Neurocomputing 149 (2015) 124–131

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Convergence analysis and application of fuzzy-HDP for nonlinear discrete-time HJB systems ☆

Yuanheng Zhu, Dongbin Zhao *, Derong Liu

The State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

ARTICLE INFO

Article history:
Received 29 May 2013
Received in revised form
4 September 2013
Accepted 11 November 2013
Available online 31 July 2014

Keywords:
Discrete-time nonlinear system
T–S fuzzy system
HDP

ABSTRACT

In this paper, a type of fuzzy system structure is applied to heuristic dynamic programming (HDP) algorithm to solve nonlinear discrete-time Hamilton–Jacobi–Bellman (DT-HJB) problems. The fuzzy system here is adopted as a 0-order T–S fuzzy system using triangle membership functions (MFs). The convergence of HDP and approximability of the multivariate 0-order T–S fuzzy system is analyzed in this paper. It is derived that the cost function and control policy of HDP can be iterated to the DT-HJB solution and optimal policy. The multivariate 0-order T–S (Tanaka–Sugeno) fuzzy system using triangle MFs is proven as a universal approximator, to guarantee the convergence of the Fuzzy-HDP mechanism. Some simulations are implemented to observe the performance of the proposed method both in mathematical solution and practical issue. It is concluded that Fuzzy-HDP outperforms traditional optimal control in more complex systems.

1. Introduction

Dynamic programming is deemed as an effective method to solve nonlinear stochastic dynamic problems using the Bellman equation [1–4]. The Bellman equation was proposed by Bellman in 1953 [5]. The access to solve nonlinear dynamic problems is to find the solution of the Hamilton–Jacobi–Bellman (HJB) equation. However, because of "the curse of dimensionality" of computation, it is difficult to solve the HJB equation at the most time. In recent years, adaptive/approximate dynamic programming (ADP) was proposed to solve the HJB equation and attracted a lot attention from many researchers [6–14]. The main concept of ADP is to approach the solution by iterations one by one step. Not giving directly the solution of an optimal control problem, ADP iterates the cost function or value function until the function converges to the HJB solution and the optimal control policy is generated. Generally, ADP approaches are classified into several categories [15,16]: "heuristic dynamic programming (HDP), action-dependent HDP (ADHDP; note the prefix "action-dependent" (AD) used hereafter), also known as Q-learning, dual heuristic dynamic programming (DHP), ADDHP, globalized DHP (GDHP), and ADGDHP". HDP is one kind of intuitive expression of ADP, because it directly constructs the approximators of the value function.

The convergence analysis of ADP has been studied a lot. In the linear discrete-time quadratic optimal control problem, Landelius [17] proved that using HDP, DHP, ADHDP and ADDHP is equivalent to iterating on the Algebraic Riccati equation. In [18,19], Al-Tamimi et al. gave a rigorous convergence proof of HDP algorithm solving the value function of the HJB in nonlinear discrete-time optimal control problems.

To avoid "the curse of dimensionality" of computation, a lot of methods are applied to ADP to approximate the value function and control policy, such as the neural network (NN) [7,19], the polynomial function [20] and the radial basis function (RBF) [21]. Fuzzy systems have been proved as universal approximators [22–24]. They have been studied extensively in theoretical researches and applied widely in practical implementations, because they do not require the attention to accurate input–output relations and are similar to the human thinking and natural language [25–27]. And a lot of works present satisfying performance combining ADP with fuzzy system, for example [28].

Our previous works mostly focused on the practical application of the integration of the fuzzy system and on-line ADP algorithms [29,30]. The results showed good performance with satisfactory training trials and success rate, together with better robustness than other methods. However, the theoretical analysis of these methods is not given, which motivates our research.

This paper pays attention to the implementation of the multivariate 0-order T–S fuzzy system to approximate the HJB solution and control policy on the basis of the HDP algorithm. Moreover, the universal approximation property of the fuzzy system is presented. Furthermore, both the linear system and more complicated nonlinear

☆This work was supported partly by National Natural Science Foundation of China (Nos. 61273136, 61034002), and Beijing Natural Science Foundation (No. 4122083).

* Corresponding author. Tel.: +86 13683277856; fax: +86 10 8261 9580.
E-mail addresses: zyh7716155@163.com (Y. Zhu), dongbin.zhao@ia.ac.cn (D. Zhao), derongliu@gmail.com (D. Liu).

http://dx.doi.org/10.1016/j.neucom.2013.11.055
0925-2312/

systems are simulated using the method. In Section 2, a general description of HDP algorithm and convergence analysis is shown. In Section 3, 0-order T–S fuzzy systems using triangle membership functions are presented as the approximators of the cost function and control policy of HDP, based on the approximation analysis. In Section 4, some applications are simulated using the presented fuzzy-HDP algorithm. In the end, the conclusion is given.

2. HDP algorithm

2.1. Background

The nonlinear discrete-time dynamic system studied in this paper is formulated as

$$x_{k+1} = f(x_k) + g(x_k)u(x_k) \tag{1}$$

where x is the state variable, u is the input variable, k is the time index and $x \in \mathbf{R}^n$, $f(x) \in \mathbf{R}^n$, $g(x) \in \mathbf{R}^{n\times m}$, $u(x) \in \mathbf{R}^m$. Here, we only consider this kind of system which has an equilibrium state at $x=0$, namely $f(0)=0$ and $g(x)=0$ and that $f(x)$ and $g(x)$ are continuous for $\forall x_k$. Besides, the system is also a stabilized system on the state set $\mathbf{\Omega} \subseteq \mathbf{R}^n$, which means that there exists a control policy for all initial state $x_0 \in \mathbf{\Omega}$, x_k can be controlled to zero as k goes infinity. An infinite-horizon cost function is defined as

$$V(x_k) = \sum_{n=k}^{\infty} (x_n^T Q x_n + u^T(x_n) R u(x_n)) \tag{2}$$

and the objective is to design a control policy which can minimize $V(x_k)$. The matrixes Q and R are all positive definite. So (2) can also be rewritten as

$$\begin{aligned} V(x_k) &= \sum_{n=k}^{\infty} (x_n^T Q x_n + u^T(x_n) R u(x_n)) \\ &= x_k^T Q x_k + u_k^T R u_k + \sum_{n=k+1}^{\infty} (x_n^T Q x_n + u^T(x_n) R u(x_n)) \\ &= x_k^T Q x_k + u_k^T R u_k + V(x_{k+1}). \end{aligned} \tag{3}$$

According to Bellman's optimality principle [23], the optimal value function $V^*(x_k)$ satisfies the DT-HJB equation and can be formulated as

$$V^*(x_k) = \min_{u_k} \{x_k^T Q x_k + u_k^T R u_k + V^*(x_{k+1})\} \tag{4}$$

and the optimal control policy is

$$u_k^* = \arg\min_{u_k} \{x_k^T Q x_k + u_k^T R u_k + V^*(x_{k+1})\} \tag{5}$$

2.2. HDP algorithm

The HDP algorithm uses the iteration method to approximate the value function and the control policy gradually to the DT-HJB solution and optimal control policy one by one step. Suppose that during one certain iteration, the value function $V^{(i)}(x)$ and the control policy $u^{(i)}(x)$ are generated, then the next iteration value function can be derived by

$$\begin{aligned} V^{(i+1)}(x_k) &= x_k^T Q x_k + u^{(i)T}(x_k) R u^{(i)}(x_k) + V^{(i)}(x_{k+1}) \\ &= x_k^T Q x_k + u^{(i)T}(x_k) R u^{(i)}(x_k) \\ &\quad + V^{(i)}(f(x_k) + g(x_k) u^{(i)}(x_k)) \end{aligned} \tag{6}$$

and the new control policy is updated by

$$\begin{aligned} u^{(i+1)}(x_k) &= \arg\min_u (x_k^T Q x_k + u^T R u + V^{(i+1)}(x_{k+1})) \\ &= \arg\min_u (x_k^T Q x_k + u^T R u \\ &\quad + V^{(i+1)}(f(x_k) + g(x_k) u)) \end{aligned} \tag{7}$$

Usually, there is an initial value function $V^{(0)}(x) = 0$.

