Policy Iteration Algorithm for Constrained Cost Optimal Control of Discrete-Time Nonlinear System

Tao Li, Qinglai Wei, Hongyang Li
The State Key Laboratory for Management and Control of Complex Systems
Institute of Automation, Chinese Academy of Sciences
Beijing, China
School of Artificial Intelligence,
University of Chinese Academy of Sciences
Beijing, China
litao2019@ia.ac.cn; qinglai.wei@ia.ac.cn; lihongyang2019@ia.ac.cn

Ruizhuo Song
School of Automation,
University of Science and Technology Beijing
Beijing, China
ruizhuosong@ustb.edu.cn

2021 International Joint Conference on Neural Networks (IJCNN) | 978-1-6654-3900-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/IJCNN52387.2021.9533395

***Abstract*—In this paper, optimal control problems with constraints on summation of auxiliary utility function are called constrained cost optimal control problems and a constrained cost policy iteration adaptive dynamic programming (ADP) algorithm is developed to solve constrained cost optimal control problems for discrete-time nonlinear systems. A convergence analysis is developed to guarantee that the iterative value functions nonincreasingly convergent to the approximate optimal value function. It is also proven that any of the iterative control policy is feasible and can stabilize the nonlinear systems. Finally, a simulation example is given to illustrate the performance of the developed constrained cost policy iteration algorithm.**

***Index Terms*—Adaptive dynamic programming (ADP), reinforcement learning, constrained cost optimal control, policy iteration.**

I. Introduction

Adaptive dynamic programming (ADP) algorithms were proposed in [1] and [2] to overcome the curse of dimensionality [3]. There are two main iterative ADP algorithms, which are value iteration ADP (value iteration for brief) algorithm [4] and policy iteration ADP (policy iteration for brief) algorithm [5].

Policy iteration algorithm for discrete-time systems was proposed in [5]. Liu *et al.* proved that for discrete-time nonlinear systems, the iterative value functions are monotonically nonincreasing and convergent to the solution of the Bellman equation [5]. In 2015, a generalized policy iteration algorithm was proposed in [6] and the admissibility of the iterative control policy are proved. In 2017, a local policy iteration algorithm was proposed in [7] to reduce the computational complexity of the policy iteration algorithm. Zhang *et al.* applied policy iteration algorithm to solve discrete-time nonzero-sum games for multiplayer [11]. In recent years, the policy iteration algorithm has received more and more attention from researchers [12]–[14].

However, almost all the discussions on the policy iteration algorithm are interested in the single performance index function [5]–[7]. Many engineering probelms require describing the goals of a system by two or more performance indices, rather than the single performance index function employed in classical optimal control [8]–[10]. Using several performance indices provides more flexibility to represent the expected behavior of the system in ways that are difficult to express otherwise. In this paper, we consider the constrained cost optimal problem for discrete-time nonlinear systems, where in addition to its standard performance index function, the control policy must satisfy constraints on summation of auxiliary utility function. Examples of these applications can be found in [15]. Yinlam Chow *et al.* [16] formulated the problem of safe reinforcement learning as a constrained Markov decision problems (CMDPs) and proposed a novel Lyapunov approach for solving them. To the best of our knowledge, there are still no discussions focused on the policy iteration adaptive dynamic programming algorithms for constrained cost optimal control problem for discrete-time nonlinear systems, which motives our research.

In this paper, a new constrained cost policy iteration (CCPI) algorithm is developed to solve undiscounted and constrained optimal control problems of discrete-time nonlinear systems. First, the CCPI algorithm is introduced to find the approximate optimal control policy under constraints on summation of auxiliary utility function. Second, the convergence properties of the iterative value functions are analyzed and it will show that any of the iterative control policies can satisfy the constraint condition. Furthermore, an effective method is developed to obtain the initial feasible control policy. In numerical examples, the control results by the CCPI algorithm will be compared with the traditional policy iteration to show the effectiveness of the developed algorithm.

This work was supported in part by the National Key Reseanch and Development Program of China (2018YFB1702300, 2018AAA0101502), and in part by the National Natural Science Foundation of China (62073321, 61873300).

978-0-7381-3366-9/21/$31.00 ©2021 IEEE

The paper is organized as follows. In Section II, the problem formulations are presented. In Section III, the CCPI algorithm is derived. The statbility and feasibility of the iterative control policy and the convergence properties of the iterative value functions are also presented in this section. Then an effective method is developed to obtain the initial admissible and feasible initial control policy. In Section IV, the neural network implementation for the optimal control scheme is discussed. In Section V, the numerical results and analyses are presented to demonstrate the effectiveness of the CCPI algorithm. Finally, in Section VI, the conclusion is drawn.

