2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Online Adaptation of Controller Parameters Based on Approximate Dynamic Programming

Wentao Guo, *Student Member, IEEE,* Feng Liu, *Member, IEEE,* Jennie Si, *Fellow, IEEE,* and Shengwei Mei, *Senior Member, IEEE*

***Abstract*—Controller parameter tuning is an integral part of control engineering practice. Existing tuning methods usually start with an accurate mathematical model of the controlled system, which may pose some challenges for practicing engineers dealing with real systems. As such, parameter optimization and adaptation are treated as two independent steps during tuning. To address these issues, we propose a new, online parameterized controller tuning method for a general nonlinear dynamic system. This tuning method is based on direct heuristic dynamic programming (direct HDP), a model-free algorithm in the approximated dynamic programming (ADP) family. By using a Lyapunov stability approach, we provide uniformly ultimately bounded (UUB) results under some mild conditions for controller parameters, the critic neural network weights, and the action neural network weights. Simulation studies based on the benchmark cart-pole system demonstrate adaptability and optimization capabilities of the proposed controller parameter tuning method.**

I. Introduction

APPROXIMATE dynamic programming (ADP) [1], or adaptive dynamic programming [2], [3], serves as a powerful tool for optimal policy searching in a Markov decision process (MDP). ADP overcomes limitations suffered by classic dynamic programming such as the "curse of dimensionality" [1], [4] by using learning and function approximation techniques. In [5], ADP design approaches are categorized into heuristic dynamic programming (HDP), dual heuristic dynamic programming (DHP), globalized dual heuristic dynamic programming (GDHP) and their action dependent variants of ADHDP, ADDHP and ADGDHP. The value function and its derivative are approximated in HDP and DHP, respectively. GDHP approximates both the value function and its derivative. In the action dependent versions of ADP, control variables are taken into account in the value function approximation. Excellent surveys on ADP can be found in [1], [2], [3].

Optimal adaptive control is an important application field of ADP. A variety of ADP based optimal adaptive control methods have been developed, including adaptive critic designs (ACD) [5], direct HDP [6], goal representation based ADP [7], time-based ADP [8], and the optimal adaptive control based on policy iteration and value iteration [2], [9], [10]. In these methods, neural networks (NNs), such as multilayer perceptrons (MLPs) and radial basis functions (RBFs), are used extensively to approximate control policies. Such neural network based controllers are effective owing to their universal approximation capability. However, neural networks are considered "black box" models in parameters and functional basis, which is not intuitive for result interpretation and gaining insight. On the other hand, the proportional-integral-derivative (PID) controller has been the work horse in engineering fields for its structural simplicity and straightforward interpretations. Tuning PID controllers in real time however, still poses challenges. Taking the best of the two world, the optimal online learning and tuning based on ADP and the structural simplicity of existing controllers such as the PID, we propose a generic method of parameterized controller tuning based on ADP.

Great attention has been paid to incorporate the learning capability of ADP into some existing controllers. In [11], the action neural network in direct HDP is replaced by a phase-shift neural network to imitate the classic lead-lag control. A novel neural network with augmented states is developed in [12] for utilizing the prior knowledge of a well-designed PID controller. In [13], adaptive-heuristic-critic (AHC) based learning algorithm is used to tune the PID controller of an unmanned ground vehicle. It should be noted that the methods in [11], [12], [13] are all designed for specific controllers, such as PID or lead-lag controllers. Besides, a reference model is required in [13].

In [14], the authors proposed a direct HDP based tuning method for the PD-type virtual inertia control (VIC) of doubly fed induction generators (DFIGs). The proposed method shows great potential in tuning parameterized controllers. In this paper, we extend the work in [14] to a general model-free online tuning method for parameterized controllers. An analysis based on Lyapunov approaches is made to prove uniformly ultimately bounded (UUB) convergence of the proposed tuning method.

In fact, great effort has been made in the classic control community to develop systematic tuning methods for parameterized controllers. Taking the most widely used PID controller for example, its tuning methods include optimization method [15], [16], heuristic method [17], [18], frequency domain based method [19], [20], time domain

This work was supported in part by the National Natural Science Foundation of China (No. 51377092, No. 51321005), the Major Projects on Planning and Operation Control of Large Scale Grid (SGCC-MPLG017-2012) of the State Grid Corporation of China, and the National High Technology Research and Development of China (863 Program, 2012AA050204).

W. Guo, F. Liu, and S. Mei are with State Key Laboratory of Power Systems, Department of Electrical Engineering, Tsinghua University, Beijing 100084, China (e-mail: gwt329@gmail.com, lfeng@tsinghua.edu.cn, meishengwei@tsinghua.edu.cn).

J. Si is with Department of Electrical Engineering, Arizona State University, Tempe, AZ 85287, USA (e-mail: si@asu.edu).

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

based method [21], methods based on internal model control (IMC) [22] and model predictive control (MPC) [23]. Compared to these methods, the proposed ADP tuning method in this paper has the following advantages. 1) The mathematic model of the controlled system is not required. 2) The design is online conducted to optimize a designer-specified objective function. 3) Controller parameters can be adapted to system changes. In summary, the proposed method is an online model-free optimal adaptive tuning method for general parameterized controllers.