In (6) and (7), the index i presents the iteration step for the value function and the control policy. It is clear to see that the value function starts from an arbitrary condition and it can be updated to approach the optimal value function gradually in the end through iterations one by one step. Next, a convergence proof is shown that HDP can approximate the DT-HJB solution and control policy, namely $V^{(i)} \to V^*$ and $u^{(i)} \to u^*$ as $i \to \infty$.

2.3. Convergence analysis

Now we introduce the convergence property of HDP by Al-Tamimi et al., while the detailed theoretical proof can be found in [18,19]. In order to prove that the iterations (6) and (7) in HDP can converge to DT-HJB solution and optimal control policy, some lemmas are needed.

Lemma 1. *[18,19] Suppose the system (1) has nonlinear discrete-time dynamic model. $V^{(i)}$ and $u^{(i)}$ are defined as (6) and (7) and $V^{(0)} = 0$.*

1. *Define any arbitrary sequence of control policies μ, and iterations $\Lambda^{(i)}$ as*

$$\Lambda^{(i+1)}(x_k) = x_k^T Q x_k + \mu^T(x_k) R \mu(x_k) + \Lambda^{(i)}(f(x_k) + g(x_k)\mu(x_k)) \tag{8}$$

   If $V^{(0)}(x_k) = \Lambda^{(0)}(x_k) = 0$, then $V^{(i)}(x_k) \le \Lambda^{(i)}(x_k)$, $\forall i$.
2. *If the system is controllable, then define any stabilizing and admissible control policy η. Let $Y(x_k)$ is defined by*

$$Y(x_k) = \sum_{n=k}^{\infty} (x_n^T Q x_n + \eta^T(x_n) R \eta(x_n)). \tag{9}$$

   Then $0 \le V^{(i)}(x_k) \le Y(x_k)$, $\forall i$.
3. *If the system is controllable and (4) has the solution $V^*(x_k)$, then define any stabilizing and admissible control policy η and $Y(x_k)$ in (9). In this way, $0 \le V^{(i)}(x_k) \le V^*(x_k) \le Y(x_k)$, $\forall i$.*
4. *Define $V^{(i)}$ and $u^{(i)}$ as (6) and (7). If the initial value $V^{(0)}(x_k) = 0$, then the sequence $\{V^{(0)}, V^{(1)}, V^{(2)}, \ldots\}$ is a nondecreasing sequence, namely $V^{(i)}(x_k) \le V^{(i+1)}(x_k)$, $\forall i$.*

On the basis of Lemma 1, now the result is deduced.

Theorem 1. *[18,19] Consider the nonlinear discrete-time dynamic system (1) and define $V^{(i)}$ and $u^{(i)}$ as (6) and (7). Suppose that the system is controllable and (4) has the solution $V^*(x_k)$. If $V^{(0)}(x_k) = 0$, then the sequence $\{V^{(i)}\}$ converge to the DT-HJB solution (4), namely $V^{(i)} \to V^*$, $u^{(i)} \to u^*$, as $i \to \infty$.*

3. Fuzzy-HDP

In the above two sections, HDP algorithm and its convergence analysis is introduced. It has been proved that $V^{(i)}$ and $u^{(i)}$ defined by (6) and (7) in HDP can converge to DT-HJB solution (4) and optimal control policy (5) [18,19]. However, $V^{(i)}$ and $u^{(i)}$ are too complex to generate definite expressions mapping from x. So some approximation methods are applied to HDP to approximate $V^{(i)}$ and $u^{(i)}$. In this paper, 0-order T–S fuzzy systems using triangle MFs are adopted as the approximators of the value function and control policy of HDP. A lot of researches have proved that fuzzy system is a universal approximator to any functions. Wang and Mendel [23] gave an approximating proof of the univariate T–S fuzzy system using triangle MFs. In this section, the theoretical analysis of multivariate T–S fuzzy system and the detailed description of fuzzy-HDP are presented.

3.1. 0-Order multivariate T–S fuzzy system

First of all, some symbol definitions are listed in the following.

$x = [x_1, x_2, \ldots, x_n]^T$ system input variable

$A_1^{j_1}, A_2^{j_2}, \ldots, A_n^{j_n} \in F(\mathbf{U})$	fuzzy sets in the domain of discourse **U**, corresponding to $x_1, x_2, \ldots, x_n$ respectively
$N_1, N_2, \ldots, N_n$	each number of fuzzy sets for $x_1, x_2, \ldots, x_n$, namely $j_1 = 1, \ldots, N_1$, $j_2 = 1, \ldots, N_2, \ldots, j_n = 1, \ldots, N_n$
$M = N_1 N_2 \cdots N_n$	total number of fuzzy rules
$m \leftrightarrow \{j_1, j_2, \ldots, j_n\}$	for the index m of each rules R_m, there exists a relative sequence $\{j_1, j_2, \ldots, j_n\}$

In this paper, triangle MFs are adopted, and denoted by

$$\mu_{A_l^{j_l}}(x_l) = \begin{cases} (x_l - X_l^{j_l-1})/(X_l^{j_l} - X_l^{j_l-1}), & x_l \in [X_l^{j_l-1}, X_l^{j_l}] \\ (X_l^{j_l+1} - x_l)/(X_l^{j_l+1} - X_l^{j_l}), & x_l \in [X_l^{j_l}, X_l^{j_l+1}] \\ 0 & \text{other conditions} \end{cases} \tag{10}$$

where $X_l^{j_l-1}$, $X_l^{j_l}$, $X_l^{j_l+1}$ are three parameters which determine the shape of triangle MFs and $l = 1, 2, \ldots, n$ is the index of the input variables. In order to simplify the symbol, $\mu_{A_l^{j_l}}(x_l)$ is replaced with $A_l^{j_l}(x_l)$. Fig. 1 shows a typical shape of triangle MFs.

It is obvious that triangle MFs have such properties

1. $A_l^{j_l}(X_l^{j_l}) = 1, \ \forall l, j_l$
2. $\forall x_l \in [X_l^{J_l}, X_l^{J_l+1}], \ \sum_{j_l=1}^{N_l} A_l^{j_l}(x_l) = A_l^{J_l}(x_l) + A_l^{J_l+1}(x_l) = 1.$

For $\forall x_l \in [X_l^{J_l}, X_l^{J_l+1}]$, J_l represents one certain fuzzy set. In other words, for any x_l, it must exist in the overlapping region of the available scope of two neighboring fuzzy sets, which can be indicated by $A_l^{J_l}$ and $A_l^{J_l+1}$ fuzzy sets. The second property means that for any input variable, there are at most two triangle MFs activated and the sum of the memberships is 1, while the other fuzzy memberships are zero.

