[5] C. Wang, T. R. Chen, and T. F. Liu, "Deterministic learning and data-based modeling and control," *Acta Autom. Sinica*, vol. 35, no. 6, pp. 693–706, Jun. 2009.
[6] R. H. Chi and Z. S. Hou, "A model-free periodic adaptive control for freeway traffic density via ramp metering," *Acta Autom. Sinica*, vol. 36, no. 7, pp. 1029–1033, Jul. 2010.
[7] J. Zhang, H. B. Xu, and H. B. Zhang, "Observer-based sampled-data control for a class of continuous nonlinear systems," *Acta Autom. Sinica*, vol. 36, no. 12, pp. 1780–1787, Dec. 2010.
[8] M. Xiang and W. R. Shi, "A cluster data management algorithm based on data correlation of wireless sensor networks," *Acta Autom. Sinica*, vol. 36, no. 9, pp. 1343–1350, May 2010.
[9] F. Wu, Y. Zhong, and Q. Y. Wu, "Mining frequent patterns over data stream under the time decaying model," *Acta Autom. Sinica*, vol. 36, no. 5, pp. 674–684, May 2010.
[10] K. F. Wang, Z. J. Li, and S. M. Tang, "Visual traffic data collection approach based on multi-features fusion," *Acta Autom. Sinica*, vol. 37, no. 3, pp. 322–330, Mar. 2011.
[11] X. K. Dong, Y. C. Fang, and Y. D. Zhang, "An improved AFM dynamic imaging method based on data fusion of neighboring point set," *Acta Autom. Sinica*, vol. 37, no. 2, pp. 214–221, Feb. 2011.
[12] G. O. Guardabassi and S. M. Savaresi, "Virtual reference direct design method: An off-line approach to data-based control system design," *IEEE Trans. Autom. Control*, vol. 45, no. 5, pp. 954–959, May 2000.
[13] S. Tan and J. F. Zhang, "Adaptive measured-data based linear quadratic optimal control of stochastic systems," *Int. J. Control*, vol. 80, no. 10, pp. 1676–1689, 2007.
[14] J. X. Xu and Z. S. Hou, "Notes on data-driven system approaches," *Acta Autom. Sinica*, vol. 35, no. 6, pp. 668–675, Jun. 2009.
[15] K. Ogata, *Modern Control Engineering*, 5th ed. Upper Saddle River, NJ: Prentice-Hall, 2009.
[16] P. J. Antsaklis and A. N. Michel, *Linear Systems*. Berlin, Germany: Birkhäuser, 2006.
[17] E. Hendricks, O. Jannerup, and P. H. Sørensen, *Linear Systems Control: Deterministic and Stochastic Methods*. Berlin, Germany: Springer-Verlag, 2008.
[18] S. Barnett, *Introduction to Mathematical Control Theory*. New York: Oxford Univ. Press, 1975.
[19] T. Li, *Controllability and Observability for Quasilinear Hyperbolic Systems*. Springfield, MO: AIMS, 2010.
[20] J. N. Juang, *Applied System Identification*. Englewood Cliffs, NJ: Prentice-Hall, 1994.
[21] I. Markovsky, J. C. Willems, S. V. Huffel, B. D. Moor, and R. Pintelon, "Application of structured total least squares for system identification and model reduction," *IEEE Trans. Autom. Control*, vol. 50, no. 10, pp. 1490–1500, Oct. 2005.
[22] M. Verhaegen and V. Verdult, *Filtering and System Identification: A Least Squares Approach*. Cambridge, U.K.: Cambridge Univ. Press, 2007.
[23] F. Ding, P. X. Liu, and G. Liu, "Multiinnovation least-squares identification for system modeling," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 40, no. 3, pp. 767–778, Jun. 2010.
[24] X. Liu and J. Lu, "Least squares based iterative identification for a class of multirate systems," *Automatica*, vol. 46, no. 3, pp. 549–554, Mar. 2010.
[25] O. Nelles, *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. New York: Springer-Verlag, 2001.
[26] W. Yu and X. Li, "Some new results on system identification with dynamic neural networks," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 412–417, Mar. 2001.
[27] X. M. Ren, A. B. Rad, P. T. Chan, and L. L. Wai, "Identification and control of continuous-time nonlinear systems via dynamic neural networks," *IEEE Trans. Ind. Electron.*, vol. 50, no. 3, pp. 478–486, Jun. 2003.
[28] S. Yilmaz and Y. Oysal, "Fuzzy wavelet neural network models for prediction and identification of dynamical systems," *IEEE Trans. Neural Netw.*, vol. 21, no. 10, pp. 1599–1609, Oct. 2010.
[29] H. L. Wei, S. A. Billings, Y. F. Zhao, and L. Z. Guo, "An adaptive wavelet neural network for spatio-temporal system identification," *Neural Netw.*, vol. 23, no. 10, pp. 1286–1299, Dec. 2010.
[30] D. Z. Cheng, H. S. Qi, and Z. Q. Li, "Model construction of Boolean network via observed data," *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 525–536, Apr. 2011.
[31] A. Alessandri, M. Baglietto, G. Battistelli, and M. Gaggero, "Moving-horizon state estimation for nonlinear systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 22, no. 5, pp. 768–780, May 2011.
[32] B. Subudhi and D. Jena, "Nonlinear system identification using memetic differential evolution trained neural networks," *Neurocomputing*, vol. 74, no. 10, pp. 1696–1709, May 2011.
[33] L. Wang, *Support Vector Machines: Theory and Applications*. Berlin, Germany: Springer-Verlag, 2005.
[34] I. Steinwart and A. Christmann, *Support Vector Machines*. New York: Springer-Verlag, 2008.
[35] L. Zhang, Y. G. Xi, and W. D. Zhou, "Identification and control of discrete-time nonlinear systems using affine support vector machines," *Int. J. Artif. Intell. Tools*, vol. 18, no. 6, pp. 929–947, 2009.
[36] B. Moore, "Principal component analysis in linear systems: Controllability, observability, and model reduction," *IEEE Trans. Autom. Control*, vol. 26, no. 1, pp. 17–32, Feb. 1981.

