Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Learning Algorithm for LQG Model With Constrained Control

Yankai Xu * **Xi Chen** **

* *Center for Intelligent and Networked Systems (CFINS), Dept. of Automation, Tsinghua Univ., Beijing, 100084, China. (Tel: 86-10-62792491; email: xuyankai99@mails.thu.edu.cn).*
** *CFINS, Dept. of Automation, Tsinghua Univ., Beijing, 100084, China. (Tel: 86-10-62783946; email: bjchenxi@tsinghua.edu.cn).*

Abstract: The paper considers a discrete-time linear quadratic Gaussian model with constrained control. It is formulated with Markov systems. With the derivative equation, a performance gradient with respect to control parameters is estimated from a sample path. Then a learning algorithm is proposed to obtain a suboptimal feedback policy in affine linear form. The learning algorithm can be implemented on-line. Its improving feature makes the algorithm attain better performance than existing approaches, and the idea can be applied to more general cases.

1. INTRODUCTION

Discrete-time optimal control systems, especially linear quadratic systems, with constrained control have been widely studied in recent years [11, 19, 5]. Control constraints may represent hard physical limits of the equipment or environmental and/or safety guidelines. The general approach to tackle such problems is dynamic programming, but it is time consuming and hard to be implemented for practical systems. For systems with stochastic disturbance, optimal control problem becomes more difficult. Chmielewski and Manousiouthakis [12] provide a short literature review on constrained linear quadratic optimal control problem with stochastic disturbance, and discuss some results of certainty equivalence between different types of models, e.g. open-loop feedback policy and close-loop feedback policy.

There are a number of suboptimal strategies for stochastic *linear quadratic regulator* (LQR) problem with constrained control. A simple policy is based on the optimal LQR for unconstrained problem (OLQRU). When applying the policy, if the control variable exceeds a control set, then restrict it to the set with a projection function (given in details in Section 2). When the variance of noise is rather small compared with control constraints, the simple policy may be good enough, but not for general cases. By using a truncated Taylor series to approximate the expected function loss in the Bellman equation, Toivonen [22] provides an affine linear form suboptimal policy. Lee and Cooley [14] suggest an open-loop optimal policy as a suboptimal policy. Following their suggestion, Perez et al. [17] give a suboptimal policy called certainty equivalent control (CEC), in which, the control adopted at each stage is the optimal control of an associated deterministic problem. Batina et al. [2] present an algorithm which can solve the constrained control problem approximately with arbitrary accuracy, however, the expected value function is computed with a randomized algorithm which yields a high computational complexity.

The approximate dynamic programming (ADP) [23, 7, 20, 21, 1] is an approach to solve dynamic programming by using simulation and function approximation to alleviate the curse-of-dimensionality. In the framework of actor-critic network, neural networks are applied to approximate the value function and the control policy. Most ADP based approaches focus on approximation of the value function, and improvement of the control policy often requires simulation when dealing with stochastic systems.

In this paper, we consider a discrete-time *linear quadratic Gaussian* (LQG) problem with constrained control. We provide a learning method which is easy for on-line implementation for a suboptimal policy. The approach belongs to the field of ADP and is called policy gradient [4] method or perturbation analysis (PA) [8]. Our approach focuses on optimization with an actor network, while the tuning of a parameterized value function is bypassed. Marbach and Tsitsiklis [15] propose a similar algorithm for Markov reward processes. It deals with discrete state space and estimates relative value with regenerative period. In continuous state space case, by estimating relative value by truncation, our approach handles the constrained optimal control problems well. We first fix a feedback policy in affine linear form with parameters unknown. So the original policy optimization problem reduces to a parameter optimization problem. Secondly, based on perturbation analysis theory in *Markov decision processes* (MDPs), the performance gradient with respect to the parameters can be estimated from sample path. Then the parameters are updated with a gradient descent method so that a better control policy is obtained. Repeating this procedure, the approach delivers an ϵ-optimal policy with arbitrary accuracy in affine linear form. Hence it can achieve better performance than many other suboptimal polices in affine linear form. Many approaches in ADP require exploratory experiments, which may intervene the normal operation

[1] Partially supported by the National Natural Science Foundations (60574064 and 60736027) of China.

of the system. Therefore these approaches are purely simulation-based algorithms. Our approach is not only a simulation-based method, but also can be implemented on-line. It needs not exploratory experiments, and the control policy can be improved step by step as the real system is running. The on-line property makes our approach more flexible for practical applications.

The paper is organized as follows. Problem formulation is given in Section 2. In Section 3, an MDP-based formulation is introduced. With derivative equation, estimates of performance gradient, and the corresponding gradient-based algorithm are given. In Section 4, we consider two special cases: scalar control set case and positive control set case. In Section 5, numerical examples and comparison with other approaches are presented. Conclusion is given in Section 6.