The rest of this paper is organized as follows. In Section II, the parameter tuning problem is formulated in mathematical terms. Framework and implementation details of the proposed parameter tuning method are also presented. In Section III, convergence results based on Lyapunov function analysis are presented. A parameter selection guidance for the online tuning method is also given. Simulation results on a benchmark cart-pole system are presented in Section IV. Conclusions are drawn in Section V.

II. ADP Based Controller Parameter Tuning Method

A. Problem formulation

In this paper, we consider the discrete-time nonlinear system (1) with the parameterized controller (2),

$$x(t+1) = F(x(t), v(t)), \tag{1}$$

$$v(t) = h(x(t), K), \tag{2}$$

where t is the discrete time, $x \in \mathbb{R}^n$ is the state vector, $v \in \mathbb{R}^m$ is the control vector, $K \in \mathbb{R}^l$ is the control parameter vector, $F(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^m \longrightarrow \mathbb{R}^n$ is the system state transformation function, $h(\cdot,\cdot) : \mathbb{R}^n \times \mathbb{R}^l \longrightarrow \mathbb{R}^m$ is the control policy.

For system (1), an instantaneous cost function at time t can be defined, which is denoted as $r(t)$, which satisfies that

$$0 \le r(t) \le r_m, \tag{3}$$

where r_m is the upper bound of $r(t)$. The definition of $r(t)$ is specified by the controller designer in accordance with the application in mind. Usually, state deviation and control effort are considered in $r(t)$.

Total cost function, or cost-to-go function, is defined as

$$J(t) = \sum_{i=1}^{\infty} \alpha^{i-1} r(t+i), \tag{4}$$

where $0 < \alpha < 1$ is the discount factor. Note that $J(t)$ is the total cost from time $t+1$.

The optimal parameter tuning problem for the parameterized controller (2) is to find a control policy (2) as a function of the parameter K so that system (1) is stable and the cost-to-go is minimized. Accordingly we have the optimal cost-to-go function,

$$J^*(t) = \min_K J(t), \tag{5}$$

and the optimal control parameter can be computed from

$$K^* = \arg\min_K J(t). \tag{6}$$

B. Tuning method based on ADP

In this paper, an ADP based controller parameter tuning method is proposed for the optimal parameter tuning problem formulated above. A schematic diagram of the ADP based controller parameter tuning method is illustrated in Fig. 1.

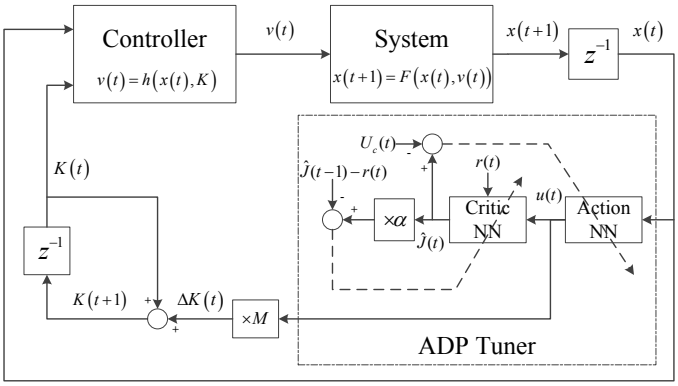

Fig. 1. Schematic of ADP based controller parameter tuning

In the proposed method, the state variable x is taken as input of the ADP tuner. A "proportional-integrator" type structure [24] is employed to update the controller parameter,

$$K(t+1) = K(t) + \Delta K(t), \tag{7}$$

where $K(0)$ is the original controller parameter setting, and the regulatory signal ΔK is obtained by

$$\Delta K(t) = M u(t), \tag{8}$$

where $M \in \mathbb{R}^+$ is a modulation factor, $u(t)$ is generated by the ADP tuner.

The ADP tuner in Fig. 1 is based on direct HDP [6]. In Fig. 1, the solid lines indicate paths for signal flow, while the dashed lines are paths for network parameter tuning. Two neural networks are involved in direct HDP, i.e., the critic neural network and the action neural network.

Critic NN takes $x(t)$ and $u(t)$ as inputs. Output of critic NN, $\hat{J}(t)$, is used to approximate the cost-to-go function $J(t)$ defined in (4). When critic NN is well trained, $\hat{J}(t)$ should satisfy the Bellman equation [4]

$$\hat{J}(t-1) = r(t) + \alpha \hat{J}(t). \tag{9}$$

That is

$$\hat{J}(t) = r(t+1) + \alpha \hat{J}(t+1) = \cdots = \sum_{i=1}^{\infty} \alpha^{i-1} r(t+i). \tag{10}$$

By comparing (4) and (10), there is $\hat{J}(t) = J(t)$ when critic NN is perfectly trained.

Action NN converts the state variable $x(t)$ to the control variable $u(t)$. Its objective is to minimize the difference between the estimated cost-to-go $\hat{J}(t)$ and its objective value $U_C(t)$. Without any loss of generality, $U_C(t)$ is set to 0 in this paper.

From the above, we can see that critic NN evaluates the current control policy and action NN tries to minimize the estimated cost-to-go. Converged action NN and critic NN thus provide a near optimal control policy.

