APPROXIMATE DYNAMIC PROGRAMMING STRATEGY FOR DUAL ADAPTIVE CONTROL

Jong Min Lee [*] **Jay H. Lee** [*,1]

[*] *School of Chemical and Biomolecular Engineering,*
Georgia Institute of Technology, Atlanta, GA 30332, USA

Abstract: An approximate dynamic programming (ADP) strategy for a dual adaptive control problem is presented. An optimal control policy of a dual adaptive control problem can be derived by solving a stochastic dynamic programming problem, which is computationally intractable using conventional solution methods that involve sampling of a complete hyperstate space. To solve the problem in a computationally amenable manner, we perform closed-loop simulations with different control policies to generate a data set that defines a subset of a hyperstate within which the Bellman equation is iterated. A local approximator with a penalty function is designed for estimation of cost-to-go values over the continuous hyperstate space. An integrating process with an unknown gain is used for illustration. *Copyright*© *2005 IFAC*

Keywords: approximate dynamic programming, dual control, adaptive control, stochastic dynamic programming

1. INTRODUCTION

Practical control problems are characterized by mismatches between model and plant, which can be caused by structural/parametric uncertainties and unknown exogenous disturbances. Uncertainties may be modeled using either deterministic bounds or stochastic processes. In the latter case, the usual approach is to combine parameter estimation and control into an adaptive control strategy. The estimator delivers information about the unknown parameters, such as their mean values and covariances. Different classes of adaptive controllers are obtained depending on how the information is utilized. The most popular approach is to perform a control calculation by assuming that the estimated parameter values are true values; this approach is referred to as the 'certainty equivalence' approach. This, however, disregards uncertainties in the parameter estimates and can lead to severe robustness problems such as the "bursting" phenomenon. In addition, the disregard of the coupling between the estimation and control makes the learning "passive," meaning the controller does not make exploratory moves to actively generate information about important parametric uncertainties.

To obtain useful information about the process dynamics, it is necessary to perturb the process in general. On the other hand, such a perturbation may not be favorable from a viewpoint of closed-loop performance. Thus, there is a conflict between information gathering and present control quality. This problem was first introduced and discussed by Fel'dbaum in his series of papers published in the early 60s (Fel'dbaum, 1960, 1960, 1961, 1961). The optimal controller has *dual* goals, and it should balance between control and exploration. By gaining more process information when needed, better control performance

[1] To whom all correspondence should be addressed. E-mail: jay.lee@chbe.gatech.edu

can be achieved in the future. Fel'dbaum also showed that Dynamic Programming (DP) can be solved to obtain the optimal solution to the dual control problem. It has been thought that the DP solution for the problem is intractable, and only a few very simple examples have been solved this way after reducing the problem size through some analytical insights into the specific problem (Åström and Helmersson, 1986). Due to the computational complexity, most researchers approached the problem by introducing cautious and active probing features to simpler suboptimal controllers in a somewhat ad hoc manner (Lindoff *et al.*, 1999; Chikkula and Lee, 2000).

In this work, an approximate dynamic programming (ADP) method is proposed to solve the dual optimal control problem. The approach enables us to combine the merits of the different starting policies systematically through interpolation and improvement of cost-to-go values in a hyperstate space. If successful, the derived ADP-based controller should demonstrate a well-balanced dual feature. Section 2 presents a dual adaptive control problem. Section 3 discusses a general procedure for applying the ADP approach to the dual optimal control problem. An example of integrator with unknown gain is presented in Section 4. Section 5 provides conclusions.

2. STOCHASTIC ADAPTIVE CONTROL

2.1 Problem Formulation

We consider a discrete time model described as

$$x(k+1) = f(x(k), u(k), \theta(k), \zeta(k)) \tag{1}$$

where $x(k)$ is a state vector, which is assumed to be measured, $u(k)$ is a manipulated input vector, $\theta(k)$ is a vector containing unknown parameters of the model, and $\zeta(k)$ is an exogenous noise, which we assume here to be independently identically distributed (i.i.d.) Gaussian. We also assume that the structure f is known and the unknown parameter θ is also described by a Gaussian process.

