Optimized GNC Techniques for Service Robotics

Romisaa Ali[1]

[4]*Dept. Computer and Control Engineering (DAUIN), Politecnico di Torino University, Turin, Italy*

Abstract

In this thesis, we propose the development of an optimal control system for autonomous robots. Our design aims to efficiently guide the robot, determining the best possible route to its destination. We leverage the state-of-the-art Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm to direct the robot. By utilizing a precise navigation system, we can ascertain the robot's position in real-time and manage its movements by adjusting its components. This algorithm, which integrates principles from deep learning and reinforcement learning, offers superior optimization capabilities for robot navigation and control. Notably, our approach facilitates navigation optimization without relying on a pre-existing map and ensures collision avoidance throughout the journey.

Keywords

Optimal control system, Autonomous robots, Twin Delayed Deep Deterministic Policy Gradient (TD3), Navigation system, Deep learning, Reinforcement learning, Optimization capabilities, Collision avoidance

1. Introduction

Reinforcement learning (RL) has become a key method for making smart control policies in service robotics. For these robots to perform effectively in navigating and adapting to dynamic environments. As we use service robots for more tasks, they should be flexible and adaptive in navigation.[1, 2, 3, 4]. The big challenge in service robotics is making good navigation plans that can deal with the surprises of everyday settings. When choosing RL method, a robot's performance and ability to adapt can change a lot. In this paper, we will discuss two significant algorithms: TRPO (Trust Region Policy Optimization) and PPO (Proximal Policy Optimization)[5, 6]. Both algorithms utilize the Trust Region Method (TRM) for optimization [7, 8]. TRM focuses on efficiently refining policies within a designated region, using the Kullback-Leibler divergence as a tool for gauging differences between policies. We will focus on two state-of-the-art methods, Twin Delayed Deep Deterministic Policy Gradient (TD3) and Soft Actor-Critic (SAC), which represent the latest advancements in the field[9, 10, 11, 12, 13], Twin Delayed Deep Deterministic Policy Gradient (TD3) and Soft Actor-Critic (SAC)[14, 15, 16], have stood out. This research looks at how well policy gradient methods[17], especially TD3 and SAC, work in planning paths for service robots. We use a free online platform and special control settings to test this. We'll see how strong the plans are, if they can work in different situations,

AIxIA'23: 22nd International Conference of the Italian Association for Artificial Intelligence, November 06–09, 2023, Rome, Italy

[†] Doctoral project supervised by Marcello Chiaberge (Department of Electronics and Telecommunications (DET), Politecnico di Torino University, Turin, Italy).

romisaa.ali@polito.it (R. Ali)
0000-0002-1234-5678 (R. Ali)

and how well the robots move in new places. The paper is set up like this: In Section 1, it confirms the significance of TRPO, PPO, TD3, and SAC algorithms. Section 2 offers a firsthand account of the author's PhD journey, detailing three experiments that explore and compare these algorithms. Lastly, Section 3 outlines a forward-looking plan, pinpointing challenges and queries aimed at optimizing deep reinforcement learning models.

2. PhD Research Journey: From Deep Reinforcement Learning Review to Comparative Experiments

Since the beginning of my PhD research, I began with an extensive literature review on recent advancements in deep reinforcement learning, paying particular attention to state-of-the-art (SOTA) Policy gradient methodologies. This review process also involved a thorough selection of SOTA algorithms based on various established criteria. My research then progressed to a series of experiments: The initial experiment was dedicated to comparing the performance of the Trust Region Policy Optimization (TRPO) and Proximal Policy Optimization (PPO) algorithms in the context of robot control. Subsequently, in the second experiment I investigated the efficiency of the Twin Delayed Deep Deterministic Policy Gradient (TD3) in optimizing navigation strategies. The final experiment sought to compare TD3 with another recent SOTA method, the Soft Actor-Critic (SAC), particularly in navigating unseen motion control environments[18].

