Guided Policy Search Methods: A Review

Jinyu Du*, Jian Fu, Cong Li
School of Automation, Wuhan University of Technology, Wuhan 430070, China
*djy_whut2014@whut.edu.cn

Abstract. Guided policy search methods (GPSs) have become important methods in the field of reinforcement learning in recent years. GPSs are a kind of policy search methods that utilize trajectory optimization methods to generate training data, guiding supervised learning. In theoretical research, GPSs combine convex optimization and deep learning, and have achieved fruitful results. In practical applications, they have achieved good results in complex control fields such as robots learning, especially manipulator operations. This paper mainly elaborates the development process and improvement route of GPSs. Firstly, the theoretical knowledge related to the GPSs is introduced. Secondly, the framework and basic methods of GPSs are analyzed; Thirdly, various improved GPSs based on the basic methods are generalized. Finally, the development and future improvement directions of GPSs are summarized, and the problems and future development trends are discussed.

1. Introduction
Reinforcement Learning (RL) is a branch of machine learning. It has developed rapidly in recent years and has achieved fruitful results in sequential decision-making problems such as robot learning. The RL methods allow the agent to use the reward in the interaction with the environment to learn the control policy. According to different policy learning methods, RL methods can be divided into three categories: RL methods based on Value Function, RL methods based on Direct Policy Search (DPS), and RL method based on Actor-Critic (AC) [1].

The methods based on value function occupy a large proportion in the field of RL, especially with the combination of deep learning and reinforcement learning, which utilizing neural network to approximate the value function. But in a task where the action and state space are large or continuous, it is hard to learn the value function with limited number of interactions. Greedy strategy is the mostly used method to maximize the value function, but a small action modification in action space will lead to a huge variance of the policy, and the convergence of policy cannot be absolutely guaranteed. DPS methods parameterize the policy and directly explore in parameter space. Compared with the approximation of the value function, the parameterization of the policy is more convenient. DPS methods usually adopt policy gradient method, which achieve good results under common cases. When the complexity of task increases or the policy parameter space becomes larger, the gradient requires a lot of computing resources, or even it is impossible to compute. The expansion of the parameter space increases the difficulty of exploration, and the number of samples to compute the gradient using the finite difference method also increases rapidly, which greatly reduces the efficiency of the algorithm. The performance of DPS is also limited by the choice of policy parameterization methods. Specific, low-dimensional parameterized policy limits the representation performance and reduces the generalization of the policy. For example, the Dynamic Movement Primitives (DMPs) policy [2] used in robot learning has a limited range of generalization.
In order to overcome the drawbacks of the DPS methods, Sergey Levine proposed a guided policy search (GPS) method framework [3]. The GPS method utilizes the Trajectory Optimization methods to guide the optimization of neural network policy parameters, avoiding falling into the local optimal dilemma. The trajectory optimization methods improve the sample efficiency with learned dynamics. Benefit from the great framework, GPS can utilize a more general neural network to parameterize the policy, which increases the ability to express and generalization, without suffering from data inefficiency. The GPS framework is similar to Imitation Learning (IL), but the difference is that IL utilizes expert data to guide the apprentice learning [4-5], while GPS uses trajectory optimization methods to guide the learning of neural network policy. The GPS framework can be regarded as using a supervised learning method in policy learning. The trajectory optimization methods provide an effective training set for supervised learning, or a training set similar to expert data.

Since the GPS method was proposed, many versions have been developed, and there are also many applications in the field of robot control, such as biped robot walking [3,7,9,18], manipulator operation [12,14,16-17] combined with vision, the control of autonomous aircraft, etc. The GPS method has its strengths, and it is believed that it can shine in more applications in the future.

2. Preliminary

Before introducing the series of GPS methods in detail, the relevant knowledge required by the GPS algorithm must be briefly introduced, such as the problem representation and symbol representation of the GPS algorithm, policy representation, trajectory optimization methods, convex optimization methods, etc.

2.1. Problem Definition

GPS framework combines the traditional optimal control with deep learning, decomposing the DPS methods into two steps. The first step is to find the parameters that maximize the expected cumulative return under the time-varying linear gaussian (TVLG) policy. The second step utilize the policy to guide the neural network learning, so that the neural network policy fits the TVLG policy. The problem of GPS method can be expressed by the following formula

\[
\min_{\mathbf{K}, \mathbf{c}, \mathbf{u}} E_{\mathbf{p}}[\ell(\mathbf{r})] \\
D_{KL}(q(\mathbf{x}) \mid \langle \pi_{\theta}(\mathbf{u} \mid \mathbf{x}) \rangle \| q(\mathbf{x}, \mathbf{u})) = 0
\]

(1)

In formula (1), \( \mathbf{x} \) and \( \mathbf{u} \) are state and action variable respectively. \( \pi_{\theta}(\mathbf{u} \mid \mathbf{x}) \) represents the parameterized policy. TVLG policy has the form \( \mathcal{N}(\mathbf{u} \mid \mathbf{K}, \mathbf{c}, \mathbf{u}) \).

2.2. Parameterization of Policy

Policy refers to a mapping from state space to action space. Assuming state variable \( \mathbf{x} \in \mathbb{R}^n \), action variable \( \mathbf{u} \in \mathbb{R}^m \), then policy \( \pi : \mathbb{R}^n \rightarrow \mathbb{R}^m \). The parameterized policy is to rewrite the mapping as a function of parameter \( \theta \). The performance of parameterized policy is determined by the way of parameterizing. In the field of robot learning, common policy parameterization methods include dynamic movement primitives, Gaussian Mixture Model (GMM) [20], Hidden Markov Model (HMM) [21], time-varying Linear Gaussian policy and neural network policy.

The DMPs method involves the knowledge of robot kinematics, and regards the skills of the robot as the combination of movement primitive. DMPs are driven by the differential equation of attractor dynamics, and their application fields are limited. The GMM and HMM methods think over the policy from the perspective of probability, and correspond each feature of the policy to the different states of the model [22]. The TVLG policy can fit any gaussian trajectory distribution, and can be used to parameterize complex high-dimensional nonlinear policy. TVLG policy is also closely related to the linear feedback controller in optimal control. Because of its time-varying linear characteristics, TVLG policy has its advantages in discontinuous, contact-rich tasks, and it has good generalization. TVLG
policy utilizes the form of conditional multivariate Gaussian $p(u_t | x_t) = \mathcal{N}(K_t, x_t + k_t, C_t)$ in the GPS frameworks, where $K_t$ is the linear feedback coefficient, $k_t$ is the linear feedforward term, and $C_t$ is the multivariate Gaussian covariance matrix, and the policy parameters can be expressed as $\theta = \{K_t, k_t, C_t\}$. For the GPS frameworks using DDP, iLQR, and iLQG [6,23,24] methods, the form of TVLG is expressed as $p(u_t | x_t) = \mathcal{N}(\bar{u}_t + K_t (x_t - \bar{x}_t) + k_t, C_t)$. With the development of neural network technology, neural networks are utilized to parameterize control policy, such as TRPO, DQN, DDPG algorithm [25-27]. The neural network policy has a high non-linearity, good fitting ability for complex policy and excellent versatility. For simple tasks, you can choose a multi-layer perceptron. For large and complex policy, such as end-to-end policy, convolutional neural networks (CNN) do well. When using neural network policy, due to the large number of parameters, the amounts of samples required for training increase rapidly. Ordinary policy search methods cannot achieve a good convergence.

