Off-Policy Reinforcement Learning for $ H_infty $ Control Design

Article *in* IEEE Transactions on Cybernetics · January 2015

DOI: 10.1109/TCYB.2014.2319577 · Source: PubMed

Huai-Ning Wu
Beihang University

Tingwen Huang
Texas A&M University at Qatar

arXiv:1311.6107v2 [cs.SY] 19 Jan 2014

Off-policy reinforcement learning for H_∞ control design

Biao Luo[a], Huai-Ning Wu[a], Tingwen Huang[b]

[a]*Science and Technology on Aircraft Control Laboratory, Beihang University (Beijing University of Aeronautics and Astronautics), Beijing 100191, P. R. China*

[b]*Texas A& M University at Qatar, PO Box 23874, Doha, Qatar*

Abstract

The H_∞ control design problem is considered for nonlinear systems with unknown internal system model. It is known that the nonlinear H_∞ control problem can be transformed into solving the so-called Hamilton-Jacobi-Isaacs (HJI) equation, which is a nonlinear partial differential equation that is generally impossible to be solved analytically. Even worse, model-based approaches cannot be used for approximately solving HJI equation, when the accurate system model is unavailable or costly to obtain in practice. To overcome these difficulties, an off-policy reinforcement leaning (RL) method is introduced to learn the solution of HJI equation from real system data, and its convergence is proved. In the off-policy RL method, the system data can be generated with arbitrary policies rather than the evaluating policy, which is extremely important and promising for practical systems. For implementation purpose, a neural network (NN) based actor-critic structure is employed and a least-square NN weight update algorithm is derived based on the method of weighted residuals. Finally, the developed NN-based off-policy RL method is tested on a linear F16 aircraft plant, and further applied to a rotational/translational actuator system.

Keywords: H_∞ control design; Reinforcement learning; Off-policy learning; Neural Network; Hamilton-Jacobi-Isaacs equation.

Email addresses: `biao.luo@hotmail.com` (Biao Luo), `whn@buaa.edu.cn` (Huai-Ning Wu), `tingwen.huang@qatar.tamu.edu` (Tingwen Huang)

1. Introduction

Reinforcement learning (RL) is a machine learning technique that has been widely studied from the computational intelligence and machine learning scope in the artificial intelligence community [1–4]. RL technique refers to an actor or agent that interacts with its environment and aims to learn the optimal actions, or control policies, by observing their responses from the environment. In [2] , Sutton and Barto suggested a definition of RL method, i.e., any method that is well suited for solving RL problem can be regarded as a RL method, where the RL problem is defined in terms of optimal control of Markov decision processes. This obviously established the relationship between the RL method and control community. Moreover, RL methods have the ability to find an optimal control policy from unknown environment, which makes RL a promising method for control design of real systems. In the past a few years, many RL approaches [5–14] have been introduced for solving control problems.

For most of practical real systems, the existence of external disturbances is usually unavoidable. To reduce the effects of disturbance, robust controller is required for disturbance rejection. One effective solution is the H_∞ control method, which achieves disturbance attenuation in the L_2-gain setting [15–17], that is, to design a controller such that the ratio of the objective output energy to the disturbance energy is less than a prescribed level. Over the past few decades, a large number of theoretical results on nonlinear H_∞ control have been reported [18–20], where the H_∞ control problem can be transformed into how to solve the so-called Hamilton-Jacobi-Isaacs (HJI) equation. However, the HJI equation is a nonlinear partial differential equation (PDE), which is difficult or impossible to solve, and may not have global analytic solutions even in simple cases.

Thus, some works have been reported to solve the HJI equation approximately [18, 21–26]. In [18], it was shown that there exists a sequence of policy iterations on the control input such that the HJI equation is successively approximated with a sequence of Hamilton-Jacobi-Bellman (HJB)-like equations. Then, the methods for solving HJB equation can be extended for the HJI equation. In [27], the HJB equation was successively approximated by a sequence of linear PDEs, which were solved with Galerkin approximation in [21, 28, 29]. In [30], the successive approximation method was extended to solve the discrete-time HJI equation. Similar to [21], a policy iteration scheme was developed in [22] for the constrained input system. For implementation purpose of this scheme, a neuro-dynamic programming approach was introduced in [31] and an online adaptive method was given in [32]. This approach suits for the case that the saddle point exists, thus a situation that the smooth saddle point does not exist was considered in [33]. In [23], a synchronous policy iteration method was developed, which is the extension of the work [34]. To improve the efficiency for computing the solution of HJI equation, Luo and Wu [35] proposed a computationally efficient simultaneous policy update algorithm (SPUA). In addition, in [36] the solution of the HJI equation was approximated by the Taylor series expansion, and an efficient algorithm was furnished to generate the coefficients of the Taylor series. It is observed that most of these methods are model-based, where the full system model is required. However, the accurate system model is usually unavailable or costly to obtain for many practical systems. Thus, some RL approaches have been proposed for H_∞ control design of linear systems [37, 38] and nonlinear systems [39] with unknown internal system model. But these methods are on-policy learning approaches [23, 32, 37–40], where the cost function should be evaluated by using system data generated with policies being evaluated. It is found that there are several drawbacks (to be discussed in Section 3) to apply the on-policy learning to solve real H_∞ control problem.

To overcome this problem, this paper introduces an off-policy RL method to solve the nonlinear continuous-time H_∞ control problem with unknown internal system model. The rest of the paper is rearranged as follows. Sections 2 and 3 present the problem description and the motivation. The off-policy learning methods for nonlinear systems and linear systems are developed in 4 and 5 respectively. The simulation studies are conducted in Section 6 and a brief conclusion is given in Section 7.

Notations: $\mathbb{R}, \mathbb{R}^n$ and $\mathbb{R}^{n\times m}$ are the set of real numbers, the n-dimensional Euclidean space and the set of all real matrices, respectively. $\|\cdot\|$ denotes the vector norm or matrix norm in $\mathbb{R}^n$ or $\mathbb{R}^{n\times m}$, respectively. The superscript T is used for the transpose and I

denotes the identify matrix of appropriate dimension. $\triangledown \triangleq \partial/\partial x$ denotes a gradient operator notation. For a symmetric matrix M, $M > (\geqslant) 0$ means that it is a positive (semi-positive) definite matrix. $\|v\|_M^2 \triangleq v^T M v$ for some real vector v and symmetric matrix $M > (\geqslant) 0$ with appropriate dimensions. $C^1(\mathcal{X})$ is function space on $\mathcal{X}$ with first derivatives are continuous. $L_2(0,\infty)$ is a Banach space, for $\forall w(t) \in L_2(0,\infty)$, $\int_0^\infty \|w(t)\|^2 dt < \infty$. Let $\mathcal{X}$,$\mathcal{U}$ and $\mathcal{W}$ be compact sets, denote $\mathcal{D} \triangleq \{(x,u,w)|x \in \mathcal{X}, u \in \mathcal{U}, w \in \mathcal{W}\}$. For column vector functions $s_1(x,u,w)$ and $s_2(x,u,w)$, where $(x,u,w) \in \mathcal{D}$ define inner product $\langle s_1(x,u,w), s_2(x,u,w)\rangle_{\mathcal{D}} \triangleq \int_{\mathcal{D}} s_1^T(x,u,w) s_2(x,u,w) d(x,u,w)$ and norm $\|s_1(x,u,w)\|_{\mathcal{D}} \triangleq \left(\int_{\mathcal{D}} s_1^T(x,u,w) s_1(x,u,w) d(x,u,w)\right)^{1/2}$.

2. Problem description

Let us consider the following affine nonlinear continuous-time dynamical system:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) + k(x(t))w(t) \tag{1}$$

$$z(t) = h(x) \tag{2}$$

where $[x_1 \ \dots \ x_n]^T \in \mathcal{X} \subset \mathbb{R}^n$ is the state, x_0 is the initial state, $u = [u_1 \ \dots \ u_m]^T \in \mathcal{U} \subset \mathbb{R}^m$ is the control input and $u(t) \in L_2(0,\infty)$, $w = [w_1 \ \dots \ w_q]^T \in \mathcal{W} \subset \mathbb{R}^q$ is the external disturbance and $w(t) \in L_2(0,\infty)$, $z = [z_1 \ \dots \ z_p]^T \in \mathbb{R}^p$ is the objective output. $f(x)$ is Lipschitz continuous on a set $\mathcal{X}$ that contains the origin, $f(0) = 0$. $f(x)$ represents the internal system model which is assumed to be *unknown* in this paper. $g(x), k(x)$ and $h(x)$ are known continuous vector or matrix functions of appropriate dimensions.

