A Generalized Policy Iteration Adaptive Dynamic Programming Algorithm for Optimal Control of Discrete-Time Nonlinear Systems with Actuator Saturation

Qiao Lin, Qinglai Wei(✉), and Bo Zhao

University of Chinese Academy of Sciences, Beijing 100190, China
{linqiao2014,qinglai.wei,zhaobo}@ia.ac.cn

Abstract. In this study, a nonquadratic performance function is introduced to overcome the saturation nonlinearity in actuators. Then a novel solution, generalized policy iteration adaptive dynamic programming algorithm, is applied to deal with the problem of optimal control. To achieve this goal, we use two neural networks to approximate control vectors and performance index function. Finally, this paper focuses on an example simulated on Matlab, which verifies the excellent convergence of the mentioned algorithm and feasibility of this scheme.

Keywords: Adaptive dynamic programming · Neural network · Optimal control · Saturating actuators

1 Introduction

In the control field, saturation nonlinearity of the actuators is universal phenomenon. So optimizing control of systems in which actuators have problem of saturating nonlinearity, is a major and increasing concern [1,2]. However, these traditional methods were proposed without considering the optimal control problem. In order to overcome this shortcoming, Lewis et al. [3] used adaptive dynamic programming (ADP) algorithm. The ADP algorithm [4–6], an effective brain-like method, which can give the solution to Hamilton-Jacobi-Bellman (HJB) equation forward-in-time, provides an important way of obtaining policy of optimizing control. The value and policy iteration algorithms [7,8] are key of the ADP algorithms. Considering the superiority of ADP algorithm, growing researchers chose ADP algorithm in terms of optimal control. Zhang et al. [9] used greedy ADP algorithm to design the infinite-time optimal tracking controller. Qiao et al. [10] applied ADP algorithm to a large wind farm and a STATCOM, with focusing on Coordinated reactive power control. Liu et al. [11] developed an optimizing controller for some systems which were discrete-time nonlinear and had control constraints by DHP. As mentioned in [12], ADP algorithm is also suitable for time-delay systems with the same saturation challenge

F. Cong et al. (Eds.): ISNN 2017, Part II, LNCS 10262, pp. 60–65, 2017.
DOI: 10.1007/978-3-319-59081-3_8

as above. However, in order to realize constrained optimal control, there is still no research using the generalized policy iteration ADP algorithm.

This paper focuses on the generalized policy iteration ADP algorithm. The present algorithm has i-iteration and j-iteration. When j is equal to zero, the proposed algorithm will be a value iteration algorithm, while becoming a policy iteration algorithm when j approaches the infinity. Firstly, the nonquadratic performance function is introduced to overcome the saturation nonlinearity. Then, the process of the generalized policy iteration algorithm is given. Lastly, the simulation results verify the efficiency of the developed method.

2 Problem Statement

We will study the following discrete-time nonlinear systems:

$$\begin{aligned} x_{k+1} &= F(x_k, u_k) \\ &= f(x_k) + g(x_k)u_k \end{aligned} \tag{1}$$

where $u_k \in \mathbb{R}^m$ is control vector, $x_k \in \mathbb{R}^n$ is the state vector, $f(x_k) \in \mathbb{R}^n$ and $g(x_k) \in \mathbb{R}^{n\times m}$ are system functions. We denote $\Omega_u = \{u_k | u_k = [u_{1k}, u_{2k}, \ldots, u_{mk}]^\mathsf{T} \in \mathbb{R}^m, |u_{ik}| \le \overline{u}_i, i = 1, 2, \ldots, m\}$, where $\overline{u}_i$ can be regarded as the saturating bound. Let $\overline{U} = diag[\overline{u}_1, \overline{u}_2, \ldots, \overline{u}_m]$.

The generalized nonquadratic performance index function is $J(x_k, \underline{u}_k) = \sum\limits_{i=k}^{\infty} \left\{x_i^\mathsf{T} Q x_i + W(u_i)\right\}$, where $\underline{u}_k = \{u_k, u_{k+1}, u_{k+2}, \ldots\}$, the weight matrix Q and $W(u_i) \in \mathbb{R}$ are positive definite.

Inspired by the paper [3], we can introduced $W(u_i) = 2\int_0^{u_i} \varLambda^{-\mathsf{T}}(\overline{U}^{-1}s)\overline{U}Rds$, where R is positive definite, $s \in \mathbb{R}^m$, $\varLambda \in \mathbb{R}^m$, $\varLambda^{-\mathsf{T}}$ denotes $(\varLambda^{-1})^\mathsf{T}$, and $\varLambda(\cdot)$ can choose $\tanh(\cdot)$.

