Proceedings of the 2007 IEEE Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL 2007)

DHP Adaptive Critic Motion Control of Autonomous Wheeled Mobile Robot

Wei-Song Lin[1], *Member*, *IEEE* and Ping-Chieh Yang[2]
Department and Institute of Electrical Engineering, National Taiwan University, Taiwan

***Abstract*—Autonomous drive of wheeled mobile robot (WMR) needs implementing velocity and path tracking control subject to complex dynamical constraints. Conventionally, this control design is obtained by analysis and synthesis of the WMR system. This paper presents the dual heuristic programming (DHP) adaptive critic design of the motion control system that enables WMR to achieve the control purpose simply by learning through trial. The design consists of an adaptive critic velocity neuro-control loop and a posture neuro-control loop. The neural weights in the velocity neuro-controller (VNC) are corrected with the DHP adaptive critic method. The designer simply expresses the control objective with a utility function. The VNC learns by sequential optimization to satisfy the control objective. The posture neuro-controller (PNC) approximates the inverse velocity model of WMR so as to map planned positions to desired velocities. Supervised drive of WMR in variant velocities supplies training samples for the PNC and VNC to setup the neural weights. In autonomous drive, the learning mechanism keeps improving the PNC and VNC. The design is evaluated on an experimental WMR. The excellent results make it certain that the DHP adaptive critic motion control design enables WMR to develop the control ability autonomously.**

***Index Terms*—Adaptive critic design, autonomous robot, neuro-control, dual heuristic programming, reinforcement learning**

I. Introduction

Autonomous wheeled mobile robot (WMR) relies on using sensors to percept its surroundings and using motion controller to drive to the destination. In the motion control, WMR should be capable of performing trajectory tracking, path following and stabilization. Since WMR is a nonholonomic dynamic system with intrinsic non-linearity, and commonly with unmodeled disturbance and unstructured, unmodeled dynamics [1]. Therefore, unless the mass is negligible [2], the motion control should consider the dynamics of WMR [3], [4]. Conventionally, this control design relies on engineers to analyze and synthesize the WMR system [5]-[7]. But usually difficulties arise from absence of accurate WMR model. Fuzzy control may skip the model but needs domain expert to construct the fuzzy rules [8], [9]. Controllers based on neural networks or neuro-fuzzy networks may construct the control function by learning the training samples [10]-[12]. But preparing appropriate training samples usually needs an existing controller. This limits the applications of neural and neuro-fuzzy learning control to a minimum. Alternatively, the adaptive critic motion control design presented in this paper enables WMR to build the control function by learning to optimize an objective function. No domain expert to setup the control rules and no existing controller to generate the training samples are required. In our laboratory, an experimental WMR was developed and its mathematical model was formulated and identified [13]. A hierarchical fuzzy control system was implemented and shown able to conduct the motion of WMR [14]. Furthermore, the experimental WMR was equipped with a stereovision system to enable autonomous path finding and collision avoidance [15]. In this paper, we assume the stereovision system foresees nearest path and the WMR system must construct the motion control function entirely through learning by trials. Essentially, this extends the definition of autonomous robot to autonomous setup of intelligence. But presently the autonomous setup is limited to the motion control. The idea is to obtain the motion control by learning to satisfy or optimize a specified objective function. Trials, actually supervised trials for the sake of safety, supply training samples to correct the neural networks. As a result, the neural networks can learn the motion control without reference to any existing controller. The dual heuristic programming (DHP) adaptive critic method [16], [17] is invoked to design the learning system. Multilayer perceptrons, a type of feedforward neural networks, are used as the basic learning structure to construct the posture neuro-controller (PNC) and velocity neuro-controller (VNC). The PNC learns to map planned positions to suitable desired velocity. The VNC learns to conduct the WMR motion so as to track the desired velocity. Supervised and autonomous drive of WMR in variant velocities supplies training samples for the PNC and VNC to correct the neural weights. The proposed design is evaluated with the experimental WMR.

This paper is organized as follows: Section II presents the architecture of the interested autonomous WMR and illustrates the blocks in the adaptive critic motion control system. Section III formulates the adaptive critic motion control design. Section IV evaluates the control performance with the experimental WMR. Section V is the conclusions.

[1] Corresponding author, Wei-Song Lin is with the Department and Institute of Electrical Engineering, National Taiwan University; Address: No.1, Sec. 4, Roosevelt Rd., Taipei 106, Taiwan; E-mail: weisong@cc.ee.ntu.edu.tw, Fax: +886-2-23638247, Phone: +886-2-33663638

[2] Ping-Chieh Yang is with the army of Taiwan and was with the Department and Intitute of Electrical Engineering, National Taiwan University; Address: No.1, Sec. 4, Roosevelt Rd., Taipei 106, Taiwan; E-mail: r93921078@ntu.edu.tw

II. ARCHITECTURE OF THE AUTONOMOUS WMR SYSTEM

The interested WMR is a four-wheeled mechanism shown in figure 1. The front wheels are passive, whereas the rear wheels are motorized independently to give the differential rotation configuration. Notations are defined as follows: d is the displacement from the point P along the X_c axis to the center of mass; r is the radius of driving wheels; m_c is the mass of the WMR body (i.e. excluding the driving wheels and their associated rotors); m_w is the mass of a single driving wheel (i.e. taking the associated rotor into account); I_c is the moment of inertia of the body; I_w is the moment of inertia of each driving wheel about the axle; I_m is the moment of inertia of each driving wheel about a wheel diameter; v is the linear velocity; w is the angular velocity; θ is the angle of orientation, $\dot{\varphi}_l$ and $\dot{\varphi}_r$ are angular speeds of left and right wheels, respectively. An experimental system of such WMR with stereovision has been assembled in our laboratory [13]-[15]. Table 1 lists the main figures. The experimental WMR is completely autonomous because data are elaborated without any external aid, and its sensors are the encoders attached to the rear wheels and the stereo camera module with digital output.

