input for the angle control is given by

$$\begin{aligned} u_{x\theta} &= k_{xp\theta} e_{x\theta}(t) + k_{xi\theta} \int e_{x\theta}(t)dt + k_{xd\theta} \dot{e}_{x\theta}(t) \\ &+ k_{xp\theta}\phi_{x1} + k_{xd\theta}\phi_{x2} + k_{xi\theta}\phi_{x3} \end{aligned}, \qquad (2)$$

where $k_{xp\theta}, k_{xd\theta}, k_{xi\theta}$ are PID gains for the pendulum control and $\phi_{x1}, \phi_{x2}, \phi_{x3}$ are neural network outputs.

The mobile pendulum position error is defined by

$$e_{xp} = x_d - x, \qquad (3)$$

where x_d is the desired cart position and x is the actual position of the cart.

The PID control input for the position control is

$$\begin{aligned} u_{xp} &= k_{px} e_x(t) + k_{ix} \int e_x(t)dt + k_{dx} \dot{e}_x(t) \\ &+ k_{ix}\phi_{x4} + k_{dx}\phi_{x5} + k_{px}\phi_{x6} \end{aligned} \qquad (4)$$

where k_{ix}, k_{dx}, k_{px} are PID gains for the cart control and $\phi_{x4}, \phi_{x5}, \phi_{x6}$ are neural network outputs.

The overall control input is the sum of two PID controller outputs, $u_{x\theta}$ in (2) and u_{xp} in (4)

$$u_x = u_{xp} + u_{x\theta}. \qquad (5)$$

In the same manners, the control input for the y axis is given by

$$u_y = u_{yp} + u_{y\theta}. \qquad (6)$$

where

$$u_{y\theta} = k_{yp\theta} e_{y\theta}(t) + k_{yi\theta}\int e_{y\theta}(t)dt + k_{yd\theta}\dot{e}_{y\theta}(t) + k_{yp\theta}\phi_{y1} + k_{yd\theta}\phi_{y2} + k_{yi\theta}\phi_{y3}$$

$$u_{yp} = k_{py}e_y(t) + k_{iy}\int e_y(t)dt + k_{dy}\dot{e}_y(t) + k_{iy}\phi_{y4} + k_{dy}\phi_{y5} + k_{py}\phi_{y6}.$$

4. NEURAL NETWORK LEARNING

Here, a two layered feed-forward neural network structure consists of w_{ij}, the weight between the input and the hidden layer, w_{jk}, the weight between the hidden and output layer, b_j, the bias weight for the hidden layer, and b_k, the bias weight for output layer. For the neural network structure, 6 inputs, 9 hidden units, and 6 output units are used as shown in Fig. 3. Inputs to neural networks are selected as error signals and their delayed signals to represent system dynamics by the neural network. The number of hidden units is selected experimentally. The number of neural network outputs is set to 6 to compensate at each component of 2 PID controllers in equations (2) and (4).

The nonlinear function for a neuron is the hyperbolic tangent function whose bound is ± 1.

$$f_{th}(x) = \frac{1-\exp(-x)}{1+\exp(-x)}. \qquad (7)$$

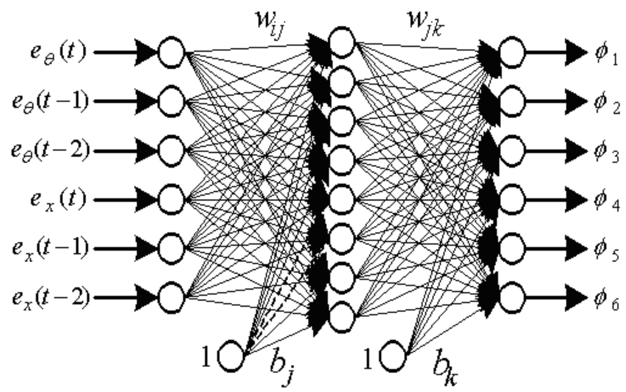


Fig. 3 Neural network structure for the x axis

From equation (2), the control input for the pendulum angle becomes

$$u_\theta = k_{P\theta}e_\theta(t) + k_{i\theta}\int e_\theta(t)dt + k_{d\theta}\dot{e}_\theta(t) + \Phi_\theta, \qquad (8)$$

where $\Phi_\theta = k_{p\theta}\phi_1 + k_{d\theta}\phi_2 + k_{i\theta}\phi_3$.

