Nonlinear Dyn (2012) 69:89–97
DOI 10.1007/s11071-011-0248-3

ORIGINAL PAPER

A novel ADP based model-free predictive control

Na Dong · Zengqiang Chen

Received: 12 November 2010 / Accepted: 10 October 2011 / Published online: 5 November 2011

Abstract Dynamic programming is a very useful tool in solving optimization and optimal control problems. Here, the Approximate Dynamic Programming (ADP) and the notion of neural networks based predictive control are combined with a model-free control method based on SPSA (Simultaneous perturbation stochastic approximation), and a novel ADP based model-free predictive control strategy for nonlinear systems is proposed. Dynamic programming is used to adjust the control parameters in the novel model-free control method and the notion of predictive control is introduced to modify the whole control structure. Finally, the proposed ADP based model-free predictive control strategy is applied to solve nonlinear tracking problems and the effectiveness of this novel control method is fully illustrated though simulation tests on two typical nonlinear systems.

Keywords Model-free control · SPSA · ADP · Neural network · Predictive control · Nonlinear tracking problem

N. Dong (✉)
Department of Automation, School of Electrical Engineering and Automation, Tianjin University, 92 Weijin Road, Tianjin, 300071, China
e-mail: alinna1110@gmail.com

Z. Chen
Department of Automation, Nankai University, 94 Weijin Road, Tianjin, 300071, China

1 Introduction

Adaptive control procedures have been developed in a variety of areas for controlling systems with imperfect information about the system (e.g., manufacturing process control, robot arm manipulation, aircraft control, etc.). Such procedures are typically limited by the need to assume that the forms of the system equations are known (and usually linear) while the parameters may be unknown. In complex physical, socioeconomic, or biological systems, however, the forms of the system equations (typically nonlinear) as well as the parameters are often unknown, making it impossible to determine the control law needed in existing adaptive control procedures. This provides the motivation for developing a control procedure that does not require a model for the underlying system.

As part of modern control theory, model-free control is an advanced control strategy. It has a innovative theory compared with other model-based control approaches. Model-free adaptive control technique has excellent performance in the aspect of adaptiveness, robustness, and nonlinearity [1]. Its typical characteristic is model-free, which is suitable for the system of nonlinear, strong coupling, strong interference, and time-varying. At the same time, some complex systems controlled by corresponding model-free controllers are very concise and effective [1].

It has been paid extensive attention since the theory of model-free control had been put forward. In theory, Professor Zhigang Han officially presented in [2] in 1994. Initial research work focused on basic form of model-free control law issues [3]. It proved the stability of a model-free control system, and provided a reasonable valid theorem, and researched the estimation method for characteristic parameter in pan-model. As well as Zhongsheng Hou, who proposed model-free control theory and applications in his Ph.D. thesis [4] in 1993–1994. Han [5] put forward a functional combination of ways based on model-free control. In the area of methods and technologies, [6] uses the I/O data to estimate the pseudo-gradient vector of the process, for the model-free control. Another idea is proposed by Dr. George Cheng [7], which realizes adaptive control for process through introducing into a neural network. Spall and Cristion [8] also proposed a model-free control strategy—the simultaneous perturbation stochastic approximation (SPSA)-based neural network (NN) controller in 1993. Their results had analyzed the concept from a theoretical approach to realization in detail, which laid a solid foundation for the following development and application of the theory.

In an effort to design more advanced model-free control algorithms with the objectives of reduced tracking errors and improved performance, we develop and evaluate a learning control technique that originated from dynamic programming. Dynamic programming is a very useful tool in solving optimization and optimal control problems. Over the years, progress has been made to circumvent the curse of dimensionality by building a system, called critic, to approximate the cost function in dynamic programming (cf. [9–15]). The idea is to approximate dynamic programming solutions by using a function approximation structure to approximate the cost function. A neural network approach for approximate dynamic programming has been developed in the literature. In the early 1970s, Werbos [16, 17] set up the basic strategy of Reinforcement Learning (RL) system for Adaptive Critic Design (ACD). A typical design of ACDs consists of three modules: Critic, Model, and Action. They are neural networks used to approximate the optimal cost function, the plant to be controlled, and the optimal controller, respectively. In ACDs, neural networks are designed to approximate the cost function $J(\cdot)$, to simulate the derivative of $J(\cdot)$, and to estimate the solution of the Hamilton–Jacobi–Bellman equation. The principle of optimal control is used inside the neural networks to build the weight updating law.

Here, the notion of ADP is introduced into a model-free control method based on SPSA [8] to optimize the control parameters. Also, the notion of predictive control [18] is successfully combined with the model-free control method mentioned above to improve the whole control structure, and finally, a novel ADP based model-free predictive control strategy for nonlinear systems is proposed. The newly proposed control strategy is then applied to solve nonlinear tracking problems. Two typical nonlinear systems are introduced for simulation tests and the effectiveness of this novel control method is fully illustrated though the testing results.