2.3. Introduction of Convex Optimization

The research on convex optimization methods has a history of a century, and convex optimization has also been widely used. If the problem to be solved is convex, the convex optimization method has significant advantages, and it also performs well in some non-convex cases. A large number of convex optimization theories are involved in the GPS series research. Before fully understanding the GPS method, it is necessary to briefly understand several convex optimization methods such as dual gradient ascent method, Bregman cross direction multiplier method and mirror descent method [28].

2.3.1. Dual Gradient Descent

Dual Gradient Descent (DGD) is widely used to solve constrained optimal problems. When the optimization problem cannot be easily solved in the primal space or it is non-convex, it is an effective method to transform the optimization problem from primal space to dual space. Suppose the problem to be solved is expressed as follows

$$\min_{x} f(x)$$

s.t. $c(x) = 0$  \hspace{1cm} (2)

For equality constraints, a Lagrangian function $\mathcal{L}(x, \lambda) = f(x) + \lambda c(x)$ can be constructed, where $\lambda$ is the Lagrangian multiplier. It is easy to get the dual function $g(\lambda)$

$$g(\lambda) = \inf_{x} \mathcal{L}(x, \lambda) = \mathcal{L}(x^*, \lambda) \quad \text{where} \quad x^* = \arg\min_{x} \mathcal{L}(x, \lambda)$$

Where inf represents the lower bound. The minimum value of the primal problem is transformed into the maximum value of the dual problem. The gradient can be solved by using the gradient descent method after solving the gradient of the dual variable. The DGD algorithm flow is as follows

Algorithm 1: Dual Gradient Descent Algorithm

1. Compute $x^* = \arg\min_{x} \mathcal{L}(x, \lambda)$
2. Compute the gradient of dual function $\frac{d\mathcal{L}}{d\lambda} = \frac{d\mathcal{L}}{dx} \cdot \frac{dx^*}{d\lambda} + \frac{d\mathcal{L}}{dx} \cdot \frac{dx}{d\lambda}$
3. Update dual variable $\lambda = \lambda + \alpha \frac{d\mathcal{L}}{d\lambda}$
4. Does the algorithm converge? If yes, end, otherwise jump to 1

2.3.2. Bregman Alternating Direction Method of Multipliers

Bregman Alternating Direction Method of Multipliers (BADMM) [29] is an improvement by introducing Bregman divergence on the basis of ADMM method [30]. The ADMM method is an
extension of the Augmented Lagrangian Method (ALM). It is an effective method for solving decomposable problems in convex optimization, especially when the problem is large. The ADMM method can equivalently decompose the objective function of the primal problem into two or more convenient subproblems, solve each subproblem simultaneously, and finally coordinate the solutions of the subproblems to obtain the global solution of the primal problem. Assuming that the problem to be solved is decomposed into two subproblems, the optimization problem formula is expressed as follows

\[
\min_x f(x) + g(y) \\
\text{s.t. } Ax + By = z
\]  

(4)

There is an augmented Lagrangian function in the ADMM method

\[
\mathcal{L}_\rho(x, y, \lambda) = f(x) + g(y) + \lambda(Ax + By - z) + \frac{\rho}{2} \|Ax + By - z\|^2_2
\]  

(5)

Where \( \frac{1}{2} \|Ax + By - z\|^2_2 \) is the penalty term, and \( \rho \) is the penalty coefficient.

The BADMM method utilizes Bregman Divergence to replace the 2-norm penalty term, and has achieved good results. Assuming there are variables \( a \) and variables \( b \), and there is a continuous differentiable continuous convex function \( \phi \), the Bregman divergence can be expressed as

\[
B_{\phi}(a, b) = \phi(a) - \phi(b) - \langle \nabla \phi(b), a - b \rangle
\]  

(6)

The flow of the BADMM method is as follows Algorithm 2.

Algorithm 2: BADMM algorithm

1. \( x^{k+1} = \arg \min_x f(x) + \lambda^k (Ax + By^k - z) + \rho B_\phi(z - Ax, By^k) \)
2. \( y^{k+1} = \arg \min_y g(y) + \lambda^k (Ax^{k+1} + By - z) + \rho B_\phi(By, z - Ax^{k+1}) \)
3. \( \lambda^{k+1} = \lambda^k + \rho(Ax^{k+1} + By^{k+1} - z) \)
4. Does the algorithm converge? If yes, end, otherwise jump to 1

2.3.3. Mirror Descent Algorithm

In other convex optimization methods, the gradient descent methods in the primal space assumes that the optimization objective function \( f \) is Lipschitz continuous with respect to the Euclidean norm. If there is \( L \) – Lipschitz continuous with respect to the infinite norm, then it is \( \sqrt{n}L \) – Lipschitz continuous with respect to the second norm, but the coefficient \( \sqrt{n} \) cannot be large enough. The gradient method does not clearly distinguish the primal space and the dual space. The variable \( x \) is iterated in the primal space, and the gradient \( \nabla f(x) \) can be regarded as a linear functional of \( x \), which forms a dual space composed of linear functionals. Mirror Descent Algorithm (MDA) [31-32] not only pays attention to the problem of the primal space, but also pays attention to the dual space.

The idea of MDA is to make full use of the primal-dual space, assuming that there is a bijective function \( \Psi \) in the primal-dual space, also called the mirror function. MDA first maps the variables \( x \) in the primal space to \( \nabla \Psi(x) \) in the dual space; then uses the gradient method in the dual space; then uses the inverse mapping \( (\nabla \Psi)^{-1} \) of the variables in the dual space back to the primal space; and finally utilizes constraints to make the inverse-mapped variables lying in the domain. The idea of MDA method is shown in Figure 1.
3. Guided Policy Search Methods

3.1. Framework of Guided Policy Search

The idea of the DPS method is to search the policy parameters to minimize the expectation of cumulative instantaneous cost $E_{\pi}(\ell(\tau))$. The most direct approach to learn parameters is to utilize the gradient descent method. The classic DPS methods include REINFORCE, GPOMDP, and NAC [33]. GPS does not search directly in the policy parameter space. Instead, as shown in Figure 2, it introduces an intermediate agent, applies the trajectory optimization method to learn the TVLG policy, and then utilizes supervised learning structure to learn a general neural network parameterized policy. In Figure 2, the first step of GPS is summarized as the Control Phase, and the second as the Supervised Phase. The control phase interacts with the environment, while the supervised phase learns the policy from the control phase. From the perspective of neural networks, GPS is a supervised learning process. From the point of view of algorithm structure, GPS method and Behavioral Cloning (BC) in imitation learning have certain similarities.

