2020 20th International Conference on Control, Automation and Systems (ICCAS 2020)
Oct. 13∼16, 2020; BEXCO, Busan, Korea

Mixed Reinforcement Learning for Efficient Policy Optimization in Stochastic Environments

Yao Mu[1], Baiyu Peng[1], Ziqing Gu[1], Shengbo Eben Li[1*], Chang Liu[2],
Bingbing Nie[1], Jianfeng Zheng[3], Bo Zhang[3]

[1]State Key Laboratory of Automotive Safety and Energy, School of Vehicle and Mobility, Tsinghua University,
Beijing, 100084, China (email:lishbo@tsinghua.edu.cn) * Corresponding author
[2]Sibley School of Mechanical and Aerospace Engineering, Cornell University,
New York, 14853, USA (email:cl775@cornell.edu)
[3]Smart Transportation Division, Didi Chuxing, 100000, China (email: zhengjianfeng@didiglobal.com)

Abstract: Reinforcement learning has the potentials of successfully control stochastic nonlinear environments in optimal manners. We propose a mixed reinforcement learning (mixed RL) algorithm by simultaneously using dual representations of environmental dynamics to search the optimal policy. Such a design has the capability of improving both learning accuracy and training speed. The dual representation includes an empirical dynamic model and a set of state-action data. The former can embed the designer's knowledge and reduce the difficulty of learning, and the latter can be used to compensate the model inaccuracy since it reflects the real system dynamics accurately. In the mixed RL framework, the additive uncertainty of stochastic model is compensated by using explored state-action data via iterative Bayesian estimator (IBE). The optimal policy is then computed in an iterative way by alternating between policy evaluation (PEV) and policy improvement (PIM). The effectiveness of mixed RL is demonstrated by a typical optimal control problem of stochastic non-affine nonlinear systems (i.e., double lane change task with an automated vehicle).

Keywords: Reinforcement learning, Bayesian estimation, Dynamic model, State-action pairs

1. INTRODUCTION

Reinforcement learning (RL) has been successfully applied in a variety of challenging tasks, such as Go game and robotic control [1, 2]. The increasing interest in RL is primarily stimulated by its data-driven nature, which requires little prior knowledge of the environmental dynamics, and its combination with powerful function approximators, e.g. deep neural networks. In spite of these advantages, many purely data-driven RL suffers from slow convergence rate in continuous action space of stochastic systems, which hinders its widespread adoption in real-world applications [3, 4].

Human beings can learn the optimal policy and achieve goals in a complex environment without much interaction since they can abstract prior knowledge from the physical world to construct a model. This corresponds to a class of model-based algorithms in RL, i.e., model-driven approaches [4–9], which search the optimal policy with known environmental model. The model can be constructed from learning data by approximate function, e.g., deep neural network, or obtained from the physical process. These methods have shown faster convergence and higher sample efficiency compared to the data-driven counterparts.

Model-based approaches can tackle the low-efficiency problem of model-free algorithms by combining model and data. Prior works can be divided into following categories, including Dyna-like algorithms [10, 11], value expansion, adaptive programming [12, 13], and sampling-based planning. Dyna-like algorithms alternate between model learning from environmental interaction, data generation and policy improvement by model-free methods, e.g., ME-TRPO [14]. Although these algorithms significantly improve the data efficiency, they cannot change the convergence speed of the RL algorithm. Value expansion algorithm utilizes the dynamic model to improve the estimation of the cumulative return [4], typically via the use of efficient model ensemble technology [6]. Such a method to improve the value estimation accuracy has been widely adopted to combine with other model-based algorithms. The adaptive dynamic programming is built upon the use of analytic gradient of state transition to control input that is calculated from the dynamic model. The dynamic model can be either learned from the interaction data or an analytical model derived from the first principle. The optimal policy is then learned via backpropagation. Typical algorithms in this category include dreamer [9], PILCO [7], iLQG [8, 15] , and SVG [16]. These algorithms converge faster than model-free methods and Dyna-like algorithms, however they usually suffer from the gradient instability caused by model mismatch and the time-varying characteristic of the models. The idea of sampling-based planning is to choose the best action by a large number of samples, and regard it as the objective for policy network. Representative algorithms include the cross-entropy method [17] in continuous space, which is used in PlaNet [18], and MCTS [19] in discrete space. Although sampling-based approaches can alleviate the problem caused by the gradient instability, such method significantly increases the computational overhead. While these four categories of model-based approaches have achieved significant progress over the past few years, their performance inevitably suffers from the model mismatch: the learned dynamic models are inclined to overfit on the local dynamics, while the physical models have inherent modeling error. These possible overfitting and model inaccuracy usually result in a locally optimal but globally unsatisfactory policy and cause an unstable training process, severely limiting the applicability of model-based RL approaches.

To overcome these challenges, this paper proposes a mixed reinforcement learning (mixed RL) algorithm that utilizes the dual representations of environmental dynamics to improve both learning accuracy and training speed. The empirical model is used as the prior information to reduce the difficulty to learn a model with satisfying generalization ability and avoid overfitting, while the model inaccuracy is iteratively compensated by interaction data using Bayesian estimation. Precisely, the contributions of this paper are as follows,

1). A dual representation of environmental dynamics, which efficiently combines the designer's knowledge and explored data, is designed to improve the model accuracy and computational efficiency in RL.

2). A mixed RL algorithm is developed by embedding an iterative Bayesian estimator (IBE) into the policy iteration process, which has superior performances on convergence speed and policy accuracy.

The rest of this paper is organized as follows. The mixed RL problem is formulated in Section 2. The mixed representation of environmental dynamics is the proposed in Section 3. The mixed RL algorithm, as well as the parametrization of the policy and value function, is developed in Section 4. The effectiveness of mixed RL problem using the double lane change task with a automated vehicle is evaluated in Section 5. Section 6. concludes this paper.

