CS230

Effect of Demonstrations for Deep Reinforcement Learning-Based Control of Concentric Tube Robots

Fredrik S. Solberg
Department of Mechanical Engineering
Stanford University
fsolberg@stanford.edu

Abstract

Concentric tube robots (CTRs) are challenging systems to control because of their nonlinear effects and unpredictable internal interactions. Fortunately, data-driven models have shown promises in CTR modeling both in terms of accuracy and efficiency. Among data-driven models, deep reinforcement learning (DRL) form a particularly interesting approach, bypassing the challenge of robot manipulation data not being independently and identically distributed by maximizing a long-term reward. Still, for DRL to be useful in real-world applications, algorithms need to lend themselves to fast inference. Demonstrations have previously shown to greatly reduce time-costly exploration in DRL for various robot tasks. This project aims to investigate the effect of demonstrations in DRL for a CTR environment. To do so, an *OpenAI Gym* simulation environment is developed. In the simulation environment, two different agents are trained by using deep deterministic policy gradient (DDPG) and hindsight experience replay (HER) with and without demonstrations for a two-tube CTR. It is found that DDPG and HER without demonstrations uses more than 14 million steps to catch up with DDPG and HER with demonstrations. If a CTR step time of 2 milliseconds is assumed, this would be analogous to more than three days of continuous operation. Moreover, the effect of demonstrations in the initialization of the DDPG policy is evident, as the agent success rate is around 50% already after the first epoch. Still, DDPG and HER without demonstrations converges to a higher success rate than DDPG and HER with demonstrations, with 75% and 66%, respectively. This might be caused by suboptimal hyperparameter selection or bias towards specific strategies caused by the demonstration, and needs to be further studied. Additionally, the success rate is assumed to be suffering from suboptimal goal generation yielding episodes that are unsolvable.

1 Introduction

Concentric tube robots (CTRs) are hyper-redundant manipulators that are composed of multiple concentrically aligned pre-curved super-elastic tubes. Their slender form factor as well as their inherently compliant nature makes them well suited for various complex and sensitive environments, such as minimally invasive surgery. However, the tubes' nonlinear behavior and unpredictable interactions make conventional kinematic modeling of CTRs challenging. Fortunately, data-driven approaches for control of continuum robots have shown to be faster and more accurate as compared to conventional inverse kinematic strategies, making learning-based control a promising paradigm for CTR control [1].

CS230: Deep Learning, Autumn 2020, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

A general challenge for supervised learning in robot control is the temporal correlation between actions and states, yielding data that is not independently and identically distributed. Ignoring these effects can lead to compounding errors [2]. This is particularly the case for CTRs, where effects such as rotation direction has an impact on the tip position [3]. Deep reinforcement learning (DRL) form a favorable alternative as it maximizes an expected *long-term* reward by allowing a software agent to interact with an environment through actions and observations. Still, for DRL to be useful in real-world applications, algorithms need to lend themselves to fast inference. This is obvious when considering that robots are unable to move faster than their physical capabilities, making time a premium.

In this project, the effect of demonstrations to overcome time-costly exploration in DRL for CTRs is investigated. More specifically, policy learning by deep deterministic policy gradient (DDPG) and hindsight experience replay (HER) with and without demonstrations is compared. DDPG and HER, hereafter denoted DDPG+HER, without demonstrations is used as the baseline model, as this previously has been suggested for CTR control in literature [4]. All analysis in this project is done in a developed *OpenAI Gym* [5] CTR simulation environment, described in Section 3.

2 Background and Related Literature

2.1 Deep Deterministic Policy Gradients (DDPG)

Only a brief outline of DDPG is given in this section. For detailed analysis, it is referred to the original work by Lillicrap et al. [6], as well as the documentation for the algorithm used for this project [7]. The outline below is based on the description in [7].