C. Online learning implementation of action NN and critic NN

Both critic NN and action NN are realized by three-layer perceptrons. A sigmoid activation function is used for the hidden layer nodes,

$$\varphi(x) = \frac{1 - e^{-x}}{1 + e^{-x}}. \tag{11}$$

In the critic NN, $w_{cij}^{(1)}$ is the network weight between the jth input node and the ith hidden node, $w_{ci}^{(2)}$ is the network weight between the ith hidden node and the output node, signals q and p are the input and output of the hidden nodes, respectively. The approximation function of critic NN can be expressed as

$$\begin{aligned} &\hat{J}(t) = \sum_{i=1}^{N_{hc}} w_{ci}^{(2)}(t) p_i(t) \\ &p_i(t) = \varphi(q_i(t)), i = 1, \cdots, N_{hc} \\ &q_i(t) = \sum_{j=1}^{n} w_{cij}^{(1)}(t) x_j(t) + \sum_{j=n+1}^{n+l} w_{cij}^{(1)}(t) u_{j-n}(t), \\ &i = 1, \cdots, N_{hc}, \end{aligned} \tag{12}$$

where N_{hc} is the number of hidden nodes in critic NN, subscript of $p(t)$, $q(t)$, $x(t)$, $u(t)$ denotes the element number in the corresponding vector. For example, $p_i(t)$ is the ith element in $p(t)$.

In the action NN, the weight $w_{aij}^{(1)}$ connects the jth input node and the ith hidden node, and $w_{aik}^{(2)}$ connects the ith hidden node and the kth output node, signals h and g are the input and output of the hidden nodes, respectively. The approximated function of action NN can be expressed as

$$\begin{aligned} &u_k(t) = \sum_{i=1}^{N_{ha}} w_{aik}^{(2)}(t) g_i(t), k = 1, \cdots, l \\ &g_i(t) = \varphi(h_i(t)), i = 1, \cdots, N_{ha} \\ &h_i(t) = \sum_{j=1}^{n} w_{aij}^{(1)}(t) x_j(t), i = 1, \cdots, N_{ha}, \end{aligned} \tag{13}$$

where N_{ha} is the number of hidden layer nodes in action NN, subscript of $g(t)$, $h(t)$, $x(t)$, $u(t)$ denotes the element number in the corresponding vector. For example, $g_i(t)$ is the ith element in $g(t)$.

In [25], it is proved that if the number of hidden layer nodes is large enough, and the input-to-hidden layer weights are randomly initialized and fixed, then approximation error of the neural network can be arbitrarily small by adjusting the hidden-to-output layer weights. Based on this result, we only consider updating the hidden-to-output layer weights of the critic NN and the action NN in direct HDP. For the ease of discussion, we denote the estimated $w_c^{(2)}$ as $\hat{w}_c \in \mathbb{R}^{N_{hc}}$, and the estimated $w_a^{(2)}$ as $\hat{w}_a \in \mathbb{R}^{N_{ha} \times l}$. The hidden layer outputs of critic NN and action NN are denoted as $\phi_c = (p_1, p_2, \cdots, p_{N_{hc}})^T$ and $\phi_a = (g_1, g_2, \cdots, g_{N_{ha}})^T$, respectively. Then the output of critic NN, action NN and the estimated controller parameter $\hat{K}$ can be expressed as

$$\hat{J}(t) = \hat{w}_c^T(t) \phi_c(t), \tag{14}$$

$$u(t) = \hat{w}_a^T(t) \phi_a(t), \tag{15}$$

$$\hat{K}(t+1) = \hat{K}(t) + M \hat{w}_a^T(t) \phi_a(t). \tag{16}$$

The approximation error of critic NN is defined as the Bellman residual [4]

$$e_c(t) = \alpha \hat{J}(t) - [\hat{J}(t-1) - r(t)]. \tag{17}$$

Critic NN is trained to minimize the following square error

$$E_c(t) = e_c^2(t)/2. \tag{18}$$

Gradient decent algorithm is employed to train critic NN,

$$\begin{aligned} &\hat{w}_c(t+1) = \hat{w}_c(t) - \alpha l_c \phi_c(t) \\ &[\alpha \hat{w}_c^T(t) \phi_c(t) + r(t) - \hat{w}_c^T(t-1) \phi_c(t-1)], \end{aligned} \tag{19}$$

where l_c is the learning rate for critic NN.

