Control Experiment of a Wheel-Driven Mobile Inverted Pendulum Using Neural Network

Seul Jung, *Member, IEEE*, and Sung Su Kim

***Abstract*—The mobile inverted pendulum is developed and tested for an intelligent control experiment of control engineers. Intelligent control algorithms are tested for the control experiment of a low cost mobile inverted pendulum system. Online learning and control using neural network of a wheel-driven mobile inverted pendulum system is presented. Neural network learning algorithm is embedded on a digital signal processing board along with primary proportional–integral–differential controllers to achieve real time control. Without knowing dynamics of the system, uncertainties in system dynamics are compensated by neural network in an online fashion. Digital filters are designed for a gyro sensor to compensate for a phase lag. Experimental studies of balancing the pendulum and tracking the desired trajectory of the cart for one dimensional motion are conducted. Results show the robustness of the proposed controller even when outer impacts as disturbance are present.**

***Index Terms*—Mobile pendulum, neural network control.**

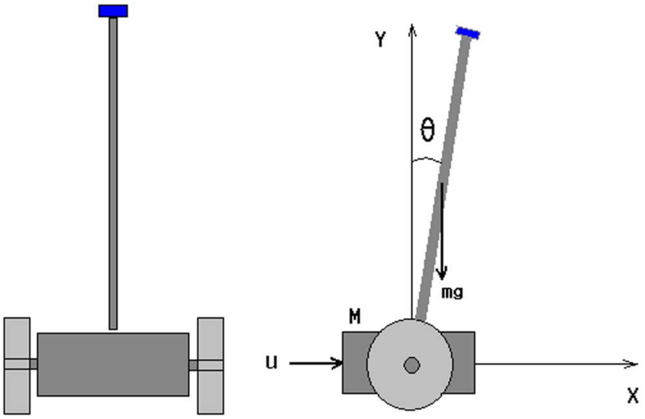

Fig. 1. Mobile pendulum structure.

I. Introduction

THE INVERTED pendulum control system has been considered as a prototype example of nonlinear system control applications. The inverted pendulum system has a single-input–multiple-output (SIMO) structure where one single input force has to control both the angle of the pendulum and the position of the cart at the same time. Numerous successful results by applying different control algorithms have been reported in the literature. Currently, the inverted pendulum system has been used for the control education purpose because various different control algorithms such as from conventional theoretical control algorithms to intelligent control algorithms can be applied and tested.

In most of inverted pendulum system control applications, the pendulum is a 1-D type which moves back and forth on the straight line. The pendulum is attached to the cart that is constrained to move on the sliding guide rail so that control of the cart is much easier by eliminating any uncertainty from cart wheels on the ground. In that framework, intelligent control methods including fuzzy logic and neural network method have been applied to prove their control effectiveness [1], [2]. Specially, for the neural network control area, research activities have been conducted with the inverted pendulum system to show its nonlinear control capability through online learning process.

Recently, numerous examples of extensions of the 1-D pendulum system have been presented. Control of two-degrees-of-freedom (DOF) inverted pendulum or a spherical pendulum moving on the x–y plane has been proposed and successfully demonstrated [3]. Further extensions of simple pendulum models include an acrobat, a pendubot, and the Furuta pendulum [4]–[10]. Increasing interest in inverted pendulum systems extends the category to a more interesting and challenging 3-D inverted pendulum problem [11].

As an extension of the same concept, a challenging problem is to control a mobile inverted pendulum system whose cart is no longer constrained to the guide rail, but moves in its terrain while balancing the pendulum as shown in Fig. 1. The 3-DOF Joe, a mobile inverted pendulum, has been developed and its successful movements have been demonstrated [12]. The dynamic equations of the JOE have been analyzed and decoupled state-space controllers were designed to keep the system in equilibrium. In [13], the mobile pendulum system is controlled by a partial feedback linearization. Furthermore, the concept of the mobile inverted pendulum system has been extended to the transportation vehicle as a human carrier [14]. This mobile pendulum carrier system has become famous after the wonderful demonstration of Segway that is built to carry a human. Now the commercialized Segway is used for carrying a robonaut in NASA projects [15].

In this brief, a mobile inverted pendulum system is developed and controlled for control applications. Since the mobile pendulum system is a non-holonomic system whose dynamics is very complicated, model-based control approaches lose their merits in this case. When without knowing the dynamic model of the system, an intelligent control approach is suitable to apply. Research motivation is to control the mobile inverted pendulum system for education of control engineers under the subject of intelligent control systems. The neural network

Manuscript received August 31, 2006; revised November 13, 2006. This work was supported in part by the KOSEF under Grant 05-2003-000-10389-0.

S. Jung is with the Chungnam National University, Daejeon, 305-764 Korea (e-mail: jungs@ cnu.ac.kr).

S. S. Kim was with the Chungnam National University, Daejeon, 305-764 Korea. He is now with the Agency for Defense Development, Daejeon, 305-152 Korea (e-mail: gubuki780616@hanmail.net).

