An Overview of Robust Reinforcement Learning

Shiyu Chen
School of Mechanical Engineering and Automation
Harbin Institute of Technology, Shenzhen
Shenzhen, China
chenshiyu@stu.hit.edu.cn

Yanjie Li†
School of Mechanical Engineering and Automation
Harbin Institute of Technology, Shenzhen
Shenzhen, China
autolyj@hit.edu.cn

***Abstract*—Reinforcement learning (RL) is one of the popular methods for intelligent control and decision making in the field of robotics recently. The goal of RL is to learn an optimal policy of the agent by interacting with the environment via trail and error. There are two main algorithms for RL problems, including model-free and model-based methods. Model-free RL is driven by historical trajectories and empirical data of the agent to optimize the policy, which needs to take actions in the environment to collect the trajectory data and may cause the damage of the robot during training in the real environment. The main different between model-based and model-free RL is that a model of the transition probability in the interaction environment is employed. Thus the agent can search the optimal policy through internal simulation. However, the model of the transition probability is usually estimated from historical data in a single environment with statistical errors. Therefore, an issue is faced by the agent is that the optimal policy is sensitive to perturbations in the model of the environment which can lead to serious degradation in performance. Robust RL aims to learn a robust optimal policy that accounts for model uncertainty of the transition probability to systematically mitigate the sensitivity of the optimal policy in perturbed environments. In this overview, we begin with an introduction to the algorithms in RL, then focus on the model uncertainty of the transition probability in robust RL. In parallel, we highlight the current research and challenges of robust RL for robot control. To conclude, we describe some research areas in robust RL and look ahead to the future work about robot control in complex environments.**

***Index Terms*—robust, reinforcement learning, dynamic programming, transition probability, model uncertainty**

I. Introduction

Reinforcement learning (RL) [1] is concerned about searching a good policy for sequential decision making problems by interacting with the environment via trail and error. Because of the potential intelligence and active learning features, especially incorporating with deep learning [2], RL is widely applied in many practical applications such as game playing [3]–[6], autonomous unmanned driving [7], [8], robot control [9]–[12], etc. According to the principle of RL, Markov decision processes (MDP) [13] usually be considered as the mathematical model to solve the policy optimization problems. Then, the objective of RL agent is to maximize the expectation of the long term return from each state. The transition probability of the states in MDP is important to estimate the value function of the states and actions. However, the model of the transition probability can not be measured exactly in many RL optimization problems. Therefore, model-free RL methods [14]–[19] are adopted to search the optimal policies of the RL agents frequently. Without the model of the transition probability, the policy evaluation and improvement are calculated through the historical trajectories generated by the current policy of the agent. In [15], a replay buffer was used to stored the transitions of the states by interacting with the environment via the policy learned by the RL algorithm. Nevertheless, the interaction between the agents and the environments is expensive in many practical applications, especially in robot control. The mechanical failure may encountered and lead to the interruption of the training during policy learning. However, model-based RL methods [20]–[22] can simulate the transitions of the states using the learned model. Thus, the interaction between the agents and the environments can be avoided, resulting in increased sample efficiency. But, the model of the transition probability is often estimated from historical data in a single environment with statistical errors. In many applications of current practice, the model errors are ignored and training the policy of the agent using the learned model as the true transition probability, which in turn affects the learned policy. Then, the optimal policy learned with the model is quite sensitive to the perturbations in the transition probability and even lead to serious trouble in performance with the real environment.

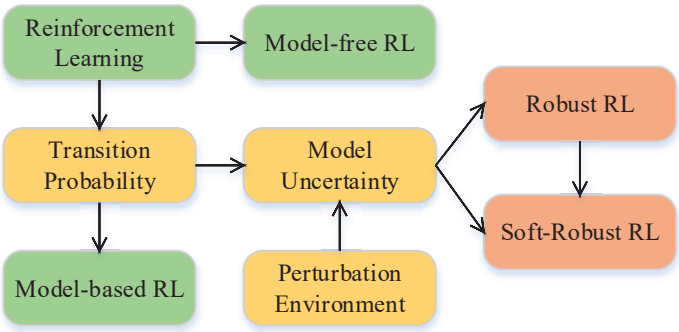

Fig. 1. The relationship between RL and robust RL.