2. FORMULATION

Consider a stochastic linear system:

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + w_k \\ \text{s.t. } u_k &\in U, k = 0, 1, \cdots, \end{aligned} \tag{1}$$

where k is discrete time epoch, $x_k \in \mathbb{R}^n$ is system state, u_k is control variable, and $U \subset \mathbb{R}^m$ is a closed and convex set which contains the origin. $w_k \in \mathbb{R}^n$ is an n-dimensional i.i.d. normal distributed random variable with mean zero and covariance ϕ. ϕ is a positive definite matrix. Denote the probability density function of random variable w as

$$p_w(x) = \frac{1}{(2\pi)^{n/2}|\phi|^{1/2}} \exp\left\{ -\frac{1}{2} x^{\mathrm{T}} \phi^{-1} x \right\}, \tag{2}$$

where $|\phi|$ is the determinant of ϕ. A and B are matrices with suitable dimensions. It is well known (see [18]) that such a system can be globally asymptotically stabilized via feedback, only in the case that all eigenvalues of the system matrix A lie on or inside the unit circle. In the paper, we assume this condition to be satisfied.

The criterion to be minimized is long-run average performance

$$\eta = \lim_{N\to\infty} \mathrm{E}\{\frac{1}{N} \sum_{k=0}^{N-1} f(x_k, u_k)\}, \tag{3}$$

where cost function $f(x, u) = x^{\mathrm{T}} C x + u^{\mathrm{T}} D u$. Assume C and D are positive semi-definite matrices. For a stable system, (3) exists. Define a projection function as

$$\Pi_U(v) = \begin{cases} v & \text{if } v \in U, \\ \arg\min\limits_{\mu\in U} ||v - \mu|| & \text{if } v \notin U. \end{cases} \tag{4}$$

The admissible control adopted at time k is $u_k = \Pi_U(\varphi(x_k))$, where $\varphi(x)$ is a function of state (called a feedback control policy). Our objective is to find a control policy $\varphi(x), \forall x \in \mathbb{R}^n$ such that $u = \Pi_U(\varphi(x))$ minimizes (3). We consider only the control policies which stabilize the system.

In this paper, we assume the control policy $\varphi(x)$ to be affine linear in x: $\varphi(x, \alpha, \beta) = \alpha x + \beta$, where α and β are parameters to be optimized. For linear quadratic system, affine linear control policy is a reasonable and widely used approximation [22, 5]. Let $\mathrm{Vec}\{M\}$ represent the vectorization of matrix M by concatenating all the columns of M, and "$\otimes$" represent Kronecker tensor product. Let $\theta = [\mathrm{Vec}(\alpha)^{\mathrm{T}}\ \beta^{\mathrm{T}}]^{\mathrm{T}}$. θ is an $(mn + m)$-dimensional vector. Then we have

$$\varphi(x, \alpha, \beta) = \varphi(x, \theta) = F\theta, \tag{5}$$

where $F = [x^{\mathrm{T}}\ 1] \otimes I_m$, and I_m is $m \times m$ identity matrix. Under a given parameter θ, system dynamic (1) becomes

$$x_{k+1} = Ax_k + B\Pi_U(F\theta) + w_k. \tag{6}$$

The original policy optimization problem reduces to a parameter optimization problem to find out the optimal parameter θ^* to minimize the performance criterion (3).

3. LEARNING ALGORITHMS

In the section, we present an iterative approach to optimize θ. At each iteration round, a control policy $\varphi(x, \theta)$ is applied to the system subject to the control set U, then the system evolves. By observing system behaviors, performance gradient with respect to θ can be estimated from sample path. Then θ is updated with a gradient descent algorithm and a better control policy is obtained. The key point of our approach lies on estimation of the performance gradient from a sample path. In the following we first present the learning algorithm for the performance gradient.

3.1 Performance gradient

It is well known that dynamic systems can be formulated as Markov processes. Define the transition function from state x to a Borel set $\mathcal{B}$ as $P(\mathcal{B}|x)$, and for ergodic systems [16], there is a unique invariant probability measure $\pi(\mathcal{B})$ such that

$$\pi(\mathcal{B}) = \int_{x\in\mathbb{R}^n} \pi(\mathrm{d}x) P(\mathcal{B}|x). \tag{7}$$