The control objective is to minimize an infinite horizon cost:

$$E\left[\sum_{t=0}^{\infty} \alpha^t \phi(x(k+t), u(k+t)) \middle| \xi(k)\right] \tag{2}$$

where $\alpha \in [0, 1)$ is a discount factor, and expectation operator E is taken over the distribution of ζ and θ. $\xi(k)$ is an information state (or *hyperstate*) at time k, which includes the process state $x(k)$ and the first two moments of a posteriori probability density function of the Gaussian parameter vector.

$$\xi(k) = \left[x(k),\ \hat{\theta}(k),\ P(k)\right]^T \tag{3}$$

where $\hat{\theta}$ and P are the conditional mean and the covariance matrix of θ (conditioned by the measurements), respectively. A feasible control policy is the one that determines $u(k)$ based on the information available at time k (i.e. $\xi(k)$).

A closed-loop optimal solution to (2) assumes that the future inputs are determined in a feedback-optimal sense, which means they are dependent on the future hyperstates. The optimal control policy can be derived by solving the following stochastic dynamic programming:

$$\begin{aligned} J^*(\xi(k)) = \min_{u(k)} E\ [\phi(x(k), u(k)) \\ +\alpha J^*(\xi(k+1))\ | \xi(k)] \end{aligned} \tag{4}$$

The above equation is also referred to as *Bellman equation.*

In real-time implementation, control action is calculated at each sample time by

$$\begin{aligned} u(k) = \mu^*(\xi(k)) = \arg\min_{u(k)} E\,[\phi(x(k), u(k)) \\ +\alpha J^*(\xi(k+1))\ | \xi(k)] \end{aligned} \tag{5}$$

Note that $\xi(k+1)$ is a stochastic variable and is affected by the choice of control action $u(k)$. The difficulty in solving the above is that the minimization, which requires expectation calculation for each evaluation of a candidate u, must be solved for all the points in a densely gridded hyperstate space. The control action influences the immediate cost ϕ, quality of future estimation (reflected through future hyperstate ξ), and future control performance. Even though the optimal controller will have the desired dual feature, the DP formulation is intractable in all but simplest cases if the conventional solution approach (e.g., value iteration, policy iteration) is taken.

2.2 Passive Learning Policies

This section introduces popular "passive" control policies, the certainty equivalence (CE) control policy and the cautious control policy. The CE policy calculates a control action at each sample time as if the estimate $\hat{\theta}(k)$ were exact:

$$u_{CE}(k) = \mu_{CE}(x(k),\ \hat{\theta(k)}) \tag{6}$$

The inputs are designed without any regard for their effects on future estimation quality, which can make the achieved performance substantially suboptimal and cause intermittent instability phenomenon known as 'bursting' (Anderson, 1985).

A simple design that takes into account the uncertainty is the 'cautious' controller that minimizes the cost function of (2) only for a single step. Note that, for a single step problem, u can be optimized as a deterministic variable. This adds a measure of caution to account for uncertainty in that the gain in the controller is decreased as the uncertainty is increased. It does not, however, take into account the effect of a control action on future estimation quality, and can lead to *turn off* of the controller if the uncertainty becomes too large. The cautious policy is also a passive learning controller because there is no active probing signal generated to improve the identification.

3. APPROXIMATE DYNAMIC PROGRAMMING

In this section, we describe the ADP procedure for solving the previously described stochastic control problem. Due to the difficulties in computing the exact solution to the DP formulation, several approximate solutions have been proposed (Wittenmark, 2002). One of them is to find an approximate solution for two-step ahead cost-to-go function (Lindoff *et al.*, 1999). It is, however, still very complex and is restricted to simple problems.

To alleviate the computational barrier of *curse-of-dimensionality*, which refers to the exponential growth of the computation with respect to the state dimension, the ADP approach attempts to solve the Bellman equation only approximately within limited confines of the hyperstate space visited in closed-loop simulations of suboptimal policies. Function approximator is used to provide continuous cost-to-go estimate based on the discrete data. The cost-to-go function approximation is then improved through either iteration of the Bellman equation (value iteration) or the iteration between the Bellman equation and the policy evaluation (policy iteration). The rationale behind this idea is that even though the state space may be huge, the dimension of the manifold for optimal control for relevant conditions may be much lower, given a small number of disturbance patterns and set-point changes occurring in real operations.