Then we can use $J^*(x_k) = \min\limits_{\underline{u}_k} J(x_k, \underline{u}_k)$ to stand for the optimal performance index function and use u_k^* to be the optimal control vector. So from the principle of discrete-time Bellman's optimality, we can obtain the optimal performance index function as

$$J^*(x_k) = \min_{u_k} \left\{ x_k^\mathsf{T} Q x_k + 2\int_0^{u_k} \varLambda^{-\mathsf{T}}(\overline{U}^{-1}s)\overline{U}Rds + J^*(x_{k+1})) \right\}. \tag{2}$$

And we can use the following equation to stand for the optimal control vector:

$$u_k^* = \arg\min_{u_k} \left\{ x_k^\mathsf{T} Q x_k + 2\int_0^{u_k} \varLambda^{-\mathsf{T}}(\overline{U}^{-1}s)\overline{U}Rds + J^*(x_{k+1}) \right\}. \tag{3}$$

The goal of this paper is to get the optimal control vector u_k^* and the optimal performance index function $J^*(x_k)$.

3 Derivation of the Generalized Policy Iteration ADP Algorithm

From [16], it's known that the traditional ADP algorithm just have one iteration procedure. However, the generalized policy iteration ADP algorithm has i-iteration and j-iteration. Specially, for i-iteration, the generalized policy iteration ADP algorithm doesn't need to solve the HJB equation, which speed the convergence rate of the developed ADP algorithm.

According to [17], if a control vector can stabilize the system (1) and make the performance index function finite at the same time, it can be concluded that the control vector is admissible.

Next, we will get that the control vector and cost function of the developed generalized policy iteration ADP algorithm are updated in each iteration. First, the cost function $V_0(x_k)$ can be initialed as follows:

$$V_0(x_k) = x_k^{\mathsf{T}} Q x_k + 2\int_0^{v_0(x_k)} \Lambda^{-\mathsf{T}}(\overline{U}^{-1}s)\overline{U}R ds + V_0(F(x_k, v_0(x_k))), \tag{4}$$

where the $v_0(x_k)$ is an initial admissible control vector. Then, for $i = 1$, the control vector $v_1(x_k)$ can be gained by:

$$v_1(x_k) = \arg\min_{u_k} \left\{ x_k^{\mathsf{T}} Q x_k + 2\int_0^{u_k} \Lambda^{-\mathsf{T}}(\overline{U}^{-1}s)\overline{U}R ds + V_0(F(x_k, u_k)) \right\}. \tag{5}$$

Then, we will introduced the second iteration procedure. Define an arbitrary non-negative integer sequence, that is $\{L_1, L_2, L_3, \ldots\}$. L_1 is the upper boundary of j_1. When j_1 increases from 0 to L_1, we can have the iterative cost function by

$$V_{1,j_1+1}(x_k) = x_k^{\mathsf{T}} Q x_k + 2\int_0^{v_1(x_k)} \Lambda^{-\mathsf{T}}(\overline{U}^{-1}s)\overline{U}R ds + V_{1,j_1}(F(x_k, v_1(x_k))), \tag{6}$$

where

$$V_{1,0}(x_k) = x_k^{\mathsf{T}} Q x_k + 2\int_0^{v_1(x_k)} \Lambda^{-\mathsf{T}}(\overline{U}^{-1}s)\overline{U}R ds + V_0(F(x_k, v_1(x_k))). \tag{7}$$

In the second iteration, the cost function changes to be $V_1(x_k) = V_{1,L_1}(x_k)$. For $i = 2, 3, 4, \ldots$, the control vector and cost function of the developed ADP algorithm are updated by:

(1) i-iteration

$$v_i(x_k) = \arg\min_{u_k} \left\{ x_k^{\mathsf{T}} Q x_k + 2\int_0^{u_k} \Lambda^{-\mathsf{T}}(\overline{U}^{-1}s)\overline{U}R ds + V_{i-1}(F(x_k, u_k)) \right\}, \tag{8}$$