The H_∞ control problem under consideration is to find a state feedback control law $u(x)$ such that the system (1) is closed-loop asymptotically stable, and has L_2-gain less than or equal to γ , that is,

$$\int_0^\infty \left(\|z(t)\|^2 + \|u(t)\|_R^2\right) dt \leq \gamma^2 \int_0^\infty \|w(t)\|^2 dt \tag{3}$$

for all $w(t) \in L_2(0,\infty)$, $R > 0$ and $\gamma > 0$ is some prescribed level of disturbance attenuation. From [18], this problem can be transformed to solve the so-called HJI equation, which is summarized in Lemma 1.

Lemma 1. *Assume the system* (1) *and* (2) *is zero-state observable. For* $\gamma > 0$, *suppose there exists a solution* $V^*(x)$ *to the HJI equation*

$$\begin{aligned}[\nabla V^*(x)]^T f(x) + h^T(x)h(x) - \frac{1}{4}[\nabla V^*(x)]^T g(x) R^{-1} g^T(x) \nabla V^*(x) \\ + \frac{1}{4\gamma^2}[\nabla V^*(x)]^T k(x) k^T(x) \nabla V^*(x) = 0.\end{aligned} \tag{4}$$

where $V^*(x) \in C^1(\mathcal{X}), V^*(x) \geqslant 0$ *and* $V^*(0) = 0$. *Then, the closed-loop system with the state feedback control*

$$u(t) = u^*(x(t)) = -\frac{1}{2} R^{-1} g^T(x) \nabla V^*(x) \tag{5}$$

has L_2*-gain less than or equal to* γ, *and the closed-loop system* (1), (2) *and* (5) *(when* $w(t) \equiv 0$ *) is locally asymptotically stable.* □

3. Motivation from investigation of related work

From Lemma 1, it is noted that the H_∞ control (5) rely on the solution of the HJI equation (4). Therefore, a model-based iterative method was proposed in [21], where the HJI equation is successively approximated by a sequence of linear PDEs:

$$[\nabla V^{(i,j+1)}]^T (f + g u^{(i)} + k w^{(i,j)}) + h^T h + \|u^{(i)}\|_R^2 - \gamma^2 \|w^{(i,j)}\|^2 = 0; \tag{6}$$

and then update control and disturbance policies with

$$w^{(i,j+1)} \triangleq \frac{1}{2}\gamma^{-2} k^T \nabla V^{(i,j+1)} \tag{7}$$

$$u^{(i+1)} \triangleq -\frac{1}{2} R^{-1} g^T \nabla V^{(i+1)} \tag{8}$$

with $V^{(i+1)} \triangleq \lim_{j\to\infty} V^{(i,j)}$. From [18, 21], it was indicated that the $V^{(i,j)}$ can converge go the solution of the HJI equation, i.e., $\lim_{i,j\to\infty} V^{(i,j)} = V^*$.

Remark 1. Note that the key point of the iterative scheme (6)-(8) depends on the solution of the linear PDE (6). Thus, several related methods were developed, such as, Galerkin approximation [21], synchronous policy iteration [23], neuro-dynamic programming approach [22, 31] and online adaptive control method [32] for constrained input systems, and Galerkin approximation method for discrete-time systems [30]. Obviously, the iteration (6)-(8) will generate two iterative loops since the control and disturbance policies are updated at the different iterative steps, i.e., the inner loop for updating disturbance policy w with index j, and the outer iterative loop for updating control policy u with index i. The outer loop will not be activated until the inner loop is convergent, which results in low efficiency. Therefore, Luo and Wu [35] proposed a simultaneous policy update algorithm (SPUA), where the control and disturbance policies are updated at the same iterative step, and thus only one iterative loop is required. □

The procedure of model-based SPUA is given in Algorithm 1.

Algorithm 1. Model-based SPUA.

- ▶ *Step 1:* Give an initial function $V^{(0)} \in \mathbb{V}_0$ ($\mathbb{V}_0 \subset \mathbb{V}$ is determined by Lemma 5 in [35]. Let $i = 0$;
- ▶ *Step 2:* Update the control and disturbance policies with

$$u^{(i)} \triangleq -\frac{1}{2}R^{-1}g^T\nabla V^{(i)} \tag{9}$$

$$w^{(i)} \triangleq \frac{1}{2}\gamma^{-2}k^T\nabla V^{(i)} \tag{10}$$

- ▶ *Step 3:* Solve the following linear PDE for $V^{(i+1)}(x)$:

$$[\nabla V^{(i+1)}]^T(f + gu^{(i)} + kw^{(i)}) + h^Th + \|u^{(i)}\|_R^2 - \gamma^2\|w^{(i)}\|^2 = 0; \tag{11}$$

where $V^{(i+1)}(x) \in C^1(\mathcal{X}), V^{(i+1)}(x) \geqslant 0$ and $V^{(i+1)}(0) = 0$.

- ▶ *Step 4:* Let $i = i + 1$, go back to Step 2 and continue. □

By constructing a fixed point equation, the convergence of Algorithm 1 is established in [35] by proving that it is essentially a Newtons iteration method for finding the fixed point. With the increase of index i, the sequence $V^{(i)}$ obtained by the SPUA with equations (9)-(11) can converge to the solution of HJI equation (4), i.e., $\lim_{i\to\infty} V^{(i)} = V^*$.

Note that both iterative equations (6) and (11) require the full system model. For the H_∞ control problem the internal system dynamic $f(x)$ is unknown, data based methods [38, 39] were suggested to solve the HJI equation online. However, most of related existing online methods are on-policy learning approaches [23, 32, 38–40]. From the definition of on-policy learning [2], the cost function should be evaluated with the data generated from the evaluating policies. For example, $V^{(i,j+1)}$ in equation (6) is the cost function of the policies $w^{(i,j)}$ and $u^{(i)}$, which means that $V^{(i,j+1)}$ should be evaluated with system data by using evaluating policies $w^{(i,j)}$ and $u^{(i)}$. It is observed that these on-policy learning approaches for solving the H_∞ control problem have several drawbacks:

- 1) For real implementation of on-policy learning methods [23, 32, 39, 40], the approximate evaluating control and disturbance policies (rather than the actual policies) are used to generate data for learning their cost function. In other words, the on-policy learning methods using the "inaccurate" data to learn their cost function, which will increase the accumulated error. For example, to learn the cost function $V^{(i,j+1)}$ in equation (6), some approximate policies $\widehat{w}^{(i,j)}$ and $\widehat{u}^{(i)}$ (rather than its actual policies $w^{(i,j)}$ and $u^{(i)}$, which are usually unknown because of estimate error) are employed to generate data;

- 2) The evaluating control and disturbance policies are required to generate data for on-policy learning, thus disturbance signal should be adjustable, which is usually impractical for most of real systems;

- 3) It is known [2, 41] that the issue of "exploration" is extremely important in RL for learning the optimal control policy, and the lack of exploration during the learning process may lead to divergency. Nevertheless, for on-policy learning, exploration is restricted because only the evaluating policies can be used to generate data. From the literature investigation, it is found that the "exploration" issue is rarely discussed in existing work that using RL techniques for control design;

- 4) The implementation structure is complicated, such as in [23, 32], three NNs are required for approximating cost function, control and disturbance policies, respectively;
- 5) Most of existing approaches [23, 32, 38–40] are implemented online, thus they are difficult for real-time control because the learning process is often time-consuming. Furthermore, online control design approaches just use current data while discard past data, which implies that the measured system data is used only once and thus results in low utilization efficiency.

To overcome the drawbacks mentioned above, we propose an off-policy RL approach to solve the H_∞ control problem with unknown internal system dynamic $f(x)$.

4. Off-policy reinforcement learning for robust H_∞ control

In this section, an off-policy RL method for robust H_∞ control design is derived and its convergence is proved. Then, a NN-based critic-actor structure is developed for implementation purpose.

4.1. Off-policy reinforcement learning

To derive the off-policy RL method, we rewrite the system (1) as:

$$\dot{x} = f + gu^{(i)} + kw^{(i)} + g[u - u^{(i)}] + k[w - w^{(i)}]. \tag{12}$$

for $\forall u \in \mathcal{U}, w \in \mathcal{W}$. Let $V^{(i+1)}(x)$ be the solution of the linear PDE (11), then taking derivative along the state of system (12) yields,

$$\frac{dV^{(i+1)}(x)}{dt} = [\nabla V^{(i+1)}]^T(f+gu^{(i)}+kw^{(i)})+[\nabla V^{(i+1)}]^T g[u-u^{(i)}]+[\nabla V^{(i+1)}]^T k[w-w^{(i)}]. \tag{13}$$

With the linear PDE (11), conducting integral on both sides of equation (13) in time interval $[t, t+\Delta t]$ and rearranging terms yield,

$$\begin{aligned}&\int_t^{t+\Delta t} [\nabla V^{(i+1)}(x(\tau))]^T g(x(\tau))[u(\tau) - u^{(i)}(x(\tau))]d\tau \\ &\quad + \int_t^{t+\Delta t} [\nabla V^{(i+1)}(x(\tau))]^T k(x(\tau))[w(\tau) - w^{(i)}(x(\tau))]d\tau \\ &\quad + V^{(i+1)}(x(t)) - V^{(i+1)}(x(t+\Delta t))\end{aligned}$$

$$= \int_t^{t+\Delta t} \left(h^T(x(\tau))h(x(\tau)) + \|u^{(i)}(x(\tau))\|_R^2 - \gamma^2\|w^{(i)}(x(\tau))\|^2\right) d\tau \tag{14}$$

Replacing linear PDE (11) in Algorithm 1 with (14) results in the off-policy RL method. To show its convergence, Theorem 1 establishes the equivalence between iterative equations (11) and (14).

