UKACC International Conference on Control 2012
Cardiff, UK, 3-5 September 2012

Optimal Control Scheme for Nonlinear Systems with Saturating Actuator Using ε-Iterative Adaptive Dynamic Programming

Xiaofeng Lin, Yuanjun Huang and Nuyun Cao
School of Electrical Engineering, Guangxi University
Nanning, China
gxulinxf@163.com

Yuzhang Lin
Department of Electrical Engineering, Tsinghua University
Beijing, China
90lyz@163.com

Abstract— **In this paper, a finite-horizon optimal control scheme for a class of nonlinear systems with saturating actuator is proposed by an improved iterative adaptive dynamic programming (ADP) algorithm. The Hamilton-Jacobi-Bellman (HJB) equation corresponding to constrained control is formulated using a suitable nonquadratic function. Then mathematical analysis of the convergence is presented, by proving that the performance index function can reach the optimum using the adaptive iteration. Finally the finite-horizon optimal control law can be obtained by the ε-iterative adaptive algorithm. The examples are given to demonstrate the effectiveness of the above methods.**

Keywords-Adaptive dynamic programming(ADP); Saturating actuators; nonlinear system; Finite time optimal control

I. INTRODUCTION

In a practical control system, the saturating actuators will reduce the system's dynamic performance, and even affect the stability of the system. Therefore seeking a better way to design control systems with saturating actuator has attached considerable attention by many researchers in recent years. The stability of discrete-time linear systems subject to actuator saturation was analyzed using a saturation-dependent Lyapunov function based on the solution of an LMI optimization problem[1]. Saberi (1996) and Sussmann(1994) proposed several processes to control saturation problems, but they did not consider non-linear systems and optimal problems (see [2],[3]). A gain-scheduled output control design for systems coping with nonlinear time-varying parameter dependent systems subject to saturated actuators was proposed in [4]. Pontryagin's Minimum Principle is a way to solve optimal control problem with Saturating Actuator. However, this needs to solve differential equations with boundary, and the result we get is an open-loop control by this way. Lyshevski designed an optimal control for a closed-loop feedback system using a non-quadratic performance index function problem to deal with the control constraints based on dynamic programming principle in [5], but the difficulty lies in the HJB equations.

Adaptive Dynamic Programming (ADP) is a powerful tool proposed by the idea of adaptive critic and reinforcement learning with dynamic programming[6]. It is solved by iterative algorithm to get an approximate solution of HJB equation, ADP has become an effective tool for optimal control problems and has achieved many results[7],[8-12]. A greedy iterative adaptive algorithm was proposed to solve the nonlinear discrete-time systems HJB equation in [8]. Iterative ADP was used to get an infinite horizon optimal control scheme for nonlinear systems with saturating actuator in [9].

However, for practical systems, a limited period of time is required to achieve control. Finite time optimal control problem could be backward solved by dynamic programming, when facing the multi-dimensional nonlinear characteristics of complex systems, the calculation will be very large, that is the problem of the "curse of dimensionality". ε-error bounds of adaptive algorithm was proposed to deal with finite time optimal control in [10],[11]. To the best of our knowledge, quite few research has been presented to deal with finite-horizon optimal control with saturating actuator. This motivates our research.

This paper aims to solve finite time optimal control problem for nonlinear systems with saturating actuator. It is organized as follows. In Section II, the problem is introduced and HJB equation corresponding to constrained control is presented. In Section III, the iterative ADP algorithm and its convergence for finite-horizon optimal control problem are derived. In Section IV, ε-optimal control algorithm is developed with the definition of finite iteration steps and ε-optimal control. In Section V, an example is given to demonstrate the effectiveness of the algorithm. In Section VI, the conclusion is drawn.

II. HJB EQUATION CORRESPONDING TO CONSTRAINED CONTROL

A. Deal with saturated problem

Consider the following class of discrete-time nonlinear systems

$$\boldsymbol{x}(k+1)=\boldsymbol{F}\left(\boldsymbol{x}(k),\boldsymbol{u}(k)\right) \qquad k=0,1,2,\ldots \tag{1}$$

Where $\boldsymbol{x}(k)\in\mathbb{R}^n$ is the state, assume the system $\boldsymbol{F}\left(\boldsymbol{x}(k),\boldsymbol{u}(k)\right)$ is Lipschitz continuous controllable on a set $\boldsymbol{\Omega}$

This work is supported by National Natural Science Foundation of China (60964002, 61034002)

containing the origin. The control $\boldsymbol{u}(k) \in \boldsymbol{\Omega}_{\boldsymbol{u}}$, and

$\boldsymbol{\Omega}_{\boldsymbol{u}} = \{\boldsymbol{u}(k) = [u_1(k), u_2(k), \cdots, u_m(k)]^{\mathrm{T}} \in \mathbf{R}^{\mathrm{m}} : |u_i(k)| \le \bar{u}_i\}$, where $\bar{u}_i$ is the i-th execution controller saturation boundary, $i = 1, ..., m$. On the other hand, the constant diagonal matrix is $\mathbf{A} = diag[\bar{u}_1, \bar{u}_2, \cdots, \bar{u}_m], \mathbf{A} \in \mathbb{R}^{m \times m}$. $\boldsymbol{F}(\mathbf{0}, \mathbf{0}) = \mathbf{0}$, hence $\boldsymbol{x} = \mathbf{0}$ is an equilibrium state of system (1) under the control $\boldsymbol{u} = \mathbf{0}$.

