International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 05 | May-2016 www.irjet.net p-ISSN: 2395-0072

# A Novel Method of Dynamic Programming for the Adaptive Optimal Control of Nonlinear Systems

**Mr. Desai Feroz[1], Mrs. M. Lakshmi Swarupa[2].**

[1]Mr. Desai Feroz, P.G. Scholar, EEE, MREC(Autonomous), Telangana, India
[2]Mrs. M. Lakshmi Swarupa, Assoc. Professor & HOD, EEE, MREC (Autonomous), Telangana, India

---

**Abstract -** Dynamic programming offers a theoretical way to solve optimal control problems. However, it suffers from the inherent computational complexity, also known as the curse of dimensionality. To achieve online approximation of the cost function and the control policy, neural networks are widely used in the previous ADP architecture. The main purpose of this paper is to develop a novel dynamic programming methodology to achieve global and adaptive suboptimal stabilization of uncertain nonlinear systems via online learning. In this paper an optimization problem, of which the solutions can be easily parameterized, is proposed to relax the problem of solving the Hamilton-Jacobi-Bellman (HJB) equation. This approach is inspired from the relaxation method used in approximate dynamic programming. It is also proposed that a relaxed policy iteration method is different for the inverse optimal control design and the results are obtained by M-file programming.

**Key Words: Global adaptive dynamic programming, Approximate Dynamic programming, Nonlinear Systems, Adaptive optimal control, Semi-definite programming (SDP), Sum of squares (SOS).**

## 1.INTRODUCTION

The main purpose of this paper is to develop a novel method to achieve adaptive optimal stabilization of nonlinear system via online learning. As the first contribution of this paper, an optimization problem is proposed to relax the Hamilton-Jacobi-Bellman (HJB) equation. The second contribution of the paper is an online learning method that implements the proposed iterative schemes using only the real-time online measurements, when the perfect system knowledge is not available. The final contribution of this paper is the Robust Design of the approximate suboptimal control policy, such that the overall system can be globally asymptotically stable in the presence of dynamic uncertainties.

## 2. PROBLEM FORMULATION AND PRELIMINARIES

In this section, we first formulate the control problem[16][17]. Then, we introduce conventional policy iteration algorithm.

### 2.1 Problem formulation

Consider the nonlinear system

$$dx/dt = f(x) + g(x)u \quad (1.1)$$

where,
x is the system state,
u is the control input,
f(x) and g(x) are locally Lipschitz functions with f(0) = 0.
In conventional optimal control theory [1], the common objective is to find a control policy u that minimizes certain performance index. In this chapter, it is specified as follows.

$$J(x_0; u) = \int_0^\infty r(x(t), u(t))dt,; \quad x(0) = x_0 \quad (1.2)$$

where,
$r(x, u) = Q(x) + u^T Ru$, with Q(x) a positive definite function, and R is a symmetric positive definite matrix.

### 2.2 Optimality and stability

Here, we recall a basic result connecting optimality and global asymptotic stability in nonlinear systems [2]. To begin with, let us give the following assumption.
*Assumption 2.1.2.* There exists $V^o \epsilon P$, such that the Hamilton-Jacobi-Bellman (HJB) equation holds[16]

$$H(V^o) = 0 \quad (1.3)$$

Where,
$H(V) = \nabla V^T(x)f(x)+Q(x)-(1/4)\nabla V^T(x)g(x)R^{-1}g^T(x)\nabla V(x)$

Under Assumption 2.1.2, it is easy to see that $V^o$ is a well-defined Lyapunov function for the closed-loop system comprised of (1.1) and

$$u^o(x) = -\frac{1}{2} R^{-1}g^T(x)\nabla V^o(x) \quad (1.4)$$

Hence, this closed-loop system is globally asymptotically stable at x = 0 [3]. Then, according to [2], $u^o$ is the optimal control policy, and the value function $V^o(x_0)$ gives the optimal cost at the initial condition x(0) = x0, i.e.,

$$V^o(x_0) = \min_u J(x_0, u) = J(x_0,u^o), \forall x_0 \epsilon R^n: \quad (1.5)$$

By Theorem 3.19 in [2], along the solutions of the closed-loop system composed of (1.1) and $u= \hat{u} = -\frac{1}{2} R^{-1}g^T\nabla\hat{V}$, it follows that $\hat{V}(x_0) = V^o(x_0)-\int_0^\infty l\, u^0 - \hat{u}\, l^2_R dt, \quad \forall x_0 \epsilon R^n$ (1.6)
Finally, comparing (1.5) and (1.6), we conclude that $V^o = \hat{V}$.

*Algorithm 2.1.1 Conventional policy iteration:*

1. Policy evaluation: For i = 1; 2; _ _ _ , solve for the cost function $V_i(x) \in \mathbb{C}^1$, with $V_i(0) = 0$, from the following partial differential equation.

$$\mathcal{L}(V_i(x), u_i(x)) = 0. \quad (1.7)$$

2. Policy improvement: Update the control policy by

$$u_{i+1}(x) = -\frac{1}{2} R^{-1} g^T(x) \nabla V_i(x) \quad (1.8)$$

The following result is a trivial extension of [4, Theorem 4], in which g(x) is a constant matrix and only stabilization over compact set is considered.