At a high level, DDPG tries to learn a deterministic policy $\mu_\theta(s)$ for state s that gives an action a that maximizes a model $Q_\phi(s, a)$, which is an approximation of the Bellman equation (e.g. approximated by neural network). Hence, the policy is learned by performing gradient ascent to solve

$$\max_{\theta} \underset{(s,a,r,s')\sim\mathcal{D}}{E}[Q_\phi(s, \mu_\theta(s))]. \tag{1}$$

Here , $\mathcal{D}$ denotes a set of transitions (s, a, r, s') defined by the state, action, reward, and next state, respectively. Concurrently, Q-learning is performed by performing gradient descent on the the mean-squared Bellman error loss

$$\min_{\phi} \underset{(s,a,r,s')\sim\mathcal{D}}{E}\left[\left(Q_\phi(s, a) - \left(r + \gamma Q_{\phi_{\text{target}}}(s', \mu_{\theta_{\text{target}}}(s'))\right)\right)^2\right]. \tag{2}$$

Note that the terminating factor originally included in [7] is removed from (2). This is because the developed CTR environment is non-terminating. The target networks θ_{target} and ϕ_{target} is updated by polyak averaging once per main network update.

2.2 Hindsight Experience Replay (HER)

The intuition behind HER is that even though the desired goal is not reached, *some* goal is reached. Therefore, after reaching *some* goal, the experience can be replayed with the same actions, but with the previously erroneous goal set as the desired goal. In this way, HER can make an agent learn to achieve arbitrary goals. Combining HER with sparse rewards has previously proven to be effective for complex robot control tasks [8]. For details about HER it is referred to the original work of Andrychowicz et al. [9].

2.3 Reinforcement Learning with Demonstrations

Previous work has combined DRL with demonstrations. For example, Shall [10] demonstrated a robot capable of learning to balance a pole by a single trial after 30 seconds of demonstration. In Shall's work, the demonstration was used to initialize policies and forward models. More recently, Nair et al. [11] proposed to use demonstrations throughout the training process in addition to in the initialization process, and showed promising results for multi-step robot tasks with varying goal

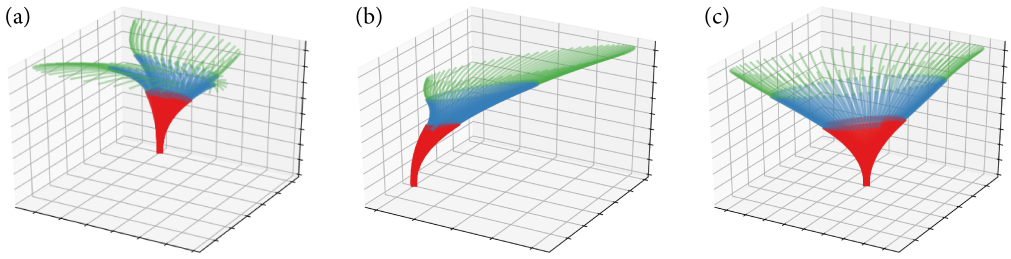

Figure 1: Effect of rotations for a three-tube concentric tube robot; 360° rotation with 5° increments of (a) the inner tube (green), (b) the mid tube (blue), and (c) the outer tube (red). All of the plots have the same starting configuration.

states, not unlike the simulation environment developed for this project. In the work by Nair et al., demonstrations was used with DDPG+HER. Also Rajeswaran et al. [12] proved the effects of demonstrations, enabling robot learning of complex high-dimensional dexterous manipulation tasks by providing a small number of human demonstrations. In this project, the algorithm proposed by Nair et al. is adopted and, therefore, it is referred to their original work for details [11].

3 Simulation Environment

One of the challenges of this project is that in order to train an agent for CTR control, an environment simulating the behaviour of a CTR is needed. This section describes the details of the developed simulation environment. The environment is developed to interface with *OpenAI Gym* [5], a framework for which a number of high-quality implementations of reinforcement learning algorithms is developed (e.g., [13, 14]).

3.1 Concentric Tube Robot Kinematics

Considering the complexity of the equations describing the kinematics of CTRs, it is outside of the scope of this article to go into the details of the equations. Instead, it is referred to the computationally fast compliant kinematic CTR model proposed by Xu and Patel [15], which forms the backbone of the developed simulation environment. Nonetheless, Figure 1 illustrates the highly nonlinear behavior and complex interaction of CTRs, where different tubes of a three-tube CTR is rotated and the overall shape of the CTR is estimated by the implemented kinematic model.