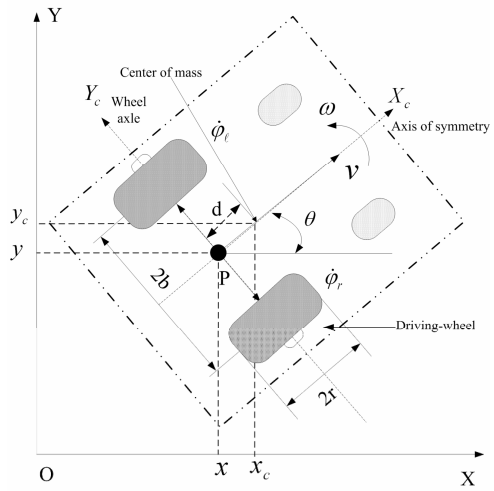

Fig. 1 A schematic top view of the experimental WMR

Table 1 Mechanical figures of the experimental WMR.

$b(m)$	$d(m)$	$r(m)$	$w_c(m)$	
0.265	0.1	0.125	0.8	
$m_c(kg)$	$m_w(kg)$	$I_c\,(kg\,m^2)$	$I_w(kg\,m^2)$	$I_m(kg\,m^2)$
110	5	1.057	0.004	0.002

Using Lagrange formalism, the dynamical model of WMR is described as [13], [18]

$$\overline{\mathbf{M}}(\mathbf{q})\dot{\mathbf{R}} + \overline{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}})\mathbf{R} + \overline{\mathbf{F}}(\dot{\mathbf{q}}) + \overline{\mathbf{G}}(\mathbf{q}) + \overline{\boldsymbol{\tau}}_d = \overline{\mathbf{B}}(\mathbf{q})\mathbf{u} \quad (1)$$

where $\mathbf{q} = [x, y, \theta, \phi_l, \phi_r]^T$ is the generalized coordinate vector to characterize WMR, $\mathbf{R} = [v, w]^T$ in which v is the linear velocity and w is the angular velocity, $\mathbf{u} = [\tau_l, \tau_r]^T$ are the input torques generated by the left and right motors. The parameter matrices in (1) are

$$\overline{\mathbf{M}}(\mathbf{q}) = \begin{bmatrix} m + \frac{2I_w}{r^2} & 0 \\ 0 & I + \frac{2b^2 I_w}{r^2} \end{bmatrix},\ \overline{\mathbf{C}}(\mathbf{q},\dot{\mathbf{q}}) = \begin{bmatrix} 0 & -\dot{\theta} m_c d \\ \dot{\theta} m_c d & 0 \end{bmatrix},\ \overline{\mathbf{B}}(\mathbf{q}) = \begin{bmatrix} \frac{2}{r} & \frac{2}{r} \\ \frac{-2b}{r} & \frac{2b}{r} \end{bmatrix}$$

where $m = m_c + 2m_w$, and $\overline{\mathbf{F}}(\dot{\mathbf{q}})$, $\overline{\mathbf{G}}(\mathbf{q})$ and $\overline{\boldsymbol{\tau}}_d$ are unknown terms corresponding to frictional, gravitational and disturbed forces, respectively. To conduct the WMR motion needs implementing velocity and trajectory tracking control. Hierarchical fuzzy control was shown a feasible approach [15], but the fuzzy rules are constructed by a domain expert. Alternatively, this paper seeks to build the motion control function entirely through learning by trials. The innovative approach is called the adaptive critic motion control system which consists of mainly a posture neuro-control loop and a velocity neuro-control loop. As shown in figure 2, the stereovision unit percepts the surroundings to find a forward path. According to the path, feedback positions and the physical limitations of WMR, the path planner produces the planed positions. The PNC learns by approximating the inverse velocity model of WMR to map the planned positions to the desired linear and angular velocities. The VNC is a DHP adaptive critic design which invokes reinforcement learning to obtain the velocity control. Learning begins with supervised drive in variant velocities to build the neural weights. In autonomous drive, the learning mechanism keeps correcting the neural weights to improve the control performance.

