Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Process Control 16 (2006) 135–156
JOURNAL OF PROCESS CONTROL
www.elsevier.com/locate/jprocont

# Choice of approximator and design of penalty function for an approximate dynamic programming based control approach

Jong Min Lee, Niket S. Kaisare, Jay H. Lee *

*School of Chemical and Biomolecular Engineering, Georgia Institute of Technology, 311 Ferst Dr. NW, Atlanta, GA 30332–0100, USA*

Received 7 September 2004; received in revised form 17 February 2005; accepted 27 April 2005

**Abstract**

This paper investigates the choice of function approximator for an approximate dynamic programming (ADP) based control strategy. The ADP strategy allows the user to derive an improved control policy given a simulation model and some starting control policy (or alternatively, closed-loop identification data), while circumventing the 'curse-of-dimensionality' of the traditional dynamic programming approach. In ADP, one fits a function approximator to state vs. 'cost-to-go' data and solves the Bellman equation with the approximator in an iterative manner. A proper choice and design of function approximator is critical for convergence of the iteration and the quality of final learned control policy, because an approximation error can grow quickly in the loop of optimization and function approximation. Typical classes of approximators used in related approaches are parameterized global approximators (e.g. artificial neural networks) and nonparametric local averagers (e.g. *k*-nearest neighbor). In this paper, we assert on the basis of some case studies and a theoretical result that a certain type of local averagers should be preferred over global approximators as the former ensures monotonic convergence of the iteration. However, a converged cost-to-go function does not necessarily lead to a stable control policy on-line due to the problem of over-extrapolation. To cope with this difficulty, we propose that a penalty term be included in the objective function in each minimization to discourage the optimizer from finding a solution in the regions of state space where the local data density is inadequately low. A nonparametric density estimator, which can be naturally combined with a local averager, is employed for this purpose.


*Keywords:* Approximate dynamic programming; *k*-nearest neighbor; Neural network

## 1. Introduction

In dynamic programming (DP), one computes an optimal policy *off-line* by solving the 'Bellman equation' [1]. The objective of DP is to obtain the optimal 'cost-to-go' function, which is a map between a starting state and minimum achievable total cost. The optimal cost-to-go information allows us to calculate an optimal control decision for any given system state without having to consider the system's dynamic behavior in a long future time horizon. Hence it can potentially reduce the large on-line computational burden associated with the math programming based approaches like model predictive control (MPC). In addition, unlike MPC, DP can be used to derive an *optimal feedback* policy for an uncertain system. However, the approach leads to an exponential growth in the computation with respect to the state dimension, and the computational and storage requirements for all problems of practical interest remain unwieldy even with today's computing hardware.

Whereas the control community was quick to discard the DP approach as a viable approach, researchers in the fields of artificial intelligence (AI) and machine

* Corresponding author. Tel.: +1 404 385 2148; fax: +1 404 894 2866.
*E-mail address:* jay.lee@chbe.gatech.edu (J.H. Lee).

0959-1524/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jprocont.2005.04.010

learning (ML) have explored the possibility of applying the theories of psychology and animal learning to solve DP problems [2]. These efforts were made under the umbrella of reinforcement learning (RL) [3] and neuro-dynamic programming (NDP) [4]. Though RL and NDP are established techniques for robotics and operations research problems, it is difficult to apply them directly to process control problems for several reasons. Most RL algorithms improve the cost-to-go function on-line by trial and error. This is potentially costly and risky for process control because one cannot, for example, operate a chemical reactor in a random manner just to explore the state space and learn. Though NDP algorithms are based on off-line update of cost-to-go function, their main premise is that a very large set of data densely populating all relevant parts of the state space for control are available. A typical chemical process, however, has a huge state space and it is unrealistic to attempt to populate all regions of the state space that may potentially come into play during operation. Furthermore, both the RL and NDP approaches assume that a failure in learning merely means a need for new data through additional exploration rather than a catastrophic situation. Finally, their common update rules based on a 'temporal difference' term are suited for "discrete" state space rather than a continuous space common for process control problems [4].

Inspired by the work done in the RL/NDP community, we have recently introduced an approximate dynamic programming (ADP) strategy suited for process control problems [5,6]. Our approach is to solve the Bellman equation for those state points sampled from *closed-loop simulations or experiments* and to obtain improved control policies through 'value-' or 'policy iteration'. Since typical process control problems involve a continuous state space, a function approximator is used to estimate a cost-to-go value for any given state. In our previous work, we have shown the efficacy of the approach through applications to several nonlinear process control problems [6,7,5,8].

One important lesson learned from our experience with the approach is that stability of learning and quality of a learned control policy are critically dependent on the structure of the function approximator. In this paper, the stability for off-line learning means that the infinity norm of the difference between cost-to-go values of current and previous rounds of iteration continues to converge within any specified tolerance value. The typical approach in the NDP and RL literature is to fit a global approximator like a neural network to cost-to-go data. While this approach has seen some notable successes in certain applications (e.g. a backgammon player at a world champion level [9]), it has also met with failures in many other applications, for example, due to the problems of local convergence and overfitting [10,11]. In certain instances, the off-line iteration would even fail to converge, with the cost-to-go approximation showing extreme nonmonotonic behavior or instability with the iteration.

The failure with a general function approximator was first explained by Thrun and Schwartz [12] with what they called an "overestimation" effect. They assumed uniformly distributed and independent error in the approximation and derived bounds on the necessary accuracy of the function approximator and discount factor. Sabes [13] showed that bias in optimality can be large when a global approximator with a linear combination of basis functions is employed. Boyan and Moore [14] listed several simple simulation examples where popular approximators fail miserably in off-line learning. Sutton [15] modified the experimental setup for the same examples and adopted a model-free on-line learning scheme to make them work. In summary, experiments with global function approximation schemes have produced mixed results, which are consistent with our experience.

Gordon [16] presented a 'stable' cost-to-go learning scheme with off-line iteration for a fixed set of states. A class of function approximators with a 'nonexpansion' property (e.g. $k$-nearest neighbor) was shown to guarantee off-line convergence of cost-to-go values during the value iteration. Gordon also provided a result for the accuracy of converged cost-to-go values in the case of 1-nearest neighbor, which has a fixed point property in the approximation. Tsitsiklis and Van Roy [17] provided a proof of convergence and its accuracy for linear function approximators when applied to finite MDPs under temporal difference learning with a particular on-line state sampling scheme. They commented that convergence properties with general nonlinear function approximators (e.g. neural network) remained unclear.

For systems with a continuous state space (infinite MDP), there are no proofs for convergence in the learning of cost-to-go function with *general* function approximators. For linear quadratic regulation problems, there exist proofs of convergence and its accuracy bounds for continuous state-action space problems with specific approximator structures [18]. Recently, a kernel-based local averaging structure was shown to have a property of convergence to optimal cost-to-go with increasing number of samples and decreasing kernel bandwidth under a certain model-free learning scheme [19,20].

In this paper we show that, even for problems involving continuous state and action spaces, the nonexpansion property guarantees stable learning in the off-line value iteration step of our ADP strategy. We back up this assertion with a theoretical result and several case studies. It still remains to be ascertained that the converged cost-to-go indeed leads to optimal or at least improved closed-loop performance. In this regard, we