2. PROBLEM DESCRIPTION

Considering a stochastic system, it consists of a deterministic environment and additive uncertainty, where the actual dynamic is mathematically described as

$$\begin{aligned} &x_{t+1} = f(x_t, u_t) + \xi_t, \\ &\xi_t \sim N(\mu, \mathcal{K}) \end{aligned} \tag{1}$$

where t is the current time, $x_t \in \mathcal{X} \subset \mathbb{R}^{\setminus}$ is the state, $u_t \in \mathcal{U} \subset \mathbb{R}^{\Updownarrow}$ is the action, $f(\cdot,\cdot)$ is the deterministic part of environmental dynamics, $\xi_t \in \mathbb{R}^n$ is the additive stochastic uncertainty with unknown mean $\mu \in \mathbb{R}^n$ and covariance $\mathcal{K} \in \mathbb{R}^{n \times n}$. In this study, we assume that the additive stochastic uncertainty ξ_t follows the Gaussian distribution and $\mathbb{E}\{|\xi_t|\} < \infty$. Parameters μ and $\mathcal{K}$ can be completely independent of (x, u) or form a functional relationship with (x, u).

As shown in Fig. 1, actual environmental dynamic contains both deterministic part $f(\cdot,\cdot)$ and uncertain part ξ_t, where $p(\xi_t)$ is the probability density of ξ_t and $p(x_{t+1})$ is the probability density of x_{t+1} under given (x_t, u_t).

The objective of mixed RL is to minimize the expectation of cumulative cost under the distribution of additive stochastic uncertainty ξ, shown as (2):

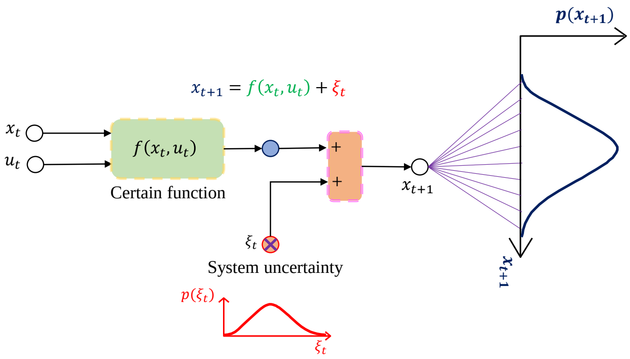

Fig. 1 Dynamics for the stochastic environment.

$$\min_{\pi} V(x_t) = \mathbb{E}_{\xi}\left\{\sum_{k=0}^{\infty} \gamma^k l\left(x_{t+k+1}, u_{t+k}\right)\right\}, \tag{2}$$

where π is the policy, $V(\cdot)$ is the state value, which is a function of current state x_t. $l(\cdot,\cdot) \geq 0$ is the utility function, which is positive definite. γ is the discounting factor limited to 0-1, and $\mathbb{E}_{\xi}(.)$ is the expectation w.r.t. the additive stochastic uncertainty ξ. Especially, the policy is a deterministic mapping:

$$u_t = \pi(x_t) \tag{3}$$

The optimal cost function is defined as

$$V^*(x_t) = \text{lnf}_{\{u_t, u_{t+1}, \ldots, u_{\infty}\}} V(x_t) \tag{4}$$

where $\{u_t, u_{t+1}, \ldots, u_{\infty}\}$ is the action sequence starting from time t. In mixed RL, the self-consistency condition (5) is used to describe the relationship of state values between current time and next time:

$$V(x_t) = \mathbb{E}_{\xi}\left\{l\left(x_{t+1}, u_t\right) + \gamma V\left(x_{t+1}\right)\right\} \tag{5}$$

By using Bellman's principle of optimality. we have the well-known Bellman equation:

$$V^*(x_t) = \min_{u_t}\left\{\mathbb{E}_{\xi}\left\{l\left(x_{t+1}, u_t\right) + \gamma V^*\left(x_{t+1}\right)\right\}\right\} \tag{6}$$

The Bellman equation implies that optimal policy can be calculated in a step-by-step backward mechanism. Therefore, optimal action is

$$\pi^*(x_t) \overset{\text{def}}{=} \arg\min_{u_t}\left\{\mathbb{E}_{\xi}\left\{l(x_{t+1}, u_t) + \gamma V^*\left(x_{t+1}\right)\right\}\right\} \tag{7}$$

where $\pi^*(\cdot)$ represent the optimal policy that maps from an arbitrary state x to its optimal action u^*. Similar to other indirect RL problems, mixed RL aims to find an optimal policy by minimizing cost (2) subject to the constraints of environmental dynamics. The searching procedure can be replaced by solving the Bellman equation in an iterative way. The performance of the generated policy depends on the accuracy of the representation of the environmental dynamics. In fact, either an empirical model or state-action samples $(x_1, u_1, \ldots, x_t, u_t, \ldots)$ can be an useful representation, which corresponds to the so-called model-driven RL and data-driven RL, respectively. The empirical model is usually inaccurate due to environmental uncertainties, which will impair the optimality of the generated policy. The state-action samples, on the other hand, have low sampling efficiency and will slow down the training process.

3. DUAL REPRESENTATION OF ENVIRONMENTAL DYNAMICS

In mixed RL, the environmental dynamics are dually represented by both an empirical model $\mathcal{M}$ and state-action data $\mathcal{D}$. The former represents the designer's knowledge about the environmental dynamics. It is defined in the whole state-action space and can be used to accelerate the training speed and improve the generalization ability. The latter comes from the real interaction between the agent and the environment. It is generally more accurate than $\mathcal{M}$, and therefore can compensate the model mismatch and improve the estimation of the uncertain part in the analytical model. Such dual representation can have accelerated training compared to purely data-driven RL while achieving better policy satisfaction than purely model-driven counterpart.