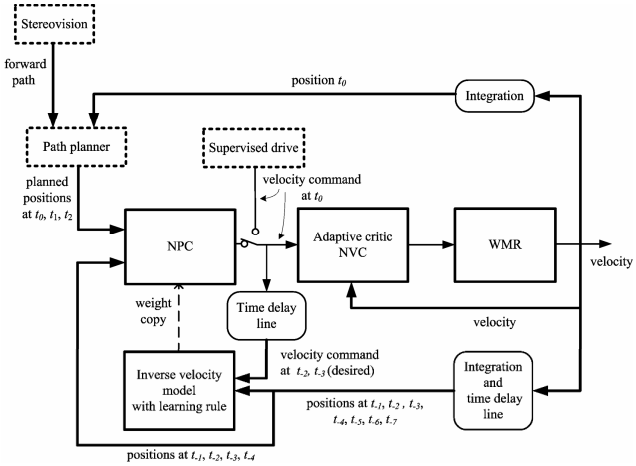

Fig. 2 Architecture of adaptive critic motion control system of WMR

III. DESIGN OF THE DHP ADAPTIVE CRITIC MOTION CONTROL SYSTEM

A. The DHP adaptive critic velocity neuro-controller (VNC)

Adaptive critic methods are usually practiced with model based learning structures such as neural or neuro-fuzzy networks. They have common roots as generalizations of dynamic programming for neural reinforcement learning approaches and have a capability of optimization over time

under conditions of noise, uncertainty, and nonlinearity [16]. Heuristic dynamic programming (HDP), dual heuristic programming (DHP), and globalized dual heuristic programming (GDHP) are the main categories of adaptive critic designs [17]. They are differentiated by the critic output. HDP uses the critic to estimate the value function in the Bellman equation of dynamic programming. In DHP, the critic approximates the derivatives of the value function to facilitate the computation in the gradient descent correcting rule. The critic in GDHP estimates both the value function and its derivatives. DHP was shown to have a superior performance to HDP and no observable improved performance by GDHP [19], [20]. Therefore, DHP is chosen for the adaptive critic motion control design of autonomous WMR.

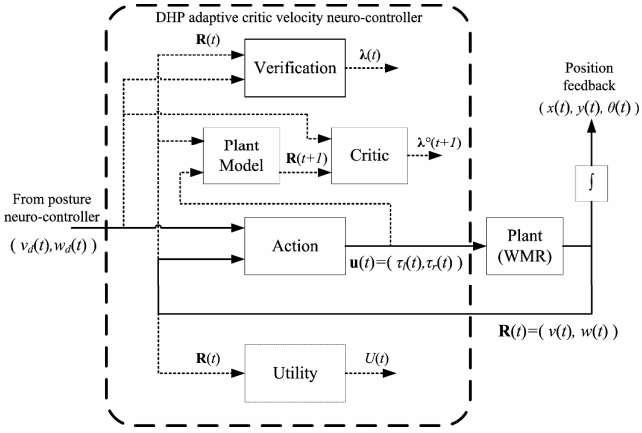

Fig. 3 Architecture of the DHP adaptive critic velocity neuro-controller

As shown in figure 3, the DHP adaptive critic VNC consists of neural networks to implement the action, critic, verification and even the plant model. In the figure, $R(t)$ represents the state variable, $u(t)$ is the control signal and $U(t)$ which depends on $R(t)$ is the primary utility function defined by the user for the specific application context. The neural weights in the action are corrected to minimize not only the present utility function alone but also the sum of all future values of $U(t)$. The verification and critic approximate the derivatives of the secondary utility function J (the value function) with respect to its state variables at present and immediate future instances. In the Bellman equation of dynamic programming, J is expressed as

$$J(t)=\sum_{k=0}^{\infty}\gamma^k U(t+k)=U(t)+\gamma J(t+1) \tag{2}$$

where γ, $0<\gamma\le 1$ is a discount factor. Accordingly, the verification and critic outputs are respectively

$$\lambda_s(t)=\frac{\partial J(t)}{\partial R_s(t)} \tag{3}$$

$$\lambda_s^{\circ}(t)=\frac{\partial J(t+1)}{\partial R_s(t+1)} \tag{4}$$

The critic critiques $\lambda(t)$ to be

$$\begin{aligned}\lambda_s^{\circ}(t)&=\frac{\partial}{\partial R_s(t)}\left(U(t)+\gamma J(t+1)\right)\\&=\frac{\partial U(t)}{\partial R_s(t)}+\sum_k\left\{\frac{\partial U(t)}{\partial u_k(t)}\frac{\partial u_k(t)}{\partial R_s(t)}\right\}+\gamma\sum_{s'}\left\{\frac{\partial J(t+1)}{\partial R_{s'}(t+1)}\frac{\partial R_{s'}(t+1)}{\partial R_s(t)}\right\}\\&\quad+\gamma\sum_{s'}\left\{\sum_{k'}\left(\frac{\partial J(t+1)}{\partial R_{s'}(t+1)}\frac{\partial R_{s'}(t+1)}{\partial u_{k'}(t)}\frac{\partial u_{k'}(t)}{\partial R_s(t)}\right)\right\}\end{aligned} \tag{5}$$

Therefore, the neural weights in the verification are corrected to minimize the following error function over time

$$E(t)=0.5\sum_s\left(\lambda_s(t)-\lambda_s^{\circ}(t)\right)^2 \tag{6}$$

Using the gradient descent method, the correcting rule of the verification is obtained as

$$\Delta w_i(t)=\alpha\frac{\partial E(t)}{\partial w_i(t)}=\alpha\left(\lambda_s(t)-\lambda_s^{\circ}(t)\right)\frac{\partial\lambda_s(t)}{\partial w_i(t)} \tag{7}$$

where α is the learning rate and w_i is a neural weight of the verification. The critic duplicates the neural weights in the verification and therefore no correcting rule is needed.