Color versions of one or more of the figures in this brief are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCST.2007.903396

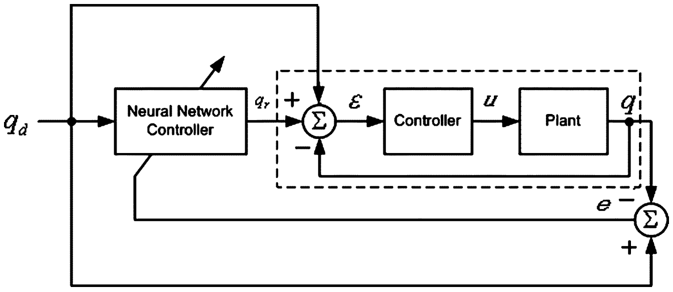

Fig. 2. Reference compensation technique control scheme.

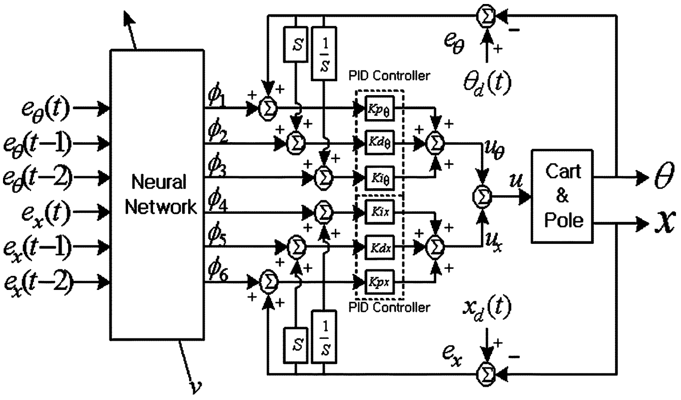

Fig. 3. Neural network control block diagram.

control technique has been applied and tested for control experiments of the mobile inverted pendulum. One of the merits of using neural network as an auxiliary controller is when complicated dynamic model of the system is not available. Neural network controllers control the system along with primary proportional–integral-differential (PID) controllers by compensating for uncertainties in dynamics in online adaptive fashion. To achieve real time control, the massive neural network learning algorithm with PID controllers has been embedded on a digital signal processing (DSP) board. Experimental results show that the performance of balancing the pendulum and tracking position of the mobile pendulum is quite well.

II. Overall System

The wheel-driven mobile pendulum system is shown in Fig. 1. The pendulum is mounted on the mobile platform whose structure is a wheel-driven mobile robot. The mobile pendulum can change the direction by differing the wheel speed driven by dc motors. Control objective is to regulate the position tracking of the cart while the pendulum is balancing.

A gyro sensor is used to detect the angle of the pendulum and encoders are used to count rotation of wheels. Dynamic behaviors of the mobile pendulum are same as the inverted pendulum for a one dimensional case, but more uncertainties are present in the mobile pendulum system since rolling on the floor causes inconsistent disturbance to the system due to irregular surface of the floor.

III. Neural Network Control

Neural network has been known as a nonlinear adaptive controller and successful control applications can be found in the literature. From among several neural control structures, in this brief, the reference compensation technique (RCT) control structure is used [16]. The RCT algorithm is similar to the well-known feedback error learning scheme [17]. The block diagram of the RCT scheme is shown in Fig. 2. The concept of the RCT is to compensate the system controlled by predetermined controllers by closing another outer loop as shown in Fig. 2. Neural network output signal q_r 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 mobile pendulum system is shown in Fig. 3. Neural network outputs are added to tracking errors to form PID controller outputs. A control input u_θ for the pendulum angle and a control input u_x for the cart position are summed together to generate an input force u to the system.

The pendulum angle error is defined as

$$e_\theta - \theta_d - \theta \tag{1}$$

where θ_d is the desired angle of the pendulum and θ is the actual angle of the pendulum.

Then a PID control input for angle control is given by

$$\begin{aligned} u_\theta = k_{p\theta} e_\theta(t) + k_{i\theta} \int e_\theta(t)dt + k_{d\theta}\dot{e}_\theta(t) + k_{p\theta}\phi_1 \\ + k_{d\theta}\phi_2 + k_{i\theta}\phi_3 \end{aligned} \tag{2}$$

where $k_{p\theta}$, $k_{d\theta}$, and $k_{i\theta}$ are PID gains for pendulum control and ϕ_1, ϕ_2, and ϕ_3 are neural network outputs.

The mobile pendulum position error is defined by

$$e_x = x_d - x \tag{3}$$

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

The PID control input for position control is

$$\begin{aligned} u_x = k_{px} e_x(t) + k_{ix} \int e_x(t)dt + k_{dx}\dot{e}_x(t) + k_{ix}\phi_4 \\ + k_{dx}\phi_5 + k_{px}\phi_6 \end{aligned} \tag{4}$$

where k_{ix}, k_{dx}, k_{px} are PID gains for cart control and ϕ_4, ϕ_5, ϕ_6 are neural network outputs.

The overall control input is the sum of two PID controller outputs, u_θ in (2) and u_x in (4)

$$u = u_x + u_\theta. \tag{5}$$

We expect the coupling action between the angle control and the position control which can be managed by the neural network controller. Next, we show how to achieve online learning and control.

