

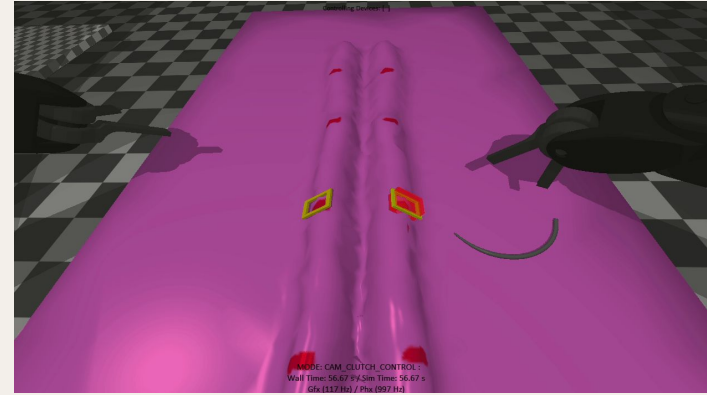
A reinforcement learning approach to robotic suturing

Checkpoint #1

Group 20

Members: Walee Attia, Jocelyn Hsu, Jihoon Kim

Mentors: Dr. Anqi Liu, Dr. Adnan Munawar, Dr. Manish Sahu, Dr. Peter Kazanzides



[4] 2021-2022 AccelNet Surgical Robotics Challenge



PROJECT OVERVIEW

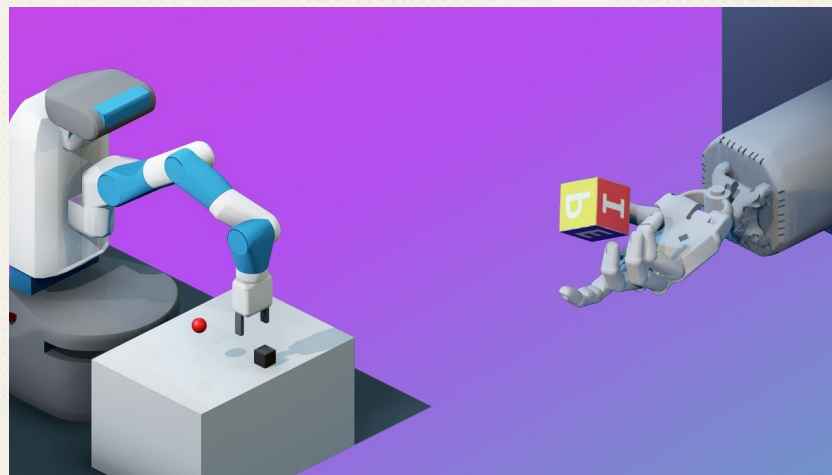
BACKGROUND

Reinforcement Learning (RL)

RL techniques have seen significant progress in the robotics domain; however, there exist a lack of platforms which offer environments conducive to medical robotics.

OpenAI Gym [5]

OpenAI Gym is an open-source RL framework that offer realistic simulation environments with easy integration.



[5] Gymnasium documentation

BACKGROUND

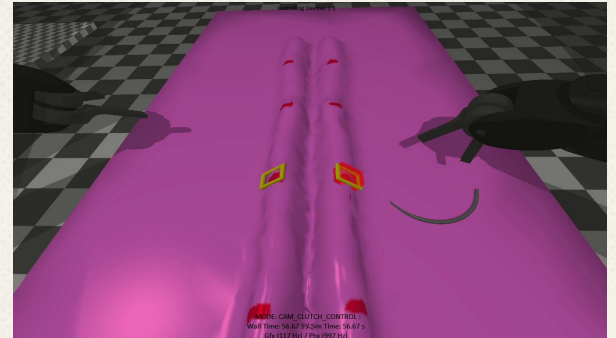
Surgical Robotics Challenge (SRC) [4]

A simulation platform to develop algorithms to address various questions in surgical robotics automation with:

- Two 7-DOF instrument arms based on the da Vinci Surgical System large needle driver
- Controllable camera based on the da Vinci Endoscopic Camera Manipulator
- Suturing phantom
- Needle with a suture

Asynchronous Multibody Framework (AMBF) [7]

A real time dynamics simulator that serves as the backbone for the Surgical Robotics Challenge environment

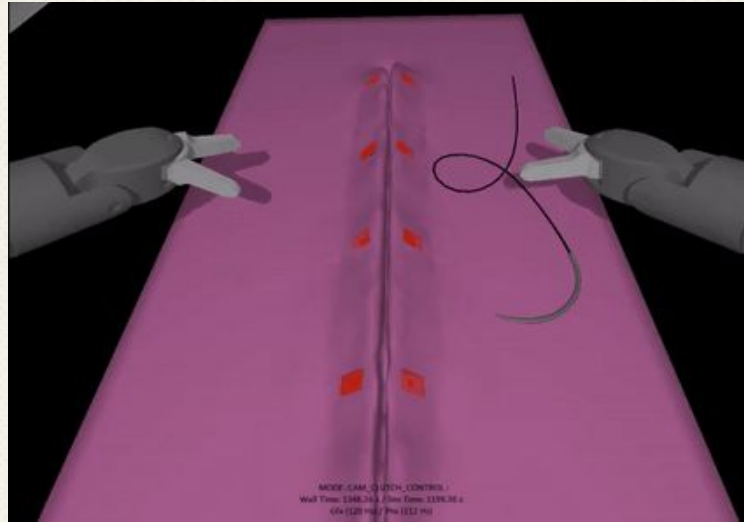


[4] 2021-2022 AccelNet Surgical Robotics Challenge

BACKGROUND

Goal

- Building a Reinforcement Learning platform addressing Challenge 2
 - Grasp needle and drive through tissue