The remainder of the paper is organized as follows: Sect. 2 reviews briefly the SPSA-based adaptive controller employed in this study. Section 3 proposes a novel ADP based model-free predictive control strategy. Section 4 provides simulation experiments and analysis of their findings. Section 5 draws conclusions.

2 Introduction of model-free control method based on SPSA

The model-free control method based on SPSA [19] was proposed by Spall in 1993 [8]. This method does not need to establish the mathematical model of controlled plant previously.

2.1 Overview of approach to control without system model

Consider a general discrete-time state space system of the form:

$$\begin{aligned} x_{k+1} &= \phi_k(x_k, u_k, w_k), \quad \text{(state)} \\ y_k &= h_k(x_k, v_k), \quad k = 0, 1, 2, \ldots \ \text{(means)} \end{aligned} \tag{1}$$

where $\phi_k(\cdot)$ and $h_k(\cdot)$ are generally unknown nonlinear functions governing the system dynamics and measurement process, u_k is the control input applied to govern the system at time $k+1$, and w_k and v_k are noise terms (not necessarily serially independent or independent of each other). Based on information contained within measurements and controls up to y_k, and u_{k-1}, the goal is to choose a control u_k in a manner so as to minimize some loss function related to the next measurement y_{k+1}. Often times, this loss function will be one that compares y_{k+1} against a target value t_{k+1}, penalizing deviations between the two.

In the approach here, a function approximator (e.g., neural network, polynomial) will be used to produce the control u_k.

2.2 Formulation of estimation problem for determining FA

Associated with the FA generating u_k will be a parameter vector θ_k, that must be estimated (e.g., the connection weights in a neural network). The adaptive control problem of finding the optimum control at time k is equivalent to finding the θ_k ($\theta_k \in \Re^p$), that minimizes some loss function $L_k(\theta_k)$:

$$L_k(\theta_k) = E\big[(y_{k+1} - t_{k+1})^T A_k (y_{k+1} - t_{k+1}) + u_k^T B_k u_k\big] \tag{2}$$

where A_k and B_k are positive semidefinite matrices reflecting the relative weight to put on deviations from the target and on the cost associated with larger values of u_k. The problem of minimizing $L_k(\theta_k)$ implies that for each k, seek the globally optimal θ_k^* such that

$$g_k(\theta_k) = \frac{\partial L_k}{\partial \theta_k} = \frac{\partial u_k^T}{\partial \theta_k} \cdot \frac{\partial L_k}{\partial u_k} = 0 \quad \text{at } \theta_k = \theta_k^* \tag{3}$$

Since $\phi_k(\cdot)$ and $h_{k+1}(\cdot)$ are generally unknown functions, the term $\frac{\partial L_k(\cdot)}{\partial u_k}$ which involves the terms $\frac{\partial h_{k+1}(\cdot)}{\partial x_{k+1}}$ and $\frac{\phi_k(\cdot)}{\partial u_k}$, is not generally computable. Hence, $g_k(\theta_k)$ is not generally available. To illustrate this fact, consider a simple scalar deterministic version of system (1). Then, under the standard squared-error loss:

$$g_k(\theta_k) = \frac{\partial (y_{k+1} - t_{k+1})^2}{\partial \theta_k} = 2(y_{k+1} - t_{k+1}) \frac{\partial h_{k+1}}{\partial x_{k+1}} \frac{\partial \phi_k}{\partial u_k} \frac{\partial u_k}{\partial \theta_k} \tag{4}$$

Here, consider a stochastic approximation (SA) algorithm of the form:

$$\hat{\theta}_k = \hat{\theta}_{k-1} - a_k (gradient\ approx.)_k \tag{5}$$

to estimate $\{\theta_k\}$, where, $\hat{\theta}_k$ denotes the estimate at the given iteration, $\{a_k\}$ is a scalar gain sequence satisfying certain regularity conditions, and the gradient approximation is such that it does not require knowledge of $\phi_k(\cdot)$ and $h_{k+1}(\cdot)$ in (1).

2.3 Parameter estimation by simultaneous perturbation stochastic approximation

Spall [19] gives a detailed analysis of the SPSA approach to optimization in the classical setting of a time-invariant loss function $L(\cdot)$ and corresponding fixed minimum.

