Neurocomputing 129 (2014) 528–539

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Fixed-final-time optimal tracking control of input-affine nonlinear systems

Ali Heydari [a,*], S.N. Balakrishnan [b]

[a] Mechanical Engineering Dept. South Dakota School of Mines and Technology, Rapid City, SD 57701, USA
[b] Mechanical and Aerospace Engineering Dept. Missouri University of Science & Technology, Rolla, MO 65409, USA

ARTICLE INFO

Article history:
Received 1 October 2012
Received in revised form
17 July 2013
Accepted 6 September 2013
Communicated by Bin He
Available online 23 October 2013

Keywords:
Fixed-final-time optimal control
Optimal tracking
Approximate dynamic programming
Adaptive critics

ABSTRACT

In this study, approximate dynamics programming framework is utilized for solving the Bellman equation related to the fixed-final-time optimal tracking problem of input-affine nonlinear systems. Convergence of the weights of the neurocontroller in the proposed successive approximation based algorithms is provided and the network is trained to provide the optimal solution to the problems with (a) *unspecified initial conditions* (b) *different time horizons,* and (c) *different reference trajectories* under certain general conditions. Numerical simulations illustrate the versatility of the proposed neurocontroller.

1. Introduction

Approximate dynamics programming (ADP) has shown a lot of promise in solving optimal control problems with neural networks (NN) as the enabling structure [1–10]. Mechanism for ADP is usually provided through a dual network architecture called the Adaptive Critics (AC) [3,2]. In the heuristic dynamic programming (HDP) class with ACs, one network, called the 'critic' network, maps the input states to output the cost and another network, called the 'action' network, outputs the control with states of the system as its inputs [4,5]. In the dual heuristic programming (DHP) formulation, while the action network remains the same as the HDP, the critic network outputs the costates with the current states as inputs [2,6,7]. The convergence proof of DHP for linear systems is presented in [8] and that of HDP for general case is presented in [4]. The Single Network Adaptive Critics (SNAC) architecture developed in [9] is shown to be able to eliminate the need for the second network and perform DHP using only one network. This process results in a considerable decrease in the offline training effort and the resulting simplicity makes it attractive for online implementation requiring less computational resources and storage memory. Similarly, the cost function based SNAC (J-SNAC) eliminates the need for the action network in an HDP scheme [10]. While [2–10] deal with discrete-time systems, some researchers have recently focused on continuous time problems [11–13].

Note that these developments in the neural network (NN) literature have mainly addressed only the *infinite horizon* or regulator type problems. Finite-horizon optimal control is relatively more difficult due to the time varying Hamilton–Jacobi–Bellman (HJB) equation resulting in a *time-to-go dependent* optimal cost function and costates. Using numerical methods [14] a two-point boundary value problem (TPBVP) needs to be solved for each set of initial condition for a given final time and it will provide only an open loop solution. The control loop can be closed using techniques like Model Predictive Control (MPC) as done in [15], however, the result will be valid only for one set of initial conditions and final time. This limitation holds for the method developed in [16] also. Ref. [17] develops a dynamics optimization scheme which gives an open-loop solution, then, optimal tracking is used for rejecting the online perturbation and deviations from the optimal trajectory. Authors of [18] used power series to solve the problem with small nonlinearities, and in [19] an approximated solution is given through the so-called Finite-horizon State Dependent Riccati Equation (Finite-SDRE) method.

Neural networks are used for solving finite-horizon optimal control problem in [20–25]. Authors of [20] developed a neurocontroller for a scalar problem with terminal constraint using AC. Continuous-time problems are considered in [21] and [22] where the time-dependent weights are calculated through a backward integration. The finite-horizon problem with *unspecified terminal*

* Corresponding author. Tel.: +1 605 3942200.
E-mail addresses: ali.heydari@sdsmt.edu (A. Heydari), bala@mst.edu (S.N. Balakrishnan).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.09.006

time and a fixed terminal state is considered in [23,24]. The neurocontroller developed in [23] can work only with one set of initial conditions and if the initial state is changed, the network needs to be re-trained to give the optimal solution for the new state. This limitation holds for [24] as well.

In many practical systems one is interested in tracking a desired signal. Examples of such systems are contour tracking in machining processes [34–35] and control of robotic manipulators [36]. In some systems the tracking is required to be carried out in a given time, see [37] for an example of such a case in an autopilot design. The constraint of final time being fixed makes the problem very difficult to solve. Missile guidance problems and launch vehicle problems are some other applications in this class of problems. Solving optimal tracking problems for nonlinear systems using adaptive critics has been investigated by researchers in [26–32]. In [26] the authors have developed a tracking controller for the system whose input gain matrix is invertible. In [27] the reference signal is limited to those which satisfy the dynamics of the system. Developments in [28–31] solve the tracking problem for the systems of nonlinear Brunovsky canonical form. Finally, the finite-horizon tracking neurocontroller developed in [32] can control only one set of initial conditions and requires the input gain matrix of the dynamics to be invertible.

In this paper, a single neural network based solution, called Finite-horizon Single Network Adaptive Critics (Finite-SNAC), is developed which embeds an approximate solution to the discrete-time HJB equation for fixed-final-time optimal tracking problems. The approximation can be made as accurate as desired using rich enough basis functions. Consequently, the offline trained network can be used to generate *online feedback control in real-time*. The neurocontroller is able to solve optimal tracking problem of *general* nonlinear control-affine dynamics for tracking either a given arbitrary trajectory or a family of trajectories which share the same, possibly nonlinear, dynamics. Once the network is trained, it will give optimal solution for *every initial condition* as long as the resulting trajectory lies on the domain for which the network is trained, hence, Finite-SNAC does not have the restrictions of some of the cited references in the field. Furthermore, a major advantage of the proposed technique is that this network provides optimal feedback solutions to any *different final time* as long as it is less than the final time for which the network is synthesized. An innovative proof is developed which shows the successive approximation based training algorithm is a contraction mapping [33].

