Available online at www.sciencedirect.com

SCIENCE DIRECT®

automatica

Automatica 41 (2005) 767–778

www.elsevier.com/locate/automatica

Stochastic iterative dynamic programming: a Monte Carlo approach to dual control☆

Adrian M. Thompson, William R. Cluett*

Department of Chemical Engineering and Applied Chemistry, University of Toronto, Toronto, Ont., Canada M5S 3E5

Received 5 July 2002; received in revised form 15 November 2004; accepted 7 December 2004

Abstract

Practical exploitation of optimal dual control (ODC) theory continues to be hindered by the difficulties involved in numerically solving the associated stochastic dynamic programming (SDPs) problems. In particular, high-dimensional hyper-states coupled with the nesting of optimizations and integrations within these SDP problems render their exact numerical solution computationally prohibitive. This paper presents a new stochastic dynamic programming algorithm that uses a Monte Carlo approach to circumvent the need for numerical integration, thereby dramatically reducing computational requirements. Also, being a generalization of iterative dynamic programming (IDP) to the stochastic domain, the new algorithm exhibits reduced sensitivity to the hyper-state dimension and, consequently, is particularly well suited to solution of ODC problems. A convergence analysis of the new algorithm is provided, and its benefits are illustrated on the problem of ODC of an integrator with unknown gain, originally presented by Åström and Helmersson (Computers and Mathematics with Applications 12A (1986) 653–662).

Keywords: Dual control; Adaptive control; Stochastic systems; Dynamic programming; Optimal control; Uncertainty

1. Introduction

The optimal dual control (ODC) idea is to predict and make use of future model estimates during the control calculation. Instead of using a fixed model for computation of the entire control policy, in ODC, later controls are computed using models predicted to result from those controls applied earlier. The resulting optimal policy balances system excitation, to improve future model accuracy, against system regulation, to achieve good immediate control. The name "dual control" arises from the need to simultaneously satisfy these two competing objectives.

ODC has proven extremely difficult to implement in practice due to several computational issues and as a result most researchers in the dual control field have been discouraged from attempting to solve the ODC problem. Instead, the recent trend has been towards development of sub-optimal approaches to dual control. Although proving simpler than ODC to implement in practice, these approaches have several disadvantages (Lindoff, Holst, & Wittenmark, 1999; Filatov & Unbehauen, 2000).

Numerically, ODC involves computation of a state- and time-dependent control policy defined over a finite control horizon to minimize a stochastic cost function subject to the dynamics of a non-linear time-varying hyper-system with continuous hyper-state[1] and control spaces. Such problems are extremely difficult to solve because (i) existing numerical approaches encounter the curse of dimensionality, which is amplified in ODC due to the presence of model parameters and their uncertainty descriptions within the hyper-state, (ii)

☆ This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Editor R.R. Bitmead.

* Corresponding author. Tel.: +1 416 978 5889; fax: +1 416 978 8605.
E-mail address: cluett@ecf.utoronto.ca (W.R. Cluett).

[1] The hyper-state is the state vector for the augmented system that includes the original system equations along with those used to predict future model adaptation.

0005-1098/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2004.12.003

non-standard probability distributions arise from the feedback of uncertainty, preventing the cost function from being written as a closed-form analytical expression and (iii) use of parameterized control policy representations is difficult because the optimal policy is invariably discontinuous over the hyper-state space.

The contribution of this paper lies in its identification of mechanisms by which ODC problems can be numerically solved without complete hyper-state discretization or nested numerical integration. We show that the original ODC problem can be made numerically tractable via a combination of iterative dynamic programming (IDP) (Luus, 2000a) and Monte Carlo methods. The result is a significant mitigation of the curse of dimensionality.

