Neural Comput & Applic (2013) 23:2093–2100
DOI 10.1007/s00521-012-1156-2

ORIGINAL ARTICLE

Optimal control problem via neural networks

Sohrab Effati · Morteza Pakdaman

Received: 22 May 2012 / Accepted: 28 August 2012 / Published online: 23 September 2012

Abstract This paper attempts to propose a new method based on capabilities of artificial neural networks, in function approximation, to attain the solution of optimal control problems. To do so, we try to approximate the solution of Hamiltonian conditions based on the Pontryagin minimum principle (PMP). For this purpose, we introduce an error function that contains all PMP conditions. In the proposed error function, we used trial solutions for the trajectory function, control function and the Lagrange multipliers. These trial solutions are constructed by using neurons. Then, we minimize the error function that contains just the weights of the trial solutions. Substituting the optimal values of the weights in the trial solutions, we obtain the optimal trajectory function, optimal control function and the optimal Lagrange multipliers.

Keywords Pontryagin minimum principle · Optimal control problem · Artificial neural networks

1 Introduction

A very important, extensive and applicable mathematical model is the optimal control problem. There are a wide variety of practical problems arising in science and engineering that have a dynamical system and the dynamic of the system must be controlled to attain an objective. In recent years, several researchers attempted to offer and extend new methods for solving optimal control problem. For example, Krabs et al. [1] proposed a mathematical model for the control of the growth of tumor cells which is formulated as a problem of optimal control theory. Modares et al. [2] presented a hybrid algorithm by integrating an improved particle swarm optimization with successive quadratic programming (SQP), for solving nonlinear optimal control problems.

The solutions of optimal control problems can be calculated either by using Pontryagin's minimum principle (PMP), which provides a necessary condition for optimality, or by solving the Hamilton–Jacobi–Bellman (HJB) partial differential equation (PDE), which is a sufficient condition (see e.g. [3, 4]). Solving the HJB–PDE is a very tedious task. Several approximation methods are proposed for solving it. Hilscher [5] considered Hamilton–Jacobi theory over time scales and its applications to linear-quadratic problems. Based on the variational iteration method, Berkani et al. [6] proposed a method for solving optimal control problems. Garg et al. [7] presented a unified framework for the numerical solution of optimal control problems using collocation at Legendre–Gauss, Legendre–Gauss–Radau, and Legendre–Gauss–Lobatto points. An adaptive multilevel generalized SQP method presented in [8] to solve PDAE-constrained (partial differential algebraic equations) optimization problems. The notion of KT-invexity from mathematical programming was extended to the classical optimal control problem by authors of [9]. Optimal control problem subject to mixed control-state constraints was investigated by Gerdts [10]. He stated the necessary conditions in terms of a local minimum principle and use of the Fischer–Burmeister

S. Effati · M. Pakdaman (✉)
Department of Applied Mathematics,
Ferdowsi University of Mashhad, Mashhad, Iran
e-mail: pakdaman.m@gmail.com

S. Effati
e-mail: s-effati@um.ac.ir

S. Effati · M. Pakdaman
Center of Excellence on Soft Computing
and Intelligent Information Processing,
Ferdowsi University of Mashhad, Mashhad, Iran

function. Buldaev [11] used perturbation methods in optimal control problems. Numerical methods based on extended one-step methods were investigated for solving optimal control problems in [12]. Variational inequalities were used to govern the existence results for optimal control problems in [13]. Chryssoverghi et al. [14] considered an optimal control problem described by nonlinear ordinary differential equations (ODEs) with control and state constraints, including point-wise state constraints. Because their problem may have no classical solutions, they formulated a relaxed form of problem and used discretization method. England et al. [15], in an interesting work, expressed optimal control problems as differential algebraic equations. Local stability of the solution to optimal control problems was analyzed by Rodriguez [16]. An approximate-analytical solution for the HJB equation proposed via homotopy perturbation method in [17]. Cheng et al. in several works [18–20] proposed a neural network solution for different types of optimal control problems. They proposed (in [18]) neural network solution for suboptimal control of non-holonomic chained form systems. In [19], they introduced a neural network solution for finite-horizon H-infinity constrained optimal control of nonlinear systems. Finally in [20], they proposed fixed-final time-constrained optimal control laws using neural networks to solve HJB equations for general affine in the constrained nonlinear systems. Vrabie and Lewis [21] presented a neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems.

In the last decade, artificial neural networks and other elements of soft computing and artificial intelligence played an important role in solving hard to solve problems arising in science and engineering phenomenons. Applying the mentioned methods in many contests was successful, and the results were comparable with the other results obtained by mathematical algorithms. Lagaris et al. [22] used artificial neural networks to solve ODEs and PDEs for both boundary value problems and initial value problems. Vrabie et al. [21] proposed a method for solving continuous-time direct adaptive optimal control for partially unknown nonlinear systems, based on a reinforcement learning scheme.

