DUAL ADAPTIVE CONTROL OF A FED-BATCH BIOREACTOR BASED ON APPROXIMATE DYNAMIC PROGRAMMING

Ha-Eun Byun[1], Boeun Kim[1] and Jay H. Lee[1*]
[1] Department of Chemical and Biomolecular Engineering, Korea Advanced Institute of Science and Technology (KAIST), 291, Daehak-ro, Yuseong-gu, Daejeon, 34141, Republic of Korea

Abstract Overview

Dual control maintains an optimal balance between control actions (exploitation) and probing actions (exploration), leading to improved process performance by actively reducing system uncertainty. The optimal solution of dual control problem can be found by stochastic dynamic programming but it is computationally intractable in most practical cases. In this study, a tailored approximate dynamic programming (ADP) method can be used. This paper addresses the dual control problem of a nonstationary batch process maximizing a given end objective while satisfying path constraints in the presence of stochastic system uncertainty. Performance of the ADP-based approach is tested on a fed-batch penicillin fermentation system with two uncertain parameters.

Keywords

Dual adaptive control, Approximate dynamic programming, Fed-batch fermentation

Introduction

The control of dynamic systems in the presence of uncertain parameters and constraints is of great interest in industrial chemical and biological processes. When measurements available, the general approach is to estimate the unknown parameters with measurements and use the parameter estimates in a deterministic control framework, known as certainty equivalence adaptive control. To obtain informative data about uncertainties, exploratory inputs to excite the system, i.e., probing inputs, are usually required. However, the certainty equivalence approach does not take into account the effect of inputs on the future uncertainty and thus the resulting probing actions are only passive or accidental (Mesbah, 2018).

The probing inputs may decrease the performance in short-term (when it conflicts with the control objective), but the improved knowledge about uncertainty can result in better control performance in the future. Thus, an optimal balancing between maximizing the information about the uncertain system and optimizing the process performance is needed, and this is called the dual control problem.

The optimal solution of the dual control problem can be found by stochastic dynamic programming (Feldbaum, 1960) but it is computationally intractable in most practical cases, particularly for systems with continuous state space. To solve the dual control problem approximately, various approaches have been suggested. One such approach is via approximate dynamic programming (ADP), also known as reinforcement learning. For the purpose of process control application with continuous state and action spaces, Lee and Lee (2009) proposed an ADP based approach for stochastic optimal control, in which dynamic programming is solved on a restricted space of hyper-state sampled through stochastic closed-loop simulations performed with suboptimal control policies. The proposed approach was applied to tracking control of ARX SISO linear system with

* To whom all correspondence should be addressed

two uncertain parameters and improved performance with dual feature was demonstrated..

This study aims to tailor the ADP method for the dual control of nonlinear and nonstationary batch processes. A batch process has a relatively short time horizon compared to a continuous process, so in the presence of uncertain parameters with poor prior knowledge, it is necessary to rapidly find good estimates before reaching the end of the batch. In this study, rather than tracing given trajectories, the ADP-based dual control problem is formulated to optimize the batch process's economic performance while meeting path constraints under uncertainty. A case study of fed-batch penicillin fermentation with two uncertain parameters is used to show improved performance of the ADP-based dual controller.

ADP-based Dual Adaptive Control

Consider the discrete-time stochastic system

$$x_{k+1} = f(x_k, u_k, \theta_k, e_k), \quad k \in \mathbb{N} \tag{1}$$

where x_k is a state vector, u_k is a control input, θ_k is a vector of unknown parameters of the model, and e_k represents exogenous noises. The state x_k is assumed to be measured and the model structure f is assumed to be known. The aim of control is to maximize the performance index represented as follow:

$$\max_{u_0,\cdots,u_{t_f-1}} E\left[\sum_{t=0}^{t_f-1} \phi(x_t, u_t) + \bar{\phi}_T(x_{t_f})\right] \tag{2}$$

subject to

$$g_i(x_t, u_t) \leq 0, \quad \text{for } i = 1, \cdots, m \tag{3}$$

where ϕ and $\bar{\phi}_T$ are stage-wise reward (profit) and terminal reward (profit), respectively. The expectation E is taken over the distribution of θ and e. g_i denotes the path constraints which should be satisfied. The inputs are to be decided based on the measured state information, and so the problem is to find the optimal control policy, which is a map between state x_k and input u_k.