The empirical model $\mathcal{M}$ is similar to (1):

$$\begin{aligned} &\mathcal{M} = \left\{x_{t+1} = f(x_t, u_t) + \xi_t^{\mathcal{M}}\right\} \\ &\xi_t^{\mathcal{M}} \sim N\left(\mu_{\mathcal{M}}, \mathcal{K}_{\mathcal{M}}\right) \end{aligned} \tag{8}$$

where the mean $\mu_{\mathcal{M}}$ and covariance $\mathcal{K}_{\mathcal{M}}$ of $\xi_t^{\mathcal{M}}$ are given in advance by designers. The given distribution can be quite different from actual distribution due to the modelling errors. Here, $\mu_{\mathcal{M}}$ and $\mathcal{K}_{\mathcal{M}}$ are taken as the prior knowledge of environmental dynamics.

The state-action data, i.e., a sequence of triples (x_j, u_j, x_{j+1}), is denoted by $\mathcal{D}$:

$$\mathcal{D} = \left\{\left(x_j^{\mathcal{D}}, u_j^{\mathcal{D}}, x_{j+1}^{\mathcal{D}}\right), j = 1, 2, \ldots, N\right\} \tag{9}$$

where $x_j^{\mathcal{D}}$ is the j-th state in $\mathcal{D}$, $u_j^{\mathcal{D}}$ is the j-th action in $\mathcal{D}$, and N is the length of data samples. Obviously, the measured data also inherently contain the distribution information of ξ, and are taken as the posterior knowledge of environmental dynamics.

If the environmental dynamics is exactly known, optimal policy $\pi^*(\cdot)$ can be computed by only using the dynamic model, which is also the most efficient RL. However, the exact model is inaccessible in reality, and thus the generated policy might not converge to $\pi^*(\cdot)$. Although collecting samples $\mathcal{D}$ is less efficient, it can be quite accurate to represent the environment, thus being able to improve the generated policy. Therefore, the mixed representation is able to utilize advantages of both model $\mathcal{M}$ and data $\mathcal{D}$ to improve training efficiency and policy accuracy.

Improve model $\mathcal{M}$ by using data $\mathcal{D}$:

We utilize data samples to improve the estimation of the additive stochastic uncertainty ξ in the analytical model $\mathcal{M}$. The uncertainty that inherently exists in a state-action triple is equal to

$$\xi_j^{\mathcal{D}} = x_{j+1}^{\mathcal{D}} - f(x_j^{\mathcal{D}}, u_j^{\mathcal{D}}) \tag{10}$$

A Bayesian estimator is adopted to fuse the distribution information of the additive stochastic uncertainty from both model $\mathcal{M}$ and data $\mathcal{D}$. The Bayesian estimator aims to maximize the posterior probability $p\left(\mu, \mathcal{K}|\mathcal{D}\right)$. In general, we introduce $p(\mu)$ and $p(\mathcal{K})$ as the the prior distribution of μ and $\mathcal{K}$, then the maximum likelihood problem becomes,

$$\begin{aligned} &\max_{\mu,\mathcal{K}} \left\{p\left(\mu, \mathcal{K}|\mathcal{D}\right)\right\} \\ \Leftrightarrow &\max_{\mu,\mathcal{K}} \left\{p\left(\mathcal{D}|\mu, \mathcal{K}\right) p(\mu) p(\mathcal{K})\right\} \end{aligned} \tag{11}$$

Under the assumption that data $\mathcal{D}$ is iid, (11) can be rewritten into an iterative form,

$$\begin{aligned} &\max_{\mu,\mathcal{K}} \left\{p\left(\xi_k^D|\mu, \mathcal{K}\right) p\left(\mathcal{D}_{k-1}|\mu, \mathcal{K}\right) p(\mu) p(\mathcal{K})\right\} \\ &\mathcal{D}_{k-1} = \left\{\xi_1^D, \xi_2^D, \ldots, \xi_{k-1}^D\right\} \end{aligned} \tag{12}$$

Therefore, we can build an iterative Bayesian estimator IBE$(\cdot)$ with the following general form,

$$\begin{bmatrix} \mu_k \\ \mathcal{K}_k \end{bmatrix} = \text{IBE}\left(\mu_{k-1}, \mathcal{K}_{k-1}, \xi_k^D\right) \tag{13}$$

Here, we discuss two simplified cases of the Bayesian estimator:

Case 1: Assume that the covariance $\mathcal{K}$ is known and μ is independent from x and u, we introduce $\mu \sim N(\mu_{\mathcal{M}}, \mathcal{K}_{\mathcal{M}})$ provided by model $\mathcal{M}$ as the prior distribution of μ. Thus, the objective function $\mathcal{L}$ of Bayesian estimation becomes,

$$\begin{aligned} \mathcal{L} =& \log\left\{\mathrm{p}(\mathcal{D}|\mu)\mathrm{p}(\mu)\right\} \\ =& \frac{1}{2}\left(\mu - \mu_M\right)^T \mathcal{K}_M^{-1}\left(\mu - \mu_M\right) \\ &+ \frac{1}{2}\sum_{j=1}^{N}\left(\xi_j^D - \mu\right)^T \mathcal{K}^{-1}\left(\xi_j^D - \mu\right) + \mathcal{C} \end{aligned} \tag{14}$$

where $p(\mu) = \mathcal{N}(\mu_{\mathcal{M}}, \mathcal{K}_{\mathcal{M}})$ is the prior distribution and $\mathcal{C}$ is a constant. The optimal estimation of μ is calculated by (15).

$$\hat{\mu} = \left(\mathcal{K}_M^{-1} + N\mathcal{K}^{-1}\right)^{-1}\left(\mathcal{K}_M^{-1}\mu_M + \mathcal{K}^{-1}\sum_{j=1}^{N}\xi_j^D\right) \tag{15}$$

The $\hat{\mu}$ can be iteratively computed by using IBE. Define $\Psi_k = \mathcal{K}_{\mathcal{M}}^{-1} + k\mathcal{K}^{-1}$, and $m_k = \sum_{j=1}^{k}\xi_j^D$, the iterative Bayesian estimator IBE$(\cdot)$ is

$$\begin{aligned} &\Psi_k = \Psi_{k-1} + \mathcal{K}^{-1}, \quad m_k = m_{k-1} + \xi_k^D \\ &\hat{\mu}_k = \left(\Psi_k\right)^{-1}\left(\mathcal{K}_{\mathcal{M}}^{-1}\mu_{\mathcal{M}} + \mathcal{K}^{-1}m_k\right) \end{aligned} \tag{16}$$