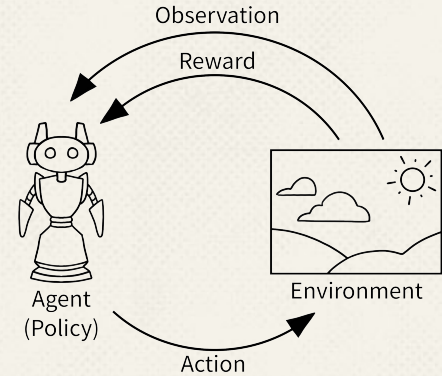




TECHNICAL PROGRESS

REINFORCEMENT LEARNING (RL)

- Reinforcement learning (RL) overview [1]:
 - Agent learns to maximize cumulative reward
 - Interaction with environment through trial and error
- RL framework components:
 - Agent, environment, states (S), actions (A), reward function (R), transition probabilities (P), discount factor (γ)
 - Policy (π) maps states to actions
 - Environment provides feedback (rewards or penalties)
- Agent's goal: maximize cumulative reward over time
 - Use of value function or Q-function to estimate expected cumulative reward
 - Optimal value function updated with Bellman equation
 - Considers current maximized reward and expected future reward



[5] Gymnasium documentation.

Optimal Value Function:

$$V^{\pi^*} = \max_a \{R(s, a) + \gamma \{P(s'|s, a)V^{\pi^*}(s')\}\}$$

RL ALGORITHMS

Deep Deterministic Policy Gradient (DDPG) [8]

- Technique for training RL agents in continuous control problems
 - Extension of Q-learning
 - Deep neural networks for value function and policy function
- **Key components:**
 - Actor network: selects actions based on current state
 - Critic network: evaluates chosen actions and state transition
- **Training:**
 - Actor network: deterministic policy gradient
 - Improve decision making by learning from feedback
 - Critic network: temporal difference (TD) learning
 - Refines assessment of actions based on observations and predicted outcomes
 - Optimize by minimizing loss function

Hindsight Experience Replay (HER) [9]

- An enhancement of DDPG algorithm
 - Addresses sparse rewards, perfect for SRCEnv
- **Main Idea:**
 - Modify reward function during training to learn from unsuccessful attempts
 - Re-label original goal with achieved goal in hindsight
 - Recalculate reward using this modified goal
- **Learning Process:**
 - Modified replay buffer stores original and re-labeled transitions
 - Agents amplies transitions to update policy and value functions
 - Converts unsuccessful episodes into successful ones, providing more learning opportunities
 - Reduces sparsity of reward signal

RL ALGORITHM IMPLEMENTATION

Tianshou: RL Platform based on PyTorch, supports OpenAI Gym Environments [10]

- API of 32 ready-to-use RL algorithms including DDPG and HER
- Seamless integration with SRCEnv (Gym-based)
- Fast-speed, Modular design streamlines development and integration of RL algorithms
- Regularly updated and maintained

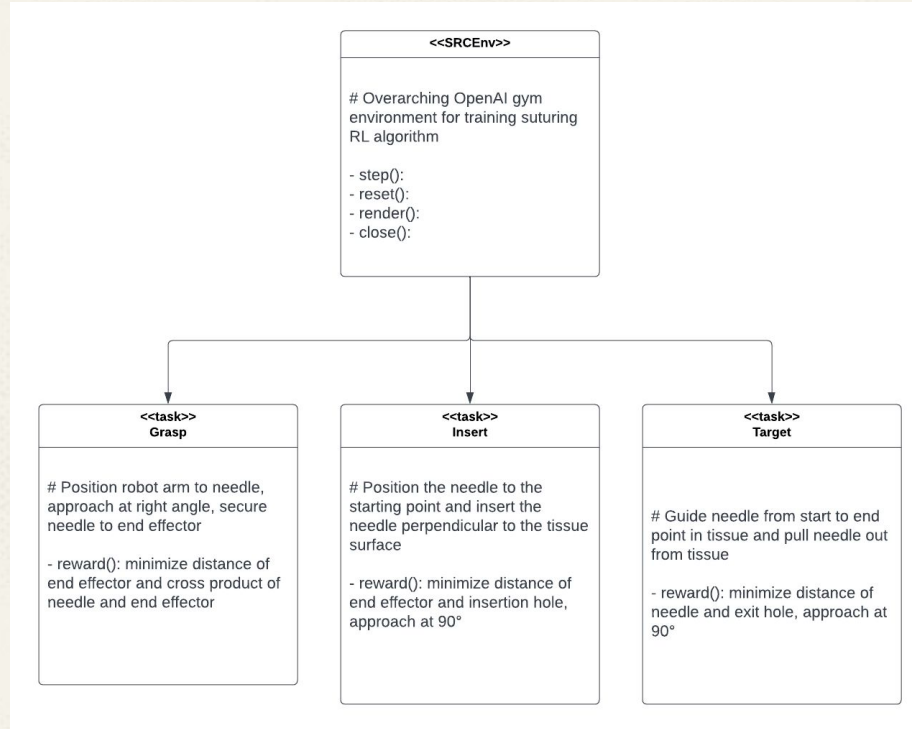
Tianshou API Overview:

- `init`: initializes policy
- `forward`: computes actions from observations
- `process_fn`: preprocesses data from replay buffer
- `learn`: learns from batch data
- `post_process_fn`: updates replay buffer after learning
- `update`: main training interface, calls `process_fn`, `learn`, and `post_process_fn` sequentially



[10] Weng, H. Chen, D. Yan, K. You, A. Duburcq,
M. Zhang, Y. Su, H. Su, and J. Zhu

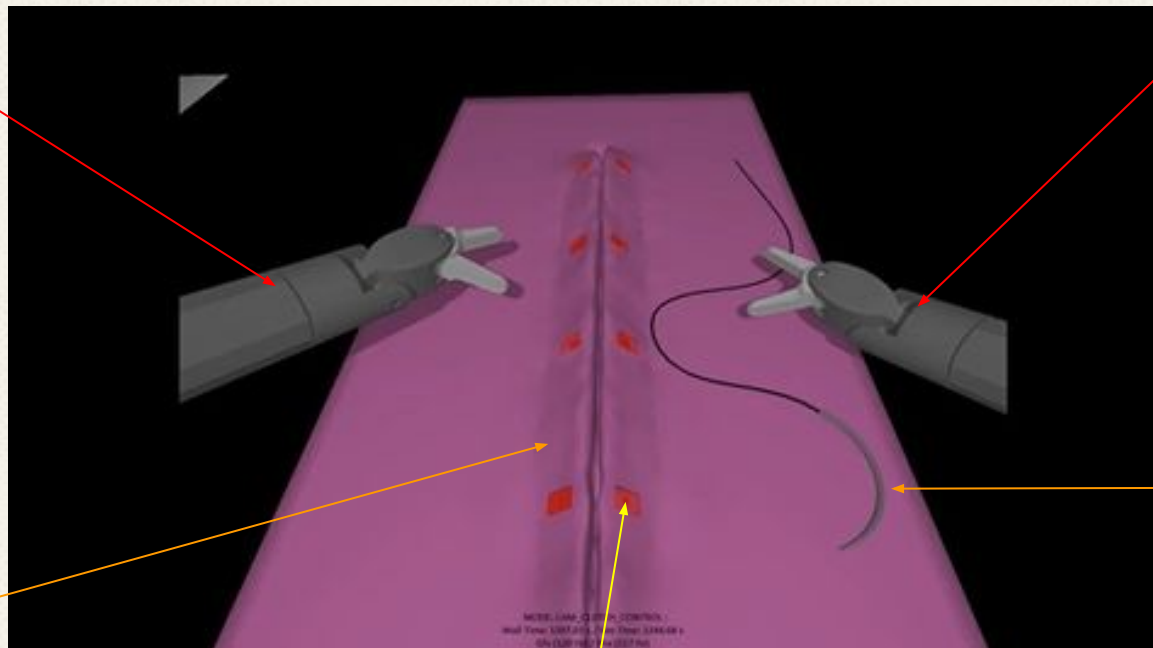
SRC ENVIRONMENT STRUCTURE



SRC ENVIRONMENT: INIT

PSM arm 1

PSM arm 2



Suturing Phantom

Needle

Insert/Target Hole

SRCEnv DEVELOPMENT

step() :

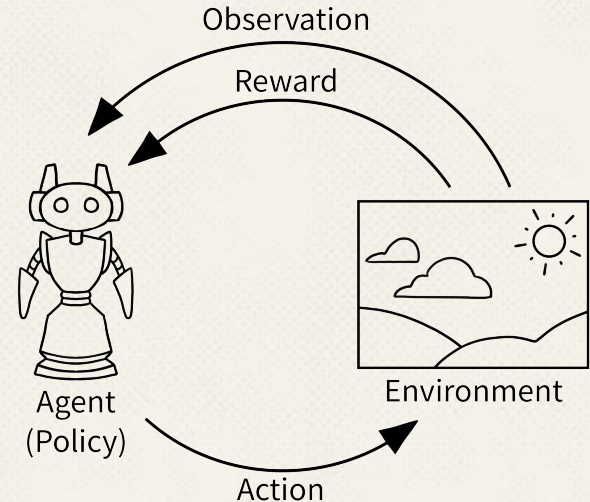
- Adjusts the robotic arm position based on the calculated action and returns the observation and reward resulting from the action.