In Sect. 2, we introduce the optimal control problem and present some basic concepts of neural network models. Section 3 contains the main idea based on neural network models. In Sect. 4, we apply the new method for solving some numerical problems, and finally Sect. 5 contains concluding remarks.

2 Preliminaries

In this paper, we consider the following type of optimal control problem:

$$\begin{aligned} \min & \int_{t_0}^{t_f} f_0(x(t),u(t),t)\,\mathrm{d}t \\ \text{s.t} & \\ & \dot{x} = g(x(t),u(t),t) \\ & x(t_0) = x_0, \end{aligned} \tag{1}$$

where $x(t) \in \Re^n$ is the state variable, $u(t) \in \Re^m$ is the control variable and $t \in \Re$. It is assumed that the integrand f_0 has continuous first and second partial derivatives with respect to all its arguments. Also we assume that t_0 and t_f are fixed and g is Lipschitz continuous on a set $\Omega \subset \Re^n$. According to the problem (1), we can construct the well-known Hamiltonian as: $H(x(t), u(t), p(t), t) = f_0(x(t), u(t), t) + p(t)\cdot g(x(t), u(t), t)$ where $p \in \Re^m$ is the costate vector. Suppose that we denote the optimal state, co-state and control functions by $x^*(t)$, $p^*(t)$ and $u^*(t)$ respectively. Then, a necessary condition for $u^*(t)$ to minimize the objective functional in (1) is that:

$$H(x^*(t),u^*(t),p^*(t),t) \le H(x^*(t),u(t),p^*(t),t) \tag{2}$$

for all $t \in [t_0, t_f]$ and for all admissible controls. Equation (2) that indicates that an optimal control must minimize the Hamiltonian is called the PMP (see [3]). Using PMP provides a necessary condition for optimality. This PMP shows that if $x(t)$, $p(t)$ and $u(t)$ are the optimal values of the state, costate and control respectively, they must satisfy the following conditions:

$$\begin{cases} \frac{\partial H(x,u,t,p)}{\partial x} = -\dot{p}(t) \\ \frac{\partial H(x,u,t,p)}{\partial p} = \dot{x}(t) \\ \frac{\partial H(x,u,t,p)}{\partial u(t)} = 0 \end{cases} \tag{3}$$

By replacing the known functions f_0 and g into the Hamiltonian, Eq. (3) gives a system of ODEs, which can be solved via numerical methods or other existing methods. In some cases, Eq. (3) introduces an straightforward ODE system that can be solved easily. But in most cases (specially in practical problems), the system cannot be easily solved, and an approximated scheme must be applied. In the next section, we try to apply neural network's ability in function approximation to solve (3).

A basic neuron based on a perceptron can be observed in Fig. 1. It is proved that we can use multi-layer perceptrons to approximate any nonlinear function with arbitrary accuracy(see [23]).

Here W is the weight vector of input layer, b is a vector containing bias weights, and V is the output layer weights.

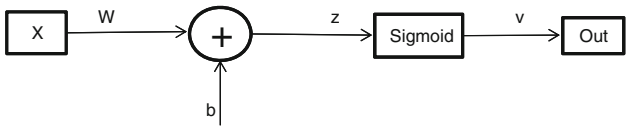

Fig. 1 Basic perceptron

It can be observed that we can calculate the output from the following formulation:

$$\begin{cases} \text{out} = \sum_{i=1}^{k} v_i \sigma(z_i) \\ z_i = \sum_{i=1}^{k} w_i x + b_i \end{cases} \tag{4}$$

where k is the number of sigmoid units. The activation function here is the sigmoid function in the following formula:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{5}$$

Based on Kolmogorov theorem, it is proved that we can implement any continuous function with a multi-layer perceptron (for more details, see [23]). According to this theorem, we use the ability of neural networks in function approximation, to approximate the state, co-state and control function for optimal control problem (1) which will be discussed in detail in the next section.

3 Main idea

In this section, we try to propose an approximation scheme for solving the equations arising in PMP (i.e., Eq. 3). We consider three neural networks for each function: state (its neural network is n_x), costate (its neural network is n_p) and the control (its neural network is n_u) function separately, where each neural network model contains its special adjustable parameters, as it can be observed in Fig. 1. Note that the structures of the neural network models must be constructed such that satisfy the initial or boundary conditions. The proposed neural network models can be proposed in the following forms:

$$\begin{cases} n_x = \sum_{i=1}^{I} v_x^i \sigma(z_x^i), \quad z_x^i = w_x^i t + b_x^i \\ n_p = \sum_{i=1}^{I} v_p^i \sigma(z_p^i), \quad z_p^i = w_p^i t + b_p^i \\ n_u = \sum_{i=1}^{I} v_u^i \sigma(z_u^i), \quad z_u^i = w_u^i t + b_u^i \end{cases} \tag{6}$$

for $i = 1, 2, \ldots, I$ where I is the number of neurons that can be different for each neural network.