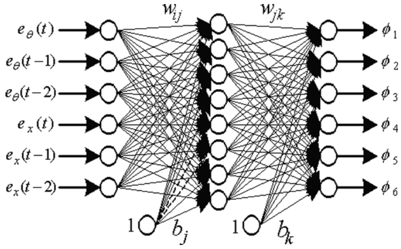

Fig. 4. Neural network structure.

IV. Neural Network Learning

Here, a two layered feed-forward neural network structure consists of w_{ij}, the weight between input and hidden layer, w_{jk}, the weight between hidden and output layer, b_j, the bias weight for hidden layer, and b_k the bias weight for output layer. For a neural network structure, six inputs, nine hidden units, and six output units are used as shown in Fig. 4. 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 (2) and (4).

Nonlinear function for a neuron is the hyperbolic tangent function whose bound is ±1

$$f_{th}(x) = \frac{1-\exp(-x)}{1+\exp(-x)}. \tag{6}$$

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

$$u_\theta = k_{P\theta} e_\theta(t) + k_{i\theta}\int e_\theta(t)dt + k_{d\theta}\dot{e}_\theta(t) + \Phi_\theta \tag{7}$$

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

$$u_x = k_{px} e_x(t) + k_{ix}\int e_x(t)dt + k_{dx}\dot{e}_x(t) + \Phi_x \tag{8}$$

where $\Phi_x = k_{ix}\phi_4 + k_{dx}\phi_5 + k_{px}\phi_6$.

If the system dynamic equation is represented as $f(\theta,\dot{\theta},\ddot{\theta},x,\dot{x},\ddot{x})$, then combining the system dynamic equation with (7) and (8) yields

$$K_P e + K_I\int edt + K_D\dot{e} = f(\theta,\dot{\theta},\ddot{\theta},x,\dot{x},\ddot{x}) - \Phi \tag{9}$$

where $\Phi = \Phi_\theta + \Phi_x$, $K_P = [k_{p\theta}, k_{px}]$, $K_I = [k_{p\theta}, k_{px}]$, $K_D = [k_{D\theta}, k_{Dx}]$, and $e = [e_\theta, e_x]^T$.

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

$$v - K_P e + K_I\int edt + K_D\dot{e}. \tag{10}$$

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

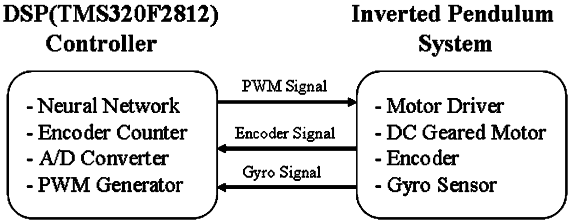

Fig. 5. Neural network control block diagram.

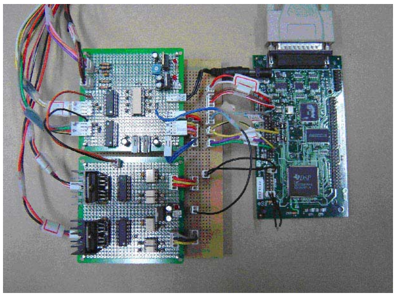

Fig. 6. Neural network controller hardware.

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

$$E = \frac{1}{2}v^2. \tag{11}$$

Differentiating (11) with the weight vector w yields

$$\frac{\partial E}{\partial w} = \frac{\partial E}{\partial v}\frac{\partial v}{\partial w} = v\frac{\partial v}{\partial w} = -v\frac{\partial \Phi}{\partial w} \tag{12}$$

where

$$\frac{\partial \Phi}{\partial w} = k_{P\theta}\frac{\partial \phi_1}{\partial w} + k_{D\theta}\frac{\partial \phi_2}{\partial w} + k_{I\theta}\frac{\partial \phi_3}{\partial w} + k_{lx}\frac{\partial \phi_4}{\partial w} + k_{Dx}\frac{\partial \phi_5}{\partial w} + k_{Px}\frac{\partial \phi_6}{\partial w}. \tag{13}$$

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

$$\Delta w_{jk} = \eta\delta_k O_j \tag{14}$$

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

$$\delta_k = -\frac{\partial E}{\partial S_k} = -\frac{\partial E}{\partial v}\frac{\partial v}{\partial S_k} = -v\frac{\partial v}{\partial \phi_k}\frac{\partial \phi_k}{\partial S_k} \tag{15}$$

where S_k is the kth summation of the output layer and ϕ_k is the kth output of the output layer. The gradient $\partial v/\partial \phi k$ can be obtained from (13) as

$$\frac{\partial v}{\partial \phi_1} = k_{p\theta x}, \quad \frac{\partial v}{\partial \phi_2} + k_{d\theta x}, \quad \frac{\partial v}{\partial \phi_3} k_{i\theta x}, \quad \frac{\partial v}{\partial \phi_4} = k_{ix}, \quad \frac{\partial v}{\partial \phi_5} = k_{dx}, \quad \frac{\partial v}{\partial \phi_6} = k_{px}. \tag{16}$$