Now assume that a multi-input single-output system can be described by 0-order T–S fuzzy model which means that the fuzzy rules have single value output. Then the fuzzy rule can be described as

$$R_m: \quad \text{If } x_1 \text{ is } A_1^{j_1}, x_2 \text{ is } A_2^{j_2}, \ldots, x_n \text{ is } A_n^{j_n}, \quad \text{then } y_m = Y_m. \tag{11}$$

If the system output adopts weighted average method, then the system output can be calculated by

$$F(x) = \sum_{m=1}^{M} w_m(x) Y_m \Big/ \sum_{m=1}^{M} w_m(x) \tag{12}$$

$$w_m(x) = A_1^{j_1}(x_1) A_2^{j_2}(x_2) \cdots A_n^{j_n}(x_n). \tag{13}$$

Next, a new theorem is presented that the 0-order multivariate T–S fuzzy system using triangle MFs can approach a given function in any precision.

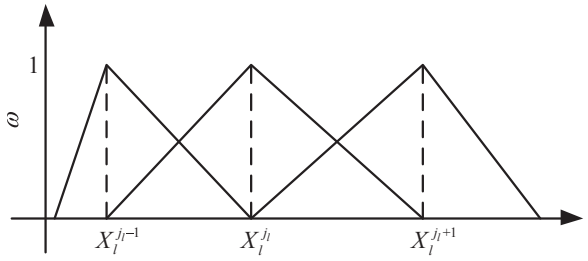


Fig. 1. Triangle MFs distribution graphs. ω indicates membership of different MFs.

3.2. Approximability of 0-order multivariate T–S fuzzy system

Theorem 2. *Assume an arbitrary function $f(x)$, $x = [x_1, x_2, \ldots, x_n]^T$. The second derivative of $f(x)$ is bounded, namely $|f''(x)| \le M$. If some data $\{(X_1^{j_1}, X_2^{j_2}, \ldots, X_n^{j_n})\}_{j_1=1,\ldots N_1, j_2=1,\ldots N_2, \ldots j_n=1,\ldots N_n}$ are sampled and their results $f(X_1^{j_1}, X_2^{j_2}, \ldots, X_n^{j_n})$ are known, then design a 0-order multivariate T–S fuzzy system which uses (10) and (11) to construct the triangle MFs and fuzzy rules. The output of the fuzzy system is $Y_m = f(X_1^{j_1}, X_2^{j_2}, \ldots, X_n^{j_n})$. Then the approximation error between the fuzzy system output $F(x)$ and $f(x)$ is infinitesimal of the square of the fuzzy interval, namely $|F(x) - f(x)| \propto O(\rho^2)$, where $\rho = \max\left\{\sqrt{(X_1^{j_1+1} - X_1^{j_1})^2 + (X_2^{j_2+1} - X_2^{j_2})^2 + \cdots + (X_n^{j_n+1} - X_n^{j_n})^2}\right\}$, and $j_1 = 1, \ldots, N_1 - 1$, $j_2 = 1, \ldots, N_2 - 1, \ldots, j_n = 1, \ldots, N_n - 1$.*

Proof. Assume $x_1 \in [X_1^{J_1}, X_1^{J_1+1}]$, $x_2 \in [X_2^{J_2}, X_2^{J_2+1}], \ldots, x_n \in [X_n^{J_n}, X_n^{J_n+1}]$. Then based on the second property of triangle MFs and (13), we have

$$\begin{aligned} \sum_{m=1}^{M} w_m(x) &= \sum_{j_1=1}^{N_1} \sum_{j_2=1}^{N_2} \cdots \sum_{j_n=1}^{N_n} A_1^{j_1}(x_1) A_2^{j_2}(x_2) \cdots A_n^{j_n}(x_n) \\ &= \sum_{j_1=J_1}^{J_1+1} \sum_{j_2=J_2}^{J_2+1} \cdots \sum_{j_n=J_n}^{J_n+1} A_1^{j_1}(x_1) A_2^{j_2}(x_2) \cdots A_n^{j_n}(x_n) \\ &= \underbrace{(A_1^{J_1}(x_1) + A_1^{J_1+1}(x_1))}_{1} \cdot \sum_{j_2=J_2}^{J_2+1} \cdots \sum_{j_n=J_n}^{J_n+1} A_2^{j_2}(x_2) \cdots A_n^{j_n}(x_n) \\ &= \sum_{j_2=J_2}^{J_2+1} \cdots \sum_{j_n=J_n}^{J_n+1} A_2^{j_2}(x_2) \cdots A_n^{j_n}(x_n) \\ &\vdots \\ &= 1. \end{aligned} \tag{14}$$

So (12) can be rewritten as

$$\begin{aligned} F(x) &= \sum_{m=1}^{M} w_m(x) Y_m \\ &= \sum_{j_1=J_1}^{J_1+1} \sum_{j_2=J_2}^{J_2+1} \cdots \sum_{j_n=J_n}^{J_n+1} (A_1^{j_1}(x_1) A_2^{j_2}(x_2) \cdots A_n^{j_n}(x_n) Y_m) \\ &= \sum_{j_1=J_1}^{J_1+1} \sum_{j_2=J_2}^{J_2+1} \cdots \sum_{j_n=J_n}^{J_n+1} \begin{pmatrix} A_1^{j_1}(x_1) A_2^{j_2}(x_2) \cdots A_n^{j_n}(x_n) \\ \cdot f(X_1^{j_1}, X_2^{j_2}, \ldots, X_n^{j_n}) \end{pmatrix} \end{aligned} \tag{15}$$

Besides, apply the second property of triangle MFs

$$\begin{aligned} f(x) &= 1 \cdot f(x) \\ &= (A_1^{J_1}(x_1) + A_1^{J_1+1}(x_1)) \cdot f(x) \\ &= (A_1^{J_1}(x_1) + A_1^{J_1+1}(x_1)) \cdot (A_2^{J_2}(x_2) + A_2^{J_2+1}(x_2)) \cdot f(x) \vdots \\ &= (A_1^{J_1}(x_1) + A_1^{J_1+1}(x_1))(A_2^{J_2}(x_2) + A_2^{J_2+1}(x_2)) \cdot (\cdots) \cdot \\ &\quad \cdot (A_n^{J_n}(x_n) + A_n^{J_n+1}(x_n)) \cdot f(x) \\ &= \sum_{j_1=J_1}^{J_1+1} \sum_{j_2=J_2}^{J_2+1} \cdots \sum_{j_n=J_n}^{J_n+1} \begin{pmatrix} A_1^{j_1}(x_1) A_2^{j_2}(x_2) \cdots A_n^{j_n}(x_n) \\ \cdot f(x_1, x_2, \ldots, x_n) \end{pmatrix}. \end{aligned} \tag{16}$$