In order to address this issue, the robust formulation of RL approach is proposed to maintain the optimization of

This work is supported by Shenzhen basic research program JCYJ20180507183837726, JCYJ20170811155028832 and National Natural Science Foundation U1813206 and 61977019.

†Corresponding author.

the policy in model uncertainty of the transition probability. Robust RL [23]–[25] aims to learn a robust optimal policy that accounts for model uncertainty of the transition probability to systematically mitigate the sensitivity of the policy in perturbed environments. The relationship between RL and robust RL is shown in Fig. 1. In [26], the worst case of the expected return value function was considered to obtain a robust policy in a conservative form. In addition, to overcome the overly conservative of the robust optimal policy, the soft-robust framework of the policy search was presented in [27]. In this overview, we focus on the model uncertainty of the transition probability in robust RL which influenced by the perturbation of the environment, such as weather conditions, change in mass, sensor noise, etc. Besides, the applications and challenges of robust RL will be discussed.

The rest of this overview is organized as follows. In section II, we formally describe the MDP formalization of RL and the model of transition probability used in RL to optimize the policies. The definition of robust RL and the uncertainty set of transition probability are given in section III, also we describe the robust formulation of RL methods used in policy search and implement the calculation process of robust RL combine with model-based RL. In addition, soft-robust RL is explained to solve the issue of overly conservative of robust RL algorithms in section IV. Then, in section V, current research and challenges of robust RL is discussed. Finally, section VI concludes this overview and proposes some research areas in robust RL for robot control in complex environments.

II. Reinforcement Learning

The main principle of RL is to establish a mapping between the state space S and action space A of agent by interacting with the environment via trail and error, which called a policy $\pi : S \mapsto A$. Then, the goal of RL is to search for the optimal policy (control strategy) π^* of the agent to achieve the maximum expected cumulative return from each state in the state space.

A. Markov Decision Processes

Formally, RL can be described as a MDP, which usually be represented as a six-element tuple $\langle S, A, R, P, \rho_0, \gamma \rangle$, where:

- S is the set of all states in agent and environment.
- A is the set of all actions generated by the agent.
- R denotes the immediate reward function $r(s_t, a_t)$ from the state s_t and action a_t at timestep t.
- P is the distribution of transition probability $p(s_{t+1}|s_t, a_t)$ that map a state-action pair at timestep t onto the next state at timestep $t+1$.
- ρ_0 denotes the distribution of starting states.
- γ is the discount factor which satisfies $\gamma \in [0, 1]$.

Then, the policy search of RL can be converted to a mathematical optimization problem. The value function $V(s)$ and action-value function $Q(s, a)$ are defined as the expected cumulative rewards under current policy π at state s and action a, respectively.

$$V_\pi(s) = \mathbb{E}_{\pi,p}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| s_0 = s\right], \tag{1}$$

$$Q_\pi(s, a) = \mathbb{E}_{\pi,p}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| s_0 = s, a_0 = a\right]. \tag{2}$$

The optimal policy can be obtained by maximizing the value functions through the optimal Bellman equations as follows:

$$V_*(s_t) = \max_{a_t} \mathbb{E}_{s_{t+1}\sim p}\left[r(s_t, a_t) + \gamma V_*(s_{t+1})\right], \tag{3}$$

$$Q_*(s_t, a_t) = \mathbb{E}_{s_{t+1}\sim p}\left[r(s_t, a_t) + \gamma \max_{a_{t+1}} Q_*(s_{t+1}, a_{t+1})\right]. \tag{4}$$