For the initial state $\boldsymbol{x}(0)$, to minimize the performance index function defined in (2) with finite control sequences as $\underline{\boldsymbol{u}}_0^{N-1} = (\boldsymbol{u}(0), \boldsymbol{u}(1), \cdots, \boldsymbol{u}(N-1))$.

$$\boldsymbol{J}\left(\boldsymbol{x}(0), \underline{\boldsymbol{u}}_0^{N-1}\right) = \sum_{k=0}^{N-1} \left\{\boldsymbol{x}(k)^{\mathrm{T}} Q \boldsymbol{x}(k) + \boldsymbol{W}(\boldsymbol{u}(k))\right\} \tag{2}$$

Where $\boldsymbol{W}(\boldsymbol{u}(k)) = 2\int_0^{\boldsymbol{u}(k)} (\mathbf{A}\tanh^{-1}(\boldsymbol{s}/\mathbf{A}))^{\mathrm{T}} \mathbf{R} ds$. The length N is determined with terminal time, this kind of optimal control problems has been called finite-horizon problems with unspecified terminal time [10].

If we set $\boldsymbol{W}(\boldsymbol{u}(k)) = \boldsymbol{u}(k)^{\mathrm{T}} \boldsymbol{R} \boldsymbol{u}(k)$ in (2), the quadratic performance index function with unconstrained control is used to design the optimal control for control system with saturated actuator, however, this system could not ensure the optimal system performance, even may lead to system instability. According to [12], let

$$\boldsymbol{W}(\boldsymbol{u}(k)) = 2\int_0^{\boldsymbol{u}(k)} \mathbf{A}\boldsymbol{\phi}^{-T}(s/\mathbf{A}) \mathbf{R} ds \tag{3}$$

Where

$$\boldsymbol{\phi}^{-1}(u(k)) = [\phi^{-1}(u_1(k)), \phi^{-1}(u_2(k)), \cdots \phi^{-1}(u_m(k))]^T$$

$|\boldsymbol{u}| \le \mathbf{A}$, $\boldsymbol{s} \in \mathbb{R}^m$, $\boldsymbol{\phi} \in \mathbb{R}^m$, let $\boldsymbol{R}$ be a diagonal positive definite, $\phi(\bullet)$ is a bounded monotonically increasing odd function belongs to $\mathbb{C}^p(p \ge 1)$ and $\boldsymbol{L}_2(\Omega)$ with $|\phi(\bullet)| \le 1$, the first derivative is a bounded constant $\boldsymbol{M}$, such function as $\phi(\bullet) = \tanh(\bullet)$. Figure 1 shows a well approaching saturation when $|u| \le 0.5$ with function $\boldsymbol{\phi}(\bullet)$. Therefore, we can guarantee the control output signal within the range of actuator saturation using the performance index function in (2).

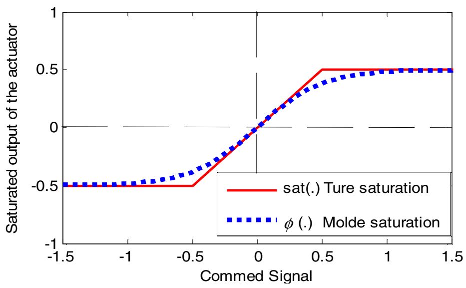

Figure 1. The model of saturation when $|u| \le 0.5$

B. *HJB equation and solution*

Definition 1 : The corresponding finite-horizon admissible control sequence of performance index function in (2) $\underline{\boldsymbol{u}}_k^{k+N-1}$ is defined as follows: For $\forall \boldsymbol{x}(k) \in \mathbb{R}^n$, there exists a control sequence $\underline{\boldsymbol{u}}_k^{k+N-1}$ satisfy $\boldsymbol{x}^f(\boldsymbol{x}(k), \underline{\boldsymbol{u}}_k^{k+N-1}) = \boldsymbol{0}$ and $\boldsymbol{J}(\boldsymbol{x}(k), \underline{\boldsymbol{u}}_k^{N-1})$ is finite. Where $N > 0$ is a positive integer, $\boldsymbol{x}^f(\boldsymbol{x}(k), \underline{\boldsymbol{u}}_k^{k+N-1})$ is the terminal state.

Let $\mathbf{C}_{\boldsymbol{x}_k}^N = \left\{\underline{\boldsymbol{u}}_k^{k+N-1} : \boldsymbol{x}^f(\boldsymbol{x}(k), \underline{\boldsymbol{u}}_k^{k+N-1}) = \boldsymbol{0}, \left|\underline{\boldsymbol{u}}_k^{k+N-1}\right| = N\right\}$ be all the admissible control sets with length N .Assume a state $\boldsymbol{x}(k)$ is a finite-horizon admissible control. By Definition 1, the optimal performance index function at the finite-horizon admissible control could be written as

$$\boldsymbol{J}^*(\boldsymbol{x}(k)) = \inf_{\underline{\boldsymbol{u}}_k^{N-1}} \left\{\boldsymbol{J}(\boldsymbol{x}(k), \underline{\boldsymbol{u}}_k^{N-1}) : \underline{\boldsymbol{u}}_k^{N-1} \in \mathbf{C}_{\boldsymbol{x}_k}^N\right\} \tag{4}$$

