Online Control Basis Selection by a Regularized Actor Critic Algorithm

Jianjun Yuan and Andrew Lamperski

Abstract— **Policy gradient algorithms are useful reinforcement learning methods which optimize a control policy by performing stochastic gradient descent with respect to controller parameters. In this paper, we extend actor-critic algorithms by adding an ℓ_1 norm regularization on the actor part, which makes our algorithm automatically select and optimize the useful controller basis functions. Our method is closely related to existing approaches to sparse controller design and actuator selection, but in contrast to these, our approach runs online and does not require a plant model. In order to utilize ℓ_1 regularization online, the actor updates are extended to include an iterative soft-thresholding step. Convergence of the algorithm is proved using methods from stochastic approximation. The effectiveness of our algorithm for control basis and actuator selection is demonstrated on numerical examples.**

I. Introduction

Reinforcement learning is a class of online algorithms in which an agent learns to make optimal decisions through trial-and-error interactions with a dynamic environment [1] [2]. Many algorithms have been proposed for this problem, including actor-only methods [3] [4], critic-only methods [5] [6], and actor-critic algorithm [7] [8].

In actor-critic methods, the actor performs a stochastic gradient descent using gradient estimates computed by the critic. The result that enables the gradient estimation is known as the *policy gradient theorem* [9], which demonstrates that the gradient with respect to control policy parameters may be computed from the state-action value function. This function is closely related to the cost-to-go function from classical dynamic programming.

In policy gradient methods, such as actor critic algorithms, it is commonly assumed that the control action is computed from a linear combination of basis functions [10]. In linear quadratic Gaussian control, it is sensible to let the basis functions be linear in the state. However, for non-linear or non-Gaussian control, the selection of a good choice of basis function is not obvious, and requires extra insight into the problem. In order to automate the process of controller parameterization, we pose the problem of *control basis selection*, which aims to choose a small collection of basis functions for control. This problem is closely related to the feature selection problem, commonly studied in statistics and machine learning [11]–[13]. In reinforcement learning, regularization has been used to compute sparse state-action value / cost-to-go functions [11] [14]. In actor-critic terminology, these methods focus on designing a sparse critic, while the current paper focuses on designing a sparse actor.

The control basis selection problem studied in this paper is closely related to methods for designing sparse linear controllers [15]. As in our work, sparse design is achieved by ℓ_1 regularization. Other closely related problems involve sensor and actuator selection [16]–[20]. Sensor or actuator selection can be achieved via a sum-of-norms regularization which can be used to drive entire rows or columns of gain matrices to zero. We demonstrate that our control basis selection framework also extends to the actuator selection problem.

All of the methods including sparse design, actuator selection, or controller selection require a system model [15]–[20]. In this paper, we propose an online, model-free approach to sparse design and actuator selection. The method can be applied to to find locally optimal solutions for nonlinear and non-Gaussian systems.

To achieve online regularization for the actor critic methods, we utilize iterative soft-thresholding [21] over the controller parameters. In general, the controller parameters converge to values near a local minimum. Convergence is proved using stochastic approximation methods [22] analogous to those from [23].

A. Notation

If x is a random variable, its expected value is denoted by $\mathbb{E}[x]$. If M is a matrix, its transpose is denoted by M^T, and $M > 0$ means M is postive definite matrix. The gradient with respect to a vector θ is denoted by ∇_θ. For a vector x, its i-th item is denoted by $[x]_i$.

II. Problem Formulation and Motivation

Consider an average cost-per-stage stochastic control problem problem [24], [25]:

$$\begin{aligned} \min_{\theta\in\Theta} J(\theta) \quad &= \lim_{N\to\infty} \tfrac{1}{N}\mathbb{E}[\textstyle\sum_{k=0}^{N-1} r(x_k,u_k)] \\ s.t. \quad & x_{k+1} = f(x_k,u_k,w_k) \\ & u_k = g(\theta,x_k,v_k) \end{aligned} \tag{1}$$

where x_k is the state, u_k is the input, w_k is independent, identically distributed (iid) process noise and v_k is iid exploration noise.

Typically, the input function has the form

$$g(\theta,x_k,v_k) = \sum_{i=1}^{q} \psi_i(x_k,v_k)\theta_i. \tag{2}$$

The functions ψ_i are called basis functions, and θ is the parameter vector. Possible choices of basis functions include linear functions, sigmoids, and Gaussian radial basis functions. The set of allowable parameter variables is given by Θ.

The authors are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN, USA, 55455 yuanx270@umn.edu, alampers@umn.edu

We will assume that Θ is the Cartesian product of compact intervals

$$\Theta = [a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_q, b_q]. \qquad (3)$$

Note that if the system dynamic equation is linear-time-invariant (LTI), which can be described as $x_{k+1} = Ax_k + B_1 u_k + B_2 w_k$, this will become the traditional infinite-horizon LQR problem, which has closed-form solution as a linear relation between u_k and x_k[1].

