UC San Diego
JACOBS SCHOOL OF ENGINEERING

Reinforcement Learning for Automating Control on the daVinci™ Surgical Robot

Jingpei Lu and Soumyaraj Sreeman Bose

Objective & Motivation

The goal of this project is to use a reinforcement learning (RL) approach to solve the reach problem on the daVinci™ Research Kit (dVRK) [1], in a manner similar to the Reacher environment in OpenAI Gym. A model of a dVRK arm should be able to move from the initial position to a randomly generated target position, using the fewest possible steps.

Background

Surgical robots are slowly becoming the norm in medicine for the convenience they offer to personnel in executing intricate procedures, while being located at a considerable distance from the emergency room. This convenience becomes a major advantage in scenarios where surgical procedures need to be urgently carried out, but access to medical personnel is restricted, such as war-zones. Hence, automation of the robot surgeon is of prime necessity and has developed into a major focus of AI research over the years.

This project will involve implementing reinforcement learning schemes to facilitate automation on the daVinci™ surgical robot. The daVinci™ is a six degree-of-freedom (DoF) system with links imitating the forearm (joint 1), the hand (joints 2-5) and two fingers/grippers (combined into joint 6). While largely operated manually, the control by computer input is also enabled on it via the daVinci™ Research Kit (dVRK)[1]. Currently, the dVRK enables control of the system in end-effector space. This efforts of this project aim to automate control of the system in joint space.

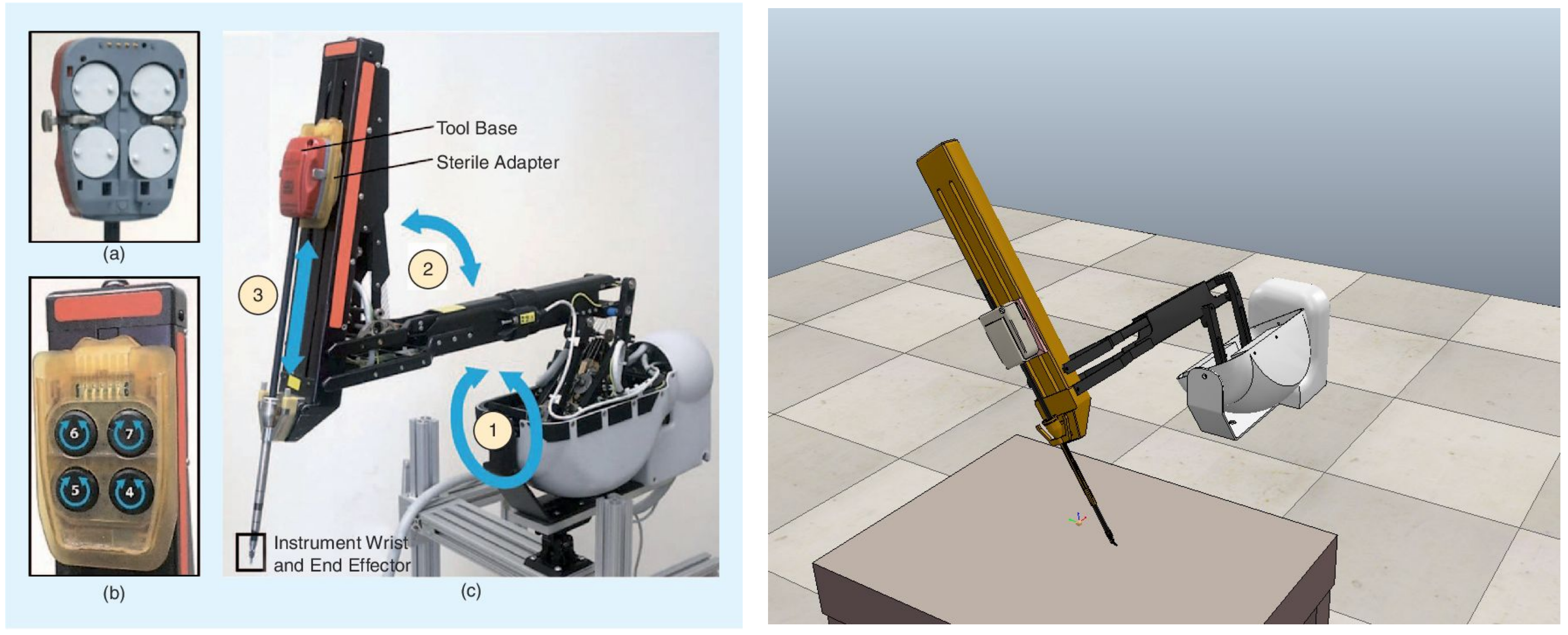
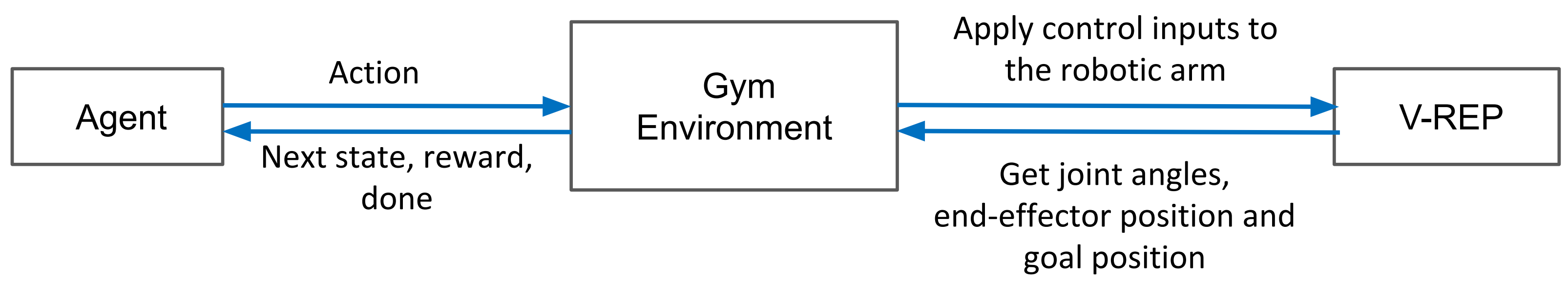