Now we are ready to use the neural networks (6) and define the main trail solutions. The trial solutions (for state, costate and control function) contain the neural networks, satisfy the initial or boundary conditions, and thus, they can be defined in the following structures:

$$\begin{cases} x_T = x_0 + (t - t_0) n_x \\ p_T = n_p \\ u_T = n_u \end{cases} \tag{7}$$

It is easy to check that x_T satisfies the initial condition ($x_T(t_0) = x_0$). Note that we may have $p(.) = 0$ for free end-points. For example, if $x(t_0)$ is free, we must have $p(t_0) = 0$, and thus, we can define p_T in (7) as: $p_T = (t - t_0)n_p$. For other initial (or boundary) conditions, we can construct appropriate trial functions.

By replacing the trial solutions into the Hamiltonian function, we can define a trial hamiltonian H_T which is conventional Hamiltonian function H where we replaced the functions x, p and u by their corresponding trial format (x_T, p_T and u_T respectively) as $H_T(x_T(t), u_T(t_f), p_T(t), t) = f_0(x_T(t), u_T(t), t) + p_T(t) \cdot g(x_T(t), u_T(t), t)$. Thus, the trial Hamiltonian function contains the weights of neural networks. Since the trial solutions (7) must satisfy conditions (3), we replace them into the Eq. (3):

$$\begin{cases} \frac{\partial H_T}{\partial x_T} + \dot{p_T} = 0 \\ \frac{\partial H_T}{\partial p_T} - \dot{x_T} = 0 \\ \frac{\partial H_T}{\partial u_T} = 0 \end{cases} \tag{8}$$

To solve the system (8), we define three error functions corresponding to each equation:

$$\begin{cases} E_1(\phi, t) = \left[\frac{\partial H_T}{\partial x_T} + \dot{p_T}\right]^2 \\ E_2(\phi, t) = \left[\frac{\partial H_T}{\partial p_T} - \dot{x_T}\right]^2 \\ E_3(\phi, t) = \left[\frac{\partial H_T}{\partial u_T}\right]^2 \end{cases} \tag{9}$$

and finally a total error function $E(\phi, t) = E_1(\phi, t) + E_2(\phi, t) + E_3(\phi, t)$, where ϕ is a vector containing all weights of three neural networks (6). Indeed, ϕ contains all weights w_x, w_p, w_u, b_x, b_p, b_u, v_x, v_p and v_u. Now instead of solving Eq. (8), we discritize the interval $[t_0, t_f]$ (by m points) and solve the following unconstrained optimization problem:

$$\min_{\phi} \sum_{k=1}^{m} E(t_k, \phi) \tag{10}$$

To solve (10), which is an unconstrained optimization problem, we can use any optimization algorithms such as steepest descent, Newton, or Quasi-Newton methods as well as the heuristic algorithms such as GA (genetic algorithm) or particle swarm optimization, etc.

After terminating the optimization step, we can replace the optimal values of the weights ϕ (containing the weights of input and output layer and the bias vector) into the Eq. (7) and conclude the trial structures of state, co-state and control functions.

The main advantages of this method are that the implementation of the algorithm is not very complicated, we can use more hidden layer or more training points over the interval $[t_0, t_f]$ to obtain more accurate approximations. Finally, the solution of state, co-state and control functions is introduced as functions of time (t) thus we can calculate

the solution at every arbitrary point over the interval $[t_0,t_f]$. Also the proposed control and state functions are differentiable which can be useful in applications.

4 Numerical simulations

In this section, we try to implement the proposed algorithm to solve four optimal control problems. We used for all problems, five parameters for each input, output and bias weights. The intervals are discretized to ten equivalent parts. For the optimization step, we used the MATLAB 7 optimization toolbox with Quasi-Newton BFGS algorithm. The user can use other optimization algorithms such as steepest descent, Newton-based methods or other heuristic algorithms such as GA or particle swarm optimization, etc.

Example 4.1 Consider the following optimization problem:

$$\begin{aligned} \min & \int_0^1 [x^2(t)+u^2(t)]\,\mathrm{d}t \\ \text{s.t} & \\ & \dot{x}=u(t) \\ & x(0)=1, x(1) \text{ is free} \end{aligned} \tag{11}$$

First we must construct the Hamiltonian function:

$$H(x,u,p,t)=x^2(t)+u^2(t)+pu(t) \tag{12}$$

Following Eq. 3, we must have:

$$\begin{cases} 2x(t)=-\dot{p} \\ \dot{x}=u(t) \\ 2u(t)+p=0 \end{cases} \tag{13}$$

because $x(1)$ is free, we have $p(1)=0$. Considering this condition and the initial condition $x(0)=1$, we can choose the trial solutions as:

$$\begin{cases} x_T=1+tn_x \\ p_T=(t-1)n_p \\ u_T=n_u \end{cases} \tag{14}$$

For this example, we used 15 weights for each neural network (five weights for each weight of input layer, output layer and the bios vector). We can see the approximate and exact solutions for $u(t)$ and $x(t)$ in Figs. 2 and 3, respectively. Figures 4 and 5 show the solution accuracy.