3.2 Observation

Observations refers to what an agent sees at each step. In the developed environment, the observed state of the CTR is represented by the rotation and translation of a tube i, α_i and β_i, respectively, where $i = 1, ..., n$ and n is the number of tubes of the CTR. Additionally, the Cartesian tip position, p, and the desired Cartesian tip position, p^*, are included in the observations. The rotation of each tube is represented as the absolute value of the quaternion of the rotation. The absolute value is used to account for the dual orientation representation of quaternions. Quaternions have previously shown to improve performance for learning CTR kinematics in supervised learning [16].

To summarize, each observation is represented as

$$(\gamma_1, ..., \gamma_n, p, p^*), \tag{3}$$

where $\gamma_i = (|q_{0i}|, |q_{3i}|, \beta_i)$ represents the state of tube i, and q_{0i} and q_{3i} the quaternion's real component and the imaginary component about the z axis, respectively.

3.3 Action

Actions refers to what an agent is allowed to do at each step. In the developed environment, the action is represented as a change in rotation and translation for each tube. Analogous to the work of Iyengar, Dwyer, and Stoyanov [4]; in this work, the rotation limit for each step is set to $\pm 5^\circ$ and the translation limit is set to ± 0.1mm. Thus, the agent can select any value in the continuous range within the limit interval. An additional constraint is appointed to the translations of the tubes to ensure that actions do not result in unsolvable CTR configurations, i.e. the tubes disappearing into each other:

$$\frac{L_1}{2}(\beta_1 + 1) \geq \frac{L_2}{2}(\beta_2 + 1) \geq ... \geq \frac{L_n}{2}(\beta_n + 1) \geq 0, \tag{4a}$$

$$\frac{L_1}{2}(\beta_1 - 1) \leq \frac{L_2}{2}(\beta_2 - 1) \leq ... \leq \frac{L_n}{2}(\beta_n - 1) \leq 0. \tag{4b}$$

Here, L_i is the length of the ith tube and $\beta_i \in [-1, 1]$, where $i = 1$ denotes the innermost tube. If an action causes the conditions in Eq. (4a,b) to be violated, the illegal action is set to zero.

3.4 Reward

Rewards refers to the score the agent receives after conducting a step. The simulation environment is developed for use with both sparse and dense rewards. However, only sparse rewards are used in this project and, therefore, described in this section. To determine the reward, the Euclidean distance, d, between the current tip position, p, and desired tip position, p^*, is calculated as

$$d = \|p - p^*\|_2^2. \tag{5}$$

The reward, r, is then defined as

$$r = \begin{cases} 0, & \text{if } d < \delta \\ -1, & \text{otherwise} \end{cases} \tag{6}$$

where δ is the goal tolerance. A goal tolerance of 1 mm is used in this project.

3.5 Logic

Each episode of the simulation environment is initialized by generating a random initial position and a goal. The goal position, p^*, is determined by solving the forward kinematics for a random set of rotations and translations, where the resulting tip position is stored as the goal position. After the goal is obtained, the agent is allowed to interact with the environment for a limited number of steps, starting from the randomly generated initial position. In this project, a maximum of 150 steps is used. The success of each step is determined as described in Section 3.4; after each step a reward is given accordingly. When the maximum number of steps is reached, a new initial position and goal position are generated and the logic repeats.

4 Method and Results

The performance of DDPG+HER with and without demonstrations is evaluated on a two-tube CTR environment. A two-tube environment is chosen to minimize computational time while still maintaining some of the nonlinear effects of CTR tube interactions. Nonetheless, DRL algorithms have indicated to comparable for two-, three-, and four-tube CTR environments [4]. Detailed CTR parameters used in the simulations can be found in Appendix A.