II. Problem Formulation

In this paper, we will study the following deterministic discrete-time systems:

$$x_{k+1} = F(x_k, u_k) \quad k = 0, 1, 2, \ldots, \tag{1}$$

where $x_k \in \mathbb{R}^n$ is the n-dimensional state vector and $u_k \in \mathbb{R}^m$ is the m-dimensional control vector. Let x_0 be the initial state and $F(x_k, u_k)$ be the system function.

The control action is determined as a function of the state, i.e., $u_k = u(x_k)$. Such a mapping $u(\cdot) : \mathbb{R}^n \to \mathbb{R}^m$ is called a control policy. For a given control policy $\mu(\cdot)$, let $\underline{x}_0 = \{x_0, x_1, \ldots\}$ be the sequence of states and $\underline{\mu}(x_0) = \{\mu(x_0), \mu(x_1), \ldots\}$ be the sequence of controls, the performance index function of $\mu(\cdot)$ is defined as

$$J_\mu(x_0) = \sum_{k=0}^{\infty} U(x_k, \mu(x_k)), \tag{2}$$

where $U(x_k, u_k)$ is a positive definite utility function for $\forall x_k, u_k$. And the cost function of $\mu(\cdot)$ is defined as

$$D_\mu(x_0) = \sum_{k=0}^{\infty} d(x_k, \mu(x_k)), \tag{3}$$

where $d(x_k, u_k)$ is a positive semi-definite constrained utility function for $\forall x_k, u_k$.

For convenience of analysis, results of this paper are based on the following assumption.

Assumption 1: The origin $x_k = 0$ is an equilibrium state of system (1) under the control $u_k = 0$, i.e., $F(0, 0) = 0$; the feedback control $u_k = u(x_k)$ satisfies $u_k = u(x_k) = 0$ for $x_k = 0$; the system function $F(x_k, u_k)$ is Lipschitz continuous on a compact set $\Omega \subset \mathbb{R}^n$ containing the origin; the system (1) is controllable on Ω.

We will study optimal control problem for (1) with constraints on cost function. The goal of this paper is to find an optimal control policy $\mu^*(\cdot)$, which stabilizes system (1), simultaneously minimizes the performance index function (2) and satisfies the constraint condition

$$D_{\mu^*}(x_0) \leq d_0, \tag{4}$$

where $d_0 > 0$ is an upper bound for the cost function. As (1) is controllable, there exists a stable control policy $\mu(\cdot)$, that moves x_0 to zero. Let $\mathscr{U}_s$ denote the set which contains all the stable control policies and $\mathscr{U}_c$ denote the set which contains all the control policies that satisfy

$$D_\mu(x_0) \leq d_0. \tag{5}$$

The goal of this paper is to solve

$$\mu^*(\cdot) = \underset{\mu \in \{\mathscr{U}_s \cap \mathscr{U}_c\}}{\arg\min} \{J_\mu(x_0) : D_\mu(x_0) \leq d_0\}, \tag{6}$$

where $\mu^*(\cdot)$ is the optimal control policy for the constrained cost optimal control problem.

III. Constraint Policy Iteration Algorithm

In this section, the CCPI algorithm is developed to solve the constrained cost optimal control problems. Stability proofs will be given to show that any of the iterative control policies can stabilize the nonlinear system. Feasibility proofs will be given to show that any of the iterative control policies satisfies the constraint condition. Convergence proofs will be given to show that the iterative value functions will converge.

Before starting, the definition of admissible and feasible policy is necessary. For the optimal control problems, the control policy must not only stabilize the control systems, but also make the performance index function finite, i.e., the admissible control policy [17].

Definition 1: A control policy $\mu(\cdot)$ is defined to be admissible with respect to (2) on Ω if $\mu(\cdot)$ is continuous on Ω, $\mu(0) = 0$, $\mu(x_k)$ stabilizes (1) on Ω, and $\forall x_0 \in \Omega$, $J_\mu(x_0)$ is finite.

For the constrained cost optimal problem, the control policy is not only admissible, but also satisfies the constraint condition (5), i.e., the feasible control policy.

Definition 2: A control policy $\mu(\cdot)$ is defined to be feasible with respect to (1) on Ω if $\mu(\cdot)$ is admissible and $\forall x_0 \in \Omega$, $D_\mu(x_0) \leq d_0$.