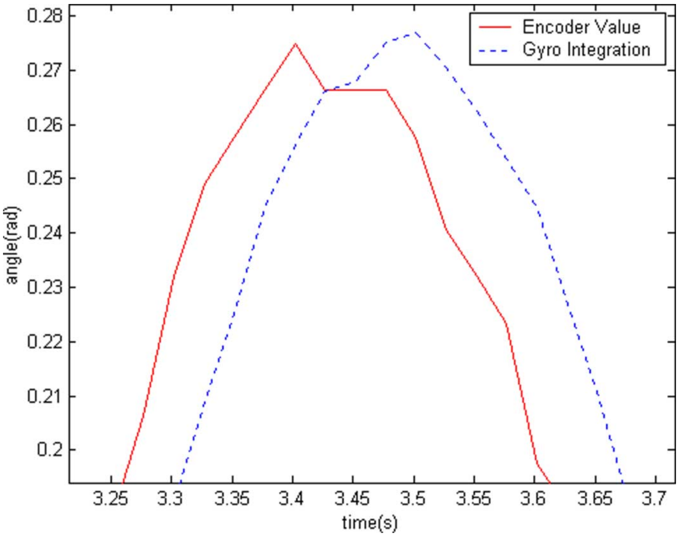

Fig. 7. Output signals from Gyro and encoder sensors.

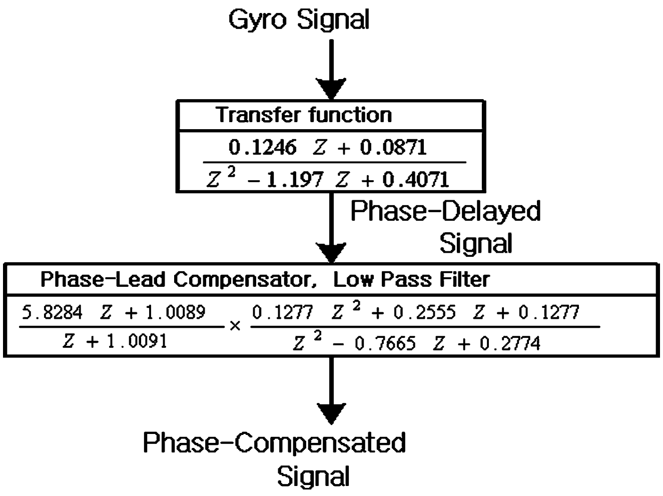
Gyro Signal
Transfer function
$\frac{0.1246 Z + 0.0871}{Z^2 - 1.197 Z + 0.4071}$
Phase-Delayed Signal
Phase-Lead Compensator, Low Pass Filter
$\frac{5.8284 Z + 1.0089}{Z + 1.0091} \times \frac{0.1277 Z^2 + 0.2555 Z + 0.1277}{Z^2 - 0.7665 Z + 0.2774}$
Phase-Compensated Signal

Fig. 8. Phase compensation process of a gyro sensor.

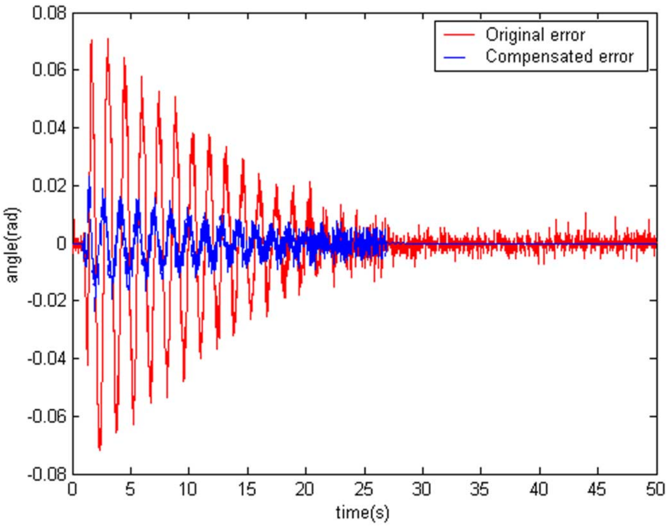

Fig. 9. Comparison between uncompensated and compensated sensor output.

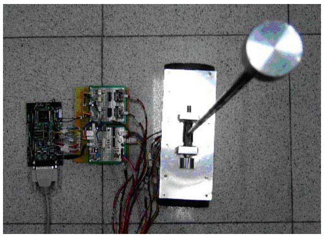

Fig. 10. Top view of the mobile pendulum system.

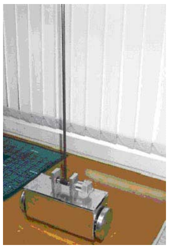

Fig. 11. Front view of the mobile pendulum system.

The weights are updated as

$$\Delta w(t) = \eta \frac{\partial \Phi}{\partial w} v + \alpha \Delta w(t-1) \tag{17}$$

$$w(t+1) = w(t) + \Delta w(t) \tag{18}$$

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

V. Hardware Implementation

A. *Neural Network Controllers*

Neural network controller hardware uses a TI TMS320F2812DSP DSP board on which neural network learning algorithm is embedded.