Approximate Dynamic Programming for Optimal Stationary Control with Control-Dependent Noise

Yu Jiang, *Student Member*, *IEEE*, and
Zhong-Ping Jiang, *Fellow*, *IEEE*

***Abstract*— This brief studies the stochastic optimal control problem via reinforcement learning and approximate/adaptive dynamic programming (ADP). A policy iteration algorithm is derived in the presence of both additive and multiplicative noise using Itô calculus. The expectation of the approximated cost matrix is guaranteed to converge to the solution of some algebraic Riccati equation that gives rise to the optimal cost value. Moreover, the covariance of the approximated cost matrix can be reduced by increasing the length of time interval between two consecutive iterations. Finally, a numerical example is given to illustrate the efficiency of the proposed ADP methodology.**

***Index Terms*— Approximate dynamic programming, control-dependent noise, optimal stationary control, stochastic systems.**

I. Introduction

Reinforcement learning [1] is one of the most important branches in learning theory. Roughly speaking, it is concerned with how an agent improves decisions at each step to achieve some long-term goal, based on interactions with and rewards received from the environment. One technique to solve reinforcement learning problems is called adaptive/approximate dynamic programming (ADP), pioneered by Werbos [2]–[5]. The strategy consists of estimating the cost function online, and making further decisions through successive iterations.

Reinforcement learning and ADP are extensively studied for Markov decision processes for both the discrete and continuous state and action spaces, see [6]–[9], for example.

Manuscript received January 13, 2011; revised June 2, 2011; accepted August 14, 2011. Date of publication September 26, 2011; date of current version December 13, 2011. This work was supported in part by the U.S. National Science Foundation, under Grant DMS-0906659 and Grant ECCS-1101401.

Y. Jiang is with the Department of Electrical and Computer Engineering, Polytechnic Institute of New York University, Brooklyn, NY 11201 USA (e-mail: yu.jiang@nyu.edu).

Z.-P. Jiang is with the Department of Electrical and Computer Engineering, Polytechnic Institute of New York University, Brooklyn, NY 11201 USA. He is also with the College of Engineering, Beijing University, Beijing 100214, China (e-mail: zjiang@poly.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2011.2165729

Over the past few years, reinforcement learning and ADP theories have been applied to feedback control designs of general dynamic systems, with an objective to achieve stability as well as optimality properties. See the tutorial [10] by Lewis and Vrabie for more details and references. The state-feedback problem has been studied in [11] and [12], and the output-based policy-iteration algorithm in [13] and [14]. Some corresponding results for the value-iteration case have been obtained in [15] and [16], and references therein.

Nearly all papers regarding ADP in solving linear optimal control problems assume that, in each iteration step, the cost function can be accurately calculated (except in [11] where a recursive least squares scheme for linear discrete-time systems is considered), mainly because most of the published ADP papers focus on deterministic systems. Therefore, the purpose of this brief is to investigate how ADP can be applied to solve the optimal stationary control problem for stochastic systems. Toward this end, we develop in this brief an ADP control algorithm that approximates the optimal cost and control policy, in the presence of both additive and multiplicative noise.

In order to analyze continuous-time stochastic systems, we use Itô's lemma to derive the relationship between the cost and state variables. Then, we show that the sequence of the expectation of cost matrices converges to the optimal value, and the covariance of the cost matrices approximated at each iteration step can be reduced by increasing the time interval between two consecutive iterations. Based on these facts, we show that the algorithm generates a sequence of control policies that converges to a neighborhood of the optimal policy in finite steps, with a probability higher than any predefined value in (0, 1).

The rest of this brief is organized as follows. In Section II, we introduce some preliminaries regarding optimal stationary control for continuous-time linear systems with additive and multiplicative noise. In Section III, we develop an ADP algorithm for online learning of the optimal cost matrix, and show the convergence of the control policies generated by the algorithm. In Section IV, a numerical example is provided to illustrate the efficiency of the proposed control scheme. Finally, Section V gives some concluding remarks.

Throughout this brief, a feedback gain matrix K, in a state-feedback law $u = -Kx$, is called a *policy*. Also, we use $\|\cdot\|$ to mean the Frobenius norm. For any $P = P^T \in \mathbb{R}^{n\times n}$, we define $\text{vec}(P) := [p_{11}, 2p_{12}, \ldots, 2p_{1n}, p_{22}, 2p_{23}, \ldots, 2p_{n-1,n}, p_{nn}]^T$, and $(P)_v := [p_{11}, p_{12}, \ldots, p_{1n}, p_{22}, p_{23}, \ldots, p_{nn}]$; for any vector $x \in \mathbb{R}^n$, we define $q(x) := [x_1^2, x_1x_2, \ldots, x_1x_n, x_2^2, \ldots, x_{n-1}x_n, x_n^2] \in \mathbb{R}^{n(n+1)/2}$.

II. Problem Formulation and Preliminaries

In this section, we study the optimal stationary control problem of continuous-time linear stochastic systems with partially unknown system dynamics, in the following form:

$$dx = (Ax + Bu)dt + Fudv + Ddw, \quad x(0) = 0 \tag{1}$$

where the state $x \in \mathbb{R}^n$ is fully measurable and available for feedback control; $u \in \mathbb{R}^m$ is the control input; $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, and $F \in \mathbb{R}^{n\times m}$ are constant matrices describing the system dynamics, with (A, B) controllable but A unknown; $D \in \mathbb{R}^{n\times n}$ is a nonsingular constant matrix; $v \in \mathbb{R}$ and $w \in \mathbb{R}^n$ are independent Brownian motions. The objective of this control problem is to determine a time-invariant policy K that minimizes the following expected cost function in the steady state:

$$J = \mathcal{E}\left[x^T Qx + u^T Ru\right] \tag{2}$$

where $\mathcal{E}[\cdot]$ denotes the expectation, $Q \geq 0$ and $R > 0$ are symmetric matrices, with $(A, Q^{1/2})$ observable.