Note that in (1), the state and input functions implicitly induce probability densities of the form $p(x_{k+1}|x_k, u_k)$ and $\pi_\theta(u_k|x_k)$. Thus, the problem can be interpreted as a Markov decision process (MDP), as studied in reinforcement learning [2].

Standard ergodicity assumptions [26] guarantee that x_k converges to a stationary distribution, which we will denote by $d_\theta(x)$. In particular, if the state space is finite, then it suffices that the Markov chain be irreducible and aperiodic. If the dynamic system and controller are linear, it suffices that the system be controllable via the noise inputs.

In the case that the x_k converges to a stationary distribution for each $\theta \in \Theta$, our original problem can be described as the following simple form:

$$\min_{\theta\in\Theta} J(\theta) = \int d_\theta(x) \int \pi_\theta(u|x) r(x,u) dx du. \qquad (4)$$

For the *control basis selection* problem, we aim to select a small number of parameters that can be used to achieve good control performance. In other words, we aim to make $J(\theta)$ small using a vector θ which is as sparse as possible. The trade-off between control performance and parameter sparsity motivates the following modified objective:

$$\min_{\theta} J(\theta) + \lambda \|\theta\|_1 \qquad (5)$$

where $\|.\|_1$ represents the ℓ_1-norm.

The ideal sparsity penalty is the ℓ_0-norm, which counts the number of non-zeros. However, minimizing the ℓ_0 norm is NP hard [27], and computationally intractable for large-scale problems. Thus, ℓ_1-norm constraint is introduced to replace the ℓ_0-norm, and it has been proved in [28] [29] that ℓ_1-norm minimization is equivalent to the ℓ_0 minimization with high probability under some technical conditions. In the next section, we will show the details on how to solve this ℓ_1-norm regularized average cost-per-stage reinforcement learning problem using a novel regularized actor-critic algorithm.

III. Regularized Actor-Critic Algorithm and Its Convergence

If $J(\theta)$ is convex in θ, the problem can be solved by the Iterative Soft-Thresholding Algorithm, which is proved to converge in [21]. The algorithm is outlined in Algorithm 1.

But for our concerned problem (5), we cannot apply the above algorithm directly, because we are assuming no plant model. In particular, $J(\theta)$ and its gradient cannot be directly computed without a model.

Algorithm 1 Algorithm to solve problem (5) with convex $J(\theta)$ and no constraint

```
Initialize $\theta^0$, $\mu < \frac{1}{L}$ where $L$ is the Lipschitz constant of $\nabla_\theta J$
while not converge do
    1. $z^k = \theta^{k-1} - \mu \nabla_\theta J(\theta^{k-1})$;
    2. $[\theta^k]_i = [z^k]_i - \mu\lambda$, if $[z^k]_i \geq \mu\lambda$;
    3. $[\theta^k]_i = [z^k]_i + \mu\lambda$, if $[z^k]_i \leq -\mu\lambda$;
    4. $[\theta^k]_i = 0$, otherwise;
end while
```

However, we will see that Algorithm 1 can be combined with policy gradient methods to solve problem (5), using approximate gradients. Previous work [9] has shown that under ergodicity assumptions, the gradient is given by:

$$\nabla_\theta J(\theta) = \int d_\theta(x) \int \nabla_\theta \pi_\theta(u|x) Q_\theta(x,u) dx du. \qquad (6)$$

Here $Q_\theta(x,u)$ is called the differential action-value function and is defined as:

$$Q_\theta(x,u) = \sum_{k=0}^{\infty} \mathbb{E}[r(x_k,u_k) - J(\theta)|x_0 = x, x_0 = x, \theta] \quad (7)$$

From [30], the gradient of the average cost can also be written as:

$$\nabla_\theta J(\theta) = \int d_\theta(x) \int \nabla_\theta \pi_\theta(u|x)[Q_\theta(x,u) \pm b(x)] dx du. \qquad (8)$$

where $b(x)$ is any function of the state only. The reason is from the following equation:

$$\int \nabla_\theta \pi_\theta(u|x) du = \nabla_\theta \int \pi_\theta(u|x) du = \nabla_\theta 1 = 0 \qquad (9)$$

As is shown in [23], to obtain the minimum variance baseline of the estimated gradient, $b(x)$ can be set to be equal to $V_\theta(x) = \int \pi(u|x) Q_\theta(x,u) du$.