Case 2: Assume that both the mean μ and covariance $\mathcal{K}$ are unknown. The same prior distribution in case 1 is applied to μ. The covariance $\mathcal{K}$ is estimated by the maximum likelihood estimation, since the parameters of the prior distribution of K are inconvenient to determine by human designer. Subsequently, the optimal estimation of μ and $\mathcal{K}$ are as follows,

$$\begin{aligned}
\hat{\mu} &= \left(\mathcal{K}_{\mathcal{M}}^{-1} + N\hat{\mathcal{K}}^{-1}\right)^{-1}\left(\mathcal{K}_{\mathcal{M}}^{-1}\mu_{\mathcal{M}} + \hat{\mathcal{K}}^{-1}\sum_{j=1}^{N}\xi_j^D\right) \\
\hat{\mathcal{K}} &= \frac{1}{N}\sum_j\left(\xi_j^D - \hat{\mu}\right)\left(\xi_j^D - \hat{\mu}\right)^T
\end{aligned} \tag{17}$$

Define $\Psi_k = \mathcal{K}_{\mathcal{M}}^{-1} + k\hat{\mathcal{K}}_k^{-1}$ and $m_k = \sum_{j=1}^{k}\xi_j^D$. Then $\hat{\mu}$ and $\hat{\mathcal{K}}$ can be iteratively computed by the following IBE,

$$\begin{aligned}
\Psi_k &= \Psi_{k-1} + \hat{\mathcal{K}}_{k-1}^{-1} \\
m_k &= m_{k-1} + \xi_k^D \\
\hat{\mu}_k &= (\Psi_k)^{-1}\left(\mathcal{K}_M^{-1}\mu_M + \hat{\mathcal{K}}_{k-1}^{-1}m_k\right) \\
\hat{\mathcal{K}}_k &= \frac{1}{k}\left\{(k-1)\hat{\mathcal{K}}_{k-1} + \left(\xi_k^D - \hat{\mu}_{k-1}\right)\left(\xi_k^D - \hat{\mu}_{k-1}\right)^T\right\}
\end{aligned} \tag{18}$$

For more general cases where ξ_t is related to x_t and u_t, i.e.,

$$\begin{aligned}
&\xi_t \sim N(\mu_t, \mathcal{K}_t) \\
&\mu_t, \mathcal{K}_t = \phi(x_t, u_t; w_\phi)
\end{aligned} \tag{19}$$

where $\phi(\cdot,\cdot)$ is a general function with parameter w_ϕ. Our goal is to infer the parameters $\psi = (\mu_w, \sigma_w)$, which determine the distribution of $w_\phi \sim N(\mu_w, \sigma_w)$. Thus, the IBE becomes the estimator of ψ, and variational Bayesian inference [20] could be used to solve such a problem, that is to minimize the KL-divergence between $p(w_\phi|\mathcal{D})$ and $q(w_\phi|\psi)$.

$$\begin{aligned}
&D_{KL}[q(w_\phi|\psi)|p(w_\phi|\mathcal{D})] \\
=&\int q(w_\phi|\psi)\log\frac{q(w_\phi|\psi)}{p(w_\phi|\mathcal{D})}dw \\
=&D_{KL}[q(w_\phi|\psi)|p(w_\phi)] - \mathbb{E}_q\{\log p(\mathcal{D}|w_\phi)\} + C
\end{aligned} \tag{20}$$

where $p(w_\phi)$ is the prior distribution of w_ϕ, C is a constant, and the objective function L can be simplified as

$$L = D_{KL}[q(w_\phi)|p(w_\phi)] - \mathbb{E}_q\{\log(D|W)\} \tag{21}$$

To solve the above problem in an iterative manner, the gradient descent method is used to find the optimal ψ, and reparametrization method ($w_\phi = \mu_w + \epsilon\sigma_w$) is utilized to ensure the objective function could be optimized via gradient propagation.

$$\psi^{k+1} = \psi^k - \alpha\frac{\partial L}{\partial w_\phi}\frac{\partial w_\phi}{\partial \psi} \tag{22}$$

4. MIXED RL ALGORITHM

4.1 Mixed RL Algorithm Framework

Existing RL algorithms that compute the optimal policy via the use of Bellman equation are known as indirect RL, and they usually involve PEV and PIM steps. Different from traditional indirect RL algorithms, mixed RL consists of three alternating steps, i.e., IBE, PEV and PIM, as shown in Fig. 2. IBE that is proposed in Section 3. is used to estimate the mean and covariance of the additive stochastic uncertainty iteratively. PEV seeks to solve a group of algebraic equations numerically governed by the self-consistency condition (5) under current-step policy π, and PIM is to search a better policy by minimizing a "weak" Bellman equation.

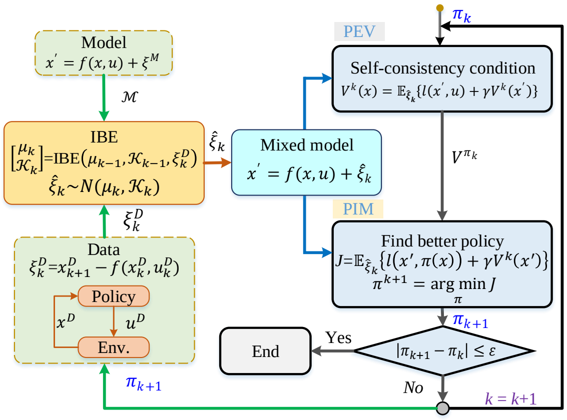
Model
$x' = f(x,u) + \xi^M$
$\mathcal{M}$
IBE
$\begin{bmatrix}\mu_k \\ \mathcal{K}_k\end{bmatrix}$=IBE$(\mu_{k-1}, \mathcal{K}_{k-1}, \xi_k^D)$
$\hat{\xi}_k \sim N(\mu_k, \mathcal{K}_k)$
$\hat{\xi}_k$
ξ_k^D
Data
$\xi_k^D = x_{k+1}^D - f(x_k^D, u_k^D)$
Policy
x^D
u^D
Env.
π_{k+1}
Mixed model
$x' = f(x,u) + \hat{\xi}_k$
PEV
π_k
Self-consistency condition
$V^k(x) = \mathbb{E}_{\hat{\xi}_k}\{l(x',u) + \gamma V^k(x')\}$
V^{π_k}
PIM
Find better policy
$J = \mathbb{E}_{\hat{\xi}_k}\{l(x', \pi(x)) + \gamma V^k(x')\}$
$\pi^{k+1} = \arg\min_\pi J$
π_{k+1}
$|\pi_{k+1} - \pi_k| \le \varepsilon$
Yes
End
No
$k = k+1$

