

Capture of Free-Floating Targets Using End-to-End Reinforcement Learning: Theoretical Foundations and Ground Demonstrations

Bahador Beigomi

A Thesis Submitted to the Faculty of Graduate Studies
In Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

Graduate Program in Mechanical Engineering
York University
Toronto, Ontario

March 2025

©Bahador Beigomi, 2025

Abstract

This doctoral research is motivated by the need to autonomously capture a free-floating target using robotic manipulators for space debris removal. Space debris poses a growing threat to near-Earth operations, exacerbating the risk of collisions and jeopardizing vital satellite services. This dissertation addresses the challenge of autonomously grasping non-cooperative free-floating, tumbling debris in microgravity by leveraging Deep Reinforcement Learning and advanced robotic manipulation. A high-fidelity simulation environment, built with the PyBullet physics engine and enhanced through domain randomization, is developed to train and evaluate RL agents under realistic contact dynamics. The agents employ hierarchical control strategies—fusing curriculum-based learning for robust path planning and tactile sensor feedback for precise force regulation—ensuring minimal disturbance to noncooperative targets.

Extensive simulation studies compare multiple DRL algorithms, including Soft Actor-Critic, demonstrating success rates exceeding 90% in grasping scenarios with varying debris shapes and dynamic conditions. To validate these policies in operational settings, a Hardware-in-the-Loop testbed at York University is constructed, integrating dual Fanuc industrial robots with active gravity compensation and real-time visual tracking. Parallel experiments at the University of Luxembourg’s Zero-G Lab further corroborate the system’s adaptability in emulated six-degree-of-freedom motion.

Results reveal that the trained agents reliably align with, and capture, free-floating targets without inducing destabilizing forces. Final hardware tests confirm high grasp stability—even for spinning debris—underscoring the feasibility of on-orbit servicing through autonomous RL-driven control. This work advances the field of space robotics by combining novel DRL methodologies with realistic experimental verifications, ultimately guiding future missions toward safer, more efficient debris removal.

Contents

	Page
Abstract	ii
Contents	vi
List of Tables	vii
List of Figures	xiii
List of Abbreviations	xiv
List of Symbols	xvi
1 Introduction	1
1.1 Background	1
1.2 Justification of Research	3
1.3 Objectives	4
1.4 Methodology	5
1.5 Dissertation Outline	6
1.6 List of Publications	7
2 Literature Review	10
2.1 Space Debris Removal	10
2.2 Space Manipulator Ground-Based Facilities	14
2.3 Robotic Grasping	19
2.4 Reinforcement Learning	21
3 Problem Formulation	27
3.1 Background	27

3.1.1	Markov Decision Process	27
3.1.2	State, Observation, and Action Space	28
3.1.3	Exploration vs. Exploitation	29
3.1.4	Advantages for Robotics	29
3.2	Algorithm	30
3.2.1	Actor (Policy Network)	34
3.2.2	Critic (Q-Value Networks)	36
3.2.3	Replay Buffer	37
3.2.4	Temperature Parameter	37
3.2.5	Network Architecture and Hyperparameters	37
3.3	Curriculum Learning	39
3.4	Learning Environment	42
3.4.1	Target Shape Selection	44
3.5	Grasp Quality Metrics	45
3.6	Direct Capturing Method for Space Debris Removal	47
4	Simulation and Results	51
4.1	Comparative Analysis of Reinforcement Learning Algorithms for Robotics Control	51
4.1.1	Definition of Actions and Observations	52
4.1.2	Reward Function Design	53
4.1.3	Performance Evaluation and Algorithm Comparison	55
4.1.4	Conclusion	59
4.2	Domain Randomization in Reinforcement Learning for Pre-Grasp of Free-Floating Targets	61
4.2.1	Definition of Actions and States	61
4.2.2	Reward Shaping	63
4.2.3	Performance Evaluation	65
4.3	The Importance of Tactile Feedback in Pre-Capture Planning with Noisy Observations	72

4.3.1	Comparison of Agents with and without Tactile Feedback	72
4.3.2	Performance Under Noisy Observations and Robustness	77
4.4	Curriculum-based Reinforcement Learning for Pre-Grasp of Rotating Free-Floating Targets	80
4.4.1	Task Definition	80
4.4.2	States and Actions	81
4.4.3	Reward Shaping	81
4.4.4	Curriculum-based Learning Framework	82
4.4.5	Performance Evaluation	85
4.4.6	Conclusion	91
4.5	End-to-End Hierarchical Reinforcement Learning Framework for Grasping Free- Floating Space Debris	92
4.5.1	Domain Randomization	92
4.5.2	Task Definition	94
4.5.3	States and Actions	95
4.5.4	Reward Shaping	97
4.5.5	Training Results	97
4.5.6	Conclusion	105
5	Ground-Based Experimental Verification and Validation	106
5.1	York University Testing Facility	106
5.1.1	Computer Vision	110
5.1.2	Robotiq 3F Gripper	112
5.1.3	Gravity Compensation	116
5.1.4	Grasping Task Demonstration	119
5.1.5	Trained Agent Implementation	122
5.2	University of Luxembourg Testing Facility	125
5.2.1	Curriculum-Based Agent Implementation	128

6 Conclusion	134
6.1 Summary of Key Conclusions	134
6.2 Original Contributions to the Field	135
6.3 Future Research Directions	136
Bibliography	138

List of Tables

2.1	Qualitative Comparison of Key Debris Removal Approaches	12
2.2	Summary of Representative RL-Based Grasping Studies	25
3.1	Summary of SAC Hyperparameters	38
4.1	Performance Metrics at Varying Maximum Relative Velocities	70
4.2	Tactile-sensing agent performance under increasing levels of Gaussian noise. . . .	78
4.3	Curriculum Learning Levels	82
4.4	Comparison of target orientation and angular velocity in each curriculum level at the start and end of an episode.	83
4.5	Domain Randomization parameters	86
4.6	Complexity Level Training Results	88
4.7	Domain Randomization Parameters	93
5.1	Denavit–Hartenberg Parameters for Fanuc M-20iD/25 Industrial Robot	108

List of Figures

2.1	Number of tracked objects in Earth orbit over time	11
2.2	Cumulative Number of Possible Collisions Predicted Over the Upcoming Years . .	11
2.3	Direct Capturing Method Components	13
2.4	Examples of ground-based facilities.	16
2.5	Further advanced facilities for on-orbit servicing and debris removal research. . . .	18
3.1	Basic reinforcement learning framework	30
3.2	Flowchart categorizing Reinforcement Learning algorithms	33
3.3	Training workflow of SAC algorithm	39
3.4	PyBullet simulation environment for grasping free-floating targets	42
3.5	Procedure for each episode during the training	44
3.6	Schematic representation of the free-floating base with a serial-link manipulator. . .	48
4.1	Illustration of the topology-based reward r_{top} , where target key points (shown in red dots) are evaluated against the gripper's convex hull (generated by blue lines). . .	54
4.2	Learning curves of DDPG, TD3, and SAC trained with (a) Reward function R_1 . (b) Reward function R_2 . Curves represent the moving average of episode rewards over training timesteps.	57
4.3	Progressive policy improvement for a free-floating target grasp scenario. (1) The agent initially prioritizes distance minimization; (2) orientation alignment is then introduced; (3) final strategies incorporate topological constraints to ensure a robust pre-grasp.	59
4.4	Dense reward components for (a) position alignment R_d and (b) orientation alignment R_θ . The rewards decrease monotonically as the gripper deviates from the target in position or orientation.	64

4.5	Graphical representation of the contact penalty term P_F . The penalty is applied when any contact force is detected by the gripper’s tactile sensors, thereby discouraging excessive force during pre-grasp.	65
4.6	Training results in simulation environment. Mean evaluation results of the best-trained agent over five random episodes.	67
4.7	Evaluation results of the best-trained agent. The term “closest distance” refers to the minimum distance between the target and the gripper’s palm, while “position” indicates the location of their center of mass. The gripper successfully positions itself in the pre-grasp stance without making contact with the free-floating moving target, and it aligns its velocity accordingly.	68
4.8	Evaluation results of the best-trained agent. Although there is slight contact between the target and the gripper’s finger, the agent manages this effectively.	69
4.9	Accumulative moving average of rewards over the training episodes for both tactile-sensing and non-tactile agents. The tactile-based agent exploits contact feedback to more accurately adjust its pose, leading to a higher overall performance and faster convergence.	73
4.10	Success rate comparison over the training process for two agents: one equipped with tactile sensor feedback and another lacking it.	74
4.11	Comparative snapshots of pre-grasping maneuvers. Each column illustrates an agent’s behavior under two different scenarios. The tactile-sensing agent effectively positions itself relative to the free-floating target, reducing unwanted collisions. The non-tactile agent either pushes the object away or stays too distant, compromising task success.	75
4.12	Random episode outcomes for each trained agent. The tactile-sensing agent attains near-maximum rewards consistently, verifying its ability to safely align with the target.	76

4.13	Random episode outcomes for each trained agent. The non-tactile agent occasionally reaches high reward levels but inevitably contacts the target, causing it to separate and leading to task failure.	77
4.14	Distribution of 1,000 Gaussian noise samples with zero mean and different standard deviations: (a) 0.01 and (b) 0.02. These noise levels represent realistic sensor noise, shifting the agent’s view of the target, challenging accurate trajectory planning. . .	78
4.15	Tactile-sensing agent’s trajectory during a representative noisy episode. Despite receiving noisy target states and consequently exhibiting momentary deviations, the agent consistently recovers and maintains a reliable pre-grasp configuration, demonstrating robust performance in uncertain environments.	79
4.16	Effect of CL on agent outcomes: Comparative analysis of Mean Reward and Success Rate over Timesteps t . Incorporating CL yields faster convergence and higher final performance.	87
4.17	Analysis of Gripper Control Actions, Velocity, and Reward Distributions Over 100 Random Episodes.	89
4.18	Snapshots of the gripper’s approach toward a moving and rotating target for two randomly selected episodes.	90
4.19	Hierarchical two-stage framework for grasping a free-floating, moving target in microgravity. The positioning agent focuses on alignment, whereas the grasping agent ensures robust closure and force regulation.	95
4.20	Training results in simulation. Mean evaluation performance over 5 random episodes at each checkpoint. The training converges around 6 million timesteps with a success rate approaching 0.92.	99
4.21	Distributions from 100 random episodes controlled by the Two-Stage RL-Trained agent.	100

4.22	Episode performance under Low Friction condition. Each subplot depicts the agent's state and action trajectories, highlighting the application of higher contact forces to mitigate slippage on an extremely slippery surface.	102
4.23	Episode performance under High Friction condition. Each subplot visualizes the agent's ability to exploit higher friction for reducing normal force after establishing stable contact, thereby optimizing reward by avoiding excessive force application. .	103
4.24	Snapshots from two evaluation episodes under Low and High Friction conditions, illustrating the agent's approach strategy and securement of the free-floating, moving target.	104
5.1	York University Testing Facility setup demonstrating the interaction between the two Fanuc robots.	107
5.2	Illustrative example of the AI-based computer vision (YOLOX) tracking the target's key features, nozzle in this case, in real time with confidence level.	111
5.3	Position tracking error of YOLOX under different translations of the target satellite. As the target moves off-axis, estimation skewness increases the location error. . .	112
5.4	Robotiq 3F gripper with independently actuated proximal finger joints and mechanical linkages for the distal joints.	113
5.5	State machine model for the Robotiq 3F gripper digital twin, enabling accurate replication of mechanical linkages and contact events.	114
5.6	Comparison of real and simulated Robotiq 3F gripper finger motions, illustrating high-fidelity replication of mechanical linkages and contact events.	115
5.7	Force sensing setup with RP-C18.3-ST thin-film pressure sensors on the Robotiq 3F gripper fingertips, enabling real-time measurement of contact forces.	116
5.8	Illustration of the ATI sensor mounted on the end-effector and the local frame definitions.	118

5.9	Snapshots demonstrating the dummy satellite’s motion in response to external forces. The robot applies gravity compensation, enabling the satellite to behave as if in a microgravity environment.	119
5.10	Trajectories of the dummy satellite in (a) joint space and (b) Cartesian space, demonstrating the robot’s ability to replicate the target’s tumbling motion.	120
5.11	Chaser robot arm trajectories in (a) joint space and (b) Cartesian space, aligning the Robotiq 3F gripper with the target satellite for grasping.	121
5.12	Snapshots of the grasping task demonstration, showcasing the coordinated motion of the chaser robot and the target satellite.	121
5.13	Overall structure of the HIL connection. Each component serves a distinct role in achieving precise control over the gripper’s motion.	123
5.14	Movement of the Target and Gripper in a Real-world Setting: Upon positioning the target at random location, the gripper aligns itself to a pre-grasp position.	124
5.15	Camera field of view illustrating the nozzle of the target satellite during the grasping task.	125
5.16	University of Luxembourg Zero-G Lab facility, showcasing the rail-mounted robotic arms, motion capture system, and illumination setup.	126
5.17	Zero-G Lab control framework, illustrating the real-time communication and control loop for the robotic arms and end-effectors.	128
5.18	OptiTrack motion capture system setup for tracking the target satellite in the Zero-G Lab.	129
5.19	Performance analysis of gripper control during a positioning task in a dynamic real-world scenario.	131
5.20	Sequential snapshots of two episodes in real-world experiment, showing its controlled approach from a random orientation to a moving, randomly positioned target. The final alignment showcases the agent’s precision in achieving a reliable pre-grasp pose under realistic conditions.	132

5.21	An example of the agent's failure in a real-world scenario. The gripper fails to align with the moving target and makes contact with it.	133
------	--	-----

List of Abbreviations

Abbreviation	Description
ADR	Active Debris Removal
AI	Artificial Intelligence
API	Application Programming Interface
ATI	ATI Industrial Automation (Force/Torque Sensor Brand)
CL	Curriculum Learning
CNN	Convolutional Neural Network
CSA	Canadian Space Agency
DDPG	Deep Deterministic Policy Gradient
DH	Denavit–Hartenberg
DoF	Degrees of Freedom
DRL	Deep Reinforcement Learning
DQN	Deep Q-Network
FANUC	Fuji Automatic Numerical Control (Robot Brand)
FK	Forward Kinematics
FT	Force-Torque
GEO	Geostationary Orbit
GJM	Generalized Jacobian Matrix
GPU	Graphics Processing Unit
GWS	Grasp Wrench Space
HIL	Hardware-in-the-Loop
HRL	Hierarchical Reinforcement Learning
IK	Inverse Kinematics
LAR	Launch Adapter Ring

LEO	Low Earth Orbit
LLM	Large Language Model
MCS	Motion Capture System
MDP	Markov Decision Process
MLP	Multi-Layer Perceptron
OOS	On-Orbit Servicing
PPO	Proximal Policy Optimization
PyBullet	Python-based Bullet Physics Engine
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
ROS	Robot Operating System
RSO	Resident Space Object
SAC	Soft Actor-Critic
SB3	Stable Baselines3
Sim2Real	Simulation-to-Reality Transfer
TCP	Transmission Control Protocol
TD3	Twin Delayed Deep Deterministic Policy Gradient
UDP	User Datagram Protocol
UR	Universal Robots (Robot Arm Brand)
YOLOX	You Only Look Once eXtended

List of Symbols

Symbol	Description
a_i	Link length in DH parameters (m)
a_ϕ, a_θ, a_ψ	Rotational actions (0.1 rad)
a_x, a_y, a_z	Translational actions (0.01 m)
α_i	Link twist in DH parameters (rad)
δ_p, δ_v	Relative pose, relative velocity (m, m/s)
d	Minimum distance between target and gripper's palm (m)
d_i	Link offset in DH parameters (m)
f_c	Closing force during approach (N)
f_g	Sustained grasping force (N)
$\dot{\theta}_i$	Joint velocity for joint i (rad/s)
$\omega_1, \omega_2, \omega_3$	Weighting gains for reward components (dimensionless)
p_v	Penalty for excessive target velocity (dimensionless)
r_d	Euclidean distance between gripper and target (m)
r_θ	Orientation difference (rad)
r_{top}	Topology-based coverage of key points in gripper hull (dimensionless)
θ_i	Joint angle for joint i (rad)
θ_{LM}	Joint angles (or configuration) from Levenberg–Marquardt solver (rad)
v_c	Finger closing velocity (mm/s)
v_{biased}	Biased sampling value in CL (dimensionless)
$\mathbf{g}$	Gravitational acceleration vector (m/s^2)
$\mathbf{r}$	Vector from sensor frame to target's center of mass (m)
τ_{ext}	External torque measured by FT sensor (Nm)
τ_g	Gravitational torque: $\tau_g = \mathbf{r} \times \mathbf{F}_g$ (Nm)

τ_{net}	Net external torque after gravity subtraction (Nm)
ω_0	Base's angular velocity (rad/s)
ω_e	End-effector's angular velocity (rad/s)
n_{in}	# of target key points inside the gripper hull (dimensionless)
$\odot$	Elementwise (Hadamard) product
$\mathbf{A}_i$	Homogeneous transform for joint i
$\mathbf{A}_e$	End-effector's orientation (rotation matrix)
$\mathbf{F}_{\text{ext}}$	External force measured by FT sensor (N)
$\mathbf{F}_g$	Gravitational force: $\mathbf{F}_g = m \mathbf{g}$ (N)
$\mathbf{F}_{\text{net}}$	Net external force after gravity subtraction (N)
F_c	Total contact force from tactile sensors (N)
G_p, G_v	Pose, velocity of the gripper (pose in m, velocity in m/s)
H_b	Inertia matrix of the base $\star$ ($\text{kg}\cdot\text{m}^2$)
$\mathbf{H}_{bm}$	Coupling inertia matrix for base and manipulator ($\text{kg}\cdot\text{m}^2$)
$\mathbf{J}_b, \mathbf{J}_m$	Base and manipulator Jacobians, respectively
$\mathbf{J}_g$	Generalized Jacobian matrix
$\mathbf{P}_e, \mathbf{A}_e$	End-effector's position and orientation (m, rotation matrix)
$\mathbf{S}$	State space vector
$\mathbf{v}_0, \omega_0$	Base's linear, angular velocity (m/s, rad/s)
$\mathbf{v}_e, \omega_e$	End-effector's linear, angular velocity (m/s, rad/s)
L_i	Current curriculum level (dimensionless)
R	Combined reward function (dimensionless)
R_1, R_2	Reward functions
R_{dist}	Distance minimization reward (dimensionless)
R_{F_N}	Normal force regulation reward (dimensionless)
R_{top}	Sparse topology reward (dimensionless)
R_{V_T}	Velocity damping reward (dimensionless)

R_e	Cumulative episode reward (dimensionless)
R_t	Reward at timestep t (dimensionless)
S_{thresh}	Success threshold for level progression (dimensionless)
$\bar{S}_{\text{eval}}$	Avg. success rate over last 10 evals (dimensionless)
T	Transformation matrix from base to end-effector
T_p, T_v	Pose, velocity of the target (pose in m, velocity in m/s)
T_{max}	Max steps per episode (dimensionless)

Chapter 1 Introduction

In this chapter, I explore the growing challenge of space debris and the essential role of robotic manipulators in mitigating its impact. I provide a comprehensive background on the increasing number of satellites and the resulting accumulation of orbital debris, emphasizing the risks they pose to current and future space missions. This chapter justifies the need for advanced control methodologies and outlines the research objectives focused on developing robust reinforcement learning-based control frameworks for autonomous debris capture. Additionally, I present an overview of the dissertation structure and highlight relevant publications that have contributed to this field of study.

1.1 Background

Over the past few decades, the proliferation of satellites and orbital missions has transformed space into a critical infrastructure for communication, Earth observation, navigation, and scientific research. Once the domain of a few government-led ventures, the near-Earth environment now hosts satellites operated by commercial entities, academic institutions, and international agencies. This intense activity accelerates debris accumulation, as defunct satellites and fragments from collisions persist in orbit. According to estimates, as of January 2021, approximately 6,542 satellites circle Earth, of which nearly 3,170 are inactive and thus classified as space debris [1, 2].

The first major satellite collision—occurred in 2009—produced large amounts of high-speed debris [3], highlighting the danger of collisions and the Kessler syndrome, wherein each collision begets further debris [4]. Residual propellants in retired rocket stages or further impacts exacerbate the hazard by generating additional fragments. In response, space debris mitigation—particularly via on-orbit servicing (OOS)—has become a focal research area. Space robots equipped with

advanced manipulators and autonomy are viewed as key enablers of tasks ranging from assembly and refueling to active debris removal [5]. Despite the fact that Kessler syndrome articulates a runaway scenario of exponentially increasing space debris, the immediate effects are also alarming. Even a single high-velocity collision can disrupt vital communication satellites or generate thousands of new fragments, risking further chain reactions. The financial costs of replacing damaged assets, mitigating collisions, and ensuring crew safety on stations like the ISS can exceed hundreds of millions of dollars per event [6, 7]. Consequently, strengthening debris removal efforts has become not only a matter of long-term orbital sustainability but also an urgent economic imperative.

Although no full-scale Active Debris Removal (ADR) mission has yet been implemented, significant research addresses autonomous debris capture and deorbit. Approaches range from passive methods (end-of-life de-orbit mechanisms, drag devices) to active retrieval using specialized servicers [8]. However, many inactive satellites remain in low-Earth orbit (LEO), where human, commercial, and defense activities make debris removal crucial. Physical capture poses formidable challenges, as many targets are noncooperative, tumbling, and lack grappling fixtures, making chaser and target attitude control difficult [9, 10, 11].

Robotic manipulators have thus emerged as a promising solution for on-orbit applications such as in-space assembly, refueling, and debris removal [12]. Going beyond traditional control methods, artificial intelligence (AI) techniques, including machine learning (ML) and reinforcement learning (RL), are increasingly advocated for space manipulation tasks [13, 14]. As missions venture farther from Earth and orbital traffic intensifies, autonomous navigation and manipulation in dynamic microgravity environments become indispensable.

This dissertation advances control methodologies and includes experimental validations for a robotic manipulator dedicated to capturing a free-floating, moving, and tumbling targets in microgravity conditions. The system introduced here is designed to handle a range of debris sizes and does not rely on target self-removal features. Closer progress toward fully autonomous ADR missions that safeguard near-Earth space operations from escalating debris threats is being made by devising robust control architectures that adapt to uncertain target motion and unsteady contact

dynamics.

Although fully autonomous ADR missions remain forthcoming, the progress described represents a key step in bridging theoretical research and on-orbit demonstrations. I strive to protect near-Earth space activities for future generations by developing reliable, AI-driven manipulation systems capable of detecting, tracking, capturing, and deorbiting space debris.

1.2 Justification of Research

Space robotic manipulators are pivotal for advancing missions such as on-orbit servicing, in-space assembly, and the critical task of space debris removal. As space becomes increasingly congested with debris, the risks to operational spacecraft and future missions escalate, underscoring the urgent need for effective debris capture and mitigation strategies. However, controlling these manipulators in the unique environment of space presents significant challenges, including limited communication bandwidth, the necessity for autonomous decision-making, stringent safety and collision avoidance requirements, and constraints on power and computational resources.

Despite notable progress in ADR, most existing solutions focus on either simplified on-orbit dynamics or assume standardized grapple fixtures. However, many debris objects in Low Earth Orbit (LEO) and beyond are severely tumbling, lack cooperative docking ports, or exhibit uncertain inertial properties. This dissertation directly tackles these gaps by introducing reinforcement-learning-based control policies capable of robustly handling free-floating, rotating debris with minimal human intervention.

Current research in this field often relies on simplified simulation environments that do not fully capture the complexities of microgravity and dynamic uncertainties inherent to space operations. Additionally, many existing control algorithms are optimized for single tasks or specific scenarios, lacking the versatility required to handle diverse and unpredictable conditions encountered in real-world applications. The integration of artificial intelligence into control strategies remains limited, particularly beyond stationary, three-degree-of-freedom (3-DoF) tabletop setups, which

restricts the adaptability and robustness of robotic manipulators in more dynamic environments.

Moreover, experimental validation in space is prohibitively expensive and logistically challenging, making ground-based testbeds that accurately emulate space conditions indispensable. In this research, I address these limitations by developing advanced RL frameworks tailored for robotic grasping in simulated microgravity environments. By utilizing state-of-the-art testing facilities—the York University Testing Facility and the University of Luxembourg Zero-G Lab—I systematically evaluate the performance, robustness, and real-world applicability of trained agents under conditions that closely mimic actual space scenarios. Through comprehensive experimentation and validation, this dissertation enhances the autonomy and reliability of space robotic manipulators, contributing to safer and more sustainable space operations.

1.3 Objectives

This dissertation explores the integration of Deep Reinforcement Learning (DRL) to overcome the challenges associated with grasp synthesis for three-finger grippers, specifically targeting the manipulation of free-floating, moving, and tumbling objects such as space debris. My primary aim is to develop an autonomous grasping framework that harnesses the agility of three-finger grippers to achieve high adaptability across a wide array of grasping scenarios. Unlike traditional techniques that rely heavily on object-specific programming, DRL facilitates the training of robotic systems capable of managing diverse objects with minimal initial configuration.

A key component of this research is the application of the Soft Actor-Critic (SAC) algorithm, a DRL approach optimized for environments with complex, high-dimensional state spaces characteristic of robotic grasping tasks. The SAC algorithm enhances training efficiency by incorporating an entropy term, which encourages exploratory behavior and enables the discovery of varied grasping strategies, thereby increasing the system’s resilience to environmental variances.

The central research question guiding my study is:

How can a DRL-based control system be developed to ensure optimal and reliable grasping of

free-floating, moving, and tumbling targets in microgravity environments?

To address this question, the research is structured around the following specific objectives:

- **Establish a High-Fidelity Microgravity Simulation Environment:** Develop a simulation platform that accurately mirrors the dynamics of microgravity, allowing for the effective training of DRL agents that can be directly applied to real-world space operations.
- **Create an Adaptive Path-Planning Strategy for Pre Capture:** Design a versatile path-planning methodology that ensures consistent and flexible performance during the pre capture phase, adaptable to various scenarios and environmental conditions.
- **Develop a Robust Force Application Policy for Post Capture:** Formulate a precise policy governing the application of force by the gripper’s fingers to moving targets, ensuring secure and stable grasps during the post capture phase.
- **Implement a Comprehensive Hardware-in-the-Loop (HIL) Validation System:** Construct an extensive HIL system to simulate the actions of the trained agent, enabling thorough validation of its performance under real-world operational conditions.

These objectives collectively aim to advance the state of the art in autonomous grasping technologies, particularly within the context of space debris removal missions. By merging sophisticated DRL techniques with realistic simulation environments and rigorous validation processes, I aspire to contribute significantly to the development of more efficient and dependable robotic systems capable of executing complex grasping tasks in challenging environments.

1.4 Methodology

In this dissertation, I leverage RL as the core control methodology to enable robots to autonomously learn grasping strategies for free-floating space debris. Unlike traditional control techniques that often rely on explicit models and carefully tuned controllers, RL agents learn by interacting with the environment—choosing actions, observing outcomes, and receiving rewards based on performance.

At its heart, RL formalizes decision-making as a Markov Decision Process (MDP). The agent observes the environment’s *state*, takes an *action*, and then transitions to a new state while receiving a numerical *reward*. Over many episodes of trial-and-error, the agent refines a *policy* that maximizes the expected cumulative reward. This iterative learning approach is especially attractive for space debris capture, where unpredictable target motion and complex contact dynamics make hand-designed solutions challenging.

A key design choice in RL is how the reward is computed. In *dense* reward settings, the agent receives frequent feedback reflecting how well it is performing at each timestep (e.g., penalizing target velocity, rewarding reduced distance to the target). In *sparse* reward scenarios, feedback might be withheld until a successful grasp is achieved, requiring more extensive exploration. In my approach, I combine both paradigms: distance-based metrics guide the agent’s incremental improvements, while binary bonuses ensure decisive, collision-free grasps.

Building on this reward framework, I adopt DRL algorithms—particularly actor-critic methods—to handle the continuous control actions required for 6-Dof motion. By encoding the robot’s pose, velocities, and tactile information into the agent’s observation space, the policy can learn to optimize positioning, orientation, and grasp stability in microgravity, reducing manual tuning of contact forces.

1.5 Dissertation Outline

This dissertation is organized into six chapters, each addressing pivotal components of space debris mitigation through advanced robotic methodologies and RL in microgravity environments. The chapters are structured as follows:

Chapter 1 introduces the research context, highlighting the significance of space debris and the necessity for robotic solutions. It outlines the research objectives, motivations, and provides an overview of the dissertation structure.

Chapter 2 reviews existing space debris management strategies, differentiating between

passive and active removal methods. It assesses current robotic systems for grasping and RL applications, identifying gaps and areas for further investigation.

Chapter 3 formulates the research problem by defining the objectives, state and action spaces, and performance metrics relevant to RL in microgravity. It presents the mathematical models and algorithms used, focusing on the MDP framework and the advantages of these models for robotic debris manipulation.

Chapter 4 describes the development of a high-fidelity simulation environment for robotic control in microgravity. It compares various RL algorithms for grasping tasks, utilizing domain randomization and curriculum-based learning to improve robustness and performance, and finally proposes an end-to-end hierarchical reinforcement learning (HRL) framework to grasp a free-floating moving target. The chapter evaluates the simulation results, demonstrating the effectiveness of the proposed methods.

Chapter 5 shifts to practical applications by detailing the hardware-in-the-loop (HIL) framework that simulates microgravity conditions. It covers the experimental setups at York University and the University of Luxembourg, including the integration of computer vision, Robotiq 3F grippers, and gravity compensation systems. This chapter validates the control policies through testing on industrial robotic platforms, highlighting the system’s adaptability and robustness in real-world scenarios.

Chapter 6 concludes the dissertation by summarizing the key findings and contributions, such as the simulation environment, path-planning strategies, and reliable grasping agents. It discusses the impact of the HIL system and proposes future research directions, including multi-agent collaboration, adaptive grasping techniques, and the integration of large language models for enhanced task planning.

1.6 List of Publications

Below is the list of all conference and peer-reviewed journal publications related to this dissertation.

Journal Publications

1. Beigomi, Bahador, Olivares-Mendez, Miguel, Mohatahem Reyaz Makhdoomi, Carol Martinez Luna, and Zheng Hong Zhu. "Curriculum-based Reinforcement Learning for 6-DoF Pre-Grasp of Floating Moving Targets" Submitted to IEEE Transaction on Automation Science and Engineering 2025. Submission No. T-ASE-2025-948
2. Beigomi, Bahador, Shi, Jian-Feng, and Zheng Hong Zhu. "An End-to-End Two-Stage Reinforcement Learning Framework for Grasping Floating Moving Space Debris in Microgravity" Submitted to Aerospace Science and Technology 2025. Submission No. AESCTE-D-25-00041.
3. Beigomi, Bahador, Al Ali, Ahmad, and Zheng Hong Zhu. "Development of 6-DoF Hardware-in-the-Loop Ground Testbed for Autonomous Robotic Space Debris Removal", Aerospace, 11(11), p.877. 2024
4. Beigomi, Bahador, and Zheng Hong Zhu. "Utilizing Deep Reinforcement Learning for Tactile-Based Autonomous Capture of Non-Cooperative Objects in Space", Aerospace Systems, 1–10. 2023

Conference Publications

1. Beigomi, Bahador, and Zheng Hong Zhu. "Improving Soft-Capture Phase Success in Space Debris Removal Missions: Leveraging Deep Reinforcement Learning and Tactile Feedback", IEEE 20th International Conference on Automation Science and Engineering (CASE) 2024
2. Beigomi, Bahador, and Zheng Hong Zhu. "Towards Real-World Efficiency: Domain Randomization in Reinforcement Learning for Pre-Capture of Free-Floating Moving Targets by Autonomous Robots", IEEE International Conference on Robotics and Automation (ICRA) 2024

3. Beigomi, Bahador, and Zheng Hong Zhu. “Adapting to Inaccurate Observation Data with SAC-Based Path-Planning Control of Gripper in Micro-Gravity”, Canadian Society for Mechanical Engineering International Congress (CSME), Toronto, Canada. May 26–29, 2024
4. Beigomi, Bahador, and Zheng Hong Zhu. ”Enhancing Robotic Grasping of Free-Floating Targets with Soft Actor-Critic Algorithm and Tactile Sensors: a Focus on the Pre-Grasp Stage.” AIAA SCITECH 2024 Forum. 2024.
5. Beigomi, Bahador, and Zheng Hong Zhu. “Deep Reinforcement Learning for Robotic Grasping with Tactile Sensor Feedback”, Canadian Society for Mechanical Engineering International Congress (CSME), Sherbrooke, Canada. May 28–31, 2023
6. Beigomi, Bahador, and Zheng Hong Zhu. “Artificial Intelligent Tactile Feedback Control for Autonomous Robotic Capture of Non-Cooperative Space Target”, AAS/AIAA Space Flight Mechanics Meeting 2023
7. Beigomi, Bahador, and Zheng Hong Zhu. “A Comprehensive Review of Space Manipulator Techniques for Recognizing Objects with Unknown Attributes”, Canadian Society for Mechanical Engineering International Congress (CSME), online, June 27–30, 2021

Chapter 2 Literature Review

This chapter presents a comprehensive overview of the existing research related to autonomous robotic grasping for space debris removal. It begins by addressing the escalating issue of space debris and the critical need for effective mitigation strategies to protect orbital operations. The review then examines the various ground-based facilities and emulation environments that are essential for developing and testing space manipulators under conditions that mimic microgravity. Following this, the chapter explores advancements in robotic grasping technologies, highlighting both traditional techniques and emerging data-driven approaches. Finally, it delves into the role of Reinforcement Learning (RL) in enhancing the adaptability and robustness of robotic systems, particularly in dynamic and uncertain environments.