Notice that finding the analytical solution to (1.7) is still non-trivial. Hence, in practice, the solution is approximated using, for example, neural networks or Galerkin's method [5]. When the precise knowledge of f or g is not available, ADP based online approximation method can be applied to compute numerically the cost functions via online data [6], [7].

In general, approximation methods can only give acceptable results on some compact set in the state space, but cannot be used to achieve global stabilization. In addition, in order to reduce the approximation error, huge computational complexity is almost inevitable. These facts may affect the effectiveness of the previously developed ADP-based online learning methods.

## 2.3. Semidefinite Programming (SDP)

According to the equivalence between SOS programs and SDPs, the SOS-based policy iteration can be reformulated as SDPs [16] [17]. Notice that we can always find two linear mappings $\imath : R^{n_{2r}} \times R^{m \times n_r} \to R^{n_{2d}}$ and $\varkappa : R^{n_{2r}} \to R^{m \times n_r}$, such that given $p \in R^{n_{2r}}$ and $k \in R^{m \times n_r}$,

$$\imath(p,k)^T [x]_{2,2d} = \mathcal{L}(p^T [x]_{2,2r}, k[x]_{1,2r-1}) \quad (1.9)$$

$$\varkappa(p,k)^T [x]_{1,2r-1} = -\frac{1}{2} R^{-1} g^T \nabla (p^T [x]_{2,2r}) \quad (1.10)$$

Then, by properties of SOS constraints [8], the polynomial $\imath(p, k)^T [x]_{2,2d}$ is SOS if and only if there exists a symmetric and positive semidefinite matrix $L \in R^{n_d \times n_d}$, such that

$$\imath(p, k)^T [x]_{2,2d} = [x]^T_{1,d} L [x]_{1,d} \quad (1.11)$$

Furthermore, there exist linear mappings $M_P : R^{n_r \times n_r} \to R^{n_{2r}}$ and $M_L : R^{n_d \times n_d} \to R^{n_{2d}}$, such that, for any vectors $p \in R^{n_{2r}}$, $l \in R^{n_{2d}}$, and symmetric matrices $P \in R^{n_r \times n_r}$ and $L \in R^{n_d \times n_d}$, the following implications are true.

$$p^T [x]_{2,2r} = [x]^T_{1,r} P [x]_{1,r} \Leftrightarrow p = M_P(P) \quad (1.12)$$

$$l^T [x]_{2,2d} = [x]^T_{1,d} L [x]_{1,d} \Leftrightarrow l = M_L(L) \quad (1.13)$$

using above assumptions policy iteration can be reformulated as follows.

*Algorithm 2.2.1 SDP-based policy iteration:*

1. Let i = 1. Let $p_0 \in R^{n_{2r}}$ and $K_1 \in R^{m \times n_d}$ satisfy $V_0 = p_0^T [x]_{2,2r}$ and $u_1 = K_1 [x]_{1,d}$.

2. Solve for an optimal solution $(p_i; P_i; L_i) \in R^{n_{2r}} \times R^{n_r \times n_r} \times R^{n_d \times n_d}$ to the following problem.

$$\min_{p,P,L} \; c^T p \quad (1.14)$$

$$\text{s.t.} \quad \imath(p,k_i) = M_L(L) \quad (1.15)$$

$$p_{i-1} - p = M_P(P) \quad (1.16)$$

$$P = P^T \geq 0 \quad (1.17)$$

$$L = L^T \geq 0 \quad (1.18)$$

where $c = \int s(x) [x]_{2,2r} dx$.

3: Go to Step 2 with $k_{i+1} = \varkappa(p_i)$ and i replaced by i+1.

## 3. ONLINE LEARNING VIA GLOBAL ADAPTIVE DYNAMIC PROGRAMMING

The proposed policy iteration method requires the perfect knowledge of the mappings $\imath$ and $\varkappa$, which can be determined if f and g are known exactly. In practice, precise system knowledge may be difficult to obtain. Hence, in this section, we develop an online learning method based on the idea of ADP to implement the iterative scheme with real-time data, instead of identifying the system dynamics. To begin with, consider the system

$$\dot{x} = f + g(u_i + e) \quad (1.19)$$

where $u_i$ is a feedback control policy and e is a bounded time-varying function, known as the exploration noise, added for the learning purpose.

