Automatica 92 (2018) 100–108

Contents lists available at ScienceDirect

Automatica

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

Brief paper

Continuous-action planning for discounted infinite-horizon nonlinear optimal control with Lipschitz values☆

Lucian Buşoniu [a,*], Előd Páll [b], Rémi Munos [c]

[a] Automation Department, Technical University of Cluj-Napoca, Romania
[b] Robotics and Biology Lab, Technische Universität Berlin, Germany
[c] Google DeepMind, London, UK

ARTICLE INFO

Article history:
Received 4 August 2016
Received in revised form 21 December 2017
Accepted 11 January 2018

Keywords:
Optimal control
Planning
Nonlinear systems
Near-optimality analysis

ABSTRACT

We consider discrete-time, infinite-horizon optimal control problems with discounted rewards. The value function must be Lipschitz continuous over action (input) sequences, the actions are in a scalar interval, while the dynamics and rewards can be nonlinear/nonquadratic. Exploiting ideas from artificial intelligence, we propose two optimistic planning methods that perform an adaptive-horizon search over the infinite-dimensional space of action sequences. The first method optimistically refines regions with the largest upper bound on the optimal value, using the Lipschitz constant to find the bounds. The second method simultaneously refines all potentially optimistic regions, without explicitly using the bounds. Our analysis proves convergence rates to the global infinite-horizon optimum for both algorithms, as a function of computation invested and of a measure of problem complexity. It turns out that the second, simultaneous algorithm works nearly as well as the first, despite not needing to know the (usually difficult to find) Lipschitz constant. We provide simulations showing the algorithms are useful in practice, compare them with value iteration and model predictive control, and give a real-time example.

1. Introduction

We consider optimal control problems that require maximizing a discounted sum of rewards (the value), along an infinitely long discrete-time trajectory of the system. Such problems are encountered in automatic control (Kirk, 2004) as well as in many other fields, including artificial intelligence (AI) (Szepesvári, 2010), operations research, medicine, economics, etc. When the system and reward function have a general form, the problem must be solved approximately with numerical algorithms. A popular class of techniques is approximate dynamic programming (Bertsekas, 2012), which computes offline a near-optimal value function and a state feedback control. Because it searches for a global solution, the complexity of dynamic programming usually grows fast with the state dimensionality (Bertsekas, 2012).

We focus instead on receding-horizon algorithms that remove the direct dependence on the state space size, at the cost of solving a new problem at each step, locally for the current state of the system. A sequence of actions (inputs) is obtained, the initial action of this sequence is applied, and the procedure is repeated online for subsequent states. In automatic control, this is called receding-horizon model predictive control (MPC) (Grüne & Pannek, 2016), while in AI it is called online planning (La Valle, 2006). Note that computation still grows with the action space size and with the search horizon. We search over the space of infinitely long sequences, using the optimistic planning (OP) class of algorithms (Munos, 2014). OP methods originate in AI and perform a branch-and-bound search over the sequences, refining the region with the best upper bound on the value — hence the "optimistic" label. The main strengths of OP are the generality of the dynamics and rewards addressed, and a tight relation between computation and near-optimality, obtained using ideas from bandit theory and reinforcement learning. Many OP variants have been proposed for discrete actions, e.g. by Hren and Munos (2008), Kocsis and Szepesvári (2006) and Máthé, Buşoniu, Munos, and Schutter (2014). Our aim in this paper is to address instead continuous actions, since they are essential in control.

Specifically, we propose two *optimistic planning algorithms with continuous actions* (OPC) that work in systems with general nonlinear dynamics and scalar, compact actions. The methods iteratively split the infinite-dimensional hyperrectangle of continuous-action sequences into smaller hyperrectangles (boxes), leading to an adaptive search horizon. They rely on a central Lipschitz

☆ The material in this paper was partially presented at the 2016 American Control Conference, July 6–8, 2016, Boston, MA, USA. This paper was recommended for publication in revised form by Associate Editor Michael V. Basin under the direction of Editor Ian R. Petersen.

* Correspondence to: Memorandumului 28, 400114 Cluj-Napoca, Romania.
E-mail addresses: lucian@busoniu.net (L. Buşoniu), elod.pall@tu-berlin.de (E. Páll), munos@google.com (R. Munos).

https://doi.org/10.1016/j.automatica.2018.03.009
0005-1098/

property of the value function over action sequences, which is satisfied e.g. when the dynamics and rewards are Lipschitz, with a small enough constant for the dynamics. Using this property, an upper bound on the optimal value is derived using the rewards and the box size. This leads to the first algorithm, which optimistically selects for splitting the box with the largest upper bound, and so it is called simply *OPC*. An essential insight is that each dimension k contributes to the bound with weight γ^k, where γ is the discount factor, and this is used to select which box dimension to split. We characterize the rate at which the box size shrinks with the number of splits, and define a measure of problem complexity, in the form of the branching factor of an associated tree (Hren & Munos, 2008). Using these concepts, we derive an overall convergence rate of the algorithm to the global infinite-horizon optimum as a function of computation, measured by the number of transitions simulated.

A limitation of this first OPC method is that it requires the Lipschitz constant. In practice the constant is difficult to find so it must be treated as a tuning parameter, which is easy to overestimate (making the algorithm conservative) or underestimate (invalidating the guarantees). So we also propose a second algorithm that expands all potentially optimistic boxes, using only the knowledge that boxes that have been split more times have smaller diameters. This algorithm is called *simultaneous OPC (SOPC)*. We analyze SOPC and show that it has nearly the same convergence rate as OPC, even though it does not need to know the value function smoothness. SOPC relies on a different tuning parameter than the Lipschitz constant, which can however be tuned much more robustly. Simulation results illustrate that SOPC outperforms OPC, and is also better than competing continuous-action planners and baseline dynamic programming and MPC solutions. We provide real-time control results with SOPC.