The objective of the action is to perform control so as to minimize J. The correcting rule is

$$\begin{aligned}\Delta w_{ik}(t)&=\beta\frac{\partial J(t)}{\partial w_{ik}(t)}=\beta\frac{\partial J(t)}{\partial u_k(t)}\frac{\partial u_k(t)}{\partial w_{ik}(t)}\\&=\beta\left(\frac{\partial U(t)}{\partial u_k(t)}+\gamma\frac{\partial J(t+1)}{\partial u_k(t)}\right)\frac{\partial u_k(t)}{\partial w_{ik}(t)}\\&=\beta\left(\frac{\partial U(t)}{\partial u_k(t)}+\gamma\sum_s\lambda_s^{\circ}(t+1)\frac{\partial R_s(t+1)}{\partial u_k(t)}\right)\frac{\partial u_k(t)}{\partial w_{ik}(t)}\end{aligned} \tag{8}$$

where β is a positive learning rate and $w_{ik}(t)$ is a neural weight of the action. The plant model predicts the immediate future state $R(t+1)$ and calculates certain partial derivatives pertaining to the plant being controlled. The objective of the VNC is to track the desired velocities as closely as possible. Therefore the primary utility function is chosen as

$$U(t)=0.25(v(t)-v_d(t))^2+0.25(\omega(t)-\omega_d(t))^2 \tag{9}$$

B. Neural networks in the velocity neuro-controller

Multilayer perceptrons (MLP) is used to implement the action, verification and critic of the VNC. The action has four inputs ($v(t), w(t), v_d(t), w_d(t)$) and two outputs corresponding to wheel's driving torques ($\tau_l(t), \tau_r(t)$). Figure 4 shows a block diagram of the action neural network. The number of hidden neurons is an experienced choice and each hidden neuron has a hyperbolic-tangent activation function. The output layer has two neurons and each with a linear activation function.

The critic neural network has four inputs ($v(t+1)$, $w(t+1)$, $v_d(t)$, $w_d(t)$) and two outputs ($\lambda^{\circ}_1(t+1)$, $\lambda^{\circ}_2(t+1)$). Figure 5 shows its architecture. The activation functions are chosen the same as that of the action network. The verification neural network has architecture identical to the critic neural network except the inputs and outputs are ($v(t), w(t), v_d(t), w_d(t)$) and ($\lambda_1(t), \lambda_2(t)$), respectively. The DHP method needs the plant model to predict the immediate future states and calculate the plant Jacobian quantities. Although neural modeling is possible, the analytic model of WMR [13] is adopted in this design.

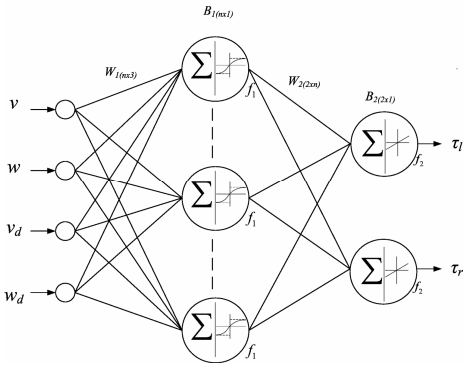

Fig. 4 Architecture of the action neural network

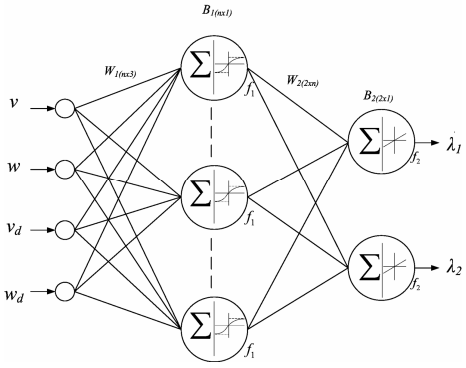

Fig. 5 Architecture of the critic and verification neural networks

C. *The posture neuro-controller (PNC)*

The PNC maps the planned positions to the desired linear and angular velocities. Therefore, learning is obtained by identifying the inverse velocity model of WMR. But for learning convergence, the inverse velocity model and the PNC have standalone neural networks. Figures 6 and 7 show the architectures of the linear and angular PNC, respectively. The number of hidden neurons is an experienced choice. Each hidden neuron has a hyperbolic-tangent activation function. Output layer has two neurons and each with a linear activation function. The linear PNC has 12 inputs organized from two planned and five feedback positions.

$$\begin{aligned} R_v = [&x(t+2)-x(t+1), y(t+2)-y(t+1), x(t+1)-x(t),\\ &y(t+1)-y(t), x(t-1)-x(t), y(t-1)-y(t),\\ &x(t-2)-x(t-1), y(t-2)-y(t-1), x(t-3)-x(t-2),\\ &y(t-3)-y(t-2), x(t-4)-x(t-3), y(t-4)-y(t-3)]^T \end{aligned} \tag{10}$$

