2008 American Control Conference
Westin Seattle Hotel, Seattle, Washington, USA
June 11-13, 2008

A Reinforcement Learning-Based Scheme for Adaptive Optimal Control of Linear Stochastic Systems

Wee Chin Wong and Jay H. Lee

Abstract— **Reinforcement learning where decision-making agents learn optimal policies through environmental interactions is an attractive paradigm for direct, adaptive controller design. However, results for systems with continuous variables are rare. Here, we generalize a previous work on deterministic linear systems, to stochastic ones, since uncertainty is almost always present and needs to be accounted for to ensure good closed-loop performance. In this work, we present convergence results and also show an example suggesting automatic controller order-reduction. We also highlight key differences between the algorithms for deterministic and stochastic systems.**

I. INTRODUCTION

Safety and economic considerations make closed-loop system identification desirable. In such situations, it may be advantageous to consider the system identification and controller design in an integrated manner. The observation that 'reinforcement learning (RL)' is a form of direct adaptive control has been made by several researchers in the field of Artificial Intelligence [2]. In RL, an autonomous agent is trained to learn an optimal control policy through interactions with the environment. This is achieved by translating feedbacks from the system of interest intelligently so that desirable actions are reinforced and undesirable ones are penalized. On the theoretical front, RL is also shown to have a strong connection to the classical optimization technique of dynamic programming.

The focus of this work is a RL method closely related to 'Q-learning' [3] for the purpose of model-free, closed-loop controller design. Typically, system identification is first performed to yield a model, based on which a controller is built. The proposed approach combines the two steps into a single task of identifying the so-called Q-function. We study this method in the context of optimal, quadratic control of linear, stochastic systems. The latter constitute a simple but descriptive and often-used model structure to describe both deterministic and stochastic effects.

Technically, RL is a means of solving infinite-horizon (and possibly discounted) stochastic optimal control problems. The main mathematical construct, given that some policy π^L is to be adhered to, is its corresponding value function, $J^L(x)$. A policy maps a state (x) to a control action (u). $J^L(x)$ represents the infinite horizon discounted cost starting from a particular state assuming the policy π^L is followed throughout the horizon. In the context of this paper, the policy is linear and gain matrix L represents π^L, i.e., $\pi^L : u = -Lx$. Denoting the mapping corresponding to the optimal policy as L_*, $J^{L_*}(x)$ is the optimal value function. An optimal action for any state x_t can be found by considering the tradeoff between the immediate benefit and the optimal value-function of the next state $(J^{L_*}(x_{t+1}))$. However, this value-function formalism does require the system's model to be known. For the rest of the paper, we omit the time subscript and/ or the argument x_t (or its equivalent) wherever it is contextually appropriate.

The $Q^L(x,u)$ function [3], mapping a state-action (x,u) pair to a real value, is more suited for the purpose of direct-adaptive control. Simply put, if u is that prescribed by π^L, then Q^L and J^L are the same. The Q function is the value of x, except that the initial action is arbitrary. Intuitively, one can expect that for the optimal policy π^{L_*}, minimizing Q^{L_*} over u gives the same effect. This means Q^{L_*} yields the optimal policy directly. Hence, the identification and controller design tasks reduce to the identification of Q^{L_*}.

Unsurprisingly, most RL research has treated problems with an artificial-intelligence bent (e.g. maze-negotiation, obstacle avoidance, game-playing and the like [4]), where the state and action spaces are finite sets. For such finite-state systems, the value-functions are typically represented as look-up tables. Convergence proofs for the corresponding RL algorithms have been well established.

The application of RL for the purpose of optimal control has been limited since dynamical systems of interest are typically described in terms of continuous variables. Naive applications of finite-state RL algorithms are subject to the 'curse-of-dimensionality', a term coined by Bellman [5]. Consequently, few analytical results have been published. The work of Bradtke et. al. [1], dealing with Linear Quadratic Regulation (LQR), is a notable exception, and represents one of the pioneering efforts in taking RL methods to the realm of control. However, the proposed algorithm is limited to deterministic systems. Landelius [6] explored this algorithm in the form of other RL variants such as Heuristic Dynamic Programming, Dual Heuristic Dynamic Programming, and Action Dependent Heuristic Dynamic Programming, but the work was still limited to the noise-free case. More recently, Al-Tamimi et. al. [7] applied a similar Q-learning approach for the purpose of $\mathcal{H}_\infty$ control.