The GPS method utilizes policy $p$ in the control phase to minimize expectation of cumulative instantaneous cost while ensuring that the neural network policy is fully learned from $p$. This requirement can be guaranteed by convex optimization methods. Commonly used convex optimization methods include DGD method, BADMM method and MDA method.

3.2. Basic Guided Policy Search Methods

3.2.1. GPS Based on Importance Sampling

GPS based on importance sampling (ISGPS) [3] is the pioneering work of the GPS series methods, which proposes the utilization of DDP to guide the policy search in parameter space. The ISGPS method first utilizes DDP to learn the TVLG policy from human demonstrations, and then iteratively utilizes the importance sampling (IS) to estimate the cumulative cost from the TVLG policy distribution $p$, instead of directly using the neural network policy $\pi_\theta$ to estimate. The formula of estimation of cumulative cost as follows

$$E_{s_0}[\ell(\tau)] = E_p \left[ \frac{\pi_\theta(\tau)}{p(\tau)} \ell(\tau) \right]$$

(7)

The utilization of IS makes it possible to use off-policy samples to estimate the cumulative cost. Therefore, ISGPS can use the TVLG policy generated by the DDP method [6] to guide the learning of
neural network policy. However, IS relies heavily on the matching of the two distributions and will produce arbitrary error results that cannot be detected. The stability of ISGPS is restricted [34]. Although there are shortcomings of instability, it provides a basic framework and improvement ideas for subsequent research.

3.2.2. GPS Based on Variational Inference

GPS based on variational inference (vGPS) [7] turns the problem of minimizing the cumulative cost into an inference problem by introducing a binary random variable \( \mathcal{O}_t \). \( \mathcal{O}_t \) represents the optimality indicator at time \( t \), that is \( \mathcal{O}_t = 1 \) when at time \( t \). We denote a distribution \( p(\mathcal{O}_t | \mathbf{x}_t, \mathbf{u}_t) \propto \exp(-\ell(\mathbf{x}_t, \mathbf{u}_t)) \). The objective function to be optimized is rewritten as

\[
p(\mathcal{O}|\theta) = \int p(\mathcal{O}|\tau)p(\tau|\theta)d\tau
\]

\[
\propto \int \exp\left(-\sum_{i=1}^{T}\ell(\mathbf{x}_i, \mathbf{u}_i)\right)p(\mathbf{x}_1)p(\mathbf{u}_1)\prod_{t=1}^{T}p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)d\tau
\]  

(8)

Take the logarithm of the objective function and use the distribution \( p(\mathcal{O}_t | \mathbf{x}_t, \mathbf{u}_t) \) to decompose it

\[
\log p(\mathcal{O}|\theta) = \mathcal{L}(q, \theta) + D_{KL}(q(\tau)\|p(\tau|\mathcal{O}, \theta))
\]

(9)

Where \( \mathcal{L}(q, \theta) \) is the variational lower bound, and its form is

\[
\mathcal{L}(q, \theta) = \int q(\tau)\log \frac{p(\mathcal{O}|\tau)p(\tau|\theta)}{q(\tau)}d\tau
\]

(10)

The second term is the KL divergence between the two distributions of \( q(\tau) \) and \( p(\tau | \mathcal{O}, \theta) \)

\[
D_{KL}(q(\tau)\|p(\tau | \mathcal{O}, \theta)) = -\int q(\tau)\log \frac{p(\tau | \mathcal{O}, \theta)}{q(\tau)}d\tau
\]

(11)

The decomposed optimization problem becomes the one that optimize \( \mathcal{L}(q | \theta) \) and \( D_{KL}(q \| p) \) respectively. The previous policy search methods are to randomly explore in the parameter space of policy \( \pi_{\theta} \), and obtain the trajectory \( q(\tau) \) by operating the target policy \( \pi_{\theta}(\mathbf{u}_t | \mathbf{x}_t) \). Although such a method can improve the policy, it relies on random exploration. vGPS proposes to directly optimize the trajectory distribution \( q(\tau) \) to minimize the expected cost. Model-based trajectory optimization methods are more powerful than random exploration, which greatly improves sample efficiency and reducing the number of iterations. The formula (11) utilizes iLQR, a variant of the DDP method, to optimize the trajectory \( q(\tau) \), and the variational lower bound is optimized by the Stochastic Gradient Descent (SGD) method.

3.2.3. GPS Based on KL Divergence Constraint

GPS based on KL divergence constraint (cGPS) [9] introduced a KL constraint between the TVLG policy and neural network policy. The purpose of the KL constraint is to make the two policy to be consistent. The problem of cGPS is described as follows

\[
\min_{\theta, q(\tau)} D_{KL}(q(\tau)\|p(\tau))
\]

s.t. \( q(\mathbf{x}_1) = p(\mathbf{x}_1) \)

\[
q(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t) = p(\mathbf{x}_{t+1}|\mathbf{x}_t, \mathbf{u}_t)
\]

(12)

\[
D_{KL}(q(\mathbf{x}_1)\|\pi_{\theta}(\mathbf{u}_1 | \mathbf{x}_1)) = 0
\]

The original form of the objective function is \( E_q[\ell(\tau)] - \mathcal{H}(q(\tau)) \), let \( p(\tau) \propto \exp(-\ell(\tau)) \) as the distribution of the cost, and then the objective function is transformed to the simple form of KL
divergence. Minimizing KL divergence means minimizing the expectation of cumulative cost and maximizing the information entropy of the trajectory distribution. Maximizing information entropy ensures the fully and widely exploration in the trajectory space, and it will not fall into a local optimal situation. Intuitively, the KL divergence constraint makes the trajectory has a higher probability where the cost is lower.

For the equation constraint problem in formula (12), we consider using the dual gradient descent method. The first and second terms in the constraint indicate that the initial state distribution and the dynamic model are consistent, which will be implicitly guaranteed. The DGD method rewrites the original problem into the form of Lagrangian function

$$
L(\theta, q, \lambda) = D_{KL}(q(\tau) \| \rho(\tau)) + 
\sum_{t=1}^{T} \lambda_t D_{KL}(q(x_t) \| \pi_\theta(u_t | x_t) \| q(x_t, u_t))
$$

(13)

For formula (13), the DGD method iteratively optimizes the trajectory $q(\tau)$, parameters $\theta$ and updates the Lagrangian multiplier $\lambda_t$. The optimization of $q(\tau)$ and $\theta$ corresponds to step 1 in algorithm 1, and the update of the Lagrangian multiplier corresponds to step 3. In the cGPS method, the variant of iLQG method is used to optimize the trajectory $q(\tau)$. For the optimization of the neural network policy, the conventional neural network optimization method such as LBFGS can be utilized.