Define the advantage function, by $A_\theta(x,u) = Q_\theta(x,u) - V_\theta(x)$. Using the identity, $\nabla_\theta \pi_\theta(u|x) = \pi_\theta(u|x) \nabla_\theta \log \pi_\theta(u|x)$, the gradient has the following form:

$$\nabla_\theta J(\theta) = \int d_\theta(x) \int \pi_\theta(u|x) \nabla_\theta \log \pi_\theta(u|x) A_\theta(x,u) dx du \qquad (10)$$

It follows that for a fixed value of θ, $\nabla_\theta \log \pi_\theta(a|s) A_\theta(s,a)$ is an unbiased estimate of the gradient $\nabla_\theta J(\theta)$.

The lemma below shows how to estimate the advantage function $A_\theta(s,a)$ from the critic part as shown in [23].

Lemma 1: Let $\hat{J}_{k+1}$ and $\hat{V}(x)$ be unbiased estimates of $J(\theta)$ and $V_\theta(x)$, respectively. Define the temporal difference error by

$$\delta_k = r(x_k,u_k) - \hat{J}_{k+1} + \hat{V}(x_{k+1}) - \hat{V}(x_k) \qquad (11)$$

Under the given policy π_θ, we have

$$\mathbb{E}[\delta_k | x_k, u_k, \theta] = A_\theta(x_k, u_k) \qquad (12)$$

The proof can be found in [23]. With the Lemma 1, we can get an unbiased estimate of the gradient to be equal to $\nabla_\theta \log \pi_\theta(u_k|x_k)\delta_k$.

[1] This assumes that the constraint set Θ is sufficiently big to contain the optimal controller.

The term $\nabla_\theta \log \pi_\theta(u_k|x_k)$ can be calculated, because we know the current state x_k, the current action u_k, and the policy configuration parameterized by θ.

For the temporal difference error δ_k, we need to know $r(x_k, u_k)$, and estimate $J(\theta)$ and $V(x)$. The term $r(x_k, u_k)$ is the known single stage cost. The cost $J(\theta)$ can be estimated by taking averages of the single-stage cost.

The biggest estimation challenge is for the state-value function V. For this term, we utilize function approximation. As in [31], we use linear basis expansion $V_\theta(x, v) \approx v^T \phi(x)$. If the state space is finite, with n states, this can also be written as $V_\theta(v) \approx \Phi v$, where Φ is an $n \times d$ matrix whose kth column ($k = 1, 2, ..., d$) is $\phi(k)$.

The parameter vector v is estimated through the temporal difference learning which is updated in the critic part. Interested readers can refer to [2] for more technical details.

For the learning algorithm, let α_k, β_k, and η_k be step sizes that satisfy the same requirements as in [23].

The incremental regularized actor-critic algorithm that is used to solve our problem is summarized in Algorithm 2.

The function Γ_i clips the value to remain in the interval, $[a_i, b_i]$ as required from (3).

The critic update part in Algorithm 2 can alternatively utilize least-squares temporal difference learning, as in [32] [33]. Due to space limits, we omit the details on this variation.

Algorithm 2 Incremental Regularized Actor Critic Algorithm

```
Initialize state $x_0$ from $p(x_0)$, function approximation parameter $v_0$, policy parameter $\theta_0$, and the eligibility vector $\chi_0 = 0$
Initialize step sizes $\alpha_0$, $\beta_0$, $\eta_0$, the decay parameter $\gamma$, and the sparsity regularization parameter $\lambda$
for k = 0,1,2,... do
    1. Obtain the action $u_k$ from $g(\theta_k, x_k, v_k)$;
    2. Observe the next state $x_{k+1}$ from $f(x_k, u_k, w_k)$;
    3. Observe the single stage cost $r_k = r(x_k, u_k)$;
    4. Average cost update: $J_{k+1} = (1 - \eta_k) J_k + \eta_k r_k$;
    5. Calculate the Temporal Difference error: $\delta_k = r_k - J_{k+1} + v_k^T \phi(x_{k+1}) - v_k^T \phi(x_k)$;
    6. Critic update:
       $v_{k+1} = v_k + \alpha_k \delta_k \chi_k$;
       Eligibility vector update: $\chi_{k+1} = \gamma \chi_k + \phi(x_k)$;
    7. Actor update: $y = \theta_k - \beta_k \nabla_\theta \log \pi_\theta(u_k|x_k) \delta_k$,
       $[\theta_{k+1}]_i = \begin{cases} \Gamma_i\Big([y]_i - \lambda\beta_k\Big) & \text{if } [y]_i \geq \lambda\beta_k \\ \Gamma_i\Big([y]_i + \lambda\beta_k\Big) & \text{if } [y]_i \leq -\lambda\beta_k \\ 0 & \text{otherwise} \end{cases}$
end for
```

IV. Convergence

This section discusses convergence of the algorithm. For simplicity, our convergence proof focuses on the finite state case. Thus, we will use sums instead of integrals to emphasize the distinction. Extension to the general case should be possible using methods from [26], [34]. Before presenting the convergence theorem, there are some assumptions that need to be clarified as described in [35].