The multi-step position inputs also imply the required velocity and acceleration for the PNC to determine the outputs. The outputs are the candidate linear velocities $[v_c(t), v_c(t+1)]$. Similarly, the angular PNC has 18 inputs as below

$$\begin{aligned} R_w = [&x(t+2)-x(t+1), y(t+2)-y(t+1), \theta(t+2)-\theta(t+1),\\ &x(t+1)-x(t), y(t+1)-y(t), \theta(t+1)-\theta(t),\\ &x(t-1)-x(t), y(t-1)-y(t), \theta(t-1)-\theta(t)\\ &x(t-2)-x(t-1), y(t-2)-y(t-1), \theta(t-2)-\theta(t-1),\\ &x(t-3)-x(t-2), y(t-3)-y(t-2), \theta(t-3)-\theta(t-2),\\ &x(t-4)-x(t-3), y(t-4)-y(t-3), \theta(t-4)-\theta(t-3)]^T \end{aligned} \tag{11}$$

The outputs are the candidate angular velocities $[w_c(t), w_c(t+1)]$. The desired velocities are taken as

$$[v_d, w_d] = [v_c(t), \sigma_1 w_c(t) + \sigma_2 w_c(t+1)] \tag{12}$$

where normally $\sigma_1 = 1$ and $\sigma_2 = 0$ while nonzero σ_2 represents feed-forward compensation. The usefulness of the feed-forward compensation in resulting smooth motion will be studied. Figure 8 shows the scheme of learning the inverse velocity model. Backpropagation with Levenberg Marquardt algorithm (LM) [21] is used to correct the neural weights. Supervised drive in variant (random) velocities supplies the training samples to build the neural weights. In autonomous drive, the inverse velocity model is corrected slightly.

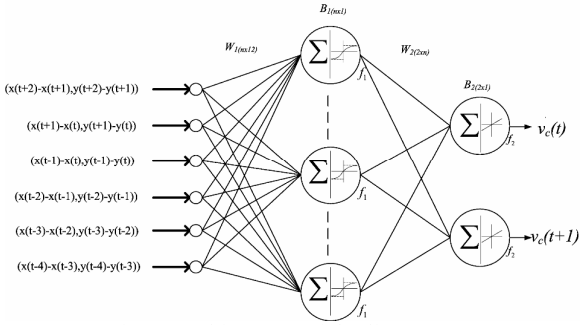

Fig. 6 Architecture of the linear PNC

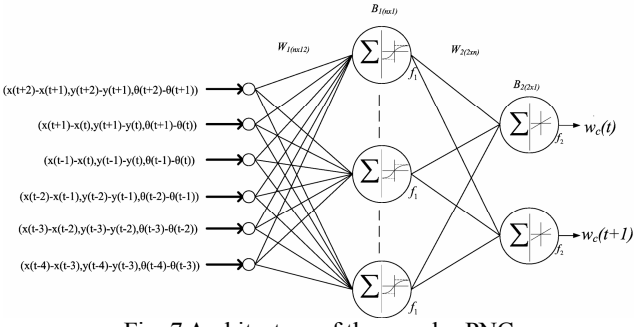

Fig. 7 Architecture of the angular PNC

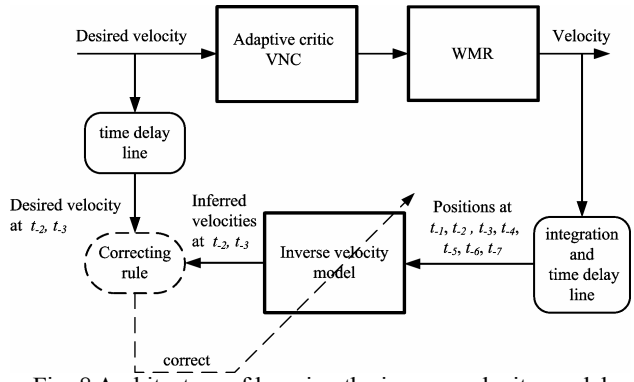

Fig. 8 Architecture of learning the inverse velocity model

D. The path planner

The stereovision unit locates a target and finds a collision free path. According to the path and considering the physical constraints of WMR, the path planner modifies the path by smoothing and produces planed positions to approach the destination. The arc-line algorithm [22] is adopted to plan a smooth path. As illustrated in figure 9, this algorithm mainly replaces the line segments around the intersection of two straight lines with a smooth curve. First, the start point $S(x_s,y_s,\theta_s)$ on the first line, the end point $E(x_e,y_e,\theta_e)$ on the second line, the intersection point $I(x_i,y_i,\theta_i)$ of these two lines, and the angle ($\phi_d = \theta_i - \theta_e$) between these two lines are found. Then we assume a value of curvature (γ) to find the transition point $T(x_t,y_t,\theta_t)$ on the first line, the distance $\gamma\tan(\phi_d/2)$ to the intersection point, and the center point $C(x_c,y_c)$. Finally, we use the arc starts at point T to replace the original straight line segments.

