

# Integration of Fuzzy Controller with Adaptive Dynamic Programming

Yuanheng Zhu, Dongbin Zhao

State Key Laboratory of Management and Control for  
Complex Systems  
Institute of Automation Chinese Academy of Sciences  
Beijing, China

e-mail: [zyh7716155@163.com](mailto:zyh7716155@163.com), [dongbin.zhao@gmail.com](mailto:dongbin.zhao@gmail.com)

Haibo He

Department of Electrical, Computer and Biomedical  
Engineering  
University of Rhode Island  
Kingston, RI 02881, USA  
e-mail: [he@ele.uri.edu](mailto:he@ele.uri.edu)

**Abstract**—Adaptive dynamic programming (ADP) is an effective method for learning while fuzzy controller has been put into use in many applications because of its simplicity and no need of accurate mathematic modeling. The combination of ADP and fuzzy control has been studied a lot. Before this paper, we have studied using ADP to learn the fuzzy rules of a Monotonic controller, which shows good performance. In this paper, a hyperbolic fuzzy model is adopted to make an improvement. In this way, both membership function and fuzzy rules are learned. With ADP algorithm, fuzzy controller has the capacity of learning and adapting. Simulations on a single cart-pole plant and a rotational inverted pendulum are implemented to observe the performance, even with uncertainties and disturbances.

**Keywords**—adaptive dynamic programming; fuzzy control; cart-pole plant; rotational inverted pendulum

## I. INTRODUCTION

Dynamic programming is deemed as an effective method to solve nonlinear stochastic dynamic problems [1~4] in all kinds of application fields. The optimization foundation is the Bellman equation, developed by Bellman in 1953 [5]. But it suffers “the curse of dimensionality” of computation when the states increase. An alternative way to solve this problem is to approximate the Bellman  $J$  function (the “value function”), which coins the idea of adaptive dynamic programming (ADP). By developing a parameterized model which is trained to approximate  $J$ , the controller is trained to minimize the function  $J$  [6]. In general, the ADP algorithm is divided into two parts, the “Critic” and the “Actor”, separately served as an approximator to  $J$  and a system controller.

Fuzzy controller is proposed as a method to convert “a linguistic control strategy based on expert knowledge into an automatic control strategy” [7][8]. Fuzzy control has two important parts, membership functions and fuzzy rules. In most cases, parameters of fuzzy IF-THEN rules and membership functions are provided according to the experience and knowledge of human experts. If the provided fuzzy rules have bad control performance, then an adaptive law is added to update the parameters, which is also called “adaptive fuzzy control” [9]. Fuzzy control can be successfully integrated with adaptive dynamic programming [10~13], to embody prior knowledge about the control objective, and improve its

learning efficiency. In our paper [14], a Monotonic fuzzy controller is developed, in which only the fuzzy rules are trained by ADP. The results confirm that the combination of ADP and fuzzy controller is efficient.

Here, some further research is studied in this paper. A complex hyperbolic fuzzy model is adopted. In this way, both membership functions and fuzzy rules are to be trained by ADP. In Section II, we present a general description of the ADP with fuzzy controller. In Section III, the new proposed method is applied to a cart-pole system, and is compared to the previous results. In Section IV, a more complex system-rotational inverted pendulum is adopted to verify the method. Section V presents some discussions about the proposed method and its simulation results. In the end, we give some conclusions.

## II. ADAPTIVE DYNAMIC PROGRAMMING WITH FUZZY CONTROLLER

### A. The Mechanism of ADP

The adopted approach proposed by Si and Wang in [15], is closely related to ADHDP. Fig. 1 is a schematic diagram of the ADP algorithm.

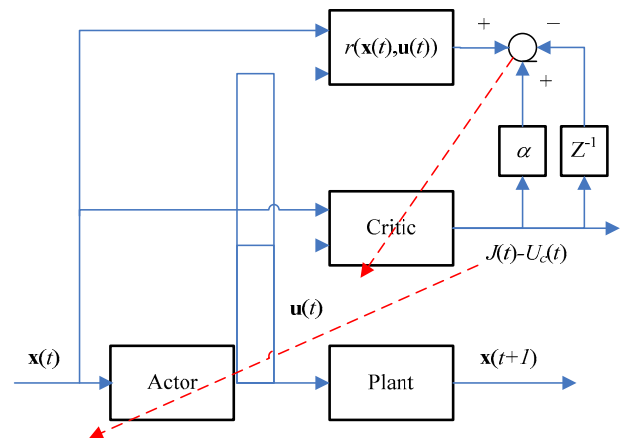


Figure 1. A schematic diagram of ADP. Signal flow is transmitted along the solid lines while parameters are trained along the dashed lines.