*Lemma 3.1.1.* Consider system (1.19). Suppose $u_i$ is a globally stabilizing control policy and there exists $V_{i-1} \in \mathcal{P}$, such that

$$\nabla V_{i-1}(f + g u_i) + u_i^T R u_i \leq 0.$$

Then, by completing the squares, it follows that

$$\nabla V^T_{i-1}(f + g u_i + g e) \leq -u_i^T R u_i - 2 u_i^T R e$$
$$= -|u_i + e|^2_R + |e|^2_R$$
$$\leq |e|^2_R$$
$$\leq |e|^2_R + V_{i-1}$$

Suppose there exist $p \in R^{n_{2r}}$ and $k_i \in R^{m \times n_k}$ such that $V = p^T [x]_{2,2r}$ and $u_i = k_i [x]_{1,d}$. Then, along the solutions of the system (1.19), it follows that

$$\dot{V} = \nabla V^T (f + g u_i) + \nabla V^T B e$$
$$= -r(x, u_i) - \mathcal{L}(V, u_i) + \nabla V^T g e$$
$$= -r(x, u_i) - \mathcal{L}(V, u_i) + 2(1/2 R^{-1} g^T \nabla V)^T R e$$
$$= -r(x, ui) - \imath(p,k_i)^T [x]_{2,2d} - 2[x]^T_{1,d} \varkappa(p)^T R e \quad (1.20)$$

where the last row is obtained by (1.9) and (1.10).

Now, integrating the terms in (1.52) over the interval [t, t + $\delta$t], we have

$$p^T([x(t)]_{2,2r} - [x(t + \delta t)]_{2,2r})$$
$$= \int_t^{t+\delta t} [r(x,u_i) + \imath(p,ki)^T [x] + 2[x]^T_{1,d} \varkappa(p)^T R e] dt \quad (1.21)$$

Eq. (1.21) implies that, given $p \in R^{n_{2r}}$, $\imath(p,k_i)$ and $\varkappa(p)$ can be directly calculated by using real-time online data, without knowing the precise knowledge of f and g.

Indeed, define

$$\sigma_e = -[[x]^T_{2,2d} \; 2[x]^T_{1,d} e^T R]^T \in R^{n_{2d}+mn_d},$$

$$\Phi_i = [\int_{t0,i}^{t1,i} \sigma_e dt \int_{t1,i}^{t2,i} \sigma_e dt \ldots \int_{tqi-1,i}^{tqi,i} \sigma_e dt]^T \in R^{q_i x (n_{2d}+mn_d)}$$

$$\Xi_i = [\int_{t0,i}^{t1,i} r(x, u_i) dt \int_{t1,i}^{t2,i} r(x, u_i) dt \ldots \int_{tqi-1,i}^{tqi,i} r(x, u_i) dt]^T \in R^{q_i},$$

$$\Theta_i = [[x]_{2,2r}|_{t0,i}^{t1,i} \ldots [x]_{2,2r}|_{tqi-1,i}^{tqi,i}]^T \in R^{qi \times n_{2r}};$$

Then, (1.21) implies

$$\Phi_i \begin{bmatrix} \imath(p,ki) \\ vec(\varkappa(p)) \end{bmatrix} = \Xi_i + \Theta_i\, p. \quad (1.22)$$

*Assumption 3.1.2*. For each i = 1; 2; _ _ _ , there exists an integer $q_{i0}$, such that when $q_i \geq q_{i0}$ the following rank condition holds.

$\text{rank}(\Phi_i) = n_{2d} + mn_d$ . (1.23)

*Remark 3.1.2.* Such a rank condition (1.23) is in the spirit of persistency of excitation (PE) in adaptive control and is a necessary condition for parameter convergence.

Given $p \in R^{n_{2r}}$ and $k_i \in R^{m \times n_d}$ , suppose Assumption 3.1.2. is satisfied and $q_i \geq q_{i0}$ for all i= 1,2,.... Then, it is easy to see that the values of $\imath(p,k_i)$ and $\varkappa(p)$ can be uniquely determined from

$$\begin{bmatrix} \imath(p,ki) \\ vec(\varkappa(p)) \end{bmatrix} = (\Phi_i^T \Phi_i)^{-1} \Phi_i^T(\Xi_i + \Theta_i\, p) \quad (1.24)$$

Now, we are ready to develop the ADP-based online implementation algorithm for the proposed policy iteration method.

*Algorithm 3.1.4. Global adaptive dynamic programming algorithm*

1: Initialization: Let $p_0$ be the constant vector such that $V_0 = p_0^T[x]_{2,2r}$, and let i = 1.

2: Collect online data: Apply $u = u_i+e$ to the system and compute the data matrices $\Phi_i$, $\Xi_i$, and $\Theta_i$, until the rank condition (1.23) in Assumption 3.1.2 is satisfied.