Denote the physical limits of WMR in one-step displacement and steering-angle as $d_{\max}$ and $\phi_{\max}$, respectively. Then by constructing a displacement vector from present position (x_p, y_p, θ_p) to a target position (x_b,y_b) selected on the planed path, the desired displacement d_p and steering angle ϕ_p can be calculated. When they violate the physical limits, the maximum allowable values are used. Then the planed position and orienting angle are calculated as

$$\begin{aligned} x_p(t+1) &= x_p(t) + d_p\cos(\phi_p+\theta_p) \\ y_p(t+1) &= y_p(t) + d_p\sin(\phi_p+\theta_p) \\ \theta_p(t+1) &= \theta_p(t) + \phi_p \end{aligned} \tag{13}$$

The above procedure is iterated once again to obtain planed positions and orienting angles for two sampling times.

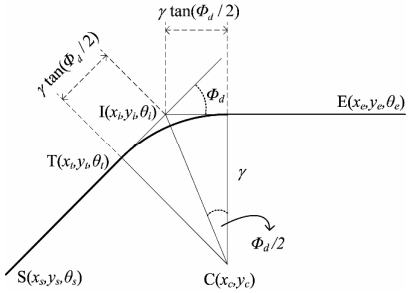

Fig. 9 Planning a smooth path by using the arc-line algorithm

IV. Performance of the DHP Adaptive Critic Motion Control System

A. Performance of the adaptive critic velocity control system

In this study, all neural weights in the neural networks of the action, critic and verification are initialized with values chosen randomly in the range [-0.1, 0.1]. The training samples are generated with (14) for 1000 samples (i=1~1000).

$$\begin{aligned} v_d(i) &= 0.5[\cos(\frac{6\pi(i-1)}{1000}+\pi)+1]e^{-0.001i} \\ w_d(i) &= [\cos(\frac{6\pi(i-1)}{1000}+\pi)+1]\sin(\frac{i-1}{40}) \end{aligned} \tag{14}$$

These training samples are used to train the adaptive critic velocity control system for 500 epochs. Then the WMR system is tested to track a set of desired velocity patterns described by the following equation:

$$v_{t2}(i) = \begin{cases} 0.002i & i \le 300 \\ 0.6 & 300 < i \le 600 \\ 1 & 600 < i \le 700 \\ 3.1 - 0.003i & 700 < i \le 900 \\ 0.4 & i > 900 \end{cases}$$

$$w_{t2}(i) = \begin{cases} -0.002i & i \le 300 \\ -0.6 & 300 < i \le 600 \\ -1 & 600 < i \le 700 \\ 0.003i - 3.1 & 700 < i \le 900 \\ -0.4 & i > 900 \end{cases} \tag{15}$$

Figures 10 and 11 show the results of velocity tracking. The accurate velocity tracking confirms the excellent performance of the adaptive critic velocity control system.

B. Performance of the inverse velocity model

In this study, all neural weights are initialized with values chosen randomly in the range [-0.1, 0.1]. The fuzzy posture controller used in [13] is connected to the adaptive critic velocity control system to conduct the WMR motion. During tracking a variety of trajectories, the desired velocities and the corresponding WMR positions at each sampling time are recorded as the training samples to train the PNC. Then the trained WMR system is commanded to track a sequence of positions calculated with

$$y = \cos(\frac{33\pi}{1000}x) \tag{16}$$

The desired velocity and the corresponding output position at every sampling time are recorded as test samples. Then the position test samples are fed into the inverse velocity model sequentially and the outputs are compared with the velocity test samples. The results are presented in Figures 12 and 13. These excellent results show the inverse velocity model is well trained.

C. Performance of the posture control system

The trained WMR system is commanded to track a path containing a right turn. The following cases study the performance of the posture control system under d_{max}=0.035 and $\phi_{\max}$=0.03.

Case 1: $\sigma_1 = 1$ and $\sigma_2 = 0$ in (12)

This case compares the responses of the posture control system with and without the path planer. Without the path planner the system has no knowledge of the right turn until it occurs. Figure 14 shows, although the path tracking is successful, large overshoot occurs around the right turn. When the path planner makes out planned positions along a smooth path with *R*=1.1. The result in figure 14 shows the overshoot is eliminated.

Case 2: effect of feed-forward compensation, $\sigma_2 \neq 0$ in (12)

This case studies the possibility of substituting the path planner by using $\sigma_2 \neq 0$ in (12). Figure 15 shows the results of using $\sigma_1 + \sigma_2 = 1$. It seems no observable improvement is obtained. Figure 16 compares the results of using $\sigma_1 = 1$ for $\sigma_2 = 0$, $\sigma_2 = 0.1$, and $\sigma_2 = 0.2$, respectively. This application shows the feed-forward compensation may change the orientation in advance to reduce the overshoot. But in general using the path planer obtains better results.

V. Conclusion

Adaptive critic motion control design extended the autonomous capability of WMR to learn the control function by trial. Eventually, theoretical analysis and synthesis of the dynamic control system were reduced to the minimum. Detailed formulations of the DHP adaptive critic motion control design were presented. The VNC corrected the neural weights by sequential optimization so that the WMR motion control was able to comply with the plant dynamics and unknown disturbances. The PNC mapped planned positions to desired velocities by implementing the inverse velocity model of WMR. The inverse velocity model was obtained by on-line neural network approximation. The overall motion control system learned to optimize the specified objective functions instead of any existing controller or representative training samples. The feasibility of the proposed design was validated on the experimental WMR and successful results were obtained. Further research should study the design strategy to guarantee the convergence of multi-loop learning systems.