This work was supported partly by National Natural Science Foundation of China (Nos. 60874043, 60921061, 61034002), and Beijing Natural Science Foundation (No. 4122083)

This work was also supported in part by the National Science Foundation (NSF) under Grant ECCS 1053717

In this paper, the Actor is a fuzzy controller, and the Critic is a network. The reinforcement signal  $r(t)$  is either a “0” or a “-1” corresponding to “success” or “failure”, respectively. The output of the Critic is expected to estimate the cost-to-go  $J(t)$ . For the Critic element, the prediction error and the objective function are defined as

$$e_c(t) = \alpha J(t) - [J(t-1) - r(t)] \quad (1)$$

$$E_c(t) = \frac{1}{2} e_c^2(t). \quad (2)$$

where  $\alpha$  is a discount factor for the infinite-horizon problem ( $1 < \alpha < 1$ ) which is set 0.95 in this paper.

The objective of training the Critic network is to lower  $E_c(t)$  close to zero. During the training process, gradient descent method (GD) is used to train parameters.

Similar to the Critic network, the Actor is expected to provide optimal action signal  $u(t)$  which is to minimize the error between the desired ultimate objective, denoted by  $U_c$ , and the function  $J$ . According to our above definition,  $U_c$  is set “0” for “success”. So the error  $e_a(t)$  and the performance error  $E_a(t)$  are defined as

$$e_a(t) = J(t) \quad (3)$$

$$E_a(t) = \frac{1}{2} e_a^2(t) \quad (4)$$

The whole process of the ADP algorithm is summarized as follows. Firstly, the Actor and Critic are both initialized randomly. Then the Actor generates an action signal  $u(t)$  based on states  $X(t)$  and the function  $J(t)$  is calculated by the Critic element, with the input of  $u(t)$  and  $X(t)$ . Combined with the previous  $J(t-1)$  and the reinforcement signal  $r(t)$ ,  $e_c(t)$  and  $E_c(t)$  are derived. Then the Critic is adapted according to  $E_c(t)$  with the gradient descent method or PSO, or some other methods, until meeting some criterion. Afterwards, the Actor is modified with  $E_a(t)$  in the same way. The next states  $X(t+1)$  are calculated from the system with the action signal  $u(t)$  generated by the modified Actor element. Then, the iteration process continues for the next cycle [16~20]. For more details, readers can refer to our previous papers.

#### B. The Hyperbolic fuzzyModel

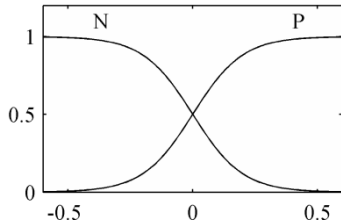


Figure 2. Hyperbolic fuzzyfunctions

In our previous research [14], the fuzzy controller is adopted with the simplest and most common one, which is composed of Monotonic membership functions and several fuzzy rules. In this paper, the membership functions are replaced by a hyperbolic fuzzy model, which is shown in Fig. 2.

The formulations are summarized as

$$\begin{aligned} \mu_{i,N} &= \frac{1}{2} [1 - \tanh(\theta_i \cdot X_i)] \\ \mu_{i,P} &= \frac{1}{2} [1 + \tanh(\theta_i \cdot X_i)] \end{aligned} \quad (5)$$

where  $i=1 \cdots n$ ,  $X_i$  is the input variables,  $\theta_i$  is the parameter of the membership function. Then, for each fuzzy control rules, the weights are calculated as

$$\omega_r = \prod_{i=1}^n \mu_{i,j_i} \quad (6)$$

where  $j_i = N$  or  $P$ ,  $r=1 \cdots 2^n$ . With fuzzy control rules  $R$ , the output of fuzzy controller  $u(t)$  is generated by

$$u(t) = \sum_{r=1}^{2^n} \omega_r \cdot R_r \quad (7)$$

#### C. The Critic Network

The critic network used in this paper is a simple feed-forward neural network with one hidden layer, with the same structure as that in the previous studies [14]. For clear comparison, only GD method is adopted in this paper. For more detailed description, refer to [14].