For any measurable function $f(x)$, define transition operator $\mathcal{P}$ as

$$\mathcal{P}f(x) = \int_{y\in\mathbb{R}^n} P(\mathrm{d}y|x) f(y). \tag{8}$$

It represents the expected cost at the next time epoch, if the current state is x, i.e., $\mathcal{P}f(x) = \mathrm{E}\{f(x_1)|x_0 = x\}$. The k-step transition operator is $\mathcal{P}^k$:

$$\mathcal{P}^k f(x) = \int_{y\in\mathbb{R}^n} P(\mathrm{d}y|x) \mathcal{P}^{k-1} f(y)$$

with $\mathcal{P}^0 f(x) = f(x)$. Then $\mathcal{P}^k f(x) = \mathrm{E}\{f(x_k)|x_0 = x\}$. Define an operator $\mathcal{A}$ as

$$\mathcal{A}f(x) = \sum_{k=0}^{\infty} (\mathcal{P}^k f(x) - \eta). \tag{9}$$

with η being long-run average performance defined in (3). LQG model is an ergodic system, and the above integrals and limitation exist [13]. For more details about transition

operator and invariant probability measure of dynamic systems, one can refer to [13, 24].

From (6) we have

$$P(\mathrm{d}y|x) = p_w(y - Ax - B\Pi_U(F\theta))\mathrm{d}y. \tag{10}$$

The transition function depends on the control parameter θ, thus it is written as $P(\mathcal{B}|x,\theta)$. For simplicity, we write cost function $f(x,u) = f(x,\Pi_U(F\theta))$ as $f(x,\theta)$. Therefore, the original problem is equivalent to an MDP with transition function $P(\mathcal{B}|x,\theta)$ and cost function $f(x,\theta)$. The performance of the MDP is written as $\eta(\theta)$. From [8, 24], we have the performance derivative equation with respect to the ith element of θ, θ_i:

$$\frac{\partial\eta(\theta)}{\partial\theta_i} = \int_{x\in\mathbb{R}^n} \pi(\mathrm{d}x)\Big[\int_{y\in\mathbb{R}^n} \frac{\partial P(\mathrm{d}y|x,\theta)}{\partial\theta_i} g(y,\theta) + \frac{\partial f(x,\theta)}{\partial\theta_i}\Big], \quad i = 1,2,\cdots,mn+m, \tag{11}$$

which provides the performance gradient

$$\nabla\eta(\theta) = \int_{x\in\mathbb{R}^n} \pi(\mathrm{d}x)\Big[\int_{y\in\mathbb{R}^n} \nabla P(\mathrm{d}y|x,\theta) g(y,\theta) + \nabla f(x,\theta)\Big], \tag{12}$$

where the symbol ∇ represents the gradient with respect to θ. In above equations, $g(x,\theta)$ is called the performance potential [8] (or bias, or relative value) of state x under control parameter θ. It is defined as

$$g(x,\theta) = \mathcal{A}f(x,\theta) = \lim_{K\to\infty} \mathrm{E}\{\sum_{k=0}^{K-1} (f(x_k,\theta) - \eta(\theta))|x_0 = x\}. \tag{13}$$

The derivative equation plays an important role in PA and performance optimization of Markov systems, especially when the actions at different states are correlated [9]. In our problem, θ is the coefficient of the feedback policy, so the control variables at different states are mutually decided by the same θ. This is not a standard MDP and policy iteration is not applicable. However, optimization with derivative equation is a proper way. There are a number of learning algorithms based on derivative equation in [10]. In the paper we only apply the results but not go deep into the details.

For LQG model, by the chain rule for differentiation, from (2) and (10) we have

$$\nabla P(\mathrm{d}y|x,\theta) = p_w(y - Ax - B\Pi_U(F\theta))\nabla(\Pi_U(F\theta)) B^{\mathrm{T}}\phi^{-1}(y - Ax - B\Pi_U(F\theta))\mathrm{d}y \tag{14}$$

and

$$\nabla f(x,\theta) = 2\nabla(\Pi_U(F\theta))D\Pi_U(F\theta). \tag{15}$$

$\nabla(\Pi_U(F\theta))$ may not exist because the projection function (4) is not differentiable on the boundary of control set U. However, the measure of the boundary is zero, it does not affect the integral in (12).

From (12), we can apply the technique of *changing measures*, which is the basic idea in *importance sampling* to estimate the derivative from sample path. A standard assumption for importance sampling is that if $\nabla P(\mathrm{d}y|x,\theta) > 0$, then $P(\mathrm{d}y|x,\theta) > 0$. The optimal control problem in the paper satisfies this assumption naturally because $p_w(x) > 0, \forall x \in \mathbb{R}^n$, thus $P(\mathrm{d}y|x,\theta) > 0$ always holds. Let

$$L(y|x,\theta) = \frac{\nabla P(\mathrm{d}y|x,\theta)}{P(\mathrm{d}y|x,\theta)} = \nabla(\Pi_U(F\theta))B^{\mathrm{T}}\phi^{-1}(y - Ax - B\Pi_U(F\theta)). \tag{16}$$

