Petrik, Vladimir; Kyrki, Ville

Influence of Recent History to System Controllability

Published: 08/12/2018

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Petrik, V., & Kyrki, V. (2018). *Influence of Recent History to System Controllability*. Paper presented at Conference on Neural Information Processing Systems, Montréal, Canada.

Influence of Recent History to System Controllability

Vladimír Petrík
Aalto University, Finland.
vladimir.petrik@aalto.fi

Ville Kyrki
Aalto University, Finland.
ville.kyrki@aalto.fi

1 Context and Problem

Reinforcement learning (RL) for physical systems such as robots has gained great interest lately. Data cost for physical systems is usually high due to required time, equipment cost, and safety limitations. For that reason, the learning is often performed with simulation models. However, the quality of simulations is limited for two reasons: First, a simulation model may capture only some of the physical phenomena present. Second, the model may be imperfectly calibrated. In that case, the simulation depends on parameters that may be directly unobservable in the physical system. Control of the system can then be considered a partially observable Markov decision process (POMDP) with respect to those parameters.

Problem Formulation We consider a nonlinear dynamic system described by the difference equation $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\theta}) + \boldsymbol{w}$, where $\boldsymbol{x}_t$ is system state at time t, $\boldsymbol{u}$ is action, $\boldsymbol{\theta}$ is set of parameters and $\boldsymbol{w}$ is Gaussian noise. We assume that the system state is observable, dynamics is Markovian and parametrized by parameters which are not directly observable in the physical world. However, the set of parameters is observable (and controllable) in the simulation during training. The stochastic dynamics is therefore described by $P_{xu\theta} \equiv p(\boldsymbol{x}_{t+1} \mid \boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{\theta})$, that is, the dynamics is Markovian if the parameters are observable.

Considering that the dynamics results from simulation, it is not expressed in closed form and it is not analytically differentiable such that optimal control approaches could be directly used. For the same reason, the straightforward approach of constructing an estimator (such as a Kalman filter variant) for the parameters is not applicable. Therefore, we use RL for training a feedback-based control policy.

Reinforcement Learning If the parameters are observable we can use RL [7] to train a deterministic feedback-based policy $\boldsymbol{u} = \pi(\boldsymbol{x}, \boldsymbol{\theta})$, such that the policy maximizes the expected reward $R(\boldsymbol{\theta}) = E_{\boldsymbol{x}_{t+1} \sim P_{xu\theta}, \boldsymbol{x}_0 \sim p(\boldsymbol{x}_0)} \left[\sum_{t=0}^{\infty} r(\boldsymbol{x}_{t+1}) \mid \boldsymbol{\theta}\right]$, where $r(\cdot)$ is immediate reward at the state $\boldsymbol{x}_t$, and $\boldsymbol{x}_0$ is start state sampled from a prior distribution $p(\boldsymbol{x}_0)$. However, for unobservable parameters we need to provide a policy which is independent of $\boldsymbol{\theta}$.

Domain Randomization Recently, the problem of unobservable parameters has received attention from the point-of-view of domain randomization (DR) [9, 3]. In DR, the policy is optimized over a distribution of parameters $p(\boldsymbol{\theta})$. The expected reward can be written as $R_{\text{DR}} = E_{\boldsymbol{\theta} \sim p(\boldsymbol{\theta})} \left[R(\boldsymbol{\theta})\right]$. Thus, the expectation is taken over the parameters in addition to the stochastic dynamics and start states. We can use RL to train policy $\boldsymbol{u}_t = \pi_{DR}(\boldsymbol{x}_t)$, which maximizes R_{DR} and is independent of the parameters. However, domain randomization is usually limited in its ability to cope with large variation of parameters in case of non-informative prior distribution.

Using Recent History The parameters could often be inferred from past observations either using classical filtering techniques [6] if closed form dynamics is known or by trained estimators [4] otherwise. However, training such an estimator is non-trivial. Therefore, we study the influence of the recent history of states and actions [8] which is sufficient for many physical systems. Considering last n states and actions, the policy is in form $\boldsymbol{u}_t = \pi_H(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \boldsymbol{u}_{t-1}, \ldots, \boldsymbol{x}_{t-n}, \boldsymbol{u}_{t-n})$ and is trained to maximize R_{DR}.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