From the method, one gets an approximation of the optimal cost-to-go function, which maps the state to the cost-to-go value under the optimal control. The cost-to-go map then can be translated into an on-line control policy, which involves solving a single-stage optimization problem rather than a multi-stage one. Its potentials have been demonstrated in several nonlinear control problems (Kaisare *et al.*, 2002).

The followings are the outline of the suggested approach for dual control problems.

- Step 1: Closed-loop simulations.

  Perform closed-loop simulations with chosen suboptimal control policies (μ^0) under all relevant operating conditions. Since the ADP algorithm derives an improved control policy from the data visited by starting policies, it is preferable to simulate with different control policies having the characteristics of cautiousness and active exploration. For example, passive controllers with dither signals can be used.

- Step 2: Approximation of the initial cost-to-go values.

  Using the simulation data, we first calculate the infinite horizon cost-to-go, $\tilde{J}^{\mu^0}$, for each state visited during the simulation according to

$$J^{\mu^0}(\xi(k)) = \sum_{t=0}^{\infty} \alpha^t \phi(x(k+t), u(k+t)) \quad (7)$$

With the data, we construct a function approximator to approximate the cost-to-go as a function of continuous hyperstate variables, denoted hereafter as $\tilde{J}^{\mu^0}$. Here we use a local averager of the following form:

$$\tilde{J}(\xi_0) = \beta_0 J_0 + \sum_{i=1}^{n} \beta_i J(\xi_i) \quad (8)$$

with

$$\sum_{i=0}^{n} \beta_i = 1, \quad \beta_i \geq 0 \quad (i = 0, \cdots, n) \quad (9)$$

where ξ_0 is a query point, J_0 is a bias term, and n is the number of neighboring points in the data set for the approximation. K-nearest neighborhood and kernel-based averagers are of this class of approximators. It can be proved that this local averaging scheme guarantees the convergence of the value iteration (Lee, 2004); however, it can still introduce significant bias in regions of the state space where the data density is inadequately low. To systematically restrict the search regions for the control actions, a penalty function based on the estimate of a local data density is employed. A nonparametric density estimate for the query point ξ_0 using a training data set Ω is given by

$$f_\Omega(\xi_0) = \frac{1}{n\sigma^{m_0}} \sum_{i=1}^{n} K\left(\frac{\xi_0 - \xi_i}{\sigma}\right) \quad (10)$$

where m_0 is the hyperstate dimension, K is a selected kernel function, and σ is a user-given bandwidth parameter. We use the following multivariate Gaussian function for the kernel:

$$K(\cdot) = \frac{1}{(2\pi\sigma^2)^{\frac{m_0}{2}}} \exp\left(-\frac{\|\xi_0 - \xi_i\|_2^2}{2\sigma^2}\right) \quad (11)$$

To use the cost-to-go approximator cautiously, we incorporate into the cost-to-go a quadratic penalty-term based on $f_\Omega(\xi_0)$.

$$\tilde{J}(\xi_0) \Longleftarrow \tilde{J}(\xi_0) + J_{\text{bias}}(\xi_0) \tag{12}$$

$$J_{\text{bias}} = AH\left(f_\Omega^{-1}(\xi_0) - \rho\right)\left[\frac{\frac{1}{f_\Omega(\xi)} - \rho}{\rho}\right]^2 \tag{13}$$

where H is a heavy-side step function, A is a scaling parameter, and ρ is a threshold value. In this work, ρ is the data density corresponding to $\|\xi_0 - \xi_i\|_2 = \sigma$, and A is calculated so that some large cost-to-go value, say $J_{\max}$, is assigned to J_{bias} at $\|\xi_0 - \xi_i\|_2 = 3\sigma$. The penalty function biases upward the estimate of cost-to-go for a query point in a manner inversely proportional to the data density, which discourages the optimizer from driving the system into unexplored regions where data density is inadequately low.

We also note that the expectation operator is not explicitly evaluated in this step, but the function approximator should smoothen the stochastic nature giving a good estimate of the expected cost-to-go value. However, this is not so critical because the off-line iteration step will refine the cost-to-go with explicit evaluation of the expectation operator.

- Step 3: Improvement of cost-to-go estimates using value iteration.