The cGPS method utilizes KL divergence to constrain the TVLG policy $p$ and $\pi_\theta$, and solve the optimization problem with DGD method, which ensures that the policy $\pi_\theta$ learns from the TVLG policy totally. To some extent, cGPS overcomes the shortcomings of instability of policy convergence in ISGPS and vGPS.

3.3. Summary of Basic GPS Methods

The last subsection introduced three basic GPS methods, which basically build up the GPS framework. The ISGPS method puts forward the prototype of GPS, and utilizes the importance sampling method to estimate the expectation of cumulative cost with the TVLG policy distribution. vGPS attempts to utilize variational inference. Finally, the cGPS introduces KL divergence to constrain the trajectory distribution and cost distribution, and utilizes the DGD method to obtain the optimal policy. Although the three methods have different ideas, they basically follow the basic framework of GPS. At the same time, they all have a common assumption: the model of the system is known. In more applications, the model is unknown. It is very difficult to obtain the model through mechanism modelling or data fitting. How to utilize the GPS method under the unknown model becomes an improvement direction. As a general constrained optimization method, does the DGD method fit the GPS framework perfectly? Using an optimization method that is more suitable to the problem itself may improve method performance. When the model is unknown, you can use the data to fit a global or local model, but it is also a good idea to directly utilize model-free methods. The next section will introduce several improved GPS methods.

4. Improvements of Guided Policy Search

4.1. GPS Under Unknown Model

In practical applications, kinetic models are mostly unknown and difficult to obtain. Although the global model can be fitted by means of system identification, it is very complicated and time-consuming. The GPS methods mentioned in Section 3 assume that the model is known, and the trajectory optimization method needs to perform local linearization on the known model. When the model is unknown, only a local linear model needs to be fitted, which is much easier than fitting a global model.

Assuming that the fitted local linear model is $p(x_{t+1} | x_t, u_t) = \mathcal{N}(f_w x_t + f_u u_t, F_w)$, it can be fitted using the Linear Regression (LR) method [35]. We first execute the TVLG policy obtained in the last iteration to on the real system to obtain $M$ sample trajectories $D = \{\tau_{1:M}\}$, and then utilize the linear
regression method to fit the coefficients $f_x$ and $f_u$, to finally fit $F_i$ based on the error. The LR formula is as follows

$$f_x, f_u = \arg \min \sum_{i=1}^{M} ||x_{t+1}^i - (f_x x_t^i + f_u u_t^i)||^2$$  \hspace{1cm} (14)$$

For high-dimensional dynamics, a large enough sample data is needed to fit a good linear model. Certain prior information can be utilized to help the model-fitting and reduce the need for samples. The Gaussian Mixture Model (GMM) method is utilized to provide prior information [36]. We first fit the GMM model on the data $(x_t, u_t, x_{t+1})^T$, and then utilize the fitted GMM model to provide the normal-inverse-Wishart prior for the mean and covariance of the local linear Gaussian model. Assume that the normal-inverse-Wishart prior has parameters $\Phi, \mu_0, m$ and $n_0, \hat{\mu}$ and $\hat{\Sigma}$ are the empirical mean and covariance on the data set. Then the posterior estimate of mean and $\mu$ covariance $\Sigma$ is

$$\mu = \frac{m\mu_0 + n_0\hat{\mu}}{m + n_0}$$  \hspace{1cm} (15)$$

$$\Sigma = \frac{\Phi + M\hat{\Sigma} + \frac{Mm}{M + m}(\hat{\mu} - \mu_0)(\hat{\mu} - \mu_0)^T}{M + n_0}$$

The GPS method under the unknown model describes the optimization problem similarly to the cGPS method. After omitting the information entropy term, it can be written as

$$\min_{p(\tau)} E_p[\ell(\tau)]$$

s.t. $D_{KL}(p(x)_x p(u|x)) = 0 \forall t$ \hspace{1cm} (16)$$

Compared with the GPS methods in Section 3, since the local model is only valid in a small neighborhood of the fitted data, constraints must be made on the trajectory changes in the trajectory optimization method when using the local linear Gaussian model. The control phase trajectory optimization problem of the GPS method of unknown model can be described as

$$\min_{p(\tau) \in N(\tau)} E_p[\ell(\tau)]$$

s.t. $D_{KL}(p(\tau)||\hat{p}(\tau)) \leq \epsilon$ \hspace{1cm} (17)$$

Where $\hat{p}(\tau)$ represents the trajectory distribution of the last iteration. For the inequality constraint problem, the DGD method can still be utilized to solve the problem, and the Lagrangian function can be written as

$$L_{u_j}(p(\tau), \eta) = E_p[\ell(\tau)] + \eta[D_{KL}(p(\tau)||\hat{p}(\tau)) - \epsilon]$$  \hspace{1cm} (18)$$

Where $\eta$ is the Lagrangian multiplier, formula (18) can be simplified to obtain

$$L_{u_j}(p(\tau), \eta) = \sum_{t} E_{p(x, u)} \left[ \ell(x_t, u_t) - \eta \log \hat{p}(u_t| x_t) \right]$$

$$-\eta \ell(p(\tau)) - \eta \epsilon$$  \hspace{1cm} (19)$$

In formula (19), let $\tilde{\ell}(x_t, u_t) = (1/\eta)\ell(x_t, u_t) - \log \hat{p}(u_t| x_t)$ as the augmented cost function, the DGD method can be utilized as in cGPS to alternately optimize the trajectory distribution $p(\tau)$ and dual variables $\eta$. The optimization of parameter $\theta$ is regarded as supervised learning with the TVLG as a part of loss function, and a good training set is obtained from trajectory optimization. Compared with the cGPS method, the GPS method under unknown model uses two nested DGD methods. The inner loop utilizes the DGD method for the problem shown in equation (19), and the outer loop uses the DGD method for the problem shown in equation (16). The outer DGD method first optimizes the supervised phase, and then optimizes the control phase using the inner DGD method. The update method of the dual variable $\lambda_\eta$ is simply updated by the formula $\lambda_\eta = \lambda_\eta + \alpha D_{KL}(p_i(x_t, u_t)||p_i(x_t, u_t))$. The update method of the dual variable $\lambda_i$ is simply updated by the formula $\lambda_i = \lambda_i + \alpha D_{KL}(p_i(x_t, u_t)||p_i(x_t, u_t))$. 