In the framework of dynamic programming, the optimal 'profit-to-go' function at time k can be represented as

$$J_k^*(\xi_k) = \max_{\boldsymbol{u}} E\left[\sum_{t=k}^{t_f-1} \phi(x_t, u_t) + \bar{\phi}_T\left(x_{t_f}\right) \mid \xi_k\right], \tag{4}$$

where hyper-state ξ_k is an extended random state including the information about uncertainty, i.e., parameter estimates and their variances, as well as x_k. J_k^* maps the hyper-state to the profit-to-go value under the optimal control, satisfying the following Bellman's optimality equation:

$$J_k^*(\xi_k) = \max_{u_k} E[\phi(x_k, u_k) + J_{k+1}^*(\xi_{k+1})|\xi_k]. \tag{5}$$

Once J_k^* is determined, the optimal control policy can be derived by solving

$$u_k = \pi^*(\xi_k) = arg\max_{u_k} E[\phi(x_k, u_k) + J_{k+1}^*(\xi_{k+1})|\xi_k]. \tag{6}$$

For each evaluation of a candidate u_k, the expectation needs to be calculated, which involves the integration of the successor hyper-state ξ_{k+1} for its all possible range. To solve the Bellman equation numerically, the value iteration or policy iteration is performed after discretization of the hyper-state space. However, this is computationally intractable in most practical cases, especially when the hyper-state space is continuous.

The ADP based approach proposed by Lee and Lee (2009) circumvents the curse-of-dimensionality of the traditional DP approach by solving the DP only for a restricted space of the hyper-state, sampled from Monte Carlo simulations of the closed-loop system with given suboptimal control policies. The same idea is employed in this study and tailored for the nonstationary batch process with path constraints. Construction and improvement of the profit-to-go approximation proceed as follows and note that these steps are performed off-line and the converged profit-to-go approximator is used on-line.

1. Perform Monte-Carlo runs of the closed-loop system with known suboptimal control policies, e.g., PID, MPC. It is recommended to simulate several policies with different characteristics in order to cover a broad range of potential operating space.
2. For each state visited during the simulation runs, calculate the profit-to-go J_k^0 using the simulation data according to

   $$J_k^0(\xi_k) = \sum_{t=k}^{t_f-1} \phi(x_t, u_t) + \bar{\phi}_T(x_{t_f})|\xi_k. \tag{7}$$

   Here the satisfaction of path constraint is considered as the stage-wise reward:

   $$\phi(x_k,\, u_k) \Leftarrow \phi(x_k,\, u_k) - \lambda \cdot \sum_{i \in G_{cv}} g_i(x_k, u_k). \tag{8}$$

   G_{cv} is a set of indexes of constraints violated. λ is a weighting parameter for the penalty for constraint violation.
3. Construct an initial function approximator $\tilde{J}$ using calculated profit-to-go values for the sampled points to approximate the profit-to-go with respect to the continuous hyper-state. In this work, a local averager, i.e., a modified *k*-nearest neighbor (*k*NN), suggested in (Lee et al., 2006) is used as the approximator. Considering the nonstationary, finite-time characteristics of the batch process, the value function approximation is performed for each time step k as below:

   $$\tilde{J}_k(\xi_{k,0}) = \sum_{i=1}^{N} w_i J_k(\xi_{k,i}) \tag{9}$$

   with

   $$w_i = \frac{1/d_i}{\sum_N 1/d_i}, \tag{10}$$

   where $\xi_{k,0}$ is a query point at time k, and N is the number of neighboring points in the data set. Each neighboring point is weighted inversely proportional to the Euclidean distance. To avoid excessive extrapolation, a quadratic penalty term based on the local density is added.

   $$\tilde{J}_k(\xi_{k,0}) \Leftarrow \tilde{J}_k(\xi_{k,0}) - J_{k,penalty}(\xi_{k,0}), \tag{11}$$

$$J_{k,penalty}(\xi_{k,0}) = A_k \cdot H\left(\frac{1}{f_\Omega(\xi_{k,0})} - \rho\right) \cdot \left[\frac{\frac{1}{f_\Omega(\xi_{k,0})} - \rho}{\rho}\right]^2. \quad (12)$$

Detailed description of the penalty term can be found in (Lee et al., 2006). Note that in this study the penalty term is set as $J_{k,max} - J_{k,min}$ whenever $J_{k,penalty}(\xi_{k,0}) \geq J_{k,max} - J_{k,min}$ to bound the profit-to-go in the iteration steps.

4. Improve the profit-to-go approximation through value iteration

$$J_k^{i+1}(\xi_k) = \max_{u_k} E\left[\phi(x_k, u_k) + \tilde{J}_{k+1}^i(\xi_{k+1}) | \xi_k\right], \quad (13)$$

where superscript i denotes ith iteration step and J_k^{i+1} is calculated for all the sampled states ξ_k from simulations. To evaluate the expectation, Monte Carlo simulation is performed and the average of data ensemble is used. The control input space is discretized and the expectation is evaluated for each candidate input. The iteration is repeated until $\left\|J_k^{i+1}(\xi) - J_k^i(\xi)\right\|_\infty$ becomes negligibly small for all k.

Once the profit-to-go values converge, it can be used on-line as a control policy by solving

$$u_k = \pi^*(\xi_k) = arg \max_{u_k} E\left[\phi(x_k, u_k) + \tilde{J}_{k+1}^{N_c}(\xi_{k+1}) | \xi_k\right] (14)$$

at each sampling time. This single-stage optimization requires much less on-line computation than the original multi-stage optimization problem.