According to the performance index function defined by equation (2) and Bellman principle of optimality, $\boldsymbol{J}^*(\boldsymbol{x}(k))$ under discrete time HJB equation can be written as

$$\boldsymbol{J}^*(\boldsymbol{x}(k)) = \min_{\boldsymbol{u}(k)} \left\{\boldsymbol{x}(k)^{\mathrm{T}} Q \boldsymbol{x}(k) + \boldsymbol{W}(\boldsymbol{u}(k)) + \boldsymbol{J}^*(\boldsymbol{F}(\boldsymbol{x}(k), \boldsymbol{u}(k)))\right\} \tag{5}$$

Define the optimal control sequence starting at k with length of N by

$$\underline{\boldsymbol{u}}_k^*(\boldsymbol{x}(k)) = \inf_{\underline{\boldsymbol{u}}_k^{k+N-1}} \left\{\boldsymbol{J}(\boldsymbol{x}(k), \underline{\boldsymbol{u}}_k^{k+N-1}) : \underline{\boldsymbol{u}}_k^{k+N-1} \in \mathbf{C}_{\boldsymbol{x}_k}^N\right\} \tag{6}$$

and define the one step optimal control vector by

$$\boldsymbol{u}^*(\boldsymbol{x}(k)) = \arg\min_{\boldsymbol{u}(k)} \left\{\boldsymbol{x}(k)^{\mathrm{T}} Q \boldsymbol{x}(k) + \boldsymbol{W}(\boldsymbol{u}(k)) + \boldsymbol{J}^*(\boldsymbol{F}(\boldsymbol{x}(k), \boldsymbol{u}(k)))\right\} \tag{7}$$

Dynamic programming is used to solving optimal control sequence in equation (6), while the first step is to determine $\boldsymbol{u}^*(\boldsymbol{x}(k+N-1))$ by equation(6) in terminal state $\boldsymbol{x}(k+N-1)$.

$$\boldsymbol{u}^*(\boldsymbol{x}(k+N-1)) = \operatorname*{argmin}_{\boldsymbol{u}(N-1)} \left\{\boldsymbol{x}(k+N-1)^{\mathrm{T}} Q \boldsymbol{x}(k+N-1) + \boldsymbol{W}(\boldsymbol{u}(k+N-1))\right\}$$
$$\boldsymbol{s.t.}\ \boldsymbol{F}(\boldsymbol{x}(k+N-1), \boldsymbol{u}(k+N-1)) = \mathbf{0} \tag{8}$$

Putting $\boldsymbol{u}^*(\boldsymbol{x}(k+N-1))$ into (5), the optimal performance index function

$$\boldsymbol{J}^*(\boldsymbol{x}(k+N-1)) = \boldsymbol{x}(k+N-1)^{\mathrm{T}} Q \boldsymbol{x}(k+N-1) + \boldsymbol{W}(\boldsymbol{u}^*(k+N-1)) \tag{9}$$

After $\boldsymbol{u}^*(\boldsymbol{x}(k+N-1))$ and $\boldsymbol{J}^*(\boldsymbol{x}(k+N-1))$ is obtained, $\boldsymbol{u}^*(\boldsymbol{x}(k+N-2))$ and $\boldsymbol{J}^*(\boldsymbol{x}(k+N-2)), \ldots$, could be determined by equation (5) and (7), finally $\boldsymbol{u}^*(\boldsymbol{x}(k))$ and $\boldsymbol{J}^*(\boldsymbol{x}(k))$ is solved. Then solving $\boldsymbol{u}^*(k) = \boldsymbol{u}^*(\boldsymbol{x}(k))$ with $\boldsymbol{x}(k)$, and putting $\boldsymbol{u}^*(k)$ into equation (1) so as to get $\boldsymbol{x}(k+1) = \boldsymbol{F}(\boldsymbol{x}(k), \boldsymbol{u}^*(k))$. By the same way, solving $\boldsymbol{u}^*(k+1)$ by $\boldsymbol{x}(k+1)$ and $\boldsymbol{u}^*(\boldsymbol{x}(k+1))$, putting $\boldsymbol{u}^*(\boldsymbol{x}(k+1))$ into equation (1) , then we can easily get $\boldsymbol{x}(k+2) = \boldsymbol{F}(\boldsymbol{x}(k+1), \boldsymbol{u}^*(k+1))$, repeating this process [9], the optimal control sequence will be obtained as $\underline{\boldsymbol{u}}_k^*(\bullet) = \{\boldsymbol{u}^*(k), \boldsymbol{u}^*(k+1), ..., \boldsymbol{u}^*(k+N-1)\}$.

However, the performance index function in this paper is non- quadratic, and the system is non-linear. Therefore, it is hard to get an analytical solution of optimal control law by solving HJB equation. On the other hand, when dealing with discrete time dynamic programming by backward method, one has to calculate and save all $\boldsymbol{J}^*(\boldsymbol{x}(k))$ and $\boldsymbol{u}^*(\boldsymbol{x}(k))$ of the sequence. Hence, it will meet difficulties when calculating finite-horizon optimal control problems by dynamic programming.