A. Derivation of the Constrained Cost Policy Iteration Algorithm

For convenience of analysis, define an operator w.r.t. an admissible control policy $\mu(\cdot)$ and a generic utility function $h(x, a)$,

$$T_{\mu,h}[V](x_k) = h(x_k, \mu(x_k)) + V(F(x_k, \mu(x_k))), \tag{7}$$

where function V is a mapping from $\mathbb{R}^n$ to $\mathbb{R}$, i.e., $V : \mathbb{R}^n \mapsto \mathbb{R}$. We denote by $T^i_{\mu,h}$ the composition of the mapping $T_{\mu,h}$ with itself i times.

Now, we give the CCPI algorithm as follows. Let $\mu_0(\cdot)$ be an arbitrary feasible control policy. For $i = 0$, compute the performance index function of $\mu_0(\cdot)$,

$$V_{\mu_0}(x_k) = \lim_{j \to \infty} T^j_{\mu_0,U}[\Psi](x_k), \tag{8}$$

and the cost function of $\mu_0(\cdot)$,

$$D_{\mu_0}(x_k) = \lim_{j \to \infty} T^j_{\mu_0,d}[\Psi](x_k). \tag{9}$$

Then the iterative control policy is computed by

$$\mu_1(x_k) = \underset{\mu(x_k) \in M_{\mu_0}(x_k)}{\arg\min} T_{\mu,U}[V_{\mu_0}](x_k), \tag{10}$$

where $M_{\mu_0} = \{\mu(\cdot) \in \mathscr{U}_a : T_{\mu,d}[D_{\mu_0}(x_k)] \leq D_{\mu_0}(x_k)\}$. For $\forall i = 1, 2, \ldots$, let $V_{\mu_i}(x_k)$ be the iterative value function constructed by $\mu_i(x_k)$, which satisfy the following GHJB equations:

$$V_{\mu_i}(x_k) = U(x_k, \mu_i(x_k)) + V_{\mu_i}(F(x_k, \mu_i(x_k))). \quad (11)$$

$V_{\mu_i}(x_k)$ is computed by

$$V_{\mu_i}(x_k) = \lim_{j\to\infty} T^j_{\mu_i,U}[\Psi](x_k) \quad (12)$$

The iterative control policy is updated by

$$\mu_{i+1}(x_k) = \arg\min_{\mu(x_k)\in M_{\mu_0}(x_k)} T_{\mu,U}[V_{\mu_i}](x_k) \quad (13)$$

In the following section, we will show the properties of the CCPI algorithm.

B. Properties of the Constraint Policy Iteration Algorithm

The monotonicity and convergence of the iterative value function, as well as the feasibility of the iterative control law will be derived.

Theorem 1: For $i = 0, 1, \ldots$, letting $V_{\mu_i}(x_k)$ and $\mu_i(x_k)$ be obtained by the constrained cost policy iteration algorithm (8)–(10) and (12)–(13), $\mu_0(\cdot)$ is feasible, then the iterative value function $V_{\mu_i}(x_k)$ is monotocially non-increasing convergent as i increases and for any $i = 0, 1, \ldots$, the iterative control law $\mu_i(x_k)$ is a feasible control law.

Proof The statement is proven by mathematical induction. First, for $i = 0$, as $\mu_0(x_k)$ is a feasible control law, then we can derive

$$\begin{aligned} T_{\mu_0,U}[V_{\mu_0}](x_k) &= U(x_k, \mu_0(x_k)) + V_{\mu_0}(F(x_k, \mu_0(x_k))) \\ &= V_{\mu_0}(x_k), \end{aligned} \quad (14)$$

and

$$\begin{aligned} T_{\mu_0,U}[D_{\mu_0}](x_k) &= d(x_k, \mu_0(x_k)) + D_{\mu_0}(F(x_k, \mu_0(x_k))) \\ &= D_{\mu_0}(x_k). \end{aligned} \quad (15)$$

Consider $i = 1$. According to (10), it is known that

$$\begin{aligned} \mu_1(x_k) = \arg\min_{\mu(x_k)} & \; T_{\mu,U}[V_{\mu_0}](x_k) \\ \text{s.t.} \quad & T_{\mu,d}[D_{\mu_0}](x_k) \leq D_{\mu_0}(x_k). \end{aligned} \quad (16)$$