(2) j-iteration

$$V_{i,j_i+1}(x_k) = x_k^{\mathsf{T}} Q x_k + 2\int_0^{v_i(x_k)} \Lambda^{-\mathsf{T}}(\overline{U}^{-1}s)\overline{U}R ds + V_{i,j_i}(F(x_k, v_i(x_k))), \tag{9}$$

where $j_i = 0, 1, 2, \ldots, L_i$,

$$V_{i,0}(x_k) = x_k^{\mathsf{T}} Q x_k + 2\int_0^{v_i(k)} \Lambda^{-\mathsf{T}}(\overline{U}^{-1}s)\overline{U}R ds + V_{i-1}(F(x_k, v_i(x_k))) \tag{10}$$

and we can get the iterative cost function by

$$V_i(x_k) = V_{i,L_i}(x_k). \tag{11}$$

From (4)–(11), we make use of $V_{i,j_i}(x_k)$ to approximate $J^*(x_k)$ and $v_i(x_k)$ to approximate u_k^*. In the following, an example is applied to illustrate the convergence and feasibility of the presented ADP algorithm.

4 Simulation Example

The following nonlinear system is mass-spring system:

$$x(k+1) = f(x_k) + g(x_k)u(k), \tag{12}$$

where

$$\begin{aligned} x_k &= \begin{bmatrix} x_{1k} \\ x_{2k} \end{bmatrix}, \\ f(x_k) &= \begin{bmatrix} x_{1k} + 0.05x_{2k} \\ -0.0005x_{1k} - 0.0335x_{1k}^3 + x_{2k} \end{bmatrix}, \\ g(x_k) &= \begin{bmatrix} 0 \\ 0.05 \end{bmatrix}, \end{aligned}$$

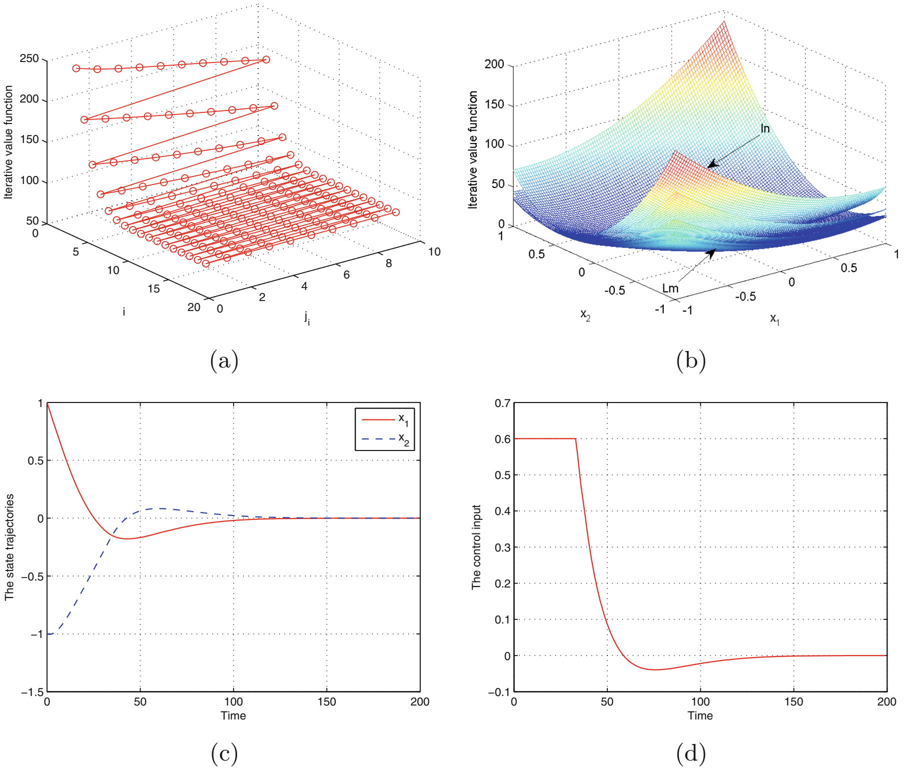
and the system is controlled with control constraint of $|u| \leq 0.6$. The cost function is defined by

$$J(x_k) = \sum_{i=k}^{\infty} \left\{ x_i^{\mathsf{T}} Q x_i + 2\int_0^{u_i} \tanh^{-\mathsf{T}}(\overline{U}^{-1}s)\overline{U}R ds \right\},$$

where $Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, $R = 0.5$, $\overline{U} = 0.6$.

The developed iteration ADP algorithm is implemented by NNs. The hidden layers of the critic network and action network both are 10 neurons. For each iteration step, we train the networks for 4000 training steps so as to make the training error become minimum. The learning rate of the above two networks both are 0.01.