III. FINITE TIME ITERATIVE ADP ALGORITHM

A. *Formula derived for iterative ADP*

For any state $\boldsymbol{x}(k)$, the performance index function $\{\boldsymbol{V}_i\}$ and control policy $\{\upsilon_i\}$ in the iterative ADP algorithm are updated by recursive iterations. The iterative starts with $i=0$ and the initial performance index function $\boldsymbol{V}_0(\boldsymbol{x}(k))=0$, the performance index function for $i=1$ is computed as

$$\begin{aligned}
\boldsymbol{V}_1(\boldsymbol{x}(k)) &= \min_{\boldsymbol{u}(k)}\left\{\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+\boldsymbol{W}(\boldsymbol{u}(k))+\boldsymbol{V}_0(\boldsymbol{F}(\boldsymbol{x}(k),\boldsymbol{u}(k)))\right\} \\
&\qquad s.t.\ \boldsymbol{F}(\boldsymbol{x}(k),\boldsymbol{u}(k))=\mathbf{0} \\
&= \min_{\boldsymbol{u}(k)}\left\{\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+\boldsymbol{W}(\boldsymbol{u}(k))\right\} \\
&\qquad s.t.\ \boldsymbol{F}(\boldsymbol{x}(k),\boldsymbol{u}(k))=\mathbf{0} \\
&= \boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+\boldsymbol{W}(\boldsymbol{u}^*(\boldsymbol{x}(k)))
\end{aligned} \tag{10}$$

Where $\boldsymbol{F}\left(\boldsymbol{x}(k),\boldsymbol{u}^*(k)\right)=\mathbf{0}$, let $\boldsymbol{u}^*(k)=\upsilon_1(\boldsymbol{x}(k))$, then (10) can be written as

$$\begin{aligned}
\boldsymbol{V}_1(\boldsymbol{x}(k)) &= \min_{\boldsymbol{u}(k)}\left\{\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+\boldsymbol{W}(\boldsymbol{u}(k))+\boldsymbol{V}_0(\boldsymbol{F}(\boldsymbol{x}(k),\boldsymbol{u}(k)))\right\} \\
&\qquad s.t.\ \boldsymbol{F}(\boldsymbol{x}(k),\boldsymbol{u}(k))=\mathbf{0} \\
&= \boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+\boldsymbol{W}(\upsilon_1(\boldsymbol{x}(k)))
\end{aligned} \tag{11}$$

where

$$\begin{aligned}
\upsilon_1(\boldsymbol{x}(k)) &= \arg\min_{\boldsymbol{u}(k)}\{\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+\boldsymbol{W}(\boldsymbol{u}(k))\} \\
&\qquad s.t.\ \boldsymbol{F}(\boldsymbol{x}(k),\boldsymbol{u}(k))=\mathbf{0}
\end{aligned} \tag{12}$$

For $i=2,3,4\ldots$, the iterative ADP algorithm is updated as follows

$$\begin{aligned}
\boldsymbol{V}_i(\boldsymbol{x}(k)) &= \min_{\boldsymbol{u}(k)}\left\{\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+\boldsymbol{W}(\boldsymbol{u}(k))+\boldsymbol{V}_{i-1}(\boldsymbol{F}(\boldsymbol{x}(k),\boldsymbol{u}(k)))\right\} \\
&= \boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+\boldsymbol{W}(\upsilon_i(\boldsymbol{x}(k)))+\boldsymbol{V}_{i-1}(\boldsymbol{F}(\boldsymbol{x}(k),\upsilon_i(\boldsymbol{x}(k))))
\end{aligned} \tag{13}$$

where

$$\upsilon_i(\boldsymbol{x}(k)) = \arg\min_{\boldsymbol{u}(k)}\{\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+\boldsymbol{W}(\boldsymbol{u}(k))+\boldsymbol{V}_{i-1}(\boldsymbol{F}(\boldsymbol{x}(k),\boldsymbol{u}(k)))\} \tag{14}$$

After i steps iteration, the performance index function sequence is obtained as $\{\boldsymbol{V}_i\}=(\boldsymbol{V}_i,\boldsymbol{V}_{i-1},\cdots,\boldsymbol{V}_1)$ and control policy sequence as $\{\upsilon_i\}=(\upsilon_i,\upsilon_{i-1},\cdots\upsilon_1)$. Note that in this case, each control sequence $\boldsymbol{u}(\cdot)$ will obey with different control policy, that is, for $i=0,1,\cdots N-1$, control sequence $\boldsymbol{u}(i)$ is obtained by υ_i respectively. However, one could prove that $\boldsymbol{V}_i(\boldsymbol{x}(k))$ is limit to $\boldsymbol{J}^*(\boldsymbol{x}(k))$ when $i\to\infty$. Therefore, the performance index function $\boldsymbol{J}^*(\boldsymbol{x}(k))$ in HJB equation will be replaced by the iterative performance index function $\boldsymbol{V}_i(\boldsymbol{x}(k))$ while control policy $\boldsymbol{u}(\boldsymbol{x})$ will be replaced by the iterative control policy.