We first recall an iterative algorithm from [17], which is based on the assumption that matrix A is perfectly known. It is shown in [17] that, for any constant matrix D with full rank, the steady-state covariance X is the solution of the following equation:

$$0 = A_cX + XA_c^T + FKXK^TF^T + DD^T \tag{3}$$

with $A_c = A - BK$. The steady-state cost can be expressed as

$$J = \text{tr}\left[PDD^T\right] \tag{4}$$

where P is the unique positive definite solution of

$$0 = PA_c + A_c^TP + Q + K^TF^TPFK + K^TRK \tag{5}$$

and the optimal feedback policy is determined as

$$K = \left(R + F^TPF\right)^{-1}B^TP. \tag{6}$$

As in [17], we assume there exist K and a unique symmetric matrix $P > 0$ satisfying both (5) and (6).

Definition 1: A control policy K is called *stabilizing* if the matrix

$$I \otimes (A - BK) + (A - BK) \otimes I + FK \otimes FK$$

is Hurwitz, where $\otimes$ denotes the Kronecker product.

In [17], it is shown that, given a stabilizing policy, (5) and (6) can be solved iteratively according to the following theorem.

Theorem 3: Let K_0 be stabilizing, and $\{P_k\}$ and $\{K_k\}$ be two sequences obtained from solving

$$0 = P_kA_k + A_k^TP_k + Q + K_k^T\left(R + F^TP_kF\right)K_k \tag{7}$$

$$K_{k+1} = \left(R + F^TP_kF\right)^{-1}B^TP_k \tag{8}$$

where

$$A_k = A - BK_k, \quad k = 0, 1, 2, \ldots \tag{9}$$

Then:

1) $P_{k+1} \leq P_k$, $k = 0, 1, \ldots$;
2) $P = \lim_{k\to\infty} P_k$ and $K = \lim_{k\to\infty} K_k$ are the unique solutions of (5) and (6).

In the next section, we will develop an algorithm to solve P online, without assuming the knowledge of A.

III. ADP for Optimal Stationary Control

The purpose of this section is to adopt the theory of ADP and develop numerical schemes for online learning and approximation of the cost matrices. Different from the ADP for deterministic systems as in [12]–[14], and [18], the approximated cost matrix is itself a random variable. Therefore, it is necessary to investigate how the convergence will be affected in the presence of both additive and multiplicative noise.

It is assumed that, only for the purpose of learning, we can introduce extra additive noise to the system, such that the additive noise Ddw is extended to $D(t)dw$, where $D(t) \in \mathbb{R}^{n\times n}$ is a matrix of periodic piecewise constant functions. However, after the policy is numerically obtained through the ADP algorithm, the extra noise is no longer used, and hence $D(t)$ is reduced to the original D.

Also, we make the following assumptions.

Assumption 1: There exist an integer $l = n(n+1)/2$ and constant matrices D_i, such that for all $i = 1, 2, \ldots, l$

$$D(mT + t) = D(t) = D_i, \quad t \in (t_{i-1}, t_i] \tag{10}$$

where $0 \le t_0 < t_1 < \cdots < t_l \le T$, m is any arbitrary nonnegative integer, and T is the period of $D(t)$.

Assumption 2: $\text{rank}[W_1^T, W_2^T, \ldots, W_l^T] = l$ where $W_i = (D_i D_i^T)_v$ for all $i = 1, 2, \ldots, l$.

Under Assumptions 1 and 2, we can develop an algorithm to compute ADP-based stationary control law, as shown below.

Remark 4: Since the covariance of the state exponentially converges to its steady-state value, all the Z's defined in the algorithm after Lemma 1 are bounded. Under Assumption 1, the nonsingularity of $(Z + m_k U)$ is guaranteed for large m_k.

Lemma 3: For all $k = 0, 1, 2, \ldots$, given $\hat{K}_k$ a stabilizing policy, we have

$$\mathcal{E}(\hat{P}_k | x) = P_k \tag{11}$$

where P_k is the cost associated with $\hat{K}_k$.

Proof: By Itô's lemma

$$\begin{aligned} d\left(x^T P_k x\right) &= dx^T P_k x + x^T P_k dx + dx^T P_k dx \\ &= x^T \left(A_k^T P_k + P_k A_k + \hat{K}_k^T F^T P_k F \hat{K}_k\right) x dt \\ &\quad + 2x^T P_k F \hat{K}_k x dv + 2x^T P_k D(t) dw \\ &\quad + \text{tr}\left(D(t) D(t)^T P_k\right) dt \\ &= -x^T \left(Q + \hat{K}_k^T R \hat{K}_k\right) x dt + W_i \text{vec}(P_k) dt \\ &\quad + \lambda_k(x) dv + \gamma_k(x,t) dw \end{aligned}$$

where $\lambda_k(x) = 2x^T P_k F \hat{K}_k x$, and $\gamma_k(x,t) = 2x^T P_k D(t)$.

Therefore, we have

$$\begin{aligned} &[q(x(t)) - q(x(t+\Delta t)) + W_i \Delta t] \text{vec}(P_k) \\ &\quad = \int_t^{t+\Delta t} \left(x^T Q x + u^T R u\right) d\tau \\ &\qquad + \int_t^{t+\Delta t} \lambda_k(x) dv + \int_t^{t+\Delta t} \gamma_k(x,t) dw. \end{aligned}$$

Algorithm ADP-based stochastic policy iteration

Let K_0 be an initial stabilizing control policy. Set the maximum number of iteration steps N, and let $k = 0$, $m_k = 1$, $\hat{K}_0 = K_0$. Apply the control input $u = -\hat{K}_0 x$ to (1), with D replaced by $D(t)$.

