Online BayesSim for Combined Simulator Parameter Inference and Policy Improvement

Author Names Omitted for Anonymous Review

Abstract— **In this paper, we study the integration of simulation parameter inference with both model-free reinforcement learning and model-based control in a novel sequential algorithm that alternates between learning a better estimation of parameters and improving the controller. Experimental results suggest that both control strategies have better performance when compared to traditional domain randomization methods.**

I. INTRODUCTION

Advancements in simulation have allowed robotics learning to become more efficient and realistic in recent years [1][2][3]. However, there is still a range of possible improvements in simulation techniques before they can capture reality with all its complexities. "Reality gap" is a term used when the environment model used in a simulator does not represent the targeted system accurately enough so we can achieve the desirable performance when deploying a robot in the real world.

It is known that oversimplified assumptions or insufficient numerical precision in solvers can play a major role in how well a simulator models its target desired system. Existing prior knowledge about simulation parameters is often incorporated through a series of trial and error experiments until a good approximation is reached. This process is inefficient and time consuming as it involves running non-optimal control strategies on expensive and fragile robots.

In this work, we build upon the idea of using probabilistic inference to learn distributions over simulation parameters [3]. This technique leverages recent advances in likelihood-free inference (LFI) [4] for Bayesian analysis to learn posteriors over simulation parameters based on rollouts obtained from the target system. Previous work [3] managed to learn distributions over parameters, but it required a reasonable initial controller that was able to explore the dynamical system in relevant regions of the state-space. Alternatively, in this paper we propose an end-to-end approach that combines posterior updates with controller improvement.

II. ONLINE BAYESSIM

Here we present the main contribution of the paper: Online BayesSim. We leverage previous work in likelihood-free inference to simultaneously improve a controller and learn a distribution over the simulator parameters. Additionally, we propose a methodology to automate the computation of a low-dimensional representation of state-action trajectories using Recurrent Neural Networks (RNN). The difficulty in representing high-dimensional time series has been one of the major reasons why LFI methods do not scale well to higher dimensional spaces. We show that with an RNN, latent representations from entire trajectories can be learnt and used directly for the posterior estimation. This removes the need to manually define meaningful summary statistics, which sometimes, can be a quite difficult and complex task.

The use of Bayesian inference can be borrowed from more traditional statistics methods such as approximate Bayesian computation (ABC) [5]. Improvements over this method such as Rejection ABC [6], Markov Chain Monte Carlo ABC (MCMC-ABC) [7], Sequential Monte Carlo ABC (SMC-ABC) [8] and finally the ϵ-free approach [4] have enabled Bayesian inference on a wide range of problems.

Formally, we start with a stochastic controller $\pi_\beta(\boldsymbol{a_t}|\boldsymbol{s_t})$ and no prior knowledge of the true parameters represented by an uniform prior $p(\boldsymbol{\theta})$. In the first iteration $\pi_\beta(\boldsymbol{a_t}|\boldsymbol{s_t})$ is initialised with samples from the uniform prior $p(\boldsymbol{\theta})$. Trajectories $\mathbf{S}^s, \mathbf{A}^s$ are collected using current $\pi_\beta(\boldsymbol{a_t}|\boldsymbol{s_t})$ which are then used to update our Mixture of Gaussians model $q_\phi(\boldsymbol{\theta}|\mathbf{z})$. New data $\mathbf{S}^r, \mathbf{A}^r$ is then collected in the target system (e.g. real environment, proxy simulator and etc) using the same controller which is subsequently used to recover a new posterior and update the control strategy. $p(\boldsymbol{\theta}|\mathbf{S}, \mathbf{A} = \mathbf{S}^r, \mathbf{A}^r)$. The prior $p(\boldsymbol{\theta})$ is then replaced by the new posterior and the algorithm iterates until we achieve the desired controller performance. Details can be seen in Algorithm 1.

III. RESULTS

A. Classic Control Tasks

Online BayesSim have been evaluated on several control tasks as shown on table I. We have compared the log-likelihood of the posteriors recovered by our algorithm against recent work in LFI. It can be seen that Online BayesSim has outperformed current work in most of the tasks. This shows that online learning coupled with iterative updates can result in sharper posteriors.