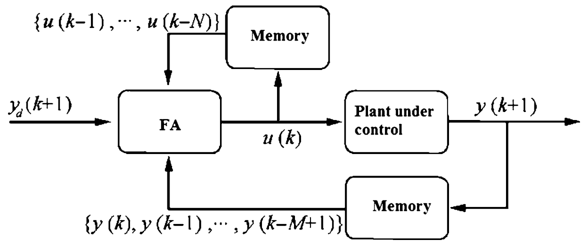

Fig. 1 Model-free control method based on SPSA

The mathematical model of controlled plant is unknown and the controller is a Function approximator. Here, the controller was considered as a multilayer neural network to produce the control u. The number of layers and notes in each layer is fixed previously. The connecting weights are then the control parameter θ, and are allowed to be updated. Its whole structure is shown in Fig. 1.

Suppose the "sliding window" of previous information available at time k, contains M previous measurements and N previous controls. Thus, at time k, the input of neural network is

$$y(k), y(k-1), \ldots, y(k-M+1), u(k-1), u(k-2), \ldots, u(k-N), y_d(k+1) \tag{6}$$

$u(k)$ is the output. The whole structure of the control strategy can be seen in Fig. 2.

$y(k)$ is the output of the controlled plant while $u(k)$ is its input at time k. $y_d(k+1)$ is the expected output at time $k+1$. The aim is to find an optimized control parameter θ_k^*, to minimize the objective function $J_k(\cdot)$ defined as:

$$J_k(\theta_k) = E\big[\big(y(\theta_k, k+1) - y_d(k+1)\big)^2\big] \tag{7}$$

The SPSA based model-free control method uses the following equation to estimate θ_k:

$$\hat{\theta}_k = \hat{\theta}_{k-1} - a_k \hat{g}_k\big(\hat{\theta}_{k-1}\big) \tag{8}$$

$\hat{\theta}_k$ is the estimation. a_k is a scalar gain. $\hat{g}_k(\hat{\theta}_{k-1})$ is the simultaneous perturbation approximation to $g_k(\hat{\theta}_{k-1})$. In particular, the lth component of $\hat{g}_k(\hat{\theta}_{k-1})$, $l = 1, 2, \ldots, L$, is given by

$$\hat{g}_{kl}\big(\hat{\theta}_{k-1}\big) = \frac{\hat{J}_k^{(+)}(\cdot) - \hat{J}_k^{(-)}(\cdot)}{2c_k \Delta_{kl}} \tag{9}$$

$\hat{J}_k^{(\pm)}(\cdot)$ are estimated values of $J_k(\widehat{\bigcap}\theta_{k-1} \pm c_k \Delta_k)$ using the observed $y_{k+1}^{(\pm)}$ and $u_k^{(\pm)}$. $u_k^{(\pm)}$ are controls based on the parameter vector $\theta_k = \hat{\theta}_{k-1} \pm c_k \Delta_k$,

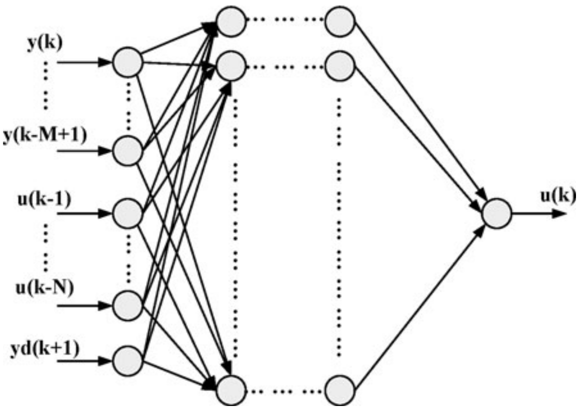

Fig. 2 Structure of the control strategy

where $\Delta_k = [\Delta_{k1}, \Delta_{k2}, \ldots, \Delta_{kL}]^T$, with Δ_{kl} independent, bounded, symmetrically distributed (about 0) random variables $\forall k, l$ identically distributed. $\{c_k\}$ is a sequence of positive numbers (typically $c_k \to 0$ or $c_k = c, \forall k$, depending on whether the system equations are stationary or non-stationary). In the approach above, at each iteration step, only two close-loop tests are needed. The estimation value of $g_k(\hat{\theta}_{k-1})$ (that is $\hat{g}_{kl}(\hat{\theta}_{k-1})$) can then be obtained. The model of the controlled plant is not needed during the whole control procedure.

3 ADP based model-free predictive control

The implementation of approximate dynamic programming usually requires the use of three modules: Critic, Model, and Action [9, 10]. These three modules perform the function of evaluation, prediction, and decision, respectively. Here, the notion of approximate dynamic programming is adopted into the above model-free control method. Artificial neural networks are used to represent the cost function in dynamic programming. The model-free controller, whose structure is actually a neural network, will perform as the action network in the dynamic programming procedure, and its parameters will be modified according to the dynamic programming rules. And thus, a better controller can be anticipated.