#### D. Training Actor with Gradient Descent (GD) Method

Compared to [14], the fuzzy membership parameters are updated to get better control performance with the GD method as

$$\begin{aligned} \Delta \theta_i(t) &= l_a(t) \left[ -\frac{\partial E_a(t)}{\partial \theta_i(t)} \right] \\ &= -l_a(t) \frac{\partial E_a(t)}{\partial J(t)} \frac{\partial J(t)}{\partial u(t)} \frac{\partial u(t)}{\partial \theta_i(t)} \\ &= -l_a(t) e_a(t) \sum_{j=1}^{N_h} \left[ \omega_{c_j}^{(2)}(t) \frac{1}{2} (1 - p_j^2(t)) \omega_{c_{j,n+1}}^{(1)}(t) \right] \quad (8) \\ &= \sum_{r=1}^{16} \left[ R_r \left( \prod_{\substack{i=1 \\ i \neq j_i}}^4 \mu_{i,j_i} \right) \frac{\partial \mu_{i,j_i}}{\partial \theta_i} \right] \end{aligned}$$

where

$$\frac{\partial \mu_{i,j_i}}{\partial \theta_i} = \begin{cases} -\frac{1}{2} \sec^2 h^2(\theta_i X_i) \cdot X_i, & \text{if } j_i = N, \\ \frac{1}{2} \sec^2 h^2(\theta_i X_i) \cdot X_i, & \text{if } j_i = P. \end{cases} \quad (9)$$

The fuzzy rules are updated with the GD method as

$$\begin{aligned}\Delta R_r(t) &= l_a(t) \left[ -\frac{\partial E_a(t)}{\partial R_r(t)} \right] \\ &= -l_a(t) \frac{\partial E_a(t)}{\partial J(t)} \frac{\partial J(t)}{\partial u(t)} \frac{\partial u(t)}{\partial R_r(t)} \\ &= -l_a(t) e_a(t) \omega_r \sum_{i=1}^{N_h} \left[ \omega_{c_i}^{(2)}(t) \frac{1}{2} (1 - p_i^2(t)) \omega_{c_{i,n+1}}^{(1)}(t) \right].\end{aligned}\quad (10)$$

### III. ADP WITH HYPERBOLIC FUZZY MODEL FOR A CART-POLE PLANT

The new proposed method is now simulated on a single cart-pole plant. The plant is a single pole mounted on a cart and the objective of the ADP algorithm is to train the membership functions and fuzzy rules from initially random values to the capability of balancing the pole.

#### A. The Cart-Pole Balancing Problem

The cart-pole plant used here which is shown in Fig. 4 can also be found in [15] and [21], and the model formulas are summarized in the following.

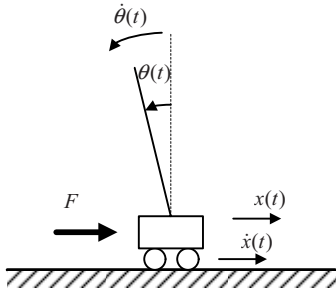


Figure 3. Schematic diagram of a single cart-pole plant

$$\frac{d^2\theta}{dt^2} = \frac{g \sin \theta + \cos \theta [-F - ml\dot{\theta}^2 \sin \theta + \mu_c \operatorname{sgn}(\dot{x})] - \frac{\mu_p \dot{\theta}}{ml}}{l \left( \frac{4}{3} - \frac{m \cos^2 \theta}{m_c + m} \right)} \quad (11)$$

$$\frac{d^2x}{dt^2} = \frac{F + ml[\dot{\theta}^2 \sin \theta - \ddot{\theta} \cos \theta] - \mu_c \operatorname{sgn}(\dot{x})}{m_c + m} \quad (12)$$

where

- $g$  9.8 m/s<sup>2</sup>, acceleration due to gravity;
- $m_c$  1.0 kg, mass of cart;
- $m$  0.1 kg, mass of pole;
- $l$  0.5 m, half-pole length;
- $\mu_c$  0.0005, coefficient of friction of cart on track;
- $\mu_p$  0.000 002, coefficient of friction of pole on cart;

$$\operatorname{sgn}(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x = 0 \\ -1, & \text{if } x < 0. \end{cases}$$

The system states  $X(t)$  are constituted of four variables: 1)  $x(t)$ , position of the cart with the reference to the center of the track; 2)  $\theta(t)$ , the angle of the pole with the reference to the vertical position; 3)  $\dot{x}(t)$ , the velocity of the cart; 4)  $\dot{\theta}(t)$ , the angular velocity of the pole.

