Robust Sim2Real Transfer by Learning Inverse Dynamics of Simulated Systems

Mohammadhossein Malmir[1]
hossein.malmir@tum.de

Josip Josifovski[1]
josip.josifovski@tum.de

Noah Klarmann[1]
noah.klarmann@tum.de

Alois Knoll[1]
knoll@mytum.de

***Abstract*—This paper presents a data-driven nonlinear disturbance observer to reduce the reality gap caused by the imperfect simulation of the real-world physics. The main focus is on increasing robustness of the closed-loop control without changing the RL algorithm or simulation model to account for the uncertainty of the real world. For this purpose, a DNN representing inverse dynamics of the deterministic source-domain environment is learned by the simulation data. The proposed approach offers a systematic way to transfer the policies trained in simulation into the real world without decreasing sample efficiency of the RL agent in contrast to domain randomization or min-max robust RL methods.**

***Index Terms*—inverse dynamics, disturbance observer, robotic manipulation, robust reinforcement learning, sim2real transfer**

I. Introduction

Directly training the RL agent on the real robots [1]–[3] has shown only few successes for merely learning simple tasks [4] due to the high sample complexity of the state-of-the-art RL algorithms [4]–[6]. A common approach to overcome the aforementioned problem is to perform learning in a simulated environment that mimics the real world and to transfer the trained policies to the physical robot afterwards [4], [7]–[16]. However, this is a challenging task since the conventional RL algorithms usually assume the same environment both for the training and the test phases [17], [18], which makes them unable to generalize across slightly varied dynamics of the environment [5], [18], and consequently fail to keep their performance when transferred to the real world [19], [20] due to the existing *reality gap* [8], [9], [12], [21].

Increasing simulation accuracy in terms of the simulated physics via accurate system identification [4], [9], [22]–[24], and the simulated perception via realistic rendering [25] is the first step toward reducing the reality gap. Furthermore, continuing the learning process in the real world lets the RL agent adapt its behavior to the new uncertain situations that it has not been previously trained for [16], [26], and it is reflected in the contexts of transfer learning [27]–[29], progressive neural networks [16], domain adaptation [30]–[33] or action adaptation [15]. Finally, improving robustness of the trained optimal policy by adding intentional uncertainties in simulation, like randomizing the impacts of actions on the environment via dynamics randomization [4], [9], [34]–[36] or randomizing the visual observations of the environment via domain randomization [8], [26], [37], helps in finding transferable policies without any real-world data. As an implication of adding uncertainties, generalization of the learned control policy is enhanced as the algorithm needs to perform well on a wider range of possible dynamics or perception of the environment. Hence, the real-world performance is improved without calling for continuation of the training on the physical system [4], [15].

II. Background

Based on the ideas of H_∞ optimal control [38], previous works in [6], [9] considered the mismatch between the source domain (e.g., the simulated environment) and the target domain (e.g., the real world) as extra disturbances added to the actions of the agent. For example, in the case of a torque-controlled robotic arm, these additive disturbances are of the type of forces or torques exerted on the joints or links of the robot or the end effector. The effect of adding these disturbance forces is similar to including uncertainty in modeling the correct dynamics of the robot (e.g., links' mass, inertia and joints friction, damping or backlash) as well as the correct parameters of the objects manipulated by the robot. In order to estimate the additive disturbances (i.e., the mismatch between the domains), a nonlinear disturbance observer is used in this work.

The objective of a disturbance observer is to incorporate an inner-feedback loop that uses the inverse of the nominal model (i.e., the deterministic simulated environment in the source domain) in order to adapt the system inputs in a way that the overall robustness of the control loop is increased [39]. Particularly, the controller becomes able to maintain its nominal performance even when external disturbances exist or the dynamics of the system are uncertain. The advantage of this approach is on its hierarchical way to solve the problem in a sense that the disturbance observer can be easily integrated with any generic controller [40] or any RL algorithm for training the agent, which is not the case for the adversarial robust RL methods (e.g., [5], [6], [41]). Recently, [42] showed how a disturbance observer can increase robustness of RL-based controllers for a partially-observable uncertain system. However, they have focused on obtaining the necessary conditions that prove sub-optimality of the control performance by assuming to know the nominal dynamics of the environment,

[1] Chair of Robotics, Artificial Intelligence and Real-time Systems, Department of Informatics, Technical University of Munich, Munich, Germany

This work has been financially supported by the ECSEL Joint Undertaking under the H2020 AI4DI project (grant agreement 826060).