In order to prove the convergence of the actor update, we extend the method of [23]. For any set-valued vector field, $h(\theta)$ Define the new vector field $\hat{\Gamma}$ by:

$$\hat{\Gamma}(h(\theta)) = \lim_{0<\eta\to 0} \Big(\frac{\Gamma(\theta + \eta h(\theta)) - \theta}{\eta}\Big) \tag{13}$$

Then consider the differential inclusion associated with the subgradient of Problem (5):

$$\dot{\theta} \in \hat{\Gamma}(-\nabla_\theta J(\theta) - \lambda g(\theta)) \tag{14}$$

where $g(\theta)$ is the subgradient of $\|\theta\|_1$ defined as:

$$[g(\theta)]_i = \begin{cases} +1 & \text{if } [\theta]_i > 0 \\ -1 & \text{if } [\theta]_i < 0 \\ [-1, 1] & \text{otherwise} \end{cases} \tag{15}$$

Let Λ be the set of the asymptotically stable equilibria for (14), i.e., the local minima of Problem (5), and let Λ^ϵ be the ϵ neighborhood of Λ, i.e., $\Lambda^\epsilon = \{\hat{\theta} | \left\|\hat{\theta} - \theta^*\right\| < \epsilon, \theta^* \in \Lambda\}$.

The function approximator can introduce bias into the estimates of $V_\theta(x)$ and its gradient. The bias can be quantified using time-scale separation. Specifically, if the critic converges faster than the actor, then for each fixed θ, the parameter v_k converges to a steady steady value v_∞. The temporal difference after critic convergence is given by:

$$\hat{\delta}_k = r(x_k, u_k) - \hat{J}_k + v_\infty^T \phi(x_{k+1}) - v_\infty^T \phi(x_k). \tag{16}$$

According to [23], the estimated gradient using $\hat{\delta}_k$ is given by:

$$\mathbb{E}[\hat{\delta}_k \nabla_\theta \log \pi_\theta(u_k|x_k)] = \nabla_\theta J(\theta) + b_\theta, \tag{17}$$

where b_θ is a bias term.

Theorem 1: For any $\epsilon > 0$*, there exists* $\zeta > 0$ *such that* $\sup_{\theta_k} \|b_{\theta_k}\| < \zeta$*, such that* θ_k *will converge to* Λ^ϵ *as* $k \to \infty$ *with probability one.*

Proof: First, we need a background lemma, which is proved in [35]. In this lemma P denotes the transition matrix induced by parameter θ: $P_{i,j} = p_\theta(x_{k+1} = j | x_k = i)$, and e denotes the vector of all ones.

Lemma 2: For any $\gamma \in [0, 1)$*, the average cost update of* J_k *and the critic update of* v_k *in Algorithm 2 converge to* $J(\theta)$ *and* v_θ *with probability one, respectively, where*

$$J(\theta) = \sum_{x \in \mathcal{X}} d_\theta(x) \sum_{u \in \mathcal{U}} \pi_\theta(u|x) r(x, u) \tag{18}$$

is the average cost under policy π_θ*, and* v_θ *is the unique solution of the equation*

$$\Pi T^\gamma(\Phi v_\theta) = \Phi v_\theta \tag{19}$$

where $T^\gamma(\Phi v)$ *is defined as:*

$$T^\gamma(\Phi v) = (1 - \gamma) \sum_{m=0}^{\infty} \gamma^m \Big(\sum_{k=0}^{m} P^k (r - J(\theta) e) + P^{k+1} \Phi v \Big) \tag{20}$$

Thus, according to Lemma 2, our average cost update and critic update will converge to the above value with probability one.

The analysis below mainly follows the ordinary differential equation (ODE) method [22]. Recall that the function approximation introduces bias into the gradient estimate, which is described in (17). To analyze this biased estimate, define a separate differential inclusion by:

$$\dot{\theta} = \hat{\Gamma}(-\nabla_\theta J(\theta) - \lambda g(\theta) - b_\theta), \tag{21}$$

and let Ξ be the set of the asymptotically stable equilibria of (21).