2.1. First Experiment

In this Experiment I aimed to compare the effectiveness of the TRPO and PPO algorithms in controlling robots within two specific environments: ANT and Humanoid, with the goal of directing the robot to move forward and fast. This experiment provides a basis for upcoming, deeper explorations in this field. this experiment presented several limitations.

2.1.1. The limitations

Reliability in Real-world Scenarios: The study encountered potential issues in achieving consistent results, suggesting that the solutions might not be reliable when implemented in real scenarios.

Overfitting: There was a notable risk of the algorithms fitting too closely to the training data, especially in the Humanoid environment, which may affect their performance in unseen or varied situations.

Transferability: The results were tailored for ANT and Humanoid robots, limiting their broader application to different robotic designs or environments.

Evaluation metrics: The primary metrics used for evaluation were average returns and training time. However, important aspects such as algorithm robustness, safety concerns, and potential scalability were not evaluated.

Absence of ROS Integration: The study did not utilize the ROS (Robot Operating System) framework, a standard in many robotic applications. This omission could pose challenges when trying to integrate or deploy the solutions on platforms that rely on ROS[19].

2.2. Second Experiment

The goal of this experiment is to assess how well the TD3 algorithm optimizes navigation policies in three varied environments: static, dynamic-wall, and dynamic-box. assessing its adaptability, effectiveness in handling diverse challenges, and its ability to generalize across different environments, Expanding upon the insights from our first experiment, this experiment addresses several of its limitations. Specifically, the training of the model was conducted within the ROS operating system.

2.2.1. The limitations

Limited Training Scenarios: If the range of training environments is too narrow, the learned policies may face difficulties when introduced to new or different settings.

Overfitting: When an algorithm is trained on a limited number of environments or datasets, it may become specialized, which can affect its performance in unfamiliar scenarios.

Incomplete Training: The training process might require additional time to fully converge and identify the most optimal policy.

Resource Limitations: Having enough computing power, including CPU and GPU, along with memory and storage, is vital. Lack of these resources can affect both learning and deployment of the algorithms.

2.3. Third Experiment

In our third experiment, our research efforts are conducted within the Robotic Operating System (ROS) framework. One of the major adjustments made in this experiment, compared to the previous ones, is the expansion of the training environments. This strategic adjustment was based on challenges identified in our previous experiments. By incorporating more numbers of environments, we aim to enhance the adaptability of our model from training scenarios to test environments. The primary objective of this experiment is to reevaluate the TD3 algorithm, which was a significant component of our second experiment. We are particularly interested in comparing its performance with the SAC algorithm. SAC, known for its high-entropy policy methodology, offers a different approach to robotic navigation optimization. This comparison, in both training and testing stages, aims to understand the effectiveness and robustness of these algorithms when applied to robotic navigation within the ROS framework. In this third experiment, we made efforts to reduce some of the limitations identified in the second experiment, robot start point and end point simulation, As shown in Figure 1, the robot navigates from the starting point to the destination..

2.3.1. The limitations

Time to Convergence: The model requires an extended period to converge and ascertain optimal policies.

Computational Resources: Addressing the convergence time limitation necessitates the deployment of heightened computational resources.

Distribution Shift: difference between the training and testing environments. If the test scenarios have different characteristics or distributions of states that the model has not encountered during training.

3. PhD Final Year: A Comprehensive Plan challenges

Deep reinforcement learning (DRL) is rapidly advancing, and there are key challenges we need to address. in this section, there are important elements we can consider to improve the DRL model, In light of the unresolved issues in my research, during my participation in the Doctoral Consortium, I aim to garner answers to the following key questions, with the guidance of my mentors and feedback from attendees:

Transfer-ability with Minimal Distribution Shift: Optimizing the packages to simplify the transition from training to testing, aiming to minimize distribution shift from training environments to testing, and ultimately to deployment scenarios, especially when each of these stages possesses its own distinct characteristics and complexities?

Adaptabilitywith Minimal Distribution Shift: How can we incorporate advanced techniques within the deep reinforcement learning model, that enable it to adapt in real-time to environmental changes, ensuring effective navigation?