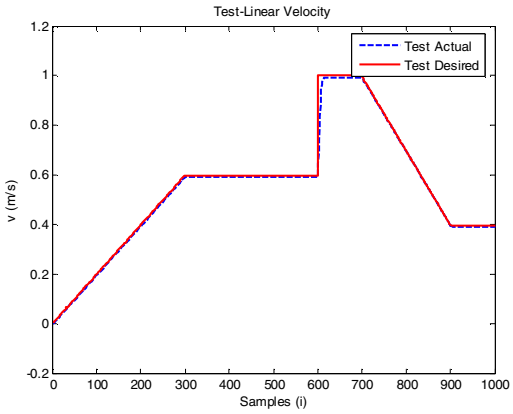

Fig. 10 Result of the linear velocity tracking

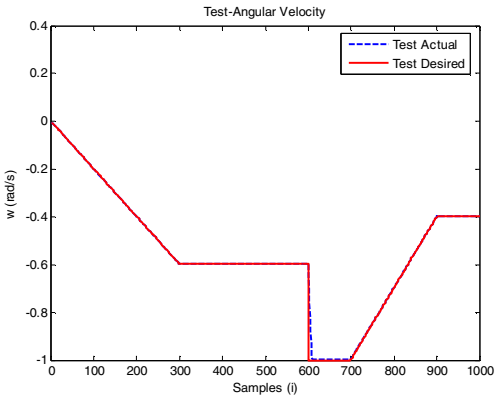

Fig. 11 Result of the angular velocity tracking

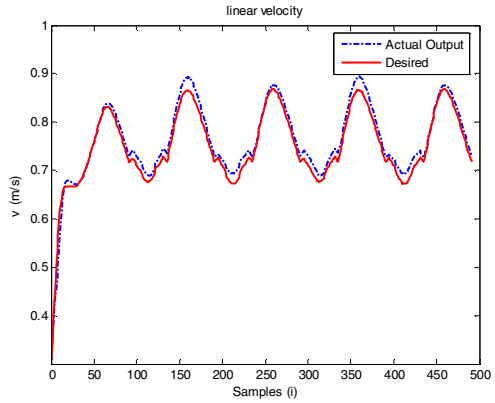

Fig. 12 Comparing model generated linear velocity (actual) with the test sample (desired)

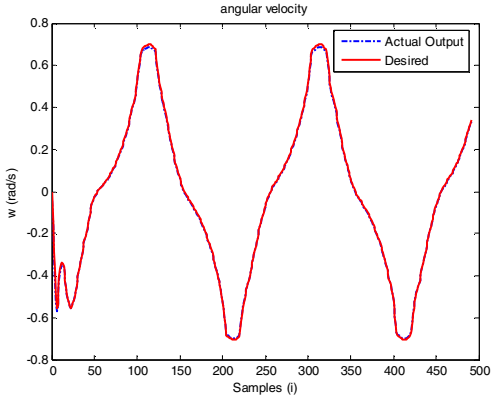

Fig. 13 Comparing model generated angular velocity (actual) with the test sample (desired)

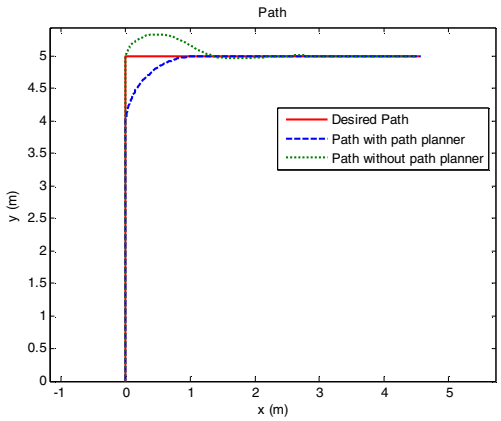

Fig. 14 Path tracking with and without the path planer

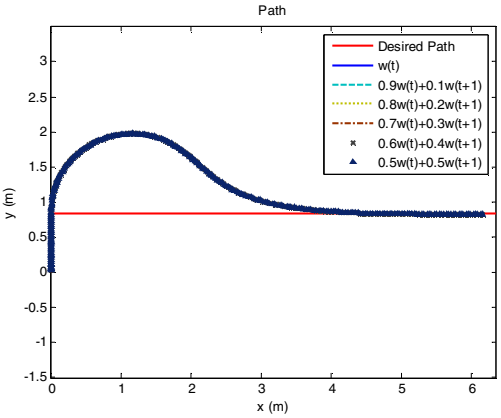

Fig. 15 Path tracking using the weighted average method

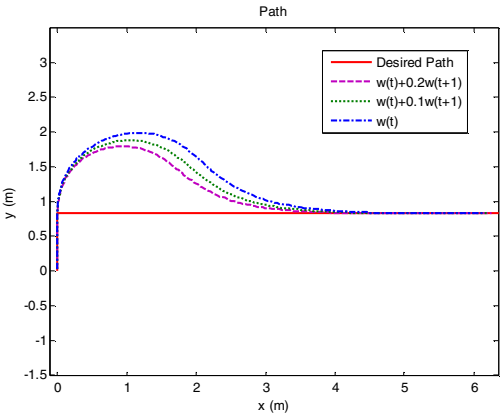

Fig. 16 Path tracking using the feed-forward compensation method

ACKNOWLEDGMENT

The authors gratefully acknowledge National Science Council of Taiwan on project NSC94-2213-E-002-049 and National Taiwan University on project NTU95R0036-07.

