Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 50-1 (2017) 11814–11821

Approximate Dynamic Programming via Penalty Functions ⋆

Paul N. Beuchat * **John Lygeros** *

* *Automatic Control Laboratory, ETH Zürich, ZH 8006 Switzerland (e-mail: {beuchatp,lygeros}@control.ee.ethz.ch).*

Abstract: In this paper, we propose a novel formulation for encoding state constraints into the Linear Programming approach to Approximate Dynamic Programming via the use of penalty functions. To maintain tractability of the resulting optimization problem that needs to be solved, we suggest a penalty function that is constructed as a point-wise maximum taken over a family of low-order polynomials. Once the penalty functions are designed, no additional approximations are introduced by the proposed formulation. The effectiveness and numerical stability of the formulation is demonstrated through examples.

Keywords: Stochastic optimal control problems, Approximate dynamic programming, Soft constraints

1. INTRODUCTION

All engineering problems place requirements on the states of a systems to ensure safe and practical operation, for example a robotic arm should not extend out beyond its defined workspace. In the Stochastic Optimal Control (SOC) framework, these requirements are encoded as constraints on the state and input spaces. However, finding an exact solution to such problems is in general intractable and leads to the curse of dimensionality (Powell, 2014). In the literature of Approximate Dynamic programming (ADP), many methods have been proposed for computing sub-optimal solutions of SOC problems, ranging from model-free to model-based methods, see Bertsekas and Tsitsiklis (1996) and Bertsekas (2013). The Linear Programming (LP) approach to ADP is a model-based method first introduced by Schweitzer and Seidmann (1985) and equipped with performance guarantees by De Farias and Van Roy (2003). Although it is possible to include state constraints with the LP approach, the approximation techniques proposed either: lead to intractable problems, require problem specific tuning, or don't respect the constraints. In this paper, we propose a formulation for including constraints that addresses all of these shortcomings.

State constraints are categorized as either: (i) *hard constraints* that must always be satisfied to avoid failure of the system, or (ii) *soft constraints* that are preferable to satisfy but for which violations do not lead to system failure. Sub-optimal policies synthesized through the LP approach to ADP cannot in general guarantee online constraint satisfaction (Chen and Blankenship, 2004). Hence we consider problems with soft constraints. Building climate control is one motivating example for the use of soft constraints (Sturzenegger et al., 2016).

Soft constraints are commonly encoded in SOC problems using penalty functions, which impose a high cost on constraint violations, leading to an overall stage cost function that can be non-smooth. In this case, the optimal cost-to-go function, which characterizes the solution of the SOC problem, will also be non-smooth (Kerrigan and Maciejowski, 2000). Recent developments of the LP approach to ADP by Wang et al. (2014) and Summers et al. (2013) used polynomial functions to improve the approximation quality. However, the approximation quality degrades for non-smooth optimal cost-to-go functions.

Recent work has suggested that improved approximations of the optimal cost-to-go function can be achieved by taking the point-wise maximum over a family of polynomial approximations, see O'Donoghue et al. (2011) and Beuchat et al. (2016). This work is promising because it allows non-smooth approximations to be constructed with minimal tuning effort. The key ingredient missing is the ability to include penalty functions into the framework. High order polynomials can encode soft constraints into the formulation of Summers et al. (2013), however, even for simple box constraints, it leads to optimization problems that are either intractably large or suffer numerical issues (Lasserre, 2001).

In this paper, we study penalty functions constructed as the point-wise maximum over a family of polynomials and propose a formulation for their inclusion into the LP approach to ADP. On the theoretical side, our main contribution is proving that our proposal is an exact reformulation, and as such it does not introduce additional approximation steps. On the practical side, our proposed formulation: (i) allows the use of non-smooth penalty functions, (ii) requires only a small increase in computational burden, and (iii) is numerically stable. Overall, our contribution broadens the applicability of the LP approach.

The paper is structured as follows. Section 2 introduces the soft constraint formulation considered and Section 3

⋆ This research was partially funded by the European Commission under the project Local4Global.

10.1016/j.ifacol.2017.08.1993

incorporates this into the theoretical framework of the LP approach to DP, additionally providing practical guidance for computing approximations. Section 4 investigates the effectiveness and behavior of our proposed formulation through numerical examples.

2. DYNAMIC PROGRAMMING (DP) FORMULATION

This section introduces the problem formulation with general constraints on the state-by-input space and uses the Bellman equation to characterize solutions. We consider infinite horizon, discounted cost, constrained stochastic optimal control problems. The formulation with hard constraints is introduced first and used to motivate the specific soft constraint formulation studied throughout the paper.

2.1 Formulation with Hard Constraints

The system is described by discrete-time dynamics over continuous state and action spaces. The state of the system at time t is $x_t \in \mathcal{X} \subseteq \mathbb{R}^{n_x}$. The system state is influenced by the control decisions $u_t \in \mathcal{U} \subseteq \mathbb{R}^{n_u}$, and the stochastic disturbance $\xi_t \in \Xi \subseteq \mathbb{R}^{n_\xi}$. In this setting, the state evolves according to the function $g : \mathcal{X} \times \mathcal{U} \times \Xi \to \mathcal{X}$ as, $x_{t+1} = g(x_t, u_t, \xi_t)$. At time t, the system incurs the stage cost $\gamma^t l(x_t, u_t)$, where $\gamma \in [0, 1)$ is the discount factor. The objective is to minimize the infinite sum of the stage costs, while ensuring the the hard constraint $(x, u) \in \mathcal{C} \subset \mathcal{X} \times \mathcal{U}$ is satisfied at all time steps.