Theorem 1. *Let $V^{(i+1)}(x) \in C^1(\mathcal{X}), V^{(i+1)}(x) \geqslant 0$ and $V^{(i+1)}(0) = 0$. $V^{(i+1)}(x)$ is the solution of equation (14) iff (if and only if) it is the solution of the linear PDE (11), i.e., equation (14) is equivalent to the linear PDE (11).*

Proof. From the derivation of equation (14), it is concluded that if $V^{(i+1)}(x)$ is the solution of the linear PDE (11), then $V^{(i+1)}(x)$ also satisfies equation (14). To complete the proof, we have to show that $V^{(i+1)}(x)$ is the unique solution of equation (14). The proof is by contradiction.

Before starting the contradiction proof, we derive a simply fact. Consider

$$\begin{aligned} \lim_{\Delta t \to 0} \frac{1}{\Delta t} \int_t^{t+\Delta t} \hbar(\tau) d\tau &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left(\int_0^{t+\Delta t} \hbar(\tau) d\tau - \int_0^t \hbar(\tau) d\tau \right) \\ &= \frac{d}{dt} \int_0^t \hbar(\tau) d\tau \\ &= \hbar(t). \end{aligned} \tag{15}$$

From (14), we have

$$\begin{aligned} \frac{dV^{(i+1)}(x)}{dt} &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \left[V^{(i+1)}(x(t+\Delta t)) - V^{(i+1)}(x(t))\right] \\ &= \lim_{\Delta t \to 0} \frac{1}{\Delta t} \int_t^{t+\Delta t} [\nabla V^{(i+1)}(x(\tau))]^T g(x(\tau))[u(\tau) - u^{(i)}(x(\tau))] d\tau \\ &\quad + \lim_{\Delta t \to 0} \frac{1}{\Delta t} \int_t^{t+\Delta t} [\nabla V^{(i+1)}(x(\tau))]^T k(x(\tau))[w(\tau) - w^{(i)}(x(\tau))] d\tau \\ &\quad - \lim_{\Delta t \to 0} \frac{1}{\Delta t} \int_t^{t+\Delta t} \left[h^T(x(\tau))h(x(\tau)) + \|u^{(i)}(x(\tau))\|_R^2 - \gamma^2 \|w^{(i)}(x(\tau))\|^2\right] d\tau. \end{aligned} \tag{16}$$

By using the fact (15), the equation (16) is rewritten as

$$\begin{aligned}\frac{dV^{(i+1)}(x)}{dt} =&[\nabla V^{(i+1)}(x(t))]^T g(x(t))[u(t) - u^{(i)}(x(t))] \\ &+ [\nabla V^{(i+1)}(x(t))]^T k(x(t))[w(t) - w^{(i)}(x(t))] \\ &- \left[h^T(x(t))h(x(t)) + \|u^{(i)}(x(t))\|_R^2 - \gamma^2 \|w^{(i)}(x(t))\|^2\right].\end{aligned} \tag{17}$$

Suppose that $W(x) \in C^1(\mathcal{X})$ is another solution of equation (14) with boundary condition $W(0) = 0$. Thus, $W(x)$ also satisfies equation (17), i.e.,

$$\begin{aligned}\frac{dW(x)}{dt} =&[\nabla W(x(t))]^T g(x(t))[u(t) - u^{(i)}(x(t))] \\ &+ [\nabla W(x(t))]^T k(x(t))[w(t) - w^{(i)}(x(t))] \\ &- \left[h^T(x(t))h(x(t)) + \|u^{(i)}(x(t))\|_R^2 - \gamma^2 \|w^{(i)}(x(t))\|^2\right].\end{aligned} \tag{18}$$

Substituting equation (18) from (17) yields,

$$\begin{aligned}\frac{d}{dt}\left(V^{(i+1)}(x) - W(x)\right) = &\left[\nabla \left(V^{(i+1)}(x) - W(x)\right)\right]^T g(x)[u - u^{(i)}(x)] \\ &+ \left[\nabla \left(V^{(i+1)}(x) - W(x)\right)\right]^T k(x)[w - w^{(i)}(x)].\end{aligned} \tag{19}$$

This means that equation (19) holds for $\forall u \in \mathcal{U}, w \in \mathcal{W}$. If letting $u = u^{(i)}, w = w^{(i)}$, we have

$$\frac{d}{dt}\left[V^{(i+1)}(x) - W(x)\right] = 0. \tag{20}$$

Then, $V^{(i+1)}(x) - W(x) = c$ for $\forall x \in \mathcal{X}$, where c is a real constant, and $c = V^{(i+1)}(0) - W(0) = 0$. Thus,$V^{(i+1)}(x) - W(x) = 0$, i.e., $W(x) = V^{(i+1)}(x)$ for $\forall x \in \mathcal{X}$. This completes the proof. □

Remark 2. It follows from Theorem 1 that the solution of equation (14) is equivalent to equation (11), and thus the convergence of the off-policy RL is guaranteed, i.e., the solution of the iterative equation (14) will converge to the solution of HJI equation (4) as iteration step i increases. Different from the equation (11) in Algorithm 1, the off-policy RL with equation (14) uses system data instead of the internal system dynamic $f(x)$. Hence, the off-policy RL can be regarded as a direct learning method for H_∞ control design, which avoids the identification of $f(x)$. In fact, the information of $f(x)$ is embedded in the online measurement of system data. That is to say, the lack of knowledge about $f(x)$ does not have any impact on the off-policy RL to obtain the solution of HJI equation (4) and the H_∞ control policy . $\square$

4.2. Implementation based on neural network

To solve equation (14) for the unknown function $V^{(i+1)}(x)$ based on system data, we develop a neural network (NN) based actor-critic structure. From the well known high-order Weierstrass approximation theorem [42], a continuous function can be represented by an infinite-dimensional linearly independent basis function set. For real practical application, it is usually required to approximate the function in a compact set with a finite-dimensional function set. We consider the critic NN for approximating the cost function on a compact set $\mathcal{X}$. Let $\varphi(x) \triangleq [\varphi_1(x) \ ... \ \varphi_L(x)]^T$ be the vector of linearly independent activation functions for critic NN, where $\varphi_l(x) : \mathcal{X} \mapsto \mathbb{R}, l = 1, ..., L, L$ is the number of critic NN hide layer neurons. Then, the output of critic NN are given by

$$\widehat{V}^{(i)}(x) = \sum_{l=1}^{L} \theta_l^{(i)} \varphi_l(x) = \varphi^T(x)\theta^{(i)} \tag{21}$$

for $\forall i = 0, 1, 2, ...$, where $\theta^{(i)} \triangleq [\theta_1^{(i)} \ ... \ \theta_L^{(i)}]^T$ is the critic NN weight vector. It follows from (9), (10) and (21) that the outputs of actor NNs for disturbance and control policies are given by:

$$\widehat{u}^{(i)}(x) = -\frac{1}{2}R^{-1}g^T(x)\nabla\varphi^T(x)\theta^{(i)} \tag{22}$$

$$\widehat{w}^{(i)}(x) = \frac{1}{2}\gamma^{-2}k^T(x)\nabla\varphi^T(x)\theta^{(i)} \tag{23}$$

for $\forall i = 0, 1, 2, ...$, and $\nabla\varphi(x) \triangleq [\partial\varphi_1/\partial x \ ... \ \partial\varphi_L/\partial x]^T$ is the Jacobian of $\varphi(x)$.