Then, for all $x_k \in \Omega$, it can be derived that

$$\begin{aligned} &T_{\mu_1,U}[V_{\mu_0}](x_k) \\ &\quad = U(x_k, \mu_1(x_k)) + V_{\mu_0}(F(x_k, \mu_1(x_k))) \\ &\quad = \min_{\mu(x_k)} \{T_{\mu,U}[V_{\mu_0}](x_k) | T_{\mu,d}[D_{\mu_0}](x_k) \leq D_{\mu_0}(x_k)\} \\ &\quad \leq T_{\mu_0,U}[V_{\mu_0}](x_k) \\ &\quad = V_{\mu_0}(x_k). \end{aligned} \quad (17)$$

It can also be derive that

$$\begin{aligned} T_{\mu_1,U}[V_{\mu_0}](x_k) &= U(x_k, \mu_1(x_k)) + V_{\mu_0}(x_{k+1}) \\ &\geq U(x_k, \mu_1(x_k)) + T_{\mu_1,U}[V_{\mu_0}](x_{k+1}) \\ &\geq \sum_{j=0}^{\infty} U(x_{k+j}, \mu_1(x_{k+j})) \\ &\quad + \lim_{N\to\infty} T_{\mu_1,U}[V_{\mu_0}](x_{k+N}). \end{aligned} \quad (18)$$

For $x_k \in \Omega$, it is known that $V_{\mu_0}(x_k)$ and $\mu_1(x_k)$ are both finite and hence $0 \leq T_{\mu_1,U}[V_{\mu_0}](x_k) < \infty$. It implies that

$$0 \leq \sum_{j=0}^{\infty} U(x_{k+j}, \mu_1(x_{k+j})) < \infty \quad (19)$$

and $x_{k+N} \to 0$ as $N \to \infty$. Thus, it is shown that $\mu_1(x_k)$ is an admissible control law.

According to (16), the iterative control law $\mu_1(x_k)$ satisfies

$$T_{\mu_1,d}[D_{\mu_0}](x_k) \leq D_{\mu_0}(x_k). \quad (20)$$

Then we have

$$\begin{aligned} D_{\mu_0}(x_k) &\geq T_{\mu_1,d}[D_{\mu_0}](x_k) \\ &\geq \lim_{i\to\infty} T^i_{\mu_1,d}[D_{\mu_0}](x_k) \\ &= D_{\mu_1}(x_k). \end{aligned} \quad (21)$$

Thus, we have

$$D_{\mu_1}(x_k) \leq D_{\mu_0}(x_k) \leq d_0. \quad (22)$$

Therefore, the iterative control law $\mu_1(x_k)$ is feasible.

According to (17) and (18), we have

$$\begin{aligned} V_{\mu_0}(x_k) &\geq T_{\mu_1,U}[V_{\mu_0}](x_k) \\ &\geq \sum_{j=0}^{\infty} U(x_{k+j}, \mu_1(x_{k+j})) + \lim_{N\to\infty} T_{\mu_1,U}[V_{\mu_0}](x_{k+N}) \\ &\geq \sum_{j=0}^{\infty} U(x_{k+j}, \mu_1(x_{k+j})) \\ &= V_{\mu_1}(x_k). \end{aligned} \quad (23)$$

Assume that the statement is true for $i = l$. As $\mu_l(x_k)$ is a feasible control law, we can derive

$$\begin{aligned} T_{\mu_l,U}[V_{\mu_l}](x_k) &= U(x_k, \mu_l(x_k)) + V_{\mu_l}(F(x_k, \mu_l(x_k))) \\ &= V_{\mu_l}(x_k), \end{aligned} \quad (24)$$

and

$$D_{\mu_l}(x_k) \leq d_0. \quad (25)$$

Similar to the proof for $i = 0$, we can prove that $\mu_{l+1}(x_k)$ is feasible and

$$\begin{aligned} V_{\mu_l}(x_k) \geq & T_{\mu_{l+1},U}[V_{\mu_l}](x_k) \\ \geq & \sum_{j=0}^{\infty} U(x_{k+j}, \mu_{l+1}(x_{k+j})) \\ & + \lim_{N\to\infty} T_{\mu_{l+1},U}[V_{\mu_l}](x_{k+N}) \\ \geq & \sum_{j=0}^{\infty} U(x_{k+j}, \mu_{l+1}(x_{k+j})) \\ = & V_{\mu_{l+1}}(x_k). \end{aligned} \quad (26)$$