B. Experiments on a physical robot

This section presents experimental results with a physical robot equipped with a skid-steering drive mechanism (Figure 1). We modelled the kinematics of the robot based on a modified unicycle model, which accounts for skidding via an additional parameter [9]. The parameters to be estimated via Online BayesSim are the robot's wheel radius, axial distance, i.e. the distance between the wheels, and the displacement of the robot's instant centre of rotation (ICR) from the robot's centre. We have used DISCO [10] as the robot's controller, a stochastic non-linear MPC based on MPPI [11]

Problem	Parameter	Prior	Online BayesSim	BayesSim RFF	ϵ-Free
CartPole	Length / Mass	$[0.1, 1.0]$	**4.66±0.22**	2.68±0.08	2.88±0.15
Pendulum	Length / Mass	$[0.1, 1.0]$	**4.076±0.12**	3.89±0.34	3.332±0.41
Fetch Push	Friction	$[0.1, 2.0]$	2.09±0.12	**2.18±0.19**	2.10±0.27
Fetch Slide	Friction	$[0.1, 2.0]$	**3.24±0.21**	3.12±0.08	2.55±0.26
Hopper	Lat. Friction	$[0.1, 0.5]$	3.25±0.33	3.134±0.11	**3.384±0.25**
Acrobot	Link Mass 1 & 2	$[0.5, 2.0]$	**2.85±0.12**	1.534±0.22	1.210±0.32
	Link Length 1 & 2	$[0.1, 1.5]$	**2.25±0.13**	1.426±0.11	1.012±0.21

TABLE I: Mean and standard deviation of log-likelihood of the joint distribution for offline and online likelihood-free methods, applied to different problems and combination of parameters

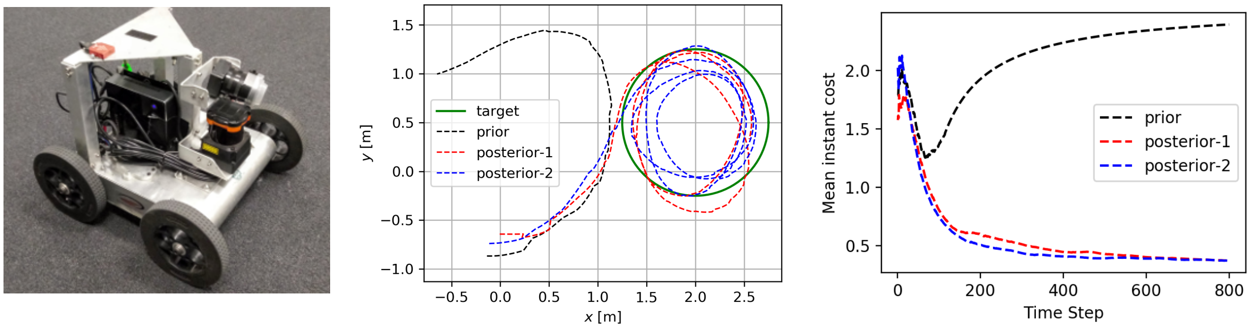

Fig. 1: (Left) Skid-steer Robot. (Center) As posteriors are refined the controller has fewer overshoots on the circular trajectory. (Right) Cumulative average of the cost over time.

Algorithm 1 Online BayesSim

```
//observed and real trajectories: $\mathbf{S}^s, \mathbf{A}^s$ and $\mathbf{S}^r, \mathbf{A}^r$
//RNN Mixture of Gaussians (MoG) estimator: $q_\phi(\boldsymbol{\theta}|\mathbf{z})$
//RL Policy: $\pi_\beta(\boldsymbol{a_t}|\boldsymbol{s_t})$
**Inputs**: total_steps, policy_train_steps, mog_train_steps, num_sampled_params, $p(\boldsymbol{\theta}_0)$
**Outputs**: $q_\phi(\boldsymbol{\theta}|\mathbf{z}), \pi_\beta(\boldsymbol{a_t}|\boldsymbol{s_t})$

Initialize weights $\boldsymbol{\beta}$ and $\boldsymbol{\phi}$ randomly

$t \leftarrow 1$
**repeat**
    $\boldsymbol{\theta}_t \sim p(\boldsymbol{\theta}_{t-1})$
    $\mathbf{S}^s, \mathbf{A}^s \leftarrow$ Run $\pi_{\beta_t}(\boldsymbol{a_t}|\boldsymbol{s_t})$ in sim with $\boldsymbol{\theta}_t$.
    $\boldsymbol{\beta}_t \leftarrow \boldsymbol{\beta}_{t-1} + \lambda \nabla \pi_{\beta_t}(\boldsymbol{a_t}|\boldsymbol{s_t})$
    $\phi_t \leftarrow \phi_{t-1} + \lambda \nabla q_\phi(\boldsymbol{\theta}_t|\psi_{\gamma_t}(\mathbf{S}^s, \mathbf{A}^s))$
    $\mathbf{S}^r, \mathbf{A}^r \leftarrow$ Run $\pi_{\beta_t}(\boldsymbol{a_t}|\boldsymbol{s_t})$ on real env.
    $p(\boldsymbol{\theta}_t|\mathbf{S}, \mathbf{A}) \leftarrow q_\phi(\boldsymbol{\theta}_t|\psi_{\gamma_t}(\mathbf{S}^r, \mathbf{A}^r))$
    $p(\boldsymbol{\theta}_t) \leftarrow p(\boldsymbol{\theta}_t|\mathbf{S}, \mathbf{A})$
    $t \leftarrow t + 1$
**until** $t <$ total_steps or convergence reached
```