Then performance gradient (12) becomes

$$\nabla\eta(\theta) = \int_{x\in\mathbb{R}^n} \pi(\mathrm{d}x)\Big[\int_{y\in\mathbb{R}^n} L(y|x,\theta)P(\mathrm{d}y|x,\theta)g(y,\theta) + \nabla f(x,\theta)\Big]. \tag{17}$$

From [10], the performance potential $g(x,\theta)$ can be estimated from a sample path by truncation. Choose a fixed integer K and estimate $g(x,\theta)$ by

$$g(x,\theta) \approx \mathrm{E}\{\sum_{k=0}^{K-1} f(x_k,\theta)|x_0 = x\} - K\eta(\theta). \tag{18}$$

From the derivative equation (11), if the potential $g(x,\theta)$ is applied, then $g(x,\theta)+c$ for any constant c is also applied to (11), since we have $\int_{y\in\mathbb{R}^n} P(dy|x,\theta) = 1$ and therefore $\int_{y\in\mathbb{R}^n} \frac{\partial P(dy|x,\theta)}{\partial\theta_i} = 0$. Then the performance potential has infinite forms which only differ on a constant. We may ignore the constant $K\eta(\theta)$ in (18) and use its first term as an estimate:

$$g(x,\theta) \approx \mathrm{E}\{\sum_{k=0}^{K-1} f(x_k,\theta)|x_0 = x\}. \tag{19}$$

By (17) and (19), with the algorithm in [10] we have the following estimate of the performance gradient from a sample path, for given N and K:

$$\nabla\hat{\eta}(\theta) = \frac{1}{N-K+1}\sum_{k=0}^{N-K}\Big[L(x_{k+1}|x_k,\theta)\sum_{l=1}^{K} f(x_{k+l},\theta) + \nabla f(x_k,\theta)\Big], \tag{20}$$

where $\nabla f(x_k,\theta)$ and $L(x_{k+1}|x_k,\theta)$ can be calculated from (15) and (16). Cao [10] shows that for ergodic systems, we have

$$\nabla\eta(\theta) = \lim_{N\to\infty,K\to\infty} \nabla\hat{\eta}(\theta), \qquad w.p.1. \tag{21}$$

3.2 Gradient descent method

So far we propose a simple gradient descent method to obtain an ϵ-optimal parameter $\hat{\theta}$.

Algorithm 1. (1) Choose an integers K, simulation length N, a small positive number $\delta > 0$, and $0 < \kappa < 1$. Choose an initial control policy $\varphi(x,\alpha^{(0)},\beta^{(0)}) = \alpha^{(0)}x + \beta^{(0)}$ which stabilizes the system. $\theta^{(0)} = [\mathrm{Vec}(\alpha^{(0)})^{\mathrm{T}}\ (\beta^{(0)})^{\mathrm{T}}]^{\mathrm{T}}$. Choose a sequence of step sizes $\gamma^{(s)}, s = 0,1,\ldots,$ satisfying:

$$\sum_{s=0}^{\infty}\gamma^{(s)} = \infty, \quad \sum_{s=0}^{\infty}(\gamma^{(s)})^2 < \infty. \tag{22}$$

Set $s = 0$.

(2) Run the system for N periods under control policy $\varphi(x, \theta^{(s)}) = F\theta^{(s)}$. Estimate $\nabla\hat{\eta}(\theta)$ by (20).
(3) (a) Let $\theta^{(s+1)} = \theta^{(s)} - \gamma^{(s)}\nabla\hat{\eta}(\theta)$.
    (b) Obtain $\alpha^{(s+1)}$ by $\theta^{(s+1)}$. If $|\rho(A + B\alpha^{(s+1)})| < 1$, go to the next step; otherwise set $\gamma^{(s)} = \kappa\gamma^{(s)}$ and go to step 3a. $\rho(M)$ represents the spectral radius of a matrix M.
(4) If $||\theta^{(s+1)} - \theta^{(s)}|| < \delta$, set $\hat{\theta} = \theta^{(s+1)}$ and algorithm stops; otherwise set $s := s + 1$, and go to step 2. $||\cdot||$ represents the L_2-norm of a vector.

We consider convergence and optimality of the algorithm. We give these conditions:

Condition 1. Both θ^* and $\theta^{(0)}$ are inner points of the stable region $\Theta = \{\theta : \theta = [Vec(\alpha)^{\mathrm{T}},\ \beta^{\mathrm{T}}]^{\mathrm{T}}, |\rho(A + B\alpha)| < 1\}$.