3: Policy evaluation and improvement: Find an optimal solution $(p_i, k_{i+}, P_i, L_i)$ to the following optimization problem

$$\min_{p,K,P,L} \; c^T p \quad (1.25)$$

$$\text{s:t:} \quad \begin{bmatrix} ML(L) \\ vec(k) \end{bmatrix} = (\Phi_i^T \Phi_i)^{-1} \Phi_i^T(\Xi_I + \Theta_i) \quad (1.26)$$

$$p_{i-1} - p = M_P\,(P) \quad (1.27)$$

$$P = P^T \geq 0 \quad (1.28)$$

$$L = L^T \geq 0 \quad (1.29)$$

Then, denote $V_i = p_i^T[x]_{2,2r}$, $u_{i+1} = k_{i+1}[x]_{1,d}$, and go to Step 2) with $i \leftarrow i + 1$.

## 4. ONLINE IMPLEMENTATION VIA GLOBAL ADAPTIVE DYNAMIC PROGRAMMING

Let $V = p^T \bar{\emptyset}$. Similar as in Section above, over the interval $[t, t + \delta t]$, we have

$$p^T [\bar{\emptyset}(x(t)) - \bar{\emptyset}\; (x(t + \delta t))]$$

$$= \int_t^{t+\delta t} [r(x, u_i) + \bar{\imath}(p,k_i)^T \bar{\sigma} + 2\bar{\sigma}\bar{\varkappa}\,(p)^T Re]\, dt \quad (1.30)$$

Therefore, (1.30) shows that, given $p \in R^{N_1}$, $\bar{\imath}(p, Ki)$ and $\bar{\varkappa}(p)$ can be directly obtained by using real-time online data, without knowing the precise knowledge of f and g.

Indeed, define [17]

$\bar{\sigma}_e = - [\bar{\sigma}^T 2\sigma^T \otimes e^T R]^T \in R^{l_1+ml}$,

$\bar{\Phi}_i = [\int_{t0,i}^{t1,i} \bar{\sigma}_e\, dt \int_{t1,i}^{t2,i} \bar{\sigma}_e\, dt ... \int_{tqi-1,i}^{tqi,i} \bar{\sigma}_e]^T \in R^{qi \times (l_1+ml)}$,

$\bar{\Xi}_i = [\int_{t0,i}^{t1,i} r(x,u_i)dt \int_{t1,i}^{t2,i} r(x,u_i)dt ... \int_{tqi-1,i}^{tqi,i} r(x,u_i)dt]^T \in R^{qi}$ ,

$\bar{\Theta}_i = [\bar{\emptyset}(x) \,|\, ^{t1,i}_{t0,i}\, [\bar{\emptyset}(x) \,|\, ^{t2,i}_{t1,i}\, .....]^T \in R^{q_i \times N_1}$

Then, (1.30) implies

$$\bar{\Phi}_i \begin{bmatrix} \bar{\imath}(p,ki) \\ vec(\bar{\varkappa}(p)) \end{bmatrix} = \bar{\Xi}_i + \bar{\Theta}_l\, p. \quad (1.31)$$

*Assumption 4.1.1.* For each i = 1; 2; _ _ , there exists an integer $q_{i0}$, such that, when $q_i \geq q_{i0}$, the following rank condition holds.

$\text{rank}(\bar{\Phi}_i) = l_1 + ml.$ (1.32)

Let $p \in R^{N_1}$ and $k_i \in R^{m \times l}$. Suppose Assumption 4.1.1 holds and assume $q_i \geq q_{i0}$, for i = 1,2, _ _ . Then, $\bar{\imath}(p,k_i)$ and $\bar{\varkappa}(p)$ can be uniquely determined by

$$\begin{bmatrix} h \\ vec(k) \end{bmatrix} = (\bar{\Phi}_i^T \bar{\Phi}_i)^{-1} \bar{\Phi}_i^T\, (\bar{\Xi}_l + \bar{\theta}_i\, p) \quad (1.33)$$

Now, we are ready to develop the ADP-based online implementation algorithm for the proposed policy iteration method.

*Algorithm 4.1.2 Global adaptive dynamic programming algorithm for non-polynomial systems*

1: Initialization: Let $p_0$ and $k_1$ satisfying $V_0 = p_0^T \bar{\emptyset}$, $u_1 = k_1 \sigma$, and $\mathcal{L}(V_0, u_1)$ and let i = 1.

2: Collect online data: Apply $u = u_i+e$ to the system and compute the data matrices $\bar{\Phi}_i$, $\bar{\Xi}_i$ , and $\bar{\theta}_i$, until the rank condition (1.32) is satisfied.

3: Policy evaluation and improvement: Find an optimal solution $(p_i, h_i, K_{i+1})$ to the following optimization problem

$$\min_{p,h,k} \; c^T p \quad (1.34)$$

$$\text{s:t:} \quad \begin{bmatrix} h \\ vec(k) \end{bmatrix} = (\bar{\Phi}_i^T \bar{\Phi}_i)^{-1} \bar{\Phi}_i^T\, (\bar{\Xi}_l + \bar{\theta}_i\, p) \quad (1.35)$$

$$h \in S^+\sigma \quad (1.36)$$

$$p_{i-1} - p \in S^+ \emptyset \quad (1.37)$$

Then, denote $Vi = p_i \bar{\emptyset}$ and $u_{i+1} = k_{i+1}\, \bar{\sigma}$.