3.1 Brief introduction of ADP

In the ADP scheme here, we focus on the discrete-time nonlinear (time-varying) dynamical (deterministic) systems. Consider a discrete-time nonlinear (time-varying) dynamical system with the following formulation:

$$x(t+1) = F\big[x(t), u(t), t\big], \quad t = 0, 1, 2, \ldots \tag{10}$$

where $x \in R^n$ represents the state vector of the system and $u \in R^m$ denotes the control action. Suppose the system's performance cost as:

$$I\big[x(i), i\big] = \sum_{k=i}^{\infty} \gamma^{k-i} U\big[x(k), u(k), k\big] \tag{11}$$

where $U(\cdot)$ is called the utility function and γ is the discount factor with $0 < \gamma \le 1$. Note that the function $I(\cdot)$ is dependent on the initial time i and the initial state $x(i)$, and it is referred to as the cost-to-go of state $x(i)$.

The objective of dynamic programming problem is to choose a control sequence $u(k)$, $k = i, i+1, \ldots,$ so that the function $I(\cdot)$ (i.e., the cost) in (11) is minimized. An optimal (control) policy has the property that no matter what previous decisions have been; the remaining decisions must constitute an optimal policy with regard to the state resulting from those previous decisions.

Suppose that one has computed the optimal cost $I^*[x(t+1), t+1]$ from time $t+1$ to the terminal time, for all possible states $x(t+1)$, and that one has also found the optimal control sequences from time $t+1$ to the terminal time. The optimal cost results when the optimal control sequence $u^*(t+1), u^*(t+2), \ldots,$ is applied to the system with initial state $x(t+1)$. Note that the optimal control sequence depends on $x(t+1)$. If one applies an arbitrary control $u(t)$ at time t and then uses the known optimal control sequence from $t+1$ on, the resulting cost will be:

$$U\big[x(t), u(t), t\big] + \gamma I^*\big[x(t+1), t+1\big] \tag{12}$$

where $x(t)$ is the state at time t and $x(t+1)$ is determined by (10). According to Bellman, the optimal cost from time t on is equal to

$$I^*\big[x(t), t\big] = \min_{u(t)}\Big(U\big[x(t), u(t), t\big] + \gamma I^*\big[x(t+1), t+1\big]\Big) \tag{13}$$

3.2 Adaptive critic designs based on ADP

A typical design of ACDs consists of three modules: Critic, Model, and Action [10, 20], as shown in Fig. 3.

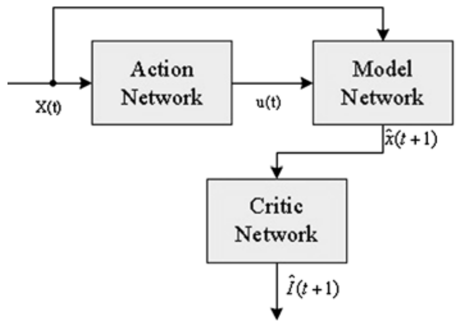

Fig. 3 The three modules of an adaptive critic design

They are neural networks used to approximate the optimal cost function, the plant to be controlled, and the optimal controller, respectively. These three parts combined together form a "Reinforcement Learning System" (RLS) or an ACD. In ACDs, neural networks are designed to approximate the cost function $I(\cdot)$, and to estimate the solution of Hamilton–Jacobi–Bellman equation. Instead of a sequence of optimal controllers, only one controller is trained to provide an approximative optimal control. The critic network outputs the function $\widehat{I}$, which is an estimate of the function I in (11). This is done by minimizing the following square tracking error measure over time:

$$\begin{aligned}\|E_h\| &= \sum_t E_h(t) \\ &= \frac{1}{2}\sum_t \big[\widehat{I}(t) - U(t) - \gamma \widehat{I}(t+1)\big]^2 \end{aligned} \tag{14}$$

In our approach, three neural networks, named $\widehat{F}(x,u)$, $\widehat{I}(x,k)$, and $\widehat{u}(x)$, respectively, are used to approximate the functions $F(x,u)$, $I(x,k)$, and $u(x)$. The neural networks $\widehat{F}(x,u)$, $\widehat{I}(x,k)$, and $\widehat{u}(x)$ will be trained whenever some observations are obtained.

3.3 Neural network based predictive control

The NN-based control strategies have been found to be effective in controlling a wide class of nonlinear processes in the past [21–23]. First, a brief introduction of NN-based predictive control (NNPC) is given in the following section.

Assume that the unknown nonlinear system is expressed as the input-output form by

$$\begin{aligned} y(t) = g\big(&y(t-1),\dots,y(t-n_a), \\ &u(t-\tau-1),\dots,u(t-\tau-n_b)\big) \end{aligned} \tag{15}$$

where $y(t)$ and $u(t)$ are the output and input of the system, respectively, $g(\cdot)$ is the unknown nonlinear function to be estimated by an NN, n_a, and n_b are the orders of the system, and τ is the plant delay as an integer number. The purpose of NNPC is to select signal $u(t)$, such that the output of the system $y(t)$ is made as close as possible to be a prespecified set-point $r(t)$.