B. Model-free RL

In general, the model of transition probability is not known in many practical problems. Then, in order to search the optimal policy of the agent, the historical trajectories generated by the agent interacts with the environment are collected to iterate for the policy search process. In [15], the authors adopted deep neural networks to estimate the action-value function and a replay buffer was used to store the transitions (s_t, a_t, r_t, s_{t+1}) which were sampled from the environment according to the exploration policy. To improve the current policy, the loss function was defined as the temporal-difference (TD) error:

$$\delta_t^{MF} = r(s_t, a_t) + \gamma Q_{\hat{\phi}}(s_{t+1}, \mu(s_{t+1})) - Q_\phi(s_t, a_t), \tag{5}$$

where $\hat{\phi}$ and ϕ are the target network parameter and current parameter of the action-value function, and $\mu(s_{t+1})$ denotes the deterministic policy at timestep $t+1$. Therefore, the parameters of the action-value function can be updated via stochastic gradient descent (SGD) algorithm [28].

However, since the unstable policy is used for interacting with the environment by trail and error during the collection of the data, the agent may suffer from some unknown fault which can lead to the interruption of the policy search program. Due to the risk of failure during training framework, model-free RL algorithms is generally limited to use in simulation environments, such as OpenAI Gym [29], DeepMind Control Suite [30], Torcs [31], etc.

C. Model-based RL

In addition to simulation applications in the field of game playing, RL methods become more and more popular used for robot control. In order to prevent the agent from interacting with the environment which is expensive for robots, model-based RL approaches are proposed to optimize the control policy of robots with internal simulation. The framework of model-based RL is shown in Fig. 2. In [32]–[34], the model of helicopter's transition dynamics was identified by supervised learning. Then, the trajectories data of the helicopter for policy

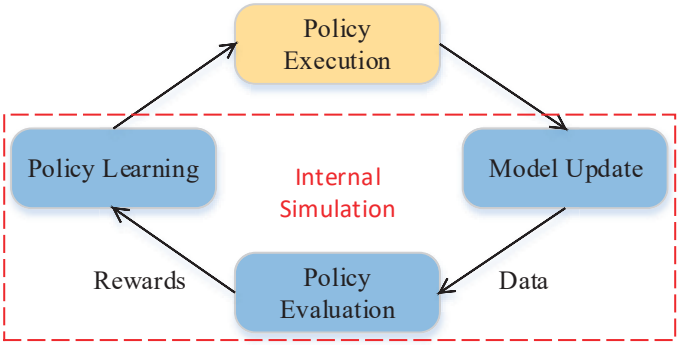

Fig. 2. The framework of model-based RL.

evaluation and improvement were generated by the model of transition dynamics. Also, a neural network was used to learn the dynamics model for quadrotor in [35] as the following formulation:

$$s_{t+1} = s_t + f_\theta(s_t, a_t), \tag{6}$$

where f_θ is a parametric function that maps from the current state and action to the change of the state.

Without interacting with the environment during policy learning, the agent can optimize the policy with less sample data which reduce the sample complexity indirectly. Consider the model of transition probability, the TD-error in (5) can be written as follows:

$$\begin{aligned}\delta_t^{MB} =& r(s_t, a_t) + \gamma \sum_{s_{t+1} \in S} p(s_{t+1}|s_t, a_t) Q_{\hat{\phi}}(s_{t+1}, \mu(s_{t+1})) \\ & - Q_\phi(s_t, a_t).\end{aligned} \tag{7}$$

D. Challenges in RL

Although many algorithms for RL are presented in recent years, and have used in many notable practical applications. There are also some challenges faced both in model-free and model-based RL:

- The optimal policy learned by the RL algorithms are mostly in a single environment, which may sensitive to the perturbation of environment and even lead to some unknown malfunctions in performance with the real complex environments.
- The model of transition probability is estimated from the historical data with some statistical errors generally. However, the model errors are ignored and the policy updated using the learned model as the true transition dynamics.
- Great gaps are lying between the simulation policy and the real environments.