Like the ODE method used in [23], we have the following equation:

$$\begin{aligned} \theta_{k+1} = \quad & \Gamma\Big(\theta_k - \beta_k \mathbb{E}[\hat{\delta}_{k,\infty} \nabla_\theta \log \pi_\theta(u_k|x_k)|\mathcal{F}_1(k)] \\ & -\beta_k \omega_1(k) - \beta_k \omega_2(k) + g(\theta_{k+1})\lambda\Big) \end{aligned} \tag{22}$$

where $\mathcal{F}_1(k) = \sigma(\theta_l, l < k)$ denotes the sequence of σ-fields generated by θ_l,$l \geq 0$, and $\omega_1(k)$, $\omega_2(k)$ are defined in (23) and (24):

$$\omega_1(k) = \hat{\delta}_k \nabla_\theta \log \pi_\theta(u_k|x_k) - \mathbb{E}[\hat{\delta}_k \nabla_\theta \log \pi_\theta(u_k|x_k)|\mathcal{F}_1(k)] \tag{23}$$

$$\omega_2(k) = \mathbb{E}[(\hat{\delta}_k - \delta_k)\nabla_\theta \log \pi_\theta(u_k|x_k)|\mathcal{F}_1(k)] \tag{24}$$

where δ_k and $\hat{\delta}_k$ are the temporal differences from the algorithm and (16), respectively.

By the step size assumptions, the critic will converge faster than the actor, so $\omega_2(k) = o(1)$. And let $M^1(k) = \sum_{l=0}^{k-1} \beta_l \omega_1(l)$, $k \geq 1$. Then according to [23], $\{M^1(k)\}$ is a convergent martingale sequence, and $\sum_{l=n}^{n_T} \beta_l \omega_2(l) \to 0$ as $n \to \infty$ for $n_T = \min\{m \geq n | \sum_{l=n}^{m} \beta_l \geq T\}$ with any $T > 0$. The argument in [23] shows that $\mathbb{E}[\hat{\delta}_k \nabla_\theta \log \pi_\theta(u_k|x_k)|\mathcal{F}_1(k)]$ converges to $\nabla_\theta J(\theta)$ as $\sup_{\theta_k} \|b_{\theta_k}\| \to 0$. Thus, the solutions (21) will converge to those of (14) as the bias goes to 0. □

V. Numerical Examples

In this section, we will use a simple example to show the effectiveness of our algorithm. Consider the problem:

$$\begin{aligned} \min_{\theta} \quad & J(\theta) + \lambda \|\theta\|_1 \\ s.t. \quad & x_{k+1} = 0.9x_k + 0.05(u_k + 0.1w_k) \\ & u_k = g(\theta, x_k, v_k) \end{aligned} \tag{25}$$

where $J(\theta) = \lim_{k\to\infty} \mathbb{E}[x_k^T Q x_k + u_k^T R u_k]$, Q = 10, R = 0.1.

First, we will show the effectiveness of the Actor Critic algorithm in finding the optimal solution when $\lambda = 0$, and controller configuration is $u_k = \theta_1 x_k + \theta_2 v_k$.

According to [36], the optimal solution for this LQR problem has the optimal linear form $u_k = Kx_k$ with optimal gain $K^* = -6.24$. Thus, the optimal solution $\theta^* = [\theta_1^* \quad \theta_2^*]^T$ is $\theta_1^* = -6.24$ and $\theta_2^* = 0$.

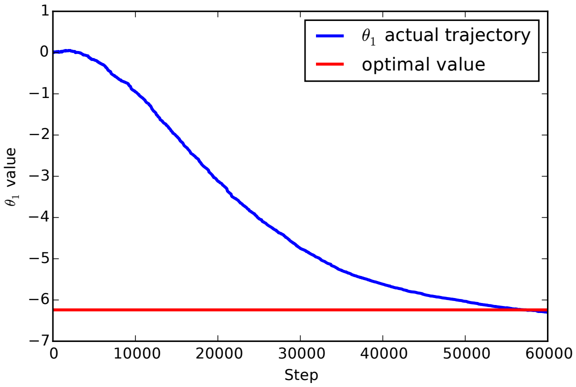

Fig. 1. θ_1 trajectories by Algorithm 2 and the optimal value line

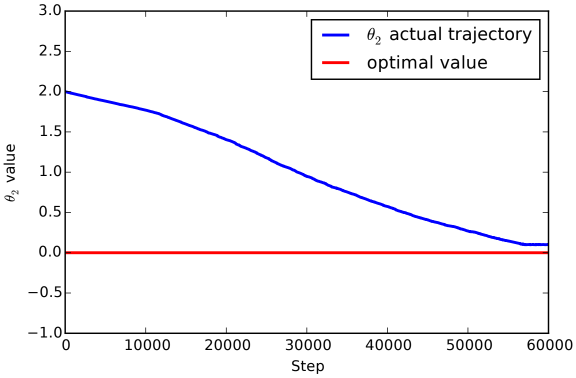

Fig. 2. Exploration noise variable θ_2 trajectories by Algorithm 2 and the optimal value line. The noise is constrained to a small positive value, to which it converges.