Fig. 2 The framework of the mixed RL algorithm.

In the first step, IBE calculates μ_k and $\mathcal{K}_k$ with the latest data ξ_k^D and the mixed model is updated accordingly, i.e.,

$$x' = f(x,u) + \hat{\xi}_k, \quad \hat{\xi}_k \sim N(\mu_k, \mathcal{K}_k) \tag{23}$$

where $[\mu_k, \mu_k]^T = \text{IBE}\left(\mu_{k-1}, \mathcal{K}_{k-1}, \xi_k^D\right)$ is defined in (13) in the cases that ξ is independent to (x,u) and $[\mu_k, \mu_k]^T = \phi(x_t, u_t; w_\phi)$ that defined in (19) in the cases that ξ is correlated to (x,u). The optimal policy is searched by policy iteration with the mixed model (23). In the second step, PEV solves (24) under the estimated distribution of ξ:

$$V^k(x) = \mathbb{E}_{\hat{\xi}_k}\left\{l\left(x', \pi^k(x)\right) + \gamma V^k\left(x'\right)\right\}, \forall x \in X \tag{24}$$

where $\pi^k(x)$ is the current policy at k-step iteration, and $V^k(x)$ is the state value to be solved under policy $\pi^k(x)$. In the third step, PIM computes an improved policy by minimizing (25):

$$\pi^{k+1}(x) = \arg\min_\pi\left\{\mathbb{E}_{\hat{\xi}_k}\left\{l\left(x', \pi(x)\right) + \gamma V^k\left(x'\right)\right\}\right\} \tag{25}$$

where $\pi^{k+1}(x)$ is the new policy. The use of estimated $\hat{\xi}_k$ naturally embeds both empirical model and state-action data into RL, which is able to improve the accuracy of the additive stochastic uncertainty ξ and x' and achieve high convergence speed. The mixed RL algorithm is summarized in Algorithm 1.

4.2 Mixed RL with Parameterized Functions

For large state spaces, both value function and policy are parameterized in mixed RL, as shown in (26). The parameterized value function with known parameter w

Algorithm 1 Mixed RL algorithm

```
Initialize IBE parameters $\hat{\mu}_0 = \mu_M$ and $\hat{\mathcal{K}}_0 = \mathcal{K}_M$
Initialize state $x_0 \in \mathcal{X}$, $k = 0$
repeat
update the mixed model by IBE (13) or (22)
    $x' = f(x, u) + \hat{\xi}_k, \quad \hat{\xi}_k \sim N(\mu_k, \mathcal{K}_k)$
PEV with mixed model:
    $V^k(x) = \mathbb{E}_{\hat{\xi}_k}\{l(x', \pi^k(x)) + \gamma V^k(x')\}$
PIM with mixed model:
    $\pi^{k+1}(x) = \arg\min_{\pi} \left\{\mathbb{E}_{\hat{\xi}_k}\{l(x', \pi(x)) + \gamma V^k(x')\}\right\}$
$k = k + 1$
until $|V^{k+1} - V^k| \le \epsilon$ and $|\pi^{k+1} - \pi^k| \le \epsilon$
```

is called the "critic", and the parameterized policy with known parameter θ is called the "actor" [21].

$$V(x) \cong V(x; w) \quad u \cong \pi(x; \theta) \tag{26}$$

The parameterized critic is to minimize the average square error (27) in PEV, i.e.,

$$J_{\text{critic}} = \mathbb{E}_{\hat{\xi}}\left\{\frac{1}{2}\left(l(x', u_\theta) + \gamma V^k(x'; w) - V^k(x; w)\right)^2\right\} \tag{27}$$

The semi-gradient of the critic is

$$\frac{\partial J_{\text{Critic}}}{\partial w} = \int p(x')\left(V^k(x; w) - V_{\text{target}}\right)\frac{\partial V^k(x; w)}{\partial w} dx' \tag{28}$$

where $V_{\text{target}} = l(x', u_\theta) + \gamma V^k(x')$ is the target of the value function's output.

The parameterized actor is to minimize the "weak" Bellman condition, i.e., to minimize the following objective function,

$$\begin{aligned} J_{\text{Actor}} &= \mathbb{E}_{\hat{\xi}}\{l(x', u_\theta) + \gamma V^k(x')\} \\ p(x'; u_\theta) &\sim N\left(f(x, u_\theta) + \hat{\mu}, \hat{\mathcal{K}}\right) \end{aligned} \tag{29}$$

where $\hat{\mu}$ and $\hat{\mathcal{K}}$ are the mean and covariance of $\hat{\xi}$. The gradient of J_{Actor} is calculated as follows,

$$\begin{aligned} \frac{\partial J_{\text{Actor}}}{\partial \theta} = \int \Bigg\{ & \left[l(x', u_\theta) + \gamma V^k(x')\right]\frac{\partial p(x'; u_\theta)}{\partial \theta} \\ & + \frac{\partial l(x', u_\theta)}{\partial \theta} p(x'; u_\theta)\Bigg\} dx' \end{aligned} \tag{30}$$

In essence, the parameterized method is called generalized policy iteration (GPI). Different from the traditional policy iteration, PEV and PIM each has only one step in GPI, which greatly improves the computational efficiency when RL is combined with neural network.