Example 4.2 Consider the following optimization problem:

$$\begin{aligned} \min & \int_0^1 [(2-x(t))^2+u^2(t)]\,\mathrm{d}t \\ \text{s.t} & \\ & \dot{x}=-0.25\sqrt{x(t)}+u(t) \\ & x(0)=0, x(1)=2 \end{aligned} \tag{15}$$

First we must construct the Hamiltonian function:

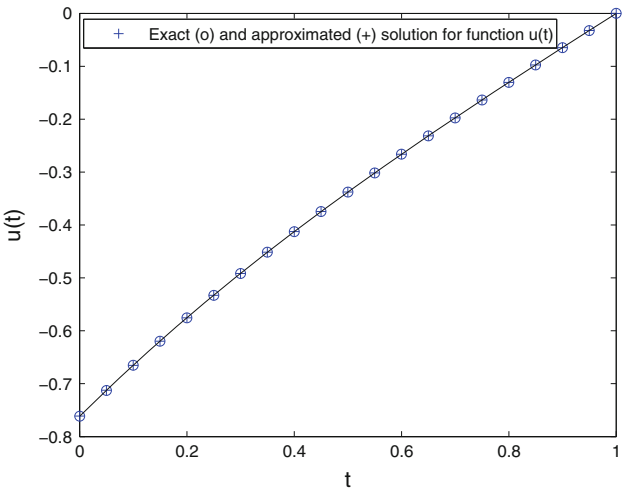

Fig. 2 Exact and approximated control function (Example 4.1)

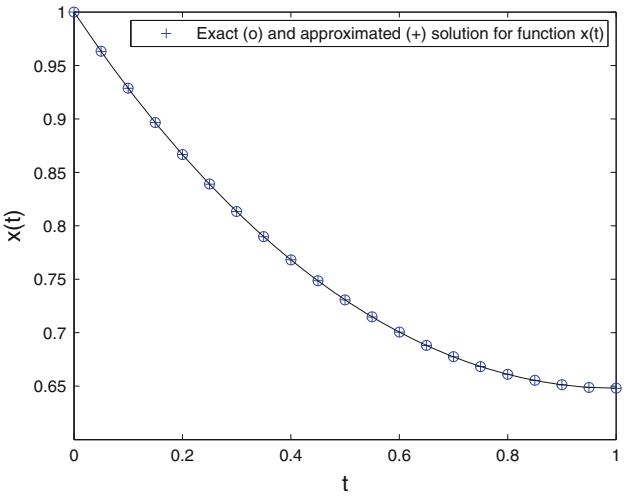

Fig. 3 Exact and approximated state function (Example 4.1)

$$H(x,u,p,t)=[(2-x(t))^2+u^2(t)]+p(-0.25\sqrt{x(t)}+u(t)) \tag{16}$$

Following Eq. 2, we must have:

$$\begin{cases} -2(2-x(t))+\frac{-0.25}{2\sqrt{x(t)}}p=-\dot{p} \\ \dot{x}=-0.25\sqrt{x(t)}+u(t) \\ 2u(t)+p=0 \end{cases} \tag{17}$$

Considering the initial conditions $x(0)=0$ and $x(1)=2$, we can choose the trial solutions as:

$$\begin{cases} x_T=2t+t(t-1)n_x \\ p_T=n_p \\ u_T=n_u \end{cases} \tag{18}$$

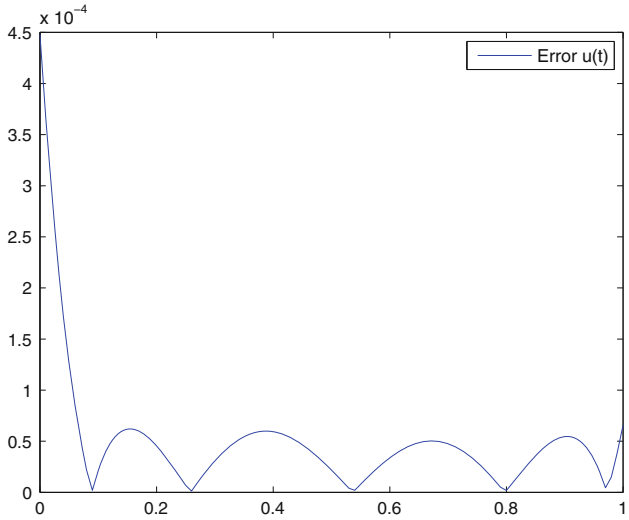

Fig. 4 Error for estimating control function (Example 4.1)