2.1 Space Debris Removal

Since the launch of the first satellite in 1957, the accumulation of space debris has become a critical challenge for satellite operations and space exploration. Close to around 70 years of activities have placed numerous satellites and rocket bodies into Earth’s orbit. In 2018, there were almost 3,000 dead satellites still circling the planet, contributing to a total mass of over 8,000 tons of debris [15]. While some have naturally re-entered the atmosphere, thousands of inactive satellites, spent rocket stages, and fragments from collisions remain on-orbit, posing substantial threats to operational spacecraft. As illustrated in Fig. 2.1 [16], the number of tracked objects in Earth’s orbit has now exceeded 30,000, exacerbating concerns for space traffic management and debris mitigation [17]. According to NASA guidelines, operators are encouraged to either de-orbit satellites within 25 years or move them to a higher “graveyard orbit,” but compliance is inconsistent [18]. In fact, removing 5 to 10 debris objects annually is deemed necessary to stabilize the orbital environment and curtail

the risk of catastrophic collisions [19].

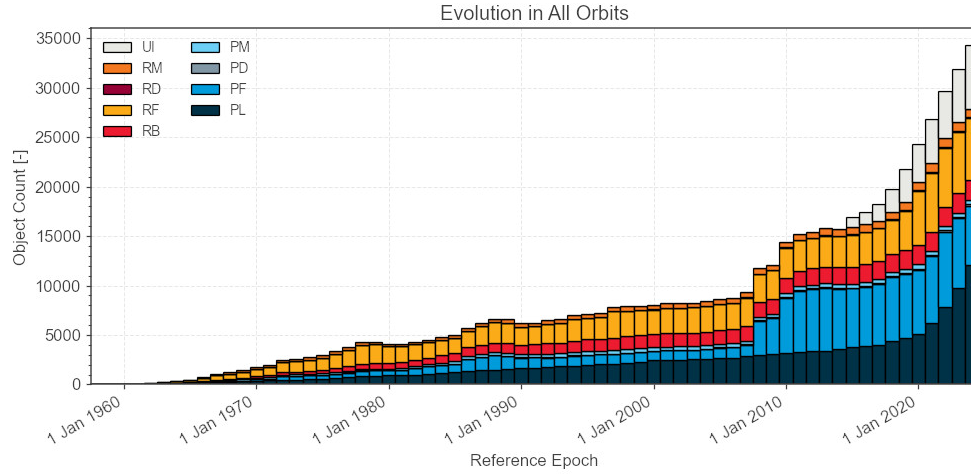


Figure 2.1: Number of tracked objects in Earth orbit over time

Fig. 2.2 (adapted from ESA's Space Environment Report 2023 [16]) underscores the potential for frequent destructive collisions if the debris population continues to expand. Projections indicate that even with an immediate halt to all new launches—an impractical scenario—hundreds of high-velocity collisions may still occur in the coming decades. Likewise, analyses show that collision probabilities for debris exceeding 1 cm in diameter have surged by over 50% in recent years [20, 21], highlighting the urgency of more effective mitigation strategies.

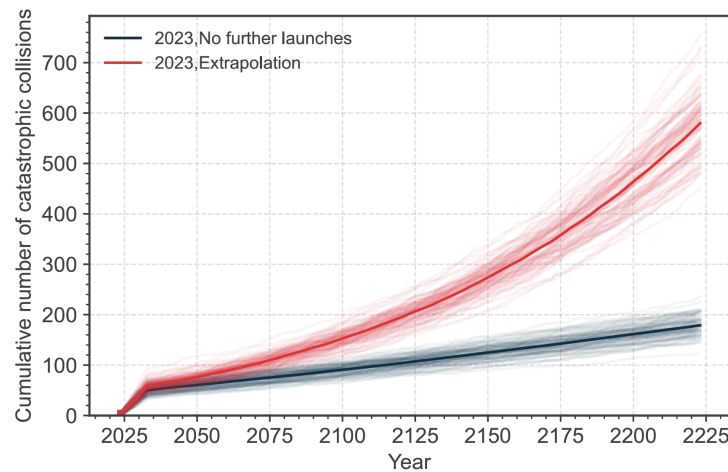


Figure 2.2: Cumulative Number of Possible Collisions Predicted Over the Upcoming Years

Beyond the numerical growth in orbital objects, the diverse nature of space debris complicates

removal efforts. Mission-related debris encompasses obsolete spacecraft, discarded rocket components, protective shields, and other hardware that may remain in low Earth orbit (LEO) for decades [22]. Accidental debris includes items lost during crewed missions, fragments from collisions and explosions, and materials shed through weathering or extreme thermal cycling [23, 24]. Among the most notable incidents is the high-speed collision between Cosmos 2251 and Iridium 33 in 2009, which generated thousands of new fragments [3], further magnifying the hazard to active satellites.

To confront this escalating threat, the aerospace community has explored several active debris removal (ADR) strategies, including harpoon-based captures, net-throwing systems, tethers, lasers and robotic approaches [25, 26]. Table 2.1 provides a qualitative comparison of these methods based on complexity, cost, and control level. Robotic arms, while more complex, offer superior dexterity and reusability. They can adjust grasping strategies in real time to accommodate diverse debris shapes and tumbling profiles, a critical advantage over most passive or tether-based methods. Some of the most advanced proposals focus on direct retrieval methods, whereby a servicer spacecraft physically grasps debris and removes it from valuable orbits. In particular, direct capture via a robotic manipulator has attracted significant interest because of its potential to handle objects of varying shapes and sizes, as well as to perform subsequent operations like inspection, refueling, or end-of-life disposal [27, 28].

Table 2.1: Qualitative Comparison of Key Debris Removal Approaches

Method	Complexity	Cost	Control Level
Nets / Harpoons	Moderate	Medium	Low precision, limited reusability
Tethers	Low	Low	Limited to certain orbits, minimal direct capture control
Lasers / Ablation	High	Potentially High	Non-contact, requires large power supply and precise targeting
Robotic Arms	High	Medium-High	High dexterity, multi-use

In direct-capture scenarios, a chaser spacecraft equipped with one or more manipulators positions itself near the target debris and extends a robotic arm to perform the grasp [29]. An illustration of this concept is given in Fig. 2.3, highlighting key components such as sophisticated vision systems and tactile sensors [30]. Although direct capture offers precise control over the

target, its implementation is technically demanding. The chaser must maintain a stable relative position, anticipate and accommodate target motion, and ensure that gripping forces do not impart excessive momentum or induce further fragmentation [31, 22].

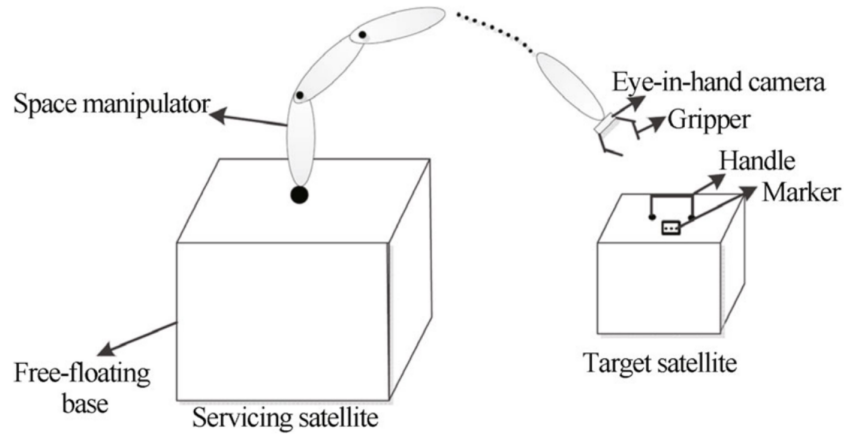


Figure 2.3: Direct Capturing Method Components

Compared to other methods, robotic arms excel when dealing with relatively slow-rotating debris (approximately $5^\circ/\text{s}$ or less) [32]. This is because higher angular velocities can lead to violent contacts, potentially causing the debris to bounce away or break apart, and can destabilize the chaser's attitude control if not carefully managed [33, 34]. Although detumbling techniques have been proposed—ranging from contact-based momentum exchange devices to tethered solutions [35, 36, 37, 38, 39]—they add mission complexity and are not always feasible for rapidly spinning or irregularly shaped objects. Consequently, current technology often constrains direct capture to targets spinning below certain angular thresholds.

Robotic manipulation of noncooperative, tumbling debris introduces several unique challenges:

- **Grasping Uncertainty:** Many debris objects lack standardized grapple fixtures or convenient gripping features, complicating the end-effector approach [22, 29].
- **Contact Dynamics:** Improper alignment or excessive approach velocity can induce large impact forces, risking damage to both the servicer and the debris. Post-contact bounce-off can also jeopardize nearby assets [3, 24].

- **Attitude Instability:** Once grasped, the combined chaser-debris system may exhibit destabilized attitude motion if torque and momentum are not carefully managed, necessitating robust control architectures [33, 34, 40].

Despite these technical hurdles, robotic arms are seen as one of the most immediately viable pathways toward flight-ready ADR demonstrations [41]. Moreover, robust manipulator control and advanced autonomy—bolstered by artificial intelligence (AI) and machine learning (ML) techniques—offer promising avenues for handling uncertainties in debris motion and on-orbit environments [13, 31]. A typical mission architecture involves phases such as approach, deployment, grasp, and post-grasp detumbling, each demanding high-precision control and real-time feedback [42, 43].

In parallel, various research initiatives are investigating complementary or alternative solutions, from Earth-based laser ablation [44] to electrodynamic tethers [45] and foam-based methods [46]. While these approaches hold promise for future debris remediation, direct capture using robotic manipulators remains a leading contender for near-term missions due to its adaptability and potential to engage in multiservice tasks (e.g., refueling, inspection, repairs).

By refining manipulator hardware, improving sensor fusion and perception algorithms, and developing robust control strategies that accommodate dynamic uncertainties, direct-capture methods will continue to mature. Ultimately, large-scale deployment of such systems could stabilize orbital debris populations, safeguard vital assets, and ensure the sustainability of space exploration and satellite operations for decades to come.

2.2 Space Manipulator Ground-Based Facilities

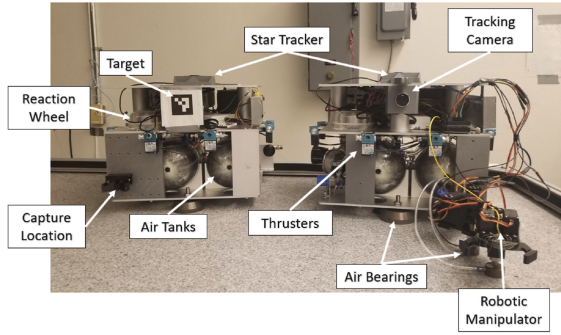
The development and validation of robotic capture technologies for on-orbit servicing (OOS) and debris removal entail numerous technical challenges. Chief among them is the dynamic control problem of a free-floating manipulator interacting with a tumbling, non-cooperative target in microgravity [47]. Achieving a *safe* capture—where the robot successfully grasps the target without

pushing it away—is crucial to prevent collision hazards. Concurrently, it is essential to stabilize the attitude of the chaser spacecraft carrying the robot. Following capture and stabilization, OOS tasks can proceed, or debris can be sent to a graveyard orbit or deorbited within the recommended 25-year limit [18, 48]. Given the high risk and cost of in-space testing, extensive validation of both hardware and control algorithms is typically conducted via ground-based facilities [49].

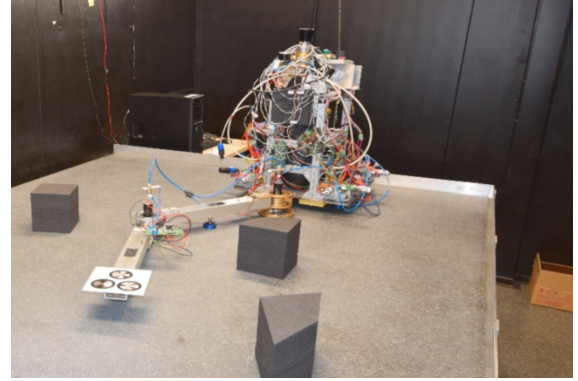
However, replicating microgravity conditions on Earth is inherently difficult and often requires multiple complementary approaches. Several methods have been proposed to emulate near-frictionless environments and validate robotic operations involving target contact. Below, I discuss notable testbeds and hardware-in-the-loop (HIL) systems that aim to bridge the gap between theoretical development and on-orbit application.

One of the most widely used methods for simulating low-gravity conditions is the air-bearing testbed, such as the planar systems at York University [50] and the Polish Academy of Sciences [51, 52]. These setups (shown in Fig. 2.4a and Fig. 2.4b) involve mounting robotic systems on high-precision air bearings that float over an exceptionally flat surface, approximating microgravity by dramatically reducing friction in planar motion. The main drawback is their restriction to two-dimensional movement, limiting the fidelity of capture and manipulation tasks that demand six-degree-of-freedom (6-DOF) motion.

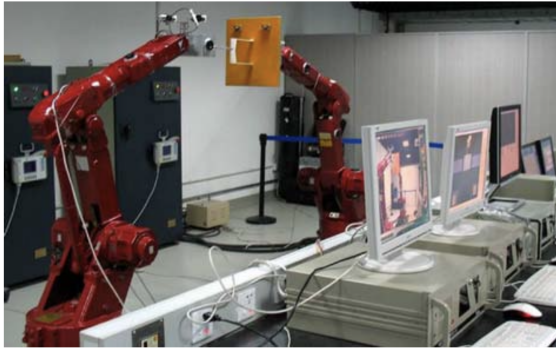
Meanwhile, HIL platforms have been developed at institutions like the Shenzhen Space Technology Center [53] and the German Aerospace Center [54]. As shown in Fig. 2.4c and Fig. 2.4d, these systems integrate physical robotic hardware with numerical simulations of spacecraft dynamics. They allow the manipulator to experience realistic contact forces when capturing a simulated target, even without full microgravity conditions. Although these facilities provide a more comprehensive environment for testing capture strategies and control algorithms, the end-effectors often remain relatively simple, featuring only basic pin-and-hole mechanisms.



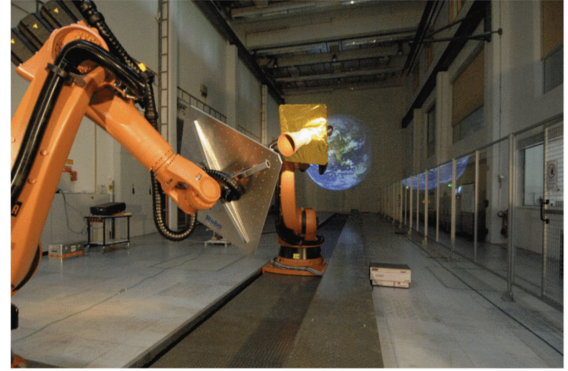
(a)



(b)



(c)



(d)

Figure 2.4: Examples of ground-based facilities.

Beyond air-bearing and HIL approaches, some laboratories employ active suspension systems for gravity compensation [55], which can theoretically allow 6-DOF robot motion by modulating tension on supporting cables. However, the dynamic coupling between the manipulator's vibrations and the suspension apparatus often causes residual forces, making it difficult to achieve accurate microgravity emulation. Neutral buoyancy setups have also been investigated [56], in which robotic arms are submerged in a tank to offset their weight. While such arrangements provide greater freedom of motion, fluid damping and added inertia can distort the system's true dynamics, especially under rapid or forceful interactions.

Parabolic flights [57, 58] and drop towers [59, 60] are considered high-fidelity methods because they offer actual free-fall conditions. These methods provide a true free-fall environment, effectively nullifying the net gravitational force experienced by the payload and thereby creating a genuine

microgravity condition. however, they afford only brief microgravity windows—from about 10 s in drop towers to around 30 s in parabolic flights—and are both space-limited and expensive, rendering them unsuitable for prolonged robotic capture testing.

More sophisticated testbeds have emerged to validate space manipulator capture missions under conditions closer to operational reality. For instance, the Canadian Space Agency’s SPDM Task Verification Facility [61] (see Fig. 2.5a) has demonstrated advanced robotic operations representative of tasks on the International Space Station. At Tsinghua University, a dual-robotic-arm testbed [62] (Fig. 2.5b) enables high-level autonomy testing for coordinated manipulation. Likewise, the China Academy of Space Technology has developed the Manipulator Task Verification Facility [63, 64], shown in Fig. 2.5c, for validating multi-arm coordination and capture stability. Meanwhile, the Zero-G Lab [65] (Fig. 2.5d) offers a versatile environment capable of emulating diverse orbital scenarios through integrated perception, control, and path planning modules.

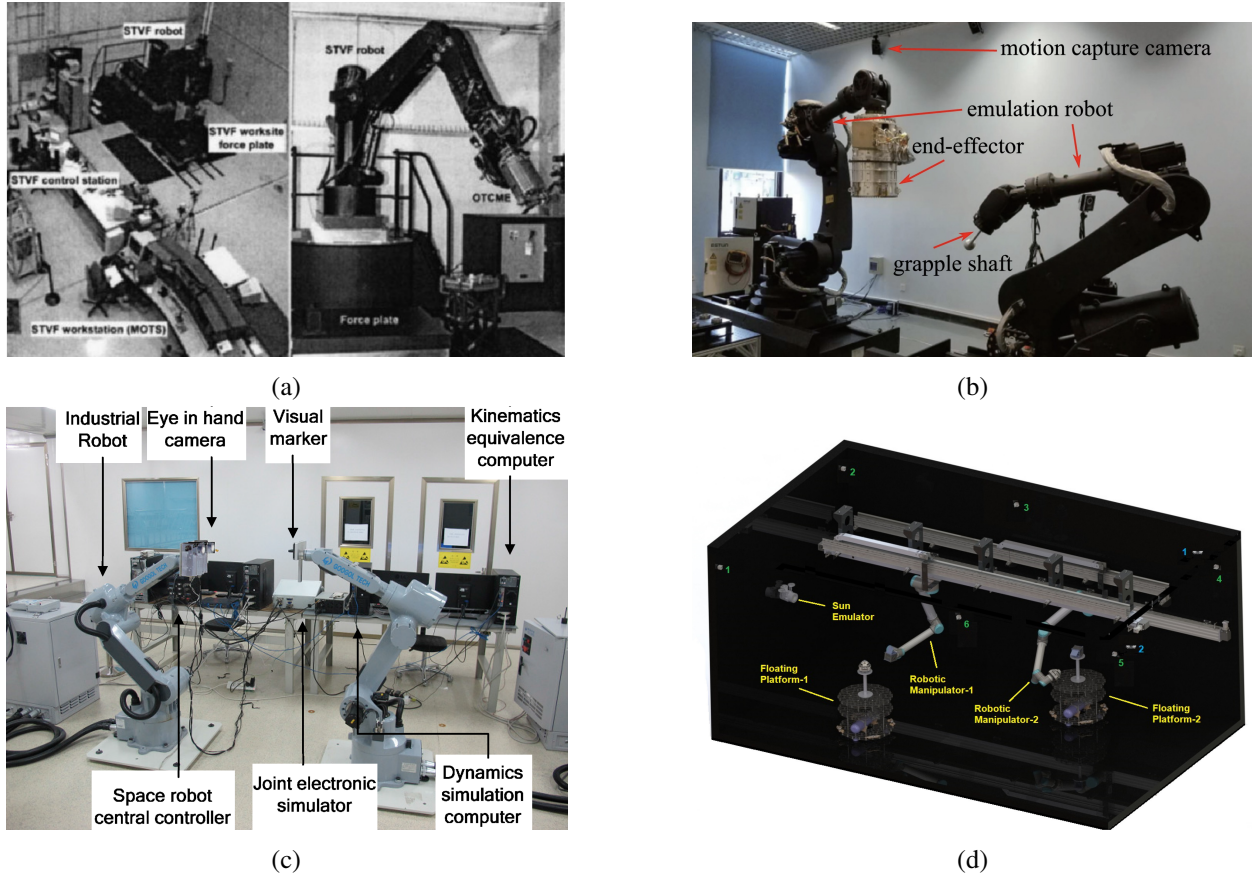


Figure 2.5: Further advanced facilities for on-orbit servicing and debris removal research.

Despite these innovative setups, most existing end-effectors in space robot testing facilities are designed primarily for basic pose-alignment and rigid fixture attachment. Such simple interfaces do not effectively replicate the complexities of capturing irregularly shaped debris or malfunctioning satellites. *Comprehensive* capture processes—requiring complex sensing feedback, improved manipulation dexterity, and post-capture stabilization—call for more advanced multi-finger grippers integrated with force/torque sensors, cameras, and tactile arrays.

To address these challenges, the present work introduces an experimental validation system that extends conventional end-effector capabilities. Specifically, I incorporate a three-finger gripper outfitted with a camera, a torque/force sensor, and tactile sensors. This allows *active gravity compensation* and precise contact force detection, enabling a more accurate simulation of true on-orbit interactions. With such sophisticated hardware, ground-based facilities can better approximate

the conditions of in-space capture, thereby improving the reliability of pre-flight testing for both next-generation OOS and ADR missions.

2.3 Robotic Grasping

Robotic grasping involves a suite of interrelated tasks, including path planning for the manipulator, optimal pre-grasp pose selection, and executing the grasp itself—often involving finger closure, force control, and sensor feedback to ensure stability [66, 67]. While these problems have historically been studied in isolation, modern advancements integrate them into a unified framework that aims to enable robots to perform robust grasping operations in real-world scenarios. Achieving this goal is vital for expanding robotic applications beyond controlled industrial settings into homes, healthcare, and outdoor environments.

Despite notable progress, robotic systems continue to display a “performance paradox,” often referred to as Moravec’s Paradox: robots can excel at tasks involving tremendous computational or physical capacity, such as beating human champions in strategic board games [68], but struggle with ostensibly simpler human tasks like object recognition and nuanced manipulation [69, 70]. Grasping is emblematic of this challenge, necessitating careful coordination of perception, motion planning, and control. Accordingly, research has focused on breaking grasping into smaller, more tractable problems, leveraging both analytical and data-driven approaches.

Conventional robotic grasping approaches rely on predefined motion paths and detailed environmental models to guarantee a stable grip [71, 72, 73]. These methods can be effective under tightly controlled conditions but often deteriorate in less predictable environments where objects may move or where model inaccuracies abound. Early works also depended heavily on exact modeling of object geometries, contact forces, and friction properties, typically requiring extensive computation and real-time pose estimation [74, 75, 76, 77].

Although such model-based strategies can achieve precise force-closure grasps, they remain susceptible to model errors and may fail to accommodate unexpected object shapes or motions

[78, 79, 80]. In scenarios demanding full 6-DoF control for both position and orientation, these methods become increasingly limited.

Physical contact between the robot’s end-effector and external objects lies at the heart of grasping and broader manipulation tasks. Accurate modeling of contact dynamics is crucial but challenging, particularly in microgravity or otherwise complex environments [81, 82]. The problem becomes even more pronounced for free-floating space manipulators, which must manage both grasp stability and base attitude control [83, 84].

Research efforts have explored advanced control frameworks—ranging from hybrid impedance controllers [85] to coordinated multi-body control [86, 87]—to handle contact interactions more effectively. Additionally, various modeling techniques account for impact forces and internal constraint forces during contact [88, 89], supporting more realistic simulations in HIL platforms [90].

Analytical approaches have historically played a central role in predicting optimal grasp points and force distributions. By leveraging detailed mathematical models of object geometry and manipulator kinematics, these methods aim to guarantee stable grasp configurations and sufficient grip forces. For instance, algorithms that search for force-closure points across an object’s surface can provide robust solutions when models are accurate [91, 92]. Yet, their applicability often hinges on computational feasibility and the fidelity of object representations. Hybrid methods that decompose complex objects into simpler primitives (e.g., superquadrics) [93] or exploit shape segmentation techniques [94] represent attempts to scale analytical approaches to real-world object sets. Others have optimized contact point selection using genetic algorithms [95] or low-dimensional posture subspaces (e.g., eigengrasps) for dexterous hands [96].

Despite these innovations, purely analytical methods still face limitations. They require precise prior knowledge of the object, can be computationally intensive, and often struggle with unmodeled variations. Consequently, many contemporary research directions seek to integrate analytical insights with data-driven techniques, trading strict mathematical guarantees for broader adaptability and robustness.

Traditional approaches, largely reliant on pre-programmed strategies and rigid models, exhibit limited flexibility and dexterity in unfamiliar or rapidly changing environments. As robots are expected to operate more autonomously—whether in industrial, domestic, or even orbital settings—these shortcomings underscore the need for more adaptive, data-driven frameworks.

2.4 Reinforcement Learning

In recent years, learning-based methods—including convolutional neural networks (CNNs) and Reinforcement Learning (RL)—have substantially improved the reliability of grasping in unstructured and dynamic environments [97, 98]. Such approaches circumvent explicit object modeling by learning grasp-relevant features directly from sensor data, enabling adaptability in scenarios where objects vary in shape, texture, or movement, and where high-fidelity models are unavailable [99, 100, 101].

Deep Reinforcement Learning (DRL) has shown remarkable potential in robotic grasping, especially in tasks involving stationary objects [102, 103, 104]. Nonetheless, there is an increasing demand for autonomous grasping technologies that can address the additional complexities of free-floating and moving entities [105]. Among the more critical issues for DRL in real-world robotics is the so-called *sim-to-real gap*: policies trained purely in simulation often degrade in performance when transferred to physical robots. Domain randomization during simulation training is one strategy to mitigate this gap, whereas HIL approaches offer another alternative. However, direct training on robots can be hazardous and time-consuming, rendering it largely impractical for many applications [106, 107, 108].

Many grasping solutions simplify the task by restricting the workspace to table-top settings and limiting the control problem to 3-DoF or 4-DoF [109, 110, 111, 112, 113]. Although effective in some contexts, such constraints become insufficient when the target is free-floating and can translate or rotate unpredictably. For that reason, more advanced methods seek 6-DoF grasping to accommodate the full range of potential object motion [114, 115, 116, 117, 118, 119].

Recent advances in machine learning—particularly deep learning—have revolutionized various domains, from computer vision to decision-making. In robotics, RL stands out as a promising avenue to combine control-theoretic principles with neural function approximators, though its application to continuous control tasks remains an active area of research. Deep learning-driven methods enable robots to learn grasping strategies by exploring the environment and receiving feedback (rewards or penalties), obviating the need for hand-tuned controllers and precise object models [120, 121]. In these paradigms, a neural network transforms sensory data (visual or tactile) into control actions—e.g., translational/rotational motions of the gripper or finger closures [122, 123]. By leveraging real-time sensor feedback, the trained policy adapts to previously unseen objects and nonstationary conditions without requiring extensive manual engineering [124].

The applications of this end-to-end, deep learning-driven framework are extensive, promising robust performance for tasks ranging from standard industrial bin-picking to complex in-home caregiving scenarios. As objects become more varied and subject to movement or deformation, learning-based approaches can adjust continuously through iterative trial-and-error, enhancing overall reliability and robustness.

A key obstacle in RL-based grasping lies in the intensive data requirements for policy training, often addressed by leveraging simulations [125]. High-fidelity physics engines allow rapid, cost-effective exploration of different control policies, while providing access to complete system state data. Consequently, researchers have successfully developed visuomotor RL policies within simulated worlds [126, 127]. That said, bridging simulation and the real world often calls for additional techniques—such as domain randomization and domain adaptation—to overcome model inaccuracies [128, 129, 130]. An alternative approach, particularly applicable when the simulated environment closely matches physical characteristics, is to train on state representations rather than raw sensor data. By eliminating vision-specific uncertainties, this technique can reduce computational overhead, making policy training more efficient.

Another breakthrough is the use of CNN-based feature extraction. Rather than depending on carefully constructed object geometries or extensive sensor fusion pipelines, these methods

directly process image data to recognize target objects and plan corresponding grasp configurations [131, 132]. This shift from manual feature engineering to learned representations significantly broadens the range of objects and environments that robots can handle. Nonetheless, full integration of DRL into real-world, continuous-control tasks remains challenging, requiring rigorous solutions for reliability and scalability [133, 134].

Recent investigations have explored more nuanced action spaces, allowing control policies to adapt to complex dynamical scenarios. Techniques like adaptive Q-learning, double deep Q-learning, and trust region policy optimization have all been employed to refine a robot’s ability to identify viable grasping maneuvers on-the-fly [135, 136, 137]. Additionally, missions aiming to capture moving or tumbling objects often segment the process into *pre-capture* (envelopment and velocity matching) and *post-capture* (force-regulated gripping) phases [138]. This sequential view underscores the delicate balance between avoiding target deflection and establishing a stable, minimal-force grasp.

Beyond vision, tactile perception is paramount for preventing unintentional collisions or excessive forces—issues that become particularly acute when the target is free-floating or delicate. By continuously measuring contact forces, robots can modulate their end-effector trajectories, pivoting from a purely kinematic pursuit to a gentler, dynamic enclosure [139, 140, 141]. This tactile-centric paradigm is especially pertinent to *pre-capture*, providing crucial feedback to halt or slow the approaching fingers and ensure minimal contact velocity. Empirical evidence indicates that incorporating tactile signals into RL training notably improves target tracking while reducing undesirable contact forces.

Whereas many DRL grasping frameworks concentrate on stationary table-top objects, more recent work has turned to the complexities of moving targets [142, 143]. Although these approaches offer partial solutions—for instance, planar dynamic grasping through object detection or rapid filtering of precomputed grasp databases—they generally assume relatively predictable, linear motions in 2D. By contrast, 6-DoF manipulation in microgravity entails handling both translation and rotation without the stabilizing effect of a fixed surface.

Curriculum Learning (CL) is a compelling strategy to overcome the large search space and intricate dynamics inherent to free-floating 6-DoF grasping. By starting with simplified tasks and incrementally increasing difficulty, CL helps the agent establish foundational competencies before tackling more elaborate or adversarial settings [144, 145]. This tiered approach has demonstrated enhanced sample efficiency and policy robustness in tasks ranging from locomotion to high-precision manipulation [146, 147, 148, 149], reducing the need for intensive human involvement in designing complex training regimens [150, 151].