Supplementary video is available at: https://youtu.be/LyqMkguLDe8 which limits the applicability of their approach on many of the real-world RL problems. In this work, a data-driven nonlinear disturbance observer is designed to increase robustness of the closed-loop controlled system and thus effectively transfer the trained policy from the simulation to the real world.

III. Method

The core idea is to reduce the reality gap by manipulating the actions imposed on the real robot in a way that the real-world environment behaves similarly to the simulated one from the input-output (actions-observations) perspective [42]. By adapting the pre-trained actions, the agent is expected to follow the optimal policy learned in simulation without the need to continue training in the real world. To achieve this goal, the disturbance observer only needs the inverted dynamics of the simulation model and not the one of the real system, which is much simpler to be realized. This is a significant advantage compared to the work in [15] where the inverse dynamics model of the real-world environment needs to be identified.

The inverse simulation model can be represented by a nonlinear dynamic system, which calculates what was the action imposed to the simulated environment (u_{k-L}) from the next observations received ($Y_{[k,L]} = [y_k, y_{k-1}, \ldots, y_{k-L}]^T$) where k is the current time step and L is the inherent delay of the system [43].

By availing the inverse dynamics model, the disturbance observer is able to find an estimated value for the disturbance ($\hat{d}_k$), which accounts for the mismatch between the source and target domains. Fig. 1 shows how the disturbance observer works in closed loop with the uncertain system of the real world and alters the optimal action u_k by rejecting the underlying disturbances to increase robustness. Accordingly, the closed-loop dynamics of the disturbance observer with the target environment becomes approximately equal to the dynamics of the source environment. This statement is shown in Fig. 1 and justifies the robustness of the approach, however, the extent of how much the approximation $\hat{d}_k \approx d_k$ remains valid should be investigated.

Training of the inverse dynamics model can occur at the very moment when the RL agent is learning the optimal policy in closed loop with the simulated environment and is shown in Fig. 2. In order to learn the inverse dynamics of the simulated environment, a feedforward neural network needs to be trained in supervised fashion by the simulation data where the input-output pairs are the simulated observations and actions.

IV. Experiments

The efficacy of the proposed method in reducing the reality gap can be evaluated by performing several experiments. These experiments should investigate the robust operation of the overall controller when it is transferred from the source domain to the target domain. Towards this end, two kinds of experiments will be taken, namely *sim2sim* and *sim2real*. In both kinds, the source domain is the nominal simulation environment where the agent has been trained. The target domain for a sim2sim experiment is a perturbed simulation

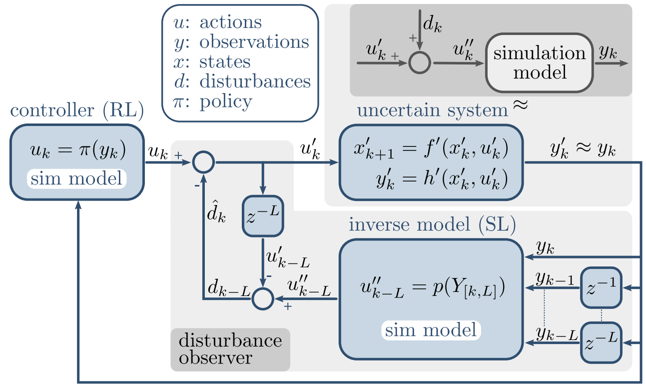

Fig. 1: Disturbance observer employs the trained inverse simulation model to achieve robustness in the target domain.

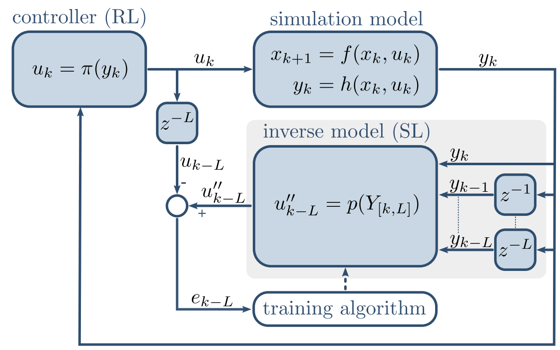

Fig. 2: An illustrative case of how the inverse simulation model could be trained in the source domain.

environment while for the sim2real experiment is the real world where finally the agent is deployed. A systematic empirical validation will be conducted to assess the performance of the proposed approach, in contrast to the related state-of-the-art methods, in terms of the increased success rate [35], expected return [9], [44], and gained robustness bounds on the parameters uncertainty.