B. *Convergence of iteration ADP*

Remark 1: According to (2) and (13), we have

$$\boldsymbol{V}_{i+1}(\boldsymbol{x}(k)) = \min_{\underline{\boldsymbol{u}}_k^{k+i}}\{\boldsymbol{J}(\boldsymbol{x}(k),\underline{\boldsymbol{u}}_k^{k+i}):\underline{\boldsymbol{u}}_k^{k+i}\in\mathbf{C}_{\boldsymbol{x}_k}^{(i+1)}\} \tag{15}$$

Prove: According to (13), we have

$$\begin{aligned}
\boldsymbol{V}_i(\boldsymbol{x}(k+i)) &= \min_{\boldsymbol{u}(k+i)}\{\boldsymbol{x}(k+i)^{\mathrm{T}}Q\boldsymbol{x}(k+i)+\boldsymbol{W}(\boldsymbol{u}(k+i))\} \\
&\qquad s.t.\ \boldsymbol{F}(\boldsymbol{x}(k+i),\boldsymbol{u}(k+i))=\mathbf{0}
\end{aligned} \tag{16}$$

Thus,

$$\begin{aligned}
\boldsymbol{V}_{i+1}(\boldsymbol{x}(k)) &= \min_{\boldsymbol{u}(k)}\{\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+\boldsymbol{W}(\boldsymbol{u}(k))+\boldsymbol{V}_i(\boldsymbol{x}(k+1))\} \\
&= \min_{\boldsymbol{u}(k)}\Bigg\{\boldsymbol{x}(k)^{\mathrm{T}}\mathrm{Q}\boldsymbol{x}(k)+\boldsymbol{W}(\boldsymbol{u}(k)) \\
&\quad +\min_{\boldsymbol{u}(k+1)}\Big\{\{\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+\boldsymbol{W}(\boldsymbol{u}(k+1)) \\
&\quad +\min_{\boldsymbol{u}(k+2)}\{\boldsymbol{x}(k+2)^{\mathrm{T}}Q\boldsymbol{x}(k+2)+\boldsymbol{W}(\boldsymbol{u}(k+2))+\cdots \\
&\quad +\min_{\boldsymbol{u}(k+i-1)}\{\boldsymbol{x}(k+i\text{-}1)^{\mathrm{T}}Q\boldsymbol{x}(k+i\text{-}1)+\boldsymbol{W}(\boldsymbol{u}(k+i\text{-}1)) \\
&\quad +\boldsymbol{V}_1(\boldsymbol{x}(k+i))\}\cdots\ \}\Big\}\Bigg\}
\end{aligned} \tag{17}$$

Then (17) could be written as

$$\begin{aligned}
\boldsymbol{V}_{i+1}(\boldsymbol{x}(k)) &= \min_{\underline{\boldsymbol{u}}_k^{k+i}}\{\boldsymbol{x}(k)^{\mathrm{T}}Q\boldsymbol{x}(k)+\boldsymbol{W}(\boldsymbol{u}(k)) \\
&\quad +\boldsymbol{x}(k+1)^{\mathrm{T}}Q\boldsymbol{x}(k+1)+\boldsymbol{W}(\boldsymbol{u}(k+1))+\cdots \\
&\quad +\boldsymbol{x}(k+i)^{\mathrm{T}}Q\boldsymbol{x}(k+i)+\boldsymbol{W}(\boldsymbol{u}(k+i))\} \\
&\qquad s.t.\ \boldsymbol{F}(\boldsymbol{x}(k+i),\boldsymbol{u}(k+i))=\mathbf{0} \\
&= \min_{\underline{\boldsymbol{u}}_k^{k+i}}\{\boldsymbol{J}(\boldsymbol{x}(k),\underline{\boldsymbol{u}}_k^{k+i}):\underline{\boldsymbol{u}}_k^{k+i}\in\mathbf{C}_{\boldsymbol{x}_k}^{(i+1)}\}
\end{aligned} \tag{18}$$

Thus (15) holds.

Three theorems as well as a corollary are given in the following. For the proof in detail, see [10],[11].

Theorem 1: Giving an arbitrary state vector $\boldsymbol{x}(k)$, assume there exist an integral i such that $\mathbf{C}_{\boldsymbol{x}_k}^{(i)}\neq\varnothing$, then $\mathbf{C}_{\boldsymbol{x}_k}^{(i+1)}\neq\varnothing$, the performance index function $\boldsymbol{V}_i(\boldsymbol{x}(k))$ is a monotonically nonincreasing sequence. i.e. $\forall i\geq 1$, $\boldsymbol{V}_{i+1}(\boldsymbol{x}(k))\leq\boldsymbol{V}_i(\boldsymbol{x}(k))$.

Theorem 2: Giving an arbitrary state vector $\boldsymbol{x}(k)$, Define the performance index function $\boldsymbol{V}_\infty(\boldsymbol{x}(k))$ as the limit of the iterative function $V_i(\boldsymbol{x}(k))$, i.e.,

$$\boldsymbol{V}_\infty(\boldsymbol{x}(k)) = \lim_{i\to\infty} \boldsymbol{V}_i(\boldsymbol{x}(k)) \tag{19}$$

Thus,

$$V_\infty(\boldsymbol{x}(k)) = \min_{\boldsymbol{u}(k)} \{\boldsymbol{x}(k)^{\mathrm{T}} Q\boldsymbol{x}(k) + \boldsymbol{W}(\boldsymbol{u}(k)) + V_\infty(\boldsymbol{x}(k+1))\} \tag{20}$$

Theorem 3: Define the performance index function $\boldsymbol{V}_\infty(\boldsymbol{x}(k))$ as (13), assume the state $\boldsymbol{x}(k)$ is controllable, then the performance index function $\boldsymbol{V}_\infty(\boldsymbol{x}(k))$ equals the optimal performance index function $\boldsymbol{J}^*(\boldsymbol{x}(k))$, i.e.

$$\lim_{i\to\infty} \boldsymbol{V}_i(\boldsymbol{x}(k)) = \boldsymbol{J}^*(\boldsymbol{x}(k)) \tag{21}$$

Corollary 1: Assume the system state $\boldsymbol{x}(k)$ is admissible controllable and the performance index function is defined in (13), and Theorem 3 holds, then the iterative control law $\upsilon_i(\boldsymbol{x}(k))$ converges to the optimal control law $\boldsymbol{u}^*(\boldsymbol{x}(k))$.