In the context of comparing indirect and direct approaches for controller synthesis, [8] and [9] explored the impact of noise (vs. external input dithering) on the convergence of the algorithm proposed by [1]. Those studies assumed full state-feedback. The conclusion drawn from the study

Wee Chin Wong and Jay H. Lee are with the Department of Chemical & Biomolecular Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA. {weechin.wong, jay.lee}@chbe.gatech.edu

978-1-4244-2079-7/08/$25.00 ©2008 AACC.

is that the Q-learning based direct approach was inferior but this conclusion was based on the Q-learning algorithm developed for noise-free systems. There remains the question of whether one can modify the algorithm to deal with noisy feedback in a systematic way.

With this in mind, an extension of Bradtke et. al.'s (1994) work to situations where the system can be described by a state-space model with stochastic noise inputs, or equivalently, an Auto-Regressive-Moving-Average model with eXogenous inputs (ARMAX), is proposed. In the control, econometrics and other communities, ARMAX models and their state-space analogs are regarded as a flexible model class [10] describing both deterministic and stochastic effects (such as disturbances).

To the best of our knowledge, in the context of RL, this work is the first extension accounting explicitly for noise inputs (modeled as Gaussian stochastic processes) and can be regarded as the Linear Quadratic Gaussian (LQG) analog of Bradtke et. al.'s LQR work. This work can also be viewed as another means of iterative LQG controller design, the benefits of which have been propounded by [11]. An important observation is that judicious use of optimal state-estimates (or equivalently, 1-step-ahead output predictors) is required for optimal closed-loop performance. This is important since users of RL algorithms [8], [9] for direct adaptive control often employ the deterministic version of the algorithm, whereas direct state feedback, even if it were available, is not desired in the noisy case.

This paper proposes a scheme whereby these estimates/ predictors may be obtained in a closed-loop setting. Without having to resort to full-fledged system-identification, the mechanics of closed-loop subspace identification methods [12] offer a means by which this can be achieved. Finally, we show simulation results indicating convergence for the case where a low-order controller is employed. This skips the model-reduction step typical of indirect methods. Since controller reduction is oftentimes viewed as more difficult than model-reduction [13], this result is useful for low order controller design.

II. Problem Statement

We consider controllable and observable systems of the innovation form, as in (1).

$$\begin{aligned} x_{t+1} &= Ax_t + Bu_t + AKe_t \\ y_t &= Cx_t + e_t \end{aligned} \tag{1}$$

where, $x_t \in \mathbb{R}^{n_x}$, $y_t \in \mathbb{R}^{n_y}$ and $u_t \in \mathbb{R}^{n_u}$ represent the state, (measurable) input and output signals respectively. $\{e_t\}$ is a sequence of zero-mean white, Gaussian noise. Matrices (A, B, C) have their usual meanings whilst K is the steady-state Kalman gain. The objective is to find an optimal policy, π^{L_*} attaining the following quantity (2)

$$J^{L_*}(\mathcal{I}_t^l) = \min \mathbb{E}_{(\cdot|\mathcal{I}_t^l)} \sum_{i=t}^{\infty} \gamma^{i-t} r_i, \quad \gamma \in (0, 1) \tag{2}$$

subject to (1), where $\mathcal{I}_t^l \triangleq \{y_0, \ldots, y_{t-l}, u_0, \ldots, u_{t-1}\}$ denotes an information vector serving as a proxy for the unmeasurable state, x_t. $l \in \mathbb{N}, l \geq 0$ indicates the amount of information available to the controller. Also, $r_i \triangleq x_i' R_x x_i + u_i' R_u u_i, R_u > 0$ reflect unwanted deviations of the state vector from the origin as well as excessive actuator movement, as defined by Euclidean norms with user-defined weighting matrices R_x and R_u. For our purpose, we assume $R_x = C'C \geq 0$ so that deviations of the measurable quantities of interest y, from the origin are penalized instead. It is well established [14] that the optimal policy is the Linear Quadratic Regulator (i.e., the linear map (4)) applied to the state estimate provided by the Kalman filter (5).

$$\hat{x}_{t|t-l} \triangleq \mathbb{E}\{x_t|\mathcal{I}_t^l\} \tag{3}$$

$$u_t = -\gamma(R_u + \gamma B^{'} S^{L_*} B)^{-1} B' S^{L_*} A\hat{x}_{t|t-l} \tag{4}$$

$$\hat{x}_{t+l|t} = A\hat{x}_{t+l-1|t-1} + Bu_{t+l-1} + \underbrace{A^l K}_{\breve{K}}(y_t - C\hat{x}_{t|t-1}) \tag{5}$$

$$\begin{aligned} S^{L_*} = \; & R_x + \gamma A^{'} S^{L_*} A - \\ & \gamma^2 A' S^{L_*} B(R_u + \gamma B^{'} S^{L_*} B)^{-1} B^{'} S^{L_*} A \end{aligned} \tag{6}$$

Clearly, this approach requires model knowledge. With this, $\mathcal{I}_t^l$ can be summarized by the pair: $\{\hat{x}_{t|t-l}, P_{t|t-l} \triangleq \mathbb{E}_{(\cdot|\mathcal{I}_t^l)}(x_t - \hat{x}_{t|t-l})(x_t - \hat{x}_{t|t-l})'\}$. It is noted that $l = 0$ corresponds to filtered estimates and $l = 1$ to one-step-ahead predicted estimates. Although these values of l are typically employed, allowing $l \geq 0$ imparts generality.