Let (15) subtracts (16) and it follows that

$$F(x) - f(x) = \sum_{j_1=J_1}^{J_1+1} \sum_{j_2=J_2}^{J_2+1} \cdots \sum_{j_n=J_n}^{J_n+1} (A_1^{j_1}(x_1) A_2^{j_2}(x_2) \cdots A_n^{j_n}(x_n) \cdot (f(X_1^{j_1}, X_2^{j_2}, \ldots, X_n^{j_n}) - f(x_1, x_2, \ldots, x_n))). \tag{17}$$

Using Taylor's expansion, expand $f(X_1^{j_1}, X_2^{j_2}, \ldots, X_n^{j_n})$ and $f(x_1, x_2, \ldots, x_n)$ at $f(X_1^{J_1}, X_2^{J_2}, \ldots, X_n^{J_n})$ respectively. Because $|f''(x)| \le M$, so it has

$$\begin{aligned} f(X_1^{j_1}, X_2^{j_2}, \ldots, X_n^{j_n}) &= f(X_1^{J_1}, X_2^{J_2}, \ldots, X_n^{J_n}) + (X_1^{j_1} - X_1^{J_1}) f'_{x_1}(X_1^{J_1}, X_2^{J_2}, \ldots, X_n^{J_n}) \\ &\quad + \cdots + (X_n^{j_n} - X_n^{J_n}) f'_{x_n} \\ &\quad \times (X_1^{J_1}, X_2^{J_2}, \ldots, X_n^{J_n}) + O(\rho^2), \end{aligned} \tag{18}$$

and

$$f(x_1,x_2,\ldots,x_n) = f(X_1^{J_1},X_2^{J_2},\ldots,X_n^{J_n})+(x_1-X_1^{J_1})f'_{x_1}(X_1^{J_1},X_2^{J_2},\ldots,X_n^{J_n}) + \cdots+(x_n-X_n^{J_n})f'_{x_n}(X_1^{J_1},X_2^{J_2},\ldots,X_n^{J_n})+O(\rho^2). \quad (19)$$

Let (18) subtracts (19), it derives

$$f(X_1^{j_1},X_2^{j_2},\ldots,X_n^{j_n})-f(x_1,x_2,\ldots,x_n) = (X_1^{j_1}-x_1)f'_{x_1}(X_1^{J_1},X_2^{J_2},\ldots,X_n^{J_n})+\cdots+(X_n^{j_n}-x_n)f'_{x_n} \times(X_1^{J_1},X_2^{J_2},\ldots,X_n^{J_n})+O(\rho^2). \quad (20)$$

Substitute (20) into (17), one can obtain

$$F(x)-f(x) = \sum_{j_1=J_1}^{J_1+1}\sum_{j_2=J_2}^{J_2+1}\cdots\sum_{j_n=J_n}^{J_n+1} \times\left(\begin{array}{l} A_1^{j_1}(x_1)A_2^{j_2}(x_2)\cdots A_n^{j_n}(x_n) \\ \cdot\left(\begin{array}{l}(X_1^{j_1}-x_1)f'_{x_1}(X_1^{J_1},X_2^{J_2},\ldots,X_n^{J_n}) \\ +\cdots+(X_n^{j_n}-x_n)f'_{x_n}(X_1^{J_1},X_2^{J_2},\ldots,X_n^{J_n}) \\ +O(\rho^2)\end{array}\right)\end{array}\right). \quad (21)$$

Now, just consider the coefficient of $f'_{x_1}(X_1^{J_1},X_2^{J_2},\ldots,X_n^{J_n})$ and the coefficient is

$$\begin{aligned}&\sum_{j_1=J_1}^{J_1+1}\sum_{j_2=J_2}^{J_2+1}\cdots\sum_{j_n=J_n}^{J_n+1}(A_1^{j_1}(x_1)A_2^{j_2}(x_2)\cdots A_n^{j_n}(x_n)\cdot(X_1^{j_1}-x_1)) \\ &= \sum_{j_1=J_1}^{J_1+1}A_1^{j_1}(x_1)(X_1^{j_1}-x_1)\cdot\underbrace{\left(\sum_{j_2=J_2}^{J_2+1}\cdots\sum_{j_n=J_n}^{J_n+1}(A_1^{j_1}(x_1)A_2^{j_2}(x_2)\cdots A_n^{j_n}(x_n))\right)}_{1} \\ &= A_1^{J_1}(x_1)(X_1^{J_1}-x_1)+A_1^{J_1+1}(x_1)(X_1^{J_1+1}-x_1) \\ &= \frac{X_1^{J_1+1}-x_1}{X_1^{J_1+1}-X_i^{J_i}}(X_1^{J_1}-x_1)+\frac{x_i-X_i^{J_i}}{X_1^{J_1+1}-X_i^{J_i}}(X_1^{J_1+1}-x_1) \\ &= 0. \end{aligned} \quad (22)$$

Likewise, the coefficients of the other first derivative are also zero. In this way, (21) can be rewritten as

$$\begin{aligned}F(x)-f(x) &= \sum_{j_1=J_1}^{J_1+1}\sum_{j_2=J_2}^{J_2+1}\cdots\sum_{j_n=J_n}^{J_n+1}O(\rho^2) \\ &= O(\rho^2).\end{aligned} \quad (23)$$

The proof is completed. □

It is obvious that the approximation error between $F(x)$ and $f(x)$ is related with the fuzzy interval ρ. The smaller the interval is selected, the more accurately $F(x)$ approaches $f(x)$. In this way, the adoption of fuzzy interval, in other words, the number of fuzzy sets is determined by the demand of approximating accuracy.

3.3. Fuzzy-HDP algorithm

It has been proved that HDP can converge to DT-HJB solution and the 0-order multi-variate T–S fuzzy systems can approach the value function and control policy of HDP. Now, a detailed description of fuzzy-HDP algorithm is presented here.

First of all, a group of proper fuzzy sets $A_1^{j_1},A_2^{j_2},\ldots,A_n^{j_n}\in F(\mathbf{U})$, $j_1=1,\ldots,N_1$, $j_2=1,\ldots,N_2,\ldots,j_n=1,\ldots,N_n$, are adopted for input variables. Then these two equations are used to approach $V^{(i)}$ and $u^{(i)}$:

$$\hat{V}^{(i)}(x_k) = \sum_{m=1}^{M}w_m(x_k)Y_{V}{}_m^{(i)}\Big/\sum_{m=1}^{M}w_m(x_k) \quad (24)$$

$$\hat{u}^{(i)}(x_k) = \sum_{m=1}^{M}w_m(x_k)Y_{u}{}_m^{(i)}\Big/\sum_{m=1}^{M}w_m(x_k) \quad (25)$$

where $w_m(x_k)$ can be calculated using (10) and (13) and $Y_V^{(i)}$ and $Y_u^{(i)}$ are the fuzzy rules of value function and control policy. Define the target value function

$$\begin{aligned}d(x_k,Y_V^{(i)}) &= x_k^TQx_k+\hat{u}^{(i)T}(x_k)R\hat{u}^{(i)}(x_k)+\hat{V}^{(i)}(x_{k+1}) \\ &= x_k^TQx_k+\hat{u}^{(i)T}(x_k)R\hat{u}^{(i)}(x_k) \\ &\quad+\sum_{m=1}^{M}w_m(x_{k+1})Y_{V}{}_m^{(i)}\Big/\sum_{m=1}^{M}w_m(x_{k+1}),\end{aligned} \quad (26)$$

where $x_{k+1}=f(x_k)+g(x_k)\hat{u}^{(i)}(x_k)$. So the target is to find out $Y_V(i+1)$ which minimize the difference between $\hat{V}^{(i+1)}(x_k)$ in (24) and $d(x_k,Y_V^{(i)})$ in (26). It is implemented over a compact set $\mathbf{\Omega}$ to calculate the variance

$$\begin{aligned}Y_V^{(i+1)} &= \arg\min_{Y_V^{(i+1)}}\left\{\int_{\Omega}\left|\left(\sum_{m=1}^{M}w_m(x_k)Y_{V}{}_m^{(i+1)}\Big/\sum_{m=1}^{M}w_m(x_k)\right)\right.\right. \\ &\quad\left.\left.-d(x_k,Y_V^{(i)})\right|^2dx_k\right\}.\end{aligned} \quad (27)$$

Moreover, it is obvious that in (24) it is a first-order polynomial on $Y_V^{(i+1)}$, so linear least-squares method can be used to solve (27).