During the trials, the controller is considered a failure when  $\theta(t)$  or  $x(t)$  are outside their predefined ranges. At that moment, the learning process stops and restarts a new trial. So the reinforcement signal is defined as

$$r = \begin{cases} 0 & \text{If } -12^\circ < \theta < 12^\circ \text{ and } -1 < x < 1, \\ -1 & \text{Otherwise.} \end{cases} \quad (13)$$

#### B. Simulation Results

Some parameters are defined before the simulations. In ADP algorithm, the learning rate of both the Critic  $l_c(t)$  and the Actor  $l_a(t)$  is 0.005; the maximum adapting cycles for the Critic  $N_c$  and the Actor  $N_a$  are 50 and 100, respectively; the training error threshold  $T_c$  and  $T_a$  are 0.05 and 0.005, respectively. The hidden nodes  $N_h$  of the critic network is 6; time step  $dt$  is 0.02s. Besides, boundaries  $B$  of four membership functions are [1m, 12°, 1.5m/s, 120°/s].

During simulations, there are at most 1000 consecutive trials in a run. If any of the 1000 trials has lasted 6000 steps, it is considered successful and the run stops. Otherwise, if the fuzzy controller still fails after 1000 trials, the run is unsuccessful.

Both bang-bang and continuous control strategies are implemented for 100 runs to calculate their success rate and average trials to success. In bang-bang control strategy, the constant force is  $\pm 10N$ , while in continuous control strategy, the conversion gain  $K_s$  is 40 multiplied to  $u$ .

To be more realistic, sensor noise is added to the state measurements. Two kinds of noise, uniform and Gaussian noise are added to the angle measurements  $\theta(t)$ . The simulation results are shown in Table I. Besides, for comparison, the results of new fuzzy controller with hyperbolic model, old fuzzy controller with Monotonic, and neural network controller [15] are listed in Table II. All controllers implement bang-bang control strategy.

TABLE I. COMPARISON OF TWO CONTROL STRATEGIES FOR CART-POLE PLANT WITH DIFFERENT KINDS OF NOISE USING FUZZY HYPERBOLIC MODEL

Noise type	Bang-bang		Continuous	
	Success rate	# of trials	Success rate	# of trials
Noise free	100%	21.67	100%	38.71
Uniform 5%	100%	21.65	100%	37.18
Uniform 10%	100%	17.46	100%	36.99
Gaussian $\sigma^2 = 0.1$	100%	25.23	100%	43.35
Gaussian $\sigma^2 = 0.2$	100%	29.87	100%	34.46

From the results, the bang-bang control is still more suitable for cart-pole plant than continuous control. It is obvious that the new method provides better performance on cart-pole system than the old one, and much better than neural network controller. With the hyperbolic fuzzy model, the added noise has a much less impact on training, mostly because of the superiority of fuzzy control and both membership functions and fuzzy rules being learned. The training trials are also less than the other two controllers when the measurement is polluted by noise. This simulation verifies that by training both membership functions and fuzzy rules, the controller is capable to provide a more efficient performance.

TABLE II. COMPARISON THE RESULTS OF THREE CONTROLLERS AT SAME SIMULATION CONDITIONS WITH BANG-BANG CONTROL STRATEGY

Noise type	Hyperbolic Fuzzy Model		Monotonic Fuzzy Model [14]		NN[15]	
	Success rate	# of trials	Success rate	# of trials	Success rate	# of trials
Noise free	100%	21.67	100%	30.42	100%	6
Uniform 5%	100%	<b>21.65</b>	100%	37.42	100%	32
Uniform 10%	100%	<b>17.46</b>	100%	41.79	100%	54
Gaussian $\sigma^2 = 0.1$	100%	<b>25.23</b>	100%	45.65	100%	164
Gaussian $\sigma^2 = 0.2$	100%	<b>29.87</b>	100%	53.28	100%	193

#### IV. ADP WITH HYPERBOLIC FUZZY MODEL FOR A ROTATIONAL INVERTED PENDULUM

As the simple cart-pole plant is simulated, now a complex system-rotational inverted pendulum (RIP) is adopted to test the new method.