Therefore, we have $V_{\mu_{i+1}}(x_k) \leq V_{\mu_i}(x_k), \forall i \geq 0$ and the iterative control law $\mu_i(x_k)$ is feasible. Let $J^{*\prime}(x_k)$ be the

optimal performance index function of unconstrained optimal control problem, i.e.,

$$J^{*\prime}(x_k) = \min_{\mu \in \mathcal{U}_s} J_\mu(x_k) \tag{27}$$

and $J^*(x_k)$ be the be the optimal performance index function of constraint optimal control problem, i.e.,

$$J^*(x_k) = \min_{\mu \in \mathcal{U}_s} \{J_\mu(x_k) : D_\mu(x_k) \le d_0\}. \tag{28}$$

Then

$$V_{\mu_i}(x_k) \ge J^*(x_k) \ge J^{*\prime}(x_k). \tag{29}$$

Then $\{V_{\mu_i}(x_k)\}$ is a monotonically nonincreasing sequence and is lower bounded by $J^{*\prime}(x_k)$, therefore the iterative function $V_{\mu_i}(x_k), \forall x_k \in \mathbb{R}^n$ is convergent. The proof is complete. ■

C. Obtaining the Initial Feasible Control Policy

Theorem 2: Suppose Assumption 1 holds. Let $\Psi(x_k) \ge 0$ be an arbitrary semipositive definite function. Let $\mu(\cdot)$ be an arbitrary control policy for system (1), which satisfies $\mu(0) = 0$. Then, $\mu(\cdot)$ is an feasible control policy if and only if $\lim_{j\to\infty} T^j_{\mu,U}[\Psi](x_k)$ and $\lim_{j\to\infty} T^j_{\mu,d}[\Psi](x_k)$ exist and $\lim_{j\to\infty} T^j_{\mu,d}[\Psi](x_k) \le d_0$.

Proof According to Theorem 3.3 in [5], $\mu(\cdot)$ is an admissiable control policy if and only if $\lim_{j\to\infty} T^j_{\mu,U}[\Psi](x_k)$ exists. Then according to Definition 2, $\mu(\cdot)$ is feasible if and only if $\lim_{j\to\infty} T^j_{\mu,d}[\Psi](x_k) \le d_0$.

According to Theorem 2, we can establish an effective iteration algorithm by repeating experiments using neural networks. The detailed implimentation of the iteration algorithm is expressed in Algorithm 1.

D. Summary of the Constraint Policy Iteration Algorithm

According to the above preparations, we can summarize the discrete-time CCPI algorithm in Algorithm 2.

IV. Neural Network Implementation

In this paper, BP neural networks are used to approximate μ_i, $V_i(x_k)$ and $D_i(x_k)$, respectively. Here, there are three networks, which are critic network, action network and cost critic network, respectively. All networks are chosen as three-layer feedforward neural network. The whole structure diagram is shown in Fig.1.

V. Simulation Studies

We now examine the performance of the developed algorithm in the following discrete-time nonlinear system

$$x_{k+1} = h(x_k) + g(x_k)\, u_k, \tag{30}$$

where

$$h(x_k) = [0.9x_{1k} + 0.1x_{2k},$$
$$-0.05\left(x_{1k} + x_{2k}\left(1 - (\cos(2x_{1k}) + 2)^2\right)\right) + x_{2k}]^\top,$$
$$g(x_k) = \begin{bmatrix} 0 \\ 0.1\cos(2x_{1k}) + 0.2 \end{bmatrix}$$

Algorithm 1 Obtain the Initial Feasible Control Policy

Initialization:

Choose a semi-positive definite function $\Psi(x_k) \ge 0$;
Initialize four neural networks $cnet1$, $cnet2$, $dnet1$ and $dnet2$ with small random weights;
Let $\Phi_0(x_k) = \Psi(x_k)$;
Give the max iteration of computation $i_{\max}$.
Choose a computation precision ε;

Iteration:

1: Establish a neural network (action network for brief) with small random weights to generate an initial control policy $\mu(\cdot)$ with $\mu(x_k) = 0$ for $x_k = 0$;

Determine whether the initial control policy is admissible

2: Let $i = 0$. Train the critic network $cnet1$ to approximate $\Phi_1(x_k)$, where $\Phi_1(x_k)$ satisfies

$$\Phi_1(x_k) = U(x_k, \mu(x_k)) + \Phi_0(x_{k+1});$$

3: Copy $cnet1$ to $cnet2$.
4: Let $i = i + 1$. Use $cnet2$ to get $\Phi_i(x_{k+1})$ and train the critic network $cnet1$ to approximate $\Phi_{i+1}(x_k)$, where $\Phi_{i+1}(x_k)$ satisfies

$$\Phi_{i+1}(x_k) = U(x_k, \mu(x_k)) + \Phi_i(x_{k+1});$$

5: Use $cnet1$ to get $\Phi_{i+1}(x_k)$ and use $cnet2$ to get $\Phi_i(x_k)$. If $|\Phi_{i+1}(x_k) - \Phi_i(x_k)| < \varepsilon$, then goto Step 7. Else goto next step;
6: If $i > i_{\max}$, then goto Step 1. Else goto Step 3;

