dVRL: Reinforcement Learning Environments for Surgical Robotics

Florian Richter[1] *Student Member, IEEE*, Ryan K. Orosco[2] *Member, IEEE*,
and Michael C. Yip[1] *Member, IEEE*

***Abstract*— Reinforcement Learning (RL) is a framework that recently has found success by integrating Artificial Intelligence to solve a variety of complex problems. We aim to bring the successes from the RL community to the surgical robotics community by presenting the first open-sourced RL environemnts for surgical robotics, dVRL[3]. By engaging the broader community, which includes both surgical robotics and non-domain experts such as reinforcement learning enthusiasts, new solutions can be contributed to problems that would have real world significance to robotic surgery and the patients that undergo those procedures. To show the effectiveness of the simulated environments, learned policies are transferred to the real robot and sucessfully accomplish surgically relevant tasks.**

I. Introduction

Reinforcement Learning (RL) is a framework that integrates Artificial Intelligence to solve variety of complex problems [1]. The framework is based on a Markov Decision Process where a control policy is learned through interaction with the environment and maximizes the long term reward, which is specified appropriately to solve the problem. This allows for model-free controllers to be found that can solve challenging, non-linear problems. Alongside the ongoing evolution and successes of RL, there is active research in automating surgical tasks [2]. One of the challenges moving forward for the surgical robotics community is that a lot of the recent work is based on hand-crafted control policies that can be difficult to both develop at scale and generalize well.

We aim to bridge these two communities by presenting dVRL, the first open-sourced RL environments for Surgical Robotics. We are motivated to engage the broader community that includes surgical robotics and also non-domain experts, such that reinforcement learning enthusiasts with no domain knowledge of surgery can still easily prototype their algorithms with such an environment and contribute to solutions that would have real world significance to robotic surgery and the patients that undergo those procedures. To evaluate performance of the presented environments, learned policies are transferred to the real robot with minimal effort to automate a surgically relevant task.

II. Methods

The environments are simulated in V-REP based on the da Vinci® Surgical System System scenes developed by Fontanelli et al. [3]. The presented environments only incorporate a single Patient Side Manipulator (PSM), as shown in Fig. 1, but the additional PSM and endoscopic camera arm can be easily added. The PSM arm is controlled in the end-effector space and, for the sake of simplicity, the orientation is held constant. By working in the end-effector space, the learned policies can be transferred to various EndoWrists which are the attachable surgical tools for the da Vinci® Surgical System and have varied kinematic chains. For the simulated scene in the environments presented, the Large Needle Driver (LND) EndoWrist is used, but it can be replaced with other EndoWrist models.

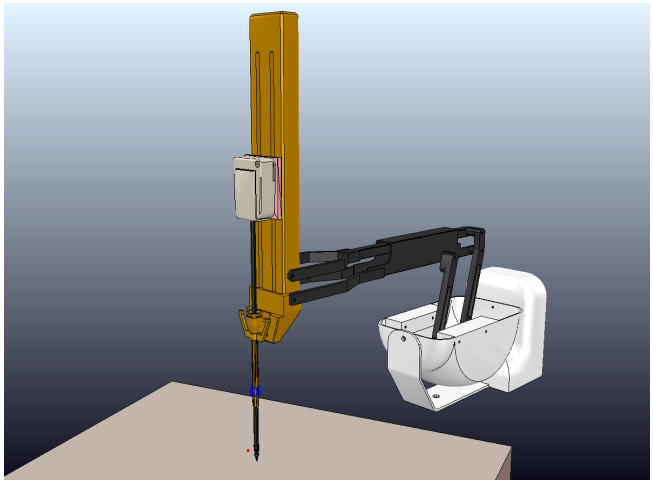

Fig. 1: Simulation scene in V-REP of the single PSM arm. This is the fundamental scene that the presented environments, PSM Reach and PSM Pick, are based on.

A. PSM Reach Environment

The goal of the PSM Reach Environment is to have the end-effector reach a target goal position g starting from any initial configuration, p_0. Note that both g and p_0 are 3-dimensional vectors describing end-effector position and when resetting the environment are randomly sampled from the workspace of the PSM arm. The state and action space of the environment is:

$$s_t = \begin{bmatrix} p_t & g \end{bmatrix} \qquad a_t = \begin{bmatrix} \Delta_t \end{bmatrix} \tag{1}$$

where Δ_t is the control input that changes the end-effector position, which is bounded by [-1, 1]. The next state equation for the position is:

$$p_{t+1} = \eta \Delta_t + p_t \tag{2}$$

where η is a scaling factor. The term η must be kept low enough to ensure stability on the real da Vinci® Surgical System. This is since every new action gives new set points to the joint level controllers. If the new set points are far from the current position, overshoot and instability can occur. The reward function is:

$$r(s_t) = -\mathbb{1}_{||p_t - g|| > \delta} \tag{3}$$

where $\mathbb{1}$ is the indicator function and δ is a threshold distance.

[1]Florian Richter and Michael C. Yip are with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093 USA. {frichter, yip}@ucsd.edu

[2]Ryan K. Orosco is with the Department of Surgery - Division of Head and Neck Surgery, University of California San Diego, La Jolla, CA 92093 USA. rorosco@ucsd.edu

[3]dVRL available at https://github.com/ucsdarclab/dVRL