  To improve the cost-to-go approximation, we perform the value iteration until convergence according to

$$\begin{aligned} J^{i+1}(\xi(k)) &= \min_{u(k)} E\left[\phi(x(k), u(k))\right. \\ &\qquad \left. + \alpha\tilde{J}^i(\xi(k+1))\right] \end{aligned} \tag{14}$$

  where superscript i denotes the ith iteration step. J^{i+1} is calculated for every $\xi(k)$ in the data set, and $\xi(k+1)$ is a successor state after applying the control action $u(k)$.

  This step is complicated by the expectation operator coupled with the minimization. The expectation operator is evaluated by sampling the *innovation* term, which is also affected by the control action. We not only sample the control actions used in the suboptimal control policies but discretize the actions with a reasonable grid size. Each candidate action gives probability distribution of the corresponding innovation, according to which the possible outcomes of hyperstate are sampled using the Monte Carlo simulation.
- Step 4: On-line implementation.

  With the converged cost-to-go values, $\tilde{J}^*$, control action in real-time is calculated from the following minimization:

$$\begin{aligned} u(k) &= \arg\min_{u(k)} E\left[\phi(\xi(k), u(k))\right. \\ &\qquad \left. + \alpha\tilde{J}^*(\xi(k+1))\right] \end{aligned} \tag{15}$$

4. EXAMPLE

The ADP strategy is now applied to the integrator process with two different scenarios: A step change in the gain parameter and continuous drifts in the parameter.

4.1 Problem Statement

Consider the integrator process (Åström and Helmersson, 1986) described by

$$y(k+1) = y(k) + bu(k) + e(k+1) \tag{16}$$

where $y(k)$ is the output, $u(k)$ is the manipulated input, $e(k)$ is a white noise, and b is an unknown parameter. e follows the normal distribution of

$$e \sim \mathcal{N}(0, \sigma^2) \tag{17}$$

Furthermore, the unknown parameter b can vary in time and its behavior is modeled as

$$b(k+1) = \phi b(k) + \gamma w(k) \tag{18}$$

where $w(k)$ is a Gaussian white noise.

The control objective is to minimize the following discounted infinite horizon objective function:

$$E\left[\sum_{t=k+1}^{\infty} \alpha^{t-(k+1)}\left[y(t)\right]^2 \middle| \mathcal{Y}_k\right] \tag{19}$$

where $\mathcal{Y}_k$ denotes the sequence of observed outputs and inputs available at time k. Given the measurements $\mathcal{Y}_k$, the estimator generates the conditional probability distribution of the parameter b as follows:

$$\hat{b}(k) = E\left\{b(k) \middle| \mathcal{Y}_k\right\} \tag{20}$$

$$P(k) = E\left\{\left[\hat{b}(k) - b(k)\right]^2 \middle| \mathcal{Y}_k\right\} \tag{21}$$

They can be calculated recursively according to

$$\begin{aligned} \hat{b}(k+1) &= \phi\hat{b}(k) + K(k)\left[y(k+1)\right. \\ &\qquad \left. - y(k) - \hat{b}(k)u(k)\right] \end{aligned} \tag{22}$$

$$K(k) = \frac{\phi P(k)u(k)}{\sigma^2 + P(k)u^2(k)} \tag{23}$$

$$P(k+1) = \frac{\phi^2\sigma^2 P(k)}{P(k)u^2(k) + \sigma^2} + \gamma^2 R_w \tag{24}$$

where R_w is the variance of w. The hyperstate of the process, $\xi(k)$, is defined as $\xi(k) = [y(k),\ \hat{b}(k),\ P(k)]^T$.

4.2 Example 1: Step disturbance

Simulation Scenarios In most cases, the following CE controller is nearly-optimal for the given problem:

$$u(k) = -\frac{y(k)}{\hat{b}(k)} \tag{25}$$

We consider a simple but somewhat idealized case where the gain b can jump from the initial value of 0.5 to a value between -15 and 15 (except 0) and the timing of the jump is known. The initial parameter value is assumed to be known exactly so that the estimator is initiated with a covariance of $P(0) = 0$. The covariance of the exogenous noise term is set as 1 ($\sigma = 1.0$) but in the particular realization we simulate, it is kept to as a zero signal up to some time period ($t = 100$). The parameter jump occurs at a certain time ($t = 10$) during that period. We reset the covariance in the estimator to 200 at the time of the jump.