For action NN, the approximation error is defined as

$$e_a(t) = \hat{J}(t) - U_C(t). \tag{20}$$

Action NN is trained to minimize the square error

$$E_a(t) = e_a^T(t) e_a(t)/2. \tag{21}$$

The gradient decent training algorithm for action NN is

$$\hat{w}_a(t+1) = \hat{w}_a(t) - l_a [\hat{w}_c^T(t) \phi_c(t)] \phi_a(t) [\hat{w}_c^T(t) C(t)], \tag{22}$$

where l_a is the learning rate for action NN, and the elements of $C(t) \in \mathbb{R}^{N_{hc} \times l}$ are

$$C_{ij}(t) = \frac{1}{2}(1 - \phi_{ci}^2(t)) w_{ci(n+j)}^{(1)}, i = 1, \cdots, N_{hc}; j = 1, \cdots, l. \tag{23}$$

For convergence consideration, the estimated parameters of critic NN and action NN, $\hat{w}_c$ and $\hat{w}_a$, are confined in an appropriate range by

$$\hat{w}_c(t) = \frac{\hat{w}_c(t)}{max|\hat{w}_c(t)|}, if\ max|\hat{w}_c(t)| > \hat{w}_{cm}, \tag{24}$$

$$\hat{w}_a(t) = \frac{\hat{w}_a(t)}{max|\hat{w}_a(t)|}, if\ max|\hat{w}_a(t)| > \hat{w}_{am}, \tag{25}$$

where $max|\cdot|$ is the largest absolute value of the components of the argument $\hat{w}_{cm}$ and $\hat{w}_{am}$ are the weight bounds for critic NN and action NN, respectively. The estimated controller parameter $\hat{K}(t)$ is also bounded by

$$|\hat{K}_i| \leq \hat{K}_{mi}, i = 1, \cdots, l. \tag{26}$$

where $\hat{K}_{mi} > 0$ is the bound for the ith controller parameter.

III. Convergence Proof

Let w_c^* and w_a^* be the optimal parameters of critic NN and action NN, respectively, where,

$$w_c^* = \arg\min_{w_c} \{\alpha \hat{J}(t) - [\hat{J}(t-1) - r(t)]\} \tag{27}$$

$$w_a^* = \arg\min_{w_a} \hat{J}(t) \tag{28}$$

Then we define the parameter estimation error as $\tilde{w}_c(t) = \hat{w}_c(t) - w_c^*$ and $\tilde{w}_a(t) = \hat{w}_a(t) - w_a^*$. The controller parameter estimation error is denoted as $\tilde{K}(t) = \hat{K}(t) - K^*$. Similar to [26], the UUB convergence properties of $\tilde{w}_c(t)$, $\tilde{w}_a(t)$ and

$\tilde{K}(t)$ are studied by the use of Lyapunov method. Taking the controller parameter error $\tilde{K}(t)$ for example, the definition of UUB is stated as:

Definition 1. (Uniformly Ultimately Bounded (UUB) [27]) The parameter $\tilde{K}(t)$ is said to be UUB if there exists a compact set $\Omega \subset \mathbb{R}^l$ so that for $\forall K(t_0) \subset \Omega$, there exists a bound ϵ and an integer $T(\epsilon) > 0$ such that $\|\tilde{K}(t)\| \leq \epsilon$ for $\forall t \geq t_0 + T(\epsilon)$.

The following assumption is made before the analysis.

Assumption 1. The optimal parameter w_c^* for critic NN in (27), w_a^* for action NN in (28), and K^* for parameterized controller in (6) are bounded, respectively,

$$\|w_c^*\| \leq w_{cm}, \|w_a^*\| \leq w_{am}, \|K^*\| \leq K_m, \tag{29}$$

where $\|\cdot\|$ is the 2-norm, w_{cm}, w_{am} and K_m are constants.

Then we have the following lemmas.

Lemma 1. Consider the critic NN with output (14) and update formula (19). Let

$$L_1(t) = \frac{1}{l_c}\tilde{w}_c^T(t)\tilde{w}_c(t). \tag{30}$$

Then the first-order difference of $L_1(t)$ satisfies that

$$\begin{aligned}\Delta L_1(t) \leq & -\alpha^2\zeta_c^2(t) - \alpha^2(1-\alpha^2 l_c\|\phi_c(t)\|^2)(\zeta_c(t) + \frac{E}{\alpha})^2 \\ & + 2(E + \frac{1}{2}\zeta_c(t-1))^2 + \frac{1}{2}\zeta_c^2(t-1),\end{aligned} \tag{31}$$

where

$$\zeta_c(t) = \tilde{w}_c^T(t)\phi_c(t) \in \mathbb{R}, \tag{32}$$

$$E = \alpha w_c^{*T}(t)\phi_c(t) + r(t) - \hat{w}_c^T(t-1)\phi_c(t-1) \in \mathbb{R}. \tag{33}$$

Proof. By substituting (19) into

$$\Delta L_1(t) = \frac{1}{l_c}(\tilde{w}_c^T(t+1)\tilde{w}_c(t+1) - \tilde{w}_c^T(t)\tilde{w}_c(t)), \tag{34}$$

and applying the Cauchy-Schwarz inequality, (31) can be obtained.

Lemma 2. Consider the action NN with output (15) and update formula (22). Let

$$L_2(t) = \frac{1}{l_a}tr(\tilde{w}_a^T(t)\tilde{w}_a(t)), \tag{35}$$

where $tr(\cdot)$ is the trace of a matrix. Then the first-order difference of $L_2(t)$ satisfies that

$$\begin{aligned}\Delta L_2(t) \leq & 2F^2\|C^T(t)\hat{w}_c(t)\|^2 \\ & + 2\|\hat{w}_a^T(t)\phi_a(t)\|^2 + 2\|w_a^{*T}(t)\phi_a(t)\|^2 \\ & - F^2(1 - l_a\|\phi_a(t)\|^2)\|C^T(t)\hat{w}_c(t)\|^2,\end{aligned} \tag{36}$$

where

$$F = \hat{w}_c^T(t)\phi_c(t) \in \mathbb{R}. \tag{37}$$

Proof. By substituting (22) into

$$\Delta L_2(t) = \frac{1}{l_a}tr(\tilde{w}_a^T(t+1)\tilde{w}_a(t+1) - \tilde{w}_a^T(t)\tilde{w}_a(t)), \tag{38}$$

and applying the property $tr(AB) = tr(BA)$ and the Cauchy-Schwarz inequality, (36) can be obtained.