IV. ε-OPTIMAL CONTROL ALGORITHM

A. ε-Optimal Control

In order to get the limit of the performance index function $\boldsymbol{V}_i(\boldsymbol{x}(k))$, iterative ADP algorithm (10) is need to perform, optimal control law $\boldsymbol{u}^*(\boldsymbol{x}(k))$ won't obtain until $i \to \infty$. However, i is usually numbered in practical, $\boldsymbol{J}^*(\boldsymbol{x}(k)) = \boldsymbol{V}_i(\boldsymbol{x}(k))$ could not hold for any finite i. Therefore, error bound ε is introduced for $\boldsymbol{V}_i(\boldsymbol{x}(k))$ and $\boldsymbol{J}^*(\boldsymbol{x}(k))$ in iterative ADP algorithm, so that the performance index function $\boldsymbol{V}_i(\boldsymbol{x}(k))$ approximates to the optimal performance index function $\boldsymbol{J}^*(\boldsymbol{x}(k))$ in finite number of steps.

Definition 2: let $\boldsymbol{\Gamma}_\infty$ be a controllable state set, $\boldsymbol{x}(k) \in \boldsymbol{\Gamma}_\infty, \varepsilon > 0$, then the number of iteration steps $K_\varepsilon(\boldsymbol{x}(k))$ for optimal control is defined as

$$K_\varepsilon(\boldsymbol{x}(k)) = \min\left\{ i : \left| V_i(\boldsymbol{x}(k)) - \boldsymbol{J}^*(\boldsymbol{x}(k)) \right| \le \varepsilon \right\} \tag{22}$$

$K_\varepsilon(\boldsymbol{x}(k))$ refers to the length of control sequence reaching equilibrium point from $\boldsymbol{x}(k)$, since $\boldsymbol{x}(k) \in \boldsymbol{\Gamma}_\infty$, thus $\lim_{i\to\infty} V_i(\boldsymbol{x}(k)) = \boldsymbol{J}^*(\boldsymbol{x}(k))$. Therefore, there exists a finite i such that

$$\left| \boldsymbol{V}_i(\boldsymbol{x}(k)) - \boldsymbol{J}^*(\boldsymbol{x}(k)) \right| \le \varepsilon \tag{23}$$

holds. Thus $\left\{ i : \left| V_i(\boldsymbol{x}(k)) - \boldsymbol{J}^*(\boldsymbol{x}(k)) \right| \le \varepsilon \right\} \ne \varnothing$, so that $K_\varepsilon(\boldsymbol{x}(k))$ is defined.

Definition 3: let $\boldsymbol{x}(k) \in \boldsymbol{\Gamma}_\infty$ be a controllable state vector, for any $\varepsilon > 0$, if $\left| V_i(\boldsymbol{x}(k)) - \boldsymbol{J}^*(\boldsymbol{x}(k)) \right| \le \varepsilon$ holds for the iterative control law $\upsilon_i(\boldsymbol{x}(k))$, then the $\upsilon_i(\boldsymbol{x}(k))$ is defined as a ε-Optimal Control $\boldsymbol{\mu}_\varepsilon^*(\boldsymbol{x}(k))$, i.e.

$$\boldsymbol{\mu}_\varepsilon^*(\boldsymbol{x}(k)) = \upsilon_i(\boldsymbol{x}(k)) = \arg\min_{\boldsymbol{u}(k)} \left\{ \begin{array}{l} \boldsymbol{x}(k)^{\mathrm{T}} Q\boldsymbol{x}(k) + \boldsymbol{W}(\boldsymbol{u}(k)) \\ +V_{i-1}(\boldsymbol{F}(\boldsymbol{x}(k), \boldsymbol{u}(\boldsymbol{x}(k)))) \end{array} \right\} \tag{24}$$

B. Summary of the ε -Optimal Control Algorithm

Step A1. Giving an initial state $\boldsymbol{x}(k)$ and an error bound ε .
Step A2. Set $i = 0$, $V_0(\boldsymbol{x}(k)) = 0$ and $K_\varepsilon(\boldsymbol{x}(k)) = 0$.
Step A3. Calculate $\upsilon_1(\boldsymbol{x}(k)) = \boldsymbol{u}^*(k)$ by (12).
Step A4. For $i = 1$, calculate $V_1(\boldsymbol{x}(k))$ by (11).
Step A5. Set $i = i + 1$ and $K_\varepsilon(\boldsymbol{x}(k)) = i$
Step A6. For $i = 2, 3, \cdots$, calculate $\upsilon_i(\boldsymbol{x}(k))$ by (14), calculate $V_i(\boldsymbol{x}(k))$ by (13).
Step A7. If $\left| V_{i+1}(\boldsymbol{x}(k)) - V_i(\boldsymbol{x}(k)) \right| \le \varepsilon$, go to step A8; then $K_\varepsilon(\boldsymbol{x}(k)) = i$ is the number of optimal control steps, ε-optimal control law is $\boldsymbol{\mu}_\varepsilon^*(\boldsymbol{x}(k)) = \upsilon_i(\boldsymbol{x}(k))$ .otherwise, go to step A5.
Step A8. Stop.