Condition 2. Performance $\eta(\theta)$ is strong convex: there exists a number $\nu > 0$ such that

$$(\nabla\eta(\theta) - \nabla\eta(\theta'))^{\mathrm{T}}(\theta - \theta') \geq \nu||\theta - \theta'||^2 \tag{23}$$

holds for all $\theta, \theta' \in \Theta$.

This condition is a strengthened form of convexity (Proposition (B.5), p. 679, [6]). Lots of convex functions, i.e., quadratic functions, satisfy this condition.

Step 3b is to guarantee the stability of the new parameter $\theta^{(s+1)}$ by reducing step size. Since penalty on state variance $x^{\mathrm{T}}Cx$ is involved in cost function, thus if the system is not stable, performance must be very bad. The performance gradient must point to the stable region. Therefore, if the current parameter makes the system stable, there exists a proper step size to obtain a new parameter under which the system is still stable. The procedure in Step 3b: $\gamma^{(s)} = \kappa\gamma^{(s)}$ will stop in finite number of reductions. As Condition 1 holds, initial parameter $\theta^{(0)}$ makes the system stable, then each iteration round obtain a stable control policy.

Please note that, for any given integer K, (19) is a biased estimate. Thus $\nabla\hat{\eta}(\theta)$ is also a biased estimate. Define

$$\hat{z}(\theta) = \mathrm{E}\{\nabla\hat{\eta}(\theta)\} = \lim_{N\to\infty} \nabla\hat{\eta}(\theta), \tag{24}$$

and $z(\theta) = \nabla\eta(\theta)$. Consider convergence of above algorithm, we have the following results:

Lemma 1. If Conditions 1 holds, and K and N are large enough, Algorithm 1 converges to $\hat{\theta}$ satisfying $\hat{z}(\hat{\theta}) = 0$.

Proof. Denote step size applied in the sth iteration round as $\tilde{\gamma}^{(s)} = \kappa^{d(s)}\gamma^{(s)}$, where $d(s)$ is the number of reduction of step size in Step 3b. From above discussion, it is obvious that step sizes $\tilde{\gamma}^{(s)}$ also satisfy condition (22), and the estimate $\nabla\hat{\eta}(\theta)$ has finite variance. It is easy to verify that Hessian matrix $\nabla^2\eta(\theta)$ is bounded. For large enough N and K, $\nabla\hat{\eta}(\theta)$ must be a proper descent direction satisfying conditions in Proposition 3.5 from page 96 of [7]. By this convergence theorem, and a stable initial parameter as Condition 1, Lemma 1 is proved. □

Theorem 2. If Conditions 1 and 2 hold, then for any small number ϵ, and a stable initial parameter $\theta^{(0)}$, there exist a positive integer $\mathbf{K}$, such that for all $K \geq \mathbf{K}$, $|\eta(\theta^*) - \eta(\hat{\theta})| \leq \epsilon$.

Proof. From definition of $\hat{z}(\theta)$ and $z(\theta)$, we have

$$\lim_{K\to\infty} \hat{z}(\theta) = z(\theta).$$

Thus for any small number $\varepsilon > 0$, there exists an integer $\mathbf{K}$, such that for all $K \geq \mathbf{K}$, we have $||z(\theta) - \hat{z}(\theta)|| \leq \varepsilon$ for all θ. Choose $\varepsilon = \sqrt{\nu\epsilon}$. Consider Taylor series expansion of $\eta(\theta^*)$ around $\hat{\theta}$:

$$\eta(\theta^*) = \eta(\hat{\theta}) + z(\hat{\theta})^{\mathrm{T}}(\theta^* - \hat{\theta}) + o(\theta^* - \hat{\theta}). \tag{25}$$

From Lemma 1,

$$||z(\hat{\theta})|| = ||z(\hat{\theta}) - \hat{z}(\hat{\theta})|| \leq \varepsilon. \tag{26}$$

Then we consider upper bound of $||\theta^* - \hat{\theta}||$. From Condition 2 we have

$$\begin{aligned}\nu||\theta^* - \hat{\theta}||^2 &\leq (z(\theta^*) - z(\hat{\theta}))^{\mathrm{T}}(\theta^* - \hat{\theta}) \\ &= -z(\hat{\theta})^{\mathrm{T}}(\theta^* - \hat{\theta}) \leq ||\theta^* - \hat{\theta}||||z(\hat{\theta})||.\end{aligned}$$

Then

$$||\theta^* - \hat{\theta}|| \leq \frac{||z(\hat{\theta})||}{\nu} \leq \frac{\varepsilon}{\nu} \tag{27}$$

From (26), (27) and Taylor series expansion (25), we have

$$|\eta(\theta^*) - \eta(\hat{\theta})| \leq \frac{\varepsilon^2}{\nu} = \epsilon. \tag{28}$$

This completes the proof. □

This algorithm is a simple steepest descent gradient method. Particularly, it's a good idea to let N increase when s increases. Since at the beginning of iteration, the parameter θ is far away from the optimum, then the estimates need not be very accurate. When θ moves towards the minimum, N increases and $\nabla\hat{\eta}(\theta)$ becomes more accurate. This idea saves computation, and is applied to examples in Section 5.