Generalizability: What technique can we employ in deep reinforcement learning models to enhance navigation effectively in unseen environments?

Efficient Training in DRL Models: How can I optimize deep reinforcement learning models to significantly reduce both training time and the need for computational resources, while still ensuring their ability to adapt to unfamiliar environments? Additionally, which strategies are most effective in achieving this goal?

Addressing Memory and Speed Challengess: How can we design training scenarios that eliminate the need for detailed simulations, thereby cutting down on memory usage and possibly accelerating the speed at which models learn? This approach aims to overcome the high memory consumption and inefficiencies often seen with traditional simulation methods.

4. ACKNOWLEDGMENT

Deep appreciation is conveyed to Professor Marcello Chiaberge for his invaluable supervision and guidance throughout this research. Warm thanks go to the SCUDO Office at Politecnico di Torino for their essential support and assistance. Recognition must also be given to REPLY Concepts company, especially to Mr. Maurizio Griva and Mr. Simone Voto, for their invaluable input and collaboration. Finally, sincere gratitude is extended to the Department of DAUIN at Politecnico di Torino, as well as to the dedicated staff of PIC4SeR, for their cooperation and support in the execution of this work. The contributions of Mr. Zifan Xu from the University of Texas at Austin were particularly instrumental to the success of this work and are deeply acknowledged.

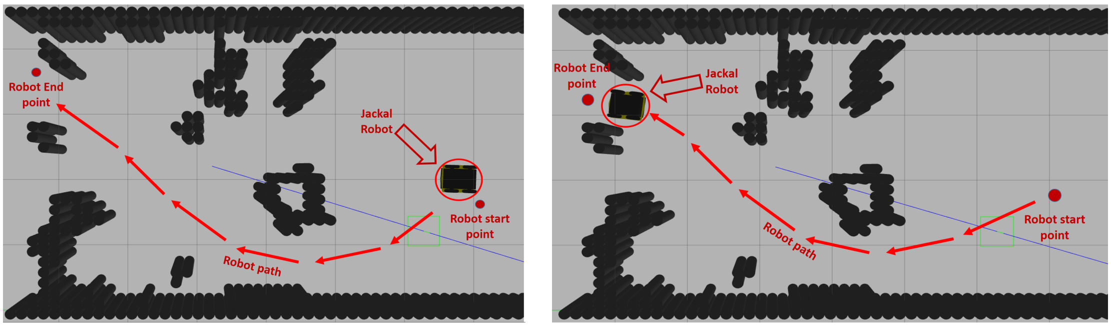

Figure 1: Robot navigation from start to end. Left: Starting point. Right: Destination.

References

[1] R. S. Sutton, A. G. Barto, Reinforcement learning - an introduction, Adaptive computation and machine learning, MIT Press, 1998. URL: https://www.worldcat.org/oclc/37293240.

[2] M. Sewak, Deep Reinforcement Learning - Frontiers of Artificial Intelligence, Springer, 2019. doi:10.1007/978-981-13-8285-7, https://doi.org/10.1007/978-981-13-8285-7.

[3] A. Sehgal, H. M. La, S. J. Louis, H. Nguyen, Deep reinforcement learning using genetic algorithm for parameter optimization, CoRR abs/1905.04100 (2019). http://arxiv.org/abs/1905.04100.

[4] A. S. Anand, J. E. Kveen, F. J. Abu-Dakka, E. I. Grøtli, J. T. Gravdahl, Addressing sample efficiency and model-bias in model-based reinforcement learning, in: 21st IEEE International Conference on Machine Learning and Applications, ICMLA 2022, Nassau, Bahamas, December 12-14, 2022, IEEE, 2022, pp. 1–6. doi:10.1109/ICMLA55696.2022.00009, https://doi.org/10.1109/ICMLA55696.2022.00009.