Determine whether the initial control policy is feasible

7: Let $i = 0$. Train the cost critic network $dnet1$ to approximate $\Phi_1(x_k)$, where $\Phi_1(x_k)$ satisfies

$$\Phi_1(x_k) = d(x_k, \mu(x_k)) + \Phi_0(x_{k+1});$$

8: Copy $dnet1$ to $dnet2$.
9: Let $i = i+1$. Use $dnet2$ to get $\Phi_i(x_{k+1})$ and train the cost critic network $dnet1$ to approximate $\Phi_{i+1}(x_k)$ $\Phi_{i+1}(x_k)$, where $\Phi_{i+1}(x_k)$ satisfies

$$\Phi_{i+1}(x_k) = d(x_k, \mu(x_k)) + \Phi_i(x_{k+1});$$

10: Use $dnet1$ to get $\Phi_{i+1}(x_k)$ and use $dnet2$ to get $\Phi_i(x_k)$. If $|\Phi_{i+1}(x_k) - \Phi_i(x_k)| < \varepsilon$, then goto Step 12. Else goto next step;
11: If $i > i_{\max}$, then goto Step 1. Else goto Step 8;
12: Use $dnet1$ to get $\Phi_{i+1}(x_k)$. If $\Phi_{i+1}(x_k) \le d_0$, then goto Step 13, else goto Step 1;
13: **return** $\mu(x_k)$ and let $\mu_0(x_k) = \mu(x_k)$.

$x_k = [x_{1k}, x_{2k}]^\top \in \mathbb{R}^2$, and $u_k \in \mathbb{R}, k = 0, 1, \ldots$. The initial state is $x_0 = [-0.87, 0.97]^\top$. The upper limit of the constraint function d_0 is set to 3.7. The utility function is the quadratic form that is expressed as $U(x_k, u_k) = x_k^\top Q x_k + u_k^\top R u_k$, where $Q = 0.1I$, $R = 0.1I$, and I is the identity matrix

Algorithm 2 Discrete-Time Constraint Policy Iteration Algorithm

Initialization:

Choose randomly an array of initial states x_0;
Choose a computation precision ε;
Give the initial feasible control policy μ_0;
Give the max iteration of computation $i_{\max}$.

Iteration:

Let the iteration index $i = 0$;

1: Construct the initial iterative value function $V_{\mu_0}(x_k)$ and the iterative cost function $D_{\mu_0}(x_k)$ according to μ_0 by

$$V_{\mu_0}(x_k) = U(x_k, \mu_0(x_k)) + V_{\mu_0}(x_{k+1})$$

and

$$D_{\mu_0}(x_k) = d(x_k, \mu_0(x_k)) + D_{\mu_0}(x_{k+1});$$

2: Update the iterative control policy by

$$\mu_1(x_k) = \underset{\mu(x_k) \in M_{\mu_0}(x_k)}{\arg\min} \{U(x_k, u_k) + V_{\mu_0}(x_{k+1})\};$$

3: Let $i = i + 1$. Construct the iterative value function $V_{\mu_i}(x_k)$, which satisfies

$$V_{\mu_i}(x_k) = U(x_k, \mu_i(x_k)) + V_{\mu_i}(F(x_k, \mu_i(x_k)));$$

4: Update the iterative control policy μ_{i+1} by

$$\mu_{i+1}(x_k) = \underset{\mu(x_k) \in M_{\mu_0}(x_k)}{\arg\min} \{U(x_k, \mu(x_k)) + V_{\mu_i}(x_{k+1})\};$$

5: If $|V_{i-1}(x_k) - V_i(x_k)| < \varepsilon$, goto Step 7. Else goto Step 6.

6: If $i < i_{\max}$, then goto Step 3. Else goto Step 8.

7: **return** $\mu_i(x_k)$ and $V_i(x_k)$.

8: **return** The algorithm dosen't converge within $i_{\max}$ iterations.

with suitable dimensions. The constrained utility function is a nonquadratic form, where the constraint utility function is expressed as

$$d(x_k, u_k) = \ln(x_k^\top Q x_k + 1) + \ln(u_k^\top R u_k + 1),$$

where $Q = 0.01I$ and $R = 0.6I$.