The following cautious controller is also derived by minimizing the one-step ahead cost-to-go function.

$$u(k) = -\frac{\hat{b}(k)}{\hat{b}^2(k) + P(k)} y(k) \tag{26}$$

ADP-based Controller

- Data Generation
  For generation of training data (hyperstate vs. cost-to-go), closed-loop simulations were performed using the following control policies: (1) The CE controller, (2) the cautious controller, and (3) the CE and cautious controllers with dithered inputs. We simulated parameter jumps from the nominal value to $b = \pm 5, \pm 10, \pm 15$. The dither signals were randomly generated from the uniform distribution of $[-0.1\ 0.1]$. Three sets of the dither signals were injected at regular intervals during quiet periods. Each set of dither signals lasted for 4 sample times. Three and five realizations of e were simulated for non-dithered policies and dithered policies, respectively. Three separate realizations of the input dithering were also simulated for each realization of the dithered policies. The parameter was modeled as a constant for this scenario, which means we set $\phi = 1$, $R_w = 0$, and the variance of σ was set as 1. The total number of simulation data obtained was 3849.
- Value Iteration
  In the value iteration step, we solve (14) with $\phi(x(k), u(k)) = y^2(k+1)$ and $\alpha = 0.98$. The expectation operator was evaluated by sampling 50 innovation values ($\epsilon(k)$) for each action candidate $u(k)$ used for the optimization.

  $$\epsilon(k) = y(k+1) - y(k) - \hat{b}(k)u(k) \tag{27}$$

  $\epsilon(k)$ has the following distribution:

  $$\epsilon(k) \sim \mathcal{N}\left(0,\ 1 + u(k)^2 P(k)\right) \tag{28}$$

  The value iteration step converged after 24 runs with

  $$\left\| \frac{J^i(\xi(k)) - J^{i-1}(\xi(k))}{J^{i-1}(\xi(k))} \right\|_\infty < 0.03 \tag{29}$$

  A distance weighted k-nearest neighbor estimator was used for the cost-to-go approximation with $k = 4$, and the quadratic penalty was designed with the parameter choices of $A = 0.87,\ \rho = 0.047,\ \sigma = 0.35,\ J_{\max} = 2500$. To bound the cost-to-go in the off-line iteration steps, the additive penalty term is set as $J_{\max}$ whenever $\tilde{J}(\xi_0) \geq J_{\max}$.
- On-line Performance
  Different parameter jump cases were simulated to compare the ADP policy with the suboptimal control policies. The parameter jump cases that had not been simulated for generating the training data set were also tested. For each case, the total cost over 50 sample times ($\sum_{t=1}^{50} y^2(t)$) were calculated. The total cost averaged over 10 realizations are compared in Table 1. Whereas the average performance of the ADP controller does not vary much with different parameters, the other control policies suffer from bursting or turn off phenomena, leading to poor average performances.

Table 1. Averaged cost over 50 sample times with 10 realizations of e

b	CE	Cautious	Dithered CE	Dithered Cautious	ADP
15	630.5	152.3	63.1	79.8	52.9
-15	936.7	179.8	116.0	93.3	50.7
10	194.6	169.6	99.7	85.5	66.3
-10	184.1	163.9	156.0	64.3	56.7
5	68.4	142.9	41.2	113.7	56.8
-5	72.0	130.5	83.4	64.7	48.0
12	630.1	109.3	60.0	52.3	51.2
-12	401.5	85.8	875.0	51.7	46.6
7	125.9	126.8	60.0	83.8	46.6
-7	345.1	167.4	84.1	65.2	60.7

The performance disparities were observed during the transient period when the parameter jump occurred and exogenous noises entered the system. Fig. 1 shows sample results of the output regulation under the three policies (CE, Cautious, and ADP). At time 10, b jumps from 0.5 to 15 and the covariance is reset to 200. White noise e enters the system at time 15. It shows that the ADP controller injects the probing signal at time