We use Algorithm 2 to solve this problem. The parameter configurations are as follows: $\theta_0 = [0 \quad 2]^T$, x_0 is randomlly obtained from the Gaussian distribution with mean equal to 0 and variance equal to 2. $\alpha_k = \frac{0.5}{10\ ^4k+30}$, $\beta_k = \frac{0.3}{10\ ^2k+90}$, $\eta_k = \frac{1}{k+1}$, and the decay parameter $\gamma = 0.9$.

θ_1 and θ_2 trajectories are shown in Fig.1 and Fig.2, respectively. From Fig.1 and 2, we can see that θ_1 and θ_2 ultimately converge near the optimal value.

Next, we use linear-nonlinear mixed controller configuration with the form:

$$\begin{aligned} u_k = \quad & \tilde{\varphi}(x_k, v_k)^T \theta \\ \tilde{\varphi}(x_k, v_k)^T = \quad & [\varphi(x_k)^T \quad x_k^T \quad v_k^T] \end{aligned} \tag{26}$$

And each item in the vector $\varphi(x_k)$ is defined as:

$$[\varphi(x_k)]_i = \exp(-\frac{(x_k - \iota_i)^T (x_k - \iota_i)}{l}) \tag{27}$$

where in our experiment, we set $l = 2$, and ι is generated uniformly in the range $[-10, 10]$ with 15 items.

The parameter setup is the same except for the stepsizes: $\alpha_k = \frac{1}{10\ ^4k+100}$, $\beta_k = \frac{1}{10\ ^2k+300}$, and $\lambda = 0.1$. θ trajectories are shown in Fig.3 and 4, respectively. From Fig.3, we can see final solution that is obtained is where there are only 3 non-zero variables. The largest value for θ, by far,

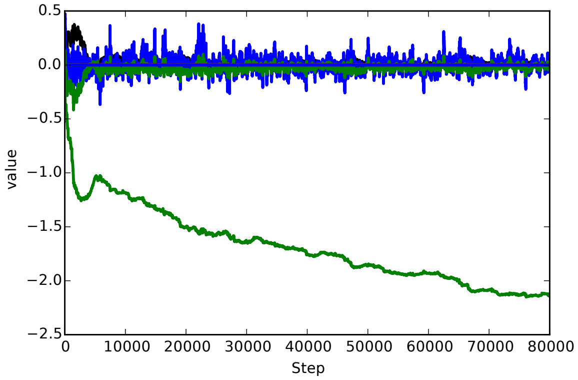

Fig. 3. θ trajectories except exploration noise variable by Algorithm 2

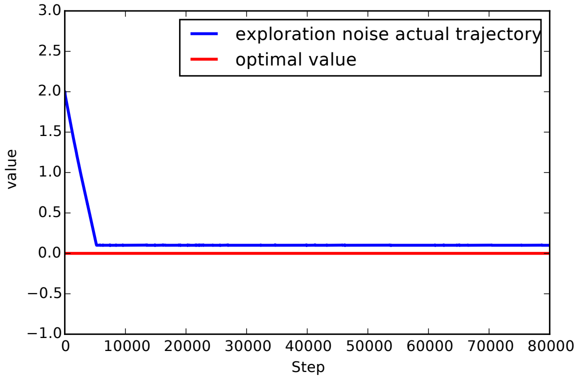

Fig. 4. Exploration noise variable trajectories by Algorithm 2

corresponds to the linear term. This makes sense, since the original problem has a linear linear optimal solution.

Finally, we will vary λ to see the process for the number of non-zeros obtained by our algorithm. This result is shown in Fig.5. We can see from Fig.5 that when λ increases, the number of non-zeros decreases as it should do.

Modification for Actuator Selection: We consider the regularized LQR problem for actuator selection, as in [20]. In this problem, a group-Lasso regularization [37] is used to select columns of a gain matrix for state feedback. The problem is given formally by:

$$\begin{aligned} &\min_{\theta} J(\theta) + \lambda \sum_{l=1}^{L} \|\theta_l\|_2 \\ &s.t. \quad x_{k+1} = Ax_k + B_1 u_k + B_2 w_k \end{aligned} \tag{28}$$

where θ_l represents the items belonging to group l. Note that in this problem, the number of groups L is equal to the number of actuators.

Our algorithm admits a straightforward modification to implement group-Lasso regularization in place of the standard Lasso. The primary change is that the subgradient is taken for the sum-of-norms penalty, rather than the ℓ_1 penalty.