V. SIMULATION STUDY

A. ε-Iterative ADP Algorithm for Saturating Actuator

Consider the following nonlinear system

$$\boldsymbol{x}(k+1) = f(\boldsymbol{x}(k)) + g(\boldsymbol{x}(k))\boldsymbol{u}(k) \tag{25}$$

where

$$f(\boldsymbol{x}(k)) = \begin{bmatrix} -0.5x_2(k) \\ sin(0.8x_1(k) - x_2(k)) + 1.8x_2(k) \end{bmatrix}$$

$$g(\boldsymbol{x}(k)) = \begin{bmatrix} -0.1x_1(k) & 0 \\ 0 & -0.8x_2(k) \end{bmatrix}, \mathbf{A} = \begin{bmatrix} \overline{u}_1 & 0 \\ 0 & \overline{u}_2 \end{bmatrix} = \begin{bmatrix} 0.5 & 0 \\ 0 & 1 \end{bmatrix}, \boldsymbol{Q} = \boldsymbol{R} = \boldsymbol{I}_{2\times 2},$$

The performance index function is

$$\boldsymbol{J}\left(\boldsymbol{x}(k), \boldsymbol{u}(\cdot)\right) = \sum_{i=k}^{k+N-1} \left\{ \boldsymbol{x}(i)^{\mathrm{T}} Q\boldsymbol{x}(i) + \boldsymbol{W}(\boldsymbol{u}(i)) \right\} \tag{26}$$

Neural networks are used to implement the iterative ADP algorithm in [7],[9],[10] with its good function approximating characteristics. The structure diagram of the iterative ADP algorithm using Neural-Network approximate function is shown in figure 2. In the diagram, critic neural network is used to approximate function $V_i(\boldsymbol{x})$, action neural network is used to control law $\upsilon_i(\boldsymbol{x}(k))$, gradient descent algorithm is used to adjust the weight by neural network training rule, the approximate proof and formula derivation can be checked in [7],[9],[12]. The critic network and the action network are chosen as three-layer back-propagation (BP) neural networks with the structures of 2–10–1 and 2–10–2. According to ε-optimal control algorithm, let $\varepsilon = 10^{-3}$, the initial state is chosen as $\boldsymbol{x}(k) = [1, -1]^{\mathrm{T}}$, learning rate $\alpha = 0.05$. For each iterative step, the critic network and the action network are trained for 100 iteration steps so as to guarantee the neural network training error is less than 10^{-6}.

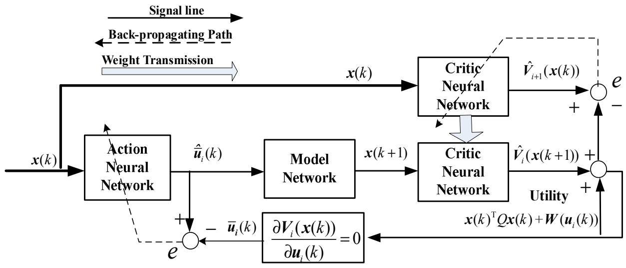

Figure 2. The structure diagram of the iterative ADP algorithm using Neural-Network approximate $V_i(x)$ and $\upsilon_i(x)$

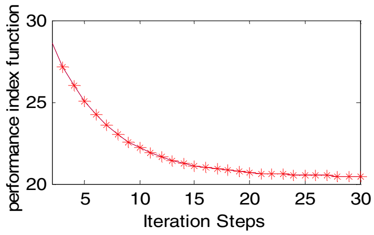

Figure 3. The convergence process of the cost function

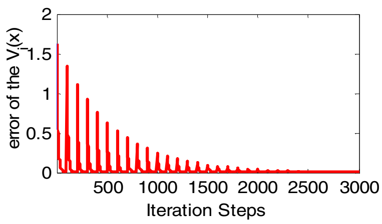

Figure 4. The error of neural-network approximate cost function

Simulation result in Figure 3 shows the iteration convergence process of the cost function $V_i(\boldsymbol{x}(k))$. When $i=30$, $|V_i(\boldsymbol{x}(k))-\boldsymbol{J}^*(\boldsymbol{x}(k))|\le\varepsilon$ holds, thus ε−optimal control law $\boldsymbol{\mu}_\varepsilon^*$ is obtained in finite step $K_\varepsilon(\boldsymbol{x}(k))=i=30$. Besides, Figure 3 also shows that $V_i(\boldsymbol{x}(k))$ satisfy the monotonically nonincreasing and constringency property in Theorem 1 since $V_{i+1}(\boldsymbol{x}(k))\le V_i(\boldsymbol{x}(k))$. Figure 4 shows the error of neural-network approximate cost function. In order to verify the control law $\boldsymbol{\mu}_\varepsilon^*$ obtained by ε-iteration algorithm, the state trajectories and the optimal control trajectories are shows in Figure 5 and Figure 6 for the state $\boldsymbol{x}(k)=[1,-1]^{\mathrm{T}}$, Figure 7 shows the state trajectories in 3d space performance. Notice that the state has been control in stable within finite step $T_f=30$ showed in Figure 6, which satisfies the theory, on the other hand, the control output $|u_1|\le\bar{u}_1=0.5$, $|u_2|\le\bar{u}_2=1$ in Figure 6 shows that control output signal actuator keeps under the constraints $|\boldsymbol{u}|\le\mathbf{A}$.