The results presented in Figure 1 and Figure 2 show the qualitative and quantitative improvement in the control task as the posterior distribution is refined. As expected, once the robot is able to collect data from the real environment and refine its knowledge of the world, the results improve significantly.

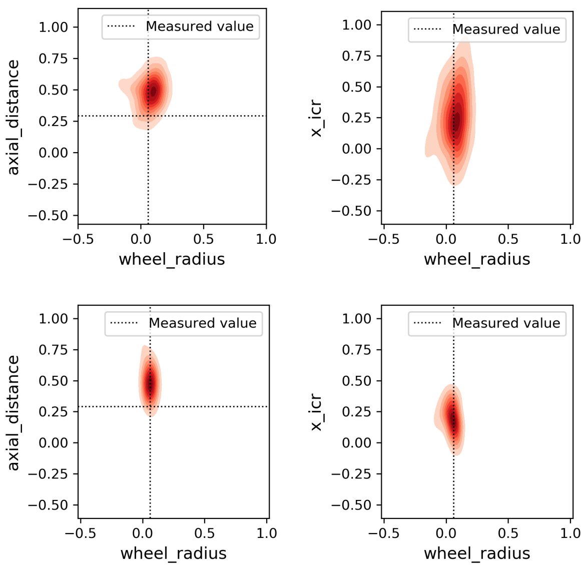

Fig. 2: Posterior distribution for the *second* iteration of online BayesSim on the experiments with a physical robot. Available measured values are indicated by a dashed line.

IV. Discussion

This paper presented a principled framework for solving the ”reality gap” problem in robotics simulators, combining parameter estimation with policy improvement. The approach is capable of leveraging these two problems within a single framework, where sequential improvements in controller performance are used to estimate better simulation parameters and the associated uncertainty.

References

[1] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, “Sim-to-real transfer of robotic control with dynamics randomization,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[2] Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox, “Closing the sim-to-real loop: Adapting simulation randomization with real world experience,” *arXiv preprint arXiv:1810.05687*, 2018.

[3] F. Ramos, R. C. Possas, and D. Fox, “Bayessim: adaptive domain randomization via probabilistic inference for robotics simulators,” *arXiv preprint arXiv:1906.01728*, 2019.

[4] G. Papamakarios and I. Murray, “Fast ε-free inference of simulation models with bayesian conditional density estimation,” in *Advances in Neural Information Processing Systems*, 2016, pp. 1028–1036.

[5] M. A. Beaumont, W. Zhang, and D. J. Balding, “Approximate bayesian computation in population genetics,” *Genetics*, vol. 4, no. 162, pp. 2025–2035, 2002.

[6] J. K. Pritchard, M. T. Seielstad, A. Perez-Lezaun, and M. W. Feldman, “Population growth of human y chromosomes: a study of y chromosome microsatellites.” *Molecular biology and evolution*, vol. 16, no. 12, pp. 1791–1798, 1999.

[7] P. Marjoram, J. Molitor, V. Plagnol, and S. Tavaré, “Markov chain Monte Carlo without likelihoods,” *Proceedings of the National Academy of Sciences*, vol. 100, no. 26, pp. 15 324–15 328, 2003.

[8] F. V. Bonassi, M. West, *et al.*, “Sequential Monte Carlo with adaptive weights for approximate bayesian computation,” *Bayesian Analysis*, vol. 10, no. 1, pp. 171–187, 2015.

[9] K. Kozłowski and D. Pazderski, “Modeling and Control of a 4-wheel Skid-steering Mobile Robot,” *Int. J. Appl. Math. Comput. Sci.*, vol. 14, no. 4, pp. 477–496, 2004.

[10] L. Barcelos, R. Oliveira, R. Possas, L. Ott, and F. Ramos, “DISCO: Double Likelihood-free Inference Stochastic Control.” [Online]. Available: http://arxiv.org/abs/2002.07379

[11] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, “Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving,” vol. 34, no. 6, pp. 1603–1622.