if $k \le N$ **then**

$$Z \leftarrow \sum_{i=0}^{m_k-1} \begin{bmatrix} q(x(Ti + t_0')) - q(x(Ti + t_1')) \\ q(x(Ti + t_1')) - q(x(Ti + t_2')) \\ \vdots \\ q(x(Ti + t_{l-1}')) - q(x(Ti + t_l')) \end{bmatrix}_{l\times l},$$

$$U \leftarrow \begin{bmatrix} (t_1 - t_0) W_1 \\ (t_2 - t_1) W_2 \\ \vdots \\ (t_l - t_{l-1}) W_l \end{bmatrix}_{l\times l},$$

$$c(t_1, t_2) \leftarrow \int_{t_1}^{t_2} (x^T Q x + u^T R u) d\tau,$$

$$t_j'(k) \leftarrow \begin{cases} t_j, & k = 0, \\ t_j + \sum_{i=0}^{k-1} m_i T, & k = 1, 2, \ldots, \end{cases}$$

$$V \leftarrow \sum_{i=0}^{m_k-1} \begin{bmatrix} c(Ti + t_0', Ti + t_1') \\ c(Ti + t_1', Ti + t_2') \\ \vdots \\ c(Ti + t_{l-1}', Ti + t_l') \end{bmatrix}_{l\times 1}$$

if $\det(Z + m_k U) \neq 0$ **then**

$$\begin{aligned} \text{vec}(\hat{P}_k) &\leftarrow (Z + m_k U)^{-1} V, \\ \hat{K}_{k+1} &\leftarrow (R + F^T \hat{P}_k F)^{-1} B \hat{P}_k, \\ k &\leftarrow k + 1, \\ m_k &\leftarrow 1. \end{aligned}$$

Apply $u = -\hat{K}_k x$ as the control input to the system.

else

$m_k \leftarrow m_k + 1$.

end if

else

Apply $u = -\hat{K}_k x$ as the control input to the system.

end if

Hence, using compact notation

$$\text{vec}(P_k) = (Z + m_k U)^{-1} (V + \Gamma) \tag{12}$$

where m_k can be any positive integer and

$$\Gamma = \sum_{i=0}^{m_k-1} \begin{bmatrix} \int_{Ti+t_0'}^{Ti+t_1'} (\lambda_k(x) dv + \gamma_k(x,t) dw) \\ \int_{Ti+t_1'}^{Ti+t_2'} (\lambda_k(x) dv + \gamma_k(x,t) dw) \\ \vdots \\ \int_{Ti+t_{l-1}'}^{Ti+t_l'} (\lambda_k(x) dv + \gamma_k(x,t) dw) \end{bmatrix}. \tag{13}$$

Finally, note that

$$\mathcal{E}\left[\text{vec}(P_k - \hat{P}_k)|x\right] = (Z + m_k U)^{-1}\,\mathcal{E}\left[\Gamma|x\right] = 0. \quad (14)$$

The proof is complete. ∎

The next lemma shows that the covariance of the approximated cost matrices can be reduced by increasing m_k.

Lemma 4: $\lim_{m_k\to\infty} \text{Var}\left[\text{vec}(\hat{P}_k)|x\right] = 0$ with probability 1.

Proof:

$$\begin{aligned}&\mathcal{E}\left[\text{vec}(\hat{P}_k - P_k)(\hat{P}_k - P_k)^T \,\middle|\, x\right]\\ &= (Z + m_k U)^{-1}\,\mathcal{E}\left[\Gamma\Gamma^T \,\middle|\, x\right](Z + m_k U)^{-T}\\ &= \left(\frac{1}{m_k T}Z + \frac{1}{T}U\right)^{-1}\mathcal{E}\left[\frac{1}{T^2 m_k^2}\Gamma\Gamma^T \,\middle|\, x\right]\left(\frac{1}{m_k T}Z + \frac{1}{T}U\right)^{-T}.\end{aligned}$$

According to the properties of Brownian motions, the entry on the sth row and pth column of $\Gamma\Gamma^T$ is

$$\begin{cases}\sum_{i=0}^{m_k-1}\int_{Ti+t_{s-1}}^{Ti+t_s}\left(|\lambda_k(x)|^2 + ||\gamma_k(x,t)||^2\right)dt, & \text{for } s = p,\\ 0, & \text{for } s \neq p.\end{cases}$$

Noticing that $\hat{K}_k$ is stabilizing, $x(t)$ has finite moments, and thus

$$\mathcal{P}\left[\lim_{\Delta t\to\infty}\frac{\int_t^{t+\Delta t}\left(|\lambda_k(x)|^2 + \|\gamma_k(x,\tau)\|^2\right)d\tau}{\Delta t^2} = 0\right] = 1$$

where we used $\mathcal{P}[\cdot]$ to denote the conditional probability given the knowledge of x.

Finally, we have

$$\lim_{m_k\to\infty}\mathcal{E}\left[\text{vec}(\hat{P}_k - P_k)(\hat{P}_k - P_k)^T|x\right] = 0 \quad (15)$$

with probability 1. ∎

Next, we study how the stochastic property of both $\hat{P}_k$ and $\hat{K}_k$ calculated in each iteration will affect the convergence. For this purpose, we write the policy updating laws from the algorithm in the following compact form:

$$\hat{K}_{k+1} = f(\hat{K}_k) + \Delta_k \quad (16)$$

for $k = 0, 1, \ldots$, where $f: \mathbb{R}^{n\times m} \to \mathbb{R}^{n\times m}$ represents the mapping from any stabilizing policy $\hat{K}_k$ to the updated policy $\hat{K}_{k+1}$ from (7) and (8). Δ_k is the difference between the policy $\hat{K}_{k+1}$ and the ideal policy K_{k+1} obtained from (8).

In order to identify conditions under which $\hat{K}_k$ is stabilizing for $k = 0, 1, 2, \ldots$, we define a Lyapunov-like function

$$v(\hat{K}_k) := \text{tr}(P_k) \quad (17)$$

where $P_k = P_k^T$ is the solution of (7) with $K_k = \hat{K}_k$. Notice that $\hat{K}_k$ is stabilizing if and only if P_k is positive definite or $v(\hat{K}_k) < \infty$.

Further, let us define a compact set

$$B_S = \{K : v(K) - v(K^*) \le S\} \quad (18)$$

where K^* is the optimal policy satisfying $f(K^*) = K^*$.