5. NUMERICAL EXPERIMENTS

In this section, the proposed mixed RL is first applied to a linear system to compare with the model-free method and the LQR algorithm based on the empirical model. Then, it is evaluated on a nonlinear system and compared with typical model-based RL methods.

5.1 Stochastic Linear System

Consider the F16 aircraft system [22] described by $\dot{x} = Ax + Bu + Dd$

$$A = \begin{bmatrix} -1.01887 & 0.90506 & -0.00215 \\ 0.82225 & -1.07741 & -0.17555 \\ 0 & 0 & -1 \end{bmatrix} \tag{31}$$
$$B = [0, 0, 5]^T, \quad D = [1, 0, 0]^T$$

The system state vector is $x = [\alpha \quad q \quad \delta_e]$, where α denotes the angle of attack, q is the pitch rate, and δ_e is the elevator deflection angle. The control input u is the elevator actuator voltage, and the disturbance $d \sim N(0.05, 0.01)$ is the wind gusts on angle of attack. Then with such a stochastic linear system equation, the optimal control problem is defined as

$$\begin{aligned} & \min_u \sum_{t=0}^{\infty} \gamma^t (x_t^T Q x_t + u_t^T R u_t) \\ & \text{s.t. } \dot{x} = Ax + Bu + Dd \end{aligned} \tag{32}$$

where the top left element of the matrix Q is considered to be 20, and all the other elements are zero. It is also assumed here that $R = 1$ and $\gamma = 0.995$. The optimal feedback control law of this stochastic system can be derived by Bellman principle of optimality [23], which is composed of both gain and bias

$$u^* = \begin{bmatrix} 0.8019 & 0.5607 & 0.0608 \end{bmatrix} x + 0.0696 \tag{33}$$

We now implement the Mixed RL Algorithm and compare the mixed RL with PPO2 [24], a widely used model-free algorithm, and the LQR algorithm with the empirical model (LQR-EM). The simulation interval is chosen as $T = 0.005(s)$. Fig. 3 and Fig. 4 show the convergence rate is significantly faster than the PPO2 algorithm and the control performance is very close to its optimal value. In contrast with mixed RL and PPO2, the LQR-EM has higher cost when testing due to the wind gusts which is not considered in the empirical model. These results confirm that the proposed method converges to the optimal solution and outperforms the model-free method and LQR-EM.

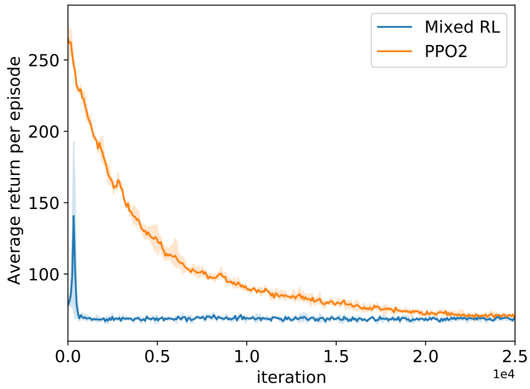

Fig. 3 Comparison of the average return of mixed RL and PPO2 per episode.

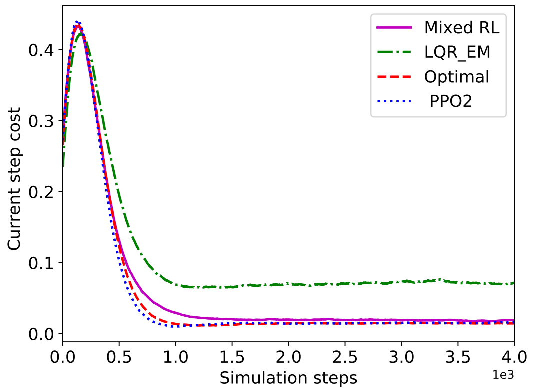

Fig. 4 Control performance comparison of mixed RL, PPO2, LQR-EM, and the optimal controller.

5.2 Stochastic Non-affine Nonlinear System

To demonstrate the advantages of mixed RL in complex systems, we compared the performance of mixed RL with the widely used model-based RL methods, including dyna algorithm with the learned model (Dyna-LM), dyna algorithm with the mixed model (Dyna-MM), adaptive dynamic programming with the learned model (ADP-LM) and adaptive dynamic programming with the empirical model (ADP-EM).

We consider a typical optimal control problem of stochastic non-affine nonlinear systems, i.e., the combined lateral and longitudinal control of an automated vehicle with stochastic disturbance (i.e., the influence of small road slope and road bumps). The vehicle is subjected to random longitudinal interference force F_{dis} in the tracking process and the vehicle dynamics is shown as follows [25],

$$\dot{x} = \begin{bmatrix} \frac{F_{yf}\cos\delta + F_{yr}}{m} - v_x r \\ \frac{aF_{yf}\cos\delta - bF_{yr}}{I_z} \\ a_x + v_y r - \frac{F_{yf}\sin\delta}{m} + \frac{F_{dis}}{m} \\ r \\ v_x \sin\phi + v_y \cos\phi \end{bmatrix} \tag{34}$$

where the state $x = \begin{bmatrix} v_y & r & v_x & \phi & y \end{bmatrix}^T$, v_y is the lateral velocity, r is the yaw rate, v_x is the difference between longitudinal velocity and desired velocity, ϕ is the yaw angle, and y is the distance between vehicle's centroid and the target trajectory. For the control input $u = [\delta \quad a_x]^T$, where δ is the front wheel angle and a_x is the longitudinal acceleration. The F_{yf} and F_{yr} are the lateral tire forces of the front and rear tires respectively, which are calculated by the Fiala tire model [26]. In the tire model,the tire-road friction coefficient μ is set as 1.0. The front wheel cornering stiffness and rear wheel cornering stiffness are set as 88000 N/rad and 94000 N/rad respectively. The mass m is set as 1500 kg, the a and b are the distances from centroid to front axle and rear axle, and set as 1.14 m and 1.40 m respectively. The polar moment of inertia I_z at centroid is set as 2420 N/rad. The random longitudinal interference force $F_{dis} \sim N(261, 32)$ and the desired velocity is set as 12 m/s [27].