Well-crafted reward functions remain instrumental in guiding RL-based grasping methods toward consistent outcomes [152]. However, reliance on simplified environments has often sidelined the interconnected challenges of full 6-DoF positioning and nuanced finger-force control [153, 154, 155, 156]. Although some multifingered solutions integrate tactile and proprioceptive data to stabilize object handling [157, 158, 159], they frequently assume targets that are largely stationary or easily modeled. Promising evidence emerges from multi-stage RL pipelines incorporating advanced perception or adversarial training [160], but verifying these methods in free-floating, microgravity-like environments remains an open research problem.

The summary of different RL-based grasping strategies is presented in Table 2.2. These methods encompass a wide spectrum of complexity, ranging from simple, static object grasping to dynamic, free-floating target capture. While each approach offers unique advantages, the ultimate goal remains to develop a unified framework capable of adapting to diverse object shapes, sizes, and motions, thereby ensuring reliable grasping in real-world scenarios. To the best of my knowledge, no single method has yet addressed the challenges of 6-DoF grasping. This is defined as the autonomous maneuvering of a robotic gripper to align and position itself optimally for grasping free-floating, moving, and rotating targets in microgravity environments, achieving a stable hold without imparting excessive forces or causing target deflection. This work aims to bridge this gap by proposing a novel, state-of-the-art RL-based grasping strategy that integrates advanced methods to enable robust, adaptive, and dexterous grasping in complex, dynamic, and uncertain conditions.

Overall, RL-based strategies for grasping—particularly those exploiting deep neural net-

Table 2.2: Summary of Representative RL-Based Grasping Studies

Study	Focus	Key Contributions	Methods	Limitations
Wu et al. [97], Lenz et al. [98]	Grasping in un-structured settings	Improved reliability; learned from sensor data	CNN-based RL on visual/tactile inputs	Static/single-object scenarios; sim-to-real gap
Mnih et al. [102], Lampe & Riedmiller [103], Quillen et al. [104]	Deep RL for manipulation	Scalable DRL policies (e.g., DQN)	Q-learning, policy gradients	High sample needs; limited generalization
Mehta et al. [106], Peng et al. [107], Exarchos et al. [108]	Bridging sim-to-real gap	Domain randomization, HIL improved transfer	Sim training, domain adaptation, HIL	Hazardous real-world training; simple task selected
Suzuki et al. [109], Chen et al. [110], Pedersen et al. [111]	Simplified grasping scenarios	Consistent results in structured spaces	Simplified spaces, end-to-end learning	Limited to static objects; 6-DoF ignored
Johns et al. [114], Zhang et al. [115], Mousavian et al. [116]	6-DoF grasp planning	Adapts to complex object poses	6-DoF planning, pose estimation, deep RL	High computation; physics modeling needed, just for stationary targets
Luo et al. [135], Van Hasselt et al. [136], Ghadirzadeh et al. [137]	Advanced RL algorithms	Faster convergence, stable policies	Off-policy RL, trust regions, replay buffers	Data-hungry; inconsistent high-DoF results
Chen et al. [142], Akinola et al. [143]	Grasping moving objects	Solutions for planar/linear targets	Detection, grasp databases, deep RL	Focus on 2D tabletop; 6-DoF/gravity ignored
Bengio et al. [144], Narvekar et al. [145]	Curriculum learning (CL)	Faster learning via staged tasks	Incremental/task-based training	Complex CL used; gaps in moving grasping
Cao et al. [152], Osa et al. [153]	Reward design, force control	Multi-finger manipulation with tactile feedback	Hierarchical RL, tactile integration	Focus on static objects; real-time limits

works—have progressed rapidly, offering robust object handling under increasingly complex and uncertain conditions. Advances in sim-to-real transfer, curriculum learning, and sensor integration (vision, tactile, force/torque) continue to elevate the potential of autonomous systems, pushing toward reliable, dexterous, and agile grasping of free-floating objects in both terrestrial and orbital domains. Yet, key gaps persist in bridging large action spaces, handling dynamically evolving targets, and guaranteeing safe, stable captures in extreme environments. Addressing these frontiers will shape the next generation of intelligent robotic grasping systems for OOS, space debris removal, and beyond.

Chapter 3 Problem Formulation

This chapter formulates the autonomous robotic grasping problem for space debris removal using Deep Reinforcement Learning (DRL). It defines the task within a Markov Decision Process (MDP) framework, specifying the states, actions, and rewards essential for grasp synthesis. The selection and implementation of the Soft Actor-Critic (SAC) algorithm are detailed, alongside the incorporation of Curriculum Learning (CL) to enhance training efficiency and policy robustness. Additionally, the chapter outlines the design of the simulation environment and the grasp quality metrics used to evaluate performance, providing a solid foundation for developing and validating the proposed autonomous grasping system.

3.1 Background

Reinforcement Learning (RL) provides a framework for autonomous decision-making through direct interaction with the environment, allowing an agent to learn from a reward signal rather than relying solely on labeled datasets [161]. In contrast to traditional supervised learning approaches, where the model is trained on predefined input-output pairs, an RL agent iteratively explores its surroundings by selecting actions, transitioning to new states, and receiving corresponding rewards [162]. This dynamic process facilitates the discovery of action sequences that maximize overall performance, thereby offering a promising strategy in fields with limited or hard-to-obtain training data.

3.1.1 Markov Decision Process

A foundational abstraction in RL is the Markov Decision Process (MDP). An MDP is described by a tuple $\langle \mathcal{S}, \mathcal{A}, P, r, \gamma \rangle$ where:

- $\mathcal{S}$ represents the set of possible states.
- $\mathcal{A}$ represents the set of possible actions.
- $P(s_{t+1} \mid s_t, a_t)$ defines the transition probability of moving to state s_{t+1} after taking action a_t in state s_t .
- $r(s_t, a_t)$ is the reward function that evaluates the desirability of each state-action pair.
- $\gamma \in [0, 1]$ is the discount factor, which modulates the importance of future rewards relative to immediate ones.

The agent's goal in an MDP is to learn a policy π that prescribes which action to take in each state, in order to maximize the expected discounted return [162, 163, 164]:

$$J(\pi) = \mathbb{E}_{\tau \sim \pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right], \quad (3.1)$$

where τ denotes the agent's trajectory through the state-action space. A higher γ emphasizes long-term rewards, whereas a lower γ places greater value on immediate gains.

3.1.2 State, Observation, and Action Space

In many practical scenarios, the agent either observes a partial description of the underlying environment (partially observed) or the complete environment description (fully observed). A fully observed system implies that the agent receives all relevant information needed for decision-making, rendering the state and observation indistinguishable. However, real-world robotic tasks often involve partial observability, where some aspects of the environment are hidden from the agent, thereby increasing the complexity of learning and planning.

Similarly, the action space can be either discrete or continuous [165, 166]. In a discrete action space, the agent selects actions from a finite set, such as moving forward, turning left, or turning right. In contrast, a continuous action space involves actions that lie in a continuous domain, such

as the joint angles of a robotic manipulator. This allows for greater control resolution but introduces more complex learning challenges.

The flexibility of RL to handle both discrete and continuous spaces makes it an attractive choice for problems involving high-dimensional control, such as robotic grasping.

3.1.3 Exploration vs. Exploitation

A pivotal aspect of RL is the balance between exploration and exploitation. While exploitation leverages existing knowledge to choose actions that maximize reward based on past experience, exploration probes new actions or strategies to gather information about the environment [162]. This dilemma is acute in real-world domains, where suboptimal exploration risks failing to discover better solutions, yet excessive exploration can slow convergence and waste valuable resources. Methods for balancing these objectives, such as decaying random action probabilities or more sophisticated exploration bonuses, are integral to effective RL design [167, 168].

3.1.4 Advantages for Robotics

RL’s capability to learn from sparse or limited data, combined with its capacity for robust discovery of optimal action sequences, makes it particularly compelling for robotic applications [166]. By framing manipulation, locomotion, or grasping tasks as MDPs, the robot can autonomously refine its control strategies to maximize reward signals that capture task success or efficiency. Moreover, RL-based algorithms excel in handling uncertainty and partial observability—key challenges in dynamic, unstructured environments [165]. As a result, RL has emerged as a powerful tool for intricate tasks such as space debris removal, non-prehensile manipulation, and multi-stage assembly.

Fig. 3.1 provides an overview of the RL feedback loop, illustrating how the agent’s actions influence the environment’s state and reward structure. This framework underscores the essential cycle of observation, action, and reward that drives policy optimization and highlights RL’s distinctiveness among machine learning paradigms for continuous control and sequential decision-making [162, 161].

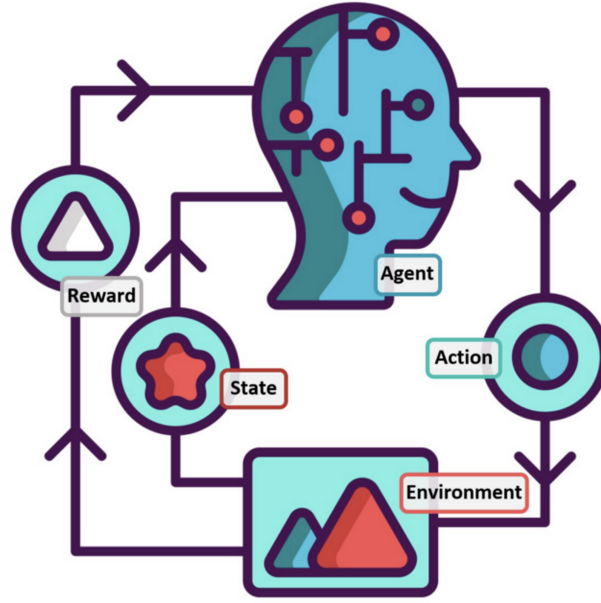


Figure 3.1: Basic reinforcement learning framework

3.2 Algorithm

In the context of DRL, policy formulation is often characterized by a nonlinear function, frequently a neural network, denoted by θ . The overarching ambition here is the fine-tuning of θ to cultivate optimal agent behavior. RL methodologies fall predominantly into three classifications: value-based, policy search, and actor-critic methods. Value-based strategies focus on deducing a value function that, in turn, informs the policy, ensuring decisions are made to maximize the expected reward. Policy search methods implement a parameterized policy, directly probing for the most effective policy parameters, often through trial and error, to optimize performance. Actor-critic methods uniquely integrate both the value function and the policy for a more rounded learning process. This combination allows for simultaneous learning of the policy (actor) and the value function (critic), leading to more efficient and effective decision-making processes. These methods are particularly advantageous in environments where both the rewards and the actions are complex and multi-dimensional [169, 170]. The method used in this work aligns with the actor-critic paradigm, where the learning process encompasses both the policy and the value function's parameters. In

such frameworks, the 'critic', informed by the value function, evaluates the actions made by the 'actor'. The actor, in response, amends its policy parameters, drawing on the critic's assessments. This synergetic relationship between the actor and critic components is pivotal in fine-tuning the decision-making process. By leveraging and using two separate neural networks for actor and critic, the model can effectively process and interpret complex, sequential, or spatial data [171]. This procedure is crucial for applications in environments with high-dimensional input spaces, like image-based or sequential decision-making tasks.

These algorithms encapsulate this dual mechanism, orchestrating the interaction between the actor and critic. It commences with the initialization of state observations and network parameters, followed by a cyclical process of action inference, execution, and subsequent reward assessment. The critic's role is pivotal in calculating the advantage, which quantifies the efficacy of actions against expected outcomes. Both networks undergo iterative updates, refining the policy and value estimations through a policy gradient approach and loss minimization, respectively. Algorithm 1 outlined the detailed of this process. This systematic approach is particularly tailored to the intricacies of robotics grasping, addressing the challenges of dynamic environments and diverse object interactions. The continuous adaptation and learning embedded within this framework significantly enhance the robot's grasping capabilities.

Model-based methods construct an internal model of the environment, which can improve sample efficiency but adds complexity and risks inaccuracies in high-dimensional settings. By contrast, model-free RL avoids explicit modeling and learns policies directly from experience, making it simpler and faster to implement—crucial for time-sensitive missions [172, 173]. This streamlined approach remains robust in dynamic scenarios like controlling a space manipulator to grasp a free-floating target, where uncertainty and rapid changes in the environment are the norm. The model-free actor-critic architecture, with its dual neural networks and iterative policy updates, is well-suited to the challenges of space debris removal, offering a versatile and adaptive solution for autonomous grasping tasks.

In this study, the application of cutting-edge, model-free RL techniques is explored to develop

sophisticated control systems for robots interacting with dynamic, free-floating targets. The focus is placed on the implementation of three advanced algorithms: Deep Deterministic Policy Gradient (DDPG), Twin Delayed Deep Deterministic Policy Gradient (TD3), and Soft Actor-Critic (SAC). These algorithms have been strategically selected for their impressive performance in solving intricate control issues, positioning them at the forefront of RL advancements. Fig. 3.2 provides an overview of the RL landscape, separating algorithms into model-free and model-based categories, and further grouping them by their underlying learning paradigms. Within the model-free branch, the actor-critic methods examined in this thesis (e.g., DDPG, TD3, and SAC) combine insights from both Q-learning and policy optimization approaches, thus leveraging the strengths of each technique to achieve robust performance in continuous control tasks.

Algorithm 1 Actor-Critic Algorithm

```

1: Initialize:
    • Actor parameters  $\theta$  (policy).
    • Critic parameters  $\phi$  (value function).
    • Discount factor  $\gamma$  and other hyperparameters.

2: for each episode do
3:   Reset environment, receive initial state  $s_0$ .
4:   while not terminal do
5:     Select action  $a_t$  using  $\pi_\theta(a_t | s_t)$ .
6:     Execute  $a_t$ , observe  $r_t$  and  $s_{t+1}$ .
7:     Estimate  $V(s_t)$  and  $V(s_{t+1})$  via the critic.
8:     Compute advantage term  $A_t \approx r_t + \gamma V(s_{t+1}) - V(s_t)$ .
9:     Update critic (adjust  $\phi$ ) to reduce value prediction error.
10:    Update actor (adjust  $\theta$ ) by increasing log-likelihood of actions with positive advantage.
11:  end while
12: end for

```

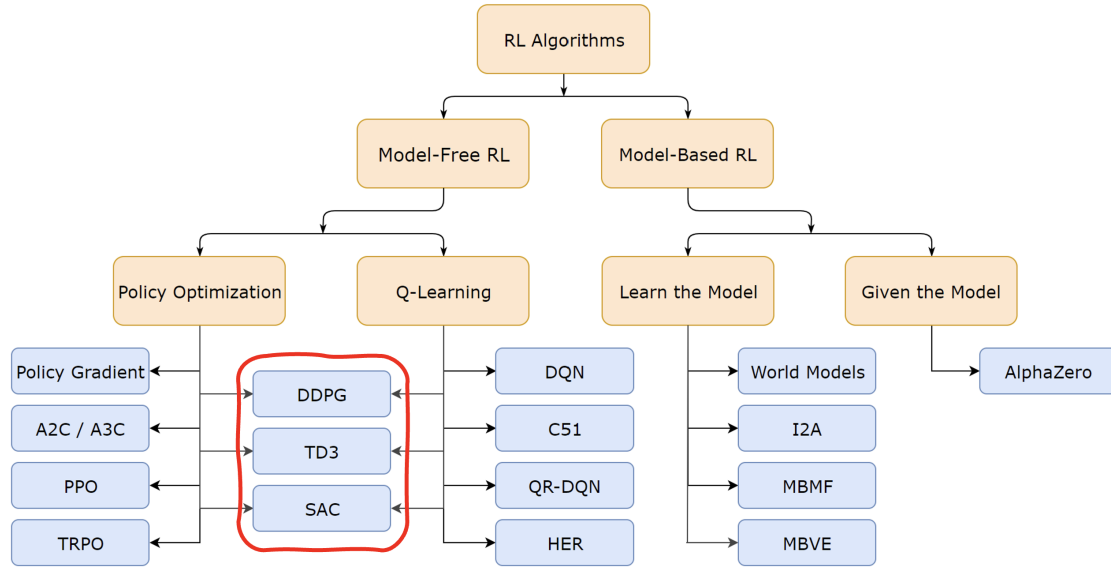


Figure 3.2: Flowchart categorizing Reinforcement Learning algorithms

DDPG, a landmark algorithm in the realm of RL, represents a significant leap forward in addressing challenges in continuous action spaces. It employs a deterministic policy update strategy within an actor-critic architecture. A notable feature of DDPG is its adoption of the Ornstein-Uhlenbeck process, which introduces noise into actions to foster exploration across diverse environments [174]. However, DDPG faced limitations in terms of stability and sample efficiency, particularly evident in environments with unpredictable or stochastic elements. This underscored the need for more refined approaches in the domain of continuous control.

To overcome these drawbacks, TD3 was developed as an advancement of DDPG. It features twin critic networks and implements delayed policy updates, effectively reducing overestimation bias and enhancing the stability of the learning process. TD3's approach of incorporating noise into target actions and using the lesser value from its twin critics for value updates significantly improved upon DDPG's framework [175]. This made TD3 a more robust option in environments characterized by fluctuating conditions and uncertainties, common in real-world applications. TD3's iterative improvements marked a significant step towards more stable and efficient learning in environments with continuous action spaces [176].

SAC builds upon its predecessors by introducing a maximum-entropy formulation that explic-

itly balances exploration and exploitation throughout training, pushing forward the capabilities of deep RL in high-dimensional continuous control. Unlike TD3’s deterministic design, which updates a single deterministic policy, SAC favors a stochastic policy that integrates an entropy term directly into the reward objective. This entropy component encourages continued exploration, preventing the agent from becoming prematurely deterministic and thus more prone to local optima [177]. By maximizing both expected return and policy entropy, SAC achieves a robust performance profile, adapting effectively to tasks where deep exploration is essential for identifying globally optimal strategies.

Compared to other actor-critic frameworks, SAC’s integration of a replay buffer and off-policy updates combines the sample efficiency typical of methods like Deep Q-Networks (DQN) [102] with the stability advantages of actor-critic techniques such as Proximal Policy Optimization (PPO) [178] and DDPG [175]. Because SAC is off-policy, it can continuously reuse past transitions to refine both the policy and the Q-value estimates, a benefit that on-policy algorithms like PPO do not fully leverage. Moreover, while DDPG also utilizes a replay buffer, SAC mitigates DDPG’s overestimation and exploration vulnerabilities through its entropy term and double Q-networks, yielding more stable learning dynamics [179]. Such a strategy makes SAC particularly compelling for robotic tasks involving high-dimensional action spaces, where thorough exploration is crucial yet often difficult to achieve.

3.2.1 Actor (Policy Network)

The SAC actor’s chief objective is to learn a parameterized stochastic policy $\pi_{\theta}(a | s)$ that maximizes the expected return while maintaining high entropy for robust exploration. In continuous action spaces, this policy is commonly modeled as a diagonal Gaussian distribution, characterized by learnable mean and variance:

$$\pi_{\theta}(a | s) = \mathcal{N}(\mu_{\theta}(s), \sigma_{\theta}(s)). \quad (3.2)$$

Here, $\mu_\theta(s)$ and $\sigma_\theta(s)$ are outputs of a neural network parameterized by θ . By sampling actions from this distribution rather than deterministically choosing the mean, the agent retains sufficient stochasticity to explore effectively.

To guide the policy towards higher returns while preserving entropy, SAC maximizes the following objective:

$$J_\pi(\theta) = \mathbb{E}_{s_t \sim \mathcal{D}, a_t \sim \pi_\theta} \left[Q_{\phi_i}(s_t, a_t) - \alpha \log \pi_\theta(a_t | s_t) \right], \quad (3.3)$$

where Q_{ϕ_i} is an ensemble of critic networks parameterized by ϕ_i , α is the temperature parameter that controls the trade-off between maximizing return and encouraging broader exploration, and $\mathcal{D}$ is the replay buffer of collected experiences. A higher value of α places more emphasis on entropy, increasing exploration, whereas a lower value makes the agent more exploitative by prioritizing the Q-value.

To enable stable and efficient gradient-based learning, the stochastic action sampling is made differentiable via the reparameterization trick. Instead of directly sampling $a_t \sim \pi_\theta(a | s_t)$, I express a_t in terms of a noise variable $\epsilon \sim \mathcal{N}(0, I)$:

$$a_t = \mu_\theta(s_t) + \sigma_\theta(s_t) \odot \epsilon, \quad (3.4)$$

where $\odot$ denotes the elementwise (Hadamard) product. This transformation allows gradients to flow from the Q-value term and the entropy term back into θ through both μ_θ and σ_θ , facilitating more precise updates of the policy parameters.

Beyond the inherent stochasticity of the Gaussian policy, additional controlled noise is injected into the output action to further enhance exploration:

$$a'_t = a_t \odot (1 + \eta_t \mathcal{M}), \quad \eta_t \in [-1, 1], \quad (3.5)$$

where $\mathcal{M} = 0.1$ is the maximum perturbation magnitude. This small yet consistent perturbation

guards against prematurely deterministic actions, increases the search radius in continuous action spaces, and promotes the policy’s resilience in noisy or uncertain environments.

During training, the actor parameters θ are updated by ascending the stochastic gradient of the policy objective $J_\pi(\theta)$. Concretely, the following computation is performed:

$$\nabla_\theta J_\pi(\theta) = \mathbb{E}_{s_t \sim \mathcal{D}, \varepsilon \sim \mathcal{N}(0, I)} \left[\nabla_\theta (Q_{\phi_i}(s_t, a_t) - \alpha \nabla_\theta \log \pi_\theta(a_t | s_t)) \right], \quad (3.6)$$

where a_t is reparameterized via $a_t = \mu_\theta(s_t) + \sigma_\theta(s_t) \odot \varepsilon$. The actor parameters are then updated as follows:

$$\theta \leftarrow \theta + \lambda_\pi \widehat{\nabla_\theta J_\pi(\theta)}, \quad (3.7)$$

where λ_π is the learning rate for the policy network, and $\widehat{\nabla_\theta J_\pi(\theta)}$ denotes the sampled gradient estimate. By iteratively applying these updates, the actor network converges toward a policy that balances exploitation of high-value actions with exploration induced by entropy maximization, ultimately enabling robust performance in continuous action domains such as microgravity grasping tasks.

3.2.2 Critic (Q-Value Networks)

SAC employs two independent Q-value networks, Q_{ϕ_1} and Q_{ϕ_2} , to combat overestimation bias, a known challenge in value-based methods. Each network is trained to minimize the discrepancy between predicted Q-values and a target based on the Bellman backup:

$$J_Q(\phi_i) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}} \left[(Q_{\phi_i}(s_t, a_t) - y_t)^2 \right], \quad (3.8)$$

where

$$y_t = r_t + \gamma \mathbb{E}_{a_{t+1} \sim \pi_\theta} \left[\min(Q_{\phi_1}(s_{t+1}, a_{t+1}), Q_{\phi_2}(s_{t+1}, a_{t+1})) - \alpha \log \pi_\theta(a_{t+1} | s_{t+1}) \right]. \quad (3.9)$$

By taking the minimum of the two Q-values during the target calculation, the algorithm counters the risk of positive bias that arises from using a single Q-value estimate. This dual-critic scheme, combined with entropy regularization, enhances learning stability and leads to more accurate value predictions [179].

3.2.3 Replay Buffer

Like other off-policy algorithms, SAC leverages a replay buffer $\mathcal{D}$ to store transitions (s_t, a_t, r_t, s_{t+1}) . This design allows the agent to train from a diverse set of prior experiences, reducing the correlation between consecutive samples and boosting sample efficiency. By sampling mini-batches of experiences in a uniformly randomized manner, SAC maintains a well-distributed training signal, stabilizing gradient updates and accelerating convergence relative to on-policy methods [102].

3.2.4 Temperature Parameter

The temperature parameter α is central to SAC’s entropy-driven exploration. Rather than keeping α fixed, the algorithm can learn its optimal value by enforcing a target entropy $\mathcal{H}_{\text{target}}$. This yields the additional objective:

$$J(\alpha) = \mathbb{E}_{a_t \sim \pi_\theta} \left[-\alpha \log \pi_\theta(a_t | s_t) + \mathcal{H}_{\text{target}} \right]. \quad (3.10)$$

Adaptive tuning of α helps the agent dynamically adjust the importance of exploration as training progresses. Early in training, a higher entropy incentive may be beneficial to explore under-sampled regions of the state-action space. Later, the agent might reduce α naturally if it needs to refine and exploit learned behaviors [179].

3.2.5 Network Architecture and Hyperparameters

In this work, I implement the SAC architecture using multilayer perceptrons (MLPs) for both actor (policy) and critic (Q-value) networks. Each MLP comprises two fully connected hidden

layers with 256 neurons per layer, employing ReLU activation functions between layers. For the policy network, a final tanh layer scales actions into the $[-1, 1]$ range. Table 3.1 summarizes the key hyperparameters used in training, including learning rates, batch sizes, and noise levels. I employ the Adam optimizer [180] to update both the actor and critic parameters. A replay buffer of size 1,000,000 stores past transitions, supporting robust off-policy training. The agent samples mini-batches of size 1024 every four episodes to balance computational cost against data diversity.

Table 3.1: Summary of SAC Hyperparameters

Network Name	Learning Rate	Optimizer	DNN Size
Policy Network	3×10^{-4}	Adam	$(s) \times 256 \times 256 \times (a)$
Critic Network	3×10^{-4}	Adam	$(s + a) \times 256 \times 256 \times 1$
Entropy Coefficient	Sample Batch	Action Noise ^a	Discount Factor γ
Learned	1024	10%	0.99
Buffer Size	Soft Update Rate τ	Training Freq	
1,000,000	0.005	4	

^a Maximum 10% change may be added or subtracted from each action.

Overall, SAC’s coupling of maximum entropy regularization, dual Q-value estimation, and off-policy data reuse produces a highly stable and sample-efficient algorithm. The workflow, depicted in Fig. 3.3, illustrates how the actor, critics, replay buffer, and adaptive α -learning converge to a cohesive training loop. Through persistent exploration and careful mitigation of overestimation, SAC advances RL’s ability to manage complex continuous control problems, laying a strong foundation for real-world robotics applications demanding both scalability and robust policy performance.

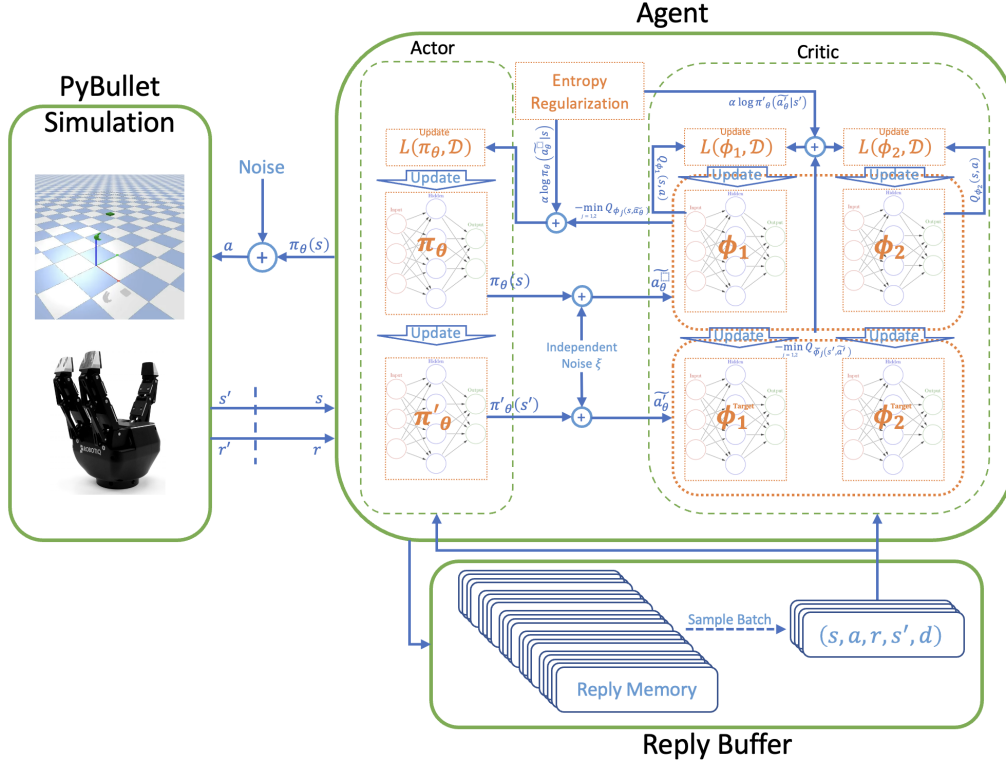


Figure 3.3: Training workflow of SAC algorithm

3.3 Curriculum Learning

In standard RL settings, agents frequently encounter difficulties when attempting to master complex tasks right from the onset, particularly in high-dimensional or rapidly changing scenarios. This challenge is further magnified when the agent must interact with free-floating moving or rotating targets, where the motion parameters can significantly enlarge the search space and introduce instability if the agent lacks robust exploration and adaptation mechanisms [181]. When an agent faces the complete challenge without intermediate steps, it may either fail to converge to a viable policy or settle prematurely into suboptimal strategies that do not generalize beyond narrow states or transitions.

To mitigate these issues, CL employs a systematic approach to task design, commencing with simpler tasks and progressively increasing complexity. Early-stage tasks may restrict the

range of rotational velocities or target orientations, allowing the agent to establish fundamental behaviors such as intercepting slow-moving targets before confronting more difficult motion patterns [145]. Beyond merely facilitating smoother exploration, this incremental buildup ensures that the agent’s learned representations are reused and refined rather than overwritten or discarded. As the agent acquires baseline proficiency, it is gradually exposed to a broader distribution of states, encompassing increasingly aggressive rotational velocities and abrupt orientation changes. This stepwise methodology reduces the risk of catastrophic forgetting in neural network policies, where the agent might otherwise lose previously acquired skills upon encountering significantly harder tasks.

A further advantage of CL lies in its capacity to enhance sample efficiency and promote faster convergence, since the agent is not overwhelmed by the full spectrum of environmental complexities at the outset [146, 182]. Each intermediate stage delivers valuable experiences pertinent to a specific subset of the overall task, thereby reducing the likelihood of unproductive exploration that often plagues standard RL. By mastering sub-components of the environment in succession, the agent accumulates a repertoire of partial solutions that serve as building blocks for managing more intricate dynamics [144]. This hierarchical acquisition of capabilities has clear parallels to skill acquisition in robotics, where tasks such as locomotion and object manipulation typically require progressive refinement of simpler skills [148, 149, 183].

Unlike Learning from Demonstration (LfD), which heavily relies on high-quality expert trajectories [184, 185], CL offers a more autonomous mechanism for policy initialization. While LfD can rapidly bootstrap an agent’s performance, its efficacy hinges on the availability and diversity of demonstrations, and it may falter in scenarios where expert data is sparse or reflects only a narrow portion of the task distribution [150, 151]. By contrast, CL makes fewer assumptions about the nature of expert knowledge and instead uses algorithmic control to expand or randomize environmental parameters in small increments. These incremental adjustments provide the agent with continual opportunities to explore new states and refine strategies without relying on an exhaustive set of human-supplied examples.

In dynamic environments, this staged learning paradigm is most effective when the schedule for task progression is coupled with adaptive triggers. Instead of strictly following a predefined sequence of difficulty increments, the agent’s performance—often gauged via success rates, reward thresholds, or a rolling average of returns—can drive the decision of when to advance to more difficult tasks [147]. Performance-based progression prevents stagnation on overly simple tasks while also guarding against abrupt escalations that could overwhelm the agent. This adaptive logic is particularly important in robotic scenarios where sudden increases in rotational velocity or erratic target motion could introduce unwieldy state transitions. The agent’s recent history of success (or failure) offers a quantitative indicator of readiness, balancing exploration and exploitation across skill levels.