##### A. The Rotational Inverted Pendulum Problem

A schematic diagram of RIP is shown in Fig. 5, and the system can be formulated as [22~23]:

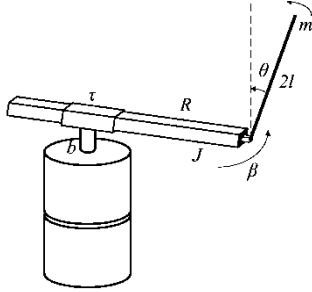


Figure 4. The schematic diagram of the rotational inverted pendulum

$$\begin{aligned} \tau_m &= mlR\ddot{\theta} \cos \theta - mlR\dot{\theta}^2 \sin \theta + b\ddot{\beta} + (J + mR^2)\ddot{\beta} \\ mg \sin \theta &= mR\ddot{\beta} \cos \theta + ml\ddot{\theta} \end{aligned} \quad (14)$$

where  $\tau_m$  is the external torque,  $\theta$  is the pendulum angle,  $\beta$  is the arm angle, and

- $g$  9.8 m/s<sup>2</sup>, acceleration due to gravity;
- $m$  0.05kg, the mass of the pendulum;
- $l$  48cm, the half length of the pendulum;

- $R$  57cm, the length of the arm;
- $J$  0.03264 kg · m<sup>2</sup>, the moment of inertia of the rotating arm;
- $b$  0.00351 kg · m<sup>2</sup>/s, the pivot's viscous friction coefficient.

Similarly, the objective of this simulation is to balance both pendulum angle and arm angle within some ranges. So the reinforcement signal is defined here as:

$$r = \begin{cases} 0 & \text{If } -12^\circ < \theta < 12^\circ \text{ and } -40^\circ < \beta < 40^\circ, \\ -1 & \text{Otherwise.} \end{cases} \quad (15)$$

The states consist of the two angles and their angular velocities, namely  $X(t) = \{\theta, \dot{\theta}, \beta, \dot{\beta}\}$ . Here, only continuous control strategy is applied to the system, and the conversion gain  $K_s$  is 10 from  $u$  to  $\tau$ .

##### B. Simulation Results

The parameters and simulation process adopted here are defined and denoted the same as the above simulation, except that only continuous control strategy is implemented here. Besides, an extra Gaussian noise is added to the measurement of pendulum angle  $\theta$ . The simulation results for two control strategies are listed in Table III. For comparison, the third column presents the simulation results using a monotonic fuzzy model.

TABLE III. COMPARISON OF TOW FUZZY CONTROLLERS AT SAME SIMULATION CONDITIONS WITH CONTINUOUS CONTROL STRATEGY

Noise type	Hyperbolic Fuzzy Model		Monotonic Fuzzy Model[14]	
	Success rate	# of trials	Success rate	# of trials
Noise free	100%	<b>35.72</b>	50%	404.30
Gaussian $\sigma^2 = 0.1$	100%	<b>72.56</b>	#	#

According to the results, training both membership functions and fuzzy rules makes a great improvement on increasing success rate and reducing training trials compared to our old method which only training fuzzy rules. The new proposed method ensures the training success rate to 100%, even disturbed by noise. It indicates that the new method is insusceptible of noise disturbance and has a perfect robustness. Fig. 6 is a typical trajectory of angles, torque and function  $J$  of the last trials during training process. From this figure, it is obvious that the fuzzy controller is gradually trained from instability to the stability of balancing the system. Fig. 7 shows the shape of membership functions after training. They all present satisfying results.

#### V. DISCUSSIONS

From the simulations, the new proposed method improves efficiently for the combination of fuzzy control and ADP. It is verified that training both membership functions and fuzzy rules ensures a higher success rate and less training trials by both cart-pole plant and RIP. Besides, the hyperbolic fuzzy model shows a better robustness than the Monotonic model and the neural network with the fact that it is more insusceptible by

noise disturbance. With the ADP algorithm implemented, the parameters of fuzzy controller are trained from random to appropriate values, rather than being provided according to the experience of experts beforehand.

Besides, some further simulations are implemented. For fuzzy controller in cart-pole system using continuous control strategy, after training successfully, even with a  $20^\circ$  initial angel, the controller can still balance the pole within about 200 steps. While for the neural network controller in [16] after training, the pole is out of control if the initial angel is above  $5^\circ$ .