The proposed GPS method under unknown model makes the GPS method no longer require an accurate mode, and its scope of application has been greatly expanded. It can be applied to a large number of scenarios that are difficult to obtain a global model, and the generalization of the method is enhanced.

4.2. GPS Based on BADMM

The GPS based on BADMM (BADMM-GPS) was proposed by Chelsea Finn [12] to train end-to-end deep neural network policy. Under the GPS framework, it is necessary to alternately optimize the trajectory distribution $p(\tau)$ and policy parameter $\theta$. This can be interpreted that the policy search problem is decomposed into two subproblems under the GPS framework, so the DGD method can be extended to the BADMM method. The optimization problem of the BADMM-GPS method is described as

$$\min_{p, \pi} E_{p}[\ell(\tau)] \quad \text{s.t.} \quad p(u|\tau) = \pi_{\theta}(u|\tau)$$

(20)

The description of the BADMM-GPS method directly uses constraints to make $p$ and $\pi_{\theta}$ equal. In practical applications, it is customary to change the constraint of trajectory distribution produced by the corresponding policy. For time $t$, the distribution of state-action pairs $(x_t, u_t)$ generated by the two policy under state $x_t$ is equal. The constraint can be rewritten as $p(u_t | x_t) p(x_t) = \pi_{\theta}(u_t | x_t) p(x_t)$. Use the BADMM method to write the Lagrangian function

$$L_p(\theta, p) = \sum_{i=1}^{T} E_{p(x_t, u_t)}[\ell(x_t, u_t)] + E_{p(x_t, \pi_{\theta}(u_t | x_t)}[\lambda_{x_t, u_t}] - E_{p(x_t, u_t)}[\lambda_{x_t, u_t}] + \nu \phi(\theta, p)$$

(21)

Where $\lambda_{x_t, u_t}$ is the Lagrangian multiplier and $\nu$ is the coefficient of the penalty term. In this method, Finn utilizes KL divergence to replace Bregman divergence as a penalty term for the convenience of calculation.

The proposed BADMM-GPS method explains the bivariate optimization problem more reasonably than the DGD method, and at the same time enables GPS to be used for large-scale optimization problems. The BADMM-GPS method fits a linear Gaussian model of the dynamic model $p(x_{t+1} | x_t, u_t) = N(f_w x_t + f_u u_t + f_0, F)$. In the local area, the model can be regarded as linear. After quadraticizing the instantaneous cost $\ell(x_t, u_t)$, it meets the requirements of the standard LQR method. Therefore, the control phase can use the standard LQR method to replace iLQG and other methods, simplifying the control phase.

4.3. GPS Based on Approximate Mirror Descent

The GPS based on approximate mirror descent method (MDGPS) was proposed by William Montgomery [14]. The problem of the GPS method is interpreted as an approximate mirror descent method. On the basis of ensuring the convergence of the algorithm, the algorithm structure is simplified and difficulty of implementation decreases. The problem of the MDGPS method is expressed as

$$p^t \leftarrow \arg \min_{p} E_{p(\tau)} \left[ \sum_{i=1}^{T} \ell(x_t, u_t) \right] \quad \text{s.t.} \quad D(p, \pi^t) \leq \epsilon$$

(22)

$$\pi^{t+1} \leftarrow \arg \min_{\pi} D(p^t, \pi)$$

The MDGPS method iteratively optimizes $p$ and parameter $\theta$. In the control phase, the KL divergence constraint $D_{\text{KL}}(p(\tau) \| \tilde{p}(\tau)) \leq \epsilon$ of the trajectory distribution can be changed to
\( D_{\text{KL}}(p(\tau) \parallel \pi_\theta(\tau)) \leq \epsilon \), which directly utilizes the trajectory distribution of the global policy. For nonlinear models, use the local linearization of the global policy \( \pi_\theta \) of the last iteration instead. Obviously, the control phase optimization utilizes the DGD method to obtain the optimal trajectory distribution. The BADMM-GPS and previous GPS methods have not considered neural network policy \( \pi_\theta \) in the control phase at all, which will lead to the problem that the GPS method may not achieve complete convergence. Therefore, the MDGPS method adds consideration to \( \pi_\theta \) in the control phase.

In the second term in formula (22), Suppose \( \Pi_\varnothing \) is the set of possible policies of all \( \pi_\theta \), that is, the set of optimized policy \( p \) in the control phase. The MDA method maps the dual space variables to the primal space, and then constrains the mapped variables to its domain. Correspondingly, supervised phase of MDGPS optimizes the moment projection of KL divergence, constraining \( \pi_\theta \) to \( \Pi_\varnothing \). From the perspective of information theory, the KL divergence is the information increment after the policy \( p \) is coded with \( \pi_\theta \), which minimizes the KL divergence between the policy distributions, that is, reduces the degree of coding imperfection as much as possible. When the KL divergence is 0, it means that the policy \( \pi_\theta \) encodes the policy \( p \) perfectly, and the neural network has completely learned the guiding policy \( p \).

The biggest advantage of the MDGPS method is that it simplifies the GPS framework, the control phase and the supervised phase are separated in form, which simplifies the description and analysis of problem and the complexity of code implementation. Although there is a certain loss in the convergence rate, the impact is not great, and its convergence is guaranteed by the convergence of the MDA method. The decoupling of the supervised phase and the control phase facilitates the subsequent improvement of the MDGPS.

4.4. GPS Based on Path Integral

Path Integral Guided Policy Search (PIGPS) method [16], which utilizes model-free trajectory-centric reinforcement learning method PI2 in the control phase, omits the process of model fitting and avoids setting the fitted parameters. The calculation time required for each iteration is greatly shortened. At the same time, PI2 utilizes the path integral, without calculating the gradient, and can adapt to many discontinuous and contact-rich scenarios where the gradient cannot be calculated. This also further expands the utilization of the GPS method.

PIGPS utilizes an improvement of PI2 which is called Path Integral Policy Improvement with Covariance Matrix Adaptation (PI2-CMA) [37-39], and combines the relative entropy strategy search (REPS). The trajectory optimization problem combining the REPS and the PI2-CMA can be expressed as

\[
\begin{align*}
\min_{\pi} E_p[I(\tau)] \quad \text{s.t.} \quad D_{\text{KL}}\left( p(u_t | x_t) \parallel \bar{p}(u_t | x_t) \right) \leq \epsilon
\end{align*}
\]

(23)

The characteristic of the REPS is to use the KL between the old and new policies as a constraint, which solves the problem that the excessively large difference between the old and the new strategies cannot steadily reduce the expected cost. This is similar to the constraint adopted in Section 4.1 because the local model is only valid in a small neighborhood. The PIGPS method is an improved version based on MDGPS. In order to use the MDGPS, the policy \( \bar{p}(u_t | x_t) \) in formula (23) is replaced by \( \pi_\theta(u_t | x_t) \). The DGD is used to solve formula (23), and its Lagrangian function can be written as

\[
L_p(p, \eta) = E_p[I(\tau)] + \eta \left[ D_{\text{KL}}\left( p(u_t | x_t) \parallel \bar{p}(u_t | x_t) \right) - \epsilon \right]
\]

(24)

The dual function \( g(\eta) \) of the above formula is
\[ g(\eta_i) = \eta_i \epsilon + \eta_i \log \frac{1}{N} \sum_{i=1}^{N} \left[ e^{-\frac{1}{\eta_i} s_{it}} \right] \] (25)

Through the optimization of the dual function, the dual variable \( \eta_i \) at each moment is obtained.