Fig. 5 shows the block diagram between a mobile pendulum system and a controller board. The controller board generates PWM signals to motor drivers and receives signals from a gyro and encoder sensors. Received signals are used to calculate tracking errors and those errors are summed with neural network outputs. Then, summed signals are used for PID calculation to drive control torques.

Fig. 6 shows the corresponding real hardware implementation. It is composed of a DSP board, encoder counters, and motor drivers.

B. *Filter Design for Gyro Sensor Compensation*

To detect the movement of the pendulum angle, a gyro sensor is used. However, the gyro sensor has a defect of the drift problem with respect to time. As the time goes, signals from the gyro sensor deviate from real data due to the temperature variation. So it requires refreshing process periodically. To find out the characteristic of the gyro sensor, we have implemented the experimental apparatus to test gyro sensors. The testing apparatus consists of a dc motor and an encoder to test and compares movements of gyro and encoder measurements

TABLE I
PID CONTROLLER GAINS

Angle	$k_{p\theta}$	-1.1828
	$k_{i\theta}$	-0.0004
	$k_{d\theta}$	-0.2151
Position	k_{px}	0.6452
	k_{ix}	0.0003
	k_{dx}	0.4086
Learning rate(η)		0.04
Momentum (α)		0.2

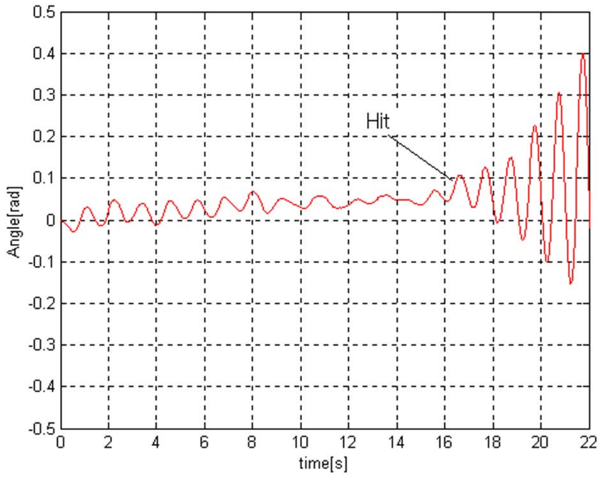

Fig. 12. Angle of the mobile inverted pendulum system by PID controllers.

for a certain period. Data from an encoder sensor is considered as more accurate.

Fig. 7 shows the comparison plots of two sensor output signals, a gyro and an encoder. We can clearly see that there is a time delay which lags about 0.05 s. This delay becomes critical in controlling the system as the time goes on. So a phase lead compensator is required to compensate for the phase lag.

In addition, to get rid of noise, the second-order low-pass IIR filter is designed. Filter parameters are designed to compensate for the phase delay. Fig. 8 shows the flow chart of compensating for the phase lag. The corresponding resultant plot is shown in Fig. 9. We see that the error between two sensors is well compensated.

VI. EXPERIMENT

A. Experimental Setups

The actual experimental setup is shown in Figs. 10 and 11. The mobile pendulum system is setup on the floor. PID controller gains are selected for the better performance experimentally and listed in Table I. Neural network parameters are also selected to have the better performance by trial and error process.

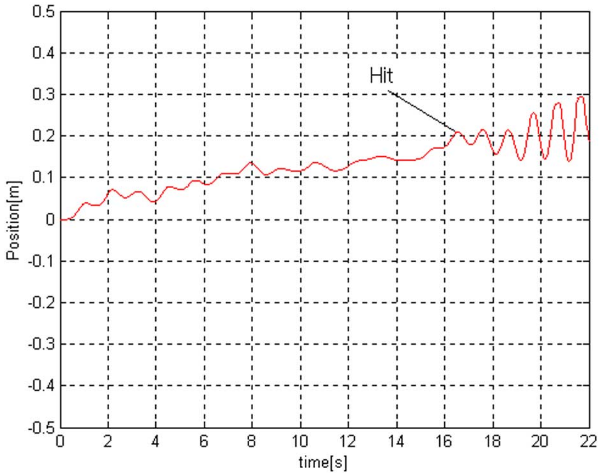

Fig. 13. Position of the mobile pendulum system by PID controllers.

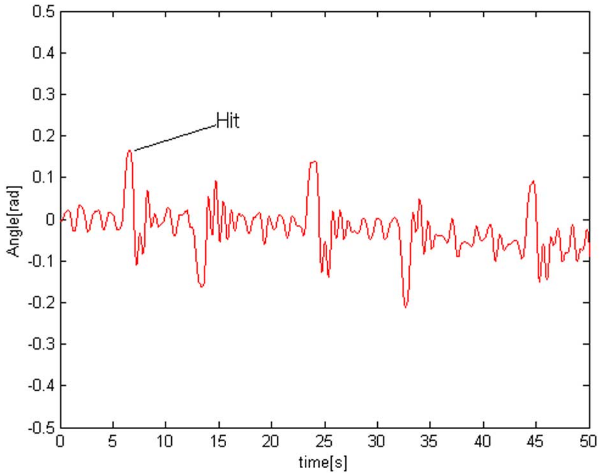

Fig. 14. Angle of the mobile pendulum system by the neural network controller.