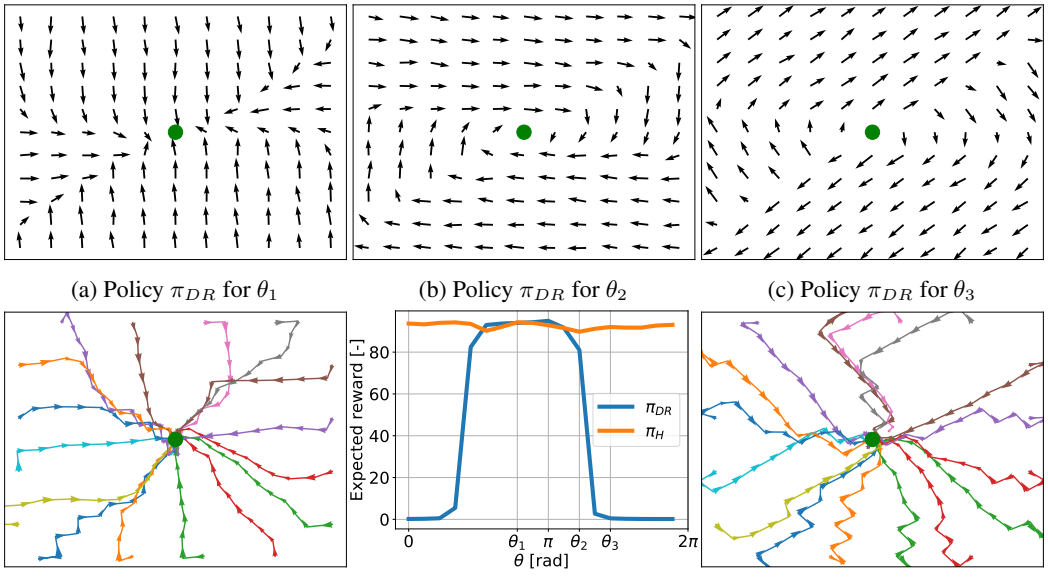
2 Experiment and Discussion

To illustrate the influence of recent history, we consider a task consisting of a holonomic 2D mobile robot with unknown but fixed orientation θ. The action is specified with respect to the robot frame and we want to achieve a goal in the origin of the world coordinate frame. Therefore, the system dynamics can be written $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + R(\theta)\boldsymbol{u}_t + \boldsymbol{w}$, where $R(\theta)$ is a rotation matrix. Results in Fig. 1 illustrate that the policy π_{DR} cannot provide a control strategy for whole range of θ because it cannot infer the robot orientation from the current state only. However, the policy π_H which considers the previous state and action can generalize over whole range of θ.

Discussion Domain randomization is not sufficient for coping with parameter uncertainties in physical systems such as robotic pushing with an unknown center of friction. In contrast, extending the state with a limited history can make the system controllable [2] even without explicit parameter estimation. Thus, existing RL approaches are sufficient to solve problems whose parameters are not directly observable.

In this brief note we only consider the case where observability is global, that is, parameters can be identified in any part of the state space. This is not the case in general; a parameter may be observable only in a part of the state space. Under local observability, memory would be needed to track the parameter, e.g. using recurrent neural network [1, 10, 5]. Memory or long history would also be beneficial if there is significant noise in the dynamics.

However, for many physical systems using the recent history of states and actions [8] may also solve the problem with an advantage of simplified policy representation and simpler training. Limited history being sufficient stems from the fact that many physical systems follow dynamics described by differential equations of low order. What are the system restrictions and how long the history should be for the system to be controllable are subjects of our future work.



(a) Policy π_{DR} for θ_1 (b) Policy π_{DR} for θ_2 (c) Policy π_{DR} for θ_3

(d) Policy π_H for θ_1 (e) Expected rewards (f) Policy π_H for θ_3

Figure 1: Top row visualizes the policy π_{DR} by showing changes of system states (quivers). The system converges towards the goal (green dot) for some values of theta (a), (b). However, it is not able to generalize across all values which result in divergence from goal (c). The system converges slowly for parameter θ_2 which results in the lower expected reward for θ_2 (e). With the policy π_H, the system converges to goal for all parameter values (e) as shown by the states evolutions in (d) and (f). Note, that first action generated by π_H may not lead toward the goal because the parameter value is unobservable. After the first action is taken, the parameter value is inferred and the system converges (see Appendix A).

Acknowledgments

This work was supported by Academy of Finland, decision 317020.

References

[1] Bram Bakker. Reinforcement learning with long short-term memory. In *Advances in neural information processing systems*, pages 1475–1482, 2002.

[2] Roland Burns. *Advanced control engineering*. Elsevier, 2001.

[3] Xi Chen, Ali Ghadirzadeh, John Folkesson, and Patric Jensfelt. Deep reinforcement learning to acquire navigation skills for wheel-legged robots in complex environments. *arXiv preprint arXiv:1804.10500*, 2018.

[4] Tuomas Haarnoja, Anurag Ajay, Sergey Levine, and Pieter Abbeel. Backprop KF: learning discriminative deterministic state estimators. *CoRR*, abs/1605.07148, 2016.

[5] Matthew Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. *CoRR, abs/1507.06527*, 7(1), 2015.

[6] Simon J Julier and Jeffrey K Uhlmann. New extension of the kalman filter to nonlinear systems. In *Signal processing, sensor fusion, and target recognition VI*, volume 3068, pages 182–194. International Society for Optics and Photonics, 1997.

[7] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[8] Long-Ji Lin and Tom M Mitchell. *Memory approaches to reinforcement learning in non-Markovian domains*. Citeseer, 1992.

[9] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. *arXiv preprint arXiv:1710.06537*, 2017.

[10] Daan Wierstra, Alexander Foerster, Jan Peters, and Juergen Schmidhuber. Solving deep memory pomdps with recurrent policy gradients. In *International Conference on Artificial Neural Networks*, pages 697–706. Springer, 2007.

A Initial Action for Policy with History

Initially, the policy π_H does not have enough information to infer the unobservable parameters because history is not yet available. Therefore, the policy needs to take some action under partial observability. The actions taken by the policy at various starting positions are shown in Fig. 2. The policy takes non-zero action only if the system state is not near the goal. Therefore, the policy is not estimating the parameters if that is not necessary for reaching the goal. This shows that knowing the parameters does not have value as such, and the value is created by the ability to achieve the task, analogous to POMDP solutions.

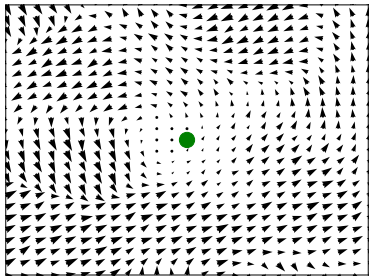


Figure 2: The initial change of system states when starting from different start states.