Lemma 5: For any given probability $0 < p < 1$ and any sufficiently small constant $d > 0$, we can always find $N_k > 0$, such that for all $m_k > N_k$, we have $\mathcal{P}[\|\Delta_k\| < d] > p$.

Proof: By Lemma 2, with probability 1 we have $\lim_{m_k\to\infty}\text{Var}[\hat{K}_{k+1}] = 0$, which implies

$$\begin{aligned}&\lim_{m_k\to\infty}\mathcal{P}\left\{\mathcal{E}\left[\|\Delta_k\|^2 = 0\,\middle|\, x\right]\right\}\\ &= \lim_{m_k\to\infty}\mathcal{P}\left\{\mathcal{E}\left[\|\hat{K}_{k+1} - f(\hat{K}_k)\|^2 = 0\,\middle|\, x\right]\right\}\\ &= \lim_{m_k\to\infty}\mathcal{P}\left\{\text{Var}\left[\hat{K}_{k+1}\right] = 0\right\}\\ &= 1.\end{aligned}$$

Hence, there exists $N_k > 0$ such that, for all $m_k > N_k$, we have $\mathcal{P}[\|\Delta_k\| < d] > p$. ∎

The next lemma points out that the policies generated using the algorithm converge to a neighborhood of the optimal policy.

Lemma 6: Given any large constant $S > 0$, there exist $\beta > 0$ and a sufficiently small constant $d > 0$, such that, for any initial stabilizing policy $\hat{K}_0$ satisfying $v(\hat{K}_0) \le v(K^*) + S$ and any bounded sequence of disturbances $\|\Delta_k\| < d$, the sequence $\{\hat{K}_k\}$ of policies generated from the algorithm satisfies:

1) $\hat{K}_k$ is stabilizing for all $k = 0, 1, 2, \ldots$;
2) $\overline{\lim_{k\to\infty}}\|\hat{K}_k - K^*\|^2 < \beta\ \overline{\lim_{k\to\infty}}\|\Delta_k\|$.

Proof: We prove this lemma in three steps.

Step 1: Given a stabilizing policy $\hat{K}_k$, satisfying $\hat{K}_k \in B_s$, there exist constants $\underline{\delta} > 0$ and $\bar{\delta} > 0$, independent of $k \in \mathbb{Z}_+$, such that the following hold:

$$\bar{\delta}\|f(\hat{K}_k) - \hat{K}_k\|^2 \ge v(\hat{K}_k) - v(f(\hat{K}_k)) \ge \underline{\delta}\|f(\hat{K}_k) - \hat{K}_k\|^2,$$

$$\bar{\delta}\|\hat{K}_k - K^*\|^2 \ge v(\hat{K}_k) - v(K^*) \ge \underline{\delta}\|\hat{K}_k - K^*\|^2.$$

Indeed, note that the minimum eigenvalue of

$$I \otimes (A - B\hat{K}_k) + (A - B\hat{K}_k) \otimes I + F\hat{K}_k \otimes F\hat{K}_k$$

is a continuous function of $\hat{K}_k$ and thus has a minimum value on the compact set B_S.

For each k, let $\hat{P}_k$ be the cost matrix associated with policy $\hat{K}_k$. Then, we have

$$\begin{aligned}&v(\hat{K}_k) - v(f(\hat{K}_k))\\ &= \text{tr}\left(\int_0^\infty e^{A_{k+1}^T\tau}(\hat{K}_k - f(\hat{K}_k))^T R_k(\hat{K}_k - f(\hat{K}_k))e^{A_{k+1}\tau}d\tau\right)\\ &= \int_0^\infty \text{tr}\left(e^{A_{k+1}^T\tau}(\hat{K}_k - f(\hat{K}_k))^T R_k(\hat{K}_k - f(\hat{K}_k))e^{A_{k+1}\tau}\right)d\tau\\ &\ge \lambda_{\min}(R_k)\ \lambda_{\min}\left(\int_0^\infty e^{A_{k+1}\tau}e^{A_{k+1}^T\tau}d\tau\right)\|\hat{K}_k - f(\hat{K}_k)\|^2\\ &\ge \underline{\delta}\ \|\hat{K}_k - f(\hat{K}_k)\|^2\end{aligned}$$

where $R_k = R + F^T P_k F$, $A_k = A - B\hat{K}_k$, for all $k = 0, 1, 2, \ldots$. $\lambda_{\min}(\cdot)$ denotes the smallest eigenvalue of a real symmetric matrix for all $\hat{K}_k \in B_S$, $\underline{\delta} = \lambda_{\min}(R_k)$ $(\int_0^\infty e^{A_{k+1}\tau}e^{A_{k+1}^T\tau}d\tau)$.

Similarly, it follows that

$$\begin{aligned}
&v(\hat{K}_k) - v(f(\hat{K}_k)) \\
&= \operatorname{tr}\left(\int_0^\infty e^{A_{k+1}^T\tau}(\hat{K}_k - f(\hat{K}_k))^T R_k(\hat{K}_k - f(\hat{K}_k))e^{A_{k+1}\tau}d\tau\right) \\
&\le \lambda_{\max}(R_k)\lambda_{\max}\left(\int_0^\infty e^{A_{k+1}\tau}e^{A_{k+1}^T\tau}d\tau\right)\|\hat{K}_k - f(\hat{K}_k)\|^2 \\
&\le \bar{\delta}\|\hat{K}_k - f(\hat{K}_k)\|^2.
\end{aligned}$$

Noting that

$$\begin{aligned}
&v(\hat{K}_k) - v(K^*) \\
&= \operatorname{tr}\left(\int_0^\infty e^{A_k^T\tau}(\hat{K}_k - K^*)^T R_k(\hat{K}_k - K^*)e^{A_k\tau}d\tau\right)
\end{aligned}$$

the second part of the lemma can be proved by following the same reasoning.