III. Role of the Q-function in Model-Free Control

Analogous to J^{L_*} is the value-function (7) corresponding to an arbitrary policy π^L, i.e. $u_t = -L\hat{x}_{t|t-l}, \forall t$. By splitting the infinite series (2), allowing u_t to be arbitrary and the subsequent control actions to be prescribed by π^L, we have the recursive form (8) and also (9) by definition, as $J^L(\mathcal{I}_t^l) = Q^L(\mathcal{I}_t^l, -L\hat{x})$ for all admissible π^L.

$$J^L(\mathcal{I}_t^l) = \mathbb{E}_{(\cdot|\mathcal{I}_t^l)} \sum_{i=t}^{\infty} \gamma^{t-i} r_i \tag{7}$$

$$Q^L(\mathcal{I}_t^l, u_t) \triangleq \mathbb{E}_{(\cdot|\mathcal{I}_t^l, u_t)}\{r_t + \gamma J^L(\mathcal{I}_{t+1})\} \tag{8}$$

$$Q^L(\mathcal{I}_t^l, u_t) = \mathbb{E}_{(\cdot|\mathcal{I}_t^l, u_t)}\{r_t + \gamma Q^L(\mathcal{I}_{t+1}^l, -L\hat{x}_{t+1|t+1-l})\} \tag{9}$$

Furthermore, since $J^{L_*}(\mathcal{I}_t^l) \leq J^L(\mathcal{I}_t^l)$ for any π^L, it must be the case that $\min_u Q^{L_*}(\mathcal{I}_t^l, u) = J^{L_*}(\mathcal{I}_t^l)$. In other words, knowledge of the Q^{L_*} gives us the optimal policy without requiring the system model. In the next sub-section, we show how, for a particular π^L, Q^L can be obtained and how π^L can be subsequently improved such that Q^{L_*} is eventually obtained. For clarity, we temporarily assume that the Kalman state-estimates and the effective Kalman gain $(\breve{K})$ are available.

A. Solution Methodology via Policy Iteration: Optimal State-Estimates Available

Similar to the work of Bradtke et. al. [1], the proposed algorithm is divided into episodes, each of which consists of i) policy evaluation and ii) policy improvement steps.

For a given episode k, the system operates in closed loop for a certain duration T, under some stabilizing policy π^{L_k}. At the end of each episode, the corresponding Q^{L_k} function is evaluated. Subsequently, a new policy whose performance is no worse than before is obtained by means of policy improvement. The combination of these steps is known as 'policy iteration' in the RL community [4] and is described below.

1) Policy Evaluation.: For any π^L, it is well-known [14] that $J^L(\mathcal{I}_t^l)$ possesses the following analytic structure : $\hat{x}_{t|t-l}^{'}S^L\hat{x}_{t|t-l} + c_J, c_J \in \mathbb{R}$, where $S^L = \gamma^2(A - BL)^{'}S^L(A - BL) + L'R_uL + R_x$ is the solution to a Lyapunov equation, and c_J some constant. Similarly, by expanding (8), rearranging terms and denoting $z_t \triangleq [\hat{x}_{t|t-l}, u_t]^{'} \in \mathbb{R}^{n_z \triangleq n_x + n_u}$, we have

$$\begin{aligned} Q^L(\mathcal{I}_t^l, u) &= z^{'}H^Lz + c_Q, c_Q \in \mathbb{R} && (10)\\ &\triangleq \bar{z}'\theta^L && (11)\end{aligned}$$

$$H^L = \begin{pmatrix} R_x + \gamma A^{'}S^LA & \gamma A^{'}S^LB \\ \gamma B^{'}S^LA & R_u + \gamma B^{'}S^LB \end{pmatrix} \qquad (12)$$