In the same way, we have the control input for the cart position

$$u_p = k_{pp}e_p(t) + k_{ip}\int e_p(t)dt + k_{dp}\dot{e}_p(t) + \Phi_p, \qquad (9)$$

where $\Phi_p = k_{ip}\phi_4 + k_{dp}\phi_{x5} + k_{pp}\phi_6$.

If the system dynamic equation is represented as $f(\theta,\dot{\theta},\ddot{\theta},p,\dot{p},\ddot{p})$, then combining the system dynamic equation with (8) and (9) yields

$$K_P e + K_I\int edt + K_D\dot{e} = f(\theta,\dot{\theta},\ddot{\theta},p,\dot{p},\ddot{p}) - \Phi, \qquad (10)$$

where $\Phi = \Phi_\theta + \Phi_p$, $K_P = [k_{p\theta}, k_{pp}]$, $K_I = [k_{p\theta}, k_{pp}]$, $K_D = [k_{d\theta}, k_{dp}]$, and $e = [e_\theta, e_p]^T$.

To learn the inverse dynamic of the system, we set the training signal as

$$v = K_P e + K_I\int edt + K_D\dot{e}. \qquad (11)$$

When the error converges, that is, when the training signal v converges to zero, the neural network output becomes $\Phi \cong f(\theta,\dot{\theta},\ddot{\theta},p,\dot{p},\ddot{p})$ so the inverse dynamic control can be accomplished.

Next is to develop on-line learning algorithm, the back-propagation algorithm for the neural controller. Define the objective function to be minimized as

$$E = \frac{1}{2}v^2. \qquad (12)$$

Differentiating (12) with the weight vector w yields

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial v}\frac{\partial v}{\partial w} = v\frac{\partial v}{\partial w} = -v\frac{\partial \Phi}{\partial w}, \qquad (13)$$

where

$$\frac{\partial \Phi}{\partial w} = k_{P\theta}\frac{\partial \phi_1}{\partial w} + k_{D\theta}\frac{\partial \phi_2}{\partial w} + k_{I\theta}\frac{\partial \phi_3}{\partial w} + k_{ip}\frac{\partial \phi_4}{\partial w} + k_{dp}\frac{\partial \phi_5}{\partial w} + k_{pp}\frac{\partial \phi_6}{\partial w} \qquad (14)$$

In details, for each output, the weight adjustment Δw_{jk} we have

$$\Delta w_{jk} = \eta\delta_k O_j, \qquad (15)$$

where η is the learning rate, O_j is the jth output of the hidden layer, and δ_k is

$$\delta_k = -\frac{\partial E}{\partial S_k} = -\frac{\partial E}{\partial v}\frac{\partial v}{\partial S_k} = -v\frac{\partial v}{\partial \phi_k}\frac{\partial \phi_k}{\partial S_k}, \qquad (16)$$

where S_k is the kth summation of the output layer and ϕ_k is the kth output of the output layer. The gradient $\frac{\partial v}{\partial \phi_k}$ can be obtained from equation (13) as

$$\frac{\partial v}{\partial \phi_1} = k_{p\theta}, \frac{\partial v}{\partial \phi_2} = k_{d\theta}, \frac{\partial v}{\partial \phi_3} = k_{i\theta},$$
$$\frac{\partial v}{\partial \phi_4} = k_{ip}, \frac{\partial v}{\partial \phi_5} = k_{dp}, \frac{\partial v}{\partial \phi_6} = k_{pp}, \quad (17)$$

The weights are updated as

$$\Delta w(t) = \eta \frac{\partial \Phi}{\partial w} v + \alpha \Delta w(t-1), \quad (18)$$
$$w(t+1) = w(t) + \Delta w(t), \quad (19)$$

where α is the momentum constant for helping the faster convergence of the error.

5. EXPERIMENTS

5.1 Experimental Setups

Fig. 4 shows the experimental setup for regulating the inverted pendulum system. Initially the pendulum angle is set to zero position by the encoder sensor.

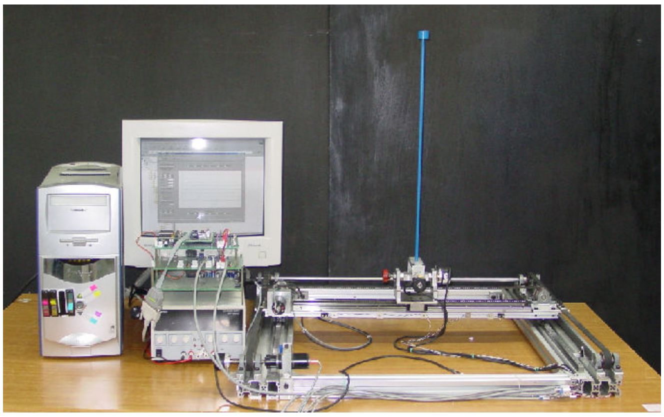


Fig. 4 Experimental setups

Fig. 5 shows the corresponding control hardware for the pendulum which consists of a DSP controller and a motor driver. The motor drivers receive PWM signals from remotely located controllers through a serial communication and change it to currents. All calculations are done in the DSP controller shown in Fig. 5.