Figure 1: An image of the dVRK patient side manipulator (PSM) and its dynamic model following a trajectory in the V-REP environment

Environment Setup


Agent
Action
Next state, reward, done
Gym Environment
Apply control inputs to the robotic arm
Get joint angles, end-effector position and goal position
V-REP

We customized a gym-like reach environment for the daVinci™ surgical robot by interfacing a dynamically-enabled simulator in a V-REP scene. Values for the dynamic parameters are derived from [3] with some scene-specific adjustments to ensure the simulator is stable. Of the latter, an important one includes enabling PD control on all the joints.

The gripper is driven to reach a random goal position within the bounds of a table by passing target positions (angles) to the first five joints. Its position is, then, recorded for reward and convergence calculations. The goal and gripper are oriented about a base frame in the space of the simulator and, their positions are always computed with respect to this base frame.

Methods and Experiments

To formulate this as an RL problem, we choose the state space to be the joint angles (for the first five joints) in radians + the goal position (x, y, z). The action space is target positions (angles) for the five joints. We also incorporate two types of reward functions. We use the negative of the distance between the end-effector position and goal position as dense reward, and 0/1 reward of successful reach as sparse reward.

To learn the policy of reaching a goal, we trained the agent using Deep Deterministic Policy Gradient (DDPG) and TD3 using dense reward. For experiments, the agents is trained on fixed goal position setting and also random goal position setting.