For comparison purposes, a double-lane change task is simulated respectively with three different RL algorithms. The task is to track the desired trajectory in the lateral direction while maintaining the desired longitudinal velocity under the longitudinal interference F_{dis}. Hence, the optimal control problem with discretized stochastic system equation is given by

$$\begin{aligned} &\min_u \sum_{t=0}^{\infty} \gamma^t \left(45\left(v_x - 12\right)^2 + 60y^2 + u^\top \begin{bmatrix} 800 & 0 \\ 0 & 1 \end{bmatrix} u \right) \\ &\quad s.t. \quad x_{t+1} = f\left(x_t, u_t\right) + \xi_t, \quad \xi_t = F_{dis}T/m \end{aligned} \tag{35}$$

where $\gamma = 0.99$ is the discounting factor, $f(\cdot, \cdot)$ is the deterministic part of the discretized system equation of (47), ξ_t is the additive stochastic uncertainty and the simulation time interval T is set as $0.005(s)$.

The convergence performance of the five algorithms mentioned above are illustrated in Fig.5. The adaptive methods (i.e., mixed RL, ADP-EM and ADP-LM) converge faster than the Dyna-like algorithms (i.e., Dyna-LM, Dyna-EM), which demonstrates the advantage of the utilization of the analytical gradient given by the dynamic model. Moreover, the mixed RL outperforms the other algorithms by having a superior convergence rate: it converges almost twice faster than the ADP-LM without oscillation. The reason why ADP-LM converges slower than both the mixed RL and ADP-EM, is the mismatch of the data distributions between two adjacent iterations and the switching characteristics of the system, which lead to the difficulties to learn an accurately enough model purely from data. All the above results confirm the effectiveness of the designer's knowledge embedded in mixed RL.

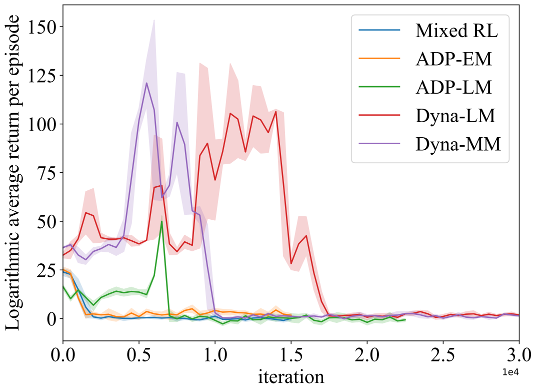

Fig. 5 Convergence rate comparison between mixed RL, model-driven RL, and data-driven RL.

It can also be noticed that ADP-EM achieves a similar convergence rate as the mixed RL. However, the control performance of ADP-EM is impaired by the model inaccuracy. To illustrate this point, we test the policies calculated by five methods in the double lane change task. As shown in Fig. 6, all five policies could stably track the target trajectory, while the tracking error is different.

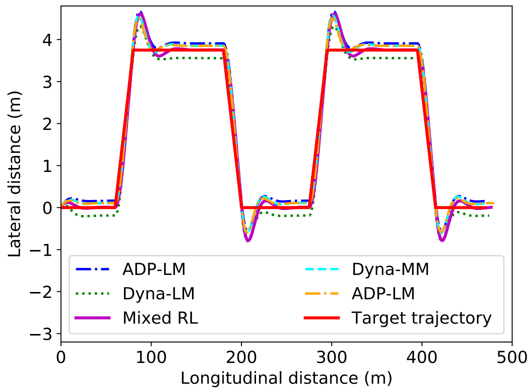

Fig. 6 Tracking performance comparison among five RL methods. The red solid line shows the reference trajectory that the vehicle needs to follow.

In particular, as shown in Fig.7, the mixed RL has the minimum longitudinal speed error, since it enables the vehicle to decelerate rapidly at sharp turns and adjust back appropriately after passing the turns. In contrast, because of the model inaccuracy, the policies generated by ADP-EM have highest speed error due to the insufficient deceleration when making turns.

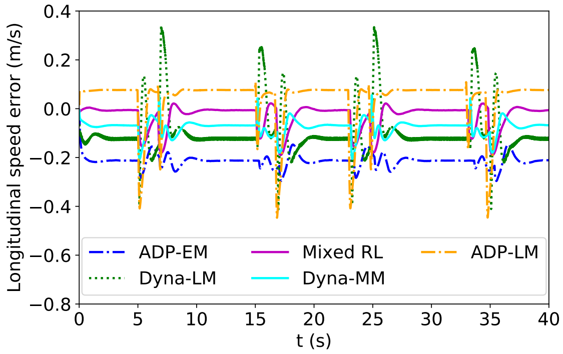

Fig. 7 Longitudinal speed error.

The mixed RL also outperforms the other five benchmark methods in terms of the lateral position error. As shown in Fig.8, the mixed RL has the minimum steady-state lateral position error, while the Dyna-LM and ADP-EM has the larger lateral position error. It is worth noting that, when making sharp turns, the mixed RL generates larger lateral position error than most benchmark methods. This is, however, an expected and desirable overshoot that is commonly observed in RL controllers, as it allows the vehicle to rapidly adjust its state so as to accurately track the trajectories in the regions that change mildly and smoothly.

In summary, mixed RL exhibits the fastest convergence speed during the training process and superior control performance in the given double lane change task. The ADP-EM has a similar convergence speed as the mixed RL, but has higher tracking error due to the model mismatch. Although the ADP-LM compensates the model inaccuracy by iteratively updating the dynamic model, it converges slower than both the mixed RL and ADP-EM due to the difficulties to learn an accurately enough model purely from data. The Dyna-like algorithms have a slower convergence rate than the mixed RL, due to the difficulties in finding the optimal policy only by state-action data.