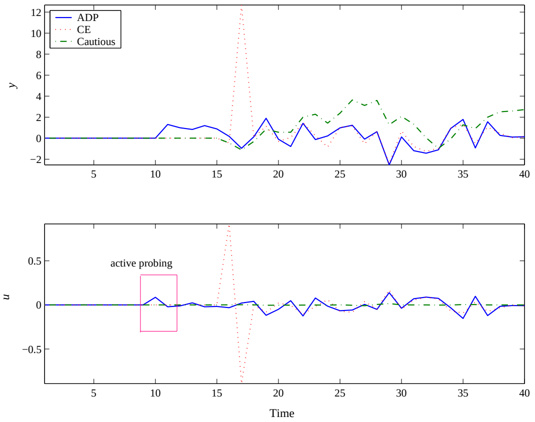

Fig. 1. A sample run of the parameter jump case ($b = 15$): y and u.

10 and achieves the best overall performance of regulation, whereas the passive policies do not move the control actions until time 15, and the performances are degraded either by bursting of the output or by turn off of the control signals.

4.3 Example 2: Time-varying gain

We consider the case with nonzero R_w and σ. To bound the parameter range well, we use $\phi = 0.9987$ and $\gamma = 0.05$, which gives b of unit variance. The performances of the CE and cautious controller deteriorate when the parameter estimate becomes close to zero even with small uncertainty. Hence, we also simulated using CE and cautious controllers with input dither signals sampled from the uniform distribution of $[-15, 15]$ when $|\hat{b}(k)| < 0.2$.

From 25 sets of simulations under the four different control policies(CE, Cautious, CE/Cautious + input dither signals), 3323 data points (hyperstate vs. cost-to-go) were obtained. The value iteration step converged after 21 runs using the same convergence criterion as in the first example, and the quadratic penalty term was designed with $K = 4$, $\sigma = 0.4$, $\rho = 0.1064$, $A = 0.8709$, and $J_{\max} = 2500$. 20 different realizations were performed and the average on-line performances are compared in Table 2.

Table 2. Averaged cost over 20 realizations of e: 500 sample times

	CE	Cautious	Dithered CE	Dithered Cautious	ADP
Avg	136120	894	13046	887	838
Max	387999	1016	327130	994	906
Min	7722	730	6708	690	654

5. CONCLUSIONS

An approximate dynamic programming strategy was suggested to solve a stochastic optimal control problem with a dual objective of identification and control. Starting from different control policies, including several passive and randomized policies, the ADP approach could derive a superior control policy that actively reduces the parameter uncertainty, leading to a robust performance. Sufficient number of simulations with dithered signals would be beneficial in order for the approach to learn a policy with the desired dual feature.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the financial support from the National Science Foundation (CTS–#0301993).

REFERENCES

Anderson, Brian D. O. (1985). Adaptive systems, lack of persistency of excitation and bursting phenomena. *Automatica* **21**(3), 247–258.

Åström, K. J. and A. Helmersson (1986). Dual control of an integrator with unknown gain. *Comp. & Maths. with Appls.* **12A**, 653–662.

Chikkula, Y. and J. H. Lee (2000). Robust adaptive predictive control of nonlinear processes using nonlinear moving average system models. *Industrial & Engineering Chemistry Research* **39**, 2010–2023.

Fel'dbaum, A. A. (1960, 1960, 1961, 1961). Dual control theory. I–IV. *Automation Remote Control* **21, 21, 22, 22**, 874–880, 1453–1464, 1–12, 109–121.

Kaisare, N. S., J. M. Lee and J. H. Lee (2002). Simulation based strategy for nonlinear optimal control: Application to a microbial cell reactor. *International Journal of Robust and Nonlinear Control* **13**(3–4), 347–363.

Lee, J. M. (2004). A Study on Architecture, Algorithms, and Applications of Approximate Dynamic Programming Based Approach to Optimal Control. PhD thesis. Georgia Institute of Technology.

Lindoff, B., J. Holst and B. Wittenmark (1999). Analysis of approximations of dual control. *International Journal of Adaptive Control and Signal Processing* **13**, 593–620.

Wittenmark, B. (2002). Adaptive dual control. In: *Control Systems, Robotics and Automation, Encyclopedia of Life Support Systems (EOLSS), Developed under the auspices of the UNESCO* (H. Unbehauen, Ed.). Eolss Publishers. Oxford, UK. Paper 6.43.15.6.