In the specific context of free-floating moving and tumbling targets, my approach employs four incremental levels, gradually intensifying the target’s rotation and orientation changes. Early stages focus on establishing fundamental position control when the target mostly exhibits linear translation, enabling the agent to master basic interception maneuvers. By the time the agent reaches higher levels of the curriculum, the orientation (ψ) and angular velocity (ω) escalate exponentially, compelling the policy to adapt to scenarios that require predictive tracking and agile control. This escalated challenge, introduced at an appropriately timed point, helps the agent develop domain-invariant strategies and fosters generalizable skill sets for six-degree-of-freedom (6-DoF) grasping [144, 145]. Furthermore, by maintaining continuity between the levels, the agent avoids significant regressions in performance—a concern that can arise if the curriculum transitions are either too abrupt or disconnected.

Finally, coupling this curriculum with real-time monitoring ensures that difficulty increments maintain an optimal zone of learning: neither trivial nor overly daunting. The environment modulates ψ and ω based on an online performance metric, such that the agent consistently trains at the frontier of its current capabilities. This fine-grained adaptive loop mitigates the tendency to converge prematurely to limited strategies and reduces the necessity for repeated trial-and-error cycles at higher complexities. In essence, a curriculum-based RL approach not only tackles the fundamental

sampling and exploration bottlenecks of challenging tasks but also nurtures robust policies that can handle the unpredictable dynamics of free-floating, rotating objects—a hallmark of many real-world robotics applications [181, 146, 182, 147].

3.4 Learning Environment

The learning environment is a critical component in developing and evaluating the RL agent’s ability to grasp free-floating moving targets. In this study, I employed the PyBullet physics engine [186] in conjunction with the Stable Baselines3 (SB3) framework [187, 188]. PyBullet was chosen for its high fidelity in simulating physical interactions, particularly its precise handling of contact dynamics between rigid bodies. Utilizing the Sequential Impulse method, PyBullet iteratively applies impulses to colliding bodies, ensuring non-penetrative and physically accurate responses that are essential for realistic simulation of contact forces and interactions during the grasping task [189]. Fig. 3.4 illustrates the PyBullet simulation environment, showcasing the free-floating target and the robotic gripper in a microgravity setting.

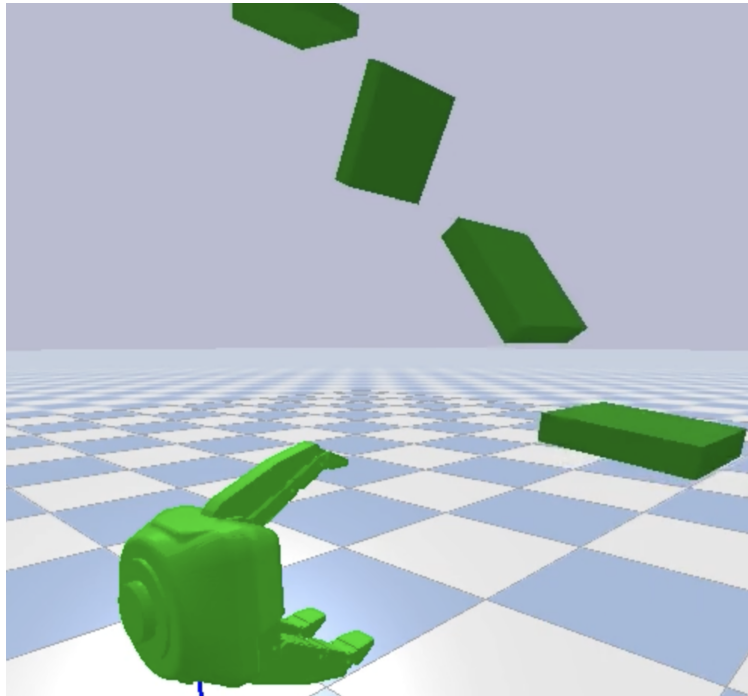


Figure 3.4: PyBullet simulation environment for grasping free-floating targets

Integrating PyBullet with the SB3 framework enables the deployment of sophisticated RL algorithms, allowing efficient and precise agent training within a robust simulation environment. RL agents inherently depend on exploration and trial-and-error to identify optimal solutions. While hardware-in-the-loop systems can yield more accurate and realistic results, they often involve significant costs, extended iteration times, and risks of physical hardware failures or damage. In contrast, simulation platforms like PyBullet provide a rapid testing environment that accelerates experimentation and eliminates the dangers associated with real-world hardware interactions. This setup facilitates extensive exploration of the agent’s action space, promoting the development of robust and transferable policies without the financial and risk-related burdens of physical experiments.

Moreover, RL agents typically require comprehensive state observations, frequent system resets, and precise reward calculations at each timestep—requirements that are challenging to consistently achieve in real-world settings due to sensor noise, environmental unpredictability, and the impracticality of repeatedly resetting physical systems. Simulations offer controlled conditions where these prerequisites can be meticulously maintained, ensuring that the agent receives accurate and consistent feedback during training. Training in simulation provides a safe and cost-effective platform for trial-and-error learning, ensuring that the agent can generalize its learned behaviors to real-world scenarios with minimal adaptation.

This methodological approach ensures exposure to a diverse range of scenarios, facilitating a comprehensive learning experience. The specifics of the training sequence and algorithmic operations are detailed in Fig. 3.5, providing a clear blueprint of the procedural methodology. By leveraging the strengths of simulation environments, the training process becomes not only more efficient but also significantly more scalable and adaptable, laying a solid foundation for deploying RL agents in complex and dynamic real-world applications.

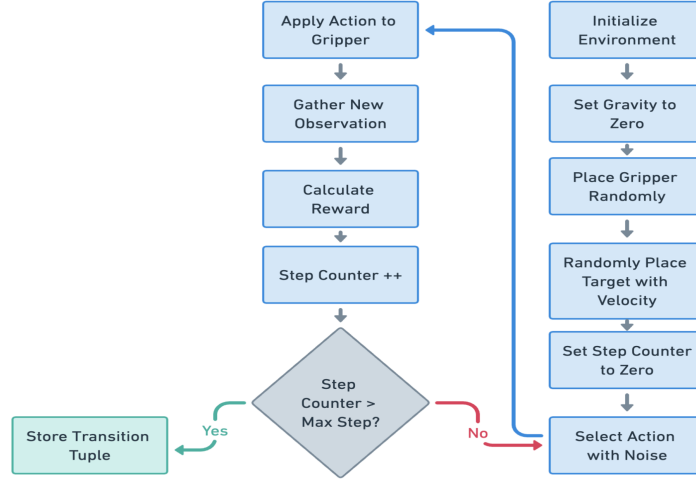


Figure 3.5: Procedure for each episode during the training

3.4.1 Target Shape Selection

A fundamental aspect of the learning environment is the selection of an appropriate target shape for the grasping task. In this research, I opted for a flat box as the target object, with dimensions of $0.2\text{m} \times 0.2\text{m} \times 0.04\text{m}$. This choice was motivated by several factors:

1. **Simplicity and Versatility:** The flat box serves as a versatile and representative model for various space debris components, such as nozzle surfaces, solar panels, and launch adapter rings (LARs). These objects can often be approximated by box-like structures, making the flat box an ideal generalized representation [190].
2. **Compatibility with Image Processing Algorithms:** Most image processing algorithms are adept at providing bounding box representations of objects [191, 192]. By selecting a brick-like target, I ensure that the visual perception system can accurately and efficiently detect and localize the target within the camera’s field of view. This alignment between the target’s geometric properties and the capabilities of image processing algorithms enhances the agent’s ability to interpret and respond to visual data effectively.
3. **Facilitation of Gripper Maneuverability:** The flat box design facilitates the Robotiq 3-finger gripper’s ability to surround and securely grip the object. The gripper can easily encase

the box, ensuring a firm and stable grasp, which is crucial for the Pre-capture phase of the grasping task.

4. **Simulation Feasibility:** The chosen box dimensions ensure that the gripper's fingers, when fully extended, are capable of encompassing the target without exceeding their maximum reach. This constraint guarantees the feasibility of executing the Pre-capturing phase within the simulation environment, allowing the agent to practice and refine its grasping strategies effectively.

3.5 Grasp Quality Metrics

Grasping free-floating targets in a microgravity environment poses significant challenges, as the absence of gravitational forces allows objects to drift freely, making them more susceptible to even minor external perturbations [193]. In such scenarios, a robust grasping mechanism must exhibit stability, precise force distribution, and strong resistance to external disturbances. To systematically assess these qualities, several grasp metrics have been developed. These metrics provide both theoretical and practical insights into grasp performance and are critical for evaluating and comparing different robotic gripping strategies [194].

Force closure is a fundamental criterion that determines whether the gripper can apply adequate contact forces in all directions required to hold the object securely. In other words, the combined forces from the gripper's contact points must be capable of counteracting external wrenches (forces and torques) to prevent the object from moving or slipping. Especially in microgravity, where the absence of weight forces can lead to unexpected object motion, achieving force closure ensures that the target remains stable once gripped. A grasp that satisfies force closure can immobilize the object, thereby enhancing both manipulation accuracy and overall grasp reliability [195].

Beyond simply achieving a force closure grasp, the Grasp Wrench Space (GWS) offers a more detailed quantitative evaluation. The GWS effectively represents the set of all wrenches (i.e., forces and torques) that the gripper can apply to the object. A larger GWS indicates that the gripper

is capable of resisting a broader range of external disturbances, including both translational and rotational motions [196, 197]. In microgravity conditions, objects can spin and drift erratically; therefore, having a robust GWS ensures the gripper can adapt to and counteract such movements. This capacity is essential for tasks like space debris capture, where unpredictable spins and velocities are common.

The Ferrari and Canny metric is a well-established measure that unifies the concepts of force closure and resistance to external wrenches into a single numerical value [198]. Essentially, it evaluates how robustly the contact points can resist worst-case disturbances in all directions. A higher value indicates a safer and more reliable grasp, offering stronger performance in real-world scenarios, including dynamic conditions often encountered in microgravity. By combining geometry, contact friction constraints, and feasible force directions, this metric provides a comprehensive way to compare different grasp configurations [199].

Post-grasp drift refers to the unintended translational or rotational motion of the object following initial contact and closure of the gripper. In terrestrial environments, gravity and friction with the supporting surface can help dampen residual motion; however, in microgravity, these natural damping effects are absent [200]. Consequently, small perturbations can lead to significant drift, jeopardizing grasp stability. My system addresses post-grasp drift by ensuring the gripper's base movement fully governs the object's motion once contact is established. This design choice effectively nullifies any independent object motion, leading to more precise control and reduced risk of object displacement.

Closely linked with post-grasp drift is the concept of object impulse control, which focuses on how effectively the gripper can absorb or counteract the object's momentum upon contact. High impulse control implies that the gripper swiftly neutralizes any existing momentum, thereby preventing oscillations or drift in the object's position and orientation. This aspect is particularly important for handling free-floating targets, as uncontrolled momentum can result in collisions with other objects or the surrounding environment. By carefully orchestrating the contact dynamics, my design ensures minimal residual motion, significantly improving grasp performance in microgravity.

Repeatability gauges how consistently the gripper can achieve a secure hold on the same object under similar conditions. A highly repeatable grasping system is crucial for automation and reliability, especially during repetitive tasks such as capturing multiple pieces of debris [194]. Shape adaptation, on the other hand, addresses the gripper’s capability to conform to objects of varying geometries and thicknesses. This is essential for scenarios where the debris or target objects may have irregular shapes or unknown dimensions. In this work, both metrics are improved by applying domain randomization during the training process. By systematically exposing the agent to a wide spectrum of object sizes, shapes, and dynamic conditions, I ensure that the learned policies are robust, flexible, and generalizable to real-world tasks.

Overall, these grasp quality metrics—Force Closure, Grasp Wrench Space, Ferrari and Canny Metric, Post-Grasp Drift, Object Impulse Control, and Grasp Repeatability & Shape Adaptation—collectively form a comprehensive framework for evaluating and optimizing robotic grippers in microgravity conditions. By addressing both the static and dynamic aspects of grasp quality, these metrics provide a thorough understanding of a gripper’s performance and reveal opportunities for further improvements, which are especially vital in high-stakes applications such as space debris removal and other orbital servicing tasks.

3.6 Direct Capturing Method for Space Debris Removal

The direct capturing method for space debris removal entails the deployment of a sophisticated robotic manipulator system designed to physically capture and manage defunct satellites, spent rocket stages, and other forms of orbital debris. This approach harnesses the capabilities of advanced robotic arm technology, which boasts enhanced dexterity and precision, crucial for the intricate task of debris manipulation in the challenging environment of space. The system is meticulously engineered to address and overcome the inherent challenges associated with dynamics and control within a microgravity setting, ensuring reliable operation when approaching and securing various types of space debris.

Central to the direct capturing method are space robotics manipulators, which serve as the primary tools for debris acquisition and handling. These manipulators are typically integrated into a free-floating robotic platform, which can be conceptualized as a serial-link manipulator anchored to a base that remains dynamically isolated from external forces and torques. As illustrated in Fig. 3.6 [201], the robotic system consists of a base that maintains a free-floating state, devoid of any fixed anchorage, thereby necessitating precise control mechanisms to manage the manipulator's movements without inducing destabilizing perturbations to the base.

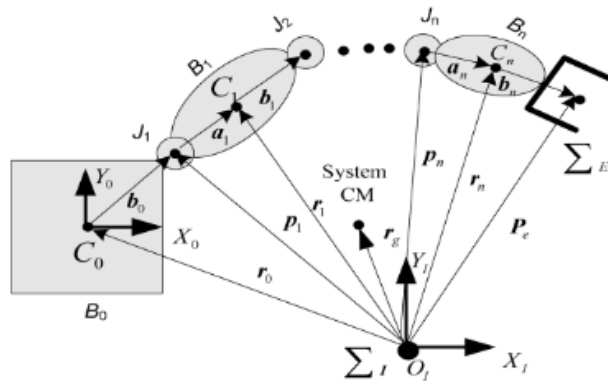


Figure 3.6: Schematic representation of the free-floating base with a serial-link manipulator.

Given the absence of a fixed base in space robot systems, the motion of the manipulator inevitably influences the position and orientation of the free-floating base. Consequently, any movement of the manipulator can disrupt the stability of the system's spatial configuration. To mitigate this, a stable reference coordinate system, referred to as the inertial coordinate system and denoted by I , is established. This inertial frame serves as the foundation for defining the spatial relationships and kinematics of the robotic manipulator.

The manipulator's configuration is described by the positions and orientations of its individual links and joints, denoted as $\mathbf{B}_i$ and $\mathbf{J}_i$, respectively. Specifically, $\mathbf{B}_0$ represents the coordinate system attached to the satellite base. The kinematic relationship governing the position and orientation of the manipulator's end-effector can be mathematically expressed as:

$$\mathbf{P}_e = \mathbf{r}_0 + \mathbf{b}_0 + \sum_{i=1}^n (\mathbf{a}_i + \mathbf{b}_i), \quad (3.11)$$

$$\mathbf{A}_e = \mathbf{A}_0 \cdot \mathbf{A}_1 \cdot \dots \cdot \mathbf{A}_n. \quad (3.12)$$

In Eq.(3.11), $\mathbf{P}_e$ denotes the position vector of the end-effector, which is a function of the base position $\mathbf{r}_0$, the base orientation $\mathbf{b}_0$, and the cumulative contributions of each link and joint in the manipulator. Eq.(3.12) defines the orientation $\mathbf{A}_e$ of the end-effector as the product of rotation matrices $\mathbf{A}_0, \mathbf{A}_1, \dots, \mathbf{A}_n$, where $\mathbf{A}_0$ is the base orientation and $\mathbf{A}_i$ represents the rotation matrix from link $i - 1$ to link i .

To analyze the dynamic behavior of the end-effector, these kinematic equations are differentiated with respect to time, yielding expressions for the linear and angular velocities of the end-effector:

$$\mathbf{v}_e = \dot{\mathbf{P}}_e = \mathbf{v}_0 + \boldsymbol{\omega}_0 \times (\mathbf{P}_e - \mathbf{r}_0) + \sum_{i=1}^n [\mathbf{K}_i \times (\mathbf{P}_e - \mathbf{P}_i)] \dot{\theta}_i, \quad (3.13)$$

$$\boldsymbol{\omega}_e = \boldsymbol{\omega}_0 + \sum_{i=1}^n \mathbf{K}_i \dot{\theta}_i. \quad (3.14)$$

In these equations, $\mathbf{K}_i$ is a unit vector that indicates the axis of rotation for joint $\mathbf{J}_i$, and $\dot{\theta}_i$ represents the angular velocity of each respective joint. The linear velocity $\mathbf{v}_e$ of the end-effector is influenced by both the base motion $\mathbf{v}_0$ and $\boldsymbol{\omega}_0$, as well as the contributions from each joint's motion. Similarly, the angular velocity $\boldsymbol{\omega}_e$ is a cumulative effect of the base's angular velocity and the manipulator's joint rotations.

By consolidating these velocity expressions, a comprehensive relationship is established that links the end-effector's velocity to both the base and manipulator motions through the Jacobian matrices:

$$\begin{bmatrix} \mathbf{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = \mathbf{J}_b \begin{bmatrix} \mathbf{v}_0 \\ \boldsymbol{\omega}_0 \end{bmatrix} + \mathbf{J}_m \dot{\boldsymbol{\Theta}}. \quad (3.15)$$

Here, $\mathbf{J}_b \in \mathbb{R}^{6 \times 6}$ is the Jacobian matrix associated with the base motion, $\mathbf{J}_m \in \mathbb{R}^{6 \times n}$ is the

Jacobian matrix related to the manipulator's motion, and $\dot{\Theta}$ is the vector of joint angular velocities.

In the context of a free-floating system, where no external forces or torques act upon the robotic platform, the principle of conservation of momentum dictates that the system's initial linear and angular momenta remain zero. This constraint leads to the following relationship:

$$\mathbf{H}_b \begin{bmatrix} \mathbf{v}_0 \\ \boldsymbol{\omega}_0 \end{bmatrix} + \mathbf{H}_{bm} \dot{\Theta} = 0 \quad \Rightarrow \quad \begin{bmatrix} \mathbf{v}_0 \\ \boldsymbol{\omega}_0 \end{bmatrix} = -\mathbf{H}_b^{-1} \mathbf{H}_{bm} \dot{\Theta} = \mathbf{J}_{bm} \dot{\Theta}. \quad (3.16)$$

In Eq.(3.16), $\mathbf{H}_b$ represents the inertia matrix of the base, while $\mathbf{H}_{bm}$ denotes the coupling inertia matrix that accounts for the interaction between the base and manipulator dynamics. By substituting the expression for the base velocities back into Eq.(3.15), The final expression is thus obtained, encapsulating the end-effector's velocity solely in terms of the manipulator's joint velocities:

$$\begin{bmatrix} \mathbf{v}_e \\ \boldsymbol{\omega}_e \end{bmatrix} = [\mathbf{J}_b \mathbf{J}_{bm} + \mathbf{J}_m] \dot{\Theta} = \mathbf{J}_g \dot{\Theta}. \quad (3.17)$$

Here, $\mathbf{J}_g$ denotes the generalized Jacobian matrix (GJM) [202], which encapsulates the combined effects of the base attitude, joint angles, and the mass properties of each component of the robotic system. The detailed formulations of the Jacobian matrices $\mathbf{J}_b$, $\mathbf{J}_m$, and $\mathbf{J}_g$ are comprehensively detailed in references [203] and [204].

Eq.(3.17) establishes the fundamental mathematical framework for the dynamic modeling of autonomous space robotic manipulators operating within a free-floating space environment. This rigorous dynamic model serves as the essential foundation upon which control policies are developed. Specifically, it underpins the design and implementation of control strategies that are informed by RL methodologies, enabling the autonomous robotic system to adaptively and efficiently execute debris capture maneuvers in the complex and variable conditions of space.

Chapter 4 Simulation and Results

This chapter details the simulation-based training and evaluation of reinforcement learning (RL) algorithms applied to robotic grasping tasks. It encompasses a comparative analysis of various Deep Reinforcement Learning (DRL) methods, investigates the impact of domain randomization and tactile feedback on policy robustness, explores curriculum-based learning for managing complex target dynamics, and introduces a Hierarchical Reinforcement Learning (HRL) framework for effective grasping of free-floating space debris. Through extensive simulations, the chapter demonstrates the effectiveness of these approaches in achieving reliable and adaptable robotic control in dynamic and uncertain environments.

4.1 Comparative Analysis of Reinforcement Learning Algorithms for Robotics Control

In this part, I present a comprehensive exploration of a model-free DRL framework designed to tackle the challenges of robotic grasping through simulation-based training. By rigorously comparing and contrasting various DRL algorithms commonly employed in robotic manipulation, I identify the most effective method for controlling a Robotiq gripper in continuous action spaces. Central to this investigation is the development and systematic evaluation of multiple reward functions, enabling me to discern the configuration that most favorably shapes grasping behaviors. Crucially, my findings demonstrate not only the viability of teaching advanced control maneuvers solely from experiences accumulated within a simulated environment, but also the capacity of these learned policies to generalize successfully to novel scenarios and adapt to significant environmental variations. This work thus validates the feasibility of employing DRL for efficient, continuous robotic control and illuminates the broader potential of machine learning methodologies to address

intricate real-world tasks within feasible temporal and computational constraints.

4.1.1 Definition of Actions and Observations

RL strategies for robotic control rely on carefully defining the action and observation spaces. While static tasks can often be managed with simpler setups, free-floating targets present unpredictable and complex motion patterns, demanding an observation–action framework that is both comprehensive in capturing dynamic interactions and streamlined for real-world deployment.

The observation space, denoted by $\mathbf{S}$, comprises 45 dimensions drawn from multiple data sources. A 12-dimensional vector represents the gripper state, including its base position, orientation, and corresponding velocities. An analogous 12-dimensional vector encodes the target (i.e., the free-floating block) state, featuring its position and velocity. Another 12-dimensional vector specifies the relative position and velocity between the gripper and the target, capturing interaction cues that are vital in dynamic environments. To reflect the gripper’s finger configuration, a 3-dimensional vector records the first joint position of each finger, exposing their angular disposition. Finally, a 6-dimensional vector provides contact information, detailing the total normal force and the number of fingertip contacts—key factors in assessing force distribution and grip stability. The complete state vector is thus given by:

$$\mathbf{S} = \{P_x, P_y, P_z, P_\phi, P_\theta, P_\psi, \dot{P}_x, \dot{P}_y, \dot{P}_z, \dot{P}_\phi, \dot{P}_\theta, \dot{P}_\psi, \delta P \dots, \delta \dot{P} \dots, F_{N_i}, n_{c_i}\} \in \mathbb{R}^{45}, \quad (4.1)$$

where each symbol represents a distinct component of the gripper or target state, inclusive of their relative positions, velocities, and relevant contact metrics.

The action space, denoted by $\mathbf{a}$, is continuous and encodes the translational and rotational movements of the gripper within its body frame. At each timestep, the agent selects a 6-dimensional action vector:

$$\mathbf{a} = \{a_x, a_y, a_z, a_\phi, a_\theta, a_\psi\} \in \mathbb{R}^6. \quad (4.2)$$

Each component $\{a_x, a_y, a_z\}$ corresponds to translational steps (up to ± 10 mm), while $\{a_\phi, a_\theta, a_\psi\}$ represent rotational adjustments (up to 0.1 rad) around the gripper’s axes. The agent’s outputs are scaled appropriately to remain within these bounds, ensuring fine-grained yet flexible maneuverability. In real-world implementations, these actions typically drive inverse kinematics modules that translate high-level motion commands into precise joint movements.

By unifying these rigorously structured observation and action spaces, the RL framework is furnished with a rich representation of the gripper–target system, enabling it to handle the fluid and unpredictable nature of free-floating targets. This design allows agents to learn robust, high-fidelity policies that effectively align, track, and ultimately grasp free-floating objects in dynamic scenarios.

4.1.2 Reward Function Design

The choice of reward function is critical in shaping the agent’s behavior and overall training effectiveness within an RL framework. To investigate how different reward formulations influence learning outcomes, two distinct functions are introduced. Each function targets specific aspects of the grasping problem, guiding the agent to minimize the distance to the free-floating target while promoting a stable and carefully aligned pre-grasp configuration.

The first reward function is defined as:

$$R_1 = \omega_1 r_d + \omega_2 r_{\text{top}} + \omega_3 p_v, \quad (4.3)$$

where each term is weighted by a gain ω_i . Here, r_d is the Euclidean (L2) distance between the gripper and the target:

$$r_d = \sqrt{(x_g - x_t)^2 + (y_g - y_t)^2 + (z_g - z_t)^2}, \quad (4.4)$$

which encourages the agent to move closer to the target. The term r_{top} is a *topology-based* reward, shown in Fig. 4.1 assessing how many of the target’s key points are located within the *convex hull*

of the gripper’s links:

$$r_{\text{top}} = \frac{n_{\text{in}}}{N_p}, \quad (4.5)$$

where N_p is the total number of key points on the target object, and n_{in} is the count of those points falling inside the gripper’s convex hull. This reward promotes an action strategy that positions the gripper to “capture” as many key points as possible.

In this work, the gripper’s *grasping volume* is approximated by constructing a convex hull around its structural components, including the centers of each finger link. This polyhedral hull effectively represents the region that the gripper can enclose when its fingers close. Consequently, if any of the target’s key points lie within this volume, it indicates that the object is well-positioned for a potential grasp. By tracking the number of key points captured inside this hull, r_{top} incentivizes the gripper to move into an orientation and position that is more likely to succeed when the final grasping motion is executed.

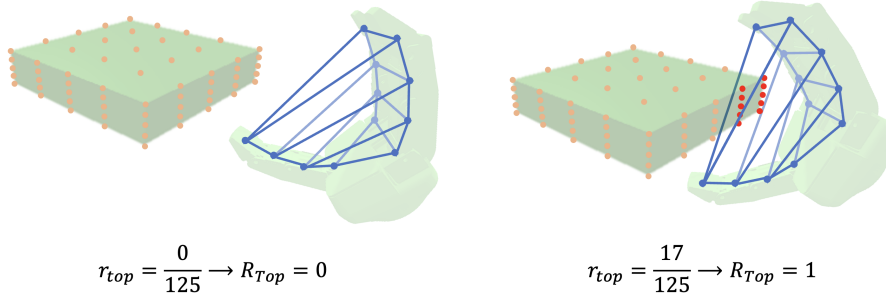


Figure 4.1: Illustration of the topology-based reward r_{top} , where target key points (shown in red dots) are evaluated against the gripper’s convex hull (generated by blue lines).

Lastly, p_v imposes a penalty if the gripper inadvertently increases the target’s velocity—an effect that could push the target away and complicate grasping. Together, these three terms reinforce distance reduction, stable alignment, and momentum control.

In the second reward function, the orientation of the gripper relative to the target and a *sparse* topology term become focal points:

$$R_2 = \omega_1 r_d + \omega_2 R_{\text{top}} + \omega_3 r_\theta, \quad (4.6)$$

where r_θ is the L2 norm of the orientation difference between the gripper and target:

$$r_\theta = \sqrt{(\phi_g - \phi_t)^2 + (\theta_g - \theta_t)^2 + (\psi_g - \psi_t)^2}, \quad (4.7)$$

encouraging the agent to align its orientation with the target. Meanwhile, R_{top} , introduced in Fig. 4.1, is the *sparse* variant of the topology term:

$$R_{\text{top}} = \begin{cases} 1 & \text{if } r_{\text{top}} > 0 \\ 0 & \text{otherwise} \end{cases}. \quad (4.8)$$

This modification triggers a binary reward for achieving a basic overlap of key points rather than proportionally rewarding partial overlaps. Consequently, the agent experiences a stronger incentive once it has entered the correct “capture zone,” bolstering confidence during training.

For both reward functions, the gains ω_1 , ω_2 , and ω_3 are set to 1.1, 1.0, and 1.5, respectively. These relative weights are chosen to balance the importance of distance minimization, precise alignment, and velocity control of the free-floating target. Although fine-tuning these values may be task-specific, the present selection provides a reasonable trade-off that emphasizes the gripper’s positioning and orientation without neglecting the management of target momentum.

In subsequent sections, I elaborate on the empirical impact of these reward formulations. By contrasting learning curves, stability, and final performance across differing reward parameters, insight is gained into how distinct grasping behaviors are fostered and the adaptability of the trained policy is influenced by each design.

4.1.3 Performance Evaluation and Algorithm Comparison

To evaluate the competence of three prominent policy-gradient algorithms—Deep Deterministic Policy Gradient (DDPG), Twin Delayed Deep Deterministic Policy Gradient (TD3), and Soft Actor-Critic (SAC)—in the task of reaching and grasping a free-floating object, I conducted extensive training trials in a zero-gravity simulation. The primary objective was twofold: first, to determine

whether these methods could learn to stably secure a free-floating target; and second, to assess their capability for generalizing to novel situations beyond their training experiences.

Each policy was trained for a cumulative total of 10,000,000 simulation steps, with each episode lasting 500 timesteps (equating to approximately 2.1 seconds in the PyBullet environment). The experiments were executed on a desktop system equipped with an Intel® Core™ i7-10600K, an NVIDIA GeForce® RTX 3070 GPU (with CUDA support), and 32 GB of RAM, thus allowing for parallelized computation. At the start of every episode, both the gripper and the free-floating target were placed at randomized initial positions and orientations. Algorithm 2 outlines the structure of each training episode in detail. Briefly, it initializes the environment, randomizes the gripper and target poses, and then iteratively selects actions, observes new states, computes rewards, and stores transitions. This loop continues until either the maximum step limit is reached or a termination condition is met.

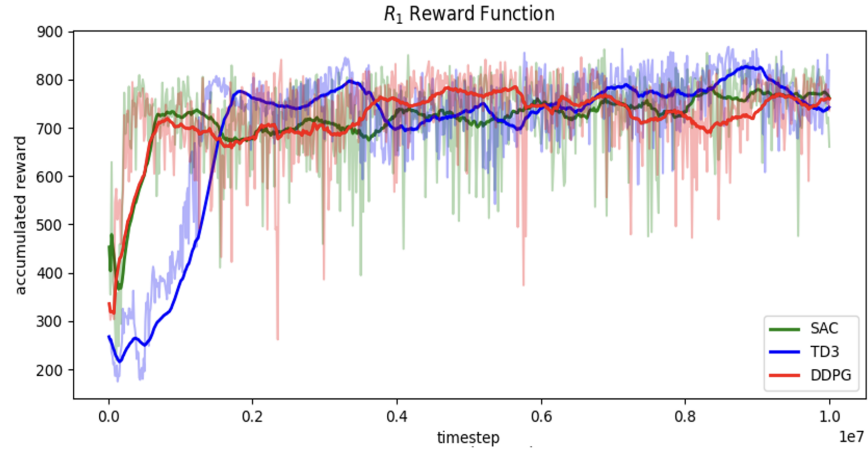
Algorithm 2 Training Episode Procedure

```

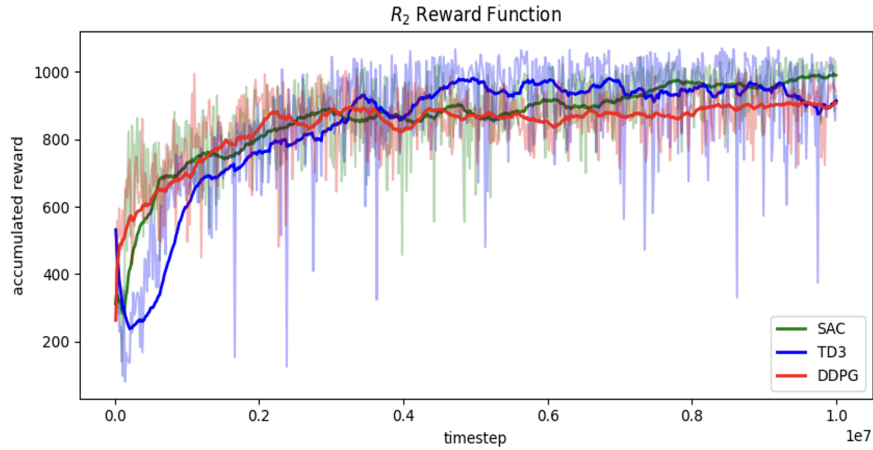
1: Initialize:
    • Initialize environment  $\mathcal{E}$ .
    • Randomize gripper position and orientation.
    • Randomize target pose around the gripper.
    • Set gravity to 0.
    • Initialize  $\text{step\_counter} \leftarrow 0$  and  $\text{done} \leftarrow 0$ .
2: while  $\text{done} = 0$  do
3:   Select Action: Choose action  $\mathbf{a}$  from the current policy and add exploration noise.
4:   Execute Action: Apply  $\mathbf{a}$  to the gripper through inverse kinematics.
5:   Observe State: Receive the next state  $\mathbf{s}'$ .
6:   Compute Reward: Calculate reward  $r$  based on the reward function  $\{R_1, R_2\}$ .
7:    $\text{step\_counter} \leftarrow \text{step\_counter} + 1$ .
8:   if  $\text{step\_counter} \geq \text{max\_steps}$  then
9:      $\text{done} \leftarrow 1$ .
10:  end if
11:  if  $\text{done} = 1$  then
12:    Terminate Episode
13:    break
14:  end if
15:  Store Transition: Save  $(\mathbf{s}, \mathbf{a}, r, \mathbf{s}', \text{done}, \text{info})$  to replay buffer.
16:   $\mathbf{s} \leftarrow \mathbf{s}'$ .
17: end while

```

Both reward functions R_1 and R_2 (see Eq.(4.3) and Eq.(4.6)) were experimented with for each of the three algorithms. Fig. 4.2 illustrates the learning curves, showing that all methods converge within approximately 4,000 episodes. Notably, SAC displays a smoother convergence profile and attains higher peak average rewards, suggesting that its entropy-regularized framework is more robust to unpredictable states or disturbances. TD3's delayed policy updates and double-Q technique appear to have a conservative learning style, which, while beneficial for stability, can slow down convergence. In contrast, DDPG's more aggressive approach leads to rapid early improvements, yet it may be less stable when exposed to abrupt changes or unseen configurations.