Step 2: We show that there exists a constant $c > 0$, independent of $k \in \mathbb{Z}_+$, such that

$$\|f(\hat{K}_k) - K^*\| \le c\|\hat{K}_k - K^*\|^2. \tag{19}$$

According to Step 1, we directly have

$$\begin{aligned}
\|f(\hat{K}_k) - K^*\| &= \|R_k^{-1}B^T P_k - (R^*)^{-1}B^T P^*\| \\
&\le \|R_k^{-1}B^T\|\ \|P_k - P^*\| + (R_k^{-1} - (R^*)^{-1})B^T P^* \\
&= (\|R_k^{-1}B^T\| + c_0)\ \left|v(\hat{K}_k) - v(K^*)\right| \\
&\le \bar{\delta}(\|R_k^{-1}B^T\| + c_0)\ \|\hat{K}_k - K^*\|^2
\end{aligned}$$

where $c_0 = \|B^T P^*\| \max_{\hat{K}_k \in B_S}(\|R_k^{-1} - (R^*)^{-1}\|/\|P^* - P_k\|)$, $R^* = R + F^T P^* F$.

Finally, defining $c = \bar{\delta}(\|R_k^{-1}B^T\| + c_0)$, the proof is complete.

Step 3: By continuity, there exists a constant $d_1 > 0$ such that, for all $K \in B_S$ and all $\|\Delta\| < d_1$

$$v(K + \Delta) \le v(K) + \alpha\|\Delta\| < +\infty \tag{20}$$

where $\alpha = \max_{\|\Delta\| \le d_1,\ K \in B_D}\{|v(K + \Delta) - v(K)|/|\Delta|\}$.

Now, define another set

$$B_d = \left\{K : \|K - K^*\| < d_2 \le \frac{2}{c}\right\} \tag{21}$$

where $d_2 > 0$ is chosen to be so small that $B_d \subset B_S$. Since $B_0 = B_S \backslash B_d$ is a compact set and the continuous function

$$g(K) := \|f(K) - K\| \tag{22}$$

is always positive for all $K \in B_0$, there exists a constant $d_3 > 0$ such that $\|F(K) - K\| \ge d_3$ for all $K \in B_0$. Thus, given $K_0 \in B_d$, by the definition of (16), it yields

$$\begin{aligned}
v(\hat{K}_{k+1}) - v(\hat{K}_k) &= v\left(f(\hat{K}_k) + \Delta_k\right) - v(\hat{K}_k) \\
&\le v(f(\hat{K}_k) - v(\hat{K}_k) + \alpha\|\Delta_k\| \\
&\le -\underline{\delta}\|f(\hat{K}_k) - \hat{K}_k\|^2 + \alpha\|\Delta_k\| \\
&= -\underline{\delta}\left(\|f(\hat{K}_k) - \hat{K}_k\|^2 - \frac{\alpha}{\underline{\delta}}\|\Delta_k\|\right).
\end{aligned} \tag{23}$$

Choose $d = \min\{d_1, d_2, \underline{\delta}d_3^2/(2\alpha)\}$. Clearly, if $\|\Delta_k\| < \underline{\delta}d_3^2/(2\alpha)$ for all $k = 1, 2, \ldots$, then

$$v(\hat{K}_{k+1}) < v(K_0) \le S \tag{24}$$

which means $\hat{K}_{k+1} \in B_S$ for all $k \in \mathbb{Z}_+$.

Furthermore, it can also be concluded that there exists an integer M such that, for all $k > M$

$$\|f(\hat{K}_k) - \hat{K}_k\| < \frac{d_3}{2}. \tag{25}$$

Next, we show that the upper limit of $\|\hat{K}_k - \hat{K}^*\|^2$ is linearly bounded by the upper limit of $\|\Delta_k\|$. From (25), it follows that

$$\begin{aligned}
\|f(\hat{K}_k) - \hat{K}_k\| &\ge \|\hat{K}_k - K^*\| - \|f(\hat{K}_k) - K^*\| \\
&\ge \|\hat{K}_k - K^*\| - c\|\hat{K}_k - K^*\|^2 \\
&> \|\hat{K}_k - K^*\| - \frac{1}{2}\|\hat{K}_k - K^*\| \\
&= \frac{1}{2}\|\hat{K}_k - K^*\|.
\end{aligned} \tag{26}$$

Therefore

$$\begin{aligned}
v(\hat{K}_{k+1}) - v(\hat{K}_k) &\le v(f(\hat{K}_k)) - v(\hat{K}_k) + \alpha\|\Delta_k\| \\
&\le -\underline{\delta}\|f(\hat{K}_k) - \hat{K}_k\|^2 + \alpha\|\Delta_k\| \\
&= -\underline{\delta}\|f(\hat{K}_k) - K^* + K^* - \hat{K}_k\|^2 + \alpha\|\Delta_k\| \\
&\le -\underline{\delta}/4\|\hat{K}_k - K^*\|^2 + \alpha\|\Delta_k\|
\end{aligned} \tag{27}$$

which, in turn, implies

$$\overline{\lim_{k\to\infty}}\|\hat{K}_k - K^*\|^2 \le 4\alpha/\underline{\delta}\,\overline{\lim_{k\to\infty}}\|\Delta_k\|. \tag{28}$$

Finally, selecting $\beta = 4\alpha/\underline{\delta}$, the proof is complete. ■

Theorem 4: Given a predefined probability $0 < p < 1$ and an arbitrarily small constant $\epsilon > 0$, there exist an integer $N > 0$ and a constant $\Delta t > 0$, such that, under the algorithm, the following holds:

$$\mathcal{P}\left[\left\|\hat{K}_N - K^*\right\| < \epsilon\right] > p. \tag{29}$$

Proof: By Lemma 6, there exist $N > 0$ and $\delta > 0$, such that

$$\left\|\hat{K}_N - K^*\right\| < \epsilon \tag{30}$$

as long as $\|\Delta_k\| < \delta$ for $k = 0, 1, 2, \ldots, N$.