Meanwhile, the policy parameters are updated as

$$Y_u^{(i)} = \arg\min_{\theta}(x_k^TQx_k+\tilde{u}^T(x_k,\theta)R\tilde{u}(x_k,\theta)+\hat{V}^{(i)}(x_{k+1})) \quad (28)$$

where $\tilde{u}(x_k,\theta)=\sum_{m=1}^{M}w_m(x_k)\theta_m/\sum_{m=1}^{M}w_m(x_k)$ and $x_{k+1}=f(x_k)+g(x_k)\tilde{u}(x_k,\theta)$. In (28) there exists no explicit relation, so a gradient descent method is adopted to update $Y_u^{(i)}$ by

$$Y_u^{(i,j+1)} = Y_u^{(i,j)}-\alpha\frac{\partial}{\partial Y_u^{(i,j)}}(x_k^TQx_k+\hat{u}^{(i,j)T}(x_k)R\hat{u}^{(i,j)}(x_k)+\hat{V}^{(i)}(x_{k+1})) \quad (29)$$

where α is a positive small step size and $x_{k+1}=f(x_k)+g(x_k)\hat{u}^{(i,j)}(x_k)$. Eq. (29) is implemented over the whole compact set $\mathbf{\Omega}$ and $Y_u^{(i,j)}\to Y_u^{(i)}$ as $j\to\infty$.

A flow chart of fuzzy-HDP algorithm is shown in Fig. 2. In the next section, some examples are implemented using fuzzy-HDP and the performance is observed.

4. Examples using Fuzzy-HDP

4.1. Linear system case

Consider a linear system with the following dynamic model:

$$x_{k+1}=Ax_k+Bu_k \quad (30)$$

where $x_k=[x_k(1),x_k(2)]^T$, $A=\left[\begin{smallmatrix}0 & 0.1\\ 0.3 & -1\end{smallmatrix}\right]$ and $B=\left[\begin{smallmatrix}0\\1\end{smallmatrix}\right]$. Besides, let Q and R in (2) be

$$Q=\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\quad R=1.$$

For linear systems, the DT-HJB solution can be simply formulated by a quadratic form of the system state, namely $V^*(x_k)=x_k^TPx_k$. The matrix P is the solution of Algebraic Riccati Equation (ARE), which can be calculated using linear-quadratic regulator (LQR) method. Here, the solution is

$$P=\begin{bmatrix}1.05600 & -0.18880\\ -0.18880 & 1.64689\end{bmatrix}.$$

Now, fuzzy-HDP is applied to this linear system. The training sets are $x_1\in[-2,2]$, $x_2\in[-1,1]$ and 41×21 points are uniformly sampled in the training set. The number of fuzzy sets for each state variable is both 11, and the fuzzy sets are all uniformly distributed over the training sets. After iterations of fuzzy-HDP, the fuzzy approximator of value function, $\hat{V}(x_k)$, is generated. The value calculated by P from ARE solution and $\hat{V}(x_k)$ is shown in Fig. 3. It is obvious that the value function generated by fuzzy-HDP is almost the same as the DT-HJB solution.

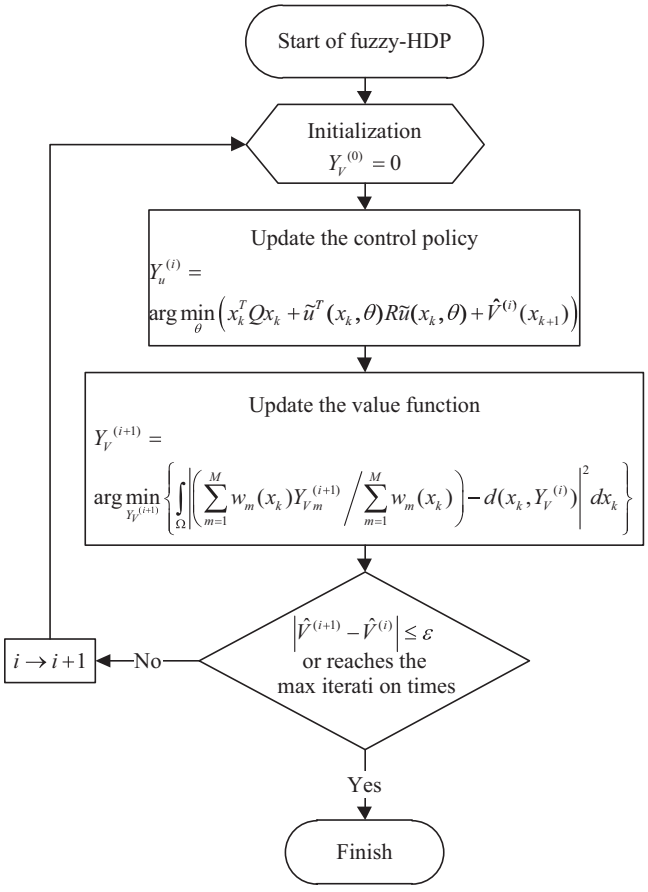


Fig. 2. A flow chart of the fuzzy-HDP algorithm.

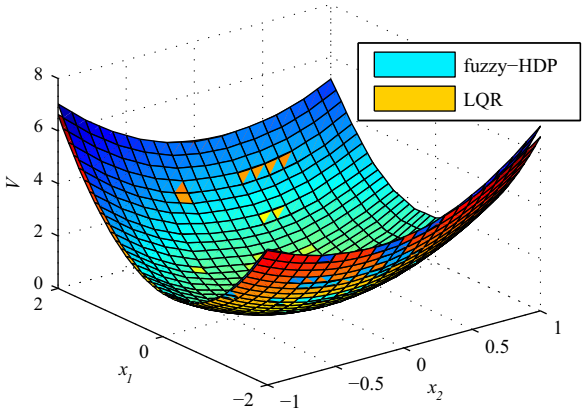


Fig. 3. The value function of fuzzy-HDP and DT-HJB solution.