Since an NN will be used to model the nonlinear plant, the configuration of the network architecture should be considered. A three-layer feed forward NN, which only has one hidden layer, is used to learn the nonlinear plant, since it has been proved that this is sufficient to represent any nonlinear function providing enough nodes are present. The activation functions are hyperbolic tangent for the hidden layer and linear for the output layer.

Since the input to the NN is:

$$\begin{aligned} \phi = \big[&y(t-1),\dots,y(t-n_a),u(t-\tau-1),\dots, \\ &u(t-\tau-n_b)\big]^T \end{aligned} \tag{16}$$

the neural model for the unknown system (15) can be expressed as

$$\hat{y}(t) = \sum_{j=1}^{n_H} \omega_j^o \cdot f\big(net_j^t\big) + b^o \tag{17}$$

$$\begin{aligned} net_j^t = &\sum_{i=1}^{n_a} \omega_{j,i}^I y(t-i) \\ &+ \sum_{i=1}^{n_b} \omega_{j,i+n_a}^I u(t-\tau-i) + b_j^I \end{aligned} \tag{18}$$

where $f(x) = 1/(1+e^{-x})$, $\hat{y}(t)$ is the NN output, net_j^t is the activation level of the jth nodes output function, n_H is the number of hidden nodes in the hidden layer, ω_j^o is the weight connecting the jth hidden node to the output node and $\omega_{j,i}^I$ the weight connecting the ith input node to the jth hidden node, b_j^I is the bias on the jth hidden node, and b^o is the bias on the output node.

The NNPC uses the output of the NN model to predict the plants dynamics to an arbitrary input from the current time t to some future time $t+k$ ($k = 1,\dots,N_n$), N_n is the prediction horizon. Based on (17) and (18), the k-step-ahead prediction outputs are

$$\hat{y}(t+k) = \sum_{j=1}^{n_H} \omega_j^o \cdot f\big(net_j(t+k)\big) + b^o \tag{19}$$

$$net_j(t+k) = \sum_{i=1}^{\min(k,n_a)} \omega_{j,i}^I y(t+k-i) + \sum_{i=k+1}^{n_a} \omega_{j,i}^I y(t+k-i) + \sum_{i=1}^{n_b} \omega_{j,i+n_a}^I u(t+k-i) + b_j^I \quad (20)$$

3.4 ADP based model-free predictive control strategy

As mentioned in the previous section, the controller of the SPSA based model-free method [8] is a neural network. Here, in order to adjust its control parameter according to the input and output information obtained at each iterative control step, the notion of ACD is incorporated and the original neural network controller in the SPSA based model-free control strategy is fixed directly as the action network in the adaptive critic design of our newly proposed ADP based model-free control strategy. The whole ACD structure acts as the controller in the novel control strategy. At each control iteration, the initial control parameters (weights of the action network) are generated by the SPSA based model-free control rules, and then they are updated by the rules of ACD as described above, taking into account the input and output information (obtained at each iteration) of the system under control.

Also, in the proposed novel control strategy, the notion of neural network based predictive control is introduced to modify the objective function $J(\cdot)$ in (7). The modified objective function, noted as $\tilde{J}(\cdot)$, takes into account the predicted system output error. By minimizing $\tilde{J}(\cdot)$, the controller can have more control power and thus, better control effects can be anticipated. $\tilde{J}(\cdot)$ can be written in the following form:

$$\tilde{J}_k(\theta_k) = E\left[\left(y(\theta_k, k+1) - y_d(k+1)\right)^2 + \lambda \sum_{i=2}^{N_p} \left(\hat{y}(k+i) - y_d(k+i)\right)^2\right] \quad (21)$$

where N_p is the number of prediction steps, $0 < \lambda < 1$ is a scalar number which can decide the portion of the predictive part in the objective function, and $\hat{y}(k+i)$ can be obtained by (19).

A schematic of the proposed ADP based model-free predictive control is given in Fig. 4.

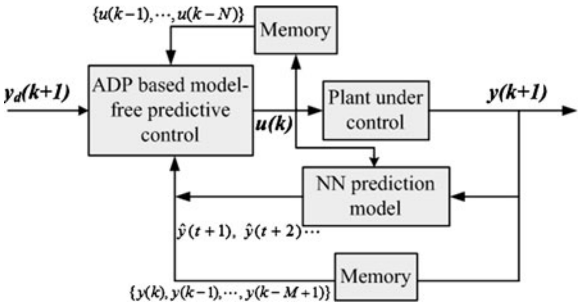

Fig. 4 ADP based model-free predictive control scheme

Based on the above knowledge, the whole ADP based model-free predictive control method can be summarized in the following steps:

A01 Build up a neural network model for the nonlinear system under control;
A02 Train the controller by SPSA based model-free control method with the modified objective function $\tilde{J}(\cdot)$ in (21), and initialize the action network by the trained values of weights;
A03 Initialize neural networks $\widehat{F}(x,u)$ and $\widehat{I}(x,k)$;
A04 Determine the control signal by $u = \widehat{u}(x_0, k)$;
A05 Run the plant to obtain $x_1 = F(x_0, u)$;
A06 Train the model network $\widehat{F}$ by $\widehat{F}(x_0, u) = x_1$;
A07 Adjust the critic network $\widehat{I}$ by minimizing (14);
A08 Adjust the action network $\widehat{u}$ by minimizing $\widehat{I}$ obtained in A07;
A09 Let $x_0 = x_1$. Repeat A05–A08 for preset times;
A10 Obtain the control signal u by the updated action network $\widehat{u}$, feed back u and complete the control loop.

4 Simulations

In order to test the performance of the proposed ADP based model-free predictive control method, a tracking problem for a typical nonlinear system is introduced here:

$$y(k+1) = \frac{y(k)y(k-1)(y(k)-2.5)}{1+y(k)^2+y(k-1)^2} + u(k) \quad (22)$$

The above system equation is not used for control. It is only used to generate the system output data in the simulation tests. The expected output is a step signal and the utility function in the ADP based method is defined as

$$U(k) = \frac{1}{2}\left[y(k) - y_d(k)\right]^2 \quad (23)$$

where y_d is the expected output.

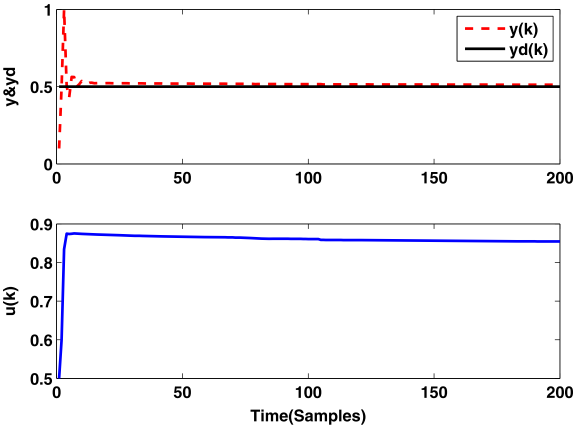

Fig. 5 Tracking result of the SPSA based model-free control method for plant (22)

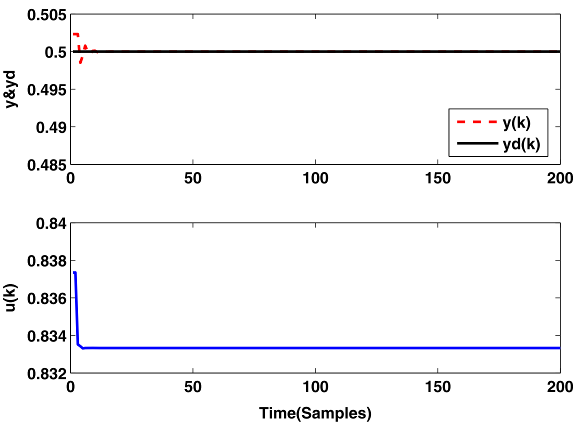

Fig. 6 Tracking result of the ADP based model-free predictive control method for plant (22)

A multilayer neural network with one hidden layer is used as the model network for system (22), and its structure is fixed as $N_{3,3,1}$. The Back Propagation (BP) algorithm [24], which is a simple but quite effective way for training neural networks is used here. Decaying gains are used in order to fulfill the requirements for convergence [25] and parameters are fixed as: $a_k = 0.05/k^{0.602}$, $ck = 0.15/k^{0.101}$ for the SPSA based model-free control.

First, apply the original SPSA based model-free control method [8]. The tracking result and the corresponding control signal is shown in Fig. 5. Then fix the parameter $\gamma = 0.9$, $\lambda = 0.5$, $N_p = 2$ for the proposed ADP based control method, and the corresponding results can be seen in Fig. 6.