Parameters:

- `self`: robotic arm position, needle position
- `action`: change in robotic arm position

Returns:

- `observation`: new robotic arm position, needle position
- `reward`: calculated reward score
- `terminated`: True if task accomplished, False otherwise
- `truncated`: True if failing to accomplish task within certain number of iterations, False otherwise
- `info`: optional diagnostic information



[5] Gymnasium documentation.

```
<<SRCEnv>>

# Overarching OpenAI gym
environment for training suturing
RL algorithm

- step():
- reset():
- render():
- close():
```

SRCEnv DEVELOPMENT

render () :

- Visualize the state of the environment from the agent's perspective through the `SimulationManager + AMBFClient`

SimulationManager + AMBFClient:

- Communication interface that connects and obtains the state of each ROS object and renders it in AMBF simulator

Parameters:

- `self`: robotic arm position, needle position

Returns:

- `None`.

close () :

- Completes and exits simulation

Parameters:

- `self`

Returns:

- `Env.unwrapped` raw state of the environment

reset () :

- Sets all environment variables to its initial state
- PSM arms and needle to original position

Parameters:

- `self`: robotic arm position, needle position

Returns:

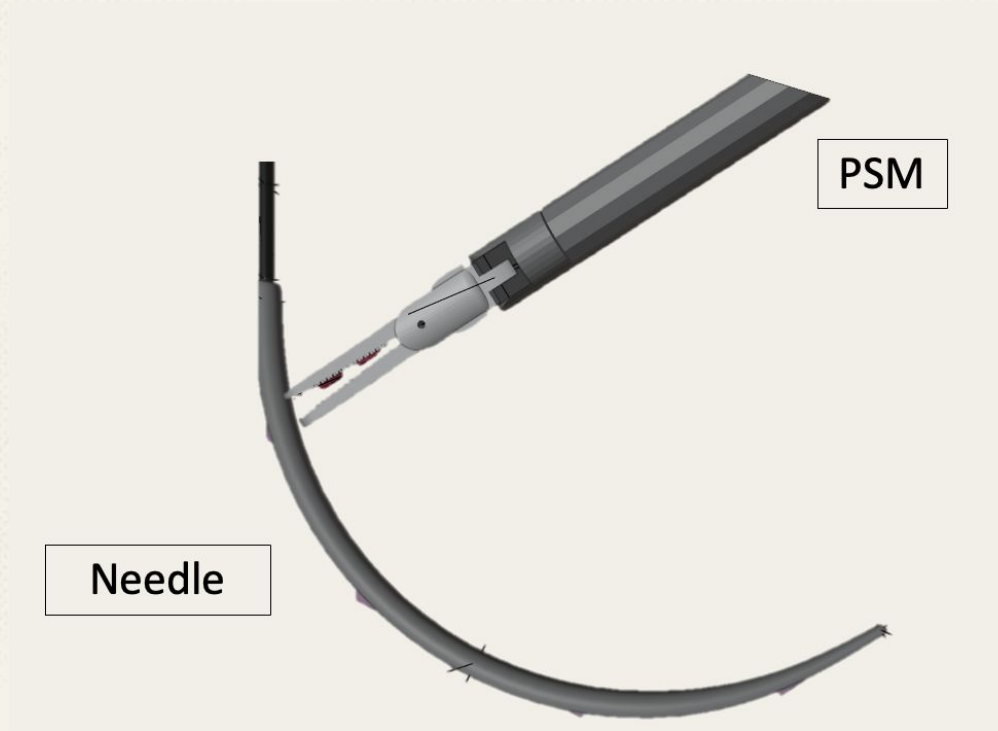
- `None`.

```
<<SRCEnv>>

# Overarching OpenAI gym
environment for training suturing
RL algorithm

- step():
- reset():
- render():
- close():
```

GRASP REWARD: IDEAL POSITION



***Not drawn to scale**

<code><task></code> Grasp
Position robot arm to needle, approach at right angle, secure needle to end effector
- reward(): minimize distance of end effector and dot product of needle and end effector

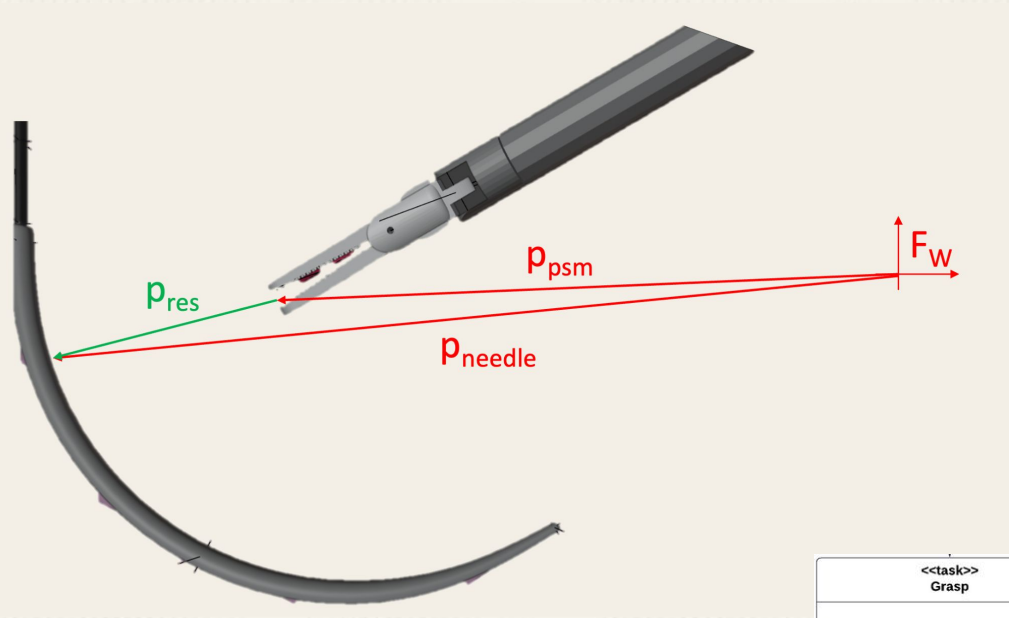
GRASP REWARD: DISTANCE

Minimize $|\mathbf{p}_{\text{res}}|$:

$$\mathbf{P}_{\text{res}} = \mathbf{P}_{\text{needle}} - \mathbf{P}_{\text{psm}}$$

$$\mathbf{p}_{\text{res}} = (x, y, z)$$

$$|\mathbf{p}_{\text{res}}| = \sqrt{x^2 + y^2 + z^2}$$



<task>
Grasp

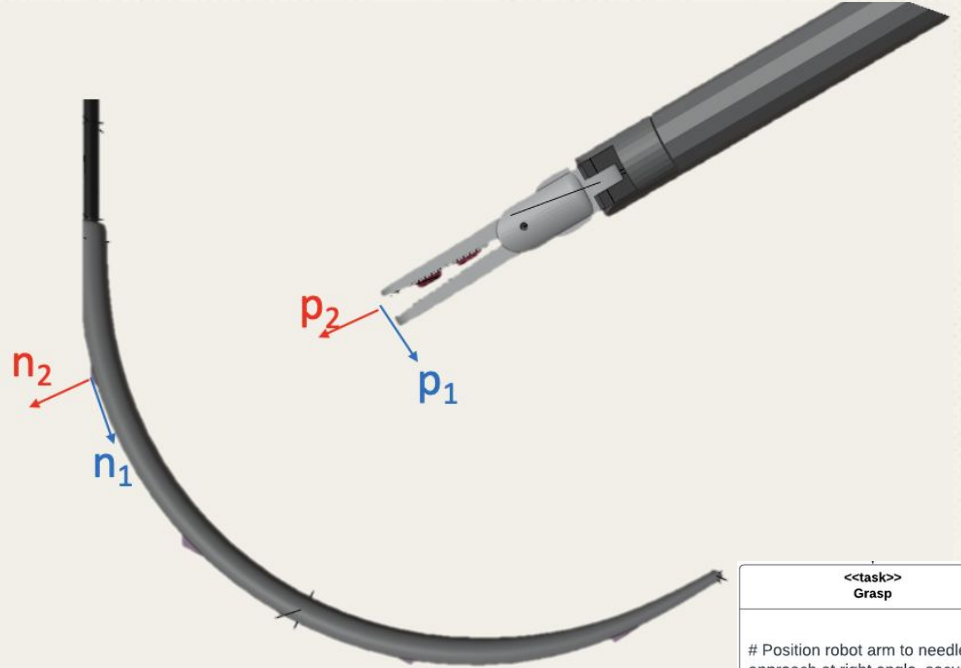
Position robot arm to needle,
approach at right angle, secure
needle to end effector

- reward(): minimize distance of
end effector and dot product of
needle and end effector

GRASP REWARD: ANGLE

$$p_1 \parallel n_1 \text{ \& } p_2 \parallel n_2:$$

$$\text{Minimize } p_1 \times n_1 + p_2 \times n_2$$



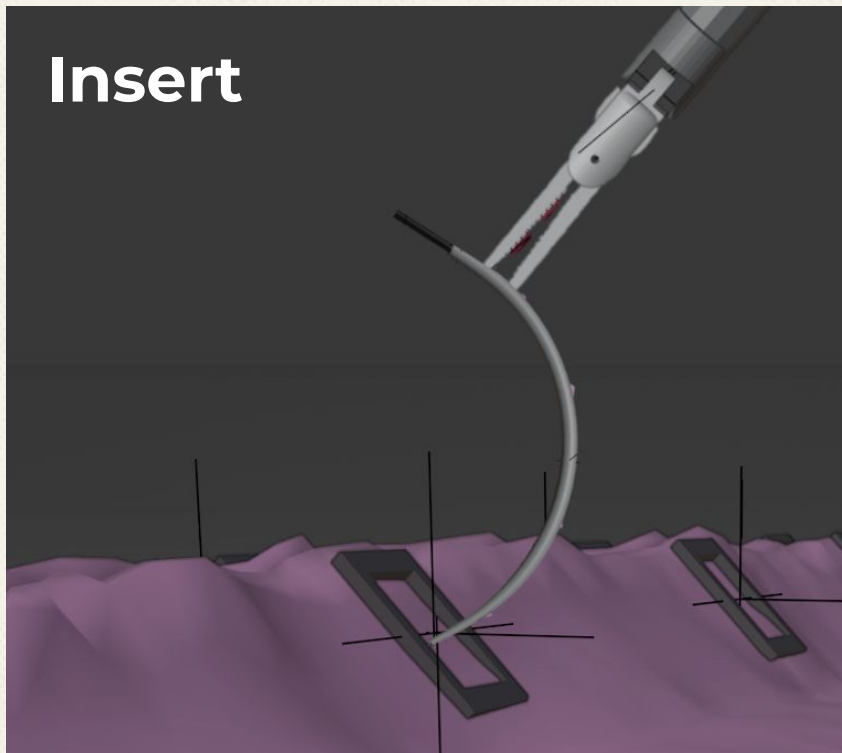
<task>
Grasp

Position robot arm to needle,
approach at right angle, secure
needle to end effector