4.2. Nonlinear system case

Now consider the following nonlinear system:

$$x_{k+1} = f(x_k) + g(x_k)u_k \tag{31}$$

where

$$x_k = [x_k(1),\ x_k(2)]^T, \quad f(x_k) = \begin{bmatrix} \frac{x_k^3(2) + x_k(2)}{1 + 2x_k^2(2)} \\ \frac{x_k(1) + x_k(2)}{1 + x_k^2(1)} \end{bmatrix}$$

and

$$g(x_k) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The training sets are $x_1 \in [-1, 1]$, $x_2 \in [-1, 1]$ and 201×201 points are uniformly sampled in the training sets. For every input variable, there are 5 fuzzy sets uniformly distributed over the scope of the training set, as shown in Fig. 4.

Using the fuzzy-HDP algorithm, the fuzzy rules of the value function and the control policy converge to a group of values, which are listed in Tables 1 and 2 respectively. Fig. 5 shows the state trajectories of the system with the trained control policy from the beginning state $x_1 = 1$, $x_2 = -1$. For comparison, linear-quadratic regulator (LQR) method is applied to the system. From Fig. 5, the state trajectories using fuzzy-HDP approach to zero faster than LQR. It is because that LQR is implemented around zero point which means that it is suitable just around zero point. However, if the state is far away from zero point, LQR is no more the optimal policy. But, fuzzy-HDP generates the control policy which is suitable for the whole state space and controls the state approach to zero faster. Figs. 6 and 7 show the trajectories of value function and control input of fuzzy-HDP respectively. It is clear to see that the value function and the control policy using fuzzy-HDP can approach the DT-HJB solution and optimal policy, and are capable of controlling the state of nonlinear systems to zero.

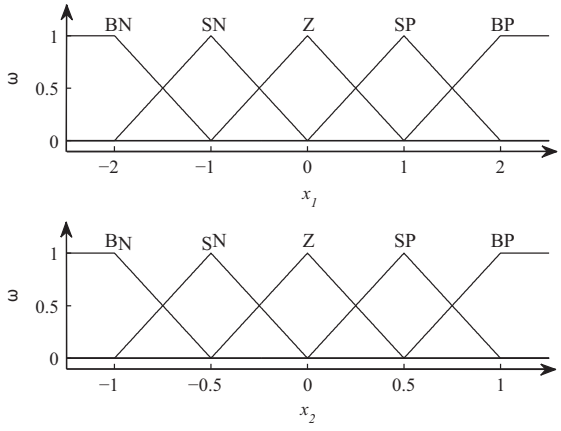


Fig. 4. Membership functions of state variables.

Table 1
The value of fuzzy rules Y_V after fuzzy-HDP algorithm.

	BN(x_2)	SN(x_2)	Z(x_2)	SP(x_2)	BP(x_2)
BN(x_1)	3.5271	2.0137	1.2046	1.5274	2.5495
SN(x_1)	3.3130	3.3130	0.3525	0.6408	1.8606
Z(x_1)	2.5449	0.6499	0	0.6476	2.5430
SP(x_1)	1.8620	0.6447	0.3495	1.3896	3.3114
BP(x_1)	2.5513	1.5294	1.2061	2.0110	3.5252

Table 2
The value of fuzzy rules Y_u after fuzzy-HDP algorithm.

	BN(x_2)	SN(x_2)	Z(x_2)	SP(x_2)	BP(x_2)
BN(x_1)	1.0029	0.7544	0.5070	0.2608	0.01411
SN(x_1)	1.2238	0.8320	0.4309	0.01730	−0.3898
Z(x_1)	1.0188	0.5000	0.0000486	−0.5000	−1.0189
SP(x_1)	0.3904	−0.01865	−0.4269	−0.8319	−1.2229
BP(x_1)	−0.01414	−0.2607	−0.5074	−0.7546	−1.0031

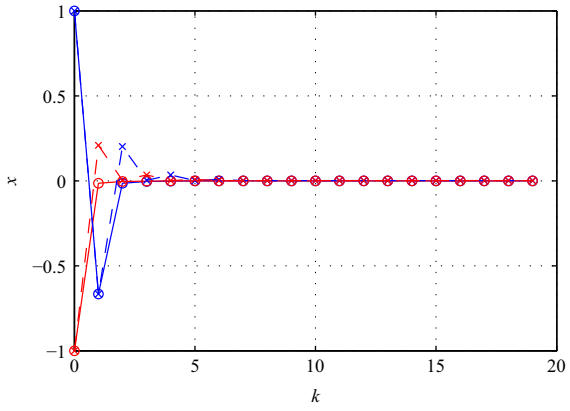


Fig. 5. State trajectory of the system with the trained fuzzy-HDP control policy compared with LQR. The blue solid line and red solid line with '○' represent x_1 and x_2 using fuzzy-HDP, while the blue dashed line and red dashed line with '×' represent x_1 and x_2 using LQR. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

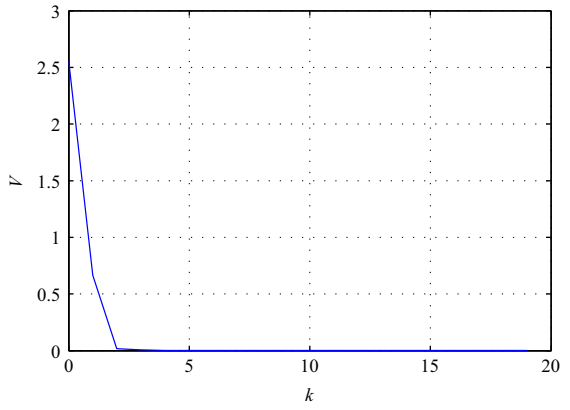


Fig. 6. Trajectory of value function using fuzzy-HDP.

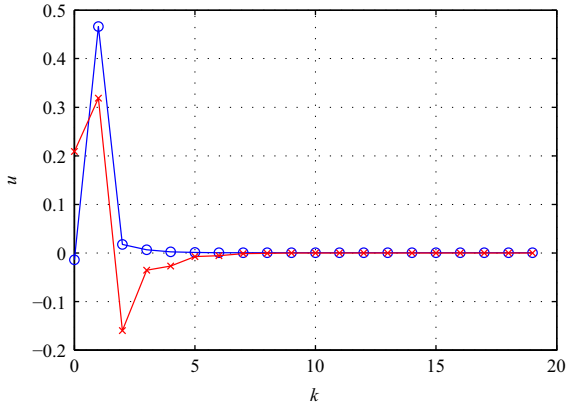


Fig. 7. Trajectory of control using fuzzy-HDP compared with LQR. The blue line with '○' represents fuzzy-HDP, while the red line with 'times' represents LQR. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

4.3. Single inverted pendulum case

Here, a more complex and more practical example is simulated —single inverted pendulum system. The system dynamical model can be formulated by

$$\dot{x}_1 = x_2, \dot{x}_2 = f(x) + g(x)u \tag{32}$$

where

$$f(x) = \frac{g \sin x_1 - mlx_2^2 \cos x_1 \sin x_1/(m_c+m)}{l(4/3 - m\cos^2 x_1/(m_c+m))}$$

and

$$g(x) = \frac{\cos x_1/(m_c+m)}{l(4/3 - m\cos^2 x_1/(m_c+m))}.$$

x_1 and x_2 are the angle and angular velocity of pendulum respectively. The other symbols are defined as

g	9.8 m/s^2, the acceleration due to gravity;
m_c	1.0 kg, the mass of cart;
m	0.1 kg, the mass of pole;
l	0.5 m, the half-pole length;
u	the control input.