To implement the developed CCPI algorithm, we choose three-layer feedforward neural networks as function approximation structures. The structures of the critic, action and cost critic neural networks are both chosen as 2–12–1. The maximum number of iteration steps is selected as $i_{\max} = 20$. The compact set Ω or the opreation region of the system is selected as $-1 \le x_1 \le 1$ and $-1 \le x_2 \le 1$. The training set $\{x_k\}$ is constructed by randomly choosing 500 samples from the compact set Ω at each iteration. For each iteration step, these networks are trained for 20000 steps using the learning rate of $\alpha = 0.01$ so that the neural network training error becomes less than 10^{-5}. The cost function of the initial control policy $D_{\mu_{\text{initial}}}(x_0) = 3.64 \le d_0$, so the initial control policy is feasible.

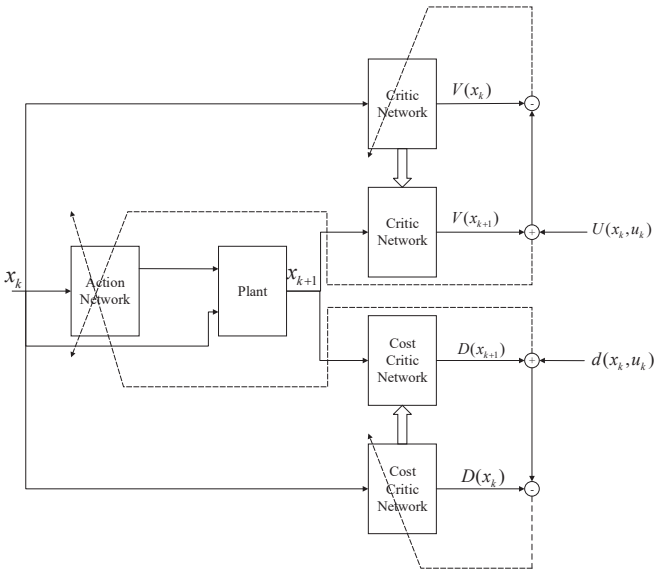

Fig. 1. Structure diagram of the algorithm

Implement the CCPI algorithm for 3 iterations to reach the comoputation precision $\varepsilon = 0.005$. To show the effectiveness of the developed CCPI algorithm, Implement the policy iteration algorithm for four iterations to reach the comoputation precision $\varepsilon = 0.01$.

For these two algorithms, the convergence trajectories of the iterative value functions are shown in Fig.2 (a), and the convergence trajectories of the cost functions of the iterative control policies are shown in Fig.2 (b). During each iteration, the iterative control policy is updated. We obtain the final control policy after convergence of the algorithm. Applying the final control policy to the given system for $T_f = 50$ time steps, we can obtain the states and controls trajectory, which are shown in Fig.3 (a), (b) and (c), respectively.

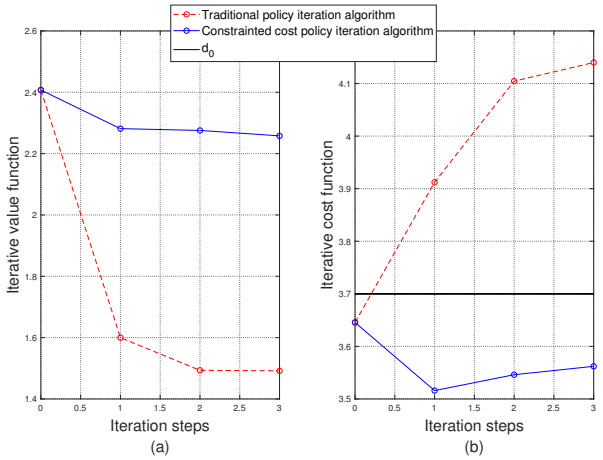

Fig. 2. Numerical results using CCPI. (a) Convergence trajectory of iterative value function. (b) Convergence trajectory of iterative cost function

For tradition policy iteration algorithm, we can see that

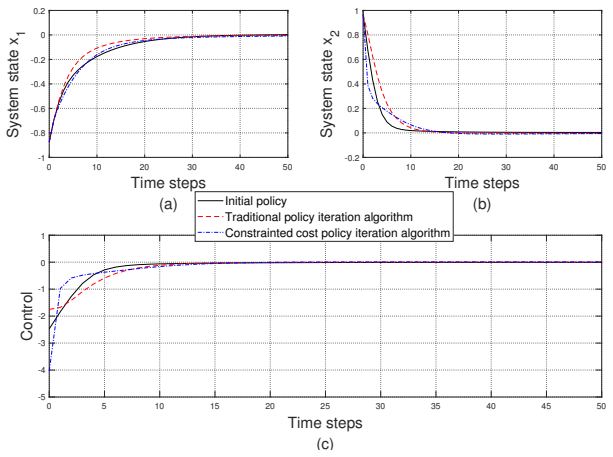

Fig. 3. (a) Trajectory of state x_1. (b) Trajectory of state x_2. (c) Trajectory of control action.