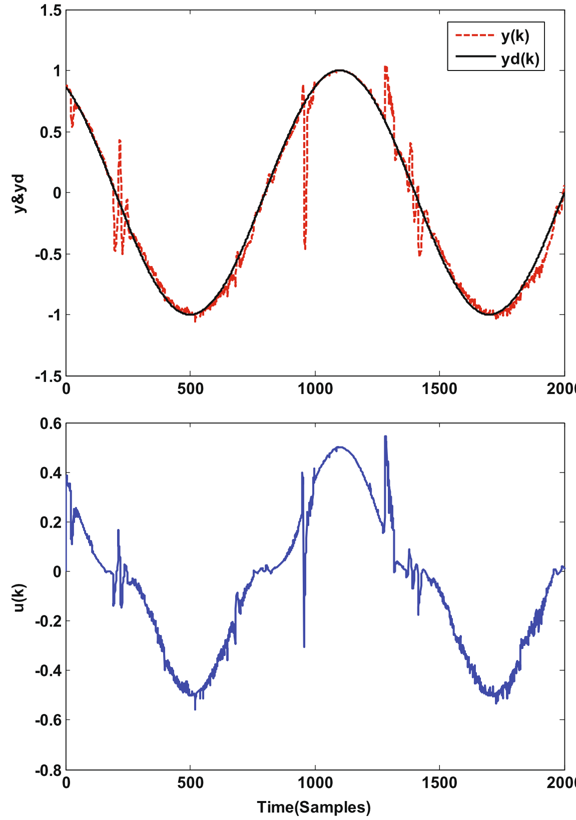

Fig. 7 Tracking result of the SPSA based model-free control method for plant (25)

Table 1 Comparison of the tracking errors

Method	Tracking err.
Original SPSA based control method	0.0043
Novel ADP based control method	9.85e–08

Define the tracking error as

$$Err = \left(\sum_{k=1}^{P} \left(y(k) - y_d(k) \right)^2 \right) \Big/ P \tag{24}$$

where P is the number of the running steps, and the comparison results can be see in Table 1. From the above comparison results, it can be seen obviously that the ADP based control method is more effective and has much higher tracking accuracy.

To further test the effectiveness of the newly proposed control method, another tracking problem is in-

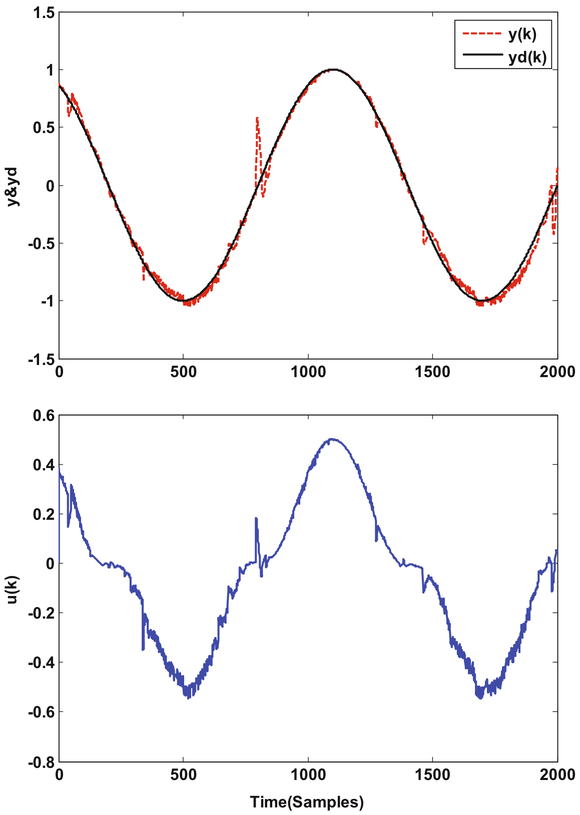

Fig. 8 Tracking result of the ADP based model-free predictive control method for plant (25)

troduced. The nonlinear system under control can be expressed as

$$y(k+1) = \frac{y(k)}{1+y(k)^2} + u(k) + 5u(k-1) \tag{25}$$

The given signal is a sinusoidal signal. The original SPSA based model-free control method and the proposed ADP based model-free predictive control strategy are both applied to solve this control problem. A multilayer neural network with the structure of $N_{3,3,1}$ is used to model this nonlinear system (25) and all the control parameters are fixed the same as for system (22). The tracking results are shown in Figs. 7 and 8 respectively, from which the effectiveness of our newly proposed ADP based control method can be seen obviously.

5 Conclusion

In this paper, a novel ADP based model-free predictive control method is proposed and then applied into nonlinear tracking problems. Two typical nonlinear systems are introduced for simulation tests. The given signals are fixed as step signal and sinusoidal signal, respectively, for these two nonlinear systems under control. Comparisons between the original SPSA based control method and the newly proposed ADP based method are made. Better control performances and much smaller tracking errors are achieved by the novel ADP based model-free predictive control method, through which its effectiveness is fully illustrated.

In our future research work, we will focus on more complex nonlinear tracking problems and try to find more application areas for our proposed ADP based model-free predictive control strategy. Also, we will do some corresponding research regarding the algorithm speed. Another open problem is one common to many applications of function approximators: namely, to develop guidelines for determining the optimal (at least approximately) structure for the FA, e.g., optimal number of hidden layers and nodes in a neural network. We will also conduct some more research in such areas.