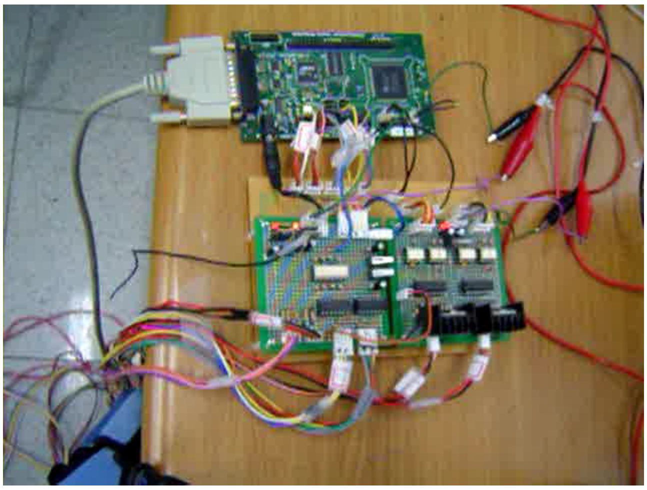


Fig. 5. Neural network control hardware

5.2 Circular Trajectory Tracking Task

The experiment is to test the desired trajectory tracking control of the pendulum system. The pendulum system is commanded to track desired circular trajectories while balancing the pendulum. The radius of the circle is 0.2m. Fig. 6 shows the tracking result. The pendulum tracks the trajectory well while maintaining the balance. The corresponding angle tracking error is within ±0.02rad as shown in Fig. 7. We see a bit larger angle error in y axis since the y axis carries the x axis. The relative position tracking error is within 3㎝ as shown in Fig. 8.

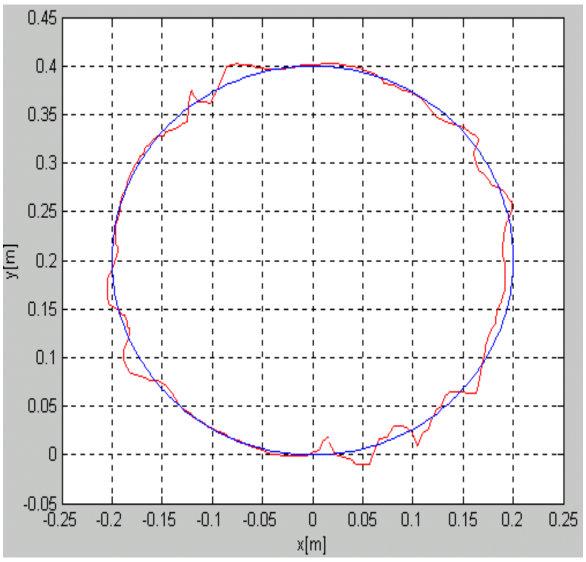
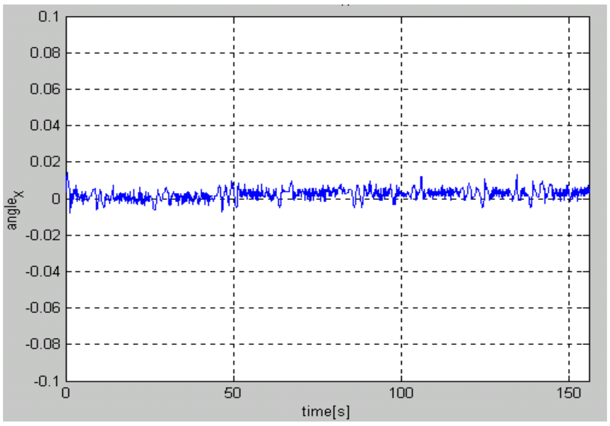
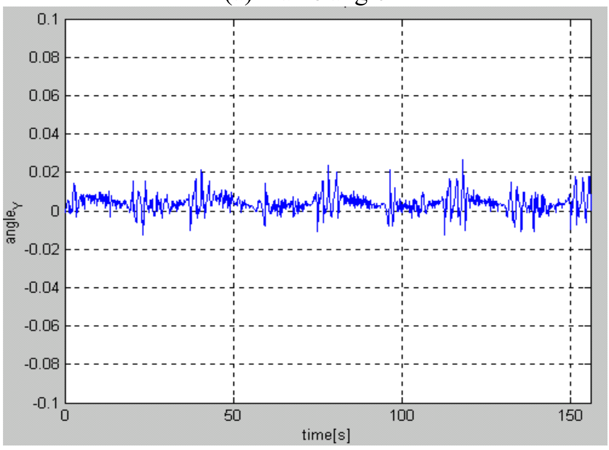