Both models is trained on Stanford University's Sherlock cluster using 19 CPU cores. Analogous to [8], each core generates experience using two parallel rollouts and uses MPI for synchronization. The performance after each epoch is evaluated by performing 10 deterministic test rollouts per MPI worker, followed by estimation of the test success rate by averaging across rollouts and MPI workers. One epoch of DDPG+HER without demonstrations consists of 50 episodes per rollout per MPI

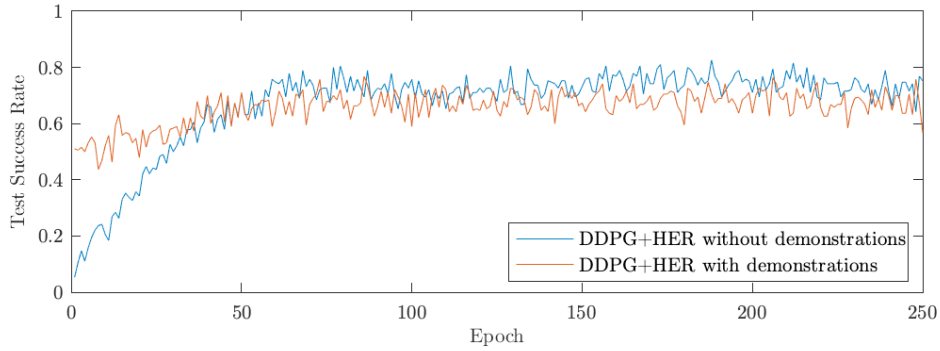

Figure 2: Success rate of trained agents. One epoch corresponds to 760 and 1900 episodes for DDPG+HER with and without demonstrations, respectively. Each episode consists of 150 steps.

worker, analogous to [8]. One epoch of DDPG+HER with demonstrations consists of 20 episodes per rollout per MPI worker, following [11]. The complete set of hyperparameters used in the analysis are detailed in Appendix B.

To provide demonstrations to the model, 100 successful examples was recorded by sequentially minimizing the difference between rotations and translations of a random goal and initial position.

The test success rate of the models is plotted in Figure 2.

5 Discussion and Future Work

From Figure 2, the effect of demonstrations is evident with the test success rate being approximately 50% after the first epoch. DDPG+HER without demonstrations catches up with the DDPG+HER with demonstrations in approximately 50 epochs, which corresponds to 95,000 episodes (50 epochs $\times$ 50 episodes $\times$ 2 rollouts $\times$ 19 MPI workers) or 14,250,000 steps (150 steps per episode). If a step time of 20 milliseconds is assumed, this would be analogous to approximately 79 hours of continuous operation. Moreover, DDPG+HER without demonstrations reaches a success rate approximately equal to the starting success of DDPG+HER with demonstrations after 30 epochs, corresponding to 57,000 episodes.

DDPG+HER with demonstrations converges to a test success rate of 66% in approximtely 40 epochs, which corresponds to 30,400 episodes. In comparison, DDPG+HER without demonstrations converges to a success rate of 75% in approximately 65 epochs, or 123,500 episodes. It is assumed that the difference in test success rate is caused by suboptimal hyperparameter selection. DDPG's sensitivity to hyperparameters is a well-known problem [17]. An alternative explanation is that the demonstrations biases the policy towards a specific strategy. Going forward, hyperparameter optimization should be conducted, in addition to exploration of how to best generate and use demonstrations throughout training. Other algorithms that addresses many of the issues with the hyperparameter sensitivity of DDPG should also be considered, such as soft actor-critic (SAC) [18] or twin-delayed deep deterministic policy gradient (TD3) [19].

Regardless, a success rate around 70% is considered promising when taking the current implementation of the simulation environment into account, where a maximum number of steps of 150 is not sufficient to span the complete workspace given the action restrictions described in Section 3.3. In future iterations of the simulation environment, care should be taken to generate goal states that is reachable from the starting position in an episode. This would likely result in a higher success rate.

6 Conclusion

It has been seen that demonstrations have the potential to greatly reduce time-costly exploration in DRL for CTRs. Still, future efforts needs to be focused on optimal hyperparameter selection and invetigation of the effects of demonstrations, as it is experienced that DDPG+HER with demonstra-

tions converges to a lower success rate as compared to DDPG+HER without demonstrations. The simulation environment also needs to be updated to make sure that generated goals are reachable within an episode.

References

[1] Wenjun Xu, Jie Chen, Henry YK Lau, and Hongliang Ren. Data-driven methods towards learning the highly nonlinear inverse kinematics of tendon-driven surgical manipulators. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 13(3):e1774, 2017.

[2] J Andrew Bagnell. An invitation to imitation. Technical report, CARNEGIE-MELLON UNIV PITTSBURGH PA ROBOTICS INST, 2015.