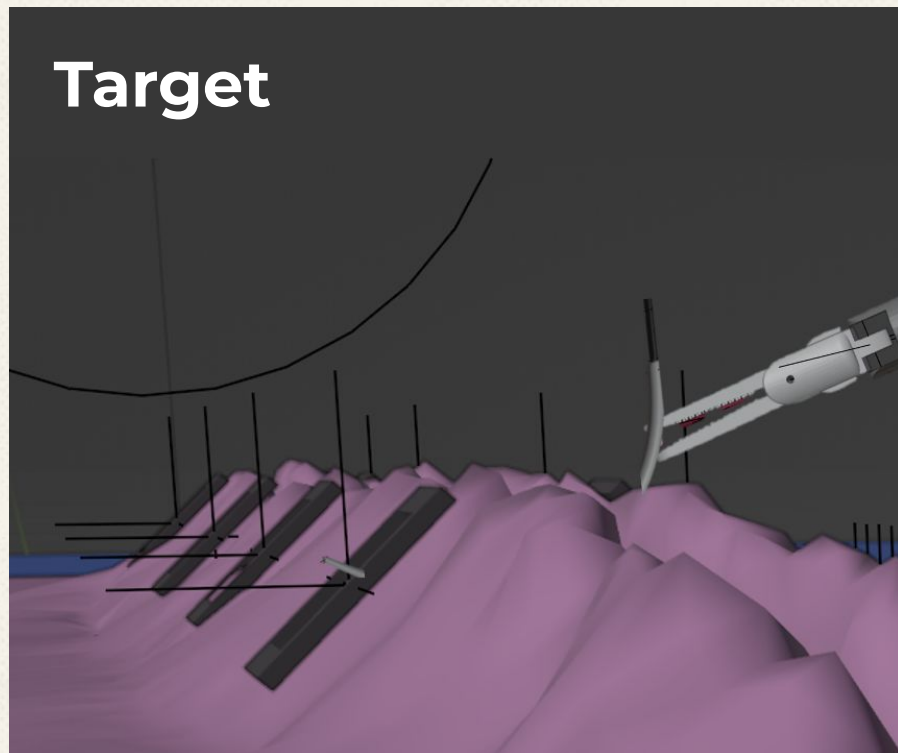
- reward(): minimize distance of
end effector and dot product of
needle and end effector

◦ INSERT & TARGET REWARD: IDEAL POSITION ◦

Insert



Target



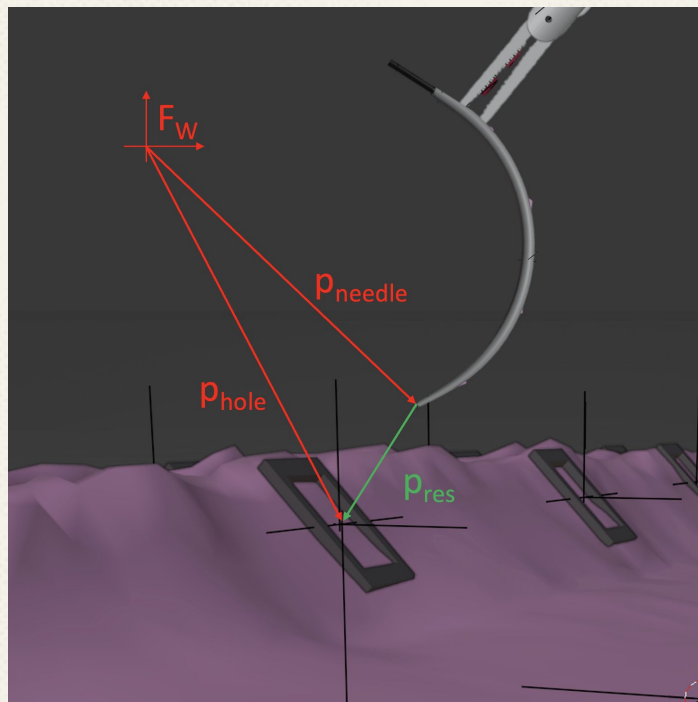
INSERT & TARGET REWARD: DISTANCE

Minimize $|\mathbf{p}_{res}|$:

$$\mathbf{p}_{res} = \mathbf{p}_{hole} - \mathbf{p}_{needle}$$

$$\mathbf{p}_{res} = (x, y, z)$$

$$|\mathbf{p}_{res}| = \sqrt{x^2 + y^2 + z^2}$$



<<task>>
Insert

Position the needle to the starting point and insert the needle perpendicular to the tissue surface