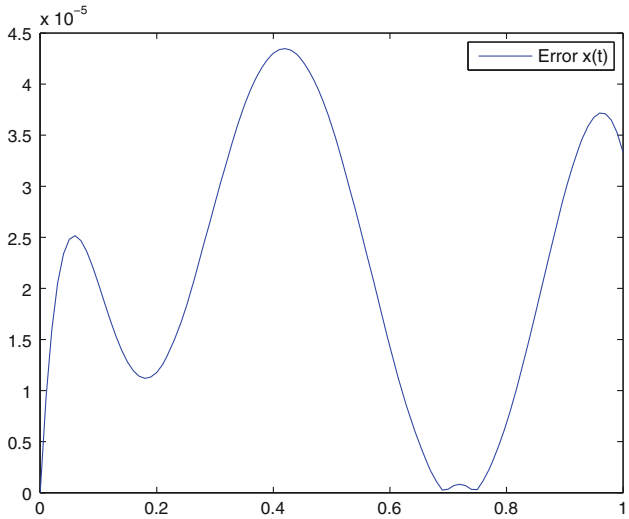

Fig. 5 Error for estimating state function (Example 4.1)

For this example, we used 15 weights for each neural network (five weights for each weight of input layer, output layer and the bios vector). We can see the approximate and exact solutions in Figs. 6 and 7. Figures 8 and 9 show the solution accuracy. This example is solved in [6] by a variational iteration method. Our results are comparable with the results in [6]; however, neural network method gives the state and control as functions of time which are differentiable and the method implementation is more simple.

Example 4.3 Consider the following optimization problem:

$$\begin{aligned} &\min && J = -x(2) \\ &\text{s.t} \\ &&& \dot{x} = \tfrac{5}{2}(-x + ux - u^2) \\ &&& x(0) = 1 \end{aligned} \tag{19}$$

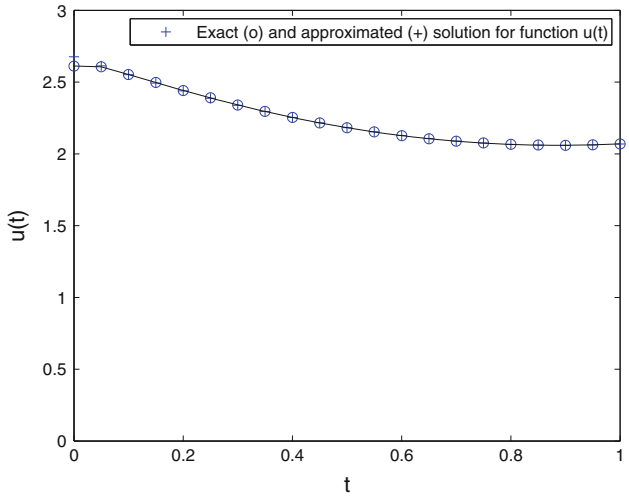

Fig. 6 Exact and approximated control function (Example 4.2)

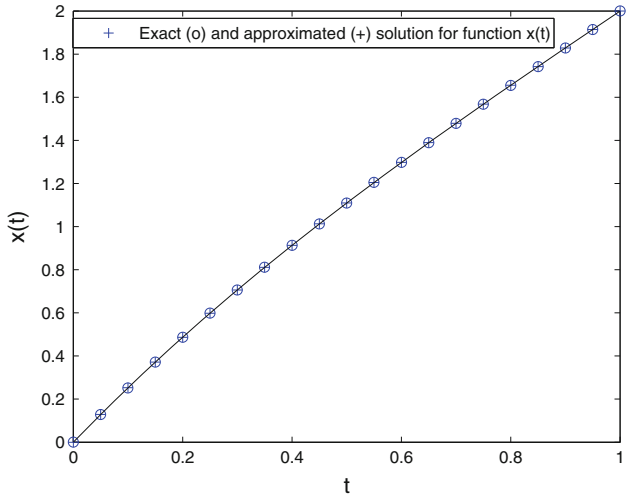

Fig. 7 Exact and approximated state function (Example 4.2)