Due to estimation errors of the critic and actor NNs (21)-(23), the replacement of $V^{(i+1)}, w^{(i)}$ and $u^{(i)}$ in the iterative equation (14) with $\widehat{V}^{(i+1)}, \widehat{w}^{(i)}$ and $\widehat{u}^{(i)}$ respectively, yields the following residual error:

$$
\begin{aligned}
\sigma^{(i)}(x(t),u(t),w(t)) \triangleq & \int_t^{t+\Delta t} [u(\tau)-u^{(i)}(x(\tau))]^T g^T(x(\tau))\nabla\varphi^T(x(\tau))\theta^{(i+1)}d\tau \\
& + \int_t^{t+\Delta t} [w(\tau)-w^{(i)}(x(\tau))]^T k^T(x(\tau))\nabla\varphi^T(x(\tau))\theta^{(i+1)}d\tau \\
& + [\varphi(x(t))-\varphi(x(t+\Delta t))]^T\theta^{(i+1)} \\
& - \int_t^{t+\Delta t} \left[h^T(x(\tau))h(x(\tau)) + \|u^{(i)}(x(\tau))\|_R^2 - \gamma^2\|w^{(i)}(x(\tau))\|^2\right] d\tau \\
= & \int_t^{t+\Delta t} u^T(\tau)g^T(x(\tau))\nabla\varphi^T(x(\tau))\theta^{(i+1)}d\tau \\
& + \frac{1}{2}\int_t^{t+\Delta t} \left(\theta^{(i)}\right)^T \nabla\varphi(x(\tau))g(x(\tau))R^{-1}g^T(x(\tau))\nabla\varphi^T(x(\tau))\theta^{(i+1)}d\tau \\
& + \int_t^{t+\Delta t} w^T(\tau)k^T(x(\tau))\nabla\varphi^T(x(\tau))\theta^{(i+1)}d\tau \\
& - \frac{1}{2}\gamma^{-2}\int_t^{t+\Delta t} \left(\theta^{(i)}\right)^T \nabla\varphi(x(\tau))k(x(\tau))k^T(x(\tau))\nabla\varphi^T(x(\tau))\theta^{(i+1)}d\tau \\
& + [\varphi(x(t))-\varphi(x(t+\Delta t))]^T\theta^{(i+1)} \\
& - \frac{1}{4}\int_t^{t+\Delta t} \left(\theta^{(i)}\right)^T \nabla\varphi(x(\tau))g(x(\tau))R^{-1}g^T(x(\tau))\nabla\varphi^T(x(\tau))\theta^{(i)}d\tau \\
& + \frac{1}{4}\gamma^{-2}\int_t^{t+\Delta t} \left(\theta^{(i)}\right)^T \nabla\varphi(x(\tau))k(x(\tau))k^T(x(\tau))\nabla\varphi^T(x(\tau))\theta^{(i)}d\tau \\
& - \int_t^{t+\Delta t} h^T(x(\tau))h(x(\tau))d\tau
\end{aligned} \tag{24}
$$

For notation simplicity, define

$$
\begin{aligned}
\rho_{\Delta\varphi}(x(t)) &\triangleq [\varphi(x(t))-\varphi(x(t+\Delta t))]^T \\
\rho_{g\varphi}(x(t)) &\triangleq \int_t^{t+\Delta t} \nabla\varphi(x(\tau))g(x(\tau))R^{-1}g^T(x(\tau))\nabla\varphi^T(x(\tau))d\tau \\
\rho_{k\varphi}(x(t)) &\triangleq \int_t^{t+\Delta t} \nabla\varphi(x(\tau))k(x(\tau))k^T(x(\tau))\nabla\varphi^T(x(\tau))d\tau \\
\rho_{u\varphi}(x(t),u(t)) &\triangleq \int_t^{t+\Delta t} u^T(\tau)g^T(x(\tau))\nabla\varphi^T(x(\tau))d\tau \\
\rho_{w\varphi}(x(t),w(t)) &\triangleq \int_t^{t+\Delta t} w^T(\tau)k^T(x(\tau))\nabla\varphi^T(x(\tau))d\tau \\
\rho_h(x(t)) &\triangleq \int_t^{t+\Delta t} h^T(x(\tau))h(x(\tau))d\tau
\end{aligned}
$$

then, expression (24) is rewritten as

$$\begin{aligned}\sigma^{(i)}(x(t),u(t),w(t)) =&\rho_{u\varphi}(x(t),u(t))\theta^{(i+1)} + \frac{1}{2}\left(\theta^{(i)}\right)^T \rho_{g\varphi}(x(t))\theta^{(i+1)} \\ &+ \rho_{w\varphi}(x(t),w(t))\theta^{(i+1)} - \frac{1}{2}\gamma^{-2}\left(\theta^{(i)}\right)^T \rho_{k\varphi}(x(t))\theta^{(i+1)} \\ &+ \rho_{\Delta\varphi}\theta^{(i+1)} - \frac{1}{4}\left(\theta^{(i)}\right)^T \rho_{g\varphi}(x(t))\theta^{(i)} \\ &+ \frac{1}{4}\gamma^{-2}\left(\theta^{(i)}\right)^T \rho_{k\varphi}(x(t))\theta^{(i)} - \rho_h(x(t)).\end{aligned} \tag{25}$$

For description convenience, expression (25) is represented as a compact form

$$\sigma^{(i)}(x(t),u(t),w(t)) = \overline{\rho}^{(i)}(x(t),u(t),w(t))\theta^{(i+1)} - \overline{\rho}_1^{(i)}(x(t)) \tag{26}$$

where

$$\begin{aligned}\overline{\rho}^{(i)}(x(t),u(t),w(t)) \triangleq&\rho_{u\varphi}(x(t),u(t)) + \frac{1}{2}\left(\theta^{(i)}\right)^T \rho_{g\varphi}(x(t)) + \rho_{w\varphi}(x(t),w(t)) \\ &- \frac{1}{2}\gamma^{-2}\left(\theta^{(i)}\right)^T \rho_{k\varphi}(x(t)) + \rho_{\Delta\varphi} \\ \overline{\rho}_1^{(i)}(x(t)) \triangleq&\frac{1}{4}\left(\theta^{(i)}\right)^T \rho_{g\varphi}(x(t))\theta^{(i)} - \frac{1}{4}\gamma^{-2}\left(\theta^{(i)}\right)^T \rho_{k\varphi}(x(t))\theta^{(i)} + \rho_h(x(t)).\end{aligned}$$

Based on the method of weighted residuals [43], the unknown critic NN weight vector $\theta^{(i+1)}$ can be computed in such a way that the residual error $\sigma^{(i)}(x,u,w)$ (for $\forall t \geqslant 0$) of (26) is forced to be zero in some average sense. Thus, projecting the residual error $\sigma^{(i)}(x,u,w)$ onto $d\sigma^{(i)}/d\theta^{(i+1)}$ and setting the result to zero on domain $\mathcal{D}$ using the inner product, $\langle\cdot,\cdot\rangle_{\mathcal{D}}$, i.e.,

$$\left\langle d\sigma^{(i)}/d\theta^{(i+1)}, \sigma^{(i)}(x,u,w)\right\rangle_{\mathcal{D}} = 0. \tag{27}$$

Then, the substitution of (26) into (27) yields,

$$\left\langle\overline{\rho}^{(i)}(x,u,w), \overline{\rho}^{(i)}(x,u,w)\right\rangle_{\mathcal{D}}\theta^{(i+1)} - \left\langle\overline{\rho}^{(i)}(x,u,w), \overline{\rho}_1^{(i)}(x)\right\rangle_{\mathcal{D}} = 0$$

and thus $\theta^{(i+1)}$ can be obtained with

$$\theta^{(i+1)} = \left\langle\overline{\rho}^{(i)}(x,u,w), \overline{\rho}^{(i)}(x,u,w)\right\rangle_{\mathcal{D}}^{-1}\left\langle\overline{\rho}^{(i)}(x,u,w), \overline{\rho}_1^{(i)}(x)\right\rangle_{\mathcal{D}}. \tag{28}$$

The computation of inner products $\left\langle\overline{\rho}^{(i)}(x,u,w), \overline{\rho}^{(i)}(x,u,w)\right\rangle_{\mathcal{D}}$ and $\left\langle\overline{\rho}^{(i)}(x,u,w), \overline{\rho}_1^{(i)}(x)\right\rangle_{\mathcal{D}}$ involve many numerical integrals on domain $\mathcal{D}$, which are computationally expensive. Thus, the Monte-Carlo integration method [44] is introduced, which is especially competitive on multi-dimensional domain. We now illustrate the Monte-Carlo integration for computing $\langle\overline{\rho}^{(i)}(x,u,w),\overline{\rho}^{(i)}(x,u,w)\rangle_{\mathcal{D}}$. Let $I_{\mathcal{D}} \triangleq \int_{\mathcal{D}} d(x,u,w)$, and $\mathcal{S}_M \triangleq \{(x_m,u_m,w_m)|(x_m,u_m,w_m) \in \mathcal{D}, m=1,2,...,M\}$ be the set that sampled on domain $\mathcal{D}$, where M is size of sample set $\mathcal{S}_M$. Then, $\langle\overline{\rho}^{(i)}(x,u,w),\overline{\rho}^{(i)}(x,u,w)\rangle_{\mathcal{D}}$ is approximately computed with