Case Study: Fed-batch Penicillin Fermentation

We illustrate the ADP-based dual control of the batch process with an example of penicillin fermentation process. The system can be described by

$$x_k = f_k(\theta_k, x_{k-1}, u_{k-1}) + e_k, \quad e_k \sim N(0, R_e) \quad (15)$$
$$y_k = h_k(x_k) + v_k, \quad v_k \sim N(0, R_v) \quad (16)$$
$$\theta_k = \theta_{k-1} + w_k, \quad w_k \sim N(0, R_w) \quad (17)$$

where e_k is exogenous noises and θ_k is a set of uncertain parameters. The system model f_k is represented by the following nonlinear differential equations (Srinivasan et al., 2002):

$$\dot{X} = \mu(S)X - \frac{u}{V}X, \quad (17)$$
$$\dot{S} = -\frac{\mu(S)X}{Y_x} - \frac{vX}{Y_p} + \frac{u}{V}(S_{in} - S), \quad (18)$$
$$\dot{P} = vX - \frac{u}{V}P, \quad (19)$$
$$\dot{V} = u, \quad (20)$$

where $\mu(S) = \mu_m S/(K_m + S + S^2/K_i)$. X, S and P are the concentrations of biomass, substrate and penicillin, respectively, and V is the liquid volume. The control input u is the feeding rate of substrate. Since the major sources of uncertainty in bioprocess are feedstock variability and cell variability, the inlet sugar concentration $S_{in} \sim N(200, 25^2)$ and the maximum growth rate $\mu_m \sim [0.01, 0.03]$ are assumed to be uncertain but constant during the whole batch in this study. For the sake of simplicity, the perfect measurement of the physical states is assumed ($R_v = 0$) and thus $y_k = x_k = [X_k\ S_k\ P_k\ V_k]^T$ and $\theta_k = [\mu_{m,k}\ S_{in,k}]^T$.

In this study, the extended Kalman filter is employed to estimate θ_k from the measurements. We set the initial covariance matrix P_0 as diag$\{0.01^2, 25^2\}$, R_e as $0.04^2 I_{4\times4}$, and R_w as diag$\{0, 0\}$. The hyper-state of the process is $\xi_k = \left[X_k, S_k, P_k, V_k, \hat{Y}_{x_{k+1|k}}, \hat{S}_{in_{k+1|k}}, P^{11}{}_{k+1|k}, P^{22}{}_{k+1|k}, P^{12}{}_{k+1|k}\right]^T$.

The control goal is to maximize P at the fixed final time $t_f = 150$ h under the uncertainties while satisfying an upper bound constraint on X, i.e., $X \leq X_{max} = 3.7\ g/L$. The control input u_k is bounded in $[0, 1]$. Thus, the profit-to-go values for the sampled points at each time step k can be calculated as follows:

$$J_k^0(\xi_k) = \sum_{i=k}^{t_f-1} \phi(x_i, u_i) + P|_{t_f}, \quad (21)$$

$$\phi(x_i, u_i) = \begin{cases} \lambda(X_{max} - X_i) & if\ X_i > X_{max} \\ 0 & otherwise \end{cases}. \quad (22)$$

For the data generation, the stochastic closed-loop simulations are performed with the suboptimal policies, i.e., a shrinking-horizon adaptive MPC with/without dithered inputs. The dithered inputs are uniformly sampled from $[-0.02, 0.02]$ and 50 runs of simulations with each suboptimal policy are conducted. With the sampled data, the value function approximator is constructed and the value iteration is performed to improve the approximation.

The performances of the certainty equivalence control, i.e., adaptive MPC, and the ADP-based dual control will be compared further.

Conclusion

This study proposes a dual control scheme for batch process based on approximate dynamic programming. The control policy from the proposed ADP-based dual control is expected to improve the batch performances compared to the passive adaptive control policy by actively reducing the parameter uncertainty within a given batch time.

References

Brooke, A., Kendrick, D., Meeraus, A. A. (1992). GAMS- A User's Guide (Release 2.25). *The Scientific Press*. San Francisco, CA.

Feldbaum, A. A. (1960). Dual control theory. I. Avtomatika i Telemekhanika, 21(9), 1240-1249.

Lee, J. M., Kaisare, N. S., & Lee, J. H. (2006). Choice of approximator and design of penalty function for an approximate dynamic programming based control approach. Journal of process control, 16(2), 135-156.

Lee, J. M., & Lee, J. H. (2009). An approximate dynamic programming based approach to dual adaptive control. Journal of process control, 19(5), 859-864.

Mesbah, A. (2018). Stochastic model predictive control with active uncertainty learning: A survey on dual control. Annual Reviews in Control, 45, 107-117.

Srinivasan, B., Bonvin, D., Visser, E., & Palanki, S. (2003). Dynamic optimization of batch processes: II. Role of measurements in handling uncertainty. Computers & chemical engineering, 27(1), 27-44.