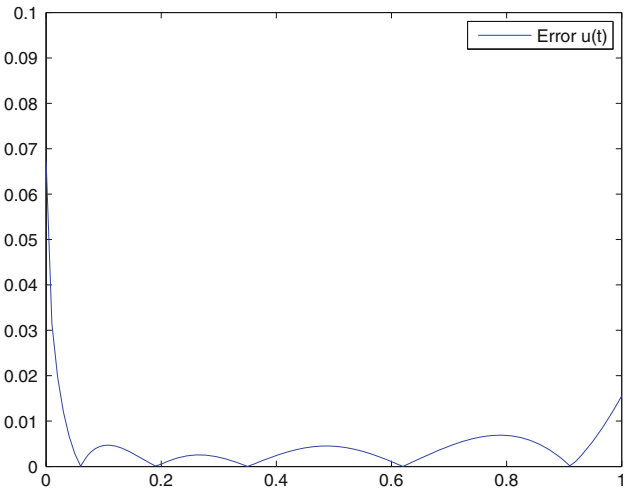

Fig. 8 Error for estimating control function (Example 4.2)

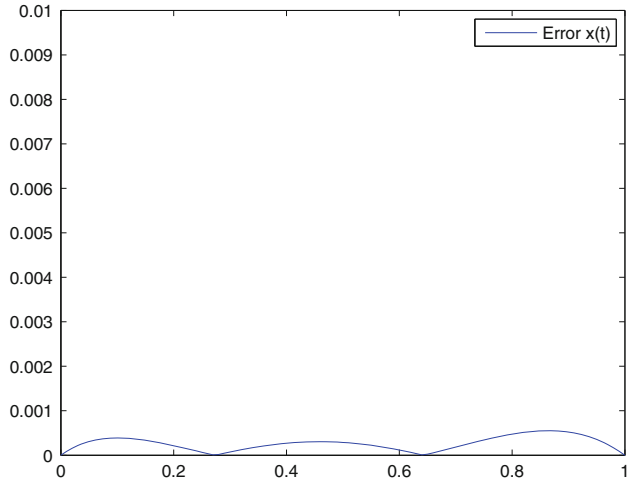

Fig. 9 Error for estimating state function (Example 4.2)

The exact state and control functions are as follows:

$$\begin{cases} x(t) = \frac{4}{1+3\exp\left(\frac{5t}{2}\right)} \\ u(t) = \frac{\dot{x}(t)}{2} \end{cases} \tag{20}$$

Figures 10 and 11 show the state function approximation and corresponding error. Figures 12 and 13 show the control function approximation and its corresponding error.

Example 4.4 Consider the following nonlinear optimal control problem [24, 25]:

$$\begin{aligned} \min & \int_0^1 u(t)^2 \, \mathrm{d}t \\ \text{s.t} & \\ & \dot{x} = 0.5x^2(t)\sin(x(t)) + u(t) \\ & x(0) = 0, \quad x(1) = 0.5, \end{aligned} \tag{21}$$

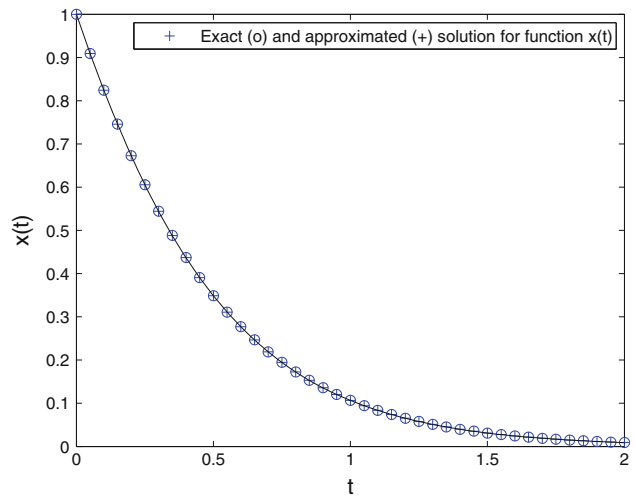

Fig. 10 Exact and approximated state function for Example 4.3

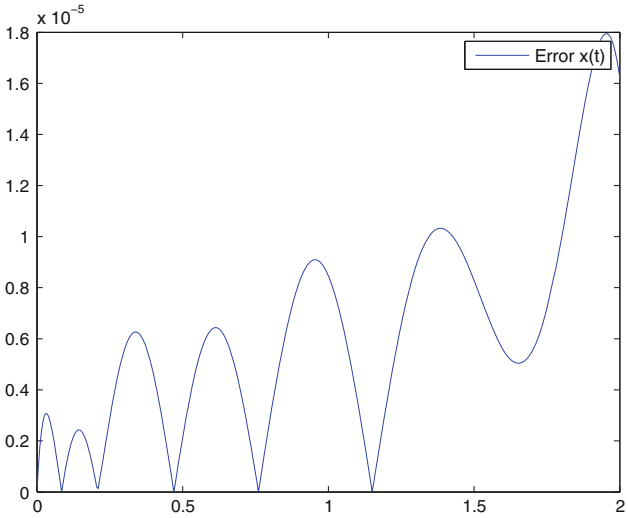

Fig. 11 Error for estimating state function (Example 4.3)