The use of Monte Carlo methods in the solution of SDP problems is not novel. Algorithms such as value iteration,[2] policy iteration, Q-learning and neuro-dynamic programming are well-known dynamic programming approaches that employ Monte Carlo sampling in stochastic settings (Mitchell, 1997; Sutton & Barto, 1998; Bertsekas & Tsitsiklis, 1996). The SIDP algorithm presented here provides an advantage over these approaches in that it does not employ complete state-space discretization and is therefore less sensitive to the curse of dimensionality. This makes it applicable to higher dimensional SDP problems, e.g. ODC, which have proven unmanageable using these other approaches. A similar effect has been achieved in other work (de Farias & van Roy, 2003; Hauskrecht & Kveton, 2004), by reformulating SDP problems into linear programming (LP) problems and approximating the large number of resulting constraints using Monte Carlo sampling. SIDP differs from these LP approaches in that it also extends to unbounded, continuous state and action spaces, its solution accuracy is independent of the state-space discretization, and it does not require basis function policy representations in order to reduce dimensional sensitivity.

In Section 2 of this paper we describe the ODC problem, present its dynamic programming formulation, and elaborate on the complexities of its numerical solution. Section 3 discusses dynamic programming in the context of ODC and explains how the numerical difficulties introduced by stochasticity can be alleviated by use of Monte Carlo methods. The new SDP solution algorithm, SIDP, is also presented in this section. A convergence analysis of SIDP is provided in Section 4. The problem of ODC of an integrator with unknown gain (Åström & Helmersson, 1986) is adopted as a benchmark in Section 5 to illustrate the advantages of the new algorithm relative to the standard dynamic programming approach.

[2] Value iteration is the standard numerical approach used for solution of dynamic programming problems and is henceforth referred to as such (cf. Section 3.1).

2. Optimal dual control

2.1. Problem description

The ODC problem of interest in this paper takes the following form:

(1) Given a single-input single-output (SISO), linear time invariant (LTI), uncertain model of a dynamical system, of the form

$$
\begin{aligned}
&y_{t+1} = \varphi_t^{\mathrm{T}}\theta_t + e_{t+1},\\
&\theta_t \sim N(\hat{\theta}_t, P_t),\\
&e_t \sim N(0, \sigma^2),\\
&\mathrm{Cov}(e_t, \theta_{t,i}) = 0 \quad \forall t, i,
\end{aligned} \tag{1}
$$

in which

$$
\varphi_t = \begin{bmatrix} u_t \\ u_{t-1} \\ \vdots \\ u_{t-n_u+1} \\ -y_t \\ -y_{t-1} \\ \vdots \\ -y_{t-n_y+1} \end{bmatrix} \quad \text{and} \quad \theta_t = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n_u} \\ a_1 \\ a_2 \\ \vdots \\ a_{n_y} \end{bmatrix},
$$

(2) with model adaptation governed by the equations

$$
\begin{aligned}
&K_t = P_t\varphi_t(\varphi_t^{\mathrm{T}}P_t\varphi_t + \sigma^2)^{-1},\\
&\hat{\theta}_{t+1} = \hat{\theta}_t + K_t(y_{t+1} - \varphi_t^{\mathrm{T}}\hat{\theta}_t),\\
&P_{t+1} = (I - K_t\varphi_t^{\mathrm{T}})P_t,
\end{aligned} \tag{2}
$$

(3) compute the feedback control policy

$$
\begin{aligned}
&u_t^* = u_t^*(y_t, y_{t-1}, \ldots, y_0, u_{t-1}, u_{t-2}, \ldots, u_0),\\
&t = 0, 1, 2, \ldots, N-1,
\end{aligned} \tag{3}
$$

that minimizes the expected tracking error

$$
J_0 = E\left\{\sum_{t=0}^{N-1}(y_{t+1} - r_{t+1})^2\right\} \tag{4}
$$

between the predicted system output, y, and a given reference signal, $r = \{r_t,\ 1 \leqslant t \leqslant N\}$,

(4) using a priori knowledge of φ_0, $\hat{\theta}_0$, P_0, σ^2 and N.

2.2. Dynamic programming formulation

The ODC problem can be formulated as an SDP problem by combining Eqs. (1) and (2) into a single "hyper-system", for which the associated hyper-state is

$$
H_t = [\tilde{\varphi}_t, \hat{\theta}_t, P_t]. \tag{5}
$$