4: Go to Step 2) with $i \leftarrow i + 1$.

## 5. ROBUST REDESIGN

Consider nonlinear system with dynamic uncertainties as follows [17]

$$\dot{w} = q(w, x) \quad (1.38)$$

$$\dot{x} = f(x) + g(x)\, [u + \Delta(w, x)] \quad (1.39)$$

where $x \in R^n$ is the system state, $w \in R^{n_w}$ is the state of the dynamic uncertainty, $u \in R^m$ is the control input, $f : R^n \rightarrow R^n$ and $g : R^n \rightarrow R^{n \times m}$ are unknown polynomial mappings with f(0) = 0.

Again, in the presence of the dynamic uncertainty, i.e. the w-subsystem, Algorithm 4.1.2. may not lead to an optimal or suboptimal control policy, since $u_i$ obtained in Algorithm 4.1.2. may not be stabilizing for the overall system (1.38)-(1.39). Therefore, to balance the tradeoff between global

robust stability and optimality, here we develop a method to redesign the control policy. Similarly as in the previous chapter, the idea is inspired from the work by [9, 10].

To begin with, we define the cost functional as

$$\min J(x_0, u) = \int_0^\infty [Q(x) + u^T R u]\, dt, \quad (1.40)$$

where $Q(x) = Q_0(x) + \varepsilon |x|^2$, with $Q_0(x)$ is a positive definite function, $\varepsilon > 0$ is a constant, R is a symmetric and positive definite matrix.

Our design objective is twofold. First, we intend to minimize the cost (1.40) for the nominal system

$$\dot{x} = f(x) + g(x)u, \quad (1.41)$$

by finding online an optimal control policy $u_o$. Second, we want to guarantee the stability of the system comprised of (1.38) and (1.39) by redesigning the optimal control policy. To this end, let us introduce the following Assumption.

*Assumption 5.1.1.* Consider the system comprised of (1.38) and (1.39).

There exist functions $\underline{\lambda}, \overline{\lambda} \in \mathcal{K}\infty$ , $\kappa_1, \kappa_2, \kappa_3 \in \mathcal{K}$, and positive definite functions W and $\kappa_4$, such that for all $w \in R^p$ and $x \in R^n$, we have

$$\underline{\lambda}(|w|) \le W(w) \le \overline{\lambda}(|w|), \quad (1.42)$$

$$|\Delta(w, x)| \le \kappa_1(|w|) + \kappa_2(|x|), \quad (1.43)$$

together with the following implication:

$$W(w) \ge \kappa_3(|x|) \Rightarrow \nabla W(w)^T q(w, x) \le -\kappa_4(w). \quad (1.44)$$

Assumption 5.1.1 implies that the w-system (1.38) is input-to-state stable (ISS) [11, 12] when x is considered as the input.

Let $V_i \in \mathcal{P}$ and $u_i$ be the cost function and the control policy obtained from Algorithm 4.1.2. Then, we know that $\mathcal{L}(V_i, u_i) \ge 0$. Also, there exist $\underline{\alpha}, \overline{\alpha} \in k\infty$, such that the following inequalities hold:

$$\underline{\alpha}(|x|) \le V^o(x) \le V_i(x) \le V_0(x) \le \overline{\alpha}(|x|), \forall x_0 \in R^n; \quad (1.45)$$

The robustly redesigned control policy is given below:

$$u_{r,i} = \rho^2(|x|^2)u_i + e \quad (1.46)$$

where $\rho(.)$ is a smooth and no decreasing function with $\rho(s) \ge 1$, $\forall s > 0$, e denotes the time varying exploration noise added for the purpose of online learning.

*Theorem 5.1.2.* Consider the closed-loop system comprised of (1.38), (1.39), and (1.46). Let $V_i \in \mathcal{P}$ and $u_i$ be the cost function and the control policy obtained from Algorithm 4.1.2 at the i-th iteration step. Then, the closed-loop system is ISS with respect to e as the input, if the following gain condition holds:

$$\gamma > \kappa_1 \circ \underline{\lambda}^{-1} \circ \kappa_3 \circ \underline{\alpha}^{-1} \circ \overline{\alpha} + \kappa_2, \quad (1.47)$$

whereis defined by

$$\gamma(s) = \varepsilon s \sqrt{\frac{\frac{1}{4} + \frac{1}{2}\rho^2(s^2)}{\lambda_{min}(R)}} \quad (1.48)$$