III. Robust Reinforcement Learning

Since the model of transition probability can not be measured accurately with the environment perturbations, the policy learned from the model will be sensitive to the test environment which is not exactly same as the training environment [36]. In order to systematically mitigate the sensitivity of the optimal policy, the robust formulation of RL methods are proposed to learn a robust optimal policy with model uncertainty of the transition dynamics [37], [38]. Define a set of uncertain model as $\mathcal{P}$, then value function of the robust RL problem can be written as follows with the worst-case:

$$V_\pi(s) = \inf_{p \in \mathcal{P}} \mathbb{E}_{\pi, p} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \middle| s_0 = s \right]. \tag{8}$$

The optimality of the decision maker is to choose a policy that maximize the value function above with the robust Bellman equation:

$$V(s_t) = \sup_{a_t \in \pi(s_t)} \inf_{p \in \mathcal{P}} \mathbb{E}_p \left[r(s_t, a_t) + \gamma V(s_{t+1}) \right]. \tag{9}$$

The sets of transition probability uncertainty were described using several models in [24], [39], such as KL-divergence, likelihood, entropy, interval matrix, etc. For finite-state, finite-action robust RL problem, the sets of the uncertain model can be used to reformulate the optimization problem as a tractable counterpart [40]. However, due to the infinity of transition dynamics in continuous state and action space, the uncertain sets of the transition probability model can not be presupposed via the above formulation. In [26], the model function was learned by manually selecting different perturbations in the environment. Then, the TD-error for policy evaluation used in robust RL can be converted to the following equation according to (7):

$$\begin{aligned}\delta_t^{R} =& r(s_t, a_t) + \gamma \inf_{p \in \mathcal{P}} \sum_{s_{t+1} \in S} p(s_{t+1}|s_t, a_t) Q_{\hat{\phi}}(s_{t+1}, \mu(s_{t+1})) \\ & - Q_\phi(s_t, a_t).\end{aligned} \tag{10}$$

IV. Soft-Robust Reinforcement Learning

The main idea of robust RL is to consider the worst-case in the environment with perturbations, we can get a relatively conservative policy by maximizing the expected return with uncertain model of the transition dynamics [41], [42]. Sometimes, the policy searched by robust RL theory is too conservative, the poor performance of the optimal policy occurred even in the nominal environment without perturbation. To address this issue, we need to find a new optimal policy to balance the robust and aggressive characters in the environment perturbation uncertainty. Soft-robust RL is a kind of nonconservative policy optimization methods which is extended from robust RL. An average model was proposed to represent the distribution of uncertainty set $\mathcal{P}$ in [27]. Thus, the update of value function followed the average model instead of the worst-case during policy evaluation. Therefore, the TD-error used in soft-robust RL algorithm is calculated as:

$$\begin{aligned}\delta_t^{SR} =& r(s_t, a_t) + \gamma \sum_{s_{t+1} \in S} \bar{p}(s_{t+1}|s_t, a_t) Q_{\hat{\phi}}(s_{t+1}, \mu(s_{t+1})) \\ & - Q_\phi(s_t, a_t),\end{aligned} \tag{11}$$

where $\bar{p}(s_{t+1}|s_t, a_t)$ denotes the average model of transition probability which is sampled from a Dirichlet distribution of different transition functions generated by the corresponding dynamic model.

V. Current Research and Challenges

Because of the complexity and perturbation in real environments, the optimal policy learned by the agent in a single scenario performs poorly and occurred to failure even slight variations of transition dynamics generally. Especially in practical applications, such as robot control, the unknown environment perturbations are ubiquitous. For example, consider the different mass of the unmanned aerial vehicle during flight control. Then, robustness to changes in transition probability is an important component for RL used in real environments. In addition to the research of robust RL above, we will now examine some other research directions with robust RL which are novel and practical.