(a)



(b)

Figure 4.2: Learning curves of DDPG, TD3, and SAC trained with (a) Reward function R_1 . (b) Reward function R_2 . Curves represent the moving average of episode rewards over training timesteps.

The sparse topology element in R_2 offers a clear performance edge by rewarding the agent once it aligns the gripper such that *at least one* key point of the target falls inside its convex hull. This binary feedback fosters a more decisive, confidence-driven policy, which tends to produce a higher average reward than the denser but more finely scaled R_1 . Nevertheless, when the target and the gripper’s fingers have shapes or spacing that make it impossible to encompass all available key points, R_1 can still guide incremental improvements by continuously rewarding partial overlap.

Fig. 4.3 captures snapshots of the policy’s behavior across training epochs. Early in training, the agent focuses heavily on reducing the gripper–target distance, unintentionally neglecting orientation alignment or topological coverage. As it becomes aware of orientation mismatches, it begins to adjust pitch, roll, and yaw to match the target’s pose more closely. Eventually, the agent learns to incorporate the topological criterion, maximizing the number of target key points lying within the gripper’s convex hull. By carefully balancing all these objectives, the agent refrains from pushing the target away (which would increase its velocity and risk losing it) and instead maneuvers into a pre-grasp configuration that synergizes low distance, proper orientation, and ample key-point containment.

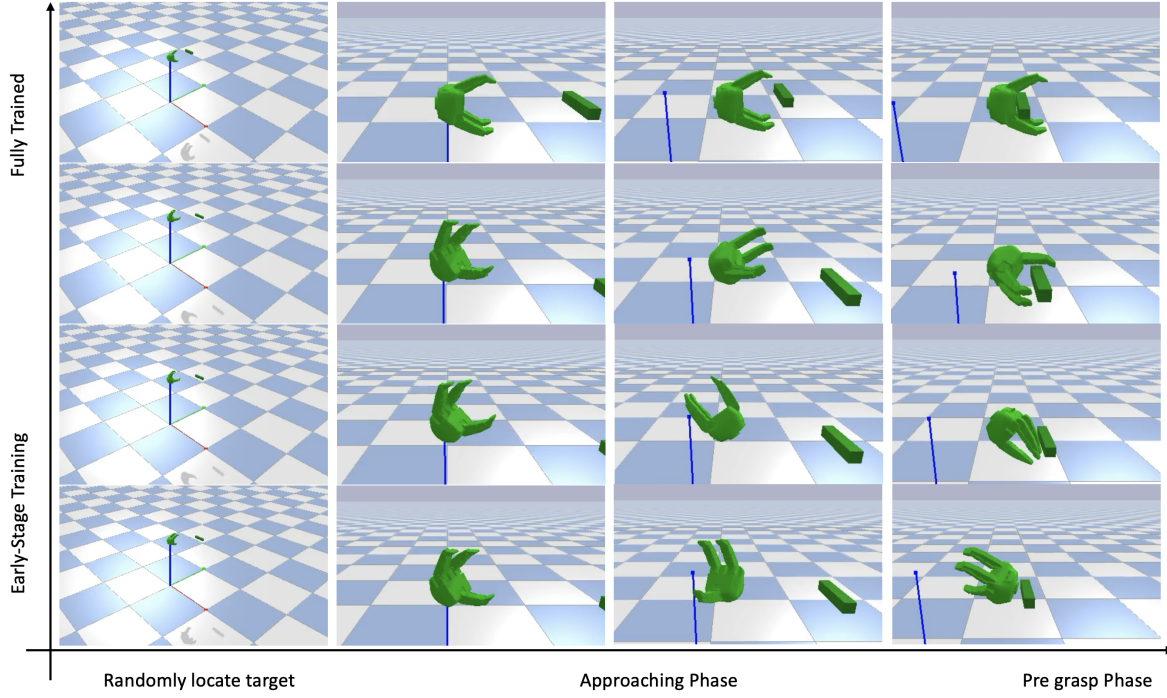


Figure 4.3: Progressive policy improvement for a free-floating target grasp scenario. (1) The agent initially prioritizes distance minimization; (2) orientation alignment is then introduced; (3) final strategies incorporate topological constraints to ensure a robust pre-grasp.

Once the agent achieves an appropriate positioning, a final “close fingers” command can be triggered. The resulting grip securely encompasses the free-floating target, reflecting the agent’s learned ability to fulfill multiple objectives: distance minimization, orientation alignment, and topological conformance. Through these iterative refinements, the agent demonstrates heightened adaptability and resilience, showcasing the potency of reward design in shaping complex robotic behaviors. This underscores the practical potential of RL-based controllers for effectively handling free-floating objects, particularly in domains where precision and minimal disturbance are paramount.

4.1.4 Conclusion

In conclusion, the comparative analysis of multiple DRL algorithms and reward function configurations confirms the viability of training a continuous-control agent for robotic grasping tasks. These

findings highlight the robustness of RL-based approaches in balancing competing objectives—such as distance minimization, orientation alignment, and topological coverage—while efficiently adapting to novel and dynamic scenarios. As a result, this work underscores the potential of DRL to serve as a practical solution for complex manipulation challenges in robotics. By leveraging the flexibility and adaptability of RL frameworks, robotic systems can be empowered to autonomously handle intricate tasks, such as grasping free-floating objects, with precision and reliability. The seamless integration of machine learning methodologies into advanced robotic control systems is thus paved by this research, enabling the development of more capable robotic platforms.

4.2 Domain Randomization in Reinforcement Learning for Pre-Grasp of Free-Floating Targets

In this section, I explore how domain randomization can empower an RL-based system to autonomously pre-grasp a free-floating, moving target. Central to this investigation is the `Fanuc_Robotiq_Grasp` environment, which systematically randomizes the initial positions and velocities of both the target and the gripper, thus demanding a highly adaptable control policy. Using a Robotiq 3F gripper, the agent must track and align with the drifting object to reach an optimal pre-grasp pose. Despite the complexities of free-floating dynamics —where even slight forces can induce substantial motion— the empirical results show that the learned policies consistently achieve precise pre-grasp positioning. This underscores the capacity of domain randomization, when combined with RL, to yield robust behaviors in dynamically uncertain scenarios, paving the way for reliable sim-to-real transfer in advanced robotic manipulation tasks.

4.2.1 Definition of Actions and States

In advanced robotic manipulation scenarios, including those requiring the interception and pre-grasp of free-floating objects, it is critical to design both the state and action spaces to capture the full complexity of the system. Unlike grasping tasks on a static tabletop, intercepting a moving target in free space adds substantial levels of uncertainty and dynamic constraints. Therefore, the observations (states) must be both sufficiently informative and straightforward to acquire from real robotic hardware.

To facilitate a direct transfer of the learned policy from simulation to real-world platforms with minimal domain shifts, I define *all* poses (of the object and the gripper) in a common inertial reference frame. Specifically, the state space includes the full six-degree-of-freedom (6-DoF) pose and velocity of the gripper, as well as the pose and velocity of the free-floating target. In order to robustly track the object in flight and predict intercept trajectories, the *relative* pose and velocity

between the target and the gripper are also incorporated. Additionally, the minimum distance between the target and the gripper’s palm is tracked to keep the agent aware of imminent collisions. Finally, direct feedback on unwanted or excessive forces upon contact is provided by the total contact force as measured by the tactile sensors. Together, these features create a 40-dimensional observation vector:

$$S = \{G_p, G_v, T_p, T_v, \delta_p, \delta_v, d, F_c\} \in \mathbb{R}^{40}, \quad (4.9)$$

where G_p and G_v denote the gripper’s pose and velocity, T_p and T_v the target’s pose and velocity, δ_p and δ_v the relative pose and velocity, d the minimum distance to the gripper’s palm, and F_c the total contact force sensed by the tactile array. By incorporating these elements, the agent acquires a comprehensive representation of the environment, allowing it to strategically approach and intercept the moving target.

The action space is formulated to capture the continuous 6-DoF movements of the gripper’s end-effector. Each action dimension, illustrated in Eq.(4.10), corresponds to a permissible translational or rotational displacement within the gripper’s local coordinate frame:

$$a = [a_x, a_y, a_z, a_\phi, a_\theta, a_\psi] \in \mathbb{R}^6. \quad (4.10)$$

At every time step, the agent can command a translation of up to 10mm along each axis and a rotation of up to 0.1 rad around each rotational axis. Internally, the network outputs continuous values in the range $[-1, 1]$, which are mapped to the corresponding physical scale for each translational and rotational DoF. These commands are then converted into joint-space commands via the robot’s forward kinematics, such that the manipulator’s end-effector—the gripper’s base—executes the specified motion. This continuous formulation grants the policy the flexibility to realize both subtle adjustments and more aggressive maneuvers as required by the current state.

4.2.2 Reward Shaping

A critical element in RL is the design of the reward function, which can greatly influence both the convergence speed and the final performance of the policy [205]. In this work, I propose a reward structure that seamlessly balances dense feedback for efficient learning with sparse penalties or bonuses that reflect crucial constraints in free-floating grasp tasks. By carefully blending these two reward styles, I ensure that the agent remains incentivized to track and align with the target while preserving safe operational constraints such as avoiding undesired collisions.

$$R = R_d + R_\theta + R_{\text{Top}} + P_F, \quad (4.11)$$

where R_d and R_θ are dense rewards that enhance the agent’s proficiency in approaching and orienting towards the target, and R_{Top} along with P_F function as more *sparse* components. Each of these components is deliberately constrained to lie between -1 and 1 . This design strategy ensures that no single component disproportionately dominates the reward unless its activation threshold is triggered, simplifying hyperparameter tuning and improving interpretability.

Dense Rewards for Precision Alignment. In order to promote accurate navigation and orientation towards the moving object, I employ the following dense terms:

$$R_d = 1 - \tanh(\|\delta_p\|), \quad (4.12)$$

$$R_\theta = 1 - \tanh(\|\delta_\theta\|), \quad (4.13)$$

where $\|\delta_p\|$ is the Euclidean (L2) norm of the gripper-target position difference, and $\|\delta_\theta\|$ is the orientation mismatch, often computed via a suitable norm of the rotation representation (e.g., angle-axis). Hence, R_d and R_θ decrease monotonically from 1 to values near 0 as $\|\delta_p\|$ and $\|\delta_\theta\|$ grow large, shown in Fig. 4.4. This mechanism shapes the agent’s behavior to systematically reduce pose

deviation, resulting in more consistent and precise trajectories.

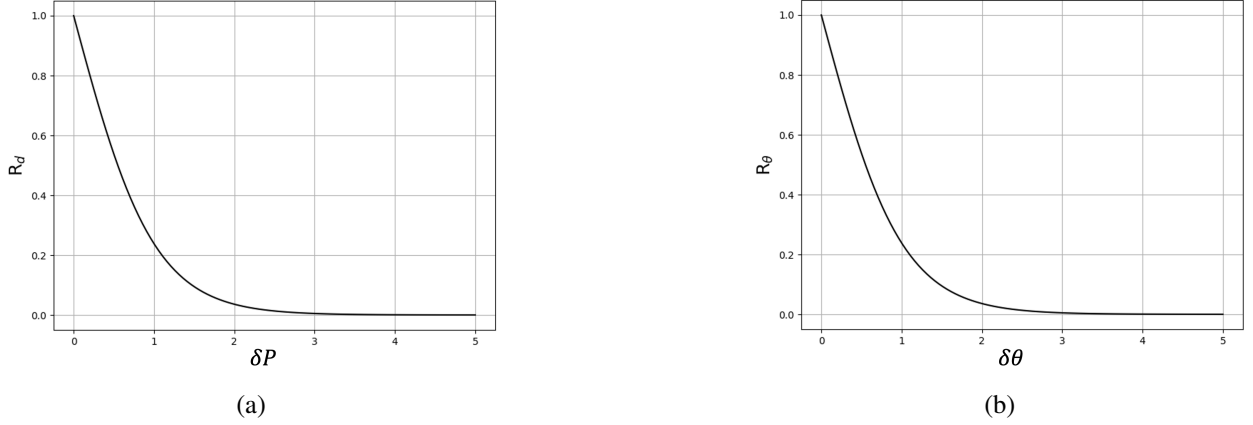


Figure 4.4: Dense reward components for (a) position alignment R_d and (b) orientation alignment R_θ . The rewards decrease monotonically as the gripper deviates from the target in position or orientation.

Sparse Rewards for Grasp Feasibility and Safety. In addition to the dense terms, I incorporate terms that reflect critical events in the grasp cycle:

$$R_{\text{Top}} = \begin{cases} 1, & \text{if } n_{\text{in}} \geq 1 \\ 0, & \text{otherwise} \end{cases}, \quad (4.14)$$

where N_p is the total number of key points on the target (e.g., corners, surface markers) and n_{in} denotes how many of these key points lie within the gripper's convex hull (Fig. 4.1). As soon as at least one key point of the object is enclosed, the agent receives a positive sparse reward. This fosters faster discovery of valid grasp postures.

A simultaneous penalty term P_F prevents the gripper from applying excessive force:

$$P_F = \begin{cases} 0, & \text{if } F_c = 0 \\ -1, & \text{if } F_c > 0 \end{cases}, \quad (4.15)$$

where F_c is the sum of the contact forces measured by the tactile sensors on the gripper. Thus, any physical interaction with the object yields a penalty, discouraging the agent from forceful collisions or prematurely clamping onto the target. This term is particularly important in free-

floating scenarios: even a brief period of contact could push the object away or destabilize its motion. As depicted in Fig. 4.5, the penalty P_F is applied whenever a contact force is detected.



Figure 4.5: Graphical representation of the contact penalty term P_F . The penalty is applied when any contact force is detected by the gripper’s tactile sensors, thereby discouraging excessive force during pre-grasp.

By integrating both dense and sparse terms, this reward formulation promotes the intermediate steps necessary for stable pre-grasp alignment while simultaneously incentivizing correct enclosure and discouraging undesirable contact. Crucially, each subcomponent is scaled identically (-1 to 1), which removes the need to manually tune separate weighting factors. As supported by experimental trials, this reward structure enables the agent to distinguish suboptimal attempts (e.g., incomplete enclosure, unintended forceful collisions) at an early stage of learning and reliably converge to policies that achieve precise and safe grasps. This approach ultimately reduces the agent’s likelihood of missing or deflecting free-floating objects and increases the probability of stable, real-world transfer without extensive post-training fine-tuning.

4.2.3 Performance Evaluation

To optimize reward structures and cultivate a robust policy, it is imperative for the robotic gripper to comprehensively explore the entirety of its operational action space. The utilization of simulation environments plays a pivotal role in systematically evaluating a diverse array of scenarios. While hardware-in-the-loop (HIL) configurations may yield more dependable outcomes, simulations

provide a rapid and secure platform for the extensive exploration of numerous potential actions. RL algorithms typically presuppose the ability to reset the system after each episode, access observations across all states, and compute rewards at every timestep—conditions that are frequently challenging to achieve in real-world applications [206].

The primary objective of this study is to rigorously assess the proposed methodology’s capability to identify optimal pre-grasp positions relative to dynamic targets and its adaptability to novel environments. For training the policy, an extensive dataset comprising 40,000 episodes was employed, with each episode encompassing 500 timesteps. At the commencement of each episode, the target is randomly positioned and endowed with a constant velocity, constrained to a maximum relative velocity of 0.4 m/s with respect to the gripper.

In the realm of RL, the success rate serves as a critical performance metric, ranging from zero to one, and encapsulates the agent’s consistency in achieving specified goals within a defined environment. In this research, the success rate is interpreted as the gripper’s ability to secure more than two positive rewards over 200 consecutive steps, thereby ensuring effective engagement in the grasping process. Consequently, the gripper dynamically adjusts its position and orientation to maximize dense rewards, as delineated by Eq.(4.12) and Eq.(4.13), while strategically positioning a key point within its grasp range without initiating contact with the target.

The agent is furnished with an enriched state space and comprehensive contact data, facilitating the learning of relative pose maintenance concerning the target and optimal pre-grasp positioning, all while meticulously avoiding unintended contacts that could potentially displace the target. Throughout the training phase, the model’s progress is periodically saved every 40 episodes. The most proficiently trained agent is subsequently evaluated through five trials, wherein it must approach a free-floating target exhibiting random initial states. Training convergence is achieved after 24,000 episodes, as illustrated in Fig. 4.6. The trained model demonstrates a commendable average reward and an impressive success rate, achieving a convergence metric of 0.91, thereby underscoring its robust learning efficiency across stochastic scenarios.

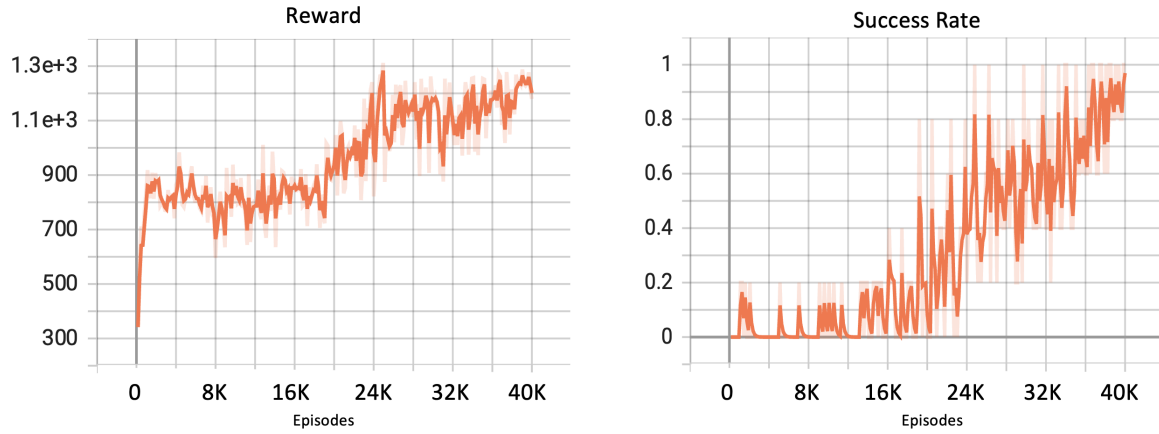


Figure 4.6: Training results in simulation environment. Mean evaluation results of the best-trained agent over five random episodes.

To rigorously evaluate the trained agent’s performance within a simulated environment, it is essential to subject it to a multitude of randomized scenarios and meticulously observe the gripper’s ability to target and grasp the object. Upon loading the agent for evaluation, all pertinent observations from the simulation are aggregated and fed into a prediction function—specifically, the trained actor network θ —thereby facilitating a streamlined input-output process. Leveraging its accumulated experience, the agent determines the most advantageous actions, which the gripper subsequently translates into precise movements. The simulation outcomes are depicted in Fig. 4.7 and Fig. 4.8, encompassing two distinct random episodes.

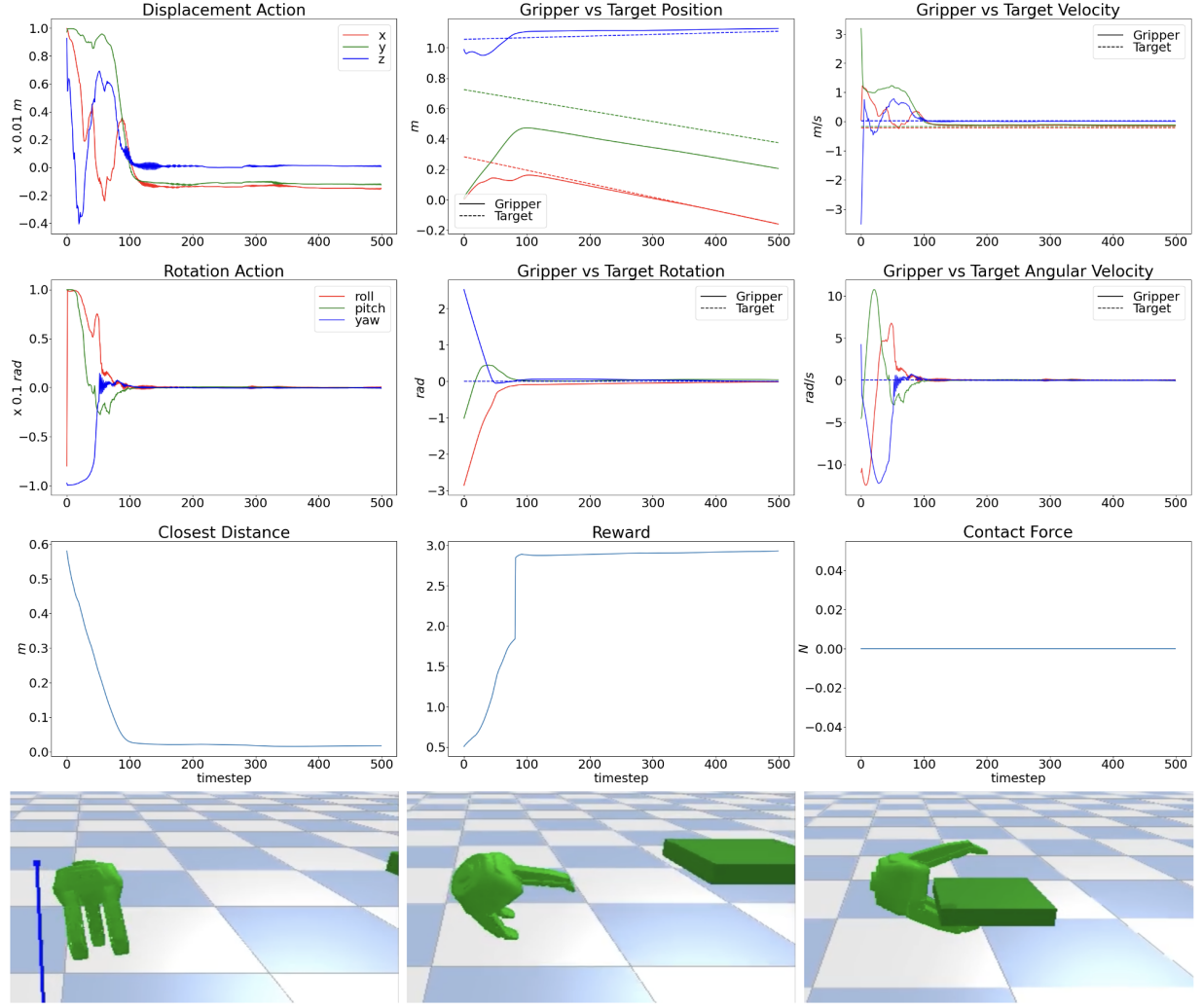


Figure 4.7: Evaluation results of the best-trained agent. The term “closest distance” refers to the minimum distance between the target and the gripper’s palm, while “position” indicates the location of their center of mass. The gripper successfully positions itself in the pre-grasp stance without making contact with the free-floating moving target, and it aligns its velocity accordingly.

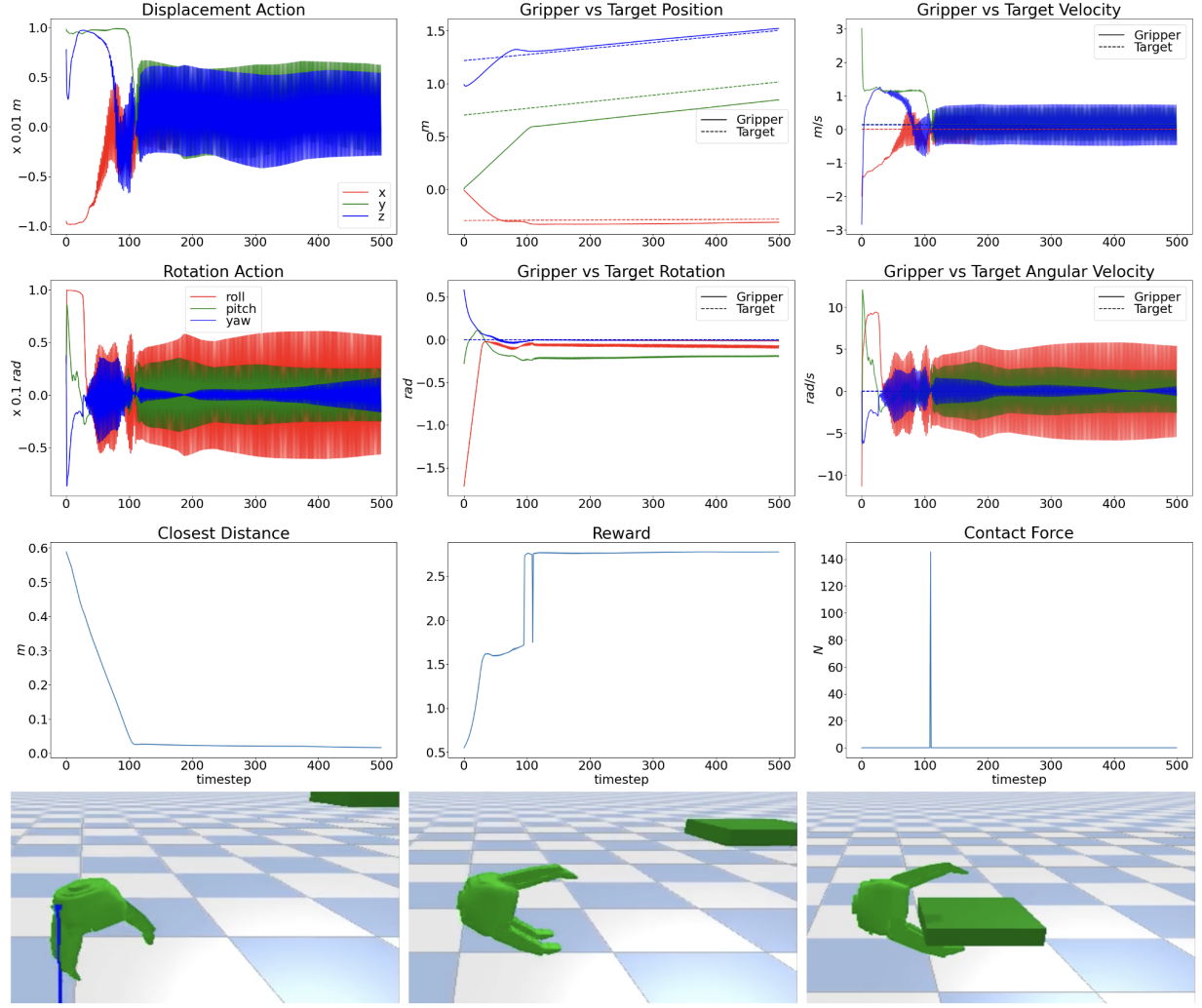


Figure 4.8: Evaluation results of the best-trained agent. Although there is slight contact between the target and the gripper’s finger, the agent manages this effectively.

Both episodes achieved success by satisfying the predefined criterion: maintaining a reward exceeding 2.0 for over 200 consecutive timesteps. In episode A, the target’s initial position was $(0.28, 0.72, 1.05)$ meters with an initial velocity of $(-0.21, -0.17, 0.03)$ m/s, whereas episode B commenced with the target at $(-0.29, 0.70, 1.21)$ meters and an initial velocity of $(0.01, 0.15, 0.14)$ m/s. The gripper’s starting position remained consistently at $(0.0, 0.0, 1.0)$ meters across both episodes; however, the initial orientations varied, with episode A initiating at $(-2.86, -1.02, 2.52)$ radians and episode B at $(-1.71, 0.28, 0.58)$ radians. In episode A, the actions executed by the gripper resulted in smooth translational displacements and rotational adjustments that effectively

guided the agent towards the target. Conversely, in episode B, the actions exhibited oscillatory behavior around the ostensibly optimal action, attributable to the incorporation of a 10% noise factor in each action choice. This stochastic perturbation was intentionally introduced to enhance policy exploration. The resultant oscillations, while indicative of the agent’s adaptive strategies to mitigate noise-induced variance, did not adversely impact the gripper’s positional or orientational stability, only affecting its velocity parameters.

Upon comprehensive analysis, it was observed that episodes with elevated relative velocities—up to 0.4 m/s—corresponded with higher failure rates. Intuitively, an increase in target velocity augments the probability of the gripper missing the target. To elucidate the influence of velocity variations on performance metrics, Table 4.1 presents the outcomes at differing maximum velocities.

Table 4.1: Performance Metrics at Varying Maximum Relative Velocities

Max Relative Velocity (m/s)	Mean Eval Reward	Mean Reward std	Success Rate
0.1	1240.11	80.86	0.96
0.2	1219.57	103.20	0.93
0.3	1208.05	185.01	0.89
0.4	1167.13	131.05	0.91

For evaluation purposes, the trained agent was subjected to 100 distinct random episodes, revealing that a reduction in maximum relative velocity generally correlates with improvements in both mean reward and success rate. This highlights a critical dependency between relative velocity and performance efficacy. Nonetheless, inherent randomness in episode parameters, such as initial positions and orientations, introduces variability in outcomes, suggesting that while overarching trends are discernible, individual episode results may exhibit deviations.

While overall success exceeded 90%, the agent did fail in situations where the target’s velocity is close to 0.4 m/s or when noisy inputs led to abrupt target’s pose changes. In these instances, the learned policy sometimes overshoot the safe approach corridor, resulting in an uncontrolled bounce.

Addressing these failures requires more robust exploration strategies or additional reward shaping for collision avoidance.

In summary, the trained agent exhibits high proficiency in grasping free-floating targets under varying dynamic conditions, demonstrating robust adaptability and efficiency. The correlation between relative velocity and performance underscores the need for optimizing target dynamics to enhance gripper reliability.

4.3 The Importance of Tactile Feedback in Pre-Capture Planning with Noisy Observations

In this work, I investigate the role of tactile sensors in a pre-grasp task and explore how excluding this feedback during training influences the learned policy. Specifically, two agents with identical architectures and training conditions are compared: one agent receives tactile sensor measurements, while the other is deprived of tactile feedback. Although the objective in the pre-grasp phase is to carefully guide the gripper toward an optimal pose relative to the rotating and translating target without making contact, the inclusion of tactile information during training provides the agent with critical insight that substantially improves performance. Beyond examining the benefits of tactile feedback, I also study the robustness of the trained agent to noisy observations. This is particularly relevant in real-world deployments where sensing systems, such as cameras, introduce inherent uncertainty.