The optimal Value function, $V^* : \mathcal{X} \to \mathbb{R}$, characterizes the solution of this constrained optimal control problem. It represents the cost-to-go from any state of the system if the optimal control policy is played. In order to write out the Bellman equation that V^* satisfies, the hard state-by-input constraint is encoded by adapting the stage cost to be infinite for constraint excursions. Defining,

$$l_{\text{hard}}(x, u) = \begin{cases} l(x, u) & \text{for } (x, u) \in \mathcal{C}\,, \\ +\infty & \text{otherwise}\,, \end{cases}$$

the optimal Value function is the solution of the Bellman equation (Bellman, 1952),

$$V^*(x) = \inf_{u \in \mathcal{U}} l_{\text{hard}}(x, u) + \gamma\, \mathbb{E}\left[V^*\left(g(x, u, \xi)\right)\right]\,, \tag{1}$$

for all $x \in \mathcal{X}$. The optimal control actions are generated via the *Greedy Policy*,

$$\pi^*(x) = \arg\min_{u \in \mathcal{U}} \; l_{\text{hard}}(x, u) + \gamma\, \mathbb{E}\left[V^*\left(g(x, u, \xi)\right)\right]\,. \tag{2}$$

By Π we denote the set of all feasible policies, i.e., $\{\pi(\cdot) : \pi(x) \in \mathcal{U},\, \forall x \in \mathcal{X}\}$.

In order for (1) to have a solution V^*, we assume that $\exists \pi \in \Pi$ with bounded infinite horizon costs for some $x \in \mathcal{X}$. In the context of hard constraints, this assumption ensures the existence of a policy satisfying $(x_t, u_t) \in \mathcal{C}$ for all t. In addition, to ensure V^* is measurable and attains the infimum, we work under Assumptions 4.2.1(a) and 4.2.1(b) of Hernández-Lerma and Lasserre (2012), which place mild requirements on the stage cost function, dynamics, and exogenous disturbance process. These assumptions apply equally for the soft-constraint formulation presented next.

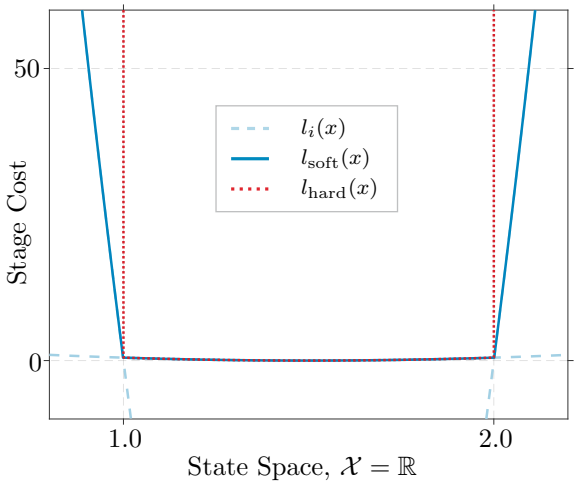

Fig. 1. Stage cost encoding of the constraint $x \in [1, 2] = \mathcal{C}$. The solid blue line is a possible l_{soft}, constructed from three l_i quadratics shown in dashed blue. The dotted red line is l_{hard}.

2.2 Formulation with Soft Constraints

As l_{hard} cannot be used in the Linear Programming (LP) approach to ADP, we describe here the soft-constraint approximation of l_{hard} that is studied in this paper. We consider soft-constraint stage cost functions of the form,

$$l_{\text{soft}}(x, u) = \max_{i \in \mathcal{I}} l_i(x, u)\,,$$

where $\mathcal{I}$ is some index set and $l_i : (\mathcal{X} \times \mathcal{U}) \to \mathbb{R}$. For each problem instance, the family of l_i functions must be designed to encode a shape similar to l_{hard}, i.e., we require:

- $l_{\text{soft}}(x, u) = l(x, u)$ for all $(x, u) \in \mathcal{C}$,
- $l_{\text{soft}}(x, u)$ grows steeply as the distance from (x, u) to $\mathcal{C}$ increases.

In section 3.8, we describe a method for constructing l_{soft} when $\mathcal{C}$ is a polytope.

To maintain the tractability of the approximation method described in Section 3.3, the l_i functions should be from the same class of functions as l. Consider, for example, a quadratic l, the point-wise maximum allows l_{soft} to grow steeply outside of $\mathcal{C}$ using only quadratics, i.e., without higher order polynomials. To illustrate the theory, we introduce an example that runs throughout the paper. Consider a system with $n_x = n_u = 1$, a nominal stage cost $l(x, u) = x^2 + u^2$, and state constraint $x \in [1, 2] = \mathcal{C}$ that should be encoded with the stage cost. Figure 1 shows the l_{soft} chosen to encode the state constraint according to the requirements above.

Given l_{soft}, the Bellman equation for this formulation is,

$$V^*(x) = \underbrace{\min_{u \in \mathcal{U}} l_{\text{soft}}(x, u) + \gamma\, \mathbb{E}\left[V^*\left(g(x, u, \xi)\right)\right]}_{(\mathcal{T}_{\text{s}} V^*)(x)}\,, \tag{3}$$

where $\mathcal{T}_{\text{s}}$ is the Bellman operator and the subscript "s" is used to make explicit the use of l_{soft}. The greedy policy is identical to (2), expect with l_{hard} replaced by l_{soft}. The next section provides exact and approximate solution methods for (3).