Lemma 3. Consider the controller parameter update formula in (16). Let

$$L_3(t) = \frac{1}{M}\tilde{K}^T(t)\tilde{K}(t). \tag{39}$$

Then the first-order difference of $L_3(t)$ satisfies that

$$\begin{aligned}\Delta L_3(t) \leq & 2\|\hat{w}_a(t)\hat{K}(t)\|^2 + 2\|\hat{w}_a(t)K^*(t)\|^2 \\ & + \|\phi_a(t)\|^2 + M\|\hat{w}_a^T(t)\phi_a(t)\|^2.\end{aligned} \tag{40}$$

Proof. By substituting (16) into

$$\Delta L_3(t) = \frac{1}{M}(\tilde{K}^T(t+1)\tilde{K}(t+1) - \tilde{K}^T(t)\tilde{K}(t)), \tag{41}$$

and applying the Cauchy-Schwarz inequality, (40) can be obtained.

Based on Lemmas 1-3, the convergence of the proposed parameter tuning method is presented in the following theorem.

Theorem 1. Let the Assumption 1 hold, and the instantaneous cost function be confined by (3). Critic NN with output (14) is updated by (19) and confined by (24). Action NN with output (15) is updated by (22) and confined by (25). The controller parameter is updated by (16) and bounded by (26). Then under conditions specified in (42), the errors between the estimated parameter and the optimal parameter, $\tilde{w}_c(t)$, $\tilde{w}_a(t)$ and $\tilde{K}(t)$, are UUB.

$$\frac{1}{\sqrt{2}} < \alpha < 1, l_c < \frac{1}{\alpha^2\|\phi_c(t)\|^2}, l_a < \frac{1}{\|\phi_a(t)\|^2}. \tag{42}$$

Proof. Construct a Lyapunov function candidate as

$$L(t) = L_1(t) + L_2(t) + L_3(t) + L_4(t), \tag{43}$$

where $L_1(t)$, $L_2(t)$ and $L_3(t)$ are defined in (30), (35) and (39), respectively, and

$$L_4(t) = \frac{1}{2}\zeta_c^2(t-1). \tag{44}$$

According to Lemma 1-3 and $\Delta L_4(t) = \frac{1}{2}\zeta_c^2(t) - \frac{1}{2}\zeta_c^2(t-1)$, there is

$$\begin{aligned}\Delta L(t) \leq & -(\alpha^2 - \frac{1}{2})\zeta_c^2(t) - \alpha^2(1-\alpha^2 l_c\|\phi_c(t)\|^2)(\zeta_c(t) + \frac{E}{\alpha})^2 \\ & - F^2(1 - l_a\|\phi_a(t)\|^2)\|C^T(t)\hat{w}_c(t)\|^2 + D^2,\end{aligned} \tag{45}$$

where D is bounded by a positive constant D_m, i.e. $|D| \leq D_m$.

Then if the condition (42) holds and

$$|\zeta_c(t)| \geq \frac{D_m}{\sqrt{\alpha^2 - \frac{1}{2}}}, \tag{46}$$

there is

$$\Delta L(t) \leq 0. \tag{47}$$

Then the UUB conclusion can be made.

Based on Theorem 1, a more convenient convergence condition is given in the following corollary.

Corollary 1. Let Assumption 1 hold. Consider the instantaneous cost as defined in (3), critic NN output defined in

(14) and weights updated by (19) and bounded by (24), action NN output defined in (15) and weights updated by (22) and bounded by (25), the controller parameter updated by (16) and bounded by (26). Under conditions specified in (48), the errors between the estimated parameters and the optimal parameters for $\tilde{w}_c(t)$, $\tilde{w}_a(t)$ and $\tilde{K}(t)$, are UUB.

$$\frac{1}{\sqrt{2}} < \alpha < 1, l_c < \frac{1}{\alpha^2 N_{hc}}, l_a < \frac{1}{N_{ha}} \tag{48}$$

Proof. According to (11), there is $|\varphi(x)| < 1$. Then $\|\phi_{hc}\| \leq N_{hc}$ and $\|\phi_{ha}\| \leq N_{ha}$. The corollary can be adapted from Theorem 1.

IV. Application in the Cart-Pole System

The proposed ADP based controller parameter tuning method is tested on the cart-pole system, a benchmark system in control theory and application.