## VI. CONCLUSIONS

Adaptive dynamic programming combined with fuzzy control has been brought to the forefront for many years. Both in theoretical analysis and actual practice, the method has caused a lot interest for researchers. In this paper, it is simulated that under-actuated systems are able to be balanced by the fuzzy controller after the membership functions and fuzzy rules being learned with the ADP algorithm. With the simplicity and easy structure of fuzzy control, the controller shows an excellent robustness and a wide applicability compared to other control methods with better control performance. Based on the ADP algorithm, the parameters of fuzzy controller are derived without the expert knowledge. But some other flaws still need further research, for example the stability and convergence of this method are not studied. We look forward to more theoretical research.

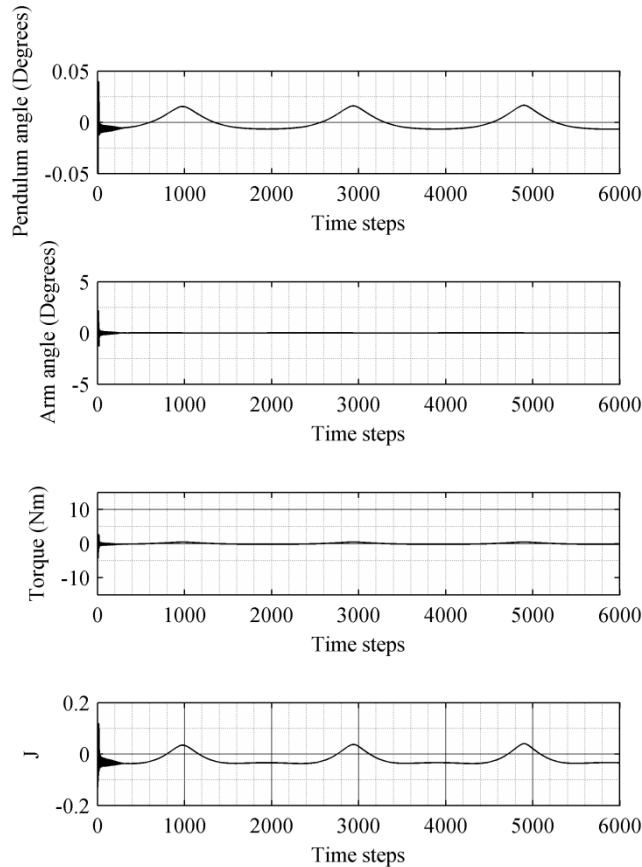


Figure 5. A typical trajectory of angles, torque and function  $J$  of the last trials during training process.

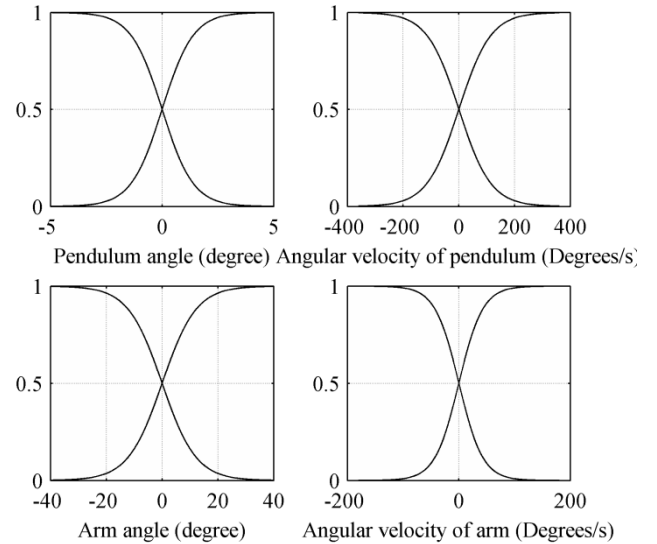


Figure 6. The shape of membership functions after training.