Different from other approaches, our approach can be applied to on-line performance optimization of real systems. By observing system behaviors, we learn the performance gradient, then obtain a better policy. Repeating the procedure, the system goes on and its performance improves step by step. There is no need for exploratory experiments that may intervene the normal operation of the system. In order to obtain accurate estimates, the approach requires a long sample path. The on-line property makes the requirement computationally feasible, particularly for the optimization of high-speed sampled-data control systems, in which hundreds of sampled data is obtained in a second and, hence, the estimation may be implemented promptly. Batina et al. [2, 3] propose a simulation-based algorithm, which is looking for a general feedback map from state to control. Their algorithm can achieve the optimal control policy with arbitrary accuracy, but has no the on-line property because of its huge computation and requirement of exploratory experiments.

4. SPECIAL CASES

In this section, we introduce two special cases: scalar control set case and positive control set case in which the above results can be simplified.

4.1 Case 1: scalar control set

If control variable $u_k \in U \subset \mathbb{R}$, then U must be an interval denoted as $U = [b_l, b_u], -\infty \leq b_l < b_u \leq \infty$. Define a subset of state space: $\chi = \{x : F\theta \in U\}$. When $x \notin \chi$, projection function is active, then $u = \Pi_U(F\theta)$ is a constant (either b_l or b_u). So we have $\nabla(\Pi_U(F\theta)) = 0$ for $x \notin \chi$. And for $x \in \chi$, $\Pi_U(F\theta) = F\theta$ and $\nabla(\Pi_U(F\theta)) = F^{\mathrm{T}}$. Then we have $\nabla f(x,\theta) = 2F^{\mathrm{T}}DF\theta$ and $L(y|x,\theta) = F^{\mathrm{T}}B^{\mathrm{T}}\phi^{-1}(y - Ax - BF\theta)$. The gradient equation (17) becomes

$$\nabla\eta(\theta) = \int_{x\in\chi} \pi(\mathrm{d}x)[\int_{y\in\mathbb{R}^n} L(y|x,\theta)P(\mathrm{d}y|x,\theta)g(y,\theta) + \nabla f(x,\theta)], \tag{29}$$

and the corresponding estimation equation (20) becomes

$$\nabla\hat{\eta}(\theta) = \frac{1}{N-K+1}\sum_{k=0}^{N-K} 1_\chi(x_k) \left[L(x_{k+1}|x_k,\theta)\sum_{l=1}^{K} f(x_{k+l},\theta) + \nabla f(x_k,\theta)\right], \tag{30}$$

where $1_\chi(x)$ is an indicator function

$$1_\chi(x) = \begin{cases} 1, & \text{if } x \in \chi, \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

Then Algorithm 1 can be implemented by using above equations.

4.2 Case 2: positive control set

If every element of control vector $u = [u_1, \ldots, u_m]^{\mathrm{T}}$ is non-negative, then U is a positive control set. Each integer $0 \leq \varpi \leq 2^m - 1$ corresponds to a binary number $\omega = (\omega_1, \omega_2, \ldots, \omega_m), \omega_i = 0$ or 1 for all $i = 1, \ldots, m$. Define an index set $I_\varpi = \{i : \omega_i = 1\}$. The control space $\mathbb{R}^m$ can be partitioned into 2^m subsets: $\mathbb{R}^m = \bigcup_{\varpi=0}^{2^m-1} \Psi_\varpi$, where $\Psi_\varpi = \{u : u_i \geq 0$ for all $i \in I_\varpi$, and $u_i < 0$ otherwise$\}$. Then we have $U = \Psi_{2^m-1}$.

Define subsets of state space $\chi_\varpi = \{x : F\theta \in \Psi_\varpi\}$. Let M be any matrix with m rows, and $\Gamma_\varpi(M)$ be a matrix whose ith row is zero if $i \notin I_\varpi$, and other rows are the same as M. For $x \in \chi_\varpi$, we have

$$\Pi_U(F\theta) = \Gamma_\varpi(F\theta), \quad \nabla(\Pi_U(F\theta)) = \Gamma_\varpi(F)^{\mathrm{T}}. \tag{32}$$

Substitute (32) into (20), the estimate of the performance gradient can be obtained and Algorithm 1 achieves a feedback control policy for positive control set case. Optimal control of LQG with positive control is not well studied because the control set is not bounded, which may be hard to handle in traditional approaches. However in our learning approach, it is settled well without any difficulties.