The control phase optimizes the TVLG policy, and its policy parameter is \( \{K_i, k_i, C_i\} \). The PIGPS method optimizes the feedforward term \( k_i \) and the covariance matrix \( C_i \), and fixes the feedback coefficient matrix \( K_i \) to a constant. With this setting, local policy can be learned with a small number of samples. According to the PI\(^2\)-CMA method, the optimization formula of parameters and is

\[ k_i = \sum_{j=1}^{N} p_{ij} \bar{k}_{ij} \] (26)

\[ C_i = \sum_{j=1}^{N} p_{ij} (\bar{k}_{ij} - k_i)(\bar{k}_{ij} - k_i)^\top \]

Where \( \bar{k}_{ij} \) represents the feedforward term of sampling, which can be calculated by the formula \( \bar{k}_{ij} = u_{ij} - K_j x_{ij} \), and \( p_{ij} \) is the weighted probability. Since the control phase requires a local policy, the step of calculating global policy parameters in PI\(^2\)-CMA is omitted.

PIGPS utilizes the PI\(^2\) method. While obtaining the advantages of model-free and suitable for discontinuous contact-rich scenarios, it also brings the inherent problems of model-free methods, and the sample efficiency is reduced to a certain extent.

4.5. GPS Based on Combination of Path Integral and LQR

Model-based GPS methods benefit from model information and are highly efficient in sample utilization. There are three types of models: the global model is known, the global model is fitted, and the local model is fitted. Fitting a model requires a lot of computing resources, which prompted the development of model-free methods. The model-free method sacrifices the sample efficiency in exchange for the simplicity of algorithm implementation and calculation. Yevgen Chebotar [17] shares the advantages of model-based and model-free methods, combining the model-based linear quadratic regulator with the PI\(^2\), and proposed a trajectory-centric reinforcement learning method called the PILQR. Applying the PILQR as a control phase optimization method to MDGPS, the PILQR-GPS method is obtained.

The PILQR method is based on the PI\(^2\) method, and its algorithm process can be divided into two parts: The first part uses fitted linear models (FLM) to estimate cost-to-go, that is, to estimate the quadraticized value function; The second part performs the PI\(^2\) update based on the residual, the residual is the difference between the real cost-to-go estimated from samples and the approximate cost-to-go estimated from the FLM. The problem to be solved by the PILQR method can be expressed as

\[
\min_{p^{(i)}} E_{p^{(i)}} \left[ S(x, u) \right] \quad \text{s.t.} \quad E_{p^{(i+1)}} \left[ D_{KL} \left( p^{(i)} \| p^{(i+1)} \right) \right] \leq \epsilon
\] (27)

Where \( S(x, u) = \sum_{j=0}^{T} t(x_j, u_j) \), which represents the cost-to-go function at time \( t \). Solving formula (27) can get the iterative relationship between the new policy \( p^{(i)} \) and the old policy \( p^{(i-1)} \)

\[
p^{(i)}(u_i \mid x_i) \propto p^{(i-1)}(u_i \mid x_i) E_{p^{(i-1)}} \left[ \exp \left( -\frac{1}{\eta_i} S(x, u) \right) \right]
\] (28)

Write the real cost-to-go in the form of the sum of estimated cost-to-go and residuals, \( S_i = \hat{S}_i + \tilde{S}_i \), and formula (28) can be deformed to obtain a two-step update of the PI\(^2\) method based on residuals
\begin{align}
\hat{p}(u_t | x_t) & \approx p^{\delta(t)}(u_t | x_t) E_{p^{\delta(t)}} \left[ \exp \left( -\frac{1}{\eta_t} \tilde{S}_t \right) \right] \\
p^{\delta(t)}(u_t | x_t) & \approx \hat{p}(u_t | x_t) E_{p^{\delta(t)}} \left[ \exp \left( -\frac{1}{\eta_t} \tilde{S}_t \right) \right] 
\end{align}

(29)

It can be seen from formula (29) that the PILQR method introduces an intermediate policy \( \hat{p} \), which serves as a bridge to communicate the model-based and model-free methods. The first formula uses the LQR method combined with FLM to estimate the cost-to-go function. The second formula uses the residual-based \( \Pi^2 \) method to improve the policy. The first formula is introduced to use model information to improve the efficiency of the sample and eliminate the problem of reduced data efficiency caused by a single \( \Pi^2 \).

5. Other GPS Methods and Summary

5.1. Other GPS Methods and Improvements

In addition to the above variants, the algorithm based on GPS framework has other improvements and applications. A brief introduction is given below.

Tianhao Zhang [13] applies model predictive control (MPC) method and GPS method to autonomous aircraft. MPC is utilized in the training phase to generate data based on the full state observations in the known training environment. The generated data guides the training of neural network policy for partial state inputs. The utilization of MPC enables the MPC-GPS method to have the ability of online training and planning.

In the control phase of the GPS method, in order to fully explore the optimal trajectory distribution under different initial states, it is necessary to perform trajectory optimization for different initial states. The training of neural network in the supervised phase needs the data in the control phase, and the learning must wait the end of the execution of control phase. Based on this problem, Ali Yahya [15] proposed the Asynchronous Distributed GPS (ADGPS) method. Multiple robots execute the control phase at the same time, and they can learn more generalized policy used in reality than one robot. In ADGPS, the control phase and the supervised phase are calculated on different computing platforms, and the two are communicated through "replay memory" in the middle. The asynchronous calculation of the supervised phase and the control phase can give full play to their respective computing capabilities and accelerate the algorithm process.

Many previous GPS methods are used to solve MDP problems and cannot be applied to non-MDP problems. Connor Schenck [41] successfully applied the GPS method to non-MDP problems, such as problems with time lag. Connor Schenck designs an experimental scene that he makes a robotic arm to pour a given amount of water into the cup, and measures the amount of poured water with a precision balance scale. Through the observation of the balance scale, we can obtain the data of water in the cup, but not the amount of poured water. There is a time lag between pouring and measuring, which makes it a non-MDP problem. In order to solve the problem with a time lag, Connor Schenck considers the state of multiple moments before the moment when fitting the model and policy. The model is expressed as \( f \left( x_{t+1} \middle| x_{t-n}, u_{t-n}, \ldots, x_t, u_t \right) \), the policy is expressed as \( \pi \left( u_t \middle| x_{t-n}, \ldots, x_t, \theta \right) \), and \( n \) is \( n \) time steps before the current moment. As long as \( n \) can cover all the lagging processes, the model and policy can predict the subsequent states and actions well.