4.3.1 Comparison of Agents with and without Tactile Feedback

The two agents considered share the same environment and reward structure, ensuring that performance differences can be attributed solely to the availability of tactile feedback. Both agents rely on the SAC algorithm, receiving a set of state variables that includes the gripper’s pose and velocity, the target’s pose and velocity, their relative differences, and the minimum directional distance between them. Concretely, these parameters form a 39-dimensional observation space for the agent without tactile sensing, while the agent equipped with tactile feedback gains a fortieth dimension capturing the accumulated contact force information. This additional tactile sensor reading serves as an early indicator of incidental contact, offering critical insight that can prevent unintended forces on the free-floating target.

$$S_{\text{no_tactile}} = (G_p, G_v, T_p, T_v, \delta_p, \delta_v, d) \in \mathbb{R}^{39}, \quad (4.16)$$

$$S_{\text{tactile}} = (G_p, G_v, T_p, T_v, \delta_p, \delta_v, d, F_c) \in \mathbb{R}^{40}. \quad (4.17)$$

By examining the training curves of these agents, it becomes evident that tactile feedback confers several advantages. First, the presence of contact-force information alerts the agent to incipient interactions with the free-floating target, effectively reducing the likelihood of inadvertently applying large forces that could repel or disturb the target. Second, as shown in Fig. 4.9, the tactile-sensing agent consistently achieves higher rewards and exhibits faster convergence toward an effective policy. The ability to sense and learn from contact signals enables it to position itself in a pre-grasp configuration more reliably while steering clear of detrimental collisions.



Figure 4.9: Accumulative moving average of rewards over the training episodes for both tactile-sensing and non-tactile agents. The tactile-based agent exploits contact feedback to more accurately adjust its pose, leading to a higher overall performance and faster convergence.

In terms of success rate—defined here as the agent’s ability to maintain a sufficiently advantageous pre-grasp pose without contacting the target for at least 200 consecutive timesteps—the tactile-sensing agent outperforms its non-tactile counterpart. Fig. 4.10 illustrates that the tactile-equipped agent not only achieves a higher peak success rate but also sustains it over a more

extensive range of training epochs, underscoring the critical role of tactile input when dealing with free-floating, lightweight objects in microgravity environments.

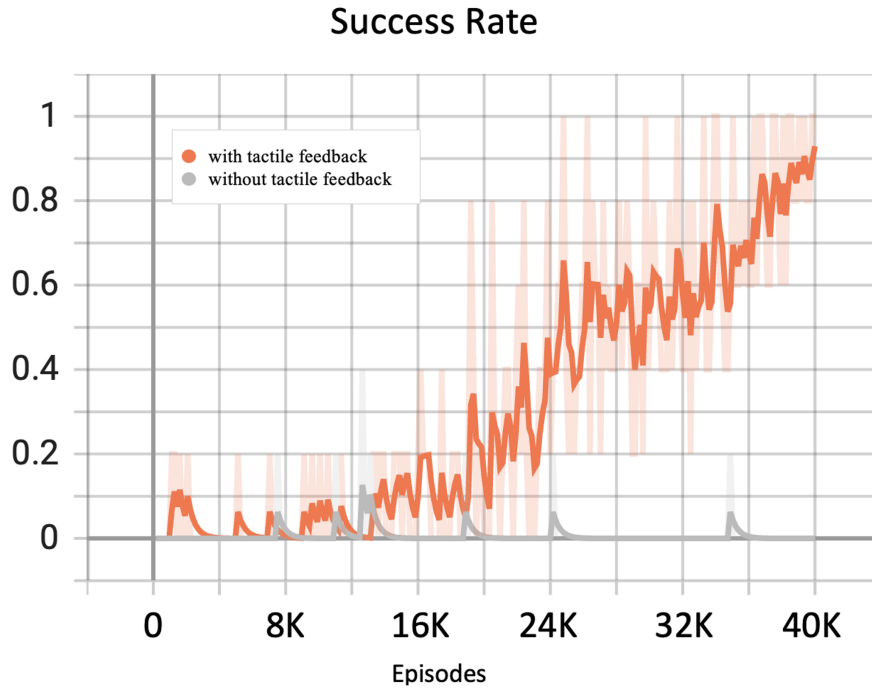


Figure 4.10: Success rate comparison over the training process for two agents: one equipped with tactile sensor feedback and another lacking it.

Snapshots of typical episodes (see Fig. 4.11) provide further insight into the different strategies employed by the agents. The tactile-sensing agent utilizes the contact cue to correct its trajectory and avoid pushing the target away. Meanwhile, the non-tactile agent oscillates between making forceful contact (risking target disturbance) and remaining overly cautious (thus missing the surrounding reward). These contrasting behaviors highlight how even minimal contact information can lead to substantially different, and ultimately more successful, control policies during pre-grasp maneuvers.

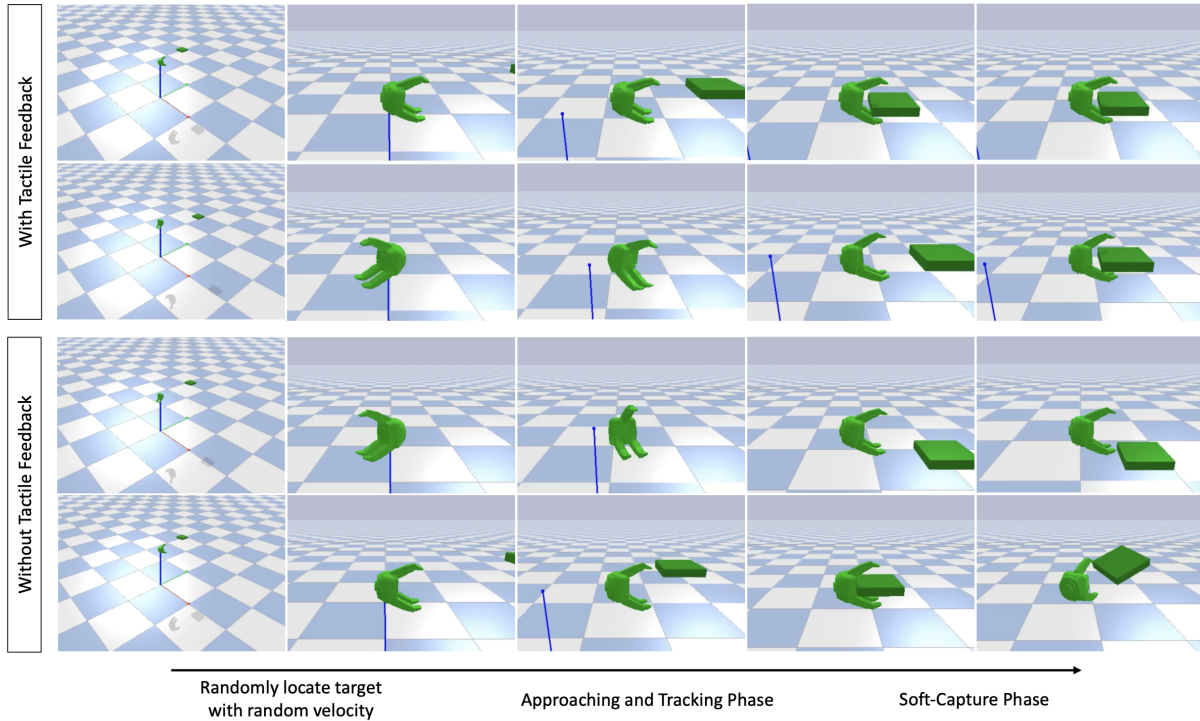


Figure 4.11: Comparative snapshots of pre-grasping maneuvers. Each column illustrates an agent’s behavior under two different scenarios. The tactile-sensing agent effectively positions itself relative to the free-floating target, reducing unwanted collisions. The non-tactile agent either pushes the object away or stays too distant, compromising task success.

A random episode analysis (see Fig. 4.12 and Fig. 4.13) reinforces these findings. The tactile-sensing agent maintains near-maximal rewards for long stretches of time, well beyond the 200-step threshold required for a successful pre-grasp. In contrast, the non-tactile agent initially aligns itself acceptably but eventually makes contact, reducing its reward and causing the target to drift off its nominal trajectory. This loss of proximity inflicts a cascading penalty, causing the agent to forfeit additional rewards.

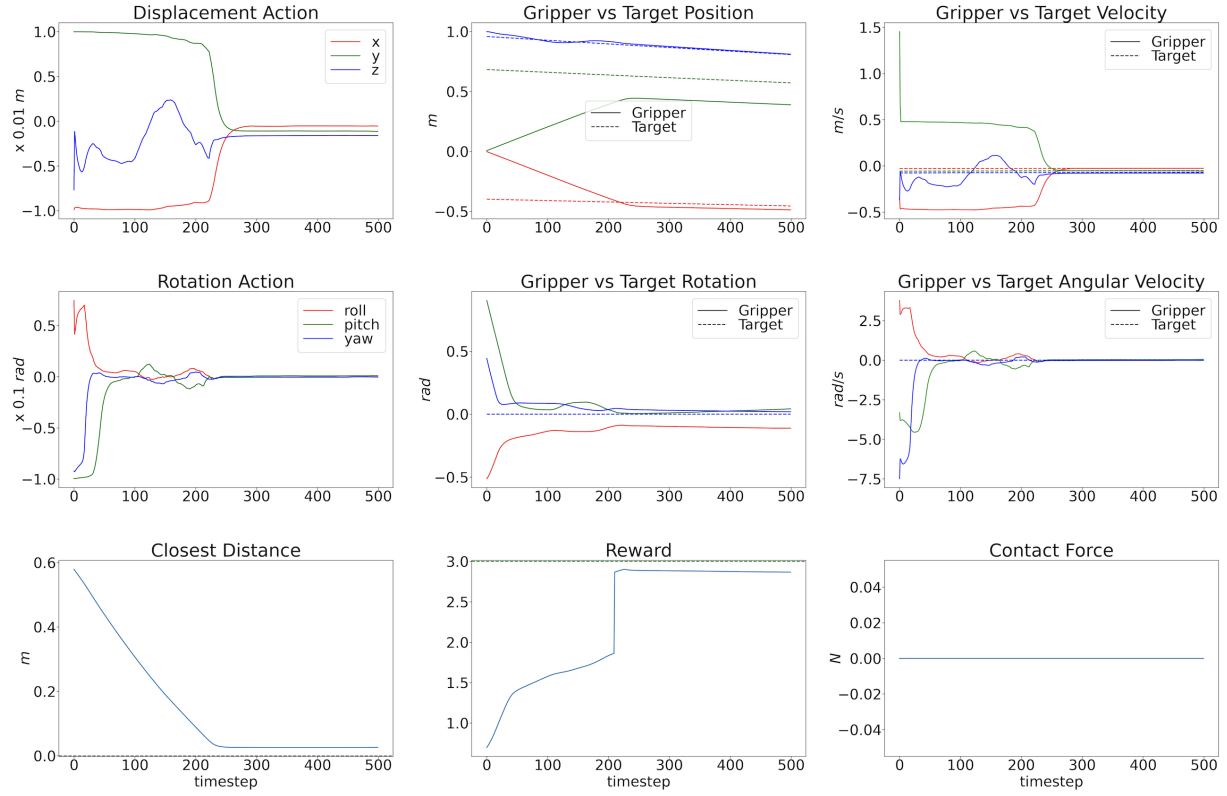


Figure 4.12: Random episode outcomes for each trained agent. The tactile-sensing agent attains near-maximum rewards consistently, verifying its ability to safely align with the target.

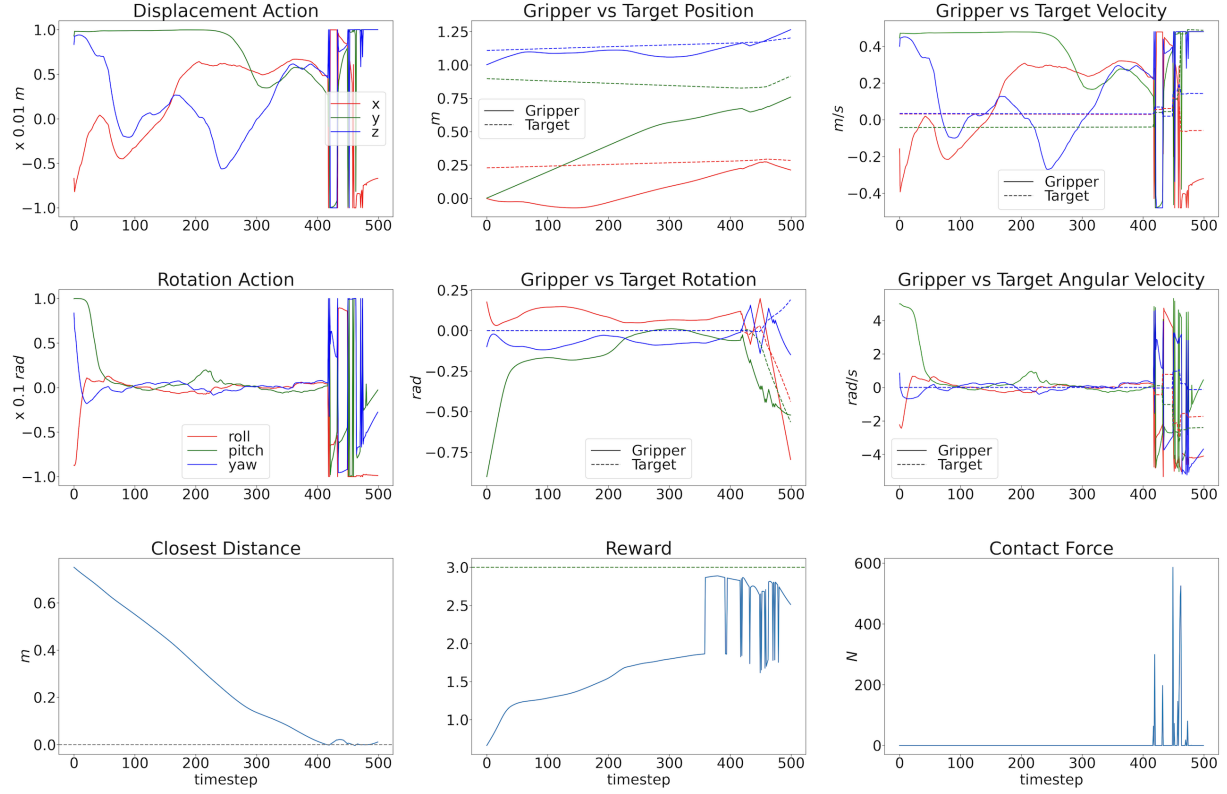


Figure 4.13: Random episode outcomes for each trained agent. The non-tactile agent occasionally reaches high reward levels but inevitably contacts the target, causing it to separate and leading to task failure.

4.3.2 Performance Under Noisy Observations and Robustness

Real-world sensing systems seldom provide perfectly accurate data, particularly in space-based operations where object tracking heavily relies on camera or other remote-sensing technologies prone to stochastic noise. To simulate this scenario, Gaussian noise with a mean of zero and standard deviations of $\sigma = 0.01$ and $\sigma = 0.02$ is introduced into the agent's observations of the target's pose and velocity. Fig. 4.14 shows the distribution of these noise samples for $N = 1000$, illustrating how the perceived position of the target could deviate by up to 0.03 m for $\sigma = 0.01$ and 0.06 m for $\sigma = 0.02$. Such deviations can be substantial in delicate tasks, especially for microgravity missions where contact forces must remain minimal to avoid disturbing the target.

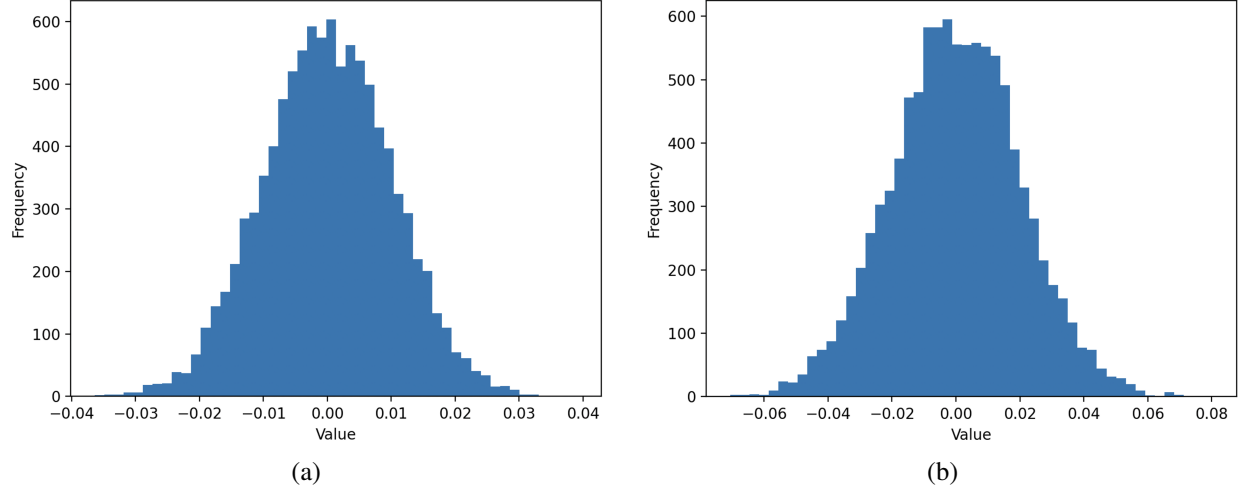


Figure 4.14: Distribution of 1,000 Gaussian noise samples with zero mean and different standard deviations: (a) 0.01 and (b) 0.02. These noise levels represent realistic sensor noise, shifting the agent’s view of the target, challenging accurate trajectory planning.

To evaluate the agent’s robustness, I tested the tactile-sensing agent on 100 random episodes under each noise condition. Table 4.2 compares the average reward and success rate as noise levels increase. While the agent’s performance naturally declines when sensor noise grows, the drop remains moderate, underscoring the agent’s ability to adapt and maintain a successful pre-grasp strategy even under substantial observation errors. Notably, the success rate remains above 0.85, demonstrating the potential for real-world feasibility where noisy sensor data is unavoidable.

Table 4.2: Tactile-sensing agent performance under increasing levels of Gaussian noise.

Noise Level	Mean Reward ($\pm$ std)	Success Rate
No Noise	2519 ± 152.3	0.92
$\mu = 0, \sigma = 0.01$	2415 ± 159.9	0.89
$\mu = 0, \sigma = 0.02$	2345 ± 183.4	0.85

A more detailed look at a sample noisy episode reveals how the agent strategically mitigates the repercussions of erroneous position and velocity measurements. As illustrated in Fig. 4.15, even when the commanded displacements or orientations occasionally deviate due to noise, the overall trajectory remains sufficiently accurate to achieve a successful pre-grasp. In effect, the policy

learned by the agent compensates for small fluctuations in the observed target state, leveraging tactile feedback to refine its final pose and prevent destabilizing contact forces.

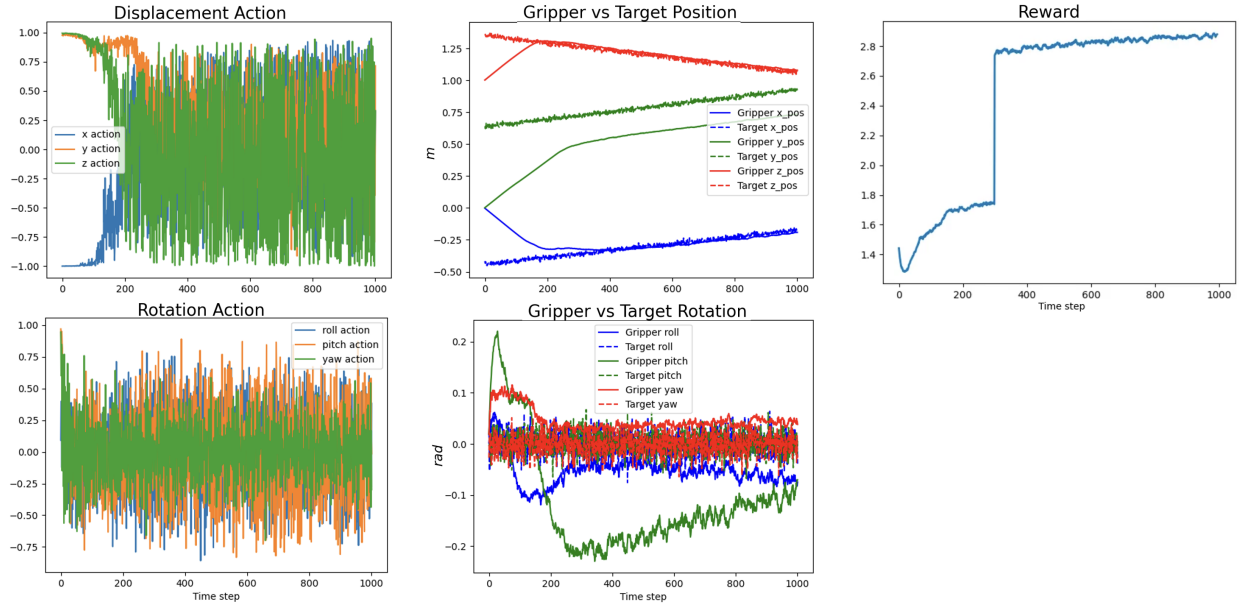


Figure 4.15: Tactile-sensing agent’s trajectory during a representative noisy episode. Despite receiving noisy target states and consequently exhibiting momentary deviations, the agent consistently recovers and maintains a reliable pre-grasp configuration, demonstrating robust performance in uncertain environments.

Overall, the simulation tests confirm that tactile feedback significantly enhances the agent’s proficiency in pre-grasp tasks by enabling it to learn a policy that skillfully balances the objective of maximizing rewards with minimizing target disturbance. Moreover, the agent demonstrates commendable robustness to noisy observations, suggesting that these findings are transferable to real-world scenarios where sensor noise is inevitable. This robustness, combined with the adaptability conferred by tactile sensing, can serve as a crucial step forward in developing reliable capture strategies for free-floating objects in microgravity.

4.4 Curriculum-based Reinforcement Learning for Pre-Grasp of Rotating Free-Floating Targets

Autonomous positioning of a robotic gripper relative to a free-floating object exhibiting both translational and rotational motion presents a significantly more complex challenge compared to conventional grasping scenarios. The inherent unpredictability and dynamic behavior of the target, especially under microgravity conditions, necessitate sophisticated control strategies to achieve precise alignment and secure pre-grasp configurations. In this subsection, I introduce a Curriculum-based RL framework designed to enhance the agent’s adaptability and performance in handling tumbling and rotating targets. By progressively increasing task complexity through Curriculum Learning (CL), the agent develops robust control strategies capable of managing dynamic and unpredictable target motions.

4.4.1 Task Definition

Autonomous positioning of a robotic gripper relative to a free-floating object exhibiting both translational and rotational motion presents a significantly more complex challenge compared to conventional grasping scenarios. In this work, the problem of ensuring a stable and precise pre-grasp configuration for a free-floating moving and rotating target in a microgravity setting is tackled. Specifically, the Robotiq 3F gripper is tasked with continuously adjusting its position, orientation, and velocity to align with a free-floating object, despite its unpredictable and dynamic behavior.

Such a task is highly relevant to applications like space debris removal, where objects often exhibit erratic motion. The free-floating nature of the object makes it extremely susceptible to external forces, meaning that even small disturbances can appreciably alter its trajectory. Under these conditions, standard RL approaches often struggle to handle fast tumbling and random motion. To address these difficulties, I employ CL to gradually introduce increasing levels of complexity. Starting with simpler scenarios focused on basic positioning, the agent learns core skills before

progressing to handle full 6-DoF rotational and translational dynamics. Ultimately, the goal is for the gripper to adapt its approach and velocity, ensuring a successful pre-grasp configuration even under the most challenging rotational velocities.

4.4.2 States and Actions

In the SAC approach, the choice of states and actions is critical. This selection intensely influences the agent’s understanding of its environment and its decision-making capabilities, directly impacting learning efficiency and performance. Therefore, a precise and thoughtful definition of states and actions is vital for optimizing the effectiveness of the SAC agent in complex, continuous action spaces.

The state S representation combines several parameters with more focus on positioning parameters: the pose and velocity of the gripper (p_G, v_G), the pose and velocity of the target (p_T, v_T), their respective differences, and the minimum distance in each direction between them, all defined within the inertial frame. This results in a 39-dimensional state space for the agent, filled with all positional information. It is assumed that the positional information of the target is accessible and supplied to the agent. The final critical input for the agent is the cumulative normal contact force exerted on the gripper, making the overall states inputs a 40-dimensional array, as discussed in detail in subsection 4.2.1.

Regarding the action space, a 6-dimensional displacement in the position and orientation of the gripper relative to its own body frame has been selected. This allows the gripper to navigate its workspace and identify the optimal path for accomplishing its task, as discussed in detail in subsection 4.2.1.

4.4.3 Reward Shaping

The reward structure employed in this framework is comprehensively discussed in subsection 4.2.2. Here, I utilize the same reward components to guide the agent’s learning process effectively.

4.4.4 Curriculum-based Learning Framework

CL structures the agent’s training by gradually increasing the complexity of the environment. Instead of forcing the agent to learn full 6-DoF motion and high angular velocities from scratch, it first masters simpler tasks, preventing overwhelming exploration. The agent progressively advances through four complexity levels, each specifying different ranges of orientation ψ and angular velocity ω . The ranges grow exponentially with increasing level i :

$$\psi_i \in \left[-\pi \cdot \frac{2^{i-1}}{16}, \pi \cdot \frac{2^{i-1}}{16} \right], \quad (4.18)$$

$$\omega_i \in \left[-1 \cdot \frac{2^{i-1}}{8}, 1 \cdot \frac{2^{i-1}}{8} \right], \quad (4.19)$$

for $i = 1, 2, 3, 4$. Initially, the agent sees only translational motion (Level 1), then progressively tackles mild, moderate, and finally rapid rotations (Levels 2–4). Table 4.3 summarizes these four levels:

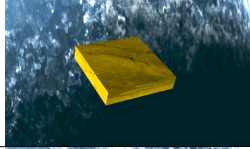
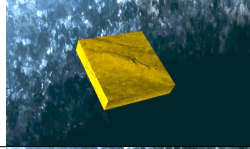
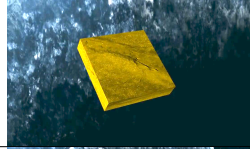
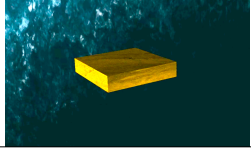
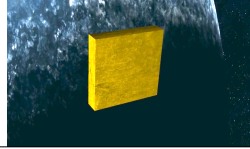
Table 4.3: Curriculum Learning Levels

CL Level	Parameter	Range
Level 1	Orientation	$\left[-\frac{\pi}{16}, \frac{\pi}{16} \right]$ rad
	Angular Velocity	$[-0.125, 0.125]$ rad/s
Level 2	Orientation	$\left[-\frac{\pi}{8}, \frac{\pi}{8} \right]$ rad
	Angular Velocity	$[-0.25, 0.25]$ rad/s
Level 3	Orientation	$\left[-\frac{\pi}{4}, \frac{\pi}{4} \right]$ rad
	Angular Velocity	$[-0.5, 0.5]$ rad/s
Level 4	Orientation	$\left[-\frac{\pi}{2}, \frac{\pi}{2} \right]$ rad
	Angular Velocity	$[-1, 1]$ rad/s

To visualize how the target is rotating in each level, Table 4.4 provides a graphical representation of the target’s orientation and angular velocity in each complexity level. In each case, the target’s orientation is started from a random value (closer to the lower bound) and its angular velocity is set to the maximum value within the specified range. It is worth mentioning that the target’s angular velocity is in the same direction in all levels, but the magnitude increases with the

level. The target’s orientation is depicted at the first, and last time steps of a 500-step episode.

Table 4.4: Comparison of target orientation and angular velocity in each curriculum level at the start and end of an episode.

Timestep	Curriculum Level 1 $\theta_0 = -0.195$ rad $\omega = 0.116$ rad/s	Curriculum Level 2 $\theta_0 = -0.377$ rad $\omega = 0.239$ rad/s	Curriculum Level 3 $\theta_0 = -0.782$ rad $\omega = 0.455$ rad/s	Curriculum Level 4 $\theta_0 = -1.084$ rad $\omega = 0.989$ rad/s
$t = 0$				
$t = 500$				

A performance threshold $S_{\text{thresh}} = 0.9$ controls level progression. After every 160 training episodes, the agent’s success rate $\bar{S}_{\text{eval}}$ over the last 10 evaluations is computed, dictating whether the agent remains at the current level, advances to the next, or—if it stagnates and achieves zero success—reverts to a previous level. Mathematically,

$$L_{(t)} = \begin{cases} L_{i+1} & \bar{S}_{\text{eval}} > S_{\text{thresh}}, \\ L_i & \bar{S}_{\text{eval}} \leq S_{\text{thresh}}, \\ L_{i-1} & \bar{S}_{\text{eval}} = 0. \end{cases} \quad (4.20)$$

At each level, both orientation and angular velocity are sampled using a non-uniform distribution that biases toward extremes:

$$v_{\text{biased}} = a + (b - a)(1 - |2u - 1|^2), \quad (4.21)$$

where $u \sim \mathcal{U}[0, 1]$, and $\{a, b\}$ are the lower and upper bounds, respectively. This ensures more frequent exposure to challenging configurations, preventing the agent from becoming complacent with only moderate rotations.

Algorithm 3 outlines the training loop. At each episode, the environment is initialized based on the current level's bounds. The agent observes the state, picks an action under the SAC policy π_θ , and updates the policy parameters using the gradient of the SAC objective (denoted symbolically in the algorithm by Eq.(3.7)). Every 160 episodes, the success rate is evaluated, deciding whether to advance, remain, or regress in complexity.

Algorithm 3 Curriculum Learning Framework

```

1: Input:  $L_i = L_1, S_{\text{thresh}} = 0.90, T = 500, j = 0$ 
2: Initialize: Policy parameters  $\theta$ 
3: for episode  $e$  do
4:   Initialize environment randomly based on  $L_i$ 
5:   Initialize agent state  $s_0$  randomly
6:   Initialize cumulative episode reward  $R_e \leftarrow 0$ 
7:   for time step  $t = 1$  to  $T$  do
8:     Sample action  $a_t \sim \pi_\theta(a_t | s_t)$ 
9:     Execute action  $a_t$ , observe  $s_{t+1}, r_t$ , and termination flag
10:     $R_e \leftarrow R_e + r_t$ 
11:    Update policy parameters via SAC gradient (Eq.(3.7))
12:  end for
13:  if  $e \bmod 160 = 0$  then
14:    Calculate  $S_{\text{eval}}^j$  based on current  $\theta$ 
15:     $\bar{S}_{\text{eval}} = \frac{1}{10} \sum_{k=j-9}^j S_{\text{eval}}^k$ 
16:    if  $\bar{S}_{\text{eval}} \geq S_{\text{thresh}}$  and  $L_i = L_4$  then
17:      Terminate training: threshold met at max complexity
18:    end if
19:    Update complexity level  $L_i$  (Eq.(4.20))
20:     $j \leftarrow j + 1$ 
21:  end if
22:  Log data  $(s_e, a_e, r_e, L_i)$  for policy evaluation
23: end for

```

Beyond the basic progression of orientation and angular velocity, several enhancements can be made:

- *Adaptive Learning Rates:* Automatically decreasing the learning rate at higher levels to allow

for more precise policy updates.

- *Multi-objective Curriculum*: Including variability in target mass or environmental disturbances to mimic real-world complexities.
- *Hierarchical Curriculum*: Decomposing the pre-grasp task into distinct sub-tasks (e.g., coarse pose alignment, fine orientation control).
- *Transfer Learning Integration*: Utilizing pre-trained networks from simpler tasks to jump-start learning at advanced levels.
- *Curriculum Regularization*: Introducing regularization to prevent overfitting to specific levels, ensuring the policy generalizes across the entire curriculum range.