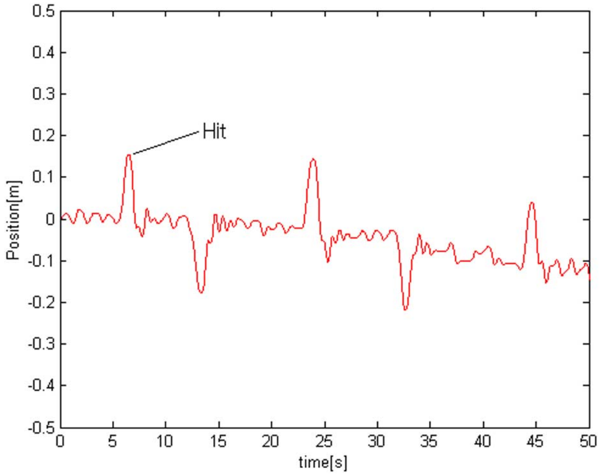

Fig. 15. Position of the mobile pendulum system by the neural network controller.

B. Experimental Results

1) PID Control: Fig. 12 shows the angle of the pendulum controlled by PID controllers only. As expected, PID controllers can maintain the balance of the pendulum, but fails to control the cart position. Fig. 13 shows that the cart keeps moving toward one direction. The performance of PID controllers is worse

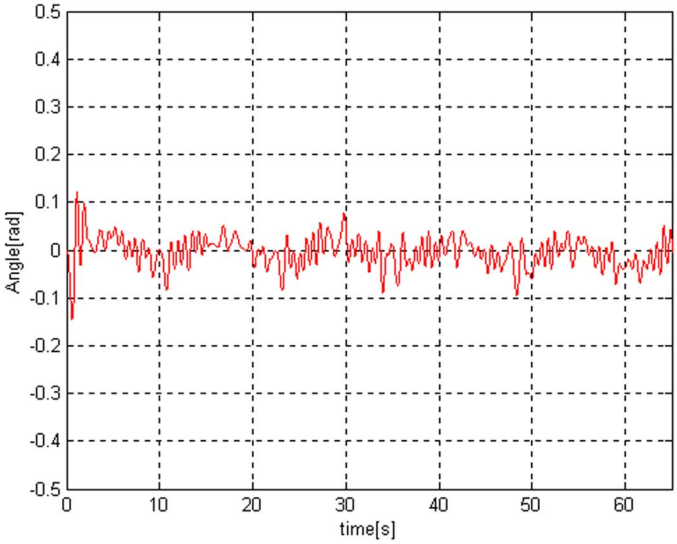

Fig. 16. Angle of mobile inverted pendulum, $T = 4\pi$ s.

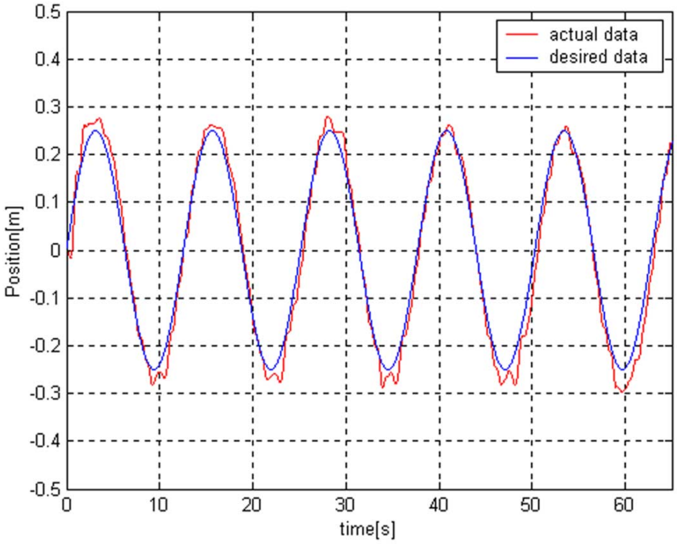

Fig. 17. Position of mobile inverted pendulum, $T = 4\pi$ s.

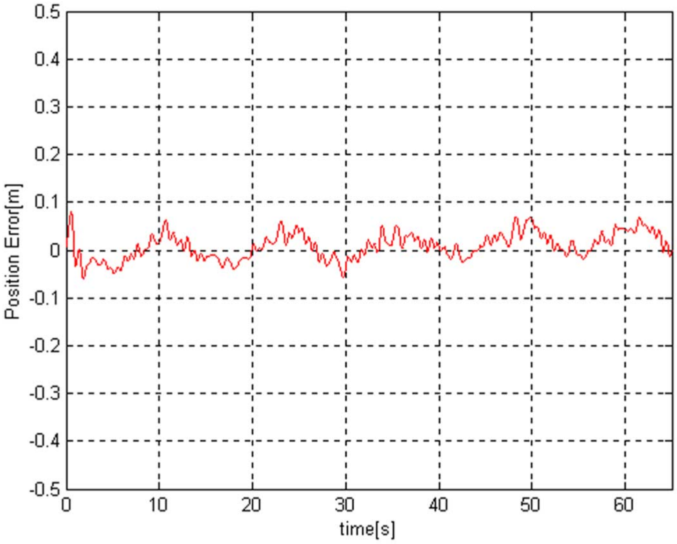

Fig. 18. Position error of mobile inverted pendulum, $T = 4\pi$ s.