[5] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, P. Abbeel, Trust region policy optimization, CoRR abs/1502.05477 (2015). http://arxiv.org/abs/1502.05477.

[6] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, CoRR abs/1707.06347 (2017). http://arxiv.org/abs/1707.06347.

[7] A. Mohapatra, Trust region methods for deep reinforcement learning, https://medium.com/

analytics-vidhya/trust-region-methods-for-deep-reinforcement-learning-e7e2a8460284, 2023.

[8] K. Lange, MM Optimization Algorithms, Technical Report, SIAM, 2016. https://www.siam.org/Publications/Books/Call-for-Book-Proposals/MM-Optimization-Algorithms.

[9] Z. Xu, B. Liu, X. Xiao, A. Nair, P. Stone, Benchmarking reinforcement learning techniques for autonomous navigation, CoRR abs/2210.04839 (2022). https://doi.org/10.48550/arXiv.2210.04839.

[10] Z. Xu, B. Liu, X. Xiao, A. Nair, P. Stone, Benchmarking reinforcement learning techniques for autonomous navigation, in: IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023, IEEE, 2023, pp. 9224–9230. https://doi.org/10.1109/ICRA48891.2023.10160583.

[11] Y. He, Y. Kim, HEV energy management strategy based on TD3 with prioritized exploration and experience replay, in: American Control Conference, ACC 2023, San Diego, CA, USA, May 31 - June 2, 2023, IEEE, 2023, pp. 1753–1758. https://doi.org/10.23919/ACC55779.2023.10156220.

[12] J. Wu, Q. M. J. Wu, S. Chen, F. Pourpanah, D. Huang, A-TD3: an adaptive asynchronous twin delayed deep deterministic for continuous action spaces, IEEE Access 10 (2022) 128077–128089. https://doi.org/10.1109/ACCESS.2022.3226446.

[13] Y. Tan, Y. Lin, T. Liu, H. Min, PL-TD3: A dynamic path planning algorithm of mobile robot, in: IEEE International Conference on Systems, Man, and Cybernetics, SMC 2022, Prague, Czech Republic, October 9-12, 2022, IEEE, 2022, pp. 3040–3045. https://doi.org/10.1109/SMC53654.2022.9945119.

[14] K. Nakhleh, M. Raza, M. Tang, M. Andrews, R. Boney, I. Hadzic, J. Lee, A. Mohajeri, K. Palyutina, Sacplanner: Real-world collision avoidance with a soft actor critic local planner and polar state representations, in: IEEE International Conference on Robotics and Automation, ICRA 2023, London, UK, May 29 - June 2, 2023, IEEE, 2023, pp. 9464–9470. https://doi.org/10.1109/ICRA48891.2023.10161129.

[15] J. B. Martin, R. Chekroun, F. Moutarde, Learning from demonstrations with SACR2: soft actor-critic with reward relabeling, volume abs/2110.14464, 2021. https://arxiv.org/abs/2110.14464.

[16] L. Chavali, T. Gupta, P. Saxena, SAC-AP: soft actor critic based deep reinforcement learning for alert prioritization, in: IEEE Congress on Evolutionary Computation, CEC 2022, Padua, Italy, July 18-23, 2022, IEEE, 2022, pp. 1–8. https://doi.org/10.1109/CEC55065.2022.9870423.

[17] S. Fujimoto, H. van Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in: Proceedings of the 35th International Conference on Machine Learning, ICML 2018, PMLR, 2018, pp. 1582–1591. http://proceedings.mlr.press/v80/fujimoto18a.html.

[18] B. Siciliano, O. Khatib, Robotics and the handbook, in: B. Siciliano, O. Khatib (Eds.), Springer Handbook of Robotics, Springer Handbooks, Springer, 2016, pp. 1–10. https://doi.org/10.1007/978-3-319-32552-1_1.

[19] K. Lange, MM Optimization Algorithms, Technical Report, SIAM, 2016. https://www.siam.org/Publications/Books/Call-for-Book-Proposals/MM-Optimization-Algorithms.