From Fig. 1(a) and (b), we can get the convergent process of the cost function $V_{i,j_i}(x_k)$ and the subsequence $V_i(x_k)$. Next, we use the optimal control vectors to control the system (12) with the initial state $x(0) = [1, -1]^{\mathsf{T}}$ for 200 time steps. Figure 1(c) and (d) display the changing curves of the state x and the control u. The effective of the presented ADP algorithm in handling optimal control problem for discrete-time nonlinear systems with actuator saturation is verified through the simulation results.



(a)

(b)

(c)

(d)

Fig. 1. Simulation results (a) Convergence of V_{i,j_i} (b) Convergence of V_i (c) State trajectories (d) Control vectors

5 Conclusion

In this paper, a novel ADP algorithm is chosen to treat the optimal control problem for discrete-time nonlinear systems with control constraint. One example demonstrates the convergence and feasibility of the presented iteration ADP algorithm. Since the time-delay problem is another hot topic in the control field, it's significant to use the developed ADP algorithm to handle the time-delay systems in the future.

Acknowledgments. This work was supported partly by the National Natural Science Foundation of China (Nos. 61374105, 61374051, 61533017, 61233001, 61273140, 61304086 and U1501251).

References

1. Saberi, A., Lin, Z., Teel, A.: Control of linear systems with saturating actuators. IEEE Trans. Autom. Control **41**(3), 368–378 (1996)
2. Sussmann, H., Sontag, E., Yang, Y.: A general result on the stabilization of linear systems using bounded controls. IEEE Trans. Autom. Control **39**(12), 2411–2425 (1994)
3. Abu-Khalaf, M., Lewis, F.: Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. Automatica **41**(5), 779–791 (2005)
4. Werbos, P.: Approximate dynamic programming for real-time control and neural modeling. In: White, D.A., Sofge, D.A. (eds.) Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches (1992)
5. Liu, D., Wang, D., Zhao, D., et al.: Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming. IEEE Trans. Autom. Sci. Eng. **9**(3), 628–634 (2012)
6. Wei, Q., Song, R., Yan, P.: Data-driven zero-sum neuro-optimal control for a class of continuous-time unknown nonlinear systems with disturbance using ADP. IEEE Trans. Neural Netw. Learn. Syst. **27**(2), 444–458 (2016)
7. Wei, Q., Liu, D., Shi, G., et al.: Optimal multi-battery coordination control for home energy management systems via distributed iterative adaptive dynamic programming. IEEE Trans. Industr. Electron. **42**(7), 4203–4214 (2015)
8. Bhasin, S., Kamalapurkar, R., Johnson, M., et al.: A novel actorcritic- identifier architecture for approximate optimal control of uncertain nonlinear systems. Automatica **49**(1), 82–92 (2013)
9. Zhang, H., Wei, Q., Luo, Y.: A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm. IEEE Trans. Syst. Man Cybern.-Part B: Cybern. **38**(4), 937–942 (2008)
10. Qiao, W., Harley, R.G., Venayagamoorthy, G.K.: Coordinated reactive power control of a large wind farm and a STATCOM using heuristic dynamic programming. IEEE Trans. Energy Convers. **24**(2), 493–503 (2009)
11. Liu, D., Wang, D., Yang, X.: An iterative adaptive dynamic programming algorithm for optimal control of unknown discrete-time nonlinear systems with constrained inputs. Inf. Sci. **220**(1), 331–342 (2013)
12. Song, R., Zhang, H., Luo, Y., et al.: Optimal control laws for time-delay systems with saturating actuators based on heuristic dynamic programming. Neurocomputing **73**, 3020–3027 (2010)
13. Vrabie, D., Vamvoudakis, K., Lewis, F.: Adaptive optimal controllers based on generalized policy iteration in a continuous-time framework. In: 17th Mediterranean Conference on Control & Automation, Thessaloniki, Greece, pp. 1402–1409 (2009)
14. Lin, Q., Wei, Q., Liu, D.: A novel optimal tracking control scheme for a class of discrete-time nonlinear systems using generalized policy iteration adaptive dynamic programming algorithm. Int. J. Syst. Sci. **48**(3), 525–534 (2017)
15. Apostol, T.: Mathematical Analysis, 2nd edn. Addison-Wesley Press
16. Wang, F., Zhang, H., Liu, D.: Adaptive dynamic programming: an introduction. IEEE Comput. Intell. Mag. **4**(2), 39–47 (2009)
17. Liu, D., Wei, Q.: Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems. IEEE Trans. Neural Netw. Learn. Syst. **25**(3), 621–634 (2014)