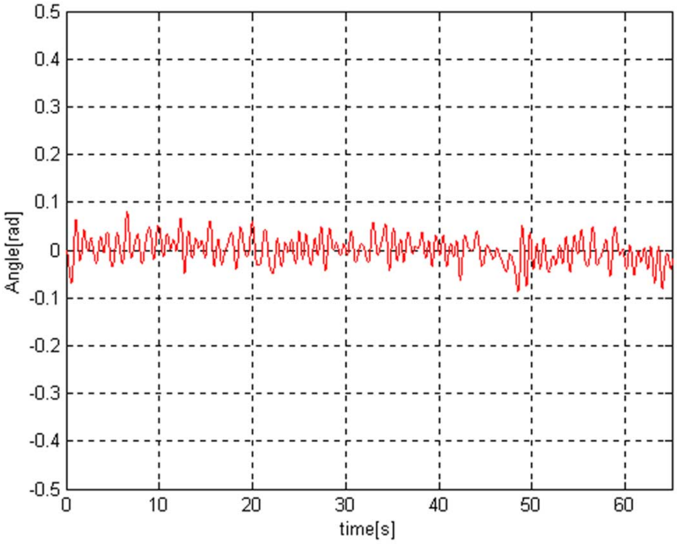

Fig. 19. Angle of mobile inverted pendulum, $T = 8\pi$ s.

when outer disturbance is present. We see that the pendulum angle starts to oscillate and fails to maintain the balance after a single hit.

2) Neural Network Control:

- *Balancing control task.* The next experiment is to use neural network to control the mobile pendulum system. The pendulum is disturbed intentionally while the mobile pendulum system is balancing. Figs. 14 and 15 show the balancing and movement of the mobile pendulum. Although there are several external disturbance impacts to the pendulum, it is quite robust to maintain the balance well. After each impact, the mobile pendulum system successfully maintains a desired position. However, after several impacts, a position error occurs due to the slippage of wheels caused by physical impacts. In comparison with results of the PID controller, while the tracking result by the PID controller goes unstable when impact occurs, the neural controller is able to maintain the stability as shown in Figs. 14 and 15.
- *Trajectory tracking control task.* The third experiment is to test desired trajectory tracking control of the mobile pendulum system. The mobile pendulum system is required to track desired sinusoidal trajectories while balancing the pendulum. Two different periods of the trajectory are tested. One trajectory is considered as two times faster than the other one, the period $T = 4\pi$ s and 8π s. And the magnitude of the sinusoidal trajectory is 0.25 m. Figs. 16–18 show the results when $T = 4\pi$ s, which is considered to be faster. The mobile pendulum tracks the trajectory well while maintaining the balance. The angle tracking error is within ±0.1 rad, and the position tracking error is within 5 cm.

Slowing down the speed by the half, the tracking error is somewhat minimized. Figs. 19–21 show the resultant plots when $T = 8\pi$ s. The angle error is within ±0.05 rad, and position tracking error is within 2 cm.

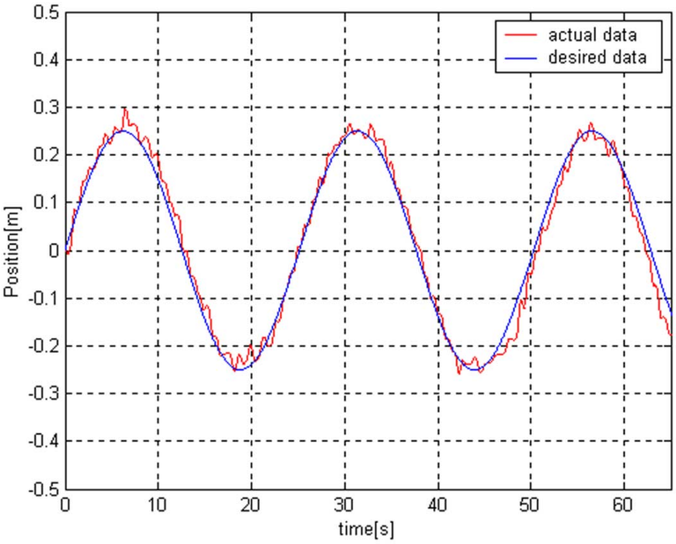

Fig. 20. Position of mobile inverted pendulum, $T = 8\pi$ s.