Here, via the $\bar{(\cdot)}$ operator, the quadratic form is converted into an equivalent inner product. θ^L constitutes a unique $\mathbb{R}^{0.5n_z(n_z+1)}$ vector representation of $H^L \in \mathbb{R}^{n_z \times n_z}$, and vice versa. Looking at (9), and letting 'tr' denote the trace operator, we have that

$$\begin{aligned} &\mathbb{E}_{(\cdot|\mathcal{I}_t^l,u_t)}\{r_t\}\\ &= \text{tr}[R_x\mathbb{E}_{(\cdot|\mathcal{I}_t^l,u_t)}(x_tx_t')] + u_t'R_uu_t\\ &= \text{tr}[R_x\mathbb{E}_{(\cdot|\mathcal{I}_t^l,u_t)}\left(\{x_t - \hat{x}_{t|t-l} + \hat{x}_{t|t-l}\}\{\cdot\}'\right)] + u_t'R_uu_t\\ &= \text{tr}[R_x\mathbb{E}_{(\cdot|\mathcal{I}_t^l,u_t)}\left(\{\hat{x}_{t|t-l}\}\{\cdot\}'\right)] + \text{tr}[R_xP_{t|t-l}] + u_t'R_uu_t\\ &= \hat{x}_{t|t-l}'R_x\hat{x}_{t|t-l} + u_t'R_uu_t + c_1, \quad c_1 \in \mathbb{R} \qquad (13)\end{aligned}$$

The second-to-last equality is due to the fact that $(x_t - \hat{x}_{t|t-l})$ is zero-mean. Here, c_1 is some real whose value is not of concern. Also, from (5), $\mathbb{E}_{(\cdot|\mathcal{I}_t^l,u_t)}\{\hat{x}_{t+1|t+1-l}\} = A\hat{x}_{t|t-l} + Bu_t = \hat{x}_{t+1|t+1-l} - \breve{K}(y_{t+1-l} - C\hat{x}_{t+1-l|t-l}) \triangleq \tilde{x}_{t+1}$, and $\tilde{x}_{t+1}$ is orthogonal to $(\hat{x}_{t+1|t+1-l} - \tilde{x}_{t+1})$. As in (13), we have

$$\mathbb{E}_{(\cdot|\mathcal{I}_t^l,u_t)}\{Q^L\left(\mathcal{I}_{t+1}^l, -L\hat{x}_{t+1|t+1-l}\right)\} = \bar{\tilde{z}}_{t+1}^{'}\theta^L + c_2 \qquad (14)$$

with $\tilde{z}_{t+1} \triangleq [\tilde{x}_{t+1}, -L\tilde{x}_{t+1}]'$ and c_2 another real number. Re-arranging (9) yields,

$$\begin{aligned} \mathbb{E}_{(\cdot|\mathcal{I}_t^l,u_t)}\{r_t\} &= \left(\bar{z}_t - \gamma\bar{\tilde{z}}_{t+1}\right)^{'}\theta^L + c\\ &\triangleq \phi_t^{'}\theta^L + c \qquad (15)\end{aligned}$$

$c \in \mathbb{R}$ is a constant that accounts for c_Q, c_1 and c_2. c is inconsequential and will be dropped for the rest of this article. $\tilde{z}$ is crucial in differentiating the stochastic and the deterministic algorithms. In the algorithm presented by Bradtke et. al. [1], one replaces z_t and $\tilde{z}_{t+1}$ in the above with $[x_t, u_t]'$ and $[x_{t+1}, -L(x_{t+1})]'$ respectively. In the stochastic case, however, direct use of state measurements to estimate θ^L, even if they were available, would not be appropriate.

At this stage, given the form of (15), it is apparent that learning the corresponding Q^L, that is policy evaluation, for any π^L is a linear regression problem and can be solved via (possibly recursive) least squares (see Appendix), with the sequence $\left\{\mathbb{E}_{(\cdot|\mathcal{I}_t^l,u_t)}(r_t)\right\}_{t=0}^T$ collected together with regressors $\{\phi_t\}_{t=0}^T$, for T larger than $\frac{n_z(n_z+1)+1}{2}$.

Specifically, if Q-learning were to be implemented online, then for some π^L and $\forall t$, we need to implement $u_t = -L\hat{x}_{t|t-1} + d_t$, where d_t is an external dithering signal, which is needed to ensure that a unique least-squares solution exists.

In general, d_t can be chosen as a random perturbation signal. For special cases where there is no requirement that controller design be done online, it is noted that $d_t = 0$ does not destroy convergence to the optimal policy, provided that u_t is the output of some pre-existing, non-linear controller such that ϕ_t is persistently exciting [16]. Policy improvement is described next.