A. The cart-pole control problem

The cart-pole system has a pole mounted on a cart, which can move linearly on a track. The control objective is to balance the system at its equilibrium point. Mathematic model of the cart-pole system is [28]

$$\begin{aligned} \ddot{\theta} &= \frac{g\sin\theta + \cos\theta[-v - ml\dot{\theta}^2\sin\theta + \mu_c sgn(\dot{s})] - \frac{\mu_p\dot{\theta}}{ml}}{l(\frac{4}{3} - \frac{m\cos^2\theta}{m_c+m})} \\ \ddot{s} &= \frac{v + ml[\dot{\theta}^2\sin\theta - \ddot{\theta}\cos\theta] - \mu_c sgn(\dot{s})}{m_c + m}, \end{aligned} \tag{49}$$

where the state vector $x = [\theta, \dot{\theta}, s, \dot{s}]^T$ has four variables: the angle of pole with respect to the vertical position θ, the position of cart s, the angular velocity $\dot{\theta}$, and the cart velocity $\dot{s}$; the control input v is the control force on the cart. All other variables and the parameter settings are the same as [6].

The cart-pole system is a nonlinear system. However, when the state trajectory is around its equilibrium point, it can be approximated by a linear system. Linear controllers such as linear quadratic regulator (LQR) can be applied to balance the cart-pole system.

Let the performance index matrixes for LQR be $Q = diag(80, 0, 20, 0)$ and $R = 2$. By solving the algebraic Riccati equation, LQR controller of the system can be obtained as

$$v(t) = 39.1255\theta(t) + 9.8863\dot{\theta}(t) + 3.1623s(t) + 4.4427\dot{s}(t). \tag{50}$$

To test the online parameter optimization capability of the proposed method, the original feedback gain of θ, denoted as $K_\theta(0)$, is set to 22. Then the original controller is

$$v(t) = 22\theta(t) + 9.8863\dot{\theta}(t) + 3.1623s(t) + 4.4427\dot{s}(t). \tag{51}$$

When the initial state is set to $x(0) = [2, 0, 0, 0]^T$, the time domain response under the original controller is illustrated in Fig. 2, which shows a very poor damping performance. In the following the proposed tuning method is used to tune K_θ.

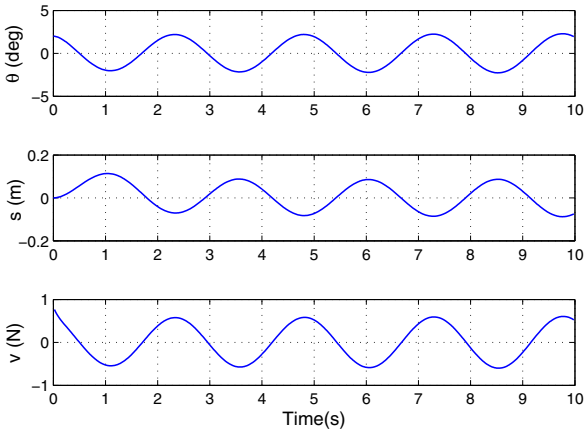

Fig. 2. Time domain response under the controller $v(t) = 22\theta(t) + 9.8863\dot{\theta}(t) + 3.1623s(t) + 4.4427\dot{s}(t)$

B. Simulation results

For implementation of the proposed tuning method, the following settings are used: the numbers of hidden layer nodes $N_{hc} = N_{ha} = 6$, the modulation factor $M = 5$. According to Corollary 1, let $\alpha = 0.95 \in [1/\sqrt{2}, 1]$, $l_c = 0.05 < 1/(0.95^2 \times 6)$, $l_a = 0.05 < 1/6$. The instantaneous cost function $r(t) = 80\theta^2(t-1) + 20s^2(t-1) + 2v^2(t-1)$. The measured states are normalized by $x_N = [12, 120, 2.4, 1.5]^T$ before presented to the action network and the critic network. The ADP tuner is called every 0.02 s.

1) Case 1: In each simulation run, the system starts from the initial state $x(0) = [2, 0, 0, 0]^T$. With ADP tuner embedded in the system, the time domain response during learning is shown in Fig. 3. And the trajectory of K_θ is shown in Fig. 4. It can be seen that the control performance is improved by online tuning the parameter. Optimal K_θ is very close to the value given by LQR. It is because that the error caused by linearization can be ignored around the equilibrium point, and the LQR controller is close to optimal.