V. Conclusion

It should be noted that the primary focus of the work is to reduce the reality gap that is caused by the imperfect simulation of the real-world physics, and not the gap caused by how the real world is perceived differently in comparison to the simulated environment. Nonetheless, the proposed idea is general enough to be combined with the existing methods on reducing the gap in perception, like domain randomization (e.g., [8]) or domain adaptation (e.g., [31]).

References

[1] S. Levine, N. Wagener, and P. Abbeel, “Learning contact-rich manipulation skills with guided policy search,” *S. Levine*,

[2] S. Levine, P. Pastor, A. Krizhevsky, and D. Quillen, *Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection*.

[3] S. Levine, C. Finn, T. Darrell, and P. Abbeel, *End-to-end training of deep visuomotor policies*.

[4] 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)*, K. Lynch and I. I. C. o. R. a. Automation, Eds., [Piscataway, NJ]: IEEE, 2018, pp. 3803–3810, ISBN: 978-1-5386-3081-5. DOI: 10.1109/ICRA.2018.8460528.

[5] M. A. Abdullah, H. Ren, H. B. Ammar, V. Milenkovic, R. Luo, M. Zhang, and J. Wang, *Wasserstein robust reinforcement learning*.

[6] L. Pinto, J. Davidson, R. Sukthankar, and A. Gupta, "Robust adversarial reinforcement learning," *International Conference on Machine Learning*, pp. 2817–2826, 2017, ISSN: 1938-7228.

[7] J. van Baar, A. Sullivan, R. Cordorel, D. Jha, D. Romeres, and D. Nikovski, "Sim-to-real transfer learning using robustified controllers in robotic tasks involving complex dynamics," in *2019 International Conference on Robotics and Automation (ICRA)*, [Piscataway, NJ]: IEEE, 2019, pp. 6001–6007, ISBN: 978-1-5386-6027-0. DOI: 10.1109/ICRA.2019.8793561.

[8] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, *Domain randomization for transferring deep neural networks from simulation to the real world*.

[9] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, *Sim-to-real: Learning agile locomotion for quadruped robots*.

[10] A. Molchanov, T. Chen, W. Hönig, J. A. Preiss, N. Ayanian, and G. S. Sukhatme, *Sim-to-(multi)-real: Transfer of low-level robust control policies to multiple quadrotors*.

[11] M. Kaspar, J. D. M. Osorio, and J. Bock, *Sim2real transfer for reinforcement learning without dynamics randomization*.

[12] R. Julian, E. Heiden, Z. He, H. Zhang, S. Schaal, J. J. Lim, G. Sukhatme, and K. Hausman, *Scaling simulation-to-real transfer by learning composable robot skills*.

[13] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis, "Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 15-Jun-19 - 20-Jun-19, pp. 12 619–12 629, ISBN: 978-1-7281-3293-8. DOI: 10.1109/CVPR.2019.01291.

[14] F. Golemo, A. A. Taiga, A. Courville, and P.-Y. Oudeyer, "Sim-to-real transfer with neural-augmented robot simulation," *Conference on Robot Learning*, pp. 817–828, 2018, ISSN: 1938-7228.

[15] P. Christiano, Z. Shah, I. Mordatch, J. Schneider, T. Blackwell, J. Tobin, P. Abbeel, and W. Zaremba, *Transfer from simulation to real world through learning deep inverse dynamics model*.

[16] A. A. Rusu, M. Večerík, T. Rothörl, N. Heess, R. Pascanu, and R. Hadsell, "Sim-to-real robot learning from pixels with progressive nets," *Conference on Robot Learning*, pp. 262–270, 2017, ISSN: 1938-7228.

[17] C. Tessler, Y. Efroni, and S. Mannor, *Action robust reinforcement learning and applications in continuous control*.

[18] C. Packer, K. Gao, J. Kos, P. Krähenbühl, V. Koltun, and D. Song, *Assessing generalization in deep reinforcement learning*.

[19] 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*.

[20] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. M. A. Eslami, M. Riedmiller, and D. Silver, *Emergence of locomotion behaviours in rich environments*.

[21] J.-B. Mouret and K. Chatzilygeroudis, "20 years of reality gap: A few thoughts about simulators in evolutionary robotics," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, New York, NY, USA: ACM, 7152017, pp. 1121–1124, ISBN: 9781450349390. DOI: 10.1145/3067695.3082052.