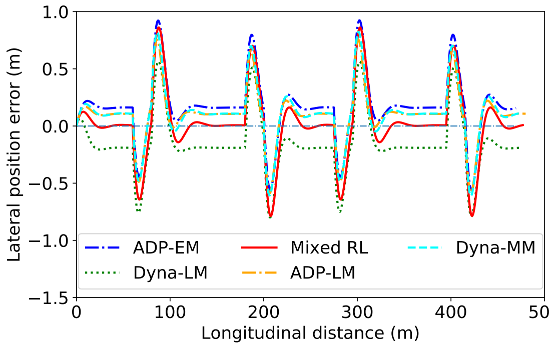

Fig. 8 Lateral position error.

6. CONCLUSION

This paper proposes a mixed reinforcement learning approach with superior performances on convergence speed and policy accuracy for non-linear systems. The mixed RL significantly improves the convergence performance by integrating the designer's knowledge with the real interaction data, and ensures the policy accuracy by embedding the iterative Bayesian estimator into the generalized policy iteration framework. The benefits of mixed RL are demonstrated in simulations using both linear system and non-affine nonlinear system. In particular, the mixed RL is shown to converge to the optimal solution for stochastic linear systems. In controlling the more challenging nonlinear systems, the mixed RL achieves faster convergence rate and more stable training process than model-free methods and the model-based algorithms that generate the policy only with learned models. In addition, the mixed RL has lower policy error than the other model-based methods that only utilize the empirical model, since the system model is refined iteratively by the Bayesian estimation. The application of the mixed RL to more general environmental dynamics and non-Gaussian uncertainties will be investigated in the future.

7. ACKNOWLEDGEMENT

This study is supported by National Key RD Program of China with 2017YFB0102603. Special thanks should be given to Tsinghua University-Didi Joint Research Center for Future Mobility.

REFERENCES

[1] E. Gibney, "Google ai algorithm masters ancient game of go," *Nature News*, vol. 529, no. 7587, p. 445, 2016.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[3] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, "Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7559–7566.

[4] V. Feinberg, A. Wan, I. Stoica, M. I. Jordan, J. E. Gonzalez, and S. Levine, "Model-based value estimation for efficient model-free reinforcement learning," *arXiv preprint arXiv:1803.00101*, 2018.

[5] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine *et al.*, "Model-based reinforcement learning for atari," *arXiv preprint arXiv:1903.00374*, 2019.

[6] J. Buckman, D. Hafner, G. Tucker, E. Brevdo, and H. Lee, "Sample-efficient reinforcement learning with stochastic ensemble value expansion," in *Advances in Neural Information Processing Systems*, 2018, pp. 8224–8234.

[7] M. Deisenroth and C. E. Rasmussen, "Pilco: A model-based and data-efficient approach to policy search," in *Proceedings of the 28th International Conference on machine learning (ICML-11)*, 2011, pp. 465–472.

[8] E. Todorov and W. Li, "A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems," in *Proceedings of the 2005, American Control Conference, 2005.* IEEE, 2005, pp. 300–306.

[9] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.

[10] R. S. Sutton, "Dyna, an integrated architecture for learning, planning, and reacting," *ACM Sigart Bulletin*, vol. 2, no. 4, pp. 160–163, 1991.

[11] R. S. Sutton, C. Szepesvári, A. Geramifard, and M. P. Bowling, "Dyna-style planning with linear function approximation and prioritized sweeping," *arXiv preprint arXiv:1206.3285*, 2012.

[12] T. Bian, Y. Jiang, and Z.-P. Jiang, "Adaptive dynamic programming and optimal control of nonlinear nonaffine systems," *Automatica*, vol. 50, no. 10, pp. 2624–2632, 2014.

[13] D. P. Bertsekas, D. P. Bertsekas, D. P. Bertsekas, and D. P. Bertsekas, *Dynamic programming and optimal control*. Athena scientific Belmont, MA, 1995, vol. 1, no. 2.

[14] T. Kurutach, I. Clavera, Y. Duan, A. Tamar, and P. Abbeel, "Model-ensemble trust-region policy optimization," *arXiv preprint arXiv:1802.10592*, 2018.

[15] W. Li and E. Todorov, "An iterative optimal control and estimation design for nonlinear stochastic system," in *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 3242–3247.

[16] N. Heess, G. Wayne, D. Silver, T. Lillicrap, T. Erez, and Y. Tassa, "Learning continuous control policies by stochastic value gradients," in *Advances in Neural Information Processing Systems*, 2015, pp. 2944–2952.

[17] J. Bekker, "Applying the cross-entropy method in multi-objective optimisation of dynamic stochastic systems," Ph.D. dissertation, Stellenbosch: Stellenbosch University, 2012.

[18] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Learning latent dynamics for planning from pixels," *arXiv preprint arXiv:1811.04551*, 2018.

[19] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[20] M. A. Chappell, A. R. Groves, B. Whitcher, and M. W. Woolrich, "Variational bayesian inference for a nonlinear forward model," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 223–236, 2008.

[21] 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*, 2000, pp. 1057–1063.

[22] J. Duan, S. E. Li, Z. Liu, M. Bujarbaruah, and B. Cheng, "Generalized policy iteration for optimal control in continuous time," *arXiv preprint arXiv:1909.05402*, 2019.

[23] D. P. Bertsekas and S. Shreve, *Stochastic optimal control: the discrete-time case*, 2004.

[24] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[25] S. E. Li, H. Chen, R. Li, Z. Liu, Z. Wang, and Z. Xin, "Predictive lateral control to stabilise highly automated vehicles at tire-road friction limits," *Vehicle System Dynamics*, pp. 1–19, 2020.

[26] Y.-H. J. Hsu, S. M. Laws, and J. C. Gerdes, "Estimation of tire slip angle and friction limits using steering torque," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 4, pp. 896–907, 2009.

[27] S. Xu, S. E. Li, B. Cheng, and K. Li, "Instantaneous feedback control for a fuel-prioritized vehicle cruising system on highways with a varying slope," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1210–1220, 2016.