Proof. Let $\chi_1 = \kappa_3 \circ \underline{\alpha}^{-1}$. Then, under Assumption 5.1.1, we immediately have the following implications

$$W(w) \ge \chi_1(V_i(x))$$

$$\Rightarrow W(w) \ge \kappa_3(\underline{\alpha}^{-1}(V_i(x))) \ge \kappa_3(|x|))$$

$$\Rightarrow \nabla W(w)^T q(w, x) \le -\kappa_4(w) \quad (1.49)$$

Define $\bar{\rho}(x) = \sqrt{\frac{\frac{1}{4} + \frac{1}{2}\rho^2(s^2)}{\lambda_{min}(R)}}$. Then, along solutions of the system comprised of (1.39), it follows that

$$\nabla V_i^T [f + g(u_{r,i} + \Delta)]$$

$$\le -Q(x) - |u_i|_R^2 + \nabla V_i^T g[(\rho^2(|x|)^2 - 1)u_i + \Delta + e$$

$$\le -Q(x) - \tilde{\rho}^2 |g^T \nabla V_i|^2 + \nabla V_i^T g(\Delta + e)$$

$$\le -Q(x) - \left|\tilde{\rho} g^T \nabla V_i - \frac{1}{2}\tilde{\rho}^{-1}\Delta\right|^2 n + \frac{1}{4}\tilde{\rho}^{-2}|\Delta + e|^2$$

$$\le -Q_0(x) - \varepsilon^2 |x|^2 + \tilde{\rho}^{-2}\Delta \max\{|\Delta|^2, |e|^2\}$$

$$\le -Q_0(x) - \tilde{\rho}^{-2}(\gamma^2 - \max\{|\Delta|^2, |e|^2\})$$

Hence, by defining $\chi_2 = \overline{\alpha} \circ (\gamma - \kappa_2)^{-1} \circ \kappa_1 \circ \underline{\lambda}^{-1}$, it follows that

$$V_i(x) \ge \max\{\chi_2(W(w)), \overline{\alpha} \circ (\gamma - \kappa_2)^{-1}(|e|)\},$$

$$\Leftrightarrow V_i(x) \ge \overline{\alpha} \circ (\gamma - \kappa_2)^{-1} \circ \max\{\kappa_1 \circ \underline{\lambda}^{-1}(W(w)), |e|\}$$

$$\Rightarrow (\gamma - \kappa_2) \circ \overline{\alpha}^{-1}(V_i(x)) \ge \max\{\kappa_1 \circ \underline{\lambda}^{-1}(W(w)), |e|\}$$

$$\Rightarrow \gamma(|x|) - \kappa_2|x| \ge \max\{\kappa_1 \circ \underline{\lambda}^{-1}(W(w)), |e|\}$$

$$\Rightarrow \gamma(|x|) - \kappa_2|x| \ge \max\{\kappa_1|w|, |e|\}$$

$$\Rightarrow \gamma(|x|) \ge \max\{|\Delta(w, x)|, |e|$$

$$\Rightarrow \nabla V_i^T [f + g(u_{r,i+1}, \Delta)] \le -Q_0(x) \quad (1.50)$$

Finally, by the gain condition, we have

$$\gamma > \kappa_1 \circ \underline{\lambda}^{-1} \circ \kappa_3 \circ \underline{\alpha}^{-1} \circ \overline{\alpha} + \kappa_2$$

$$\Rightarrow Id > (\gamma - \kappa_2)^{-1} \circ \kappa_1 \circ \underline{\lambda}^{-1} \circ \kappa_3 \circ \underline{\alpha}^{-1} \circ \overline{\alpha}$$

$$\Rightarrow Id > \overline{\alpha} \circ (\gamma - \kappa_2)^{-1} \circ \kappa_1 \circ \underline{\lambda}^{-1} \circ \kappa_3 \circ \underline{\alpha}^{-1} \circ \overline{\alpha}$$

$$\Rightarrow Id > \chi_2 \circ \chi_1 \quad (1.51)$$

The proof is thus completed by the small-gain theorem [13].

Similarly as in the previous section, along the solution of the system (1.39) and (1.46), it follows that

$$\dot{V} = \nabla V^T (f + g u_{r,i})$$

$$= \nabla V^T (f + g u_i) + \nabla V^T g\tilde{e}$$

$$= -r(x, u_i) - \mathcal{L}(V, u_i) + \nabla V^T g\tilde{e}$$

$$= -r(x, u_i) - \mathcal{L}(V, u_i) + 2(\tfrac{1}{2}R^{-1}g^T\beta\nabla V)^T R\tilde{e}$$

$$= -r(x, u_i) - \iota(p, k_i)^T [x]_{2,2d} - 2[x]_{1,d}^T \kappa(p)^T R\tilde{e} \quad (1.52)$$

where $\tilde{e} = \rho^2(|x|^2) - 1$.