When the GPS method trains the global policy, it needs to give all the training samples, which requires that all the initial states are known at once, and the local policy corresponding to each initial state is learned in the control phase. For the situation that the initial state cannot be given at one time or can be given sequentially, Biao Sun [42] proposed a linear online version of the GPS method. Assuming that multiple initial states are given sequentially, the control phase only learns a local policy \( p_{\text{pre}} \) of the initial state in one iteration, and combines the previous neural network policy \( p_{\text{pre}} \) and the current local
policy to learn a new global neural network policy $p_{global}$. Biao Sun proposed an acting policy $p_{act}$ for sample collection. The acting policy is a linear coupling between $p_{cur}$ and $p_{pre}$, namely $p_{act} = \alpha p_{pre} + \beta p_{cur}$. Fangzhou Xiong [43] proposed Sequential Multitask Learning in GPS (SMT-GPS) for the same problem. SMT-GPS regards different initial conditions of the same problem as a multi-task problem, and giving tasks sequentially as a process of life-long learning. SMT-GPS does not utilize simple linear policy coupling, but utilizes the Elastic Weight Consolidation (EWC) method when learning the global policy to extract key information from the global policy of the previous task and combine it with the local policy of the current task to learn new global policy. Compared with the former, this method has more profound theoretical support and eliminates blindness.

5.2. Summary of Multiple GPS Methods
This paper introduces a variety of GPS methods and their improvements. Sergey Levine and other researchers have gradually improved the GPS method step-by-step and expanded the scope of utilization of the algorithm. Table 1 summarizes the GPS methods and main improvements described in this paper. Table 2 summarizes several main improvements and main representatives to provide ideas for future improvements. Wen Sun [44] proposed a Dual Policy Iteration (DPI) framework, which is similar to GPS algorithm ideas. The framework puts forward the idea of dual-policy learning, which are reactive policy and non-reactive policy. Non-reactive policy is learned in the training phase and are used to guide the policy improvement of reactive policy, while reactive policy is used to generate sample data in the testing phase. The TVLG policy in GPS is used for the non-reactive policy under the DPI framework, while the neural network policy corresponds to the reactive policy. Under the guidance of this framework, the GPS method will have a more general meaning.

Table 1  Main improvements and analysis of advantages and disadvantages of various GPS methods

| Method       | Main Improvements                                                                 |
|--------------|-----------------------------------------------------------------------------------|
| ISGPS        | 1. Use DDP method to generate guiding samples;                                     |
|              | 2. Use IS to enable policy search to use guiding samples.                           |
|              | 3. Use the variational inference method to reinterpret the cumulative              |
|              | optimization problem;                                                             |
| vGPS         | 2. Use iLQR instead of DDP                                                         |
|              | 1. Use KL divergence to constrain $p$ and $\pi_{\theta}$;                          |
| cGPS         | 2. Use iLQG variants to optimize the control phase                                  |
| GPS under    | 1. Use LR and GMM methods to fit local linear models;                               |
| unknown model| 2. Add new and old trajectory distribution constraints                               |
| BADMM-GPS    | 1. Use BADMM instead of DGD method                                                  |
| MDGPS        | 1. Interpret the GPS problem as an approximate mirror descent method;              |
|              | 2. The constraints between new and old trajectory distribution are improved to     |
|              | between the new local and old global policy trajectory distribution                |
| PIGPS        | 1. Use model-free PI² method as the control phase optimization method;              |
|              | 2. Data sampling uses global policy                                                |
| PILQR-GPS    | 1. Improve the control phase to the PI² method based on cost-to-go residuals;      |
|              | 2. Use LQR-FLM method to estimate cost-to-go and optimize intermediate policy        |
Table 2. Main improvement directions of GPS methods

| Improvement Directions                  | Main Representatives                                      |
|----------------------------------------|----------------------------------------------------------|
| Improvements of the GPS framework      | 1. ISGPS, DGD, BADMM, MDGPS                               |
| Improvements of control phase          | 1. DDP, iLQR, iLQG, LQR, PI², PILQR                      |
| Improvements of supervised phase       | 1. MLP, CNN                                              |
| Improvements of model fitting          | 1. Known model;                                          |
|                                        | 2. fitted local linear model LR+GMM;                      |
|                                        | 3. model-free                                            |
|                                        | 1. Constraint between new and old trajectory distribution;|
| Other minor improvements               | 2. Adaptive upper bound of constraints $\epsilon$         |
|                                        | 3. On-policy or off-policy;                               |
|                                        | 4. Different TVLG policy.                                |
|                                        | 5. MDP or non-MDP                                        |

6. Conclusion

The GPS method was first proposed by Sergey Levine, and after several years of development, several versions of improved GPS methods were born. Although the development time is still short, it also shows the power of the algorithm in the field of robot learning. The data in the behavioral cloning method in imitation learning comes from expert demonstrations, and the DPS methods conduct random exploration in the policy parameter space. For tasks with high dimensions and complex policy, these two kinds of methods are not well applicable. The GPS method can solve the above-mentioned dilemma by introducing a trajectory optimization method to generate a guide trajectory and accelerate the process of policy search.

The GPS method also has certain shortcomings. Its scope of application is restricted by the control phase method. For different tasks, different methods need to be adopted, which lacks a certain versatility. For complex application scenarios such as robot walking, the control phase method cannot obtain a good TVLG policy under random initial conditions, which requires learning from demonstrations using expert data. An important issue in the RL method is how to set the reward function. The quality of the reward function will directly determine the quality of the RL method. Similarly, the setting of the cost function in the GPS method is also very important. The theoretical knowledge involved in GPS method is too complicated, mainly including (random) optimal control, neural network, convex optimization, policy search, model fitting, stochastic process, information theory, probability theory and robot-related knowledge. The theoretical foundation for researchers is very high, and the implementation is difficult.

Compared with the TRPO, DDPG, DQN and other methods developed in the same period, the GPS method is not so popular, but it still has great potential and broad prospects in the application fields of robots. It is awaiting in-depth research by the majority of researchers to develop the potential capabilities of GPS algorithms.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants 61773299, 51575412.

References

[1] Guo X. Reinforcement learning in simple language: an introduction to principles [M]. Publishing House of Electronics Industry, 2018.

[2] Ijspeert A J, Nakanishi J, Hoffmann H, et al. Dynamical movement primitives: learning attractor models for motor behaviors[J]. Neural computation, 2013, 25(2): 328-373.