the optimal control policy of the unconstrained cost optimal control problem is obtained after 3 iterations, but the iterative cost function exceeds d_0 because the policy iteration algorithm completely ignores the constraints. The suboptimal performance index function is obtained by the CCPI algorithm after 3 iterations and any of the iterative control policy is feasible. The iterative cost function is always less than d_0 during the iteration process in CCPI algorithm. This example shows that the CCPI algorithm has convergence and feasibility on nonlinear systems.

VI. Conclusion

In this paper, a constrained cost policy iteration adaptive dynamic programming (ADP) algorithm is developed to solve infinite horizon undiscounted constrained cost optimal control problems for discrete-time nonlinear systems. A convergence analysis is developed to guarantee that the iterative value function is nonincreasingly convergent to the suboptimal performance index function. It is also proven that any of the iterative control policy is feasible and can stabilize the nonlinear systems. Finally, a simulation example is given to illustrate the performance of the present method.

Acknowledgment

References

[1] P. J. Werbos, "Advanced forecasting methods for global crisis warning and models of intelligence," *General Syst. Yearbook*, vol. 22, pp. 25–38, Jan. 1977.

[2] P. J. Werbos, "A menu of designs for reinforcement learning over time," in *Neural Networks for Control*, W. T. Miller, R. S. Sutton, and P. J. Werbos, Ed., Cambridge, MA, USA: MIT Press, 1991, pp. 67–95.

[3] R. E. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton Univ. Press, 1957.

[4] Q. Wei, D. Liu and H. Lin, "Value iteration adaptive dynamic programming for optimal control of discrete-time nonlinear system," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 840–853, Mar. 2016.

[5] D. Liu, Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 3, pp. 621–634, 2013.

[6] D. Liu, Q. Wei and P. Yan, "Generalized policy iteration adaptive dynamic programming for discrete-time nonlinear systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 12, pp. 1577–1591, Dec. 2015.

[7] Q. Wei, D. Liu, Q. Lin and R. Song, "Discrete-time optimal control via local policy iteration adaptive dynamic programming," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3367–3379, Oct. 2017.

[8] V. G. Lopez and F. L. Lewis, "Dynamic multiobjective control for continuous-time systems using reinforcement learning," *IEEE Trans. Autom. Control*, vol. 64, no. 7, pp. 2869–2874, July 2019.

[9] C. Wu, B. Chen and W. Zhang, "Multiobjective H_2/H_∞ control design of the nonlinear mean-field stochastic jump-diffusion systems via fuzzy approach," *IEEE Trans. Fuzzy Syst.*, vol. 27, no. 4, pp. 686–700, Apr. 2019.

[10] H. Han, Z. Liu, Y. Hou and J. Qiao, "Data-driven multiobjective predictive control for wastewater treatment process," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2767–2775, Apr. 2020.

[11] H. Zhang, H. Jiang, C. Luo and G. Xiao, "Discrete-time nonzero-sum games for multiplayer using policy-iteration-based adaptive dynamic programming algorithms," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3331–3340, Oct. 2017.

[12] W. Guo, J. Si, F. Liu and S. Mei, "Policy approximation in policy iteration approximate dynamic programming for discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2794–2807, July 2018.

[13] M. Liang, D. Wang and D. Liu, "Neuro-optimal control for discrete stochastic processes via a novel policy iteration algorithm," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 11, pp. 3972–3985, Nov. 2020.

[14] S. He, H. Fang, M. Zhang, F. Liu and Z. Ding, "Adaptive optimal control for a class of nonlinear systems: the online policy iteration approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 2, pp. 549–558, Feb. 2020.

[15] E. Altman, *Constrained Markov decision processes*, Florida, USA:CRC Press, 1999.

[16] Y. Chow, O. Nachum and et al, "A lyapunov-based approach to safe reinforcement learning," in *Proceeding of Advances in Neural Information Processing Systems 31*, 2018.

[17] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming," *IEEE Trans. Syst., Man, Cybern., Cybern.*, vol. 38, no. 4, pp. 943-949, Aug. 2008.