We tested the modification using system matrices: $Q = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$, $R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$, $A = \begin{bmatrix} 0.72 & 0.5 \\ 0 & 0.05 \end{bmatrix}$, $B_1 = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix}$, $B_2 = \begin{bmatrix} 0.005 & 0 \\ 0 & 0.005 \end{bmatrix}$. According to [36], the optimal gain matrix is: $K^* = \begin{bmatrix} 4.02 & 2.84 \\ 0.80 & 0.80 \end{bmatrix}$

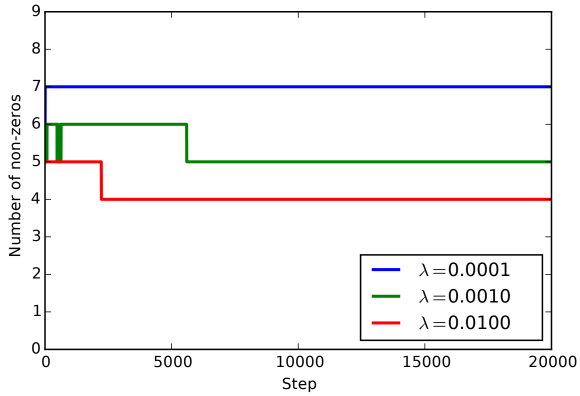

Fig. 5. Number of non-zero variables' trajectories by Algorithm 2 for different λ

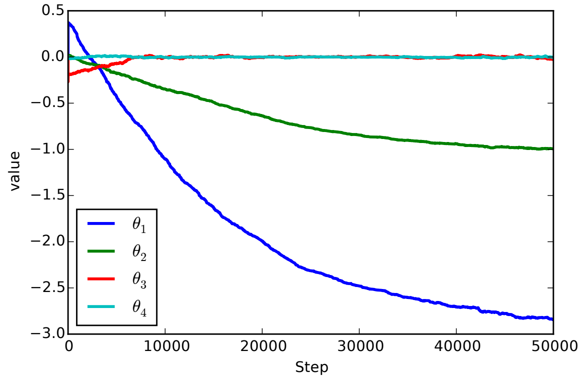

Fig. 6. θ trajectories except exploration noise variable

The Actuator Selection Problem (28) is solved with $\lambda = 0.01$, stepsizes: $\alpha_k = \frac{0.7}{10 \;\; ^4k+100}$, $\beta_k = \frac{0.5}{10 \;\; ^2k+300}$, and the decay parameter $\gamma = 0.9$. The controller configuration is: $u_k = \varphi(x_k, v_k)^T\theta$, where $\varphi(x_k, v_k) = \begin{bmatrix} x_k^T & 0 & 0 \\ 0 & x_k^T & 0 \\ 0 & 0 & v_k \end{bmatrix}$ In this case, the groups are given by:

$$\begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}, \quad \begin{bmatrix} \theta_3 \\ \theta_4 \end{bmatrix}, \quad \theta_5, \tag{29}$$

where θ_5 corresponds to exploration noise.

Fig.6 and 7 show the trajectories for our parameter θ. We can see that θ_3 and θ_4 stay near 0 all the time, while θ_1 and θ_2 reach to some non-zero values, which means that it chooses the first actuator. We also run the algorithm in [20], which gives us the same actuator selection result.

VI. Conclusion and Future Work

In this paper, we deal with the problem of control basis selection in the case when the system model is unavailable. We propose a regularized actor-critic algorithms to select and optimize the sparse controller configuration. Furthermore, possible modification is performed to solve the actuator

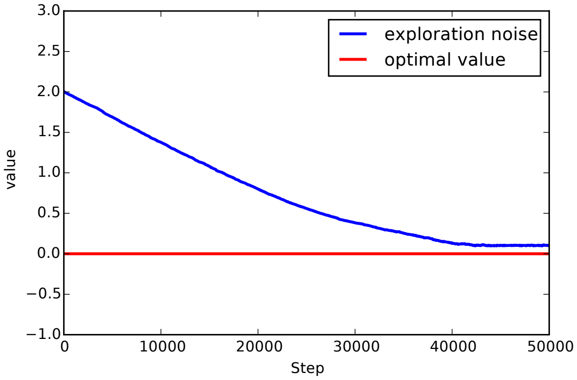

Fig. 7. Exploration noise variable trajectories

selection problem. Numerical examples show the effectiveness of our proposed algorithms. In the future, we plan to accelerate the gradient estimation to reduce the estimate's variance. Our methodology may also be useful for online sensor selection.

References

[1] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *Journal of artificial intelligence research*, vol. 4, pp. 237–285, 1996.

[2] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*, vol. 1. MIT press Cambridge, 1998.

[3] P. W. Glynn, "Stochastic approximation for monte carlo optimization," in *Proceedings of the 18th conference on Winter simulation*, pp. 356–365, ACM, 1986.

[4] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, no. 3-4, pp. 229–256, 1992.

[5] S. J. Bradtke, B. E. Ydstie, and A. G. Barto, "Adaptive linear quadratic control using policy iteration," in *American Control Conference, 1994*, vol. 3, pp. 3475–3479, IEEE, 1994.