Comparing the developed controller in this paper with the available intelligent controllers in the literature, the closest ones are [20,25]. As compared to [20], in this study, only one network is needed for computing the control, and this idea has been generalized to *tracking* with free final state. Moreover, convergence proofs are provided. The differences between this study and [25] are (a) solving *tracking* problem versus the problem of brining the states to zero in [25] (b) using time varying weights for the neural networks as opposed to the time invariant weights in that reference, (c) developing a 'backward in time' training algorithm versus the pure 'iterative' algorithm in [25], and (d) providing a completely different convergence proof in this study. The advantages of the development in this study versus the available finite-horizon optimal tracking methods in the literature [32,37] are providing solutions for different initial conditions and different final-times, without needing to retrain the network as in [32], or needing to recalculate the series of differential Riccati equation till they converge as in [37]. Moreover, the restriction of requiring an invertible input-gain matrix in [32] does not exist here. Finally the advantage of this study versus the MPC approach utilized in [15] for optimal tracking is having a negligible computational load in here versus the huge real-time computational load in MPC for online numerical calculation of the optimal solution at each instant, as detailed in [15]. In here, however, once the networks are trained offline, the online calculation of the control is as simple as a feeding the states to the network to get the costate vector and hence, the control.

The rest of the paper is organized as follows: Finite-SNAC is developed in section II. Relevant convergence theorems are presented in Section 3. A modified version of the controller for higher versatility is proposed in Section 4, and the numerical results and analyses are presented in Section 5. Finally, the conclusions are given in Section 6.

2. Theory of Finite-SNAC

Consider the nonlinear continuous-time input-affine system

$$\dot{x}(t) = f_c(x(t)) + g_c(x(t))u(t), \tag{1}$$

where $x(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^l$ denote the state and the control vectors at time t, respectively, and parameters n and l are the dimension of the state and control vectors. Smooth functions $f_c : \mathbb{R}^n \to \mathbb{R}^n$ and $g_c : \mathbb{R}^n \to \mathbb{R}^{n\times l}$ are the system dynamics and the initial states are given by $x(0)$. Given reference signal $r(t) \in \mathbb{R}^n$ for $t \in [0\ t_f]$, where the initial time is selected as zero and the final time is denoted by t_f, the objective is selecting a control history $u(t)$, $t \in [0\ t_f]$, such that the cost function given below is minimized.

$$J = \frac{1}{2}(x(t_f) - r(t_f))^T S(x(t_f) - r(t_f)) + \frac{1}{2}\int_0^{t_f} ((x(t) - r(t))^T Q_c(x(t) - r(t)) + u(t)^T R_c u(t))dt.$$

Symmetric matrices $S \in \mathbb{R}^{n\times n}$, $Q_c \in \mathbb{R}^{n\times n}$ and $R_c \in \mathbb{R}^{l\times l}$ are the penalizing matrices for the final states, states, and control vectors, respectively. Matrices S and Q_c are positive semi-definite and matrix R_c is a positive definite matrix. Superscript T denotes transpose operation.

In many practical applications, discretization is used since the states are estimated and the control is calculated at discrete times and not calculated continuously, though the description of dynamics is continuous. The approach selected in this paper is discretizing the problem using a small sampling time Δt to have.

$$x_{k+1} = f(x_k) + g(x_k)u_k,\ k \in \{0, 1, \ldots, N-1\}, \tag{2}$$

$$J = \frac{1}{2}(x_N - r_N)^T S(x_N - r_N) + \frac{1}{2}\sum_{k=0}^{N-1}((x_k - r_k)^T Q(x_k - r_k) + u_k^T R u_k), \tag{3}$$

where integer k denotes the time index, $N \equiv t_f/\Delta t$, $x_k \equiv x(k\Delta t)$, $u_k \equiv u(k\Delta t)$, and $r_k \equiv r(k\Delta t)$. Using Euler integration scheme one has $f(x_k) \equiv x_k + \Delta t f_c(x_k)$, $g(x_k) \equiv \Delta t g_c(x_k)$, $Q \equiv \Delta t Q_c$, and $R \equiv \Delta t R_c$.

Remark 1:. The assumption that discrete-time system (2) is obtained through 'discretizing' a continuous dynamics is utilized in convergence analysis of the developed algorithms in this paper. Excluding systems with inherent discrete evolution, including hybrid systems, all the physical systems are continuous; therefore, this assumption does not impose a limitation on the results obtained here for such systems.

The cost-to-go at each time step k and state vector x_k, denoted by $J(x_k, k)$, is equal to

$$J(x_k, k) = \frac{1}{2}(x_N - r_N)^T S(x_N - r_N) + \frac{1}{2}\sum_{\bar{k}=k}^{N-1}((x_{\bar{k}} - r_{\bar{k}})^T Q(x_{\bar{k}} - r_{\bar{k}}) + u_{\bar{k}}^T R u_{\bar{k}}),$$

which leads to the recursive equation

$$J(x_k, k) = \tfrac{1}{2}((x_k - r_k)^T Q(x_k - r_k) + u_k^T R u_k) + J(x_{k+1}, k+1),\ k \in \{0, 1, \ldots, N-1\}. \tag{4}$$