A. Wasserstein Robust RL

For high-dimensional continuous control tasks, a robust RL algorithm with Wasserstein distance [43] was proposed in [44], recently. Different from the robust methods for discrete state and action spaces [45], the policy and transition dynamics are parameterized with θ and ϕ, respectively. Then, the learning objective of the Wasserstein robust RL is obtained as follows:

$$\begin{aligned} &\max_{\theta} \left[\min_{\phi} \mathbb{E}_{\tau \sim p_{\theta}^{\phi}(\tau)} \left[R_{total}(\tau) \right] \right] \\ &s.t. \quad \mathbb{E}_{(s,a)\sim\pi(\cdot)\rho_{\pi}^{\phi_0}(\cdot)} \left[\mathcal{W}_2^2(\mathcal{P}_{\phi}(\cdot|s,a), \mathcal{P}_0(\cdot|s,a)) \right] \leq \epsilon, \end{aligned} \tag{12}$$

where τ is a trajectory of state-action pairs, $p_{\theta}^{\phi}(\tau)$ denotes the trajectory density function, $R_{total}(\tau)$ is the return of τ, $\rho_{\pi}^{\phi_0}(\cdot)$ is a uniform distribution over state-action pairs sampled from a trajectory, $\mathcal{P}_0$ denotes the reference dynamics, $\mathcal{W}_2^2(\mathcal{P}_{\phi}(\cdot|s,a), \mathcal{P}_0(\cdot|s,a))$ is the Wasserstein distance between $\mathcal{P}_{\phi}$ and the reference dynamics and ϵ is a hyperparameter which used to specify the degree of robustness.

Then, updating the policy and transition dynamics parameters according to (12). Because of the nonlinear presentation of policy and transition probability, the robust RL algorithm with Wasserstein constraints performs well on unseen dynamical systems.

B. Robust Adversarial RL

Motivated by the gap between simulation and real-world environment, and the model errors in training and test scenarios, a robust adversarial RL method was given in [46]. Two agents were trained at the same time, including a protagonist and an adversary. Then, the protagonist learned a optimal policy that is robust to any disturbances created by the adversary by gaining the rewards of adversary when the protagonist suffer from failure. The adversarial MDP is described as a tuple $\langle S, A_1, A_2, P, R, \gamma, \rho_0 \rangle$, where A_1 and A_2 are the action spaces of protagonist and adversary, respectively. $P : S \times A_1 \times A_2 \times S \to \mathbb{R}$ denotes the transition probability and $R : S \times A_1 \times A_2 \to \mathbb{R}$ is the reward of the two agents. The expected return function of protagonist can be written as:

$$R_1 = \mathbb{E}_{a_1\sim\mu(s), a_2\sim\nu(s)} \left[\sum_{t=0}^{\infty} \gamma^t r_1(s, a_1, a_2) \right], \tag{13}$$

where $\mu(s)$ and $\nu(s)$ are the policies of two agents. Then, the optimal policy of the protagonist can be learned by maximizing the expected return in (13):

$$R_1^* = \max_{\mu} \min_{\nu} R_1(\mu, \nu). \tag{14}$$

C. Distributionally Robust RL

Consider the risk during learning process of the agent, a risk-averse exploration method called distributionally robust soft actor-critic was proposed to prevent poor decisions in [47]. Maximum entropy theory was used to balance the risk-averse and exploration strategy during policy optimization. Due to the distributionally robust Bellman operators, lower bound guarantee on the policy state-values was available. Then, the method performed robust with estimation errors in both discrete and continuous action spaces of RL.

D. Challenges in Robust RL

With the large gaps and model errors of transition dynamics between simulation and real-world settings, Robust RL is an important pipeline to improve the robustness of the optimal policy in uncertain environment perturbations. Various of algorithms have presented to search the robust optimal policy with model uncertainty during learning process. However, some challenges are still existing and need to be resolved urgently.

- In general, the sets of uncertain transition probability are obtained via assumption in advance, which may not represent the perturbed environment accurately.
- The robust policy searched by the robust RL methods is overly conservative sometimes, it is important to keep the balance between the robustness and aggressivity of the optimal policy.
- The goal of robust RL is to let the decision maker more robust with transition dynamics uncertainty in environment perturbations. Then, it is a huge challenge that transfer the simulation environment to real-world robot control without failure occurrence even in unknown complex environments.