2) Policy Improvement.: For arbitrary policies π^{L_k} and $\pi^{L_{k+1}}$, if $Q^{L_k}(\cdot, L_{k+1}(\cdot)) \leq J^{L_k}(\cdot) = Q^{L_k}(\cdot, L_k(\cdot))$, then $J^{L_{k+1}}(\cdot) \leq J^{L_k}(\cdot)$. Therefore one way of achieving policy improvement is by minimizing $Q_k^L(\cdot, u)$ with respect to the second argument. Given the analytical form of Q^L, and by denoting k as an episode index, we have (16)

$$\begin{aligned} -L_{k+1}\hat{x}_{t|t-l} &= \arg\min_u Q^{L_k}(\cdot, u)\\ &= -(H_{22}^{L_k})^{-1}H_{21}^{L_k}\hat{x}_{t|t-l} \qquad (16)\end{aligned}$$

Here $H_{ij}^{L_k}$ is the i,j-th block sub-matrix of H_k^L for the k-th episode. Summarizing, we have

$$J^{L_*} \leq \ldots \leq J^{L_{k+1}}(\cdot) \leq \min_u Q^{L_k}(\cdot, u) \leq J^{L_k}(\cdot) \leq \ldots \qquad (17)$$

If persistent excitation conditions are satisfied, and if the system (A, B) is controllable, the above steps would converge to the optimal policy. The proof follows that of the deterministic case presented by [1].

Upon convergence $(k \rightarrow \infty)$, we verify optimality by noting that, for arbitrary x, $\min_u x'\left(Q^{L_*}(\cdot, u)\right)x = x'\left(H_{11}^{L_*} - H_{12}^{L_*}(H_{22}^{L_*})^{-1}H_{21}^{L_*}\right)x$. The matrix in parentheses coincides with the solution to the control Riccati equation, (6).

B. Remarks

At this juncture, the need for state-estimates $(z, \tilde{z})$ is clear. Section IV and beyond discusses the means of achieving this. For the purpose of illustration, the proposed algorithm is demonstrated on a simple problem. This highlights the

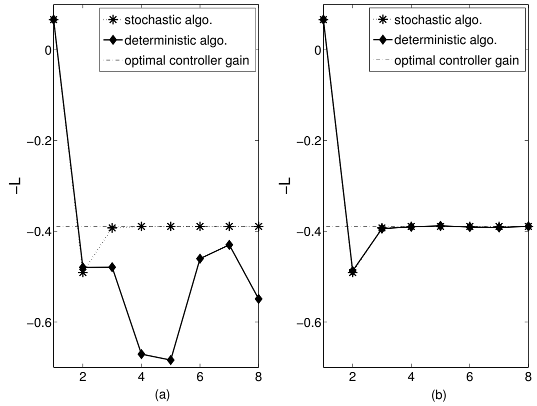

Fig. 1. (a) Learnt controller gains vs. episodes for $\frac{\mathbb{E}(d_t^2)}{\mathbb{E}(e_t^2)} = 0.01$; (b) Learnt controller gains vs. episodes for $\frac{\mathbb{E}(d_t^2)}{\mathbb{E}(e_t^2)} = 1000$

differences between the deterministic and stochastic versions of the RL-based algorithms for direct adaptive control. A second reason, as mentioned earlier, is to clarify the controller synthesis studies [8], [9], that were done to compare indirect and direct Q-learning approaches in the presence of system noise. The latter was based on the method developed for noise-free systems.

A 1st-order scenario where $A = 0.8; B = 1.5; C = 1; K = 1.2; \mathbb{E}[e_t^2] = 10; R_x = R_u = l = 1; g = 0.995$ serves as an illustration. In this case, $L_* = 0.389$. Convergence results using the deterministic algorithm (assuming full state-feedback) against those computed via the algorithm developed in this paper (assuming known $z, \tilde{z}$) are compared by considering various dither-to-noise ratios. Fig. 1 summarizes findings for a typical realization.

If the levels of dithering (simulated as white, zero-mean Gaussian noise) significantly exceed that of the system noise (as in Fig. 1b), then the true system responds as if it were driven only by exogenous inputs. Naturally, the deterministic algorithm performs well (in terms of convergence to the optimal policy) in this case. However, when the noise is significant compared to the dither, as in Fig. 1a, the deterministic algorithm shows very erratic behavior while the stochastic one exhibits smooth convergence. This means that based on the deterministic version of the algorithm, one might draw false conclusions regarding the levels of dither required for convergence to the optimal policy, as was done by [8], [9].