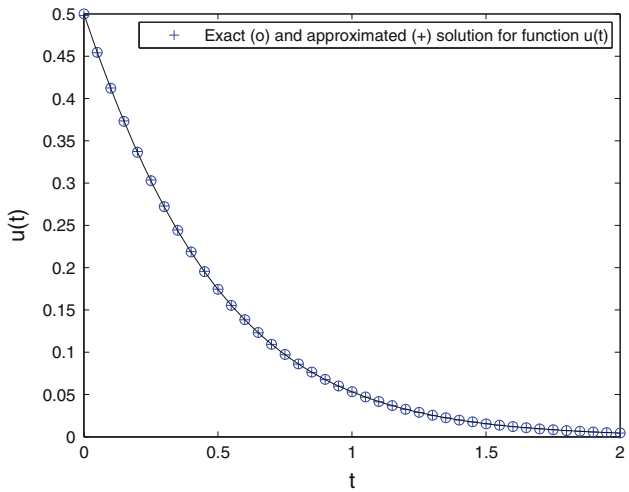

Fig. 12 Exact and approximated control function (Example 4.3)

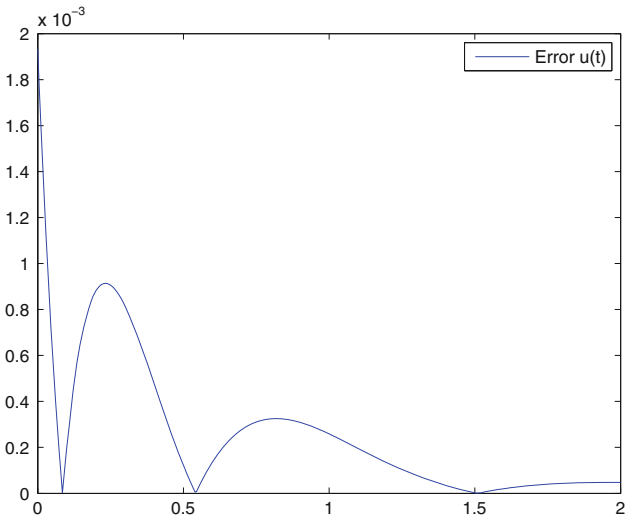

Fig. 13 Error for estimating control function (Example 4.3)

This problem is solved in [25] by a variational method. For this example, we have $H(x(t), u(t), p(t), t) = u(t)^2 + p(0.5x^2(t)sin(x(t)) + u(t))$. Thus conditions (3) can be driven as the following system:

$$\begin{cases} \dot{x}(t) = 0.5x^2(t)\sin(x(t)) - 0.5p(t) \\ \dot{p}(t) = -p(t)x(t)\sin(x(t)) - 0.5p(t)x^2(t)\cos(x(t)) \\ 2u(t) + p(t) = 0, \end{cases} \tag{22}$$

with the initial and final conditions $x(0) = 0$ and $x(1) = 0.5$. This system is solved by numerical methods (Euler method), and the results are displayed and compared with the results obtained by neural networks in Figs. 14 and 15. The optimal value of the objective functional in neural network method is $J^* = 0.2353$ and $x(t_f)$ is exactly equal to 0.5. Comparing our result with the obtained results in [25] shows the accuracy of the method based on neural networks.

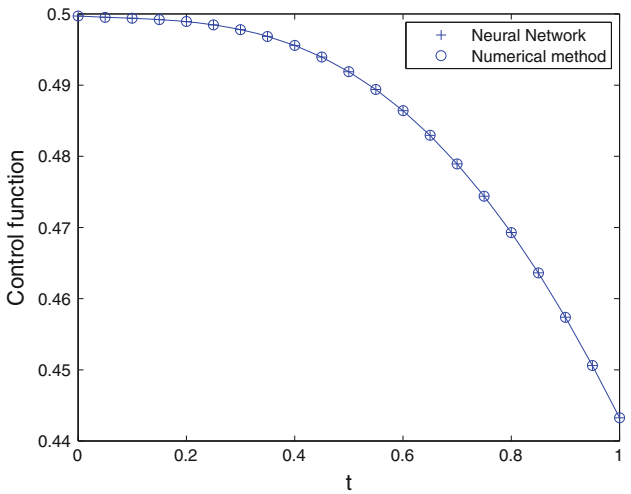

Fig. 14 Control function (Example 4.4)

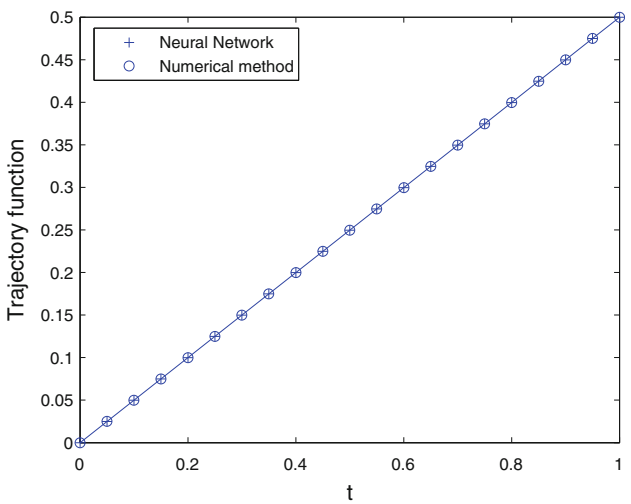

Fig. 15 State function (Example 4.4)