$$\begin{aligned}\langle\overline{\rho}^{(i)}(x,u,w),\overline{\rho}^{(i)}(x,u,w)\rangle_{\mathcal{D}} &= \int_{\mathcal{D}} \left(\overline{\rho}^{(i)}(x,u,w)\right)^T \overline{\rho}^{(i)}(x,u,w)d(x,u,w) \\ &= \frac{I_{\mathcal{D}}}{M}\sum_{m=1}^{M}\left(\overline{\rho}^{(i)}(x_m,u_m,w_m)\right)^T \overline{\rho}^{(i)}(x_m,u_m,w_m) \\ &= \frac{I_{\mathcal{D}}}{M}\left(Z^{(i)}\right)^T Z^{(i)} \end{aligned} \tag{29}$$

where $Z^{(i)} \triangleq \left[\left(\overline{\rho}^{(i)}(x_1,u_1,w_1)\right)^T \ \ldots \ \left(\overline{\rho}^{(i)}(x_M,u_M,w_M)\right)^T\right]^T$. Similarly,

$$\begin{aligned}\left\langle\overline{\rho}^{(i)}(x,u,w),\overline{\rho}_1^{(i)}(x)\right\rangle_{\mathcal{D}} &= \frac{I_{\mathcal{D}}}{M}\sum_{m=1}^{M}\left(\overline{\rho}^{(i)}(x_m,u_m,w_m)\right)^T \overline{\rho}_1^{(i)}(x_m) \\ &= \frac{I_{\mathcal{D}}}{M}\left(Z^{(i)}\right)^T \eta^{(i)} \end{aligned} \tag{30}$$

where $\eta^{(i)} \triangleq \left[\overline{\rho}_1^{(i)}(x_1) \ \ldots \ \overline{\rho}_1^{(i)}(x_M)\right]^T$. Then, the substitution of (29) and (30) into (28) yields,

$$\theta^{(i+1)} = \left[\left(Z^{(i)}\right)^T Z^{(i)}\right]^{-1}\left(Z^{(i)}\right)^T \eta^{(i)}. \tag{31}$$

It is noted that the critic NN weight update rule (31) is a least-square scheme. Based on the update rule (31), the procedure for robust H_∞ control design with NN-based off-policy RL is presented in Algorithm 2.

Algorithm 2. NN-based off-policy RL for robust H_∞ control design.

- ▶ *Step 1:* Conduct closed-loop simulation on system (1) with different initial state, control and disturbance signals. Collect online system data (x_m,u_m,w_m) for sample set $\mathcal{S}_M$, and then compute $\rho_{\Delta\varphi}(x_m), \rho_{g\varphi}(x_m), \rho_{k\varphi}(x_m)$, $\rho_{u\varphi}(x_m,u_m)$, $\rho_{w\varphi}(x_m,w_m)$ and $\rho_h(x_m)$;

- ▶ *Step 2:* Select initial critic NN weight vector $\theta^{(0)}$ such that $\widehat{V}^{(0)} \in \mathbb{V}_0$. Let $i = 0$;
- ▶ *Step 3:* Compute $Z^{(i)}$ and $\eta^{(i)}$, and update $\theta^{(i+1)}$ with (31);
- ▶ *Step 4:* Let $i = i+1$. If $\|\theta^{(i)} - \theta^{(i-1)}\| \leq \xi$ (ξ is a small positive number), stop iteration and $\theta^{(i)}$ is employed to obtain the H_∞ control policy with (22), else go back to Step 3 and continue. □

Remark 3. Note that Algorithm 2 has two parts: the first part is Step 1 for data processing, i.e., measure online system data (x, u, w) for computing $\rho_{\Delta\varphi}, \rho_{g\varphi}, \rho_{k\varphi}, \rho_{u\varphi}, \rho_{w\varphi}$ and ρ_h; the second part is Steps 2-4 for offline learning the solution of the HJI equation (4). □

Algorithm 2 can be viewed as an off-policy learning method according to references [2, 45, 46], which overcomes the drawbacks mentioned in Section 3, i.e.,

- 1) In the off-policy RL algorithm (i.e., Algorithm 2), the control u and disturbance w can be arbitrarily on $\mathcal{U}$ and $\mathcal{W}$, where no error occurs during the process of generating data, and thus the accumulated error (in the on-policy learning methods mentioned in Section 3) can be reduced;
- 2) In the Algorithm 2, the control u and disturbance w can be arbitrarily on $\mathcal{U}$ and $\mathcal{W}$, and thus disturbance w does not required to be adjustable;
- 3) In the Algorithm 2, the cost function $V^{(i+1)}$ of control and disturbance policies $(u^{(i)}, w^{(i)})$ can be evaluated by using system data generated with other different control and disturbance signals (u, w). Thus, the obvious advantage of the developed off-policy RL method is that it can learn the cost function and control policy from system data that are generated according to a more exploratory or even random policies;
- 4) The implementation of Algorithm 2 is very simple, in fact *only one* NN is required, i.e., critic NN. This means that once the critic NN weight vector $\theta^{(i+1)}$ is computed via (31), the action NNs for control and disturbance policies can be obtained based on (22) and (22) accordingly;

- 5) The developed off-policy RL method learns the H_∞ control policy offline, which is then used for real-time control. Thus, it is much more practical than online control design methods since less computational load will generate during real-time application. Meanwhile, note that in Algorithm 2, once the terms $\rho_{\Delta\varphi}, \rho_{g\varphi}, \rho_{k\varphi}, \rho_{u\varphi}, \rho_{w\varphi}$ and ρ_h are computed with sample set $\mathcal{S}_M$ (i.e., Step 1 is finished), no extra data is required for learning the H_∞ control policy (in Steps 2-4). This means that the collected data set can be utilized repeatedly, and thus the utilization efficiency is improved compared to the online control design methods.

5. Off-policy reinforcement learning for linear H_∞ control

In this section, the developed NN-based off-policy RL method (i.e., Algorithm 2) is simplified for linear H_∞ control design. Consider the linear system:

$$\dot{x}(t) = Ax(t) + B_2u(t) + B_1w(t) \tag{32}$$

$$z(t) = Cx \tag{33}$$

where $A \in \mathbb{R}^{n\times n}$, $B_1 \in \mathbb{R}^{n\times q}$, $B_2 \in \mathbb{R}^{n\times m}$ and $C \in \mathbb{R}^{p\times n}$. Then, the HJI equation (4) of the linear system (32) and (33) results in an algebraic Riccati equation (ARE) [37, 47]:

$$A^TP + PA + Q + \gamma^{-2}PB_1B_1^TP - PB_2R^{-1}B_2^TP = 0 \tag{34}$$

where $Q = C^TC$. If ARE (34) has a stabilizing solution $P \geqslant 0$, the solution of the HJI equation (4) of the linear system (32) and (33) is $V^*(x) = x^TPx$, and then the linear H_∞ control policy (5) is accordingly given by

$$u^*(x) = -R^{-1}B_2^TPx. \tag{35}$$

Consequently, $V^{(i)}(x) = x^TP^{(i)}x$, then the iterative equations (9)-(11) in Algorithm 1 are respectively represented with

$$u^{(i)} = -R^{-1}B_2^TP^{(i)}x \tag{36}$$

$$w^{(i)} = \gamma^{-2}B_1^TP^{(i)}x \tag{37}$$

$$\overline{A}_i^T P^{(i+1)} + P^{(i+1)}\overline{A}_i + \overline{Q}^{(i)} = 0 \tag{38}$$

where $\overline{A}_i \triangleq A + \gamma^{-2} B_1 B_1^T P^{(i)} - B_2 R^{-1} B_2^T P^{(i)}$ and $\overline{Q}^{(i)} \triangleq Q - \gamma^{-2} P^{(i)} B_1 B_1^T P^{(i)} + P^{(i)} B_2 R^{-1} B_2^T P^{(i)}$.