- reward(): minimize distance of end effector and insertion hole

<<task>>
Target

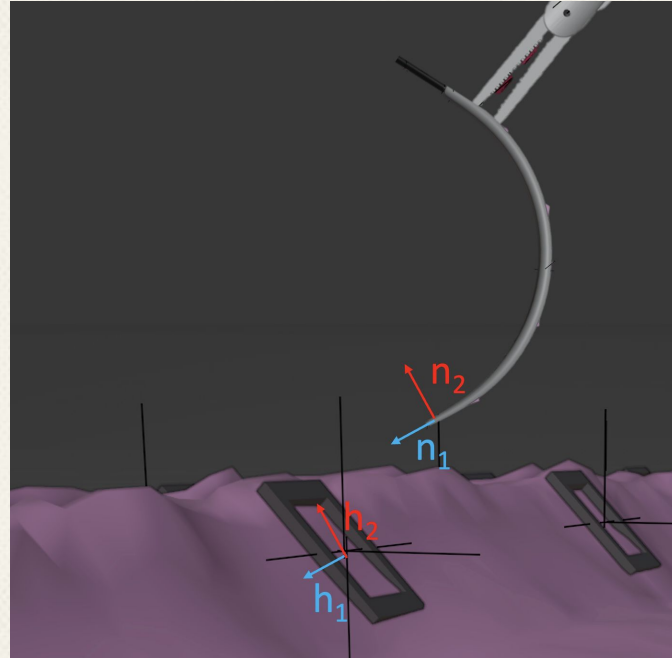
Guide needle from start to end point in tissue and pull needle out from tissue

- reward(): minimize distance of needle and exit hole

INSERT & TARGET REWARD: ANGLE

$$h_1 \parallel n_1 \text{ \& } h_2 \parallel n_2:$$

$$\text{Minimize } h_1 \times n_1 + h_2 \times n_2$$



<<task>>
Insert

Position the needle to the starting point and insert the needle perpendicular to the tissue surface

- reward(): minimize distance of end effector and insertion hole

<<task>>
Target

Guide needle from start to end point in tissue and pull needle out from tissue

- reward(): minimize distance of needle and exit hole



DOCUMENTATION

Software Design Specifications

Software Design Requirements

for

A reinforcement learning approach to robotic suturing

Group 20: Walee Attia (wattia1), Jocelyn Hsu (jhsu37), Jihoon Kim (jkim620)

Mentors: Dr. Anqi Liu, Dr. Adnan Munawar, Dr. Manish Sahu, Dr. Peter Kazanzides

April 12, 2023

1

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Project Scope	3
2	Surgical Robotics Challenge Environment	5
2.1	Challenge Requirements [1]	5
2.2	Functional Requirements	5
2.3	Design	5
2.4	Reward algorithm	8
3	Reinforcement Learning	11
3.1	Functional Requirements	11
3.2	Design	11
3.3	Implementation, Training, and Evaluation	13
4	Other	14
4.1	Instructions for running	14

2

Debugging Documentation

Debugging Documentation

for

A reinforcement learning approach to robotic suturing

Group 20: Walee Attia (wattia1), Jocelyn Hsu (jhsu37), Jihoon Kim (jkim620)

Mentors: Dr. Anqi Liu, Dr. Adnan Munawar, Dr. Manish Sahu, Dr. Peter Kazanzides

April 12, 2023

Debugging Issue

Issue	Resolution
Import errors with forked surgical robotics challenge	Changed SRC imports to backtrack to forked version; Developed req.py for installing all necessary modules
Very slow rendering of AMBF simulation	Turn off visualization when running ambf simulator executable with '-g 0' flag; Created simple render function that prints locations of each component
Compatibility of ROS / AMBF with Mac (specifically with Silicon chip)	Manually fix import statements, add https://raw.githubusercontent.com/torvalds/linux/master/arch/x86/kvm/cpuid.c to directory
Frame transformations between different coordinate frames (needle to world)	PSM base already in world coordinates; Use Blender software to click parts of components and get relationship within needle; Use Needle Kinematics for computation of frame transformations of needle to world



MANAGEMENT PLAN

PREVIOUS DELIVERABLES

Minimal: Open-source OpenAI Gym environment to train RL algorithms

- Transfer of SRC to OpenAI Gym environment (sandbox env)
- Core functionality for OpenAI Gym: `make()`, `reset()`, `step()`, `render()`
- Environments built to accomplish SRC Challenge #2: grasp needle and drive through tissue [4]
- Documentation of our work through entire development pipeline
- Compatibility with dVRK in the LCSR lab

Expected: Benchmarked performances of the SOTA RL algorithms

- Literature review of state-of-the-art (SOTA) RL Algorithms
- RL algorithm training for automated suturing task (SRC #2)
- Performance evaluation (accuracy and efficiency) of RL algorithms & 2021-2022 SRC winners

Maximum: Submission as NeurIPS (Datasets and Benchmark track), additional SRC tasks

- Paper for NeurIPS conference (database and benchmarking track), deadline TBA [6]
- Additional SRC challenges [4]:
 - Challenge 1: Find the needle
 - Challenge 3: Suture phantom

UPDATED DELIVERABLES

Minimal: Open-source OpenAI Gym environment to train RL algorithms

- Transfer of SRC to OpenAI Gym environment (sandbox env)
- Core functionality for OpenAI Gym: `make()`, `reset()`, `step()`, `render()`
- Environments built to accomplish SRC Challenge #2: grasp needle and drive through tissue [4]
- Documentation of our work through entire development pipeline
- Compatibility with dVRK in the LCSR lab

Expected: Benchmarked performances of the SOTA RL algorithms

- Literature review of state-of-the-art (SOTA) RL Algorithms
- RL algorithm training for automated suturing task (SRC #2)
- Performance evaluation (accuracy and efficiency) of RL algorithms & 2021-2022 SRC winners