IV. Obtaining State Estimates from HOARX State-Space Realizations

Obtaining the state estimates (or equivalent quantities) can be achieved in a closed-loop setting by means of a High-Order Auto-Regressive with eXogenous input (HOARX) model. This is commonly employed in closed-loop subspace identification [12] techniques as a pre-estimation step. The main idea, provided that $\lambda(\mathbb{A} \triangleq A - AKC) < 1$, is to approximate (1) as such:

$$y_t \approx \sum_{i=1}^{q} \underbrace{C\mathbb{A}^{i-1}AK}_{a_i} y_{t-i} + \sum_{i=1}^{q} \underbrace{C\mathbb{A}^{i-1}B}_{b_i} u_{t-i} + e_t \qquad (18)$$

For simplicity of exposition, we consider Single-Input-Single-Output (SISO) systems, while noting that an extension to its Multiple-Input-Multiple-Output counterparts is straightforward. This approximation can be made arbitrarily accurate for sufficiently large values of $q \in \mathbb{N}$. Since the regressors are uncorrelated with e_t, unbiased estimates $\{\hat{a}_i, \hat{b}_i\}_{i=1}^{q}$ can be obtained via a least squares method. A persistently exciting d_t ensures that the regressor matrix used in the obtaining the HOARX parameters is of full column rank. Note that these parameters estimated are used to calculate state estimates needed for the Q-learning algorithm but are not used for the design of the controller.

Unlike subspace methods where further steps are required to find a particular (A, B, C, K)-tuple, we employ a particular state-space realization with states that are fully measurable, and simultaneously allows a convenient means for controller order-reduction.

A. A Useful State-Space Realization

Consider a direct-realization of the HOARX approximation (18):

$$\begin{pmatrix} y_t \\ y_{t-1} \\ \vdots \\ y_{t-s+1} \\ u_{t-1} \\ u_{t-2} \\ \vdots \\ u_{t-s+1} \end{pmatrix} = \begin{pmatrix} a_1 & a_2 & \dots & a_s & b_2 & b_3 & \dots & b_s \\ 1 & \dots & 0 & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots & & \vdots \\ 0 & \dots & 1 & 0 & \dots & 0 & \dots & 0 \\ 0 & \dots & \dots & \dots & 0 & \dots & \dots & 0 \\ 0 & \dots & \dots & 0 & 1 & \dots & 0 & 0 \\ \vdots & & & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & \dots & 0 & 0 & \dots & 1 & 0 \end{pmatrix} \begin{pmatrix} y_{t-1} \\ y_{t-2} \\ \vdots \\ y_{t-s} \\ u_{t-2} \\ u_{t-3} \\ \vdots \\ u_{t-s} \end{pmatrix}$$
$$+ \begin{pmatrix} b_1 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} u_{t-1} + \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{pmatrix} e_t$$
$$y_t = \begin{pmatrix} 1 \; 0 \dots 0 \end{pmatrix} \begin{pmatrix} y_t \\ y_{t-1} \\ \vdots \\ y_{t-s+1} \\ u_{t-1} \\ u_{t-2} \\ \vdots \\ u_{t-s+1} \end{pmatrix} \qquad (19)$$

Here, $s \le q$ is employed to impart generality; s effectively serves as a parameter determining controller complexity. This corresponds to a stochastic state-space realization with full-feedback. This situation can be treated as a special case of that explained earlier (i.e. select $l = 0$).

Here, $C = I$ and measurement noise is absent. Therefore, the optimal time-invariant observer is open-loop and as such $\hat{x}_{t|t} = x_t$ once the effect of initial conditions has dissipated. When policy L_k is evaluated, the algorithm described earlier then specializes to:

$$\begin{aligned}
\mathbb{E}_{(\cdot|\mathcal{I}_t^0, u_t)}\{r_t\} &= y_t' y_t + u_t' R_u u_t && (20)\\
\hat{x}_{t|t} &= [y_t, .., y_{t-s+1}, u_{t-1}, .., u_{t-s+1}]' && (21)\\
\tilde{x}_{t+1} &= [\hat{y}_{t+1}, y_t, .., y_{t-s+2}, u_t, .., u_{t-s+2}]' && (22)\\
z_t &= [\hat{x}_{t|t}, u_t]' && (23)\\
\tilde{z}_{t+1} &= [\tilde{x}_{t+1}, -L\tilde{x}_{t+1}]' && (24)\\
\hat{y}_{t+1} &= \sum_{i=1}^{s} (a_i y_{t+1-i} + b_i u_{t+1-i}) && (25)
\end{aligned}$$

B. The Policy Iteration Algorithm

The following is a summary of the proposed algorithm. The underlying assumption is that the learnt controller is to be implemented on-line. Before the policy iteration begins, assume that the system is under some sub-optimal (e.g. PI[1]) control for a period of time longer than $2q$, where a sufficiently large $q \in \mathbb{N}$ is chosen. R_u, complexity-control factor s, and discount factor g, are similarly pre-specified.