REFERENCES

[1] D. T. Greenwood, *Principles of Dynamics*, Prentice-Hall, 1988.

[2] T. H. Lee, F. H. F. Leung, and P. K. S. Tam, "Position control for wheeled mobile robots using a fuzzy logic controller," in *IECON '99 Proceedings of The 25th Annual Conference of the IEEE Industrial Electronics Society,* vol. 2, pp. 525 -528, 1999.

[3] A. M. Bloch, M. Reyhanoglu, and N. H. McClamroch, "Control and stabilization of nonholonomic dynamic systems," *IEEE Trans. Automatic Control*, vol. 37, pp. 1746-1757, 1992.

[4] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol.1, pp. 384 -389, 1990.

[5] P.-S. Tsai, T.-F. Wu, F.-R. Chang and L.-S. Wang, "Tracking control of nonholonomic mobile robot using hybrid structure," *The 6th World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, Florida, 2002.

[6] R. Colbaugh, E. Barany, and K. Glass, "Adaptive control of nonholonomic robotic systems," *Journal of Robotic Systems*, vol. 15, no. 7, pp. 365-393, 1998.

[7] K. H. Park, S. B. Cho, Y.-W. Lee, " Optimal tracking control of a nonholonomic mobile robot," in *Proceedings ISIE 2001 IEEE International Symposium on Industrial Electronics*, vol. 3, pp. 2073 -2076, 2001.

[8] S. Lee, T. M. Adams, and B. Ryoo, "A fuzzy navigation system for mobile construction robots," *Automation in Construction*, vol. 6, pp. 97-107, 1997.

[9] S. Pawlowski, K. Kozlowski, and W. Wroblewski, "Fuzzy logic implementation in mobile robot control," in *Proceedings of the Second International Workshop on Robot Motion and Control*, pp. 65 -70, 2001.

[10] R. Fierro and F.L. Lewis, "Control of a nonholonomic mobile robot using neural networks," *IEEE Trans. on Neural Networks*, vol. 9, no. 4, pp. 589-600, 1998.

[11] K. S. Narendra and K. Parthasathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. On Neural Networks*, vol. 1, no. 1, pp. 4-27, March 1990.

[12] J.-S. Jang, Chuen-Tsai Sun, "Neuro-fuzzy modeling and control," *Proceedings of the IEEE*, vol. 83, Issue 3, pp. 378-406, March 1995.

[13] Wei-Song Lin, Chin-Lung Huang, Ming-Kang Chuang and Ghing-Chieh Liu, "Modeling a wheeled mobile robot for autonomous navigation design**,"** *IASTED International Conference on Modeling, Identification and Control*, pp. 275-280, Grindelwald, Switzerland, Feb. 2004

[14] Wei-Song Lin, Chin-Lung Huang and Ming-Kang Chuang, "Hierarchical fuzzy control for autonomous navigation of wheeled robots," *IEE Proc. Control Theory and Applications*, vol. 152, issue 5, pp. 598-606, Sept. 2005.

[15] Wei-Song Lin, Ming-Kang Chuang and Glorious Tien, "Autonomous mobile robot navigation using stereovision," in *Proceedings of the IEEE International Conference on Mechatronics*, Taipei, Taiwan, pp. 410-415, July 10-12, 2005.

[16] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Contorl*, White and Sofge, Eds. New York: Van Nostrand Reinhold, pp. 493-525.

[17] D. Prokhorov and D. Wunsch, "Adaptive critrc designs," *IEEE Trans. Neural Networks*, vol. 8, pp. 997-1007, Sept. 1997.

[18] X. Yun, and Y. Yamamoto, "Internal dynamics of a wheeled mobile robot," in *Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems,* pp.1288-1293, 1993.

[19] D. Prokhorov and R. Santiago, and D. Wunsch, "Adaptive critrc designs: a case study for Neurocontrol," *Neural Networks*, vol. 8, pp. 1367-1372, 1995.

[20] G. G. Lendaris, and T. T. Shannon, "Application considerations for the DHP methodology," in *Proceedings of the International Joint Conference on Neural Networks'98 (IJCNN'98)*, Anchorage, IEEE Press, pp 1013-1018, March, 1998.

[21] B. M. Wilamowski, "Neural network architectures and learning," *IEEE International Conference on Industrial Technology*, 10-12 Dec. 2003 Page(s):TU1 - T12 vol.1.

[22] W. L. Nelson, "Continuous steering-function control of robot carts," *IEEE Transactions on Industrial Electronics*, vol.36, no.3, pp.330-337, Aug.1989.