Maximum: Submission as NeurIPS (Datasets and Benchmark track), additional SRC tasks

- Paper for NeurIPS conference (database and benchmarking track), deadline 6/1 for abstract, 6/7 for full paper [6]

TECHNICAL DEPENDENCIES

Dependency	Need	Source	Date Needed	Status	Contingency Plan
Swipe access to Robotarium, LCSR	Env Development	Dr. Adnan Munawar	3/1/2023	Completed	N/A
Access to dVRK systems at the Robotarium, LCSR	Test simulation environment	Dr. Adnan Munawar	3/15/2023	Completed	Linux Virtual Machine w/ AMBF + ROS
Rockfish GPU Access	Benchmarking	Dr. Anqi Liu	4/15/2023	Completed	Google cloud
SRC Winning Algorithms	Benchmarking	Dr. Adnan Munawar	4/15/2023	Completed	N/A

TIMELINE: MILESTONES + DEADLINES

Milestone	Output	Start Date	End Date	Status
Transfer SRC Environment to OpenAI Gym	GitHub Repo w/ MVP of environment	2/23/23	3/29/2023	Completed
Reward functions: Challenge #2 <ul style="list-style-type: none"> Reward functions for needle grasp + insert Test reward functions exhaustively 	Tested code	3/29/2023	4/26/2023	In-progress
Reward functions: Challenge #2 <ul style="list-style-type: none"> Reward functions for needle target Test reward functions exhaustively 	Tested code	4/5/2023	4/26/2023	In-progress
Benchmarking: Challenge #2 <ul style="list-style-type: none"> Trained RL algorithm for Challenge #2 Evaluate RL performance w/ SRC 2022 winners 	Model results Evaluation spreadsheet	4/12/2023	5/5/2023	In-progress
NeurIPS <ul style="list-style-type: none"> Paper submission for Dataset and Benchmark track 	Submitted paper	4/26/2023	6/1/2023 Abstract 6/7/2023 Full-paper	Not Started

TIMELINE: GANTT CHART

	Member	February				March				April				May				June	
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2
Preliminary Research																			
Set up OpenAI Gym, ROS, AMBF	All	█	█																
Lit review of SRC, AMBF	All	█	█	█	█														
RL online course	All	█	█	█	█														
Env Dev: Challenge 2																			
Code Documentation	All					█	█	█	█										
Transfer SRC env to OpenAI Gym	All				█	█	█	█	█										
Benchmarking: Challenge 2																			
Develop grasp reward function	Jihoon								█	█	█	█	█						
Develop insert reward function	Walee								█	█	█	█	█						
Develop target reward function	Jocelyn								█	█	█	█	█						
Benchmarking Documentation	All										█	█	█	█					
Lit review on SOTA RL models	All																		
Evaluate RL algorithm	All											█	█	█					
NeurIPS																			
Write paper	All														█	█	█	█	
Code cleanup for NeurIPS submission	All														█	█	█	█	

█ Completed

█ In-progress

█ Not started

MEETINGS

- Responsibilities:
 - Jocelyn: Development of SRCEnv
 - Jihoon: RL Pipeline
 - Walee: Reward function / RL
- Undergraduate meetings:
 - Weekly, Mondays & Fridays 11AM
- Team meetings:
 - Weekly, Wednesdays 9AM (w/ Dr. Liu & Dr. Munawar)
- Communication
 - Microsoft Teams, JHU email (mentors + students)
 - Discord (students)

REFERENCES

- [1] Introduction to reinforcement learning with David Silver. DeepMind. (n.d.). Retrieved February 19, 2023, from <https://www.deepmind.com/learning-resources/introduction-to-reinforcement-learning-with-david-silver>
- [2] Richter, F., Orosco, R. K., & Yip, M. C. (2019). Open-sourced reinforcement learning environments for surgical robotics. arXiv preprint arXiv:1903.02090.
- [3] Medical Open Network for Artificial Intelligence. MONAI. (n.d.). Retrieved February 19, 2023, from <https://monai.io/index.html>
- [4] 2021-2022 AccelNet Surgical Robotics Challenge (online). Collaborative Robotics Toolkit (CRTK). Retrieved February 19, 2023, from <https://collaborative-robotics.github.io/surgical-robotics-challenge/challenge-2021.html>
- [5] Gymnasium documentation. Basic Usage. (n.d.). Retrieved February 19, 2023, from https://gymnasium.farama.org/content/basic_usage/
- [6] NeurIPS 2023. Neural Information Processing Systems Foundation. (n.d.). Retrieved February 19, 2023, from <https://nips.cc/Conferences/2023>
- [7] A. Munawar, Y. Wang, R. Gondokaryono, and G. S. Fischer, "A real-time dynamic simulator and an associated front-end representation format for simulating complex robots and environments," in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, Nov. 2019. [Online]. Available: <https://doi.org/10.1109/iros40897.2019.89685688>
- [8] P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015. [Online]. Available: <https://arxiv.org/abs/1509.02971>
- [9] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," Advances in neural information processing systems, vol. 30, 2017
- [10] J. Weng, H. Chen, D. Yan, K. You, A. Duburcq, M. Zhang, Y. Su, H. Su, and J. Zhu, "Tianshou: A highly modularized deep reinforcement learning library," arXiv preprint arXiv:2107.14171, 2021. 9