[3] Levine S, Koltun V. Guided policy search[C]//International Conference on Machine Learning.
[4] Asfour T, Azad P, Gyarfas F, et al. Imitation learning of dual-arm manipulation tasks in humanoid robots[J]. International Journal of Humanoid Robotics, 2008, 5(02): 183-202.

[5] Abbeel P, Ng A Y. Apprenticeship learning via inverse reinforcement learning[C]//Proceedings of the twenty-first international conference on Machine learning. 2004: 1.

[6] Tassa Y, Mansard N, Todorov E. Control-limited differential dynamic programming[C]//2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014: 1168-1175.

[7] Levine S, Koltun V. Variational policy search via trajectory optimization[C]//Advances in neural information processing systems. 2013: 207-215.

[8] Moon T K. The expectation-maximization algorithm[J]. IEEE Signal processing magazine, 1996, 13(6): 47-60.

[9] Levine S, Koltun V. Learning complex neural network policies with trajectory optimization[C]//International Conference on Machine Learning. 2014: 829-837.

[10] Levine S, Abbeel P. Learning neural network policies with guided policy search under unknown dynamics[C]//Advances in Neural Information Processing Systems. 2014: 1071-1079.

[11] Sergey L, Wagener N, Abbeel P. Learning contact-rich manipulation skills with guided policy search[C]//Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA. 2015: 26-30.

[12] Levine S, Finn C, Darrell T, et al. End-to-end training of deep visuomotor policies[J]. The Journal of Machine Learning Research, 2016, 17(1): 1334-1373.

[13] Zhang T, Kahn G, Levine S, et al. Learning deep control policies for autonomous aerial vehicles with mpc-guided policy search[C]//2016 IEEE international conference on robotics and automation (ICRA). IEEE, 2016: 528-535.

[14] Montgomery W, Levine S. Guided policy search as approximate mirror descent[J]. arXiv preprint arXiv:1607.04614, 2016.

[15] Yahya A, Li A, Kalakrishnan M, et al. Collective robot reinforcement learning with distributed asynchronous guided policy search[C]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017: 79-86.

[16] Chebotar Y, Kalakrishnan M, Yahya A, et al. Path integral guided policy search[C]//2017 IEEE international conference on robotics and automation (ICRA). IEEE, 2017: 3381-3388.

[17] Chebotar Y, Hausman K, Zhang M, et al. Combining model-based and model-free updates for trajectory-centric reinforcement learning[J]. arXiv preprint arXiv:1703.03078, 2017.

[18] Levine S. Motor skill learning with local trajectory methods[D]. Stanford University, 2014.

[19] Bubnicki Z. Modern control theory[M]. Berlin: Springer, 2005.

[20] Muhl M, Gienger M, Hellbach S, et al. Task-level imitation learning using variance-based movement optimization[C]//2009 IEEE International Conference on Robotics and Automation. IEEE, 2009: 1177-1184.

[21] Asfour T, Azad P, Gyarfas F, et al. Imitation learning of dual-arm manipulation tasks in humanoid robots[J]. International Journal of Humanoid Robotics, 2008, 5(02): 183-202.

[22] Zeng C, Yang C, Li Q, et al. Research progress on human-robot skill transfer[J]. Acta Automatica Sinica, 2019, 45(10): 1813-1828.

[23] van den Berg J. Iterated LQR smoothing for locally-optimal feedback control of systems with non-linear dynamics and non-recursive cost[C]//2014 American Control Conference. IEEE, 2014: 1912-1918.

[24] Todorov E, Li W. A generalized iterative LQG method for locally-optimal feedback control of constrained nonlinear stochastic systems[C]//Proceedings of the 2005, American Control Conference, 2005. IEEE, 2005: 300-306.

[25] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level control through deep reinforcement learning[J]. nature, 2015, 518(7540): 529-533.

[26] Lillicrap T P, Hunt J J, Pritzel A, et al. Continuous control with deep reinforcement learning[J]. arXiv preprint arXiv:1509.02971, 2015.
[27] Schulman J, Levine S, Abbeel P, et al. Trust region policy optimization[C]//International conference on machine learning. 2015: 1889-1897.
[28] Boyd S, Boyd S P, Vandenberghe L. Convex optimization[M]. Cambridge university press, 2004.
[29] Wang H, Banerjee A. Bregman alternating direction method of multipliers[C]//Advances in Neural Information Processing Systems. 2014: 2816-2824.
[30] Boyd S, Parikh N, Chu E. Distributed optimization and statistical learning via the alternating direction method of multipliers[M]. Now Publishers Inc, 2011.
[31] Miyashita M, Yano S, Kondo T. Mirror descent search and its acceleration[J]. Robotics and Autonomous Systems, 2018, 106: 107-116.
[32] Beck A, Teboulle M. Mirror descent and nonlinear projected subgradient methods for convex optimization[J]. Operations Research Letters, 2003, 31(3): 167-175.
[33] Deisenroth M P, Neumann G, Peters J. A survey on policy search for robotics[M]. now publishers, 2013.
[34] Bishop C M. Pattern recognition and machine learning[M]. Springer, 2006.
[35] Weisberg S. Applied linear regression[M]. John Wiley & Sons, 2005.
[36] Khansari-Zadeh S M, Billard A. BM: An iterative algorithm to learn stable non-linear dynamical systems with gaussian mixture models[C]//2010 IEEE International Conference on Robotics and Automation. IEEE, 2010: 2381-2388.
[37] Theodorou E, Stulp F, Buchli J, et al. An iterative path integral stochastic optimal control approach for learning robotic tasks[J]. IFAC Proceedings Volumes, 2011, 44(1): 11594-11601.
[38] Theodorou E, Buchli J, Schaal S. A generalized path integral control approach to reinforcement learning[J]. The Journal of Machine Learning Research, 2010, 11: 3137-3181.
[39] Stulp F, Sigaud O. Path integral policy improvement with covariance matrix adaptation[J]. arXiv preprint arXiv:1206.4621, 2012.
[40] Peters J, Mülling K, Altun Y. Relative entropy policy search[C]//AAAI. 2010, 10: 1607-1612.
[41] Schenck C, Fox D. Guided policy search with delayed sensor measurements[J]. arXiv preprint arXiv:1609.03076, 2016.
[42] Sun B, Xiong F, Liu Z, et al. A Linear Online Guided Policy Search Algorithm[C]//International Conference on Neural Information Processing. Springer, Cham, 2017: 434-441.
[43] Xiong F, Sun B, Yang X, et al. Guided policy search for sequential multitask learning[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2018, 49(1): 216-226.
[44] Sun W, Gordon G J, Boots B, et al. Dual policy iteration[C]//Advances in Neural Information Processing Systems. 2018: 7059-7069.