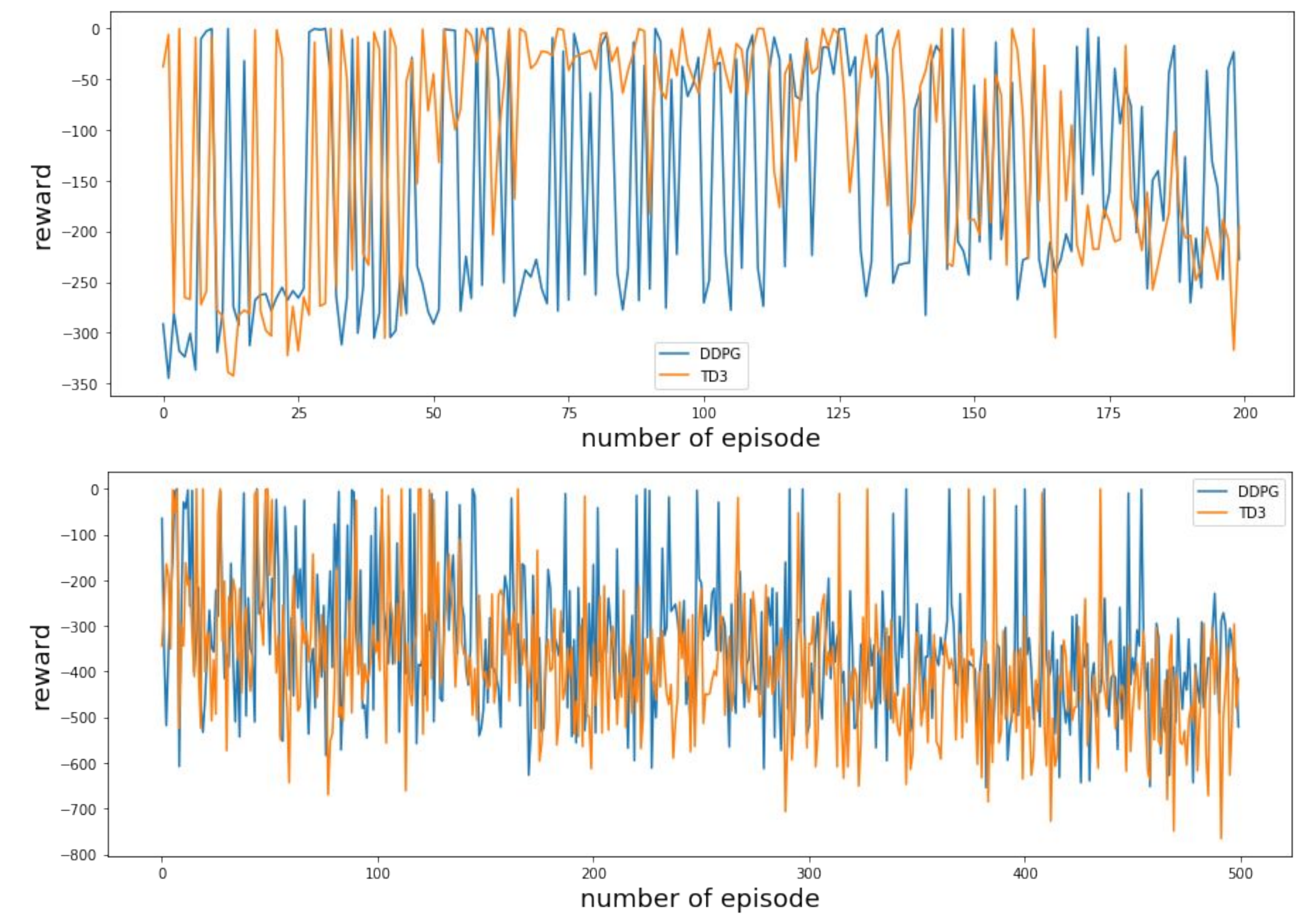


Figure 2. The learning curve using DDPG algorithm. Above: fixed goal. Below: random goal.

Discussion

While the dynamic model allows for the successful training of algorithms such as DDPG and TD3, there are problems which arise in the system that call for attention. A major anomaly (documented below) is the apparent reorientation of the entire model as it is reset over a considerable number of episodes. A plausible reason for this situation is that some of the state variables, though being sampled from a fixed range of values, may have eventually compounded in value after multiple episodes.

Another pitfall experienced by using this model is its inability to learn from a sparse reward setup. Unlike the non-dynamic case where the gripper is set to a particular position and reaches it unfailingly, joint space-control does not guarantee the gripper precisely reaching a target. However, it manages to sensitize the system to its surroundings and allows us to realize the extremes it can reach, thus helping iteratively develop a true representation by giving us the breadth of responses.

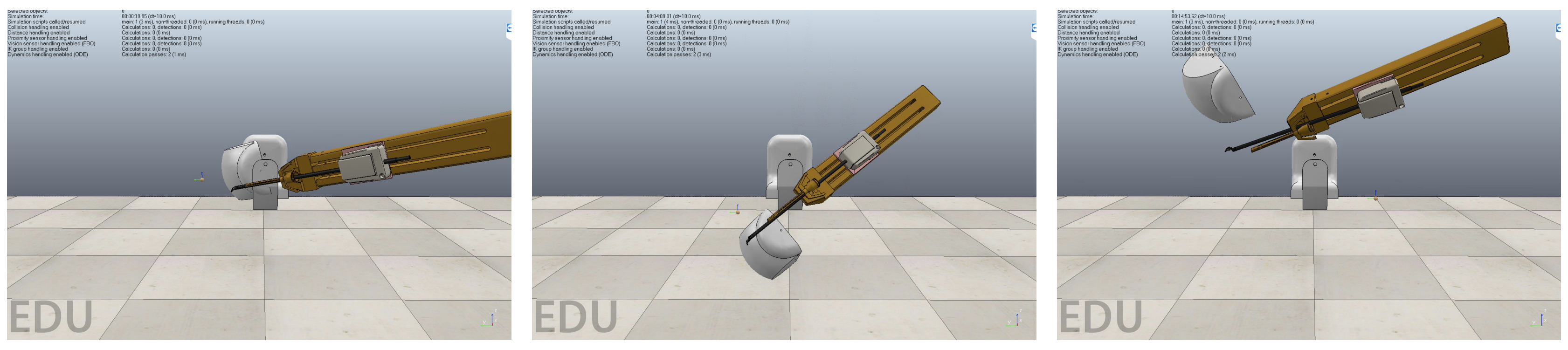


Figure 3: The dVRK arm itself gets reoriented while being trained to reach a point: one among the anomalies rising in generating a solution of the problem

Scope for future work

Future investigations into solving this problem can take two particular routes. The first involves improving the dynamic model being used, since some failure cases in training may be attributed to the system hitting a limit in the dynamics. The second route can involve the identification and application of methods that enable learning from the experience of "failure in attaining goals", but for dense rewards.

Contact

Jingpei Lu: jil360@ucsd.edu
Soumyaraj Sreeman Bose: ssbose@ucsd.edu
Jingpei and Soumyaraj are affiliated with the Department of Electrical & Computer Engineering and the Department of Mathematics, respectively, at UC San Diego.

References

1. The daVinci™ Research Kit (dVRK) developed for academic purposes can be found at https://github.com/jhu-dvrk/sawIntuitiveResearchKit/wiki
2. F. Richter, R. K. Orosco, and M. C. Yip, "Open-sourced reinforcement learning environments for surgical robotics," arXiv preprintarXiv:1903.02090, 2019
3. Y. Wang, R. Gondokaryono, A. Munawar, G. S. Fischer, "A Dynamic Model Identification Package for the da Vinci Research Kit", arXiv abs: 1902:10875, 2019