Fig. 6 Circle trajectory tracking result

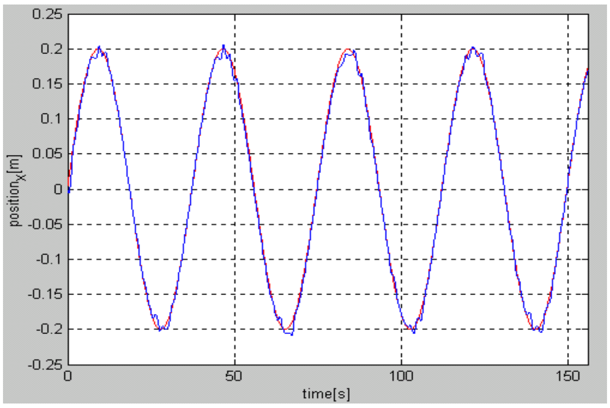


(a) x axis angle

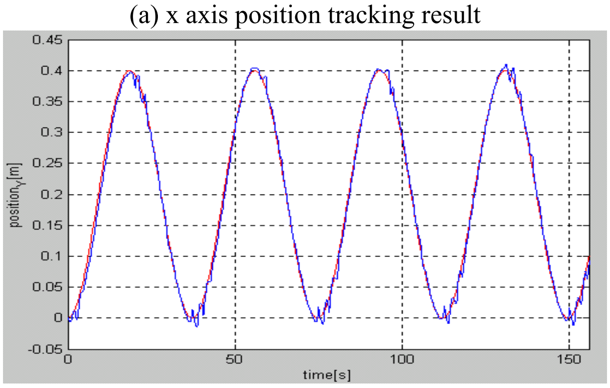


(b) y axis angle

Fig. 7 Angle balancing of inverted pendulum system



(a) x axis position tracking result



(b) y axis position tracking result

Fig. 8 Position tracking of inverted pendulum system

6. CONCLUSION

The successful hardware implementation of the neural network controller has been presented. A cost effective DSP controller has been designed to calculate the on-line learning algorithm of two neural networks. Although there exist coupling effects between two axes, neural network was able to decouple and compensate for uncertainties. The neural controller successfully regulates the position of the cart while balancing the pendulum which is quite difficult since the system is coupled and nonlinear.

ACKNOWLEDGEMENT

This research was financially in part supported by the Ministry of Commerce, Industry, and Energy (MOICE) and Korea Industrial Technology Foundation (KOTEF) through the Human Resource Training Project for Regional Innovation.

REFERENCES

Hung, T. H., Yeh, M. F, and Lu, H. C.(1997), “A PI-like fuzzy controller implementation for the inverted pendulum system,” *Proc. of IEEE Conference on Intelligent Processing Systems*, pp. 218-222

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

Kim, S. S. and Jung , S. (2004), “Hardware implementation of a real time neural network controller with a DSP and an FPGA board”, *IEEE ICRA*, pp. 4639-4644

Spong, M. W(1995). “The swing up control problem for the acrobat”, *IEEE Control Systems Magazine*, 15, pp. 72-79

White, W. and Fales, R.(1999) “Control of double inverted pendulum with hydraulic actuation : a case study”, *Proc. Of the American Control Conference*, pp.495-499,

Jung, S. and Cho, H. T.(2004) "Decentralized neural network reference compensation technique for PD controlled two degrees of freedom inverted pendulum,” *International Journal of Control, Automations, and Systems,* **vol. 2, no. 1**, pp. 92-99

Fer H. and Enns, D.(1996), “An application of dynamic inversion to stabilization of a triple inverted pendulum on a cart”, *IEEE Conf. on Control Applications*, pp. 708-714

Shen, J., Samyal, A. K., Chaturvedi, N., Bernstein, D. and McClamroch, H.(2004) “Dynamics and control of a 3 D pendulum”, *IEEE Conf. on Decision and Control*