$\{\hat{a}_i, \hat{b}_i\}_{i=1}^q$ can be obtained (see Section IV) if the input to the plant is subject to external dithering, i.e. $u_t = f(y_t) + d_t$, where f denotes the PI control law. Each episode k is assumed to be of a duration $T > \frac{n_z(n_z+1)}{2}, n_z := (s)(n_y) + (s-1)(n_u) + n_u$

For $k = 1, 2, \ldots$

- *Policy Evaluation.* The system is run in closed loop under the current policy $L_k : u_t = -L_k \hat{x}_{t|t-l} + d_t$ (21). d_t is simulated as white, zero-mean, Gaussian noise. Collection of scalar reinforcements $\{\mathbb{E}_{\cdot|\mathcal{I}_{t,u_t}^l}(r_t)\}_{t=0}^T$ and regressors (15) $\{\phi_t\}_{t=0}^T$ proceeds and a least squares estimate of θ^{L_k} is generated at the end of the k-th episode. Recall that θ^{L_k} is equivalent to H^{L_k}. If so desired, the estimates $\{\hat{a}_i, \hat{b}_i\}$ are refined continuously across all episodes by means of recursive least squares.
- *Policy Improvement.* With knowledge of H^{L_k}, policy improvement is carried out as in (16)
- *Termination.* The preceding steps cease upon ϵ-convergence (i.e. $||L_{k+1} - L_k|| \leq \epsilon$).

Note that the first policy evaluation step is carried out with the original PI controller in place.

V. Example

We demonstrate the efficacy of the algorithm on the following system. The underlying system is taken to be of the form

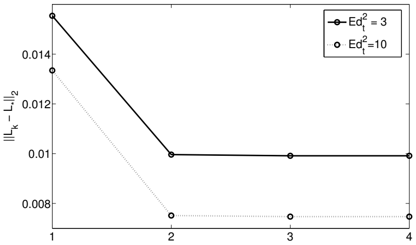

Fig. 2. Plot of error vs. episodes for various levels of dithering

$$\begin{aligned}
x_{t+1} &= \begin{pmatrix} 0.6 & -0.7 \\ 1 & 0 \end{pmatrix} x_t + \begin{pmatrix} 1.1 \\ 0.3 \end{pmatrix} u_t + w_t\\
y_t &= \begin{pmatrix} 0 & 1 \end{pmatrix} x_t + v_t && (26)
\end{aligned}$$

where w_t and v_t are uncorrelated , zero-mean Gaussian stochastic processes, with covariances $diag([1, 1])$ and 1 respectively. As such, the time-invariant Kalman gain is $(0.219, 0.718)'$ and $\mathbb{E}e_t^2 = 3.56$; R_u = 15 and $g = 0.999$ are the other user-defined quantities.

The algorithm is initialized with a PI controller with gain, 0.075 and an integral time 4. The total number of episodes is set to 4, and the duration of each episode is $T = 4000$ time-units. We show the results for the case where $q = s = 15$. For reference, the optimal controller gain, L_* for the pre-supposed realization (19) can be computed using the true values of $\{a_i, b_i\}_{i=1}^q$. Results of a typical stochastic realization is shown in Fig. 2 for $\mathbb{E}d_t^2 = 3, 10$. Since the system is time-invariant, learning of $\{\hat{a}_i, \hat{b}_i\}$ was done only during the initialization phase. The non-zero error is the result of imperfect $\{\hat{a}_i, \hat{b}_i\}$ and should decrease in the event that learning of the latter parameters occurs throughout the policy iteration phase. Numerical experiments indicate convergence for large values of q where (19) does not represent a controllable system. As stated earlier, the controllability premise is required to prove convergence of the algorithm. Whether this condition is also necessary is the subject of future work. This was also observed in the context of the deterministic algorithm, where a concatenated input-output vector served the provided state feedback [15].

As expected, the higher the levels of dithering, the smaller the error, $||L_k - L_*||_2$, $\forall k$.

A. Controller Reduction

Convergence behavior where $s < q$ needs to be explored. Consider the same scenario where now, $s = 2$, and $q = 15$. In this case, using the true $\{a_i, b_i\}_{i=1}^q$, we compute L_* as that corresponding to (19) where $s = 2$. Convergence results are shown in Fig. 3 and indicate that our proposed algorithm serves the role of automatic controller order-reduction.

As before, developing convergence proofs for the reduced order case is the aim of future work.