VI. Conclusion

In this overview, we mainly discussed the framework of robust RL with model uncertainty of transition dynamics and the relationship with RL. We gave the calculation formulations of policy evaluation for robust and soft-robust RL algorithms. Also, we analyzed other research areas, challenges of robust RL and its applications for robot control. In future work, we plan to extend the robust RL methods to real-world robot control, especially in complex perturbed environments. Besides, we will consider different parameters of the robot as perturbations, such as mass, friction, torque, etc. We believe

that robust RL has tremendous potential for robot control in real complex environments.

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, 2018.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] L. Kaiser, M. Babaeizadeh, P. Milos, B. Osinski, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, A. Mohiuddin, R. Sepassi, G. Tucker, and H. Michalewski, "Model-Based Reinforcement Learning for Atari," mar 2019. [Online]. Available: http://arxiv.org/abs/1903.00374

[4] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with Deep Reinforcement Learning," 2013. [Online]. Available: http://arxiv.org/abs/1312.5602

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, feb 2015. [Online]. Available: http://www.nature.com/articles/nature14236

[6] D. Silver, A. Huang1, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016. [Online]. Available: http://dx.doi.org/10.1038/nature16961

[7] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, 2017.

[8] X. Pan, Y. You, Z. Wang, and C. Lu, "Virtual to real reinforcement learning for autonomous driving," *arXiv preprint arXiv:1704.03952*, 2017.

[9] S. Li, T. Liu, C. Zhang, D. Y. Yeung, and S. Shen, "Learning unmanned aerial vehicle control for autonomous target following," in *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2018-July. International Joint Conferences on Artificial Intelligence Organization, jul 2018, pp. 4936–4942. [Online]. Available: https://www.ijcai.org/proceedings/2018/685

[10] R. Polvara, M. Patacchiola, S. Sharma, J. Wan, A. Manning, R. Sutton, and A. Cangelosi, "Autonomous Quadrotor Landing using Deep Reinforcement Learning," sep 2017. [Online]. Available: http://arxiv.org/abs/1709.03339

[11] M. P. 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 2011*, 2011, pp. 465–472.

[12] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[13] M. L. Puterman, *Markov Decision Processes.: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, 2014.

[14] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[15] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," sep 2015. [Online]. Available: https://goo.gl/J4PIAz http://arxiv.org/abs/1509.02971

[16] J. Schulman, S. Levine, P. Moritz, M. Jordan, and P. Abbeel, "Trust region policy optimization," in *32nd International Conference on Machine Learning, ICML 2015*, vol. 3, 2015, pp. 1889–1897.

[17] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *31st International Conference on Machine Learning, ICML 2014*, vol. 1, 2014, pp. 605–619.

[18] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized Experience Replay," nov 2015. [Online]. Available: http://arxiv.org/abs/1511.05952

[19] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2018, pp. 3215–3222.

[20] A. S. Polydoros and L. Nalpantidis, "Survey of Model-Based Reinforcement Learning: Applications on Robotics," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 86, no. 2, pp. 153–173, may 2017.

[21] N. R. Ke, A. Singh, A. Touati, A. Goyal, Y. Bengio, D. Parikh, and D. Batra, "Learning Dynamics Model in Reinforcement Learning by Incorporating the Long Term Future," mar 2019. [Online]. Available: http://arxiv.org/abs/1903.01599

[22] H. van Hasselt, M. Hessel, and J. Aslanides, "When to use parametric models in reinforcement learning?" jun 2019. [Online]. Available: http://arxiv.org/abs/1906.05243

[23] J. Morimoto and K. Doya, "Robust reinforcement learning," *Neural Computation*, vol. 17, no. 2, pp. 335–359, 2005. [Online]. Available: https://www.mitpressjournals.org/doi/abs/10.1162/0899766053011528

[24] A. Nilim and L. E. Ghaoui, "Robust control of Markov decision processes with uncertain transition matrices," *Operations Research*, vol. 53, no. 5, pp. 780–798, 2005.

[25] G. N. Iyengar, "Robust dynamic programming," *Mathematics of Operations Research*, vol. 30, no. 2, pp. 257–280, 2005.