Similar to the derivation of the off-policy RL method for nonlinear H_∞ control design in Section 4, rewrite the linear system (32) as

$$\dot{x} = Ax + B_2 u^{(i)} + B_1 w^{(i)} + B_2[u - u^{(i)}] + B_1[w - w^{(i)}]. \tag{39}$$

Based on equations (36)-(39), the equation (14) is given by

$$\begin{aligned}
&\int_t^{t+\Delta t} x^T(\tau) P^{(i+1)} B_2 \left[u(\tau) + R^{-1} B_2^T P^{(i)} x(\tau)\right] d\tau \\
&\quad + \int_t^{t+\Delta t} x^T(\tau) P^{(i+1)} B_1 \left[w(\tau) - \gamma^{-2} B_1^T P^{(i)} x(\tau)\right] d\tau \\
&\quad + [x(t) - x(t+\Delta t)]^T P^{(i+1)} [x(t) - x(t+\Delta t)] \\
&= \int_t^{t+\Delta t} x^T(\tau) \overline{Q}^{(i)} x(\tau) d\tau
\end{aligned} \tag{40}$$

where $P^{(i+1)}$ is a $n \times n$ unknown matrix to be learned. For notation simplicity, define

$$\begin{aligned}
\rho_{\Delta x}(x(t)) &\triangleq x(t) - x(t+\Delta t) \\
\rho_{xx}(x(t)) &\triangleq \int_t^{t+\Delta t} x(\tau) \otimes x(\tau) d\tau \\
\rho_{ux}(x(t), u(t)) &\triangleq \int_t^{t+\Delta t} u(\tau) \otimes x(\tau) d\tau \\
\rho_{wx}(x(t), w(t)) &\triangleq \int_t^{t+\Delta t} w(\tau) \otimes x(\tau) d\tau
\end{aligned}$$

where $\otimes$ denotes Kronecker product. Each term of equation (40) can be written as:

$$\begin{aligned}
\int_t^{t+\Delta t} x^T(\tau) P^{(i+1)} B_2 u(\tau) d\tau &= \rho_{ux}^T(x(t), u(t)) (B_2^T \otimes I) vec(P^{(i+1)}) \\
\int_t^{t+\Delta t} x^T(\tau) P^{(i+1)} B_2 R^{-1} B_2^T P^{(i)} x(\tau) d\tau &= \rho_{xx}^T(x(t)) (P^{(i)} B_2 R^{-1} B_2^T \otimes I) vec(P^{(i+1)}) \\
\int_t^{t+\Delta t} x^T(\tau) P^{(i+1)} B_1 w(\tau) d\tau &= \rho_{wx}^T(x(t), w(t)) (B_1^T \otimes I) vec(P^{(i+1)}) \\
\gamma^{-2} \int_t^{t+\Delta t} x^T(\tau) P^{(i+1)} B_1 B_1^T P^{(i)} x(\tau) d\tau &= \gamma^{-2} \rho_{xx}^T(x(t)) (P^{(i)} B_1 B_1^T \otimes I) vec(P^{(i+1)})
\end{aligned}$$

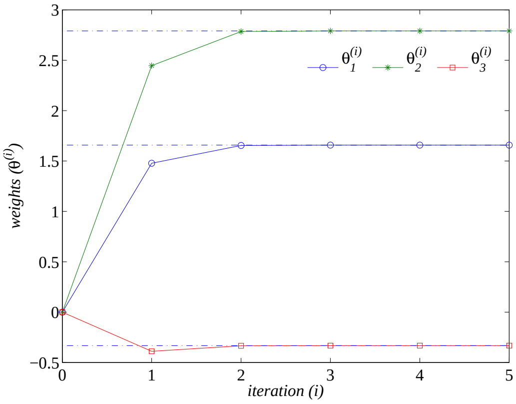

Figure 1: For the linear F16 aircraft plant, the critic NN weights $\theta_1^{(i)} \sim \theta_3^{(i)}$ at each iteration.

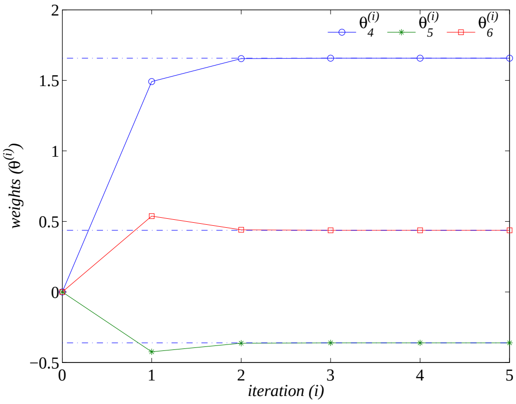

Figure 2: For the linear F16 aircraft plant, the critic NN weights $\theta_4^{(i)} \sim \theta_6^{(i)}$ at each iteration.

$$
\begin{aligned}
[x(t) - x(t+\Delta t)]^T P^{(i+1)} [x(t) - x(t+\Delta t)] =& \rho_{\Delta x}^T (x(t)) vec(P^{(i+1)}) \\
\int_t^{t+\Delta t} x^T(\tau) \overline{Q}^{(i)} x(\tau) d\tau =& \rho_{xx}^T (x(t)) vec(\overline{Q}^{(i)})
\end{aligned}
$$

where $vec(P)$ denotes the vectorization of the matrix P formed by stacking the columns of P into a single column vector. Then, equation (40) can be rewritten as

$$
\overline{\rho}^{(i)}(x(t), u(t), w(t)) vec(P^{(i+1)}) = \overline{\rho}_1^{(i)}(x(t)) \tag{41}
$$

with

$$
\begin{aligned}
\overline{\rho}^{(i)}(x(t), u(t), w(t)) =& \rho_{ux}^T(x(t), u(t))(B_2^T \otimes I) + \rho_{wx}^T(x(t), w(t))(B_1^T \otimes I) + \rho_{\Delta x}^T(x(t)) \\
&+ \rho_{xx}^T(x(t))[(P^{(i)} B_2 R^{-1} B_2^T \otimes I) - \gamma^{-2}(P^{(i)} B_1 B_1^T \otimes I)] \\
\overline{\rho}_1^{(i)}(x(t)) =& \rho_{xx}^T(x(t)) vec(\overline{Q}^{(i)}).
\end{aligned}
$$

It is noted that equation (41) is equivalent to the equation (26) with residual error $\sigma^{(i)} = 0$. This is because no cost function approximation is required for linear systems. Then, by collecting sample set $\mathcal{S}_M$ for computing $\rho_{ux}, \rho_{wx}, \rho_{xx}$ and $\rho_{\Delta x}$, a more simpler least-square scheme (31) can be derived to obtain the unknown parameter vector $vec(P^{(i+1)})$ accordingly.

6. Simulation studies

In this section, the efficiency of the developed NN-based off-policy RL method is tested on a F16 aircraft plant. Then, it is applied to the rotational/translational actuator (RTAC) nonlinear benchmark problem.

6.1. Efficiency test on linear F16 aircraft plant

Consider a F16 aircraft plant that used in [23, 37, 39, 48], where the system dynamics is described by a linear continuous-time model:

$$\dot{x} = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 \\ 0 & 0 & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} w \tag{42}$$

$$z = x. \tag{43}$$

where the system state vector is $x = [\alpha \; q \; \delta_e]^T$, α denotes the angle of attack, q is the pitch rate and δ_e is the elevator deflection angle. The control input u is the elevator actuator voltage and the disturbance w is wind gusts on angle of attack. Select $R = 1$ and $\gamma = 5$ for the L_2-gain performance (3). Then, solve the associated ARE (34) with the MATLAB command CARE, we obtain

$$P = \begin{bmatrix} 1.6573 & 1.3954 & -0.1661 \\ 1.3954 & 1.6573 & -0.1804 \\ -0.1661 & -0.1804 & 0.4371 \end{bmatrix}.$$

For linear systems, the solution of the HJI equation is $V^*(x) = x^T P x$, thus the complete activation function vector for critic NN is $\varphi(x) = [x_1^2 \; x_1x_2 \; x_1x_3 \; x_2^2 \; x_2x_3 \; x_3^2]^T$ of size $L = 6$. Then, the idea critic NN weight vector is $\theta^* = [p_{11} \; 2p_{12} \; 2p_{13} \; p_{22} \; 2p_{23} \; p_{33}]^T = [1.6573 \; 2.7908 \; -0.3322 \; 1.6573 \; -0.3608 \; 0.4371]^T$. Letting initial critic NN weight $\theta_l^{(0)} = 0 (l = 1, ..., 6)$ and iterative stop criterion $\xi = 10^{-7}$, Algorithm 2 is applied to learn the solution of the ARE. Figures 1-2 give the critic NN weight $\theta^{(i)}$ at each iteration, in which the dash lines represent idea values of θ^*. It is observed from the figures that the critic NN weights converge to

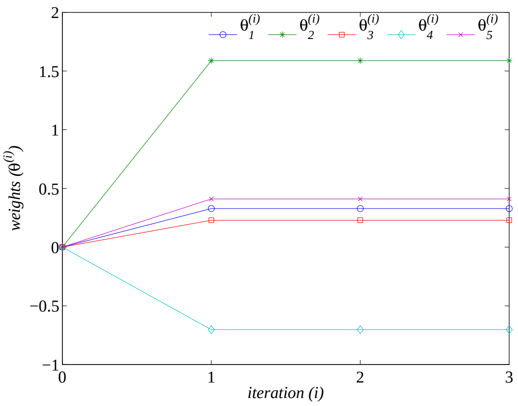

Figure 3: For the RTAC system, the critic NN $\theta_1^{(i)} \sim \theta_5^{(i)}$ weights at each iteration.