[1] Proportional-Integral control, see [17]

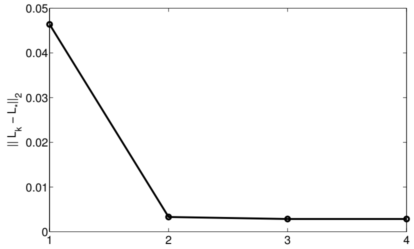

Fig. 3. Plot of error vs. episodes, $\mathbb{E}d_t^2 = 10$

VI. Conclusion, Limitations and Future Research

This paper extends the RL paradigm for model-free, adaptive control to linear, stochastic systems subject to a quadratic cost-function. A model-free algorithm is derived by borrowing certain concepts from closed-loop system identification. However, the need for complete system identification is circumvented. Results suggestive of automatic controller reduction have also been presented.

However, there remains certain issues to be clarified and opportunities to be explored.

- *Convergence Proofs.* Although empirical results are promising, there needs to be convergence proofs for the case of over-and-under modeling.
- *Extension to Nonlinear Systems.* For tighter levels of control, the ARMAX idealization may not be appropriate. An extension of this work to nonlinear, stochastic systems may follow the contribution of [15], where local RL controllers were employed in the context of high-dimensional LQR problems. Since RL methods are direct, the issue of the transmission of (bias and variance) modeling errors to controller error, in the case of the indirect approach, is avoided. We postulate that RL provides an automatic mechanism yielding superior closed-loop performance.

References

[1] Steven J. Bradtke, B. Erik Ydstie, and Andrew G. Barto. Adaptive linear quadratic control using policy iteration. In *Proceedings of the American Control Conference*, pages 3475–3476, 1994.

[2] R.S. Sutton, A.G. Barto, and R.J. Williams. Reinforcement learning is direct adaptive optimal control. In *Proceedings of the 1991 American Control Conference*, Boston, Massachusetts, 1991.

[3] C.J. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge University, 1989.

[4] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT-Press, 1998.

[5] Richard E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[6] T. Landelius. *Reinforcement Learning and Distributed Local Model Synthesis*. PhD thesis, Linkoping University, Sweden, 1997.

[7] Asama Al-Tamimi, Frank L. Lewis, and Murad Abu-Khalaf. Model-free q-learning designs for linear discrete-time zero-sum games with application to h-infinity control. *Automatica*, 43:473–481, 2007.

[8] S. Hagen and B. Krose. Linear quadratic regulation using reinforcement learning. In *Belgian-Dutch Conference on Machine Learning*, pages 39–46, 1998.

[9] S. H.G. ten Hagen. *Continuous State Space Q-Learning for Control of Nonlinear Systems*. PhD thesis, University of Amsterdam, 2001.

[10] L. Ljung. *System Identification: Theory for the User*. PTR Prentice Hall, Upper Saddle River, N.J, 2nd edition, 1999.

[11] M.H. Hsiao, J.K. Huang, and D.E. Cox. *Journal of Dynamic Systems, Measurement and Control*, 118(2):366–372, 1996.

[12] Joe Qin. An overview of subspace identification. *Computers and Chemical Engineering*, 30:1502–1513, 2006.

[13] Goro Obinata and Brian D.O. Anderson. *Model Reduction for Control System Design*. Springer, 2000.

[14] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2000.

[15] Steven Bradtke. *Incremental Dynamic Programming for On-Line Adaptive Optimal Control*. PhD thesis, University of Massachusetts, Amherst, 1994.

[16] G. Goodwin and K. Sin. *Adaptive Filtering Prediction and Control*. 1984.

[17] Dale E. Seborg, Thomas F. Edgar, and Duncan A. Mellichamp. *Process Dynamics and Control*. Wiley, 2nd edition, 2003.

Appendix

We briefly present a version of the recursive least squares algorithm. Given that one is presented sequentially with time-indexed samples $\{r_t, \phi_t\}_{t=0}^{T}$, the algorithm [16] is:

$$\begin{aligned}\hat{\theta}_t &= \hat{\theta}_{t-1} + G_t(r_t - \phi'_{t-1}\hat{\theta}_{t-1}) \\ G_t &= P_{t-1}\phi_t(I + \phi'_t P_{t-1}\phi_t)^{-1} \\ P_t &= (I - G_t\phi'_t)\,P_{t-1} \end{aligned} \tag{27}$$

Persistency of excitation relates to the following condition

$$\rho_1 I \leq \frac{1}{N}\sum_{i=1}^{N}\phi_{t-i}\phi'_{t-i} \leq \rho_2 I, \forall t \geq N_0, N \geq N_0 \tag{28}$$

where $\rho_1, \rho_2, \text{and} N_0$ are positive numbers