5. NUMERICAL EXAMPLES

Example 1. Scalar control set example:

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad f(x,u) = Cx^2 + Du^2.$$

where $A = 0.983, B = 1, C = 1, D = 1$. The covariance matrix of w is 0.04, control set $U = [-0.01,\ 0.09]$. This example appears in [12].

Table 1. Comparison

Suboptimal polices	$\varphi(x)$	η
OLQRU (CEC)	$-0.604x$	0.689
Toivonen [22]	$-0.801x - 0.113$	0.636
Our approach (Algorithm 1)	$-0.586x - 0.217$	0.577

In the example, α and β are two scalars. Initial control policy can be chosen arbitrary provided it stabilize the system. For a fair comparison, Algorithm 1 starts from an arbitrary given parameter: $\theta^{(0)} = [-0.6\ \ 0.2]^{\mathrm{T}}$. Choose $K = 500$, $N = [\![100,000 \times (\frac{s+4}{5})]\!]$, where the symbol $[\![x]\!]$ represents the nearest integer of x. $\delta = 0.001$, $\gamma^{(s)} = \frac{1}{50s}$, $\kappa = 0.5$. Algorithm 1 terminates at iteration round 41, and its improvements on performance and control policy are shown in Figure 1 and 2. The comparison with other approaches is shown in Table 1. The first row is to apply OLQRU to the constrained problem. In scalar case, the policy obtained by CEC [17] is the same as OLQRU. The second row is a suboptimal policy from Toivonen [22], and the third line is the policy obtained by Algorithm 1. It shows that Algorithm 1 achieves the best performance. It is obvious that when the control set U is not symmetrical, the optimal policy has a drift from the origin. In our approach, the unsymmetrical U is involved in learning procedure, so we have a control policy with drift β. However, such drift is absent in OLQRU. In Toivonen's policy, a truncated Taylor series is used approximately so that the result is not good enough. The performance of our approach is 16% better than OLQRU and 9% better than Toivonen's policy.

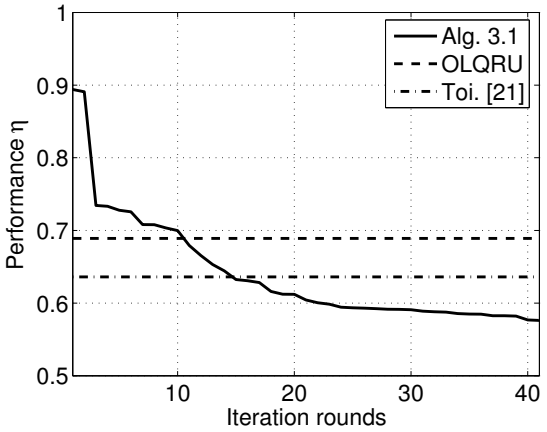

Fig. 1. Performance improvement

Example 2. Positive control set example:

$$x_{k+1} = Ax_k + Bu_k + w_k, \quad f(x,u) = Cx^2 + u^{\mathrm{T}}Du.$$

where $A = 0.983, B = [-0.5,\ 1], C = 1, D = \begin{bmatrix} 1 & 0.3 \\ 0.7 & 2 \end{bmatrix}$. The variance of w is 0.04.

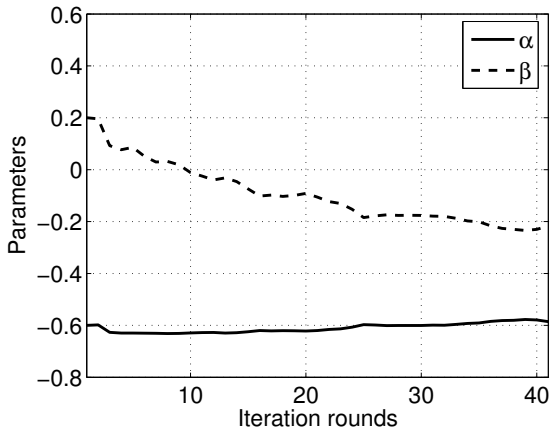

Fig. 2. Improvement of parameters

In this example, we take OLQRU as its initial policy, that is, $\varphi(x) = [0.40490 \quad -0.42040]^{\mathrm{T}}x$. Step size $\gamma = 1/(10s)$. Other parameter settings are the same as Example 1. Algorithm 1 terminates at iteration round 39, and the feedback control policy is $\varphi(x,\theta) = [0.4780 \quad -0.4329]^{\mathrm{T}}x + [0.0548 \quad -0.0035]^{\mathrm{T}}$. The performance is 0.091, while the performance under OLQRU is 0.104. Our approach achieves 14% improvement.