4.4.5 Performance Evaluation

I conduct extensive simulations to train and evaluate the proposed Curriculum-based RL framework. Each episode lasts a maximum of 500 timesteps. The microgravity environment is modeled so that small disturbances can induce large changes in the target’s motion. To prepare the policy for real-world deployment, domain randomization is employed [106] by varying:

- The target’s initial position, orientation, linear and angular velocities.
- The gripper’s initial orientation.
- The target’s mass, thickness, and environmental noise parameters.

Table 4.5 illustrates some of these randomization parameters, ensuring that the agent’s policy becomes robust to diverse, unpredictable scenarios.

Table 4.5: Domain Randomization parameters

Parameter	Range	Units
Gripper Orientation (XYZ)	$[-\frac{\pi}{2}, \frac{\pi}{2}]$	rad
Target Position (XYZ)	$[-0.4, 0.4]$	m
Target Orientation (Y)	$[-\pi, \pi]$	rad
Target Linear Velocity (XYZ)	$[-0.1, 0.1]$	m/s
Target Angular Velocity (Y)	$[-1, 1]$	rad/s
Target Thickness	$[0.01, 0.06]$	m
Target Mass	$[10, 50]$	kg

Fig. 4.16 shows how CL (green lines) significantly improves both reward and success rate relative to a baseline RL agent lacking curriculum (grey lines). Initially, the agent was trained for roughly 20 million timesteps on a translational-only target (zero rotational velocity), forming a baseline policy. Then, I switched on the CL framework, which systematically introduced rotations.

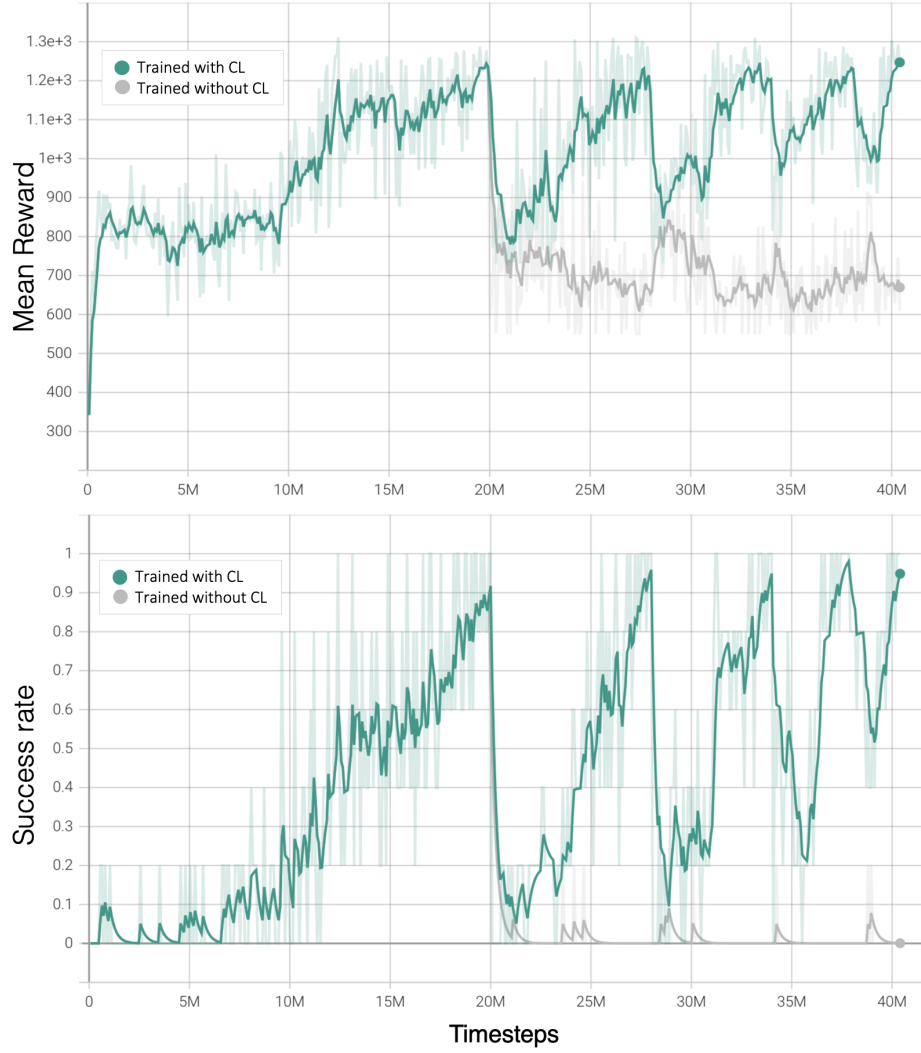


Figure 4.16: Effect of CL on agent outcomes: Comparative analysis of Mean Reward and Success Rate over Timesteps t . Incorporating CL yields faster convergence and higher final performance.

After each increase in complexity level, an expected dip in performance was observed, followed by rapid recovery. This phenomenon illustrates how the agent, having mastered simpler tasks, can quickly adapt its learned representations to handle more complex rotations. The baseline agent, in contrast, stagnated and failed to robustly handle higher levels of rotational velocity. Table 4.6 quantitatively shows the agent’s progression through the four levels:

These results underscore several key findings:

- The *steps to mastery* (i.e., the timesteps required for the agent to reliably surpass the performance threshold) decrease with each successive level, reflecting the agent’s capacity to

Table 4.6: Complexity Level Training Results

Complexity Level	Steps to Mastery	Avg. Initial 5 Evals	
		Mean Rewards	Success Rate
$L1$	8,080,000	839.76	0.00
$L2$	5,920,000	824.02	0.08
$L3$	3,840,000	931.87	0.44
$L4$	2,560,000	1150.99	0.76

leverage prior knowledge.

- The *mean rewards* and *success rates* improve significantly as the curriculum progresses, indicating that the agent can effectively handle increasingly rapid tumbling behaviors.

To evaluate the trained agent’s performance, it is tested across 100 distinct, randomized episodes. At each timestep t , the agent is loaded, and all necessary observations are collected to form the state array s_t . This array is passed into the trained actor network θ , which selects an optimal deterministic action. The chosen action is then converted into precise movements by the gripper, leveraging the agent’s accumulated experience.

Fig. 4.17 demonstrates the distribution of actions, velocity adjustments, and rewards over these 100 episodes. Consistent with the objectives, the agent aims to minimize distance and orientation errors, align velocities, and accumulate positive rewards (above 2 units). By maintaining these behaviors over a continuous 100-timestep window, it achieves a high success rate, reflected in the distribution of key performance metrics. In particular, plots (a) and (d) show how the agent modulates both positional and orientation actions, while plots (b) and (e) illustrate how these actions effectively reduce the pose differences between the gripper and the moving, rotating target. Plots (c) and (f) highlight the agent’s efforts to match its linear and angular velocities with those of the target, ensuring smooth tracking. Plot (g) reveals the accumulated rewards, underscoring how the agent consistently secures high reward signals, and plots (h) and (i) show the distributions of the gripper’s linear and angular velocities.

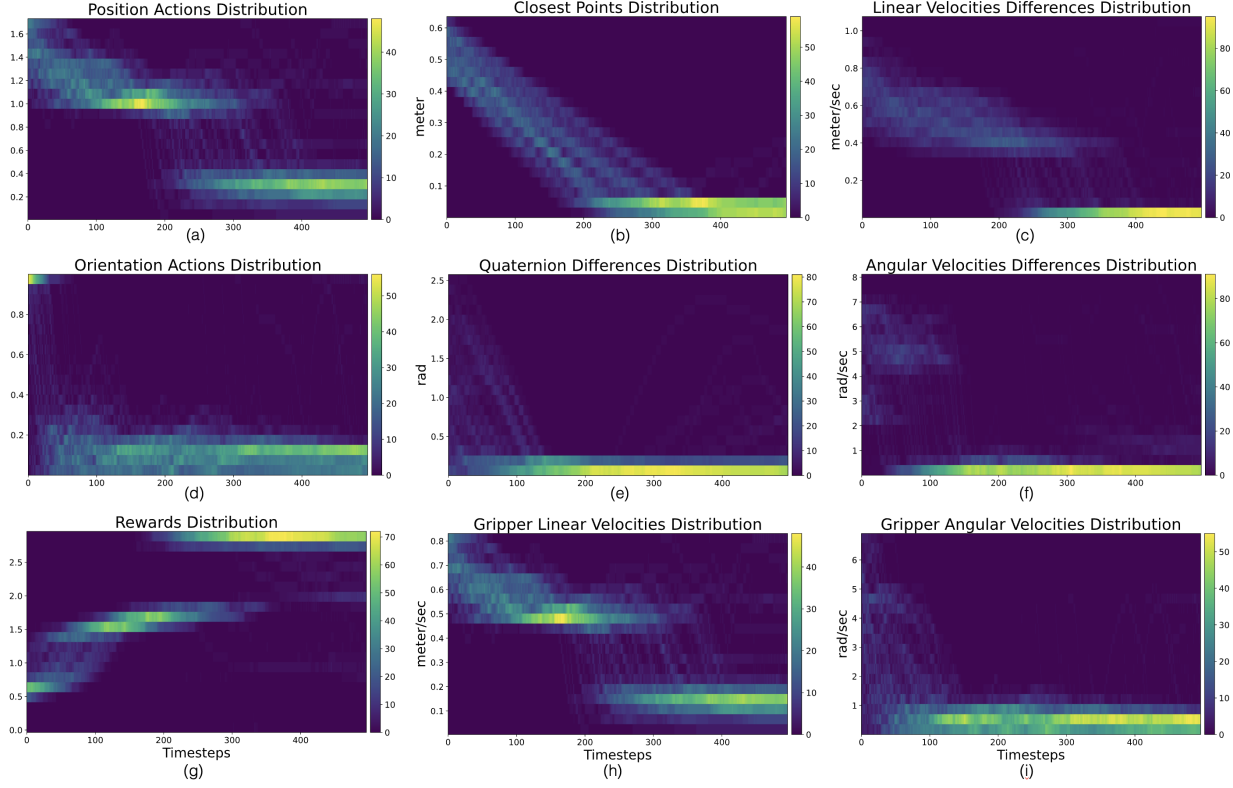


Figure 4.17: Analysis of Gripper Control Actions, Velocity, and Reward Distributions Over 100 Random Episodes.

Fig. 4.18 presents snapshots from two different randomly selected episodes, capturing key moments as the agent navigates the free-floating, rotating target. By continuously recalibrating its position and orientation, the gripper adeptly compensates for ongoing target motion, minimizing pose misalignments and velocity disparities. The sequence highlights the agent’s learned control strategies, showcasing how it leverages real-time feedback to adapt under dynamic conditions while maintaining collision-free operation.

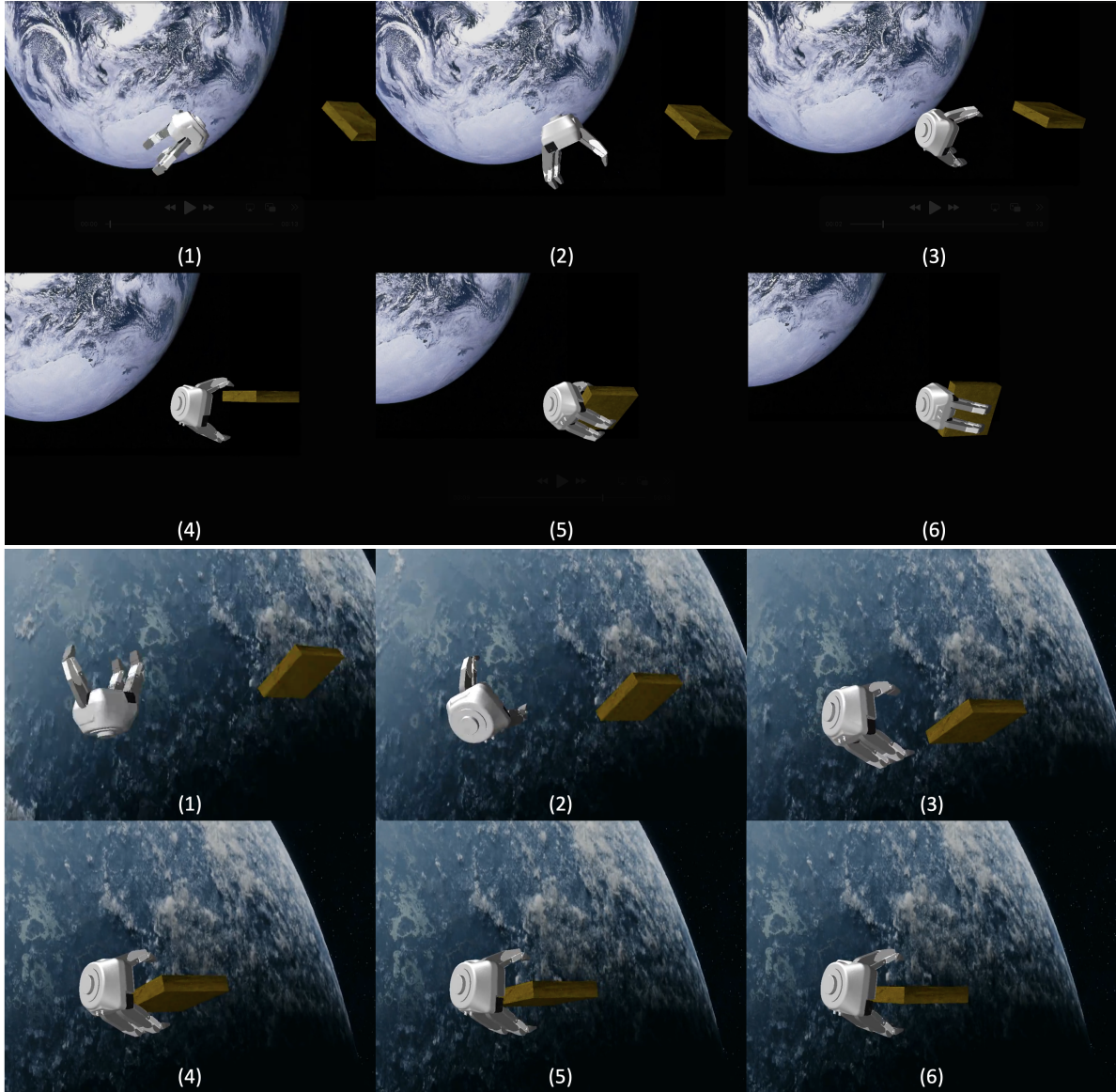


Figure 4.18: Snapshots of the gripper’s approach toward a moving and rotating target for two randomly selected episodes.

Overall, the combination of quantitative performance metrics and qualitative episode snapshots verifies that the trained agent successfully navigates a wide array of challenging scenarios, achieving robust, real-time adaptation to a moving and rotating target. These results underscore the potential of Curriculum-based RL to enhance the agent’s adaptability and performance in complex, dynamic tasks, particularly those involving free-floating, rotating bodies in microgravity environments.

4.4.6 Conclusion

In summary, this subsection presented a comprehensive Curriculum-based RL approach for tackling the pre-grasp of rotating free-floating targets in microgravity. By progressively increasing task complexity, the agent avoids the pitfalls of naive RL methods, which often fail in highly dynamic conditions. Key contributions include:

- A CL framework that systematically adjusts rotational complexity, combined with *biased sampling* toward challenging scenarios.
- Demonstration of significantly higher performance and faster convergence than baseline RL, as validated by mean rewards, success rates, and the ability to handle rapid target rotations.
- Extensive use of domain randomization to promote robust, transferable policies.

The results show that the agent efficiently leverages experience from simpler subtasks to excel under complex rotational dynamics. Future work will extend these policies to hardware implementations —particularly in space-related robotics— where robust pre-grasping is essential for operations like debris capture. This study thus underscores the potential of curriculum strategies to achieve reliable, high-precision control in challenging, real-world tasks involving free-floating, rotating bodies.

4.5 End-to-End Hierarchical Reinforcement Learning Framework for Grasping Free-Floating Space Debris

In this section, I present a holistic, hierarchical approach for grasping free-floating, moving debris under microgravity conditions using a two-stage RL framework tailored to space debris removal. First, a dedicated *positioning agent* aligns the gripper’s position and orientation with the free-floating target, precisely synchronizing the velocities of both entities to neutralize their relative motion. This alignment ensures minimal disturbance, preventing inadvertently applied external forces from displacing the debris. Once the pregrasp configuration is stable, a *grasping agent* takes over to securely close the three-finger gripper around the target. By coupling exact momentum neutralization with force modulation that adapts to the debris properties (shape, mass, friction, etc.), this approach ensures that the target is firmly immobilized while preserving its structural integrity. Ultimately, this end-to-end strategy demonstrates that advanced RL techniques can address real-world space debris capture challenges by granting robust control of an object’s motion in uncertain orbital environments. In the following subsections, the key methodological components of developed framework, such as domain randomization, reward shaping, states/actions design, and hierarchical training protocols, are detailed.

4.5.1 Domain Randomization

To enhance adaptability and reliability in space debris capture scenarios, the training pipeline incorporates extensive *domain randomization*. This strategy systematically varies the environment and object-related parameters in each episode, compelling both the positioning and grasping agents to generalize across a wide spectrum of conditions [207, 208]. As shown in Table 4.7, the critical parameters such as the gripper’s orientation, the target’s initial position, the target’s velocity, the target’s thickness, and its mass are randomized. Additionally, I incorporate variability in lateral, spinning, and rolling friction coefficients to reflect the diverse material properties and surface

conditions encountered in orbit. By exposing the agents to these parameter variations, this approach fosters a versatile skill set suitable for real-world capture scenarios.

Table 4.7: Domain Randomization Parameters

Parameter	Range	Units
Gripper Orientation (XYZ)	$[-\frac{\pi}{2}, \frac{\pi}{2}]$	rad
Target Position (XYZ)	$[-0.4, 0.4]$	m
Target Linear Velocity (XYZ)	$[-0.1, 0.1]$	m/s
Target Thickness	$[0.01, 0.06]$	m
Target Mass	$[10, 50]$	kg
Target Lateral Friction	$[0.4, 0.9]$	–
Target Spinning Friction	$[0.001, 0.01]$	–
Target Rolling Friction	$[0.0, 0.02]$	–

One key reason for including domain randomization is to ensure that the learning agents do not overfit to a single configuration, especially in an environment as complex and dynamic as orbit, where the shape, size, and dynamics of debris may vary substantially. For instance, randomizing the target thickness ensures that the debris fits within the open gripper fingers, but can still pose different frictional requirements. The randomly chosen target mass directly influences how much force must be exerted to prevent slippage or maintain stability in microgravity. In addition, friction properties such as lateral, spinning, and rolling friction can vary significantly, reflecting the diversity of surface finishes, coatings, and depositions in real debris.

Lateral friction quantifies how easily the object can slip out of the grip. A low lateral friction is representative of highly polished metal surfaces, forcing the agent to carefully regulate both normal force and alignment to avoid slippage. Conversely, high lateral friction is akin to rougher or coated surfaces that allow more robust gripping with reduced risk of lateral sliding.

Spinning friction governs the resistance of an object to in-plane or out-of-plane rotational forces. In the case of low spinning friction, the debris can rotate unexpectedly in the grip if not properly secured, which necessitates precise torque management from the agent to avert rotational instability.

Rolling friction affects the ease with which an object can roll, pivot, or slide at the edges of contact. When rolling friction is very low, the gripper has to apply more subtle, carefully oriented

forces to ensure that the target does not ‘roll out’ from the grip during maneuvers.

Overall, these friction properties place significant demands on the gripper’s control strategies for ensuring stable and secure grasps under various slipperiness or adherence conditions. Empirical results presented in later sections demonstrate how the learned policy adapts to these friction parameter variations, thus validating the importance of domain randomization in training robust, generalizable RL agents.

4.5.2 Task Definition

The task of grasping free-floating debris in microgravity is inherently complex due to the absence of gravitational settling and the associated inertial properties of the debris. In this framework, I decompose the overarching problem into two major stages:

1. **Pregrasp alignment (Positioning Agent):** A specialized positioning agent is trained to align and synchronize with the free-floating target. This agent governs the gripper’s base motion and orientation, ensuring that the relative velocity between the gripper and the target is minimized, thereby preventing unnecessary perturbations.
2. **Secure Grasp Execution (Grasping Agent):** Once the target is successfully within the nominal pregrasp configuration, responsibility shifts to the grasping agent. This agent orchestrates the three-finger closing action, controlling variables such as closing velocity, closing force, and grasping force. Its goal is to establish and maintain stable contact without causing excessive or uneven forces that might push the debris away or damage it.

Throughout each timestep t , the grasping agent receives states describing the gripper’s internal dynamics and contact interactions (detailed in subsection 4.5.3), and outputs the control actions for each finger. During this phase, tactile feedback is crucial: the agent monitors contact forces and potential slippage events while the positioning agent concurrently decelerates the target-gripper system, converging both velocities to zero in a controlled manner. Fig. 4.19 illustrates the synergy between both agents.

τ_i for each of the three fingers ($i \in \{1, 2, 3\}$).

- **Distance Metrics and Contact Forces:** Minimum distance d_i from each fingertip to the target, the number of contact points c_i per fingertip, normal forces f_{n_i} at each fingertip, overall cumulative normal force F_n , and lateral forces f_{l_i} capturing tangential interaction with the target.

The complete state vector is thus:

$$S_t = [\theta_i, \dot{\theta}_i, \tau_i, d_i, f_{n_i}, F_n, f_{l_i}] \in \mathbb{R}^{28}, \quad i \in \{1, 2, 3\}. \quad (4.22)$$

By including explicit contact information, the agent can adapt its grasping actions to account for ongoing slippage or the need to adjust finger positions and forces.

Action Space. I opt for three continuous control inputs that directly map to the gripper’s hardware-level actuations:

$$A_t = [v_c, f_c, f_g] \in \mathbb{R}^3, \quad (4.23)$$

where:

- v_c (*closing velocity*) governs how fast the fingers close inwards;
- f_c (*closing force*) sets the immediate force threshold during the finger-closing action;
- f_g (*grasping force*) represents the sustained force applied once contact is established, ensuring the object remains stable without excessive crushing.

All actions are continuous and bounded (e.g., between 0 and 5) to respect the physical constraints of typical industrial or space-grade robotic grippers. By dynamically adjusting these parameters, the agent is capable of balancing swift closure (crucial for intercepting moving debris) with the subtlety required to prevent slippage or damage to sensitive components.

4.5.4 Reward Shaping

A carefully designed reward function is critical to guide the learning process effectively. In this study, I combine three normalized reward components to reflect the competing objectives of *stable*, *gentle grasping* and *velocity reduction of the target*:

$$R_t = R_{F_N} + R_{\text{dist}} + R_{V_T}, \quad (4.24)$$

where each $R_i \in [0, 1]$, ensuring R_t lies between 0 and 3. I utilize the form $1 - \tanh(X)$ on each component so that increasing X (i.e., undesirable states) reduces the corresponding reward and motivates corrective actions by the agent.

- R_{F_N} (**Normal Force Regulation**): Penalizes excessive normal force to prevent damage or overly tight grasps. Encourages using only as much force as necessary for secure capture.
- R_{dist} (**Distance Minimization**): Rewards bringing each fingertip close to the target without incurring impulsive impacts. A small distance fosters immediate grasp stability and reduces the chance of target ejection.
- R_{V_T} (**Target Velocity Damping**): Rewards bringing the target’s velocity to near-zero. Once contact is established, the agent must coordinate with the positioning agent to dampen any residual motion.

By blending these components, the agent is strongly encouraged to smoothly converge on a stable grasp, minimize slippage, and keep the target’s motion under tight control. This nuanced reward design proved essential to training a policy that operates effectively in the multi-constraint environment of microgravity debris capture.

4.5.5 Training Results

I evaluated the end-to-end HRL approach in a high-fidelity simulated environment. The training ran for over 9,000 episodes, each episode capped at $T_{\text{max}} = 1500$ timesteps. The positioning

agent first attempts to align the gripper with the free-floating target, ensuring stable proximity for grasping. After the positioning agent achieves 200 consecutive timesteps of successful alignment, the grasping agent is activated to close the fingers. Algorithm 4 outlines the hierarchical training loop, demonstrating how the two agents operate sequentially within each episode. In a nutshell, the positioning agent first attempts to reach and stabilize the target. Once reached, the grasping agent takes over, closing the fingers and tracking contact. Throughout this process, the grasping policy is updated as it executes actions. If by the end of the episode the target is secured, the grasp is deemed successful.

Algorithm 4 End-to-End Hierarchical Training Workflow

```

1: Initialize: Positioning agent  $\pi_{\text{pos}}$ , Grasping agent  $\pi_{\text{grasp}}$ 
2: Set episode length:  $T_{\text{max}} = 1500$ 
3: for episode = 1 to  $N$  do
4:   Reset environment: Randomize initialization parameters
5:    $t \leftarrow 0$ 
6:   Load Positioning agent policy:  $\text{load}(\pi_{\text{pos}})$ 
7:   for  $t = 1$  to  $T_{\text{max}}$  do
8:      $a_p \leftarrow \pi_{\text{pos}}(s_p)$  ▷ Positioning agent predicts action
9:      $s_{t+1} \leftarrow f(s_t, a_p)$  ▷ State transition
10:    if Positioning fails or !reached then
11:      Terminate and continue to next episode
12:    end if
13:    if Positioning succeeds or reached then
14:      Transition to grasping phase:  $t_{\text{reached}} = t$ 
15:      for  $t_{\text{reached}}$  to  $T_{\text{max}}$  do
16:         $a_g \leftarrow \pi_{\text{grasp}}(s_g)$  ▷ Grasping agent actions
17:        Perform finger closure actions
18:        Track gripper-finger contact with target
19:        if grasped then
20:          Positioning agent reduces target velocity
21:        end if
22:         $\pi_{\text{grasp}} \leftarrow \text{Policy\_Update}(\pi_{\text{grasp}}, r_t, s_t, a_g)$ 
23:      end for
24:      if Object velocity at  $T_{\text{max}} \leq 0.001$  m/s then
25:        Grasping success: Target controlled
26:      end if
27:    end if
28:  end for
29: end for

```

A success is defined as: (1) all three gripper fingers make contact for at least 100 consecutive timesteps, and (2) the target's velocity is reduced below 0.001 m/s by the end of the episode. Meeting these criteria ensures that the system has effectively neutralized the target's motion, indicating a stable and reliable grasp. The learned policy is periodically evaluated every 60k training timesteps on five random test scenarios to gauge performance. Fig. 4.20 shows that after approximately 6 million timesteps, the training converges with an average success rate near 0.92 and a high mean cumulative reward. This consistency over random episodes indicates strong learning efficiency and robust generalization.

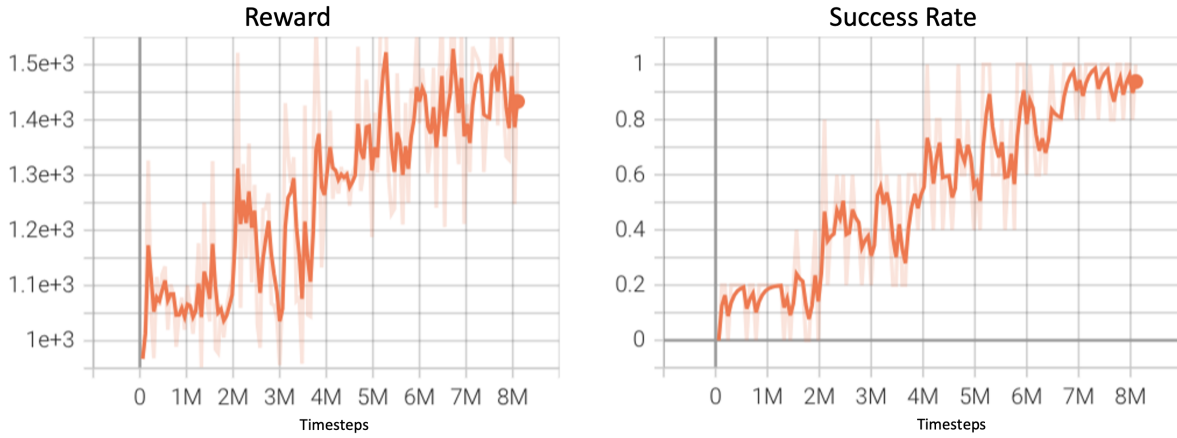


Figure 4.20: Training results in simulation. Mean evaluation performance over 5 random episodes at each checkpoint. The training converges around 6 million timesteps with a success rate approaching 0.92.

To rigorously assess the performance and robustness of the trained grasping agent, I conducted extensive evaluations across a variety of randomized scenarios within a simulated environment. This evaluation framework was designed to test the agent's ability to consistently and reliably grasp a free-floating moving target under diverse conditions, including variations in friction, mass, and geometry.

100 distinct randomized episodes were conducted using the final, best-trained grasping agent. Each episode initiated with the positioning agent, which was deemed successful if it accurately placed the gripper in the designated pre-grasp pose. Subsequently, the grasping agent was activated to perform finger closure and secure the target. Out of these 100 test episodes, the grasping

agent succeeded in 93 instances, yielding a success rate of 93%. This high success rate underscores the agent’s reproducibility in executing precise finger closure and effective velocity reduction across varied frictional, mass, and geometric conditions.

Fig. 4.21 presents the statistical distributions derived from these 100 episodes. Specifically, subplot (a) illustrates the minimum distance achieved between the gripper and the target in each episode, highlighting the agent’s ability to consistently reduce this distance from various initial positions. Subplots (b) and (c) track the linear velocities of the gripper and the target, respectively, demonstrating how both velocities converge to near-zero values by the end of successful episodes. This convergence is crucial for stabilizing the target before and during the grasp. Subplot (d) captures the distribution of grasp rewards, which begins at zero during the positioning phase and increases as the agent initiates and completes finger closure, reflecting the agent’s proficiency in transitioning from positioning to grasping effectively.

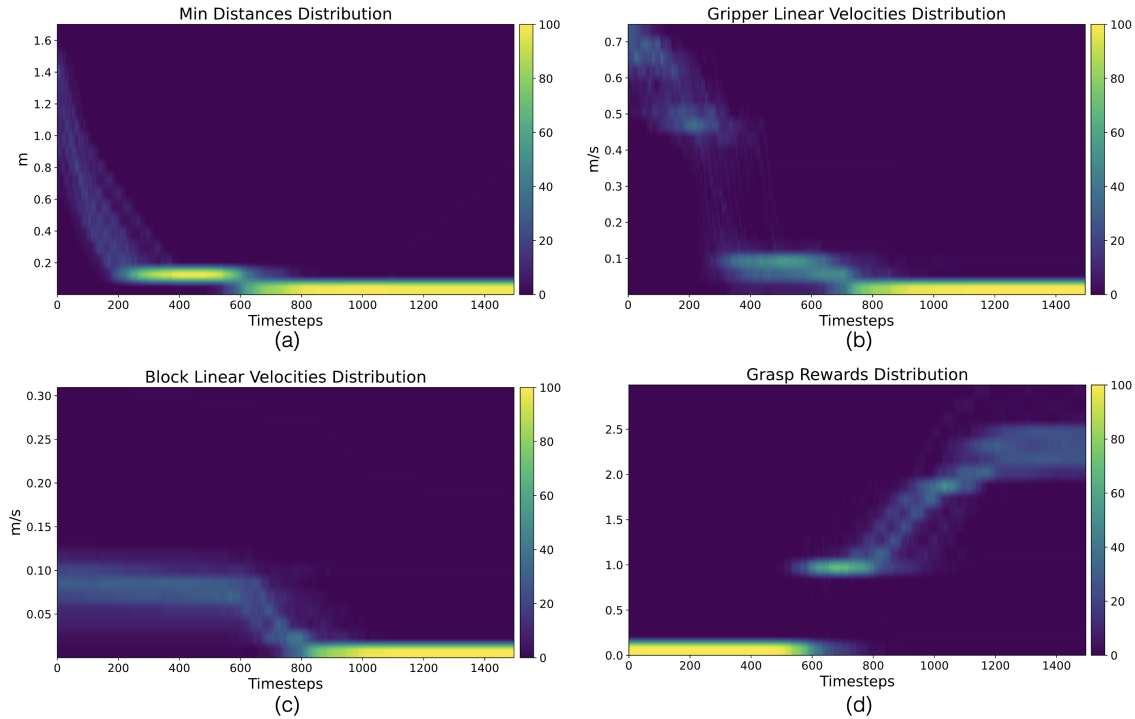


Figure 4.21: Distributions from 100 random episodes controlled by the Two-Stage RL-Trained agent.