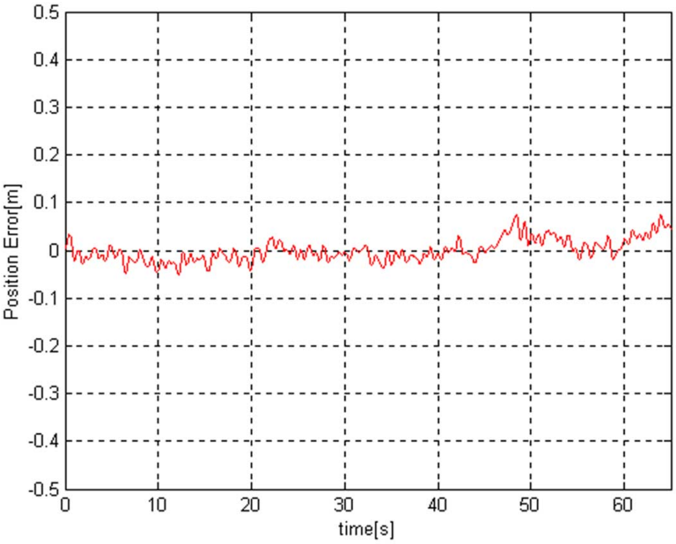

Fig. 21. Position error of mobile inverted pendulum, $T = 8\pi$ s.

VII. Conclusion

In this brief, control and implementation of a wheeled drive mobile robot-based inverted pendulum system are presented. Relying on a gyro sensor and encoders, the mobile pendulum was able to maintain balance and to track desired trajectories. The gyro sensor delay problem was compensated by digital filters. The neural network controller combined with PID controllers was implemented on DSP hardware.

The neural controller successfully controls both the balance of the pendulum and the position of the mobile robot without knowing any dynamics of the system. Even when outer disturbances are present, the neural network controller shows the robustness to maintain the balance while PID controllers failed.

Acknowledgment

The authors would like to thank the reviewers for their valuable comments and suggestions that have improved the quality of this brief.

References

[1] T. H. Hung, M. F. Yeh, and H. C. Lu, “A PI-like fuzzy controller implementation for the inverted pendulum system,” in *Proc. IEEE Conf. Intell. Process. Syst.*, 1997, pp. 218–222.

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

[3] S. S. Kim and S. Jung, “Hardware implementation of a real time neural network controller with a DSP and an FPGA board,” in *Proc. IEEE ICRA*, 2004, pp. 4639–4644.

[4] F. Cheng, G. Zhong, Y. Li, and Z. Xu, “Fuzzy control of a double inverted pendulum,” *Fuzzy Sets Syst.*, vol. 79, pp. 315–321, 1996.

[5] M. W. Spong, P. Corke, and R. Lozano, “Nonlinear control of the inertia wheel pendulum,” *Automatica*, vol. 37, pp. 1845–1851, 2001.

[6] M. W. Spong, “The swing up control problem for the acrobat,” *IEEE Control Syst. Mag.*, vol. 15, no. 2, pp. 72–79, Feb. 1995.

[7] W. White and R. Fales, “Control of double inverted pendulum with hydraulic actuation: A case study,” in *Proc. Amer. Control Conf.*, 1999, pp. 495–499.

[8] S. Jung and H. T. Cho, “Decentralized neural network reference compensation technique for PD controlled two degrees of freedom inverted pendulum,” *Int. J. Control, Autom., Syst.*, vol. 2, no. 1, pp. 92–99, 2004.

[9] H. Fer and D. Enns, “An application of dynamic inversion to stabilization of a triple inverted pendulum on a cart,” in *Proc. IEEE Conf. Control Appl.*, 1996, pp. 708–714.

[10] H. T. Cho and S. Jung, “Neural network position tracking control of an inverted pendulum by an x-y table robot,” in *Proc. IEEE IROS*, 2003, pp. 1210–1215.

[11] J. Shen, A. K. Samyal, N. Chaturvedi, D. Bernstein, and H. McClamroch, “Dynamics and control of a 3 D pendulum,” in *Proc. IEEE Conf. Dec. Control*, 2004, pp. 323–328.

[12] F. Grasser, A. Darrigo, S. Colombi, and A. Rufer, “JOE: A mobile, inverted pendulum,” *IEEE Trans. Ind. Electron.*, vol. 49, no. 1, pp. 107–114, Feb. 2002.

[13] K. Pathak, J. Franch, and S. Agrawal, “Velocity and position control of a wheeled inverted pendulum by partial feedback linearization,” *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 505–513, Jun. 2005.

[14] H. Tirmant, M. Baloh, L. Vermeiren, T. M. Guerra, and M. Parent, “B2, an alternative two wheeled vehicle for an automated urban transportation system,” in *Proc. IEEE Intell. Veh. Symp.*, 2002, pp. 594–603.

[15] R. O. Ambrose, R. T. Savely, S. M. Goza, P. Strawser, M. A. Diftler, I. Spain, and N. Radford, “Mobile manipulation using NASA’s robonaut,” in *Proc. IEEE ICRA*, 2004, pp. 2104–2109.

[16] S. Jung and T. C. Hsia, “Neural network inverse control techniques for PD controlled robot manipulator,” *ROBOTICA*, vol. 19, no. 3, pp. 305–314, 2002.

[17] M. Miyamoto, M. Kawato, T. Setoyama, and R. Suzuki, “Feedback error learning,” *IEEE Trans. Neural Netw.*, vol. 1, pp. 251–265, 1988.