[6] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[7] V. R. Konda and V. S. Borkar, "Actor-critic–type learning algorithms for markov decision processes," *SIAM Journal on control and Optimization*, vol. 38, no. 1, pp. 94–123, 1999.

[8] V. R. Konda and J. N. Tsitsiklis, "Actor-critic algorithms.," in *NIPS*, vol. 13, pp. 1008–1014, 1999.

[9] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation.," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 12, pp. 1057–1063, Citeseer, 1999.

[10] L. Busoniu, R. Babuska, B. De Schutter, and D. Ernst, *Reinforcement learning and dynamic programming using function approximators*, vol. 39. CRC press, 2010.

[11] J. Z. Kolter and A. Y. Ng, "Regularization and feature selection in least-squares temporal difference learning," in *Proceedings of the 26th annual international conference on machine learning*, pp. 521–528, ACM, 2009.

[12] K. Lounici, M. Pontil, S. Van De Geer, and A. B. Tsybakov, "Oracle inequalities and optimal inference under group sparsity," *The Annals of Statistics*, pp. 2164–2204, 2011.

[13] P. J. Bickel, Y. Ritov, and A. B. Tsybakov, "Simultaneous analysis of lasso and dantzig selector," *The Annals of Statistics*, pp. 1705–1732, 2009.

[14] B. A. Pires and C. Szepesvári, "Statistical linear estimation with penalized estimators: an application to reinforcement learning," *arXiv preprint arXiv:1206.6444*, 2012.

[15] F. Lin, M. Fardad, and M. R. Jovanović, "Design of optimal sparse feedback gains via the alternating direction method of multipliers," *IEEE Transactions on Automatic Control*, vol. 58, no. 9, pp. 2426–2431, 2013.

[16] N. Matni and V. Chandrasekaran, "Regularization for design," in *53rd IEEE Conference on Decision and Control*, pp. 1111–1118, IEEE, 2014.

[17] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2009.

[18] V. Roy, S. P. Chepuri, and G. Leus, "Sparsity-enforcing sensor selection for doa estimation," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on*, pp. 340–343, IEEE, 2013.

[19] S. Kondoh, C. YATOMI, and K. Inoue, "The positioning of sensors and actuators in the vibration control of flexible systems.," *JSME international journal. Ser. 3, Vibration, control engineering, engineering for industry*, vol. 33, no. 2, pp. 145–152, 1990.

[20] N. K. Dhingra, M. R. Jovanović, and Z.-Q. Luo, "An admm algorithm for optimal sensor and actuator selection," in *53rd IEEE Conference on Decision and Control*, pp. 4039–4044, IEEE, 2014.

[21] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on pure and applied mathematics*, vol. 57, no. 11, pp. 1413–1457, 2004.

[22] H. Kushner and G. G. Yin, *Stochastic approximation and recursive algorithms and applications*, vol. 35. Springer Science & Business Media, 2003.

[23] S. Bhatnagar, R. S. Sutton, M. Ghavamzadeh, and M. Lee, "Natural actor–critic algorithms," *Automatica*, vol. 45, no. 11, pp. 2471–2482, 2009.

[24] H. Kwakernaak and R. Sivan, *Linear optimal control systems*, vol. 1. Wiley-interscience New York, 1972.

[25] D. P. Bertsekas, *Dynamic Programming and Optimal Control: Approximate Dynamic Programming*, vol. 2. Athena Scientific, 2012.

[26] S. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*. New York, NY, USA: Cambridge University Press, 2nd ed., 2009.

[27] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM journal on computing*, vol. 24, no. 2, pp. 227–234, 1995.

[28] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE transactions on information theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[29] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal ??1-norm solution is also the sparsest solution," *Communications on pure and applied mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[30] E. Greensmith, P. L. Bartlett, and J. Baxter, "Variance reduction techniques for gradient estimates in reinforcement learning," *Journal of Machine Learning Research*, vol. 5, no. Nov, pp. 1471–1530, 2004.

[31] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 674 – 690, 1997.

[32] J. A. Boyan, "Least-squares temporal difference learning," in *ICML*, pp. 49–56, Citeseer, 1999.

[33] J. Peters, S. Vijayakumar, and S. Schaal, "Natural actor-critic," in *Machine Learning: ECML 2005*, pp. 280–291, Springer, 2005.

[34] V. R. Konda and J. N. Tsitsiklis, "Onactor-critic algorithms," *SIAM journal on Control and Optimization*, vol. 42, no. 4, pp. 1143–1166, 2003.

[35] J. N. Tsitsiklis and B. Van Roy, "Average cost temporal-difference learning," *Automatica*, vol. 35, no. 11, pp. 1799–1808, 1999.

[36] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice Hall, 1996.

[37] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 68, no. 1, pp. 49–67, 2006.