[22] M. NEUNERT, T. BOAVENTURA, and J. BUCHLI, "Why off-the-shelf physics simulators fail in evaluating feedback controller performance - a case study for quadrupedal robots," in *Advances in Cooperative Robotics*, M. O. Tokhi and G. S. Virk, Eds., New Jersey: WORLD SCIENTIFIC, 2017, pp. 464–472, ISBN: 978-981-314-912-0. DOI: 10.1142/9789813149137{\textunderscore}0055.

[23] S. Zhu, A. Kimmel, K. E. Bekris, and A. Boularias, *Fast model identification via physics engines for data-efficient policy search*.

[24] W. Yu, J. Tan, C. K. Liu, and G. Turk, *Preparing for the unknown: Learning a universal policy with online system identification*.

[25] S. James and E. Johns, *3d simulation for robot arm control with deep q-learning*.

[26] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, *Training deep networks with synthetic data: Bridging the reality gap by domain randomization*.

[27] M. Cutler, T. J. Walsh, and J. P. How, "Real-world reinforcement learning via multifidelity simulators," *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 655–671, 2015, ISSN: 1552-3098. DOI: 10.1109/TRO.2015.2419431.

[28] S. Barrett, M. E. Taylor, and P. Stone, "Transfer learning for reinforcement learning on a physical robot," *Ninth International Conference on Autonomous Agents and Multiagent Systems - Adaptive Learning Agents Workshop (AAMAS - ALA)*, 2010.

[29] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey," *Journal of Machine Learning Research*, vol. 10, no. 1, pp. 1633–1685, 2009.

[30] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, *Decaf: A deep convolutional activation feature for generic visual recognition*.

[31] E. Tzeng, C. Devin, J. Hoffman, C. Finn, P. Abbeel, S. Levine, K. Saenko, and T. Darrell, *Adapting deep visuomotor representations with weak pairwise constraints*.

[32] K. Fang, Y. Bai, S. Hinterstoisser, S. Savarese, and M. Kalakrishnan, *Multi-task domain adaptation for deep learning of instance grasping from simulation*.

[33] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, S. Levine, and V. Vanhoucke, *Using simulation and domain adaptation to improve efficiency of deep robotic grasping*.

[34] A. Rajeswaran, S. Ghotra, B. Ravindran, and S. Levine, *Epopt: Learning robust neural network policies using model ensembles*.

[35] I. Mordatch, K. Lowrey, and E. Todorov, "Ensemble-cio: Full-body dynamic motion planning that transfers to physical humanoids," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, W. Burgard, Ed., Piscataway, NJ: IEEE, 2015, pp. 5307–5314, ISBN: 978-1-4799-9994-1. DOI: 10.1109/IROS.2015.7354126.

[36] R. Antonova, S. Cruciani, C. Smith, and D. Kragic, *Reinforcement learning for pivoting task*.

[37] F. Sadeghi and S. Levine, *Cad2rl: Real single-image flight without a single real image*.

[38] T. Başar and P. Bernhard, *H_infinity-Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*, Second edition, ser. Modern Birkhäuser Classics. Boston, MA: Birkhäuser Boston, 2008, ISBN: 9780817647575. DOI: 10.1007/978-0-8176-4757-5.

[39] E. Sariyildiz, R. Oboe, and K. Ohnishi, "Disturbance observer-based robust control and its applications: 35th anniversary overview," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 3, pp. 2042–2053, 2020, ISSN: 0278-0046. DOI: 10.1109/TIE.2019.2903752.

[40] W.-H. Chen, "Disturbance observer based control for nonlinear systems," *IEEE/ASME Transactions on Mechatronics*, vol. 9, no. 4, pp. 706–710, 2004, ISSN: 1083-4435. DOI: 10.1109/TMECH.2004.839034.

[41] J. Morimoto and K. Doya, "Robust reinforcement learning," *Neural computation*, vol. 17, no. 2, pp. 335–359, 2005, ISSN: 0899-7667. DOI: 10.1162/0899766053011528.

[42] J. W. Kim, H. Shim, and I. Yang, "On improving the robustness of reinforcement learning-based controllers using disturbance observer," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, [Piscataway, NJ]: IEEE, 2019, pp. 847–852, ISBN: 978-1-7281-1398-2. DOI: 10.1109/CDC40024.2019.9028930.

[43] S. Sundaram, *Inversion of linear systems*.

[44] S. Koos, J.-B. Mouret, and S. Doncieux, "Crossing the reality gap in evolutionary robotics by promoting transferable controllers," in *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation*, M. Pelikan, Ed., ser. ACM Digital Library, New York, NY: ACM, 2010, p. 119, ISBN: 9781450300728. DOI: 10.1145/1830483.1830505.