In contrast to much of the work in nonlinear MPC (Grüne & Pannek, 2016), which uses a fixed finite horizon, OPC and SOPC directly explore the space of infinite-horizon solutions, and therefore our near-optimality bounds and convergence rates are with respect to the global, infinite-horizon optimum. E.g. the closest work to ours is the optimistic MPC method of Xu, van den Boom, and Schutter (2016), which only works for small fixed control horizons (and max-plus systems). OPC and SOPC instead adaptively increase the horizon as much as the computation allows. Moreover, typical MPC methods are derivative-based, while our methods only rely on Lipschitz values, so at the cost of more computation, they can handle dynamics and rewards that are nondifferentiable at some discrete points (on a set of measure zero).

In planning, several other optimistic methods have been proposed for continuous actions, but without an analysis; to our knowledge OPC and SOPC are the first to guarantee a convergence rate. Lipschitz planning (LP) (Hren, 2012) uses a similar upper bound but lacks the insight on the impact γ^k, so it uses a heuristic rule to choose which dimension to split. Our earlier method called simultaneous optimistic optimization for planning (SOOP) (Buşoniu, Daniels, Munos, & Babuška, 2013) is similar to SOPC in that it expands many boxes at once, but uses a heuristic for selecting these boxes, which turns out to be worse in our simulations. Other continuous-action planners only optimize over finite horizons, e.g. HOOT (Mansley, Weinstein, & Littman, 2011) or sequential planning (Hren, 2012).

Our new planners apply the *principle* of optimistic optimization (OO) (Munos, 2011) to control, while the *analysis* of OO does not work because its assumptions are not satisfied for infinite-horizon continuous-input problems. Thus, we must provide novel analysis adapted to this setting. The present paper is an extended and revised version of Buşoniu, Páll, and Munos (2016), and the material on OPC largely originates in that paper. The major novelty compared to Buşoniu et al. (2016) is the SOPC method, with almost as good analytical guarantees and much better practical performance than OPC. Even for OPC, we provide here extra insight that due to space limits was not available in Buşoniu et al. (2016). The simulations are extended and the real-time results are new.

Next, Section 2 formalizes the problem and Section 3 describes OPC and SOPC. Section 4 analyzes the two algorithms, while Section 5 provides numerical results. Section 6 concludes the paper. Supplementary material is available at http://busoniu.net/files/papers/sopc_suppl.pdf.

2. Problem statement

We consider an optimal control problem for a discrete-time nonlinear system $x_{k+1} = f(x_k, u_k)$, where $x \in X \subseteq \mathbb{R}^p$, $u \in U$, and U will be described in our main assumption below. A function $\rho : X \times U \to \mathbb{R}$ assigns a numerical reward $r_k = \rho(x_k, u_k)$ to each state–action pair. Under a fixed initial state x_0, define an infinitely-long sequence of actions $\boldsymbol{u}_\infty = (u_0, u_1, \ldots)$ and its infinite-horizon discounted value:

$$v(\boldsymbol{u}_\infty) = \sum_{k=0}^{\infty} \gamma^k \rho(x_k, u_k) \tag{1}$$

where $\gamma \in (0, 1)$ is the discount factor, and $x_{k+1} = f(x_k, u_k)\ \forall k \geq 0$. The objective is to find (a near-optimal approximation of) the optimal value $v^* := \sup_{\boldsymbol{u}_\infty} v(\boldsymbol{u}_\infty)$ and an action sequence that achieves this (near-optimal) value. Very general conditions that ensure the existence of optimal sequences are provided e.g. by Bertsekas and Shreve (1978).

We impose some assumptions that allow us to derive efficient algorithms.

Assumption 1. The following conditions hold.

(i) The rewards are bounded in $[0, 1]$.
(ii) The action is a real scalar, bounded in the unit interval, so that $U = [0, 1]$.

The main role of reward boundedness 1(i), together with discounting, is to ensure that for any sequence the values in (1) are bounded to $[0, \frac{1}{1-\gamma}]$. Our planning algorithms and analysis rely on this property. Note that many other works in control use discounting, e.g. Filar, Gaitsgory, and Haurie (2001) and Katsikopoulos and Engelbrecht (2003). Bounded costs are typical in AI methods for optimal control, such as reinforcement learning (Szepesvári, 2010). One way to achieve boundedness is by saturating a possibly unbounded original reward function, which changes the optimal solution but is often sufficient in practice. Another example is when physical limitations in the system are modeled by saturating the states and actions, from which a reward bound follows.

The scalar action from Assumption 1(ii) could in principle be generalized to multiple dimensions; however, computation would grow very fast with action dimensionality, so in practice this will not work for more than a few dimensions. (In the supplementary material, we briefly explain and empirically test such an extension for two dimensions.) The compact nature of U is fundamental, since our algorithm numerically refines this action space. In both Assumptions 1(i) and (ii), the unit interval is taken only for convenience, and can be achieved by rescaling any bounded interval.

A crucial requirement is a Lipschitz property of v with respect to its argument $\boldsymbol{u}_\infty$, "one-sided" around optimal sequences.

Assumption 2. There exists $L_v > 0$ so that for any optimal sequence $\boldsymbol{u}^*_\infty$ and any other sequence $\boldsymbol{u}_\infty \in U^\infty$:

$$v(\boldsymbol{u}^*_\infty) - v(\boldsymbol{u}_\infty) \leq L_v \sum_{k=0}^{\infty} \gamma^k \left| u^*_k - u_k \right|. \tag{2}$$