By Lemma 2, there is $\Delta t_0 > 0$ such that for all $\Delta t > \Delta t_0$, we have

$$\mathcal{P}\left(\|\Delta_k\| < \delta\right) \ge \sqrt[N]{p}. \tag{31}$$

Therefore

$$\begin{aligned}
\mathcal{P}\left[\left\|\hat{K}_N - K^*\right\| < \epsilon\right] &\ge \mathcal{P}\left[\bigcap_{k=1}^N \left(\|\Delta_k\| < \delta\right)\right] \\
&= \prod_{k=1}^N \mathcal{P}\left(\|\Delta_k\| < \delta\right) = p.
\end{aligned}$$

The proof is complete. ■

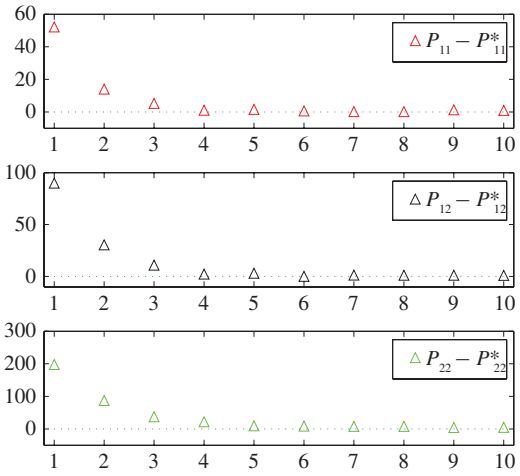

Fig. 1. Entries of the cost matrix approximated at each iteration step.

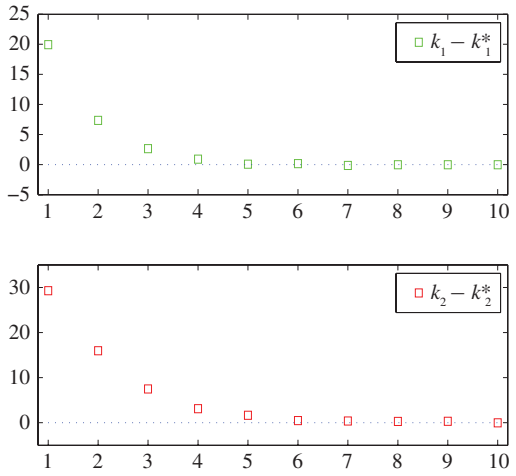

Fig. 2. Control policies updated at each iteration step.

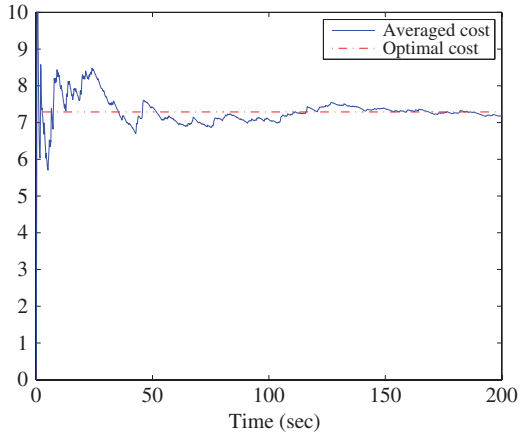

Fig. 3. Plot of the averaged cost.

IV. Simulation Results

In this section, we solve a numerical example using the algorithm developed in Section III.

Consider the following second-order controllable system:

$$dx = \left(\begin{bmatrix} a_1 & 1 \\ a_2 & a_3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \right) dt + \begin{bmatrix} 0 \\ 0.05 \end{bmatrix} u dv + D dw \tag{32}$$

where the state $x = [x_1, x_2]^T$ is fully measurable and available for controller design, $a_1 < 1$, $a_2, a_3 \in [-5, 5]$ are unknown parameters, v and $w = [w_1, w_2]^T$ are standard Brownian motions, with v, w_1, and w_2 pairwise independent. Since the ranges of a_1, a_2, and a_3 are given, an initial stabilizing control law can be chosen as $u = -[20, 30]x$. For learning purpose, we add extra additive noise to the system, so $D = I_2$ is replaced by $D(t)$ which is a time-varying matrix satisfying

$$D(t) = \begin{cases} \begin{bmatrix} 0 & 20 \\ 10 & 0 \end{bmatrix}, & \frac{t}{\delta t} \in (3l, 3l+1] \\ \begin{bmatrix} 10 & 10 \\ 0 & 10 \end{bmatrix}, & \frac{t}{\delta t} \in (3l+1, 3l+2] \\ \begin{bmatrix} 10 & 0 \\ 10 & 10 \end{bmatrix}, & \frac{t}{\delta t} \in (3l+2, 3l+3] \end{cases} \tag{33}$$

where $\delta t = 100s$ and $l = 0, 1, 2, \ldots$.

The objective is to minimize the following expected cost in the steady state:

$$J = \mathcal{E}\left(x_1^2 + x_2^2 + 10u^2 \right). \tag{34}$$

The system is numerically simulated in MATLAB, with time step $dt = 0.0005\ s$. We run the program from 0 to 6000 s, and choose $T = 200\ s$, hence the control policy is updated 10 times. Also, only for the purpose of simulation, we set $a_1 = -10$, $a_2 = 0.9$, and $a_3 = 0.2$.

To illustrate the convergence of the cost matrix and the control policy, we plot their values after each iteration, as shown in Figs. 1 and 2. It can be seen that, after four iterations, we obtain $|\text{tr}(\hat{P}_{10} - P^*)| = 0.16$.

Note that, besides random noise, the approximation error may also be caused by the inaccuracy of numerically solving the integrals.

Finally, since the ADP-based controller is obtained, we apply it to the original system (32), and remove the extra additive noise so that $D(t)$ is reduced to the original constant matrix D. In Fig. 3, we plot the averaged cost

$$\hat{J}(t) = \frac{1}{t} \int_0^t \left(x_1^2(\tau) + x_2^2(\tau) + 10u^2(\tau) \right) d\tau \tag{35}$$

using the solid curve, whereas the dashed line denotes the optimal cost as in (4).

It is clearly seen that the cost is minimized.

V. Conclusion

This brief presented new results for optimal stationary control with additive and multiplicative noise. The ADP

theory was applied to relax the assumption on the knowledge of system matrices. It has been shown that the expectation of the approximated cost matrix converges to its optimal value, and the covariance can be reduced by increasing the time interval between two consecutive iterations. It is an interesting topic of current investigation to extend this result to finite-horizon optimal control for stochastic systems and also for nonlinear systems.