6. CONCLUSION

The paper presents a learning algorithm for optimization of LQG problem with constrained control. The iterative algorithm converges to an ϵ-optimal policy in affine linear form with arbitrary accuracy. The on-line property makes the algorithm suitable for implementation on practical systems. Examples illustrate that the control policy obtained with Algorithm 1 is better than other suboptimal polices.

The similar idea of the paper can be easily applied to handle optimal control problem of nonlinear systems. In this case, the assumption of affine linear feedback policy may not be feasible and an approximation of feedback control policy with more complex structure becomes necessary. This may be our future work.

REFERENCES

[1] M. Abu-Khalaf and F. L. Lewis, Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach, *Automatica*, 41: 779-791, 2005.

[2] I. Batina, A. A. Stoorvogel and S. Weiland, Stochastic disturbance rejection in model predictive control by randomized algorithms, *Proceedings of the American Control Conference*, Arlington, June 2001.

[3] I. Batina, Model predictive control for stochastic systems by randomized algorithms, *Ph.D. Thesis*, Eindhoven University of Technology, Netherlands, 2004.

[4] J. Baxter and P. L. Bartlett, Infinite-horizon policy-gradient estimation, *J. Art. Intell. Res.*, 15: 319-350, 2001.

[5] A. Bemporad, M. Morari, V. Dua and E. N. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, *Automatica*, 38: 3-20, 2002.

[6] D. P. Bertsekas, *Nonlinear Programming, Second Edition*, Belmont, MA: Athena, 1999.

[7] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Belmont, MA: Athena, 1996.

[8] X. R. Cao and H. F. Chen, Perturbation Realization, Potentials, and Sensitivity Analysis of Markov Processes, *IEEE Trans. Automat. Contr.*, 42: 1382-1393, 1997.

[9] X. R. Cao and H. T. Fang, Gradient-Based Policy Iteration: An example, *Proceedings of the 41st IEEE Conference on Decision and Control*, Las Vegas, Nevada USA, December 2002.

[10] X. R. Cao, A basic formula for online policy gradient algorithms, *IEEE Trans. Automat. Contr.*, 50: 696-699, 2005.

[11] D. J. Chmielewski and V. Manousiouthakis, On constrained infinite-time linear quadratic optimal control, *Systems and Control Letters*, 29: 121-129, 1996.

[12] D. J. Chmielewski and V. Manousiouthakis, On constrained infinite-time linear quadratic optimal control with sotchastic distrubances, *Journal of Process Control*, 15: 383-391, 2005.

[13] O. Hernandez-Lerma and J. B. Lasserre, Policy iteration for average cost Markov control processes on Borel spaces, *Acta Appl. Math.*, 47: 125-154, 1997.

[14] J. H. Lee and B. L. Cooley, Optimal feedback control strategies for state-space systems with stochastic parameters, *IEEE Trans. Automat. Contr.*, 43: 1469-1475, 1998.

[15] P. Marbach and J. N. Tsitsiklis, Approximate gradient methods in policy-state optimization of Markov reward processes, *Discrete Event Dynamic Systems: Theory and Applications*, 13: 111-148, 2003.

[16] S. P. Meyn and R. L. Tweedie, *Markov Chains and Stochastic Stability*, Springer-Verlag, London, 1993.

[17] T. Perez, H. Haimovich and G. C. Goodwin, On optimal control of constrained linear systems with imperfect state information and stochastic distrubances, *Int. J. of Robust and Nonlinear Control*, 14: 379-393, 2004.

[18] A. Saberi, A. A. Stoorvogel, and P. Sannuti, *Control of Linear Systems with Regulation and Input Constraints*, London: Springer-Verlag, 2000.

[19] P. O. M. Scokaert and J. B. Rawlings, Constrained linear quadratic regulation, *IEEE Trans. Automat. Contr.*, 43: 1163-1169, 1998.

[20] J. Si, A. Barto, W. Powell and D. Wunsch, *Handbook of Learning and Approximate Dynamic Programmingm*, New Jersey: Wiley, 2004.

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

[22] H. T. Toivonen, Suboptimal control of linear discrete stochastic systems with linear input constraints, *IEEE Trans. Automat. Contr.*, 28: 246-248, 1983.

[23] P. J. Werbos, Approximate dynamic programming for real-time control and neural modeling, In: D. A. White and D. A. Sofge (Eds.), *Handbook of Intelligent Control*, New York: Van Nostrand Reinhold, 1992.

[24] K. J. Zhang, Y. K. Xu, X. Chen and X. R. Cao, Policy iteration based feedback control, *Automatica*, to appear.