## REFERENCES

- [1] R. Bellman and S. Dreyfus, *Applied Dynamic Programming*. Princeton, N. J.: Princeton Univ. Press, 1962.
- [2] T. Borghers and R. Sarin, "Learning through reinforcement and replicator dynamics," *J. Economic Theory*, vol. 77, no. 1, pp. 1–17, 1997.
- [3] J. Dalton and S. N. Balakrishnan, "A neighboring optimal adaptive critic for missile guidance," *Math. Comput. Modeling*, vol. 23, no. 1, pp. 175–188, 1996.
- [4] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Trans. Automat. Contr.*, vol. 42, pp. 674–690, May 1997.
- [5] P. J. Werbos, "Using adaptive dynamic programming to understand and replicate brain intelligence: The next level design," in *Neurodynamics of Higher-Level Cognition and Consciousness*, R. Kozma, Ed. Berlin, Germany: Springer-Verlag, 2007.
- [6] P. J. Werbos, "Foreword - ADP: The key direction for future research in intelligent control and understanding brain intelligence," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 38, no. 4, pp. 898–900, Aug. 2008.
- [7] C.-C. Lee, "Fuzzy logic in control systems: fuzzy logic controller-Part I," *IEEE Trans. Syst., Man, Cybern.*, vol. 20, pp. 404–418, 1990.
- [8] A. Katbab, "Fuzzy logic and controller design-a review," in *Proc. IEEE Southeastcon*, Raleigh, NC, Mar. 1995, pp. 443–449.
- [9] L.-X. Wang, "Stable adaptive fuzzy control of nonlinear systems," *IEEE Trans. Fuzzy Syst.*, vol. 1, pp. 146–155, Jan. 1993.
- [10] T. Shanon, G. Lendaris (2000), "Adaptive critic based approximate dynamic programming for tuning fuzzy controllers," in *Proc. IEEE Int. Conf. Fuzzy Systems*, San Antonio, May 2000.
- [11] G. G. Lendaris, R. A. Santiago, and M. S. Carroll, "Dual heuristic programming for fuzzy control," in *Proc. IFSA/NAFIPS Conference*, Vancouver, B.C., July 2002.
- [12] S. Mohagheghi, G. K. Venayagamoorthy, and R. G. Harley, "Adaptive critic designs based neuro-fuzzy controller for a static compensator in a multimachine power system," *IEEE Trans. Power Syst.*, vol. 21, no. 4, pp. 1744–1754, Nov. 2006.
- [13] T. Li, D. B. Zhao, and J. Q. Yi, "Adaptive dynamic neuro-fuzzy system for traffic signal control," in *Proc. IEEE Int. Joint Conf. Neural Networks*, Hong Kong, June 2008, pp. 1840–1846.
- [14] D. B. Zhao, Y. H. Zhu, H. B. He, "Neural and Fuzzy Dynamic Programming for Under-actuated Systems," in *Proc. IEEE Int. Joint Conf. Neural Networks*, Brisbane, June 10–15, 2012, accepted.

- [15] J. Si and Y. T. Wang, "On-line learning control by association and reinforcement," *IEEE Trans. on Neural Networks*, vol.12, no.2, pp. 264–276, 2001.
- [16] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control*, D. A. White and D. A. Sofge, Eds. New York: Van Nostrand Reinhold, 1992, ch. 13.
- [17] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Networks*, vol. 8, no. 5, pp. 997–1007, Sept. 1997.
- [18] D. B. Zhao, J. Q. Yi, and D. R. Liu, "Particle swarm optimized adaptive dynamic programming," in *Proc. IEEE Int. Symp. Approximate Dynamic Programming and Reinforcement Learning*, 2007, pp. 32–37.
- [19] H. He and B. Liu, "A hierarchical learning architecture with multiple-goal representations based on adaptive dynamic programming," in *Proc. IEEE Int. Conf. Networking, Sensing and Control*, Chicago, April, 2010, pp. 286–291.
- [20] T. Li, D. B. Zhao, and J. Q. Yi. "Heuristic dynamic programming strategy with eligibility traces," in *Proc. American Control Conference*. Seattle, USA, 2008. pp. 4535–4540.
- [21] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Syst., Man. and Cyhern.* vol. 13, pp. 834–846, Oct. 1983.
- [22] N. Chanapan, S. Panaudomsup, J. Ngamwiwit. and N. Komine., "Experimental study of rotational inverted pendulum," in *Proc. of Korea Automatic Control Conference*, Korea, October 2000.
- [23] A. I. Cahyadi, D. Isarakorn, T. Benjanarasuth, J. Ngamwiwit, and N. Komine, "Application of coefficient diagram method for rotational inverted pendulum control," in *Control, Automation, Robotics and Vision Conference*, 2004, pp. 1769–1773.