[26] D. J. Mankowitz, N. Levine, R. Jeong, A. Abdolmaleki, J. T. Springenberg, T. Mann, T. Hester, and M. Riedmiller, "Robust Reinforcement Learning for Continuous Control with Model Misspecification," jun 2019. [Online]. Available: http://arxiv.org/abs/1906.07516

[27] E. Derman, D. J. Mankowitz, T. A. Mann, and S. Mannor, "Soft-robust actor-critic policy-gradient," *arXiv preprint arXiv:1803.04848*, 2018.

[28] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[29] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[30] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. d. L. Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, T. Lillicrap, and M. Riedmiller, "DeepMind Control Suite," jan 2018. [Online]. Available: http://arxiv.org/abs/1801.00690

[31] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner, "Torcs, the open racing car simulator," *Software available at http://torcs. sourceforge. net*, vol. 4, no. 6, 2000.

[32] A. Y. Ng, H. J. Kim, M. I. Jordan, and S. Sastry, "Autonomous helicopter flight via reinforcement learning," in *Advances in Neural Information Processing Systems*, 2004, pp. 799–806.

[33] A. Y. Ng, A. Coates, M. Diel, V. Ganapathi, J. Schulte, B. Tse, E. Berger, and E. Liang, "Autonomous inverted helicopter flight via reinforcement earning," *Springer Tracts in Advanced Robotics*, vol. 21, pp. 363–372, 2006.

[34] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," in *Advances in Neural Information Processing Systems*, 2007, pp. 1–8. [Online]. Available: http://www.cs.stanford.edu/

[35] N. O. Lambert, D. S. Drew, J. Yaconelli, S. Levine, R. Calandra, and K. S. J. Pister, "Low-Level Control of a Quadrotor With Deep Model-Based Reinforcement Learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4224–4230, 2019. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8769882/

[36] T. Osogami, "Robustness and risk-sensitivity in markov decision processes," in *Advances in Neural Information Processing Systems*, 2012, pp. 233–241.

[37] S. Mannor, O. Mebel, and H. Xu, "Robust mdps with k-rectangular uncertainty," *Mathematics of Operations Research*, vol. 41, no. 4, pp. 1484–1509, 2016.

[38] S. H. Lim, H. Xu, and S. Mannor, "Reinforcement learning in robust markov decision processes," in *Advances in Neural Information Processing Systems*, 2013, pp. 701–709.

[39] A. Nilim and L. El Ghaoui, "Robustness in markov decision problems with uncertain transition matrices," in *Advances in Neural Information Processing Systems*, 2004, pp. 839–846.

[40] A. Ben-Tal, D. Den Hertog, A. De Waegenaere, B. Melenberg, and G. Rennen, "Robust solutions of optimization problems affected by uncertain probabilities," *Management Science*, vol. 59, no. 2, pp. 341–357, 2013.

[41] E. Delage and S. Mannor, "Percentile optimization for markov decision processes with parameter uncertainty," *Operations research*, vol. 58, no. 1, pp. 203–213, 2010.
[42] H. Xu and S. Mannor, "The robustness-performance tradeoff in markov decision processes," in *Advances in Neural Information Processing Systems*, 2007, pp. 1537–1544.
[43] L. Rüschendorf, "The wasserstein distance and approximation theorems," *Probability Theory and Related Fields*, vol. 70, no. 1, pp. 117–129, 1985.
[44] M. A. Abdullah, H. Ren, H. B. Ammar, V. Milenkovic, R. Luo, M. Zhang, and J. Wang, "Wasserstein robust reinforcement learning," *arXiv preprint arXiv:1907.13196*, 2019.
[45] L. Hansen and T. J. Sargent, "Robust control and model uncertainty," *American Economic Review*, vol. 91, no. 2, pp. 60–66, 2001.
[46] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 2817–2826.
[47] E. Smirnova, E. Dohmatob, and J. Mary, "Distributionally robust reinforcement learning," *arXiv preprint arXiv:1902.08708*, 2019.