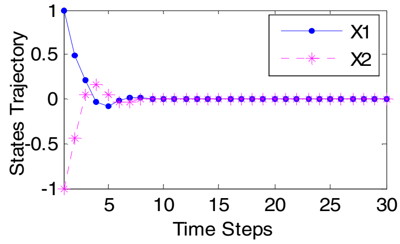

Figure 5. The state trajectories

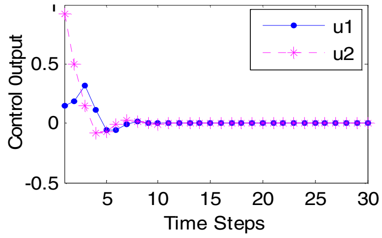

Figure 6. The optimal control trajectories

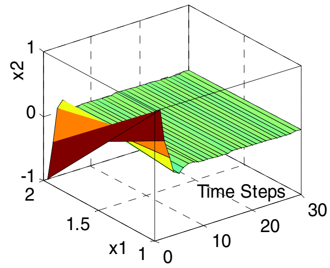

Figure 7. The state trajectories in 3d space performance

B. Comparison of simulation

In order to contrast with the controller with saturating actuator, the state trajectories without actuators saturation is showed in Figure 8 and the optimal control trajectories without actuators saturation is showed in Figure 9. In Figure 9, control output satisfy $|u_1|\ge 0.5$, $|u_2|\ge 1$, thus the control signal will be distorted with saturating actuator, while it keeps under the constraints in Figure 6. Comparing results with Figure 6 and Figure 9, result shows that ε-Iterative adaptive dynamic programming works effective for finite-horizon optimal control scheme for nonlinear systems with saturating actuator.

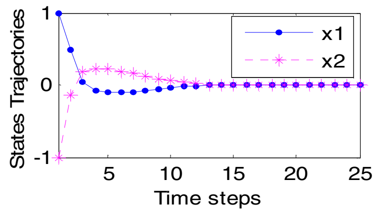

Figure 8. The state trajectories without actuators saturation

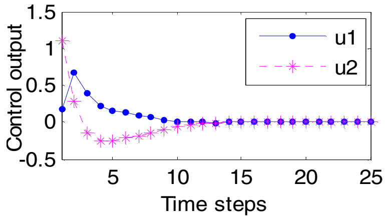

Figure 9. The optimal control trajectories without actuators saturation

VI. CONCLUSION

In this paper, a functional performance index function was used to deal with saturating actuator control with constraints effectively. Furthermore, discrete HJB equation of nonlinear systems was derived. Finite horizon iteration ADP algorithm for control with saturation was developed via mathematical analysis. Finally the finite-horizon optimal control was obtained by the ε−iterative adaptive algorithm.

REFERENCES

[1] Yong-Yan Cao, Zongli Lin, "Stability analysis of discrete-time systems with actuator saturation by a saturation-dependent Lyapunov function," Automatica,Vol.39, 2003, pp. 1235─1241.

[2] Saberi, A Z,Lin,A Teel, "Control of Linear Systems with Saturating Actuators," IEEE Transactions on Automatic Control, Vol 41, NO.3, March 1996, pp. 368─378.

[3] Sussmann H,E D Sontag,Y Yang, "A General Result on the Stabilization of Linear Systems Using Bounded Controls," IEEE Trans. Automatic Control, Vol.39, No.12, December 1994, pp. 2411─2425.

[4] Marc Jungers a, Eugênio B. Castelan, "Gain-scheduled output control design for a class of discrete-time nonlinear systems with saturating actuators," Systems & Control Letters, Vol.60, 2011, 169─173.

[5] Lyshevski, SE, "Optimal Control of Nonlinear Continuou Time Systems: Design of Bounded Controllers ViaGen-ralized Nonquadratic Functionals," American Control Conference, June 1998, pp.205─209.

[6] Werbos P J, "Approximate dynamic programming for reall controlll and neural modeling．Handbook of Intelligent Control," Neural Fuzzy and Adaptive Approaches．New York：Van No strand Reinhold, 1992

[7] Wei Qing-lai, Zhang Huang-guang, Liu De-rong, Zhao Yan, "An Optimal Control Scheme for a Class of Discrete-time Nonlinear Systems with Time Delays Using Adaptive Dynamic Programming," Acta Automatica Sinica, Vol 36, NO.1, 2010, pp. 121─129.

[8] Al-Tamimi, F L. Lewis, and M Abu-Khalaf, "Model Free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control," Automatica, Vol. 43, NO.3, Mar 2007, pp473─481.

[9] Luo Yan-Hong, "Research on Adaptive and Optimal Control for Nonlinear Systems Based on Neural Networks," Northerestern University, 2008, pp.33─59.

[10] Wang Fei-Yue, Jin Ning, Liu Derong, Wei Qing-Lai, "Adaptive Dynamic Programming for Finite-Horizon Optimal Control of Discrete-Time Nonlinear Systems With ε-Error Bound," Transactions on Neural Networks, Vol. 22, 2011, pp. 24─36.

[11] Wang Fei-Yue, Jin Ning, Liu Derong, Hou Zeng-Guang, "adaptive Dynamic Programming with epsilon-Error Bound for Nonlinear Discrete-Time Systems Using Neural Networks," Transactions on Neural Networks, 2008.

[12] Abu-Khalaf M, Lewis F L, " Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach," Automatica 2005, Vol 41, NO.5, pp. 779–791.