Control Engineering Practice 73 (2018) 13–25

Contents lists available at ScienceDirect

Control Engineering Practice

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

Incremental model based online dual heuristic programming for nonlinear adaptive control

Ye Zhou *, Erik-Jan van Kampen, Qi Ping Chu

Delft University of Technology, Kluyverweg 1, 2629HS Delft, The Netherlands

ARTICLE INFO

Keywords:
Reinforcement learning
Online learning
Dual heuristic programming
Adaptive control
Nonlinear control

ABSTRACT

Dual heuristic programming has gained an increasing interest in recent years because it provides an effective process for optimal adaptive control of uncertain nonlinear systems. However, it requires an off-line stage to train a global system model from a representative model, which is often infeasible to obtain in practice. This paper presents a new and efficient approach for online self-learning control based on dual heuristic programming. This method uses a recursive least square method to online identify an incremental model of the system instead of a global system model. The presented incremental model based dual heuristic programming method can adaptively generate a near-optimal controller online without a priori information of the system dynamics or an off-line training stage. To compare the online adaptability of the conventional dual heuristic programming method and the newly proposed method, two numerical experiments are performed: an online reference tracking task and a fault-tolerant control task. The results reveal that the proposed method outperforms the conventional dual heuristic programming method in online learning capacity, efficiency, accuracy, and robustness.

1. Introduction

Adaptive control strategies are the foundation for controlling nonlinear systems with uncertainties. To solve these control problems, Reinforcement Learning (RL) offers an option without using accurate system models (Khan, Herrmann, Lewis, Pipe, & Melhuish, 2012). RL is a self-learning method, in which actions are trained in order to minimize the cost-to-go from interaction with the environment. These self-learning methods link bio-inspired artificial intelligence techniques to the field of optimal control and adaptive control to overcome some of the limitations and challenges of traditional model-based control methods in practical applications. Approximate Dynamic Programming (ADP) is an RL method aiming to solve optimal control problems with large or continuous state spaces (Enns & Si, 2003; Ferrari & Stengel, 2004; Hanselmann, Noakes, & Zaknich, 2007; Si, 2004; Wang, Zhang, & Liu, 2009; Yadav, Padhi, & Balakrishnan, 2007). They apply an approximation of the cost-to-go of states and/or an approximation towards the optimal control policy so as to tackle the 'curse of dimensionality'. Therefore, these methods belong to optimal adaptive control (Khan et al., 2012), and the trained policy is near-optimal.

As a class of ADP methods, Adaptive Critic Designs (ACDs) have shown great success in optimal adaptive control of nonlinear problems and practical applications (Ferrari & Stengel, 2004; Khan et al., 2012; Prokhorov & Wunsch, 1997; Si, 2004; Wang, He, & Liu, 2017). They are also known as Actor-Critics (ACs) because they separate evaluation (critic) and improvement (actor) using parametric structures. The critic adopts Temporal Difference (TD) methods to approximate the cost-to-go function, while the actor adapts its parameters towards the optimal policy by applying the principle of optimality (Khan et al., 2012; Sutton & Barto, 1998). Although they are called ACs, they often need an extra structure to approximate the global system model so as to close the update path of the actor, the critic, or both. The critic, actor, and system model can be implemented with nonlinear function approximators, such as Artificial Neural Networks (ANN). With these approximators, ACDs can identify the system dynamics globally and then adaptively generate the control laws.

ACDs can generally be categorized into three groups: (1) Heuristic Dynamic Programming (HDP), which is the most basic form and uses the critic to approximate the cost-to-go; (2) Dual Heuristic Programming (DHP), in which the critic approximates the derivatives of the cost-to-go with respect to the critic inputs; and (3) Globalized Dual Heuristic Programming (GDHP), which approximates both the cost-to-go and its derivatives. Several studies comparing the before-mentioned ACDs have shown that both DHP and GDHP outperform HDP in success rate and precision (Prokhorov & Wunsch, 1997; Venayagamoorthy, Harley, &