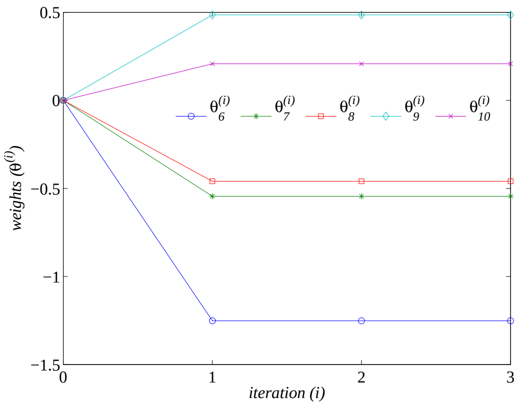

Figure 4: For the RTAC system, the critic NN $\theta_6^{(i)} \sim \theta_{10}^{(i)}$ weights at each iteration.

the idea values of θ^* at $i = 5$ iteration. Then, the efficiency of the developed off-policy RL method is verified.

6.2. Application to the rotational/translational actuator nonlinear benchmark problem

The RTAC nonlinear benchmark problem [31, 35, 49] has been widely used to test the abilities of control methods. The dynamics of this nonlinear plant poses challenges because the rotational and translation motions are coupled. The RTAC system is given as follows:

$$\dot{x} = \begin{bmatrix} x_2 \\ \dfrac{-x_1 + \zeta x_4^2 \sin x_3}{1 - \zeta^2 \cos^2 x_3} \\ x_4 \\ \dfrac{\zeta \cos x_3 (x_1 - \zeta x_4^2 \sin x_3)}{1 - \zeta^2 \cos^2 x_3} \end{bmatrix} + \begin{bmatrix} 0 \\ \dfrac{-\zeta \cos x_3}{1 - \zeta^2 \cos^2 x_3} \\ 0 \\ \dfrac{1}{1 - \zeta^2 \cos^2 x_3} \end{bmatrix} u + \begin{bmatrix} 0 \\ \dfrac{1}{1 - \zeta^2 \cos^2 x_3} \\ 0 \\ \dfrac{-\zeta \cos x_3}{1 - \zeta^2 \cos^2 x_3} \end{bmatrix} w \qquad (44)$$

$$z = \sqrt{0.1} I x \qquad (45)$$

where $\zeta = 0.2$. For the L_2-gain performance (3), let $R = 1$ and $\gamma = 6$.

Then, the developed off-policy RL method is used to solve the nonlinear H_∞ control

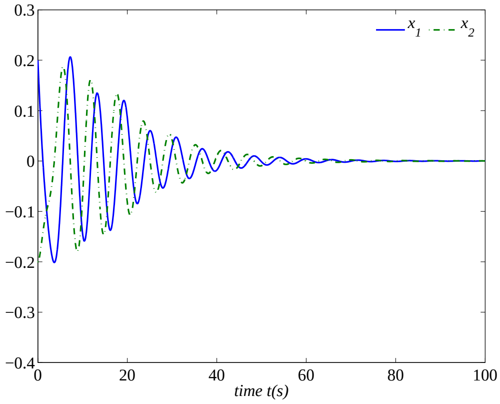

Figure 5: The state trajectories $x_1(t), x_2(t)$ of the closed-loop RTAC system.

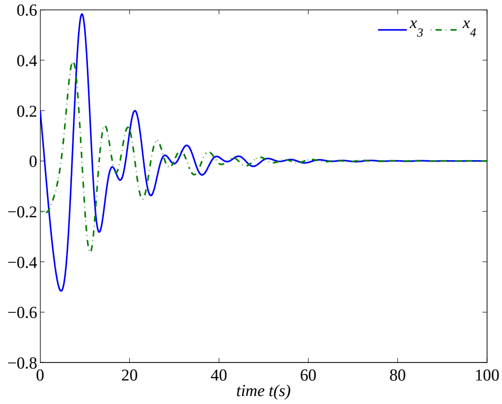

Figure 6: The state trajectories $x_3(t), x_4(t)$ of the closed-loop RTAC system.

problem of system (44) and (45). Select the critic NN activation function vector as

$$\varphi(x) = [x_1^2 \;\; x_1x_2 \;\; x_1x_3 \;\; x_1x_4 \;\; x_2^2 \;\; x_2x_3 \;\; x_2x_4 \;\; x_3^2 \;\; x_3x_4 \;\; x_4^2 \;\; x_1^3x_2 \;\; x_1^3x_3 \\ x_1^3x_4 \;\; x_1^2x_2^2 \;\; x_1^2x_2x_3 \;\; x_1^2x_2x_4 \;\; x_1^2x_3^2 \;\; x_1^2x_3x_4 \;\; x_1^2x_4^2 \;\; x_1x_2^3]^T$$

of size $L = 20$. With the initial critic NN weight $\theta_l^{(0)} = 0(l = 1, ..., 20)$ and iterative stop criterion $\xi = 10^{-7}$, Algorithm 2 is applied to learn the solution of the HJI equation. It is found that the critic NN weight vector converges fast to

$$\theta^{(3)} = [0.3285 \; 1.5877 \; 0.2288 \; -0.7028 \; 0.4101 \; -1.2514 \; -0.5448 \; -0.4595 \; 0.4852 \; 0.2078 \\ -1.3857 \; 1.7518 \; 1.1000 \; 0.5820 \; 0.1950 \; -0.0978 \; -1.0295 \; -0.2773 \; -0.2169 \; 0.2463]^T$$

at $i = 3$ iteration. Figures 3-4 show first 10 critic NN weights (i.e., $\theta_1^{(i)} \sim \theta_{10}^{(i)}$) at each iteration. With the convergent critic NN weight vector $\theta^{(3)}$, the H_∞ control policy can be computed with (22). Under the disturbance signal $w(t) = 0.2r_1(t)e^{-0.2t}cos(t), (r_1(t) \in [0, 1]$ is a random number), closed-loop simulation is conducted with the H_∞ control policy. Figures 5-7 give the trajectories of state and control policy. To show the relationship between L_2-gain and time, define the following ratio of disturbance attenuation as

$$r_d(t) = \left(\frac{\int_0^t (\|z(\tau)\|^2 + \|u(\tau)\|_R^2) d\tau}{\int_0^t \|w(\tau)\|^2 d\tau} \right)^{\frac{1}{2}}.$$

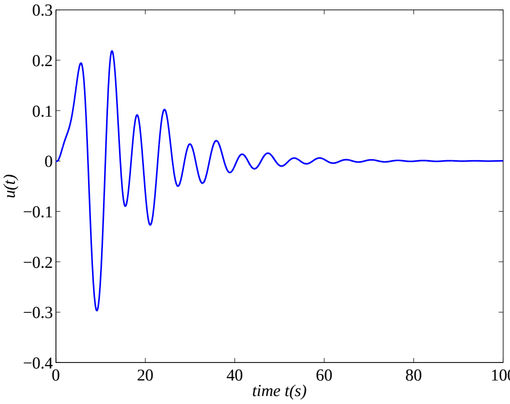

Figure 7: The control trajectory $u(t)$ of the closed-loop RTAC system.

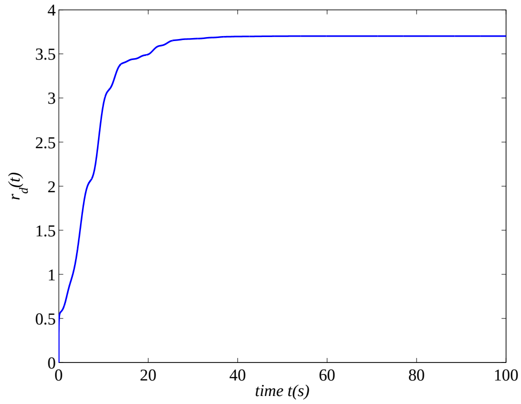

Figure 8: The curve $r_d(t)$ of the closed-loop RTAC system.

Figure 8 shows the curve of $r_d(t)$, where it converges to $3.7024(< \gamma = 6)$ as time increases, which implies that the designed H_∞ control law can achieve an prescribed L_2-gain performance level γ for the closed-loop system.

7. Conclusions

A NN-based off-policy RL method has been developed to solve the H_∞ control problem of continuous-time systems with unknown internal system model. Based on the model-based SPUA, an off-policy RL method is derived, which can learn the solution of HJI equation from the system data generated by arbitrary control and disturbance signals. The implementation of the off-policy RL method is based on an actor-critic structure, where only one NN is required for approximating the cost function, and then a least-square scheme is derived for NN weights update. The effectiveness of the proposed NN-based off-policy RL method is tested on a linear F16 aircraft plant and a nonlinear RTAC problem.

References

[1] L. P. Kaelbling, M. L. Littman, A. W. Moore, Reinforcement learning: A survey, Journal of Artificial Intelligence Research 4 (1996) 237–285.

[2] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, Cambridge Univ Press, Massachusetts London, England, 1998.

[3] D. P. Bertsekas, Dynamic Programming and Optimal Control, Vol. 1, Nashua: Athena Scientific, 2005.