REFERENCES

[1] A. G. Barto and R. S. Sutton, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[2] P. J. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," Ph.D. dissertation, Committee Appl. Math., Harvard Univ., Cambridge, MA, 1974.

[3] P. J. Werbos, "Neural networks for control and system identification," in *Proc. IEEE Conf. Decis. Control*, vol. 1. Tampa, FL, Dec. 1989, pp. 260–265.

[4] P. J. Webros, *A Menu of Designs for Reinforcement Learning Over Time, Neural Networks for Control*. Cambridge, MA: MIT Press, 1990.

[5] P. J. Werbos, "Approximate dynamic programming for real-time control and neural modeling," in *Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approaches*, vol. 15. New York: Van Nostrand, 1992, pp. 493–525.

[6] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement Learning and Dynamic Programming Using Function Approximators*. Boca Raton, FL: CRC Press, 2010.

[7] X. Xu, D. Hu, and X. Lu, "Kernel-based least squares policy iteration for reinforcement learning," *IEEE Trans. Neural Netw.*, vol. 18, no. 4, pp. 973–992, Jul. 2007.

[8] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. New York: Wiley, Sep. 2007.

[9] H. Yu and D. Bertsekas, "Basis function adaptation methods for cost approximation in MDP," in *Proc. IEEE Symp. Adapt. Dyn. Program. Reinforce. Learn.*, Mar.–Apr. 2009, pp. 74–81.

[10] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE Circuits Syst. Mag.*, vol. 9, no. 3, pp. 32–50, Sep. 2009.

[11] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *Proc. Amer. Control Conf.*, vol. 3. Jul. 1994, pp. 3475–3479.

[12] D. Vrabie, O. Pastravanu, M. Abu-Khalaf, and F. L. Lewis, "Adaptive optimal control for continuous-time linear systems based on policy iteration," *Automatica*, vol. 45, no. 2, pp. 477–484, Feb. 2009.

[13] Y. Jiang and Z. P. Jiang, "Approximate dynamic programming for output feedback control," in *Proc. 29th Chin. Control Conf.*, Beijing, China, Jul. 2010, pp. 5815–5820.

[14] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 41, no. 1, pp. 14–25, Feb. 2011.

[15] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control," *Automatica*, vol. 43, no. 3, pp. 473–481, Mar. 2007.

[16] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time nonlinear HJB solution using approximate dynamic programming: Convergence proof," *IEEE Trans. Syst., Man, Cybern., Part B: Cybern.*, vol. 38, no. 4, pp. 943–949, Aug. 2008.

[17] D. Kleinman, "Optimal stationary control of linear systems with control-dependent noise," *IEEE Trans. Autom. Control*, vol. 14, no. 6, pp. 673–677, Dec. 1969.

[18] F. Wang, H. Zhang, and D. Liu, "Adaptive dynamic programming: An introduction," *IEEE Comp. Intell. Mag.*, vol. 4, no. 2, pp. 39–47, May 2009.

Reduced-Size Kernel Models for Nonlinear Hybrid System Identification

Van Luong Le, Gérard Bloch, and Fabien Lauer

***Abstract*— This brief paper focuses on the identification of nonlinear hybrid dynamical systems, i.e., systems switching between multiple nonlinear dynamical behaviors. Thus the aim is to learn an ensemble of submodels from a single set of input-output data in a regression setting with no prior knowledge on the grouping of the data points into similar behaviors. To be able to approximate arbitrary nonlinearities, kernel submodels are considered. However, in order to maintain efficiency when applying the method to large data sets, a preprocessing step is required in order to fix the submodel sizes and limit the number of optimization variables. This brief paper proposes four approaches, respectively inspired by the fixed-size least-squares support vector machines, the feature vector selection method, the kernel principal component regression and a modification of the latter, in order to deal with this issue and build sparse kernel submodels. These are compared in numerical experiments, which show that the proposed approach achieves the simultaneous classification of data points and approximation of the nonlinear behaviors in an efficient and accurate manner.**

***Index Terms*— Hybrid dynamical systems, kernel methods, sparse models, switched regression, system identification.**

I. INTRODUCTION

Hybrid dynamical systems have been extensively studied by the control community over the recent years as a potential class of dynamical models to approximate the behavior of complex cyber-physical systems. Despite this significant amount of work, the major issue of obtaining a model of the system from experimental data remains open. Formally, this problem, known as hybrid system identification [1], takes the form of a nonconvex optimization problem involving a large number of integer variables that depends on the number of data. Consequently, most proposed methods do not apply to large data sets.

More specifically, hybrid (dynamical) systems are a class of discrete-time autoregressive with external input (ARX) systems of the form (in the single-input single-output case)

$$y_i = f_{\lambda_i}(\boldsymbol{x}_i) + e_i \tag{1}$$

where $\boldsymbol{x}_i = [y_{i-1} \ \ldots \ y_{i-n_a}, \ u_{i-n_k} \ \ldots \ u_{i-n_k-n_c+1}]^T$ is the *continuous state* (or regression vector) of dimension p containing the lagged n_a outputs y_{i-k} and n_c inputs u_{i-n_k-k}, where n_k is the pure delay. The *discrete state* (or mode) $\lambda_i \in \{1, \ldots, n\}$ determines which one of the n subsystems

Manuscript received January 14, 2011; revised September 27, 2011; accepted September 28, 2011. Date of publication October 25, 2011; date of current version December 13, 2011.

V. L. Le and G. Bloch are with the Centre de Recherche en Automatique de Nancy (CRAN), Université de Lorraine, Centre National de la Recherche Scientifique, Vandoeuvre-lès-Nancy F-54500, France (e-mail: Luong.Le-Van@ensem.inpl-nancy.fr; gerard.bloch@esstin.uhp-nancy.fr).

F. Lauer is with LORIA, Université Henri Poincaré Nancy 1, Vandoeuvre-lès-Nancy F-54506, France (e-mail: fabien.lauer@loria.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2011.2171361