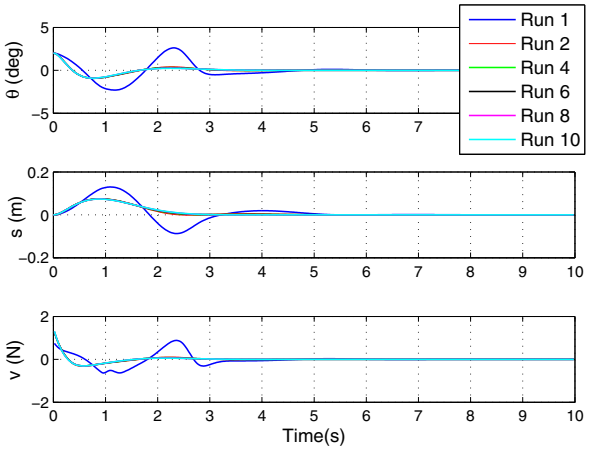

Fig. 3. Time domain response during learning

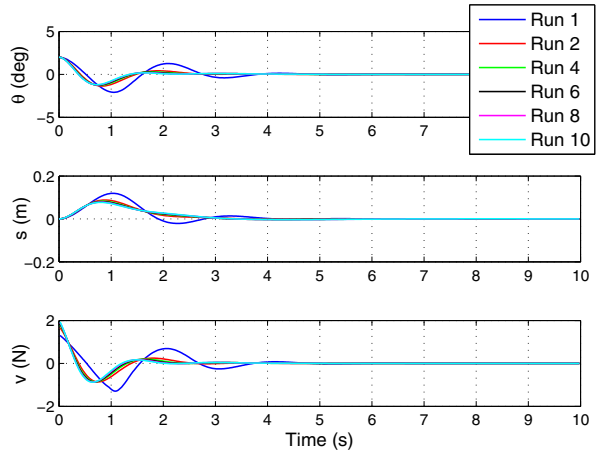

Fig. 6. Time domain response during learning

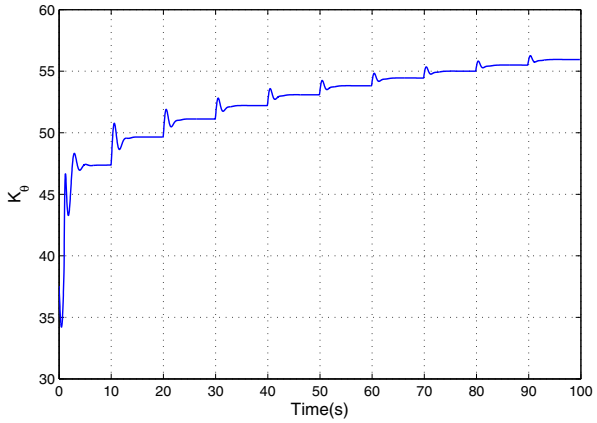

Fig. 7. Trajectory of K_θ after a large disturbance

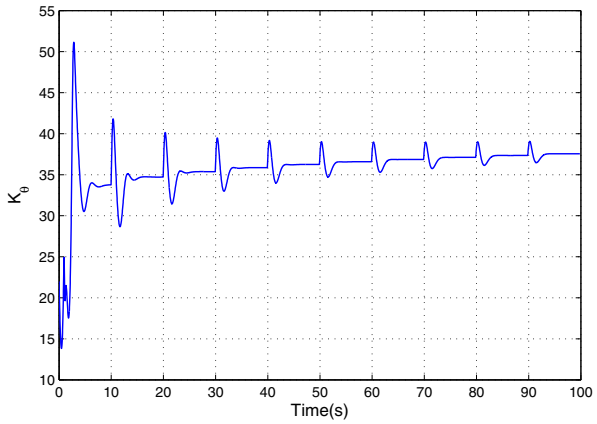

Fig. 4. Trajectory of K_θ during learning

2) Case 2: Then we test the proposed online tuning method under a large disturbance. The cart mass is increased to 2 kg from 1 kg. When using the parameter obtained in case 1, starting from $x(0) = [2, 0, 0, 0]^T$, the time domain response is shown in Fig. 5. It can be seen that the parameter is no longer optimal when the controlled system changes. Then we implement the ADP tuner. The time domain response during learning is shown in Fig. 6. The trajectory of K_θ is shown in Fig. 7. It can be seen that the performance is improved during learning, which demonstrates the online adaptation capability of the ADP tuner.

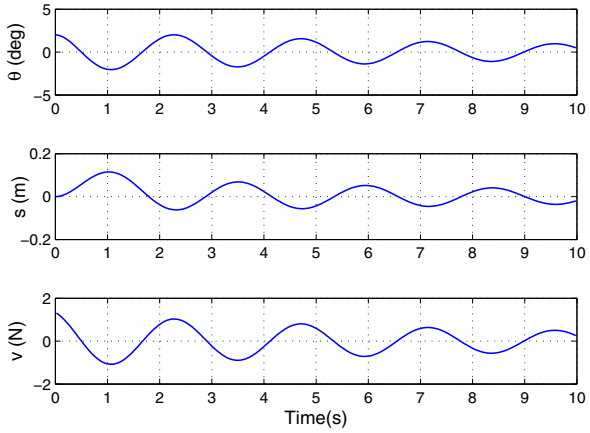

Fig. 5. Time domain response after a large disturbance

V. Conclusion

In this paper, we have developed an online and model-free controller parameter tuning method based on the direct HDP. The online tuning capability provided by the proposed method can complement an existing baseline controller beyond its designed operating range. The UUB stability results of the proposed method are proved by Lyapunov approaches for key variables such as the weights in the online learning controller and the overall controller gain.

By using a benchmark cart-pole system, we demonstrate the effectiveness of the proposed learning controller.

References

[1] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, *Handbook of learning and approximate dynamic programming: scaling up to the real world*. Piscataway, NJ: Wiley-IEEE Press, 2004.

[2] F. L. Lewis, D. Vrabie, and K. G. Vamvoudakis, "Reinforcement learning and feedback control: Using natural decision methods to design optimal adaptive controllers," *IEEE Control Syst. Mag.*, vol. 32, pp. 76–105, Dec. 2012.