5 Concluding remarks

This paper presented an approximated solution of optimal control problems, based on the neural network approach. One of the advantages of the proposed method is that, for attaining more accurate solutions, we can use more hidden layers, more training points and also we are allowed to use heuristic algorithms in optimization step such as GA and PSO or other existing unconstrained optimization algorithms. The proposed solution is a differentiable function (for state, co-state and control functions). Work is in progress to apply the method to approximate the solution of HJB equation and also for problems arising in calculus of variations.

References

1. Krabs W, Pickl S (2010) An optimal control problem in cancer chemotherapy. Appl Math Comput 217:1117–1124
2. Modares H, Naghibi Sistani MB (2011) Solving nonlinear optimal control problems using a hybrid IPSOSQP algorithm. Eng Appl Artif Intell 24:476–484
3. Kirk DE (2004) Optimal control theory—an introduction. Dover Publications, Mineola, NY
4. Lewis F, Syrmos VL (1995) Optimal control. Wiley, New York
5. Hilscher RS, Zeidan V (2012) Hamilton–Jacobi theory over time scales and applications to linear-quadratic problems. Nonlinear Anal 75:932–950
6. Berkani S, Manseur F, Maidi A (2012) Optimal control based on the variational iteration method. Comput Math Appl 64:604–610
7. Garg D, Patterson M, Hagera WW, Raoa AV, Bensonb DA, Huntington GT (2010) A unified framework for the numerical solution of optimal control problems using pseudo-spectral methods. Automatica 46:1843–1851
8. Clever D, Lang J, Ulbrich S, Ziems JC (2010) Combination of an adaptive multilevel SQP method and a space-time adaptive PDAE solver for optimal control problems. Procedia Comput Sci 1:1435–1443
9. de Oliveira VA, Silva GN, Rojas-Medar MA (2009) KT-invexity in optimal control problems. Nonlinear Anal 71:4790–4797
10. Gerdts M (2008) A non-smooth Newtons method for control-state constrained optimal control problems. Math Comput Simul 79:925–936
11. Buldaev AS (2008) Perturbation methods in optimal control problems. Ecol Model 216:157–159
12. Salama AA (2006) Numerical methods based on extended one-step methods for solving optimal control problems. Appl Math Comput 183:243–250
13. Zhou YY, Yangb XQ, Teob KL (2006) The existence results for optimal control problems governed by a variational inequality. J Math Anal Appl 321:595–608
14. Chryssoverghi I, Coletsos I, Kokkinis B (2006) Discretization methods for optimal control problems with state constraints. J Comput Appl Math 191:1–31
15. England R, Gomez S, Lamourc R (2005) Expressing optimal control problems as differential algebraic equations. Comput Chem Eng 29:1720–1730
16. Rodriguez A (2004) On the local stability of the solution to optimal control problems. J Econ Dyn Control 28:2475–2484

17. Saberi Nik H, Effati S, Shirazian M (2012) An approximate-analytical solution for the Hamilton–Jacobi–Bellman equation via homotopy perturbation method. Appl Math Model 36(11):5614–5623
18. Cheng T, Sun H, Qu Z, Lewis FL (2009) Neural network solution for suboptimal control of non-holonomic chained form system. Trans Inst Measurement Control 31(6):475–494
19. Cheng T, Lewis FL (2007) Neural network solution for finite-horizon H-infinity constrained optimal control of nonlinear systems. J Control Theory Appl 5(1):1–11
20. Cheng T, Lewis FL, Abu-Khalaf M (2007) Fixed-final-time-constrained optimal control of nonlinear systems using neural network HJB approach. IEEE Trans Neural Netw 18(6):1725–1737
21. Vrabie D, Lewis FL (2009) Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems. Neural Netw 22:237–246
22. Lagaris IE, Likas A (2012) Hamilton–Jacobi theory over time scales and applications to linear-quadratic problems. IEEE Trans Neural Netw 9(5):987–1000
23. Keckman V (2001) Learning and soft computing. MIT press, Cambridge, MA
24. Rubio JE (1986) Control and optimization, the linear treatment of nonlinear problems. Manchester University Press, Manchester
25. Shirazian M, Effati S (2012) Solving a class of nonlinear optimal control problems via he's variational iteration method. Int J Control Automat Syst 10(2):249–256