[3] Junhyoung Ha, Georgios Fagogenis, and Pierre E Dupont. Modeling tube clearance and bounding the effect of friction in concentric tube robot kinematics. *IEEE Transactions on Robotics*, 35(2):353–370, 2018.

[4] Keshav Iyengar, George Dwyer, and Danail Stoyanov. Investigating exploration for deep reinforcement learning of concentric tube robot control. *International Journal of Computer Assisted Radiology and Surgery*, 2020.

[5] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[6] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[7] Joshua Achiam. Spinning Up in Deep Reinforcement Learning. 2018.

[8] Matthias Plappert, Marcin Andrychowicz, Alex Ray, Bob McGrew, Bowen Baker, Glenn Powell, Jonas Schneider, Josh Tobin, Maciek Chociej, Peter Welinder, et al. Multi-goal reinforcement learning: Challenging robotics environments and request for research. *arXiv preprint arXiv:1802.09464*, 2018.

[9] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30:5048–5058, 2017.

[10] Stefan Schaal. Learning from demonstration. In *Advances in neural information processing systems*, pages 1040–1046, 1997.

[11] Ashvin Nair, Bob McGrew, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Overcoming exploration in reinforcement learning with demonstrations. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6292–6299. IEEE, 2018.

[12] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *arXiv preprint arXiv:1709.10087*, 2017.

[13] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines. `https://github.com/openai/baselines`, 2017.

[14] Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. `https://github.com/hill-a/stable-baselines`, 2018.

[15] R Xu and RV Patel. A fast torsionally compliant kinematic model of concentric-tube robots. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 904–907. IEEE, 2012.

[16] Reinhard Grassmann, Vincent Modes, and Jessica Burgner-Kahrs. Learning the forward and inverse kinematics of a 6-dof concentric tube continuum robot in se (3). In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5125–5132. IEEE, 2018.

[17] Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search provides a competitive approach to reinforcement learning. *arXiv preprint arXiv:1803.07055*, 2018.

[18] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[19] Scott Fujimoto, Herke Van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

A Concentric Tube Robot Parameters

Table 1: CTR parameters used in the simulation

Parameter	**Tube 1**	**Tube 2**
Length (mm)	215	120.2
Curved length (mm)	14.9	21.6
Inner diameter (mm)	1.0	3.0
Outer diameter (mm)	2.4	3.8
Stiffness (GPa)	50	50
Torsional stiffness (GPa)	23	23
Curvature $(1/\mathrm{m})$	15.82	11.8

B Hyperparameters

For DDPG+HER without demonstrations, the following hyperparameters were used:

- Actor and critic networks: 3 layers with 256 units each and ReLU nonlinearities
- Adam optimizer with $1 \cdot 10^{-3}$ for training both actor and critic
- Buffer size: 10^6 transitions
- Polyak-averaging coefficient: 0.95
- Action L2 norm coefficient: 1.0
- Observation clipping: $[-200, 200]$
- Batch size: 256
- Rollouts per MPI worker: 2
- Number of MPI workers: 19
- Cycles per epoch: 50
- Batches per cycle: 40
- Test rollouts per epoch: 10
- Probability of random actions: 0.3
- Scale of additive Gaussian noise: 0.2
- Probability of HER experience replay: 0.8
- Normalized clipping: $[-5, 5]$

For DDPG+HER with demonstrations, the following hyperparameters were used (differing hyperparameters is bolded):

- Actor and critic networks: 3 layers with 256 units each and ReLU nonlinearities
- Adam optimizer with $1 \cdot 10^{-3}$ for training both actor and critic
- Buffer size: 10^6 transitions
- Polyak-averaging coefficient: 0.95
- Action L2 norm coefficient: 1.0
- Observation clipping: $[-200, 200]$
- **Batch size: 1024**
- Rollouts per MPI worker: 2
- Number of MPI workers: 19
- **Cycles per epoch: 20**
- Batches per cycle: 40
- Test rollouts per epoch: 10
- **Probability of random actions: 0.1**
- **Scale of additive Gaussian noise: 0.1**
- Probability of HER experience replay: 0.8
- Normalized clipping: $[-5, 5]$