Therefore, we can redefine the data matrices as follows. Indeed, define

$$\bar{\sigma}_e = -[\bar{\sigma}^T 2\sigma^T \otimes e^T R]^T \in R^{l_1 + ml},$$

$$\bar{\Phi}_i = [\int_{t0,i}^{t1,i} \bar{\sigma}_e\, dt \int_{t1,i}^{t2,i} \bar{\sigma}_e\, dt \ldots \int_{tqi-1,i}^{tqi,i} \bar{\sigma}_e]^T \in R^{qi \times (l_1 + ml)},$$

$$\bar{\Xi}_i = [\int_{t0,i}^{t1,i} r(x, u_i)dt \int_{t1,i}^{t2,i} r(x, u_i)dt \ldots \int_{tqi-1,i}^{tqi,i} r(x, u_i)dt]^T \in R^{qi},$$

$$\bar{\Theta}_i = [\bar{\Theta}(x)|_{t0,i}^{t1,i} [\bar{\Theta}(x)|_{t1,i}^{t2,i} \ldots\ldots]^T \in R^{q_i \times N_1}$$

Then, the global robust adaptive dynamic programming algorithm is given below.

**Algorithm 5.1.3. The global robust adaptive dynamic programming algorithm**

1: Initialization: Let $p_0$ and $k_1$ satisfying $V_0 = p_0^T \bar{\emptyset}$, $u_1 = k_1 \sigma$, and $\mathcal{L}(V_0, u_1)$, and let i = 1.

2: Collect online data: Apply $u = u_{r,i} = \rho^2(|x|^2)u_i + e$ to the system and compute the data matrices $\Phi_i$, $\Xi_i$, and $\Theta_i$ until the rank condition in Assumption 4.1.1, is satisfied.

3: Policy evaluation and improvement: Find an optimal solution $(p_i, h_i, k_{i+1})$ to the following optimization problem

$$\min_{p,h,k} \; c^T p \tag{1.53}$$

$$\text{s:t: } \begin{bmatrix} h \\ vec(k) \end{bmatrix} = (\bar{\Phi}_i^T \bar{\Phi}_i)^{-1} \bar{\Phi}_i^T (\bar{\Xi}_i + \bar{\Theta}_i p) \tag{1.54}$$

$$h \in S^+ \sigma \tag{1.55}$$

$$p_{i-1} - p \in S^+ \emptyset \tag{1.56}$$

Then, denote $V_i = p_i \bar{\emptyset}$ and $u_{i+1} = k_{i+1} \bar{\sigma}$.

4: Go to Step 2) with $i \leftarrow i + 1$.

## 6. NUMERICAL EXAMPLE

This section provides a numerical example to illustrate the effectiveness of the proposed algorithms[17].

### 6.1 Jet engine dynamics

Consider the following system, which is inspired by the jet engine surge and stall dynamics in [14, 15]

$$\dot{x} = -ax^2 - ax(2y + y^2) \tag{1.57}$$

$$\dot{y} = -by^2 - cy^3 - (u + 3xy + 3x) \tag{1.58}$$

where x > 0 is the normalized rotating stall amplitude, y is the deviation of the scaled annulus-averaged flow, u is the deviation of the plenum pressure rise and is treated as the control input, $a \in [0.2, 0.5]$, $b \in [1.2, 1.6]$, $c \in [0.3, 0.7]$ are uncertain constants.

In this example, we assume the variable x is not available for real-time feedback control due to a 0.2s time-delay in measuring it. Hence, the objective is to find a control policy that only relies on y. The cost function we used here is

$$J = \int_0^\infty (5y^2 + u^2)\, dt \tag{1.59}$$

and an initial control policy is chosen as

$$u_{r,1} = \frac{1}{2}\rho^2(y^2)(2p - 1.47p^2 - 0.45x^3) \tag{1.60}$$

with $\rho(s) = \sqrt{2}$.

Only for the purpose of simulation, we set a= 0:3, b = 1:5, and c = 0:5. The control policy is updated every 0:25s. The simulation results are provided in Chart 1 and chart 2. It can be seen that the system performance has been improved via online learning.