Acknowledgements Firstly, we would like to express our sincere gratitude to Prof. Derong Liu, professor of Electrical and Computer Engineering, University of Illinois at Chicago, for his kindly assistance and valuable suggestions during the process of our paper writing.

This work was supported in part by the National High Technology Research Development Project (863 Project) of China Under Grant 2009$AA04Z$132, the Natural Science Foundation of China Under Grant 61174094, and the Specialized Research Fund for the Doctoral Program of Higher Education of China Under Grant 20090031110029.

References

1. Cheng, S.X.: Model-free adaptive (MFA) control. IEE Comput. Control Eng. **15**(3), 28–33 (2004)
2. Han, Z.G., Wang, D.: Controller without model. J. Nat. Sci. Heilongjiang Univ. **11**(4), 29–35 (1994)
3. Shen, Y.L., Wang, D.: The design and realization of controller without model. J. Nat. Sci. Heilongjiang Univ. **15**(2), 24–27 (1998)
4. Hou, Z.S.: The parameter identification, adaptive control and model free learning adaptive control for nonlinear systems. Ph.D. thesis, North-eastern University, Shenyang, China (1994)
5. Han, Z.G.: Designing problem of model free controller. Control Eng. China **9**(3), 19–22 (2002)
6. Hou, Z.S.: Non-parametric model and its adaptive control theory. Science Press, Beijing (1999)

7. Cheng, S.X.: Model-free adaptive process control. United States Patent, 6055524, April 25 (2000)
8. Spall, J.C., Cristion, J.A.: Model-free control of general discrete-time systems. In: Proceedings of the 32nd IEEE Conference on Decision and Control, San Antonio, TX, December 15–17 (1993)
9. Murray, J.J., Cox, C.J., Lendaris, G.G., Seaks, R.: Adaptive dynamic programming. IEEE Trans. Syst. Man Cybern., Part C Appl. Rev. **32**(2), 140–153 (2002)
10. Prokhorov, D.V., Wunsch, D.C.: Adaptive critic designs. IEEE Trans. Neural Netw. **8**(5), 997–1007 (1997)
11. Si, J., Wang, Y.T.: On-line learning control by association and reinforcement. IEEE Trans. Neural Netw. **12**(2), 264–276 (2001)
12. Werbos, P.J.: Building and understanding adaptive systems: a statistical/numerical approach to factory automation and brain research. IEEE Trans. Syst. Man Cybern. **SMC-17**, 7–20 (1987)
13. Hendzel, Z.: An adaptive critic neural network for motion control of a wheeled mobile robot. Nonlinear Dyn. **50**(4), 849–855 (2007)
14. Jin, N., Liu, D.: Discrete-time e-adaptive dynamic programming algorithm using neural networks. In: IEEE International Symposium on Intelligent Control Part of 2008 IEEE Multi-conference on Systems and Control, San Antonio, TX, September 3–5 (2008)
15. Liu, D., Jin, N.: Finite horizon discrete-time approximate dynamic programming. In: Proceedings of the 2006 IEEE International Symposium on Intelligent Control, Munich, Germany, October 4–6 (2006)
16. Werbos, P.J.: Beyond regression: New Tools for Prediction and Analysis in the Behavioral Sciences. Ph.D. thesis, Harvard Univ., Cambridge (1974)
17. Werbos, P.J.: Advanced forecasting methods for global crisis warning and models of intelligence. Gen. Syst. Yearbook **22**, 25–38 (1977)
18. Li, X., Chen, Z.Q., Yuan, Z.Z.: Simple recurrent neural network-based adaptive predictive control for nonlinear systems. Asian J. Control **4**(2), 231–239 (2002)
19. Spall, J.C.: Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. IEEE Trans. Autom. Control **37**(3), 332–341 (1992)
20. Prokhorov, D.V., Wunsch, D.C.: Adaptive critic designs. IEEE Trans. Neural Netw. **8**(5), 997–1007 (1997)
21. Tan, Y., Cauwenberghe, A.: Nonlinear one-step-ahead control using neural networks: control strategy and stability design. Automatica **32**(12), 1701–1706 (1996)
22. Noriega, J.R., Wang, H.: A direct adaptive neural-network control for unknown nonlinear systems and its application. IEEE Trans. Neural Netw. **9**(1), 27–34 (1998)
23. Yildirim, S.: A proposed hybrid neural network for position control of a walking robot. Nonlinear Dyn. **52**(3), 207–215 (2008)
24. Rumelhart, D., McClelland, J.: Parallel Disttibnctl Processing: Explorations in the Micro-Structure of Cognition, vol. 1. MIT Press, Cambridge (1986)
25. Spall, J.C., Cristion, J.A.: Model-free control of nonlinear stochastic systems with discrete-time measurements. IEEE Trans. Autom. Control **43**(9), 1198–1210 (1998)