The training sets are $x_1 \in [-5^\circ, 5^\circ]$, $x_2 \in [-20^\circ/s, 20^\circ/s]$ and 201×201 points are uniformly sampled in the training set. The fuzzy sets are adopted the same as above. Using the fuzzy-HDP algorithm, the fuzzy rules of the value function and the control policy converge to a group of values, which are listed in Tables 3 and 4 respectively.

Fig. 8 shows the state trajectories of the system with the trained control policy from the beginning state $x_1 = 2^\circ$, $x_2 = -1^\circ/s$. For comparison, linear-quadratic regulator (LQR) method is also applied to the system. The linearization of the nonlinear dynamic system is conducted at 0 point. Fig. 8 shows the result of using fuzzy-HDP better than the result of using LQR, with the fact that the former takes less time to stabilize the pendulum angle than the latter. Mostly it is in that the control policy generated by LQR method is calculated under the linear dynamic matrix. But the linear dynamic matrix is the linearization of nonlinear dynamic system at zero point. So the LQR control policy is the optimal policy around the zero point while it is not far away from the zero point. The value function and the control policy generated by fuzzy-HDP are always the approximators of the DT-HJB solution and the optimal policy. Figs. 9 and 10 show the trajectories of the value function and the control input respectively. From the results, it is shown that even a complex and practical single inverted pendulum can be controlled using fuzzy-HDP and has a very excellent performance. The whole system is balanced to the equilibrium point. The value function output is also reduced to zeros gradually.

Table 3
The value of fuzzy rules Y_V after fuzzy-HDP algorithm.

	BN(x_2)	SN(x_2)	Z(x_2)	SP(x_2)	BP(x_2)
BN(x_1)	19.8032	13.1843	6.4174	2.1362	1.0204
SN(x_1)	14.2141	9.1212	3.6068	0.9264	2.0485
Z(x_1)	8.6050	2.3152	0.1356	2.2788	8.3574
SP(x_1)	2.2022	0.8322	3.5114	9.0415	14.0625
BP(x_1)	0.9789	2.0913	6.3721	13.1611	19.8204

Table 4
The value of fuzzy rules Y_u after fuzzy-HDP algorithm.

	BN(x_2)	SN(x_2)	Z(x_2)	SP(x_2)	BP(x_2)
BN(x_1)	3.3296	3.1705	2.1409	0.9431	−0.2064
SN(x_1)	2.8902	2.7100	1.6631	0.4694	−0.7966
Z(x_1)	2.4692	1.1927	0.000015	−1.1927	−2.3852
SP(x_1)	1.0377	−0.3887	−1.6636	−2.7143	−2.8584
BP(x_1)	0.1766	−0.9632	−2.1407	−3.1686	−3.3377

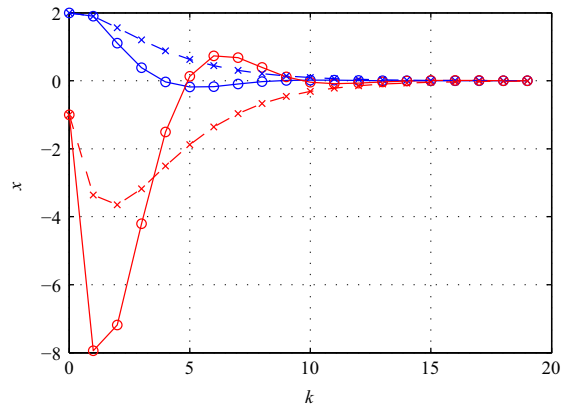


Fig. 8. State trajectory of the system with the trained fuzzy-HDP control policy compared with LQR. The blue solid line and red solid line with '○' represent x_1 and x_2 using fuzzy-HDP, while the blue dashed line and red dashed line with '×' represent x_1 and x_2 using LQR. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

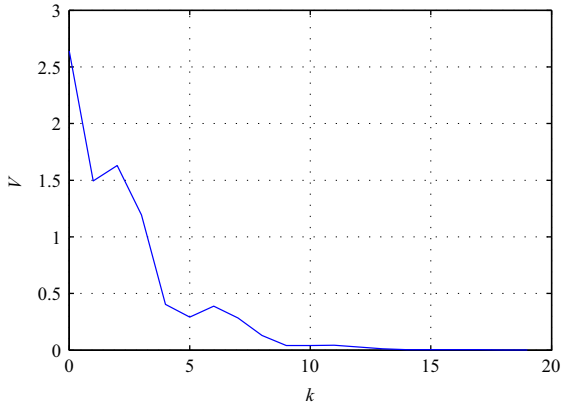


Fig. 9. Trajectory of value function using fuzzy-HDP.

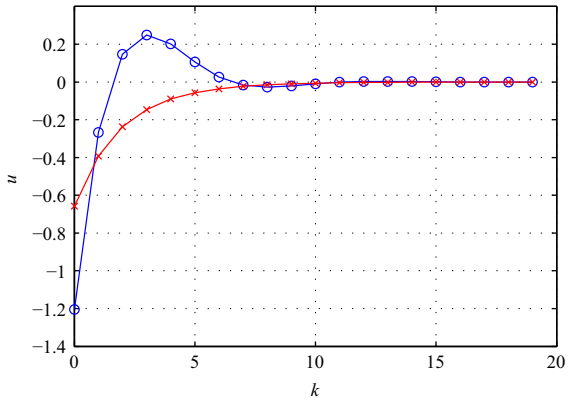


Fig. 10. Trajectory of control using fuzzy-HDP compared with LQR. The blue line with '○' represents fuzzy-HDP, while the red line with '×' represents LQR. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

5. Conclusion

In this paper, a fuzzy-HDP scheme is proposed to solve DT-HJB problems. By iterating, the value function and control policy can converge to the DT-HJB solution and optimal control policy. Besides, 0-order T–S fuzzy systems using triangle MFs are adopted to approximate the value function and the control policy, taking the advantage that fuzzy system has the property of highly nonlinear feature and the basis on human thought and natural language. Meanwhile, some convergence proof of HDP and theoretical analysis of fuzzy system approximation property are presented in this paper. Furthermore, three cases are simulated and the results are compared to LQR. While solving linear dynamic problem, fuzzy-HDP generates the same value function and optimal policy as LQR. For nonlinear system, fuzzy-HDP has a better control performance than LQR.

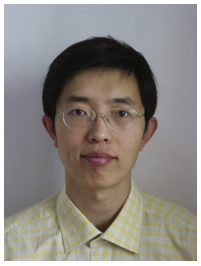
The 0-order T–S fuzzy system is just one kind of fuzzy system with simple structure. For more accurate approximating precision, 1-order T–S or more complex fuzzy systems need our further research.