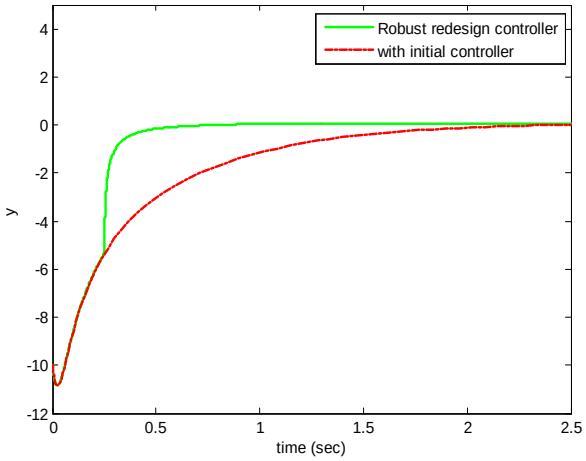


**Chart -1**: Simulation of the jet engine: Trajectories of y

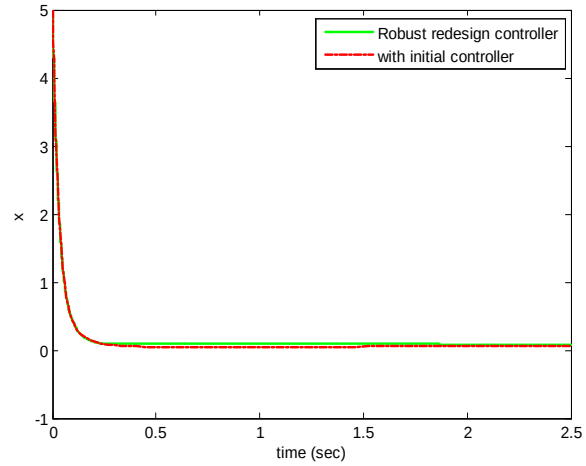


**Chart -2**: Simulation of the jet engine: Trajectories of x

## 8. CONCLUSIONS

This paper has proposed a Novel method of dynamic programming for the adaptive optimal control of nonlinear systems. In particular, a new policy iteration scheme has been developed. Different from conventional policy iteration, the new iterative technique does not attempt to solve a partial differential equation but a convex optimization problem at each iteration step.

It has been shown that, this method can find a suboptimal solution to continuous-time nonlinear optimal control problems [1]. In addition, the resultant control policy is globally stabilizing. In the presence of dynamic uncertainties, robustification of the proposed algorithms and their online implementations has been addressed, by integration with the ISS property [11, 12] and the nonlinear small-gain theorem [9, 13]. When the system parameters are unknown, conventional ADP methods utilize neural networks to approximate online the optimal solution, and a large number of basis functions are required to assure high approximation accuracy on some compact sets. Thus, neural-network-based

ADP schemes may result in slow convergence and loss of global asymptotic stability for the closed-loop system. Here, the proposed method has overcome the two above-mentioned shortcomings, and it yields computational benefits.

## REFERENCES

[1] F.L. Lewis, D. Vrabie, and V.L.Syrmos. Optimal Control, 3rd ed. Wiley, New York, 2012.

[2] R. Sepulchre, M. Jankovic, and P. Kokotovic. Constructive Nonlinear Control. Springer Verlag, New York, 1997.

[3] H. K. Khalil. Nonlinear Systems, 3rd Edition. Prentice Hall, Upper Saddle River, NJ, 2002.

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

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

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

[7] Y. Jiang and Z. P. Jiang. Robust adaptive dynamic programming and feedback stabilization of nonlinear systems. IEEE Transactions on Neural Networks and Learninig Systems, 25(5):882-893, 2014

[8] G. Blekherman, P. A. Parrilo, and R. R. Thomas, editors. Semidefinite Optimization and Convex Algebraic Geometry. SIAM, Philadelphia, PA, 2013.

[9] Z. P. Jiang, A. R. Teel, and L. Praly. Small-gain theorem for ISS systems and applications. Mathematics of Control, Signals and Systems, 7(2):95-120, 1994.

[10] L. Praly and Y. Wang. Stablization in spite of matched unmodeled dynamics and an equivalent definition of input-tp-state stability. Mathematics of Control, Signals and Systems, 9(1):1-33, 1996.

[11] E. D. Sontag. Smooth stabilization implies coprime factorization. IEEE Transactions on Automatic Control, 35(4):473-476, 1990.

[12] E. D. Sontag and Y. Wang. On characterizations of the input-to-state stability property. Systems and Control Letters, 24(5):351-359, 1995.

[13] Z. P. Jiang, I. M. Mareels, and Y. Wang. A Lyapunov formulation of the nonlinear small-gain theorem for interconnected ISS systems. Automatica, 32(8):1211-1215, 1996.

[14] M. Krstic, D. Fontaine, P. V. Kokotovic, and J. D. Paduano. Useful nonlinearities and global stabilization of bifurcations in a model of jet engine surge and stall. IEEE Transactions on Automatic Control, 43(12):1739-1745, 1998.

[15] F. Moore and E. Greitzer. A theory of post-stall transients in axial compression systems: Part 1 development of equations. Journal of engineering for gas turbine and power, 108(1):68-76, 1986.

[16] Y. Jiang and Z. P. Jiang. Global adaptive dynamic programming for continuous time nonlinear systems. IEEE Transactions on Automatic Control, vol 60, No. 11, November 2015.

[17] Y. Jiang. Robust Adaptive Dynamic Programming for Continuous-Time Linear and Nonlinear Systems. Phd Thesis, UMI Dissertation Publishing, ProQuest CSA, 789 E. Eisenhower Parkway.

"Mr. Desai Feroz was born in India, in 1989. He obtained the BTech degree in EIE from JNTU, Hyderabad. Currently, he is a final year MTech candidate working in Control and Simulation Lab (CS lab) at Mallareddy Engineering College, Hyderabad, Telangana, India."

"Mrs. M. Lakshmiswarupa, Associate Professor & HOD, M.Tech (Phd), LMISTE, MIEEE, currently is the HOD of EEE department in Mallareddy engineering college. She has published 10 papers in international journals and 15 papers in international conferences."