[3] F.-Y. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: an introduction," *IEEE Comput. Intell. Mag.*, vol. 4, pp. 39–47, May 2009.

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

[5] D. V. Prokhorov and D. C. Wunsch, "Adaptive critic designs," *IEEE Trans. Neural Netw.*, vol. 8, pp. 997–1007, Sept. 1997.

[6] J. Si and Y.-T. Wang, "Online learning control by association and reinforcement," *IEEE Trans. Neural Netw.*, vol. 12, pp. 264–276, Mar. 2001.

[7] H. He, Z. Ni, and J. Fu, "A three-network architecture for on-line learning and optimization based on adaptive dynamic programming," *Neurocomputing*, vol. 78, pp. 3–13, Feb. 2012.

[8] T. Dierks and S. Jagannathan, "Online optimal control of affine nonlinear discrete-time systems with unknown internal dynamics by using time-based policy update," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, pp. 1118–1129, July 2012.

[9] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, pp. 1513–1525, Oct. 2013.

[10] D. Liu and Q. Wei, "Policy iteration adaptive dynamic programming algorithm for discrete-time nonlinear systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, pp. 621–634, Mar. 2014.

[11] C. Lu, J. Si, and X. Xie, "Direct heuristic dynamic programming for damping oscillations in a large power system," *IEEE Trans. Syst., Man, Cybern. B*, vol. 38, pp. 1008–1013, Aug. 2008.

[12] J. Sun, F. Liu, J. Si, and S. Mei, "Direct heuristic dynamic programming with augmented states," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN'2011)*, (San Jose, CA), pp. 3112–3119, July/Aug. 2011.

[13] X. Xu and H.-G. He, "Neural-network-based learning control for the high-speed path tracking of unmanned ground vehicles," in *Proc. Int. Conf. Mach. Learning Cybern. (ICMLC'2002)*, (Beijing, China), pp. 1652–1656, Nov. 2002.

[14] W. Guo, F. Liu, J. Si, and S. Mei, "Incorporating approximate dynamic programming-based parameter tuning into PD-type virtual inertia control of DFIGs," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN'2013)*, (Dallas, TX), pp. 1–7, Aug. 2013.

[15] A. B. Rad, W.-L. Lo, and K. M. Tsang, "Self-tuning PID controller using Newton-Raphson search method," *IEEE Trans. Ind. Electron.*, vol. 44, pp. 717–725, Oct. 1997.

[16] J.-G. Juang, M.-T. Huang, and W.-K. Liu, "PID control using presearched genetic algorithms for a MIMO system," *IEEE Trans. Syst., Man, Cybern. C*, vol. 38, pp. 716–727, Sept. 2008.

[17] Z.-Y. Zhao, M. Tomizuka, and S. Isaka, "Fuzzy gain scheduling of PID controllers," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 1392–1398, Sept./Oct. 1993.

[18] F. Karray, W. Gueaieb, and S. Al-Sharhan, "The hierarchical expert tuning of PID controllers using tools of soft computing," *IEEE Trans. Syst., Man, Cybern. B*, vol. 32, pp. 77–90, Feb. 2002.

[19] K. Kim, P. Rao, and J. A. Burnworth, "Self-tuning of the PID controller for a digital excitation control system," *IEEE Trans. Ind. Appl.*, vol. 46, pp. 1518–1524, July/Aug. 2010.

[20] K. Li, "PID tuning for optimal closed-loop performance with specified gain and phase margins," *IEEE Trans. Control Syst. Technol.*, vol. 21, pp. 1024–1030, May 2013.

[21] G. K. I. Mann, B.-G. Hu, and R. G. Gosine, "Time-domain based design and analysis of new PID tuning rules," *Control Theory and Applications, IEE Proceedings*, vol. 148, pp. 251–261, May 2001.

[22] W. Tan, "Unified tuning of PID load frequency controller for power systems via IMC," *IEEE Trans. Power Syst.*, vol. 25, pp. 341–350, Feb. 2010.

[23] M. R. Katebi and M. H. Moradi, "Predictive PID controllers," *Control Theory and Applications, IEE Proceedings*, vol. 148, pp. 478–487, Nov. 2001.

[24] S. Mohagheghi, Y. del Valle, G. K. Venayagamoorthy, and R. G. Harley, "A proportional-integrator type adaptive critic design-based neurocontroller for a static compensator in a multimachine power system," *IEEE Trans. Ind. Electron.*, vol. 54, pp. 86–96, Feb. 2007.

[25] B. Igelnik and Y.-H. Pao, "Stochastic choice of basis functions in adaptive function approximation and the functional-link net," *IEEE Trans. Neural Netw.*, vol. 6, pp. 1320–1329, Nov. 1995.

[26] F. Liu, J. Sun, J. Si, W. Guo, and S. Mei, "A boundedness result for the direct heuristic dynamic programming," *Neural Networks*, vol. 32, pp. 229–235, Aug. 2012.

[27] F. L. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systemss*. New York, NY: Taylor and Francis, 1999.

[28] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike adaptive elements that can solve difficult learning control problems," *IEEE Trans. Syst., Man, Cybern.*, vol. 13, pp. 834–846, Sept./Oct. 1983.