References

[1] R. Bellman, S. Dreyfus, Applied Dynamic Programming, Princeton University Press, Princeton, NJ, 1962.
[2] T. Borgers, R. Sarin, Learning through reinforcement and replicator dynamics, J. Econ. Theory 77 (1) (1997) 1–17.
[3] J. Dalton, S.N. Balakrishnan, A neighboring optimal adaptive critic for missile guidance, Math. Comput. Model. 23 (1) (1996) 175–188.
[4] J.N. Tsitsiklis, B. Van Roy, An analysis of temporal-difference learning with function approximation, IEEE Trans. Autom. Control 42 (1997) 674–690.
[5] P.J. Werbos, Using adaptive dynamic programming to understand and replicate brain intelligence: the next level design, in: R. Kozma (Ed.), Neurodynamics of Higher-Level Cognition and Consciousness, Springer-Verlag, Berlin, Germany, 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. 38 (4) (2008) 898–900.
[7] J. Si, Y.T. Wang, On-line learning control by association and reinforcement, IEEE Trans. Neural Netw. 12 (2) (2001) 264–276.
[8] D.B. Zhao, J.Q. Yi, D.R. Liu, Particle swarm optimized adaptive dynamic programming, in: Proceedings of the 2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, 2007, pp. 32–37.
[9] T. Li, D.B. Zhao, J.Q. Yi, Heuristic dynamic programming strategy with eligibility traces, in: Proceedings of the American Control Conference, IEEE, Seattle, USA, 2008, pp. 4535–4540.
[10] D.B. Zhao, Y.J. Dai, Z. Zhang, Computational intelligence in urban traffic signal control: a survey, IEEE Trans. Syst. Man Cybern. C 42 (4) (2012) 485–494.
[11] D.B. Zhao, X.R. Bai, F.Y. Wang, J. Xu, W.S. Yu, DHP method for ramp metering of freeway traffic, IEEE Trans. Intell. Transp. Syst. 12 (4) (2011) 990–999.
[12] D. Xu, D.B. Zhao, J.Q. Yi, X.M. Tan, Trajectory tracking control of omnidirectional wheeled mobile manipulators: robust neural network based sliding mode approach, IEEE Trans. Syst. Man Cybern. B: Cybern. 39 (3) (2009) 788–799.
[13] Z. Jiang, Y. Jiang, Robust adaptive dynamic programming for linear and nonlinear systems: an overview, Eur. J. Control 19 (5) (2013) 417–425.
[14] D. Liu, D. Wang, X. Yang, An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs, Inf. Sci. 220 (20) (2013) 331–342.
[15] P.J. Werbos, Approximate dynamic programming for real-time control and neural modeling, in: D.A. White, D.A. Sofge (Eds.), Handbook of Intelligent Control, Van Nostrand Reinhold, New York, 1992 (Chapter 13).
[16] D.V. Prokhorov, D.C. Wunsch, Adaptive critic designs, IEEE Trans. Neural Netw. 8 (5) (1997) 997–1007.
[17] T. Landelius, Reinforcement learning and distributed local model synthesis (Ph.D. dissertation), Linkoping University, Sweden, 1997.
[18] A. Al-Tamimi, F.L. Lewis, M. Abu-Khalaf, Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof, IEEE Trans. Syst. Man Cybern. B 38 (4) (2008) 943–949.
[19] A. Al-Tamimi, F.L. Lewis, Discrete-time nonlinear HJB solution using approximate dynamic programming: convergence proof, in: Proceedings of the IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, Honolulu, HI, April 2007, pp. 38–43.
[20] A. Al-Tamimi, M. Abu-Khalaf, F.L. Lewis, Adaptive critic designs for discrete time zero-sum games with application to Hinfty control, IEEE Trans. Syst. Man Cybern. B 37 (1) (2007) 240–247.
[21] X.M. Tan, D.B. Zhao, J.Q. Yi, D. Xu, Adaptive integrated control for omnidirectional mobile manipulators based on neural-network, Int. J. Cogn. Inform. Natural Intell. 4 (4) (2009) 34–53.
[22] E.Q. Zhang, S.J. Shi, Z.X. Weng, Fuzzy systems using triangle MFs as interpolation functions, Acta Autom. Sin. 27 (6) (2001) 784–890.
[23] L.X. Wang, J.M. Mendel, Fuzzy basis function, universal approximation, and orthogonal least squares learning, IEEE Trans. Neural Netw. 3 (5) (1992) 807–814.
[24] H. Ying, General SISO Takagi–Sugeno fuzzy systems with linear rule consequent are universal approximators, IEEE Trans. Fuzzy Syst. 6 (1998) 582–587.
[25] C.C. Lee, Fuzzy logic in control systems: fuzzy logic controller—Part I, IEEE Trans. Syst. Man Cybern. 20 (1990) 404–418.

[26] A. Katbab, Fuzzy logic and controller design—a review, in: Proceedings of IEEE South-Eastern '95. 'Visualize the Future', Raleigh, NC, March 1995, pp. 443–449.
[27] L.X. Wang, Stable adaptive fuzzy control of nonlinear systems, IEEE Trans. Fuzzy Syst. I (1993) 146–155.
[28] J. Zhang, H. Zhang, Y. Luo, H. Liang, Nearly optimal control scheme using adaptive dynamic programming based on generalized fuzzy hyperbolic model, Acta Autom. Sin. 39 (2) (2013) 142–148.
[29] Y.H. Zhu, D.B. Zhao, H.B. He, Integration of fuzzy controller with adaptive dynamic programming, in: Proceedings of the 10th World Congress on Intelligent Control and Automation (WCICA'2012), Beijing, China, July 2012, pp. 310–315.
[30] D.B. Zhao, Y.H. Zhu, H.B. He, Neural and fuzzy dynamic programming for under-actuated systems, in: 2012 International Joint Conference on Neural Networks (IJCNN'2012), Brisbane, Australia, June 2012, pp. 1895–1901.

Yuanheng Zhu received the B.S. degree in school of management and engineering from Nanjing University, Nanjing, China, in July 2010. He is currently a Ph.D. candidate with the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, China. His current research interests lie in the area of computational intelligence, adaptive dynamic programming, and fuzzy systems.



Dongbin Zhao received the B.S., M.S., Ph.D. degrees in August 1994, August 1996, and April 2000 respectively, in materials processing engineering from Harbin Institute of Technology, China. Dr. Zhao was a postdoctoral fellow in humanoid robot at the Department of Mechanical Engineering, Tsinghua University, China, from May 2000 to January 2002.

He is currently a professor at the State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, China. He has published one book and over thirty international journal papers. His current research interests lie in the area of computational intelligence, adaptive dynamic programming, robotics, intelligent transportation systems, and process simulation.

Dr. Zhao is an Associate Editor of the IEEE Transactions on Neural Networks and Learning Systems, and Cognitive Computation.

Derong Liu received the Ph.D. degree in electrical engineering from the University of Notre Dame in 1994. Dr. Liu was a Staff Fellow with General Motors Research and Development Center, Warren, MI, from 1993 to 1995. He was an Assistant Professor in the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, from 1995 to 1999. He joined the University of Illinois at Chicago in 1999, and became a Full Professor of electrical and computer engineering and of computer science in 2006. He was selected for the "100 Talents Program" by the Chinese Academy of Sciences in 2008. He has published 10 books. Dr. Liu has been an Associate Editor of several IEEE publications. Currently, he is the Editor-in-Chief of the IEEE Transactions on Neural Networks and Learning Systems, and an Associate Editor of the IEEE Transactions on Control Systems Technology. He was an elected AdCom member of the IEEE Computational Intelligence Society (2006–2008). He received the Faculty Early Career Development (CAREER) award from the National Science Foundation (1999), the University Scholar Award from University of Illinois (2006–2009), and the Overseas Outstanding Young Scholar Award from the National Natural Science Foundation of China (2008).