[4] W. B. Powell, Approximate Dynamic Programming: Solving the Curses of Dimensionality, Vol. 703, Hoboken, N.J.: John Wiley & Sons, 2007.

[5] J. J. Murray, C. J. Cox, G. G. Lendaris, R. Saeks, Adaptive dynamic programming, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 32 (2) (2002) 140–153.

[6] D. Liu, D. Wang, D. Zhao, Q. Wei, N. Jin, Neural-network-based optimal control for a class of unknown discrete-time nonlinear systems using globalized dual heuristic programming, IEEE Transactions on Automation Science and Engineering 9 (3) (2012) 628–634.

[7] D. Vrabie, F. L. Lewis, Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems, Neural Networks 22 (3) (2009) 237–246.

[8] H. Zhang, L. Cui, X. Zhang, Y. Luo, Data-driven robust approximate optimal tracking control for unknown general nonlinear systems using adaptive dynamic programming method, IEEE Transactions on Neural Networks 22 (12) (2011) 2226–2236.

[9] D. Liu, D. Wang, H. Li, Decentralized stabilization for a class of continuous-time nonlinear interconnected systems using online learning optimal control approach, IEEE Transactions on Neural Networks and Learning Systems (2013) In Press.

[10] Q. Yang, S. Jagannathan, Reinforcement learning controller design for affine nonlinear discrete-time systems using online approximators, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 42 (2) (2012) 377–390.

[11] H. Zhang, Y. Luo, D. Liu, Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints, IEEE Transactions on Neural Networks 20 (9) (2009) 1490–1503.

[12] Z. Ni, H. He, J. Wen, Adaptive learning in tracking control based on the dual critic network design, IEEE Transactions on Neural Networks and Learning Systems 24 (6) (2013) 913–928.

[13] Q. Wei, D. Liu, A novel iterative θ-adaptive dynamic programming for discrete-time nonlinear systems, IEEE Transactions on Automation Science and Engineering (2013) In Press.

[14] H. Zhang, Q. Wei, Y. Luo, A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 38 (4) (2008) 937–942.

[15] K. Zhou, J. C. Doyle, K. Glover, Robust and Optimal Control, Prentice Hall New Jersey, 1996.

[16] A. v. d. Schaft, L_2-Gain and Passivity in Nonlinear Control, Springer-Verlag New York, Inc., 1996.

[17] T. Başar, P. Bernhard, H_∞ Optimal Control and Related Minimax Design Problems: A Dynamic

Game Approach, Springer, 2008.

[18] A. v. d. Schaft, L_2-gain analysis of nonlinear systems and nonlinear state-feedback H_∞ control, IEEE Transactions on Automatic Control 37 (6) (1992) 770–784.

[19] A. Isidori, W. Kang, H_∞ control via measurement feedback for general nonlinear systems, IEEE Transactions on Automatic Control 40 (3) (1995) 466–472.

[20] A. Isidori, A. Astolfi, Disturbance attenuation and H_∞-control via measurement feedback in nonlinear systems, IEEE Transactions on Automatic Control 37 (9) (1992) 1283–1293.

[21] R. W. Beard, Successive Galerkin approximation algorithms for nonlinear optimal and robust control, International Journal of Control 71 (5) (1998) 717–743.

[22] M. Abu-Khalaf, F. L. Lewis, J. Huang, Policy iterations on the Hamilton–Jacobi–Isaacs equation for H_∞ state feedback control with input saturation, IEEE Transactions on Automatic Control 51 (12) (2006) 1989–1995.

[23] K. G. Vamvoudakis, F. L. Lewis, Online solution of nonlinear two-player zero-sum games using synchronous policy iteration, International Journal of Robust and Nonlinear Control 22 (13) (2012) 1460–1483.

[24] Y. Feng, B. Anderson, M. Rotkowitz, A game theoretic algorithm to compute local stabilizing solutions to HJBI equations in nonlinear H_∞ control, Automatica 45 (4) (2009) 881–888.

[25] D. Liu, H. Li, D. Wang, Neural-network-based zero-sum game for discrete-time nonlinear systems via iterative adaptive dynamic programming algorithm, Neurocomputing 110 (13) (2013) 92–100.

[26] N. Sakamoto, A. v. d. Schaft, Analytical approximation methods for the stabilizing solution of the Hamilton–Jacobi equation, IEEE Transactions on Automatic Control 53 (10) (2008) 2335–2350.

[27] G. N. Saridis, C.-S. G. Lee, An approximation theory of optimal control for trainable manipulators, IEEE Transactions on Systems, Man and Cybernetics 9 (3) (1979) 152–159.

[28] R. W. Beard, G. N. Saridis, J. T. Wen, Galerkin approximations of the generalized Hamilton-Jacobi-Bellman equation, Automatica 33 (12) (1997) 2159–2177.

[29] R. Beard, G. Saridis, J. Wen, Approximate solutions to the time-invariant Hamilton–Jacobi–Bellman equation, Journal of Optimization Theory and Applications 96 (3) (1998) 589–626.

[30] S. Mehraeen, T. Dierks, S. Jagannathan, M. L. Crow, Zero-sum two-player game theoretic formulation of affine nonlinear discrete-time systems using neural networks, IEEE Transactions on Cybernetics (2013) In Press.

[31] M. Abu-Khalaf, F. L. Lewis, J. Huang, Neurodynamic programming and zero-sum games for constrained control systems, IEEE Transactions on Neural Networks 19 (7) (2008) 1243–1252.

[32] H. Modares, F. L. Lewis, M.-B. N. Sistani, Online solution of nonquadratic two-player zero-sum games arising in the H_∞ control of constrained input systems, International Journal of Adaptive Control and

Signal Processing (2013) In Press.

[33] H. Zhang, Q. Wei, D. Liu, An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games, Automatica 47 (1) (2011) 207–214.

[34] K. G. Vamvoudakis, F. L. Lewis, Online actor–critic algorithm to solve the continuous-time infinite horizon optimal control problem, Automatica 46 (5) (2010) 878–888.

[35] B. Luo, H.-N. Wu, Computationally efficient simultaneous policy update algorithm for nonlinear H_∞ state feedback control with Galerkin's method, International Journal of Robust and Nonlinear Control 23 (9) (2013) 991–1012.

[36] J. Huang, C.-F. Lin, Numerical approach to computing nonlinear H_∞ control laws, Journal of Guidance, Control, and Dynamics 18 (5) (1995) 989–994.

[37] H.-N. Wu, B. Luo, Simultaneous policy update algorithms for learning the solution of linear continuous-time H_∞ state feedback control, Information Sciences 222 (2013) 472–485.

[38] D. Vrabie, F. Lewis, Adaptive dynamic programming for online solution of a zero-sum differential game, Journal of Control Theory and Applications 9 (3) (2011) 353–360.

[39] H.-N. Wu, B. Luo, Neural network based online simultaneous policy update algorithm for solving the HJI equation in nonlinear H_∞ control, IEEE Transactions on Neural Networks and Learning Systems 23 (12) (2012) 1884–1895.

[40] H. Zhang, L. Cui, Y. Luo, Near-optimal control for nonzero-sum differential games of continuous-time nonlinear systems using single-network ADP, IEEE Transactions on Cybernetics 43 (1) (2013) 206–216.

[41] S. B. Thrun, Efficient exploration in reinforcement learning, Tech. rep., Carnegie Mellon University Pittsburgh, PA (1992).

[42] R. Courant, D. Hilbert, Methods of Mathematical Physics, Vol. 1, Wiley, 2004.

[43] B. A. Finlayson, The Method of Weighted Residuals and Variational Principles: With Applications in Fluid Mechanics, Heat and Mass Transfer, Vol. 87, New York: Academic Press, Inc., 1972.

[44] G. Peter Lepage, A new algorithm for adaptive multidimensional integration, Journal of Computational Physics 27 (2) (1978) 192–203.

[45] D. Precup, R. S. Sutton, S. Dasgupta, Off-policy temporal-difference learning with function approximation, in: Proceedings of the 18th International Conference on Machine Learning, 2001, pp. 417–424.

[46] H. R. Maei, C. Szepesvári, S. Bhatnagar, R. S. Sutton, Toward off-policy learning control with function approximation, in: Proceedings of the 27th International Conference on Machine Learning, 2010, pp. 719–726.

[47] M. Green, D. J. Limebeer, Linear Robust Control, Prentice-Hall, Englewood Cliffs, NJ, 1995.

[48] B. L. Stevens, F. L. Lewis, Aircraft Control and Simulation, Wiley-Interscience, 2003.

[49] G. Escobar, R. Ortega, H. Sira-Ramirez, An L_2 disturbance attenuation solution to the nonlinear

benchmark problem, International Journal of Robust and Nonlinear Control 8 (4-5) (1999) 311–330.