* Corresponding author.
E-mail addresses: Y.Zhou-6@tudelft.nl (Y. Zhou), E.vanKampen@tudelft.nl (E. van Kampen), Q.P.Chu@tudelft.nl (Q.P. Chu).

https://doi.org/10.1016/j.conengprac.2017.12.011
Received 1 July 2017; Received in revised form 27 November 2017; Accepted 30 December 2017

Wunsch, 2002). The main reason is that the critic of the DHP and the GDHP directly outputs the derivatives of the cost-to-go, which reduces the error introduced by the derivation backward through the critic of the HDP (Si & Wang, 2001). Although the GDHP did not show distinct advantages over the DHP, the computational complexity is considerably higher due to the second derivative terms (Prokhorov & Wunsch, 1997; Si & Wang, 2001). Therefore, the proposed method in this paper is mainly related to the DHP.

In addition, Action Dependent (AD) variations of these three original versions have been developed by directly connecting the output of the actor to the input of the critic (Enns & Si, 2003; Ni, He, Zhong, & Prokhorov, 2015; Prokhorov & Wunsch, 1997; van Kampen, Chu, & Mulder, 2006). The AD forms may reduce the dependency on the system model. However, from the theoretical point of view, the actor output is not necessarily (usually not) an input to the critic, which estimates the value/cost function; from the practical perspective, the extra input will increase the complexity of the critic. Furthermore, previous studies comparing HDP and its AD form have reported that HDP controllers have a higher success rate in an auto-landing task (Prokhorov & Wunsch, 1997), besides which it can operate in a wider range of flight conditions and adapts faster to the changed plant dynamics in controlling an F-16 aircraft model (van Kampen et al., 2006).

Online learning control with ACDs has been studied for years and is still one of the most active areas in RL today. Conventional ACDs often have two learning phases (Enns & Si, 2003; Ferrari & Stengel, 2004; Prokhorov & Wunsch, 1997; van Kampen et al., 2006; Wang, Liu, Wei, Zhao, & Jin, 2012): off-line learning and online learning. The main reason is that the identification of the global system model is not a trivial task, which needs certain time and usually an off-line learning phase beforehand (Farrell, Sharma, & Polycarpou, 2005; Lombaerts, Oort, Chu, Mulder, & Joosten, 2010; Sghairi, De Bonneval, Crouzet, Aubert, & Brot, 2008; Sonneveldt, Van Oort, Chu, & Mulder, 2008, 2009; Tang, Roemer, Ge, Crassidis, Prasad, & Belcastro, 2009; Van Oort, Sonneveldt, Chu, & Mulder, 2010). However, this off-line identification stage needs representative simulation models, which are also difficult to obtain in practice. During the online phase, extra computing cost is required to adaptively perform the approximation of the system with unforeseen dynamics, such as the resulting changes from the changes in the actor, a time-varying component in the system, uncertainties in the environment, and unexpected changes due to failures. Several studies (Ni et al., 2015; Si & Wang, 2001) have suggested to remove the global system model and to exploit previous critic outputs and/or inputs instead. Although this technique has been successfully applied to many ACD methods, it can only relieve the off-line learning phase of the AD forms. An accurate global system model still plays an important role in most ACDs, especially in DHP and GDHP because the update of both the critic and the actor depends on the system model.

This paper aims at increasing the feasibility of ACDs to practical applications without a priori information. A systematic approach is proposed for developing online ACD controllers, more specifically for DHP, based on the incremental control technique. This incremental technique has been successfully applied to design adaptive controllers, such as Incremental Nonlinear Dynamic Inversion (INDI) (Sieberling, Chu, & Mulder, 2010; Simplício, Pavel, van Kampen, & Chu, 2013), Incremental BackStepping (IBS) (Acquatella, van Kampen, & Chu, 2013) and incremental adaptive sliding mode control (Putro & Holzapfel, 2016), to deal with system nonlinearities. However, these methods have not addressed optimization or synthesis of designed closed-loop systems. Incremental Approximate Dynamic Programming (iADP) (Zhou, van Kampen, & Chu, 2015, 2017) was proposed for off-line near-optimal control of unknown nonlinear systems without using system models. This approach uses a quadratic function to approximate the cost function. Therefore, it is suitable for many practical control problems with approximately convex cost functions.

In this paper, Incremental model based Dual Heuristic Programming (IDHP) is developed for online adaptive control of unknown nonlinear systems. It uses a linear time-varying approximation of the original system to replace the global system model in conventional DHP. In addition, a Recursive Least Square (RLS) technique is used to identify the incremental model when assuming a sufficiently high sample rate for discretization. This method belongs to model-free control because it does not need any a priori information of the system dynamics at the beginning nor online identification of the global nonlinear system, but only the online identified incremental model.

The remainder of this paper is structured as follows. Section 2 starts with a brief introduction of the conventional DHP algorithm and then focuses on the development of the IDHP method. Section 3 introduces the nonlinear air vehicle model and discusses some related issues to achieve the implementation of the DHP and IDHP methods. Then, Section 4 applies these two algorithms to two illustrative control tasks and compares their performance with regard to success rates, tracking errors, settling time, and robustness in different initial states and failures. Lastly, Section 5 concludes the advantages and disadvantages of using the incremental approach with DHP and addresses the challenges and possibilities of the future research.

2. Incremental model based dual heuristic programming design

This section develops an online adaptive controller for unknown nonlinear systems, namely Incremental model based Dual Heuristic Programming (IDHP). The major difference with the conventional DHP is that IDHP does not use a nonlinear function approximator to approach the global system model. Instead, it exploits an online identified incremental model. Therefore, this method can adapt the controller online without a priori knowledge of the system dynamics or off-line learning of the system model. The rest of this section will briefly introduce the conventional DHP and then focus on the IDHP algorithm and adaptation rules.

2.1. DHP framework and global system model

DHP methods are most favored within the ACD category because they have higher success rate and accuracy than the HDP and lower computational complexity than the GDHP (Prokhorov & Wunsch, 1997; Si & Wang, 2001; Venayagamoorthy et al., 2002). Conventional DHP controllers use three nonlinear function approximators to approach the actor, the critic, and the system dynamical model with weights (or more generally called model parameters) $\mathbf{w}_a$, $\mathbf{w}_c$, and $\mathbf{w}_m$, respectively, as shown in Fig. 1. The Back-Propagation (BP) algorithms of both the critic and the actor are based on the system model.

The DHP uses the system model to approximate the dynamics of the global system. The inputs of the system model are the current state, $\mathbf{x}_t \in \mathcal{R}^n$, and the control input, $\mathbf{u}_t \in \mathcal{R}^m$, based on which it outputs the estimated next state, $\hat{\mathbf{x}}_{t+1} \in \mathcal{R}^n$. The system model weights $\mathbf{w}_m(t)$ are updated by minimizing the model error. The error is defined as the difference between the measured state $\mathbf{x}_t$ and the estimated state $\hat{\mathbf{x}}_t$:

$$E_m(t) = \frac{1}{2}\mathbf{e}_m(t)^T\mathbf{e}_m(t), \tag{1}$$

where

$$\mathbf{e}_m(t) = \mathbf{x}_t - \hat{\mathbf{x}}_t. \tag{2}$$

The update rule for system model weights are formulated according to the gradient-descent algorithm with a learning rate η_m:

$$\mathbf{w}_m(t+1) = \mathbf{w}_m(t) + \Delta\mathbf{w}_m(t), \tag{3}$$

$$\Delta\mathbf{w}_m(t) = -\eta_m \cdot \frac{\partial E_m(t)}{\partial \hat{\mathbf{x}}_t}\frac{\partial \hat{\mathbf{x}}_t}{\partial \mathbf{w}_m(t)}. \tag{4}$$

Nonlinear function approximators, such as artificial neural networks, can identify the system dynamics globally. However, online identification of the global model is not a trivial task. It needs a certain time