To further evaluate the agent’s adaptability and robustness, I examined its performance under two distinct and extreme frictional conditions: a highly slippery target (*low friction*: lateral fric-

tion=0.4, spinning friction=0.001, rolling friction=0.00) with a more adherent target (*high friction*: lateral friction=0.9, spinning friction=0.01, rolling friction=0.02). These conditions were selected to represent the boundaries of friction randomization parameters (Table 4.7), thereby testing the agent’s ability to modulate its grip strategy in response to varying surface interactions.

Fig. 4.22 and Fig. 4.23 illustrate the agent’s performance under low and high friction conditions, respectively. In the low friction scenario, characterized by minimal lateral, spinning, and rolling friction coefficients, the agent employs a strategy of maintaining higher contact forces to prevent the target from slipping out of the gripper. This is evident in the state and action trajectories, where the agent carefully adjusts its finger closure force to ensure sufficient grip despite the slippery surface. The higher contact forces compensate for the reduced friction, securing the target without inducing unwanted motion. Additionally, both the gripper’s and the target’s velocities are effectively reduced to near zero by the end of the episode, demonstrating the agent’s capability to stabilize the target under challenging frictional conditions.

Conversely, in the high friction scenario with elevated friction coefficients, the agent leverages the increased surface resistance to minimize unnecessary force application post-contact. This adaptive strategy is showcased in the action trajectories, where the agent reduces the normal force exerted by the gripper upon establishing stable contact, optimizing reward by avoiding excessive force that could stress the target. The agent swiftly aligns its velocity with that of the target, ensuring minimal disturbance and maintaining control throughout the grasping process. Consequently, the agent maximizes accumulated rewards efficiently by balancing force application with the benefits of higher frictional resistance.

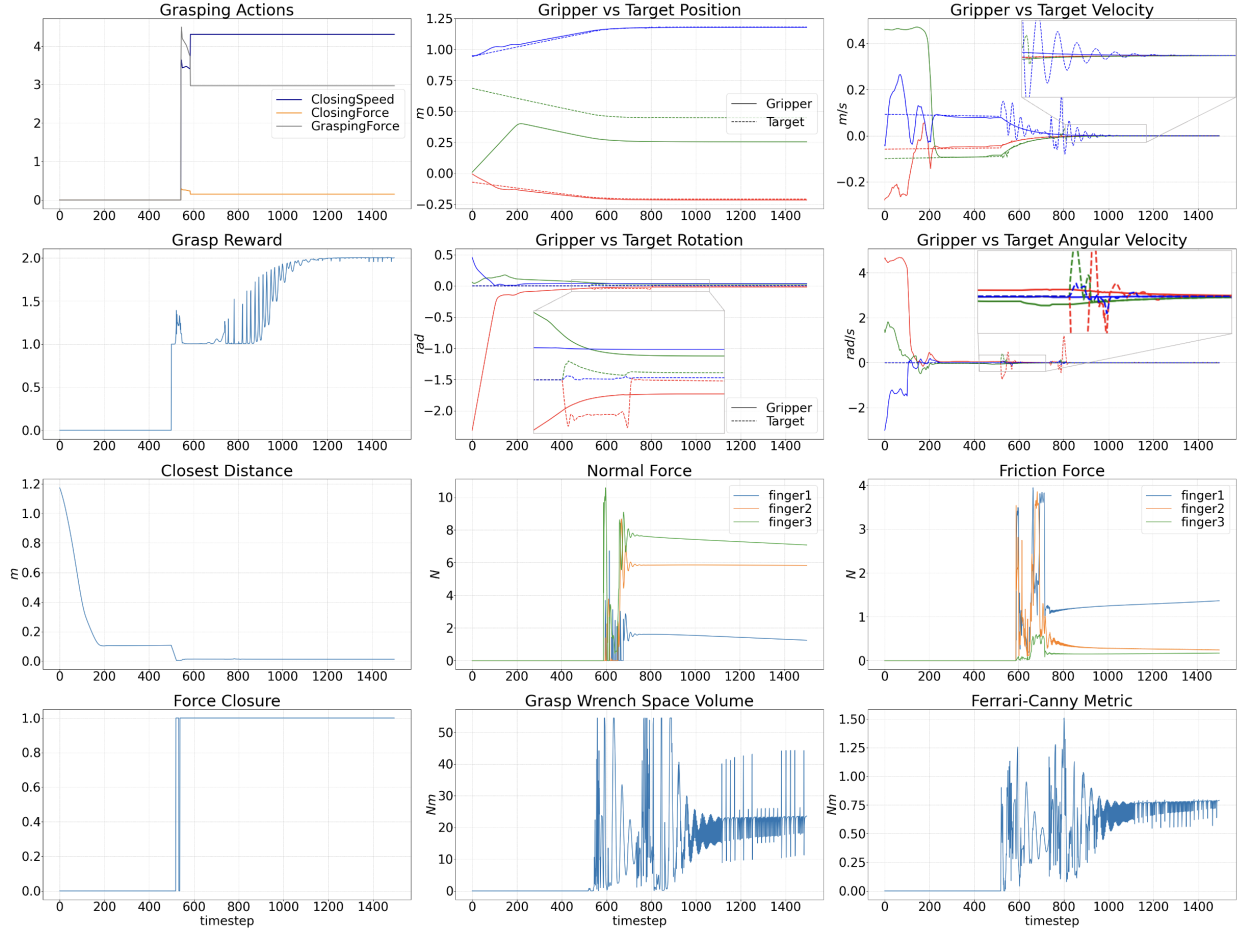


Figure 4.22: Episode performance under Low Friction condition. Each subplot depicts the agent’s state and action trajectories, highlighting the application of higher contact forces to mitigate slippage on an extremely slippery surface.

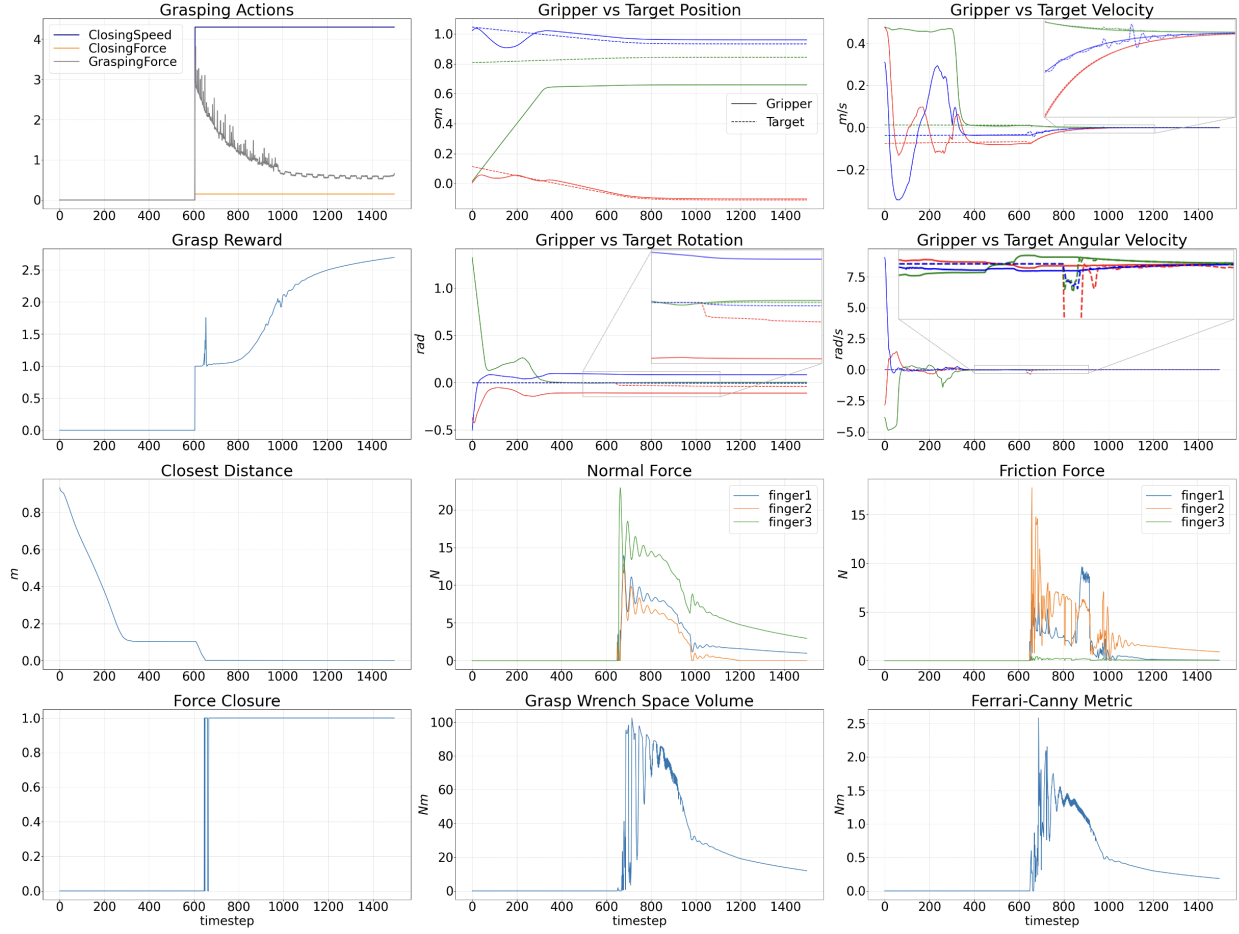


Figure 4.23: Episode performance under High Friction condition. Each subplot visualizes the agent’s ability to exploit higher friction for reducing normal force after establishing stable contact, thereby optimizing reward by avoiding excessive force application.

Fig. 4.24 provides visual snapshots from representative episodes under both low and high friction conditions. These images capture critical phases of the grasping process, including the approach strategy, finger closure, and securement of the target. In both scenarios, the agent demonstrates a calculated approach towards the target, adjusting its trajectory to account for the target’s movement and rotation. The timing and force of finger closure are meticulously controlled, ensuring that the gripper secures the target without inducing excessive movement or slippage. The final snapshots reveal a stable grasp, highlighting the effectiveness of the agent’s adaptive strategies in diverse frictional environments.

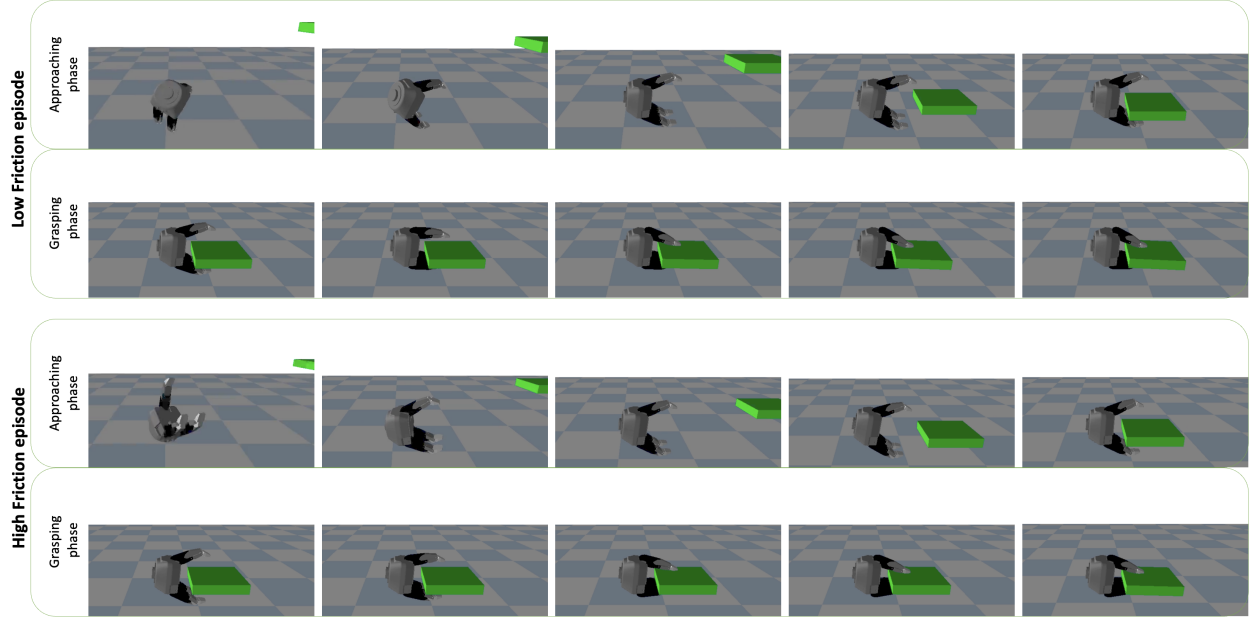


Figure 4.24: Snapshots from two evaluation episodes under Low and High Friction conditions, illustrating the agent’s approach strategy and securement of the free-floating, moving target.

In both friction scenarios, the agent successfully reduces the target’s velocity to near zero, demonstrating the synergy of careful alignment followed by robust finger closure. The hierarchical two-stage end-to-end approach, comprising a positioning agent followed by a grasping agent, facilitates this adaptability. The positioning agent ensures precise alignment and velocity matching, creating optimal conditions for the grasping agent to execute secure and efficient finger closure. This decomposition allows for specialized control strategies tailored to different phases of the grasping process, enhancing overall performance and reliability.

Quantitative metrics further elucidate the agent’s performance. The Ferrari-Canny metric assesses the gripper’s ability to resist out-of-plane torques on a flat, brick-like target. The results indicate a relatively low Ferrari-Canny score, reflecting inherent challenges in resisting complex off-axis disturbances due to the contact forces acting predominantly in a single plane. Conversely, the Grasp Wrench Space (GWS) volume remains high, indicative of the Robotiq 3-finger gripper’s capability to generate a wide range of forces in multiple directions. This versatility contributes to the agent’s ability to handle diverse grasping scenarios effectively.

The comparative analysis of episode performances under extreme friction conditions elucidates

the agent’s robust and adaptable behavior. In high friction scenarios, the agent optimizes reward by reducing normal force post-contact, leveraging the increased surface resistance to maintain a secure grip with minimal force application. Conversely, in low friction conditions, the agent compensates for reduced friction by sustaining higher contact forces to prevent target slippage, ensuring grasp stability despite the challenging surface properties.

4.5.6 Conclusion

Overall, this end-to-end HRL framework effectively addresses the complexities of grasping free-floating, moving debris in microgravity by decomposing the problem into two key stages. The first stage aligns the manipulator to the debris while the second stage focuses on tactile-driven, secure finger closure. It has been shown that through domain randomization and carefully designed state/action spaces and rewards, high success rates across challenging frictional and mass conditions can be achieved by RL agents. The approach paves the way for more advanced and reliable real-world capture missions, where robust adaptive behavior and minimal risk of debris ejection are mission-critical.

Chapter 5 Ground-Based Experimental Verification and Validation

This chapter validates the developed reinforcement learning (RL) frameworks for robotic grasping in simulated microgravity through experiments at two advanced facilities: the York University Testing Facility and the University of Luxembourg Zero-G Lab. The York University setup utilizes a hardware-in-the-loop (HIL) system with dual Fanuc industrial robots to mimic space-like conditions, enabling tests on tumbling targets, soft capture strategies, and sensory integrations. Meanwhile, the Zero-G Lab employs rail-mounted robotic arms, motion capture systems, and dynamic lighting to replicate six-degree-of-freedom (6-DoF) target motions, allowing us to assess hierarchical RL frameworks in diverse scenarios. Additionally, the impact of computer vision and tactile feedback on enhancing grasp accuracy and reliability was examined. Through a series of controlled experiments and real-world demonstrations, this chapter demonstrates the effectiveness of the developed RL approaches in achieving precise and stable grasping of free-floating space debris, showcasing the successful transition from simulation to practical application.

5.1 York University Testing Facility

The York University Testing Facility is equipped with two Fanuc M-20iD/25 industrial robots, orchestrated to emulate space situational behaviors within a HIL setup. This experimental system is designed to validate the feasibility of debris capture by space robotic manipulators, focusing on tumbling target maneuvers, soft capture control strategies, and the integration of various sensory capabilities. A schematic of the testing facility is illustrated in Fig. 5.1.

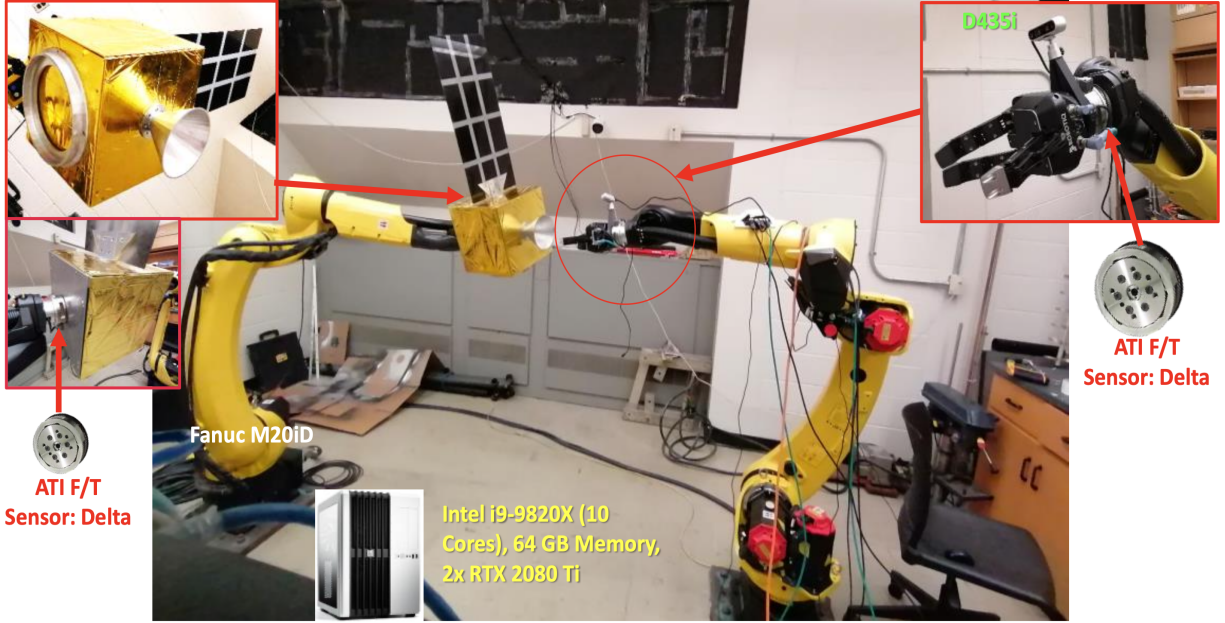


Figure 5.1: York University Testing Facility setup demonstrating the interaction between the two Fanuc robots.

Within the facility, one robot is equipped with a Robotiq 3F gripper and is responsible for executing the motion commands of a trained agent operating under microgravity conditions. This agent controls the gripper to manipulate a dummy satellite, which serves as a stand-in for an actual space object. The second robot handles the dummy satellite, replicating the dynamics of a free-floating target in space. The dummy satellite is outfitted with a nozzle, solar panel, and a launch adapter ring (LAR), accurately mimicking the structural and functional aspects of a real satellite, thereby providing a robust platform for agent testing.

Both robots incorporate gravity compensation mechanisms facilitated by dual FT sensors attached to their end-effectors. These sensors enable precise adjustment of the robots' movements to simulate a microgravity environment effectively. Additionally, the robot controlled by the trained agent is equipped with a camera system that supplies real-time data on the target's position and orientation, enhancing the accuracy of motion control.

To manage the intricate dynamics of the system, I have developed a custom control API using Python [209]. This API emulates the dynamic responses of a free-floating spacecraft-borne robot

and its corresponding target by leveraging multibody dynamics and contact models tailored for a microgravity environment. Communication between the Fanuc robot controllers and an external PC is established via an Ethernet connection. Utilizing the StreamMotion protocol, each of the six robotic joints is individually controlled by streaming joint commands through the Python API to the Fanuc control box.

These joint commands are transmitted using UDP communication, ensuring that commands are sent and received within a strict 4 ms timestep and maintaining the correct sequence numbers for synchronization. This setup guarantees that the joint commands are timely and accurately reflected in the robots' movements.

The transformation of displacement actions in position and orientation, received by the trained agent, into precise robot movements is achieved through the application of Denavit–Hartenberg (DH) parameters. Table 5.1 presents the DH parameters for the Fanuc M-20iD/25 robotic manipulator. These parameters facilitate the calculation of both forward and inverse kinematics, enabling the physical robots to replicate the motions of the simulated space robotic end-effectors and targets.

Table 5.1: Denavit–Hartenberg Parameters for Fanuc M-20iD/25 Industrial Robot

Joint	θ_i [rad]	a_i [m]	d_i [m]	α_i [rad]
1	θ_1	0	0.075	$\frac{\pi}{2}$
2	θ_2	0	0.84	0
3	θ_3	0	0.215	$-\frac{\pi}{2}$
4	θ_4	-0.89	0	$\frac{\pi}{2}$
5	θ_5	0	0	$-\frac{\pi}{2}$
6	θ_6	-0.89	0	0

The DH parameters define the kinematic relationships between consecutive joints and links of the robot, enabling precise modeling of the robot's spatial configuration. Specifically, for each joint i , the parameters include:

- **Joint Angle (θ_i):** The angle between the x_i axis and the x_{i+1} axis, measured around the z_i axis.
- **Link Length (a_i):** The distance between the z_i and z_{i+1} axes along the common normal.

- **Link Offset** (d_i): The distance along the z_i axis from the origin of frame i to the intersection with the common normal.
- **Link Twist** (α_i): The angle between the z_i axis and the z_{i+1} axis, measured around the x_i axis.

These parameters are instrumental in deriving the forward and inverse kinematic equations, which are essential for translating the agent's displacement actions into executable robot joint commands.

Forward kinematics (FK) involves determining the position and orientation of the robot's end-effector based on the given joint parameters. Using the DH parameters, the homogeneous transformation matrix A_i for each joint i is defined as:

$$A_i = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5.1)$$

The overall transformation matrix T from the base frame to the end-effector frame is obtained by the sequential multiplication of the individual transformation matrices:

$$T = A_0 A_1 A_2 A_3 A_4 A_5 A_6, \quad (5.2)$$

where each A_i corresponds to the transformation from frame $i - 1$ to frame i . This matrix T encapsulates both the position and orientation of the end-effector relative to the base frame.

Inverse kinematics (IK) involves determining the joint parameters $\{\theta_i\}$ that achieve a desired end-effector pose, defined by the transformation matrix T . In this work, the IK is calculated using a numerical method, more specifically the Levenberg-Marquardt algorithm, that leverages the previous joint angles to find solutions that are closest to them. This approach helps in achieving smooth and continuous motion by minimizing abrupt changes in joint configurations.

The accurate computation of both forward and inverse kinematics using DH parameters ensures

that the robot can precisely mimic the motions required for manipulating the dummy satellite in the microgravity simulation environment. This capability is essential for validating the control strategies developed for capturing space debris.

The 6-DoF HIL robotic experimental system has been constructed and subjected to preliminary testing. Initial results demonstrate the system’s capabilities in computer vision, sensing, and motion control. The setup effectively simulates the capture of tumbling debris by robotic manipulators in space, validating the proposed control strategies and the system’s sensory integration. These findings underscore the potential of the York University Testing Facility as a robust platform for advancing space robotics research.

5.1.1 Computer Vision

In this work, an AI-based computer vision system was developed to track the relative motion and orientation of a target satellite using a hand–eye camera configuration. Specifically, the YOLOX [210] algorithm was adopted to detect and localize key structural features of the target, namely the nozzle, the LAR, and the solar panel. During training, synthetic datasets were generated by randomly placing the corresponding 3D models of these features in a virtual environment (Blender), thereby simulating diverse viewpoints and lighting conditions. An Intel Realsense D435 depth camera was used to collect and validate real-world data, demonstrating robust performance after model deployment.

The computer vision pipeline first identifies the position of the target features in the camera image frame, then infers the overall attitude by combining the individual feature detections from varying perspectives. By exploiting the depth information from the D435 camera in conjunction with the 2D detections, the relative 6-DoF pose of the target with respect to the camera is estimated. Consequently, real-time pose tracking is achieved, which is essential for accurate satellite capture and manipulation tasks. The result, shown in Fig. 5.2, demonstrates the YOLOX algorithm’s ability to track the target’s key features in approaching phase.

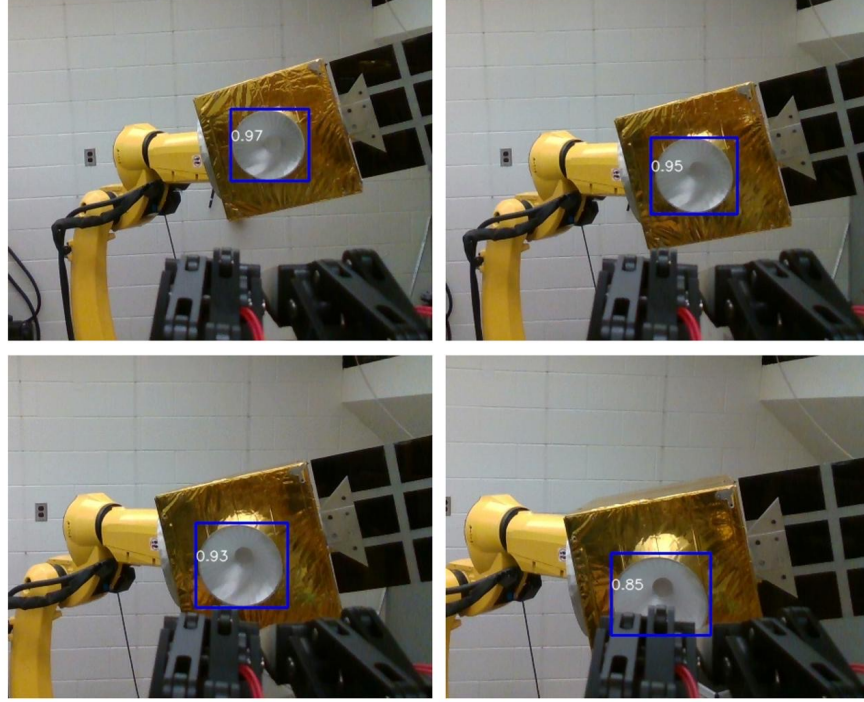


Figure 5.2: Illustrative example of the AI-based computer vision (YOLOX) tracking the target’s key features, nozzle in this case, in real time with confidence level.

To validate the system’s accuracy, a series of displacement tests was carried out using a Fanuc industrial robot, wherein the target satellite was translated relative to the camera by 0.5 m along the optical axis, downward by 0.5 m, and sideward by 0.5 m. No rotation of the target was introduced during these tests. Since the Fanuc robot’s positional accuracy is 0.02 mm, its end-effector trajectory was adopted as ground truth. The aim was to assess the positional accuracy of YOLOX under different viewing angles.

Fig. 5.3 presents the position tracking error of the YOLOX algorithm. Notably, the position error remains minimal when the target is moved along the camera’s optical axis. However, as the target moves laterally out of the optical axis, the error can increase up to 3 cm ($\sim 6\%$). This phenomenon arises because the nozzle appears skewed, causing the bounding-box center—interpreted as the nozzle’s position—to deviate from the true physical center of the nozzle. Although further refinement is needed to mitigate this skew-induced error, the achieved accuracy is sufficient for the downstream control tasks, such as gripper manipulation via a RL agent.

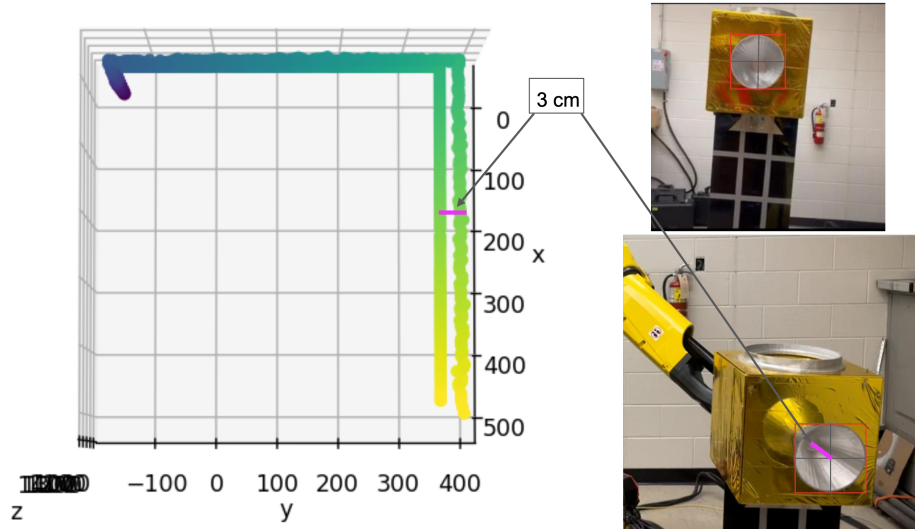


Figure 5.3: Position tracking error of YOLOX under different translations of the target satellite. As the target moves off-axis, estimation skewness increases the location error.

5.1.2 Robotiq 3F Gripper

The Robotiq 3F gripper is a versatile and adaptive end-effector designed to handle a broad spectrum of objects, ranging from rigid geometric shapes to irregular and delicate items. As depicted in Fig. 5.4, the gripper comprises three fingers, each actuated by an independent motor located within the palm, providing the necessary range of motion to conform to various target geometries. By exerting torque on the proximal (first) joint of each finger, the actuation is mechanically transferred to the remaining two joints, thereby enabling the compliant multi-link behavior essential for capturing noncooperative space debris.



Figure 5.4: Robotiq 3F gripper with independently actuated proximal finger joints and mechanical linkages for the distal joints.

Beyond its three primary actuators (one for each finger), the Robotiq 3F gripper is also equipped with a fourth actuator that adjusts the distance between two of the fingers, permitting both pinch and wide grasp modes [211]. In this study, I focused on the primary grasping functionality and did not employ the fourth actuator, as the application primarily required a stable enclosure of the target for capture and manipulation. The inherent compliance of this end-effector, stemming from its linkage-based joint architecture, makes it particularly advantageous for space robotics tasks where object shape and contact dynamics can be highly uncertain.

To minimize the simulation-to-reality (sim2real) gap, a *digital twin* of the Robotiq 3F gripper was developed, integrating high-fidelity mechanical linkages, finger kinematics, and actuator constraints. Digital twins, in the context of robotic systems, are virtual replicas of physical assets that dynamically reflect their real-world counterparts through continual synchronization of sensor data, control commands, and performance metrics [212]. In the developed framework, the digital twin serves not only to replicate the kinematic and dynamic behaviors of the gripper but also to capture the nuances of contact events during object interaction.

A state machine model underpins the digital twin to accurately reproduce the gripper’s kinematic evolution and mechanical linkages, ensuring that simulation states mirror real-world behavior. As illustrated in Fig. 5.5, each state and transition corresponds to a critical phase in the gripper operation, such as *no contact*, *initial contact*, *fully enclosed*, and *contact force saturation*. This

modular approach facilitates fine-grained control and monitoring of finger motion, transitioning between states when torque or position thresholds are met. By employing explicit transitions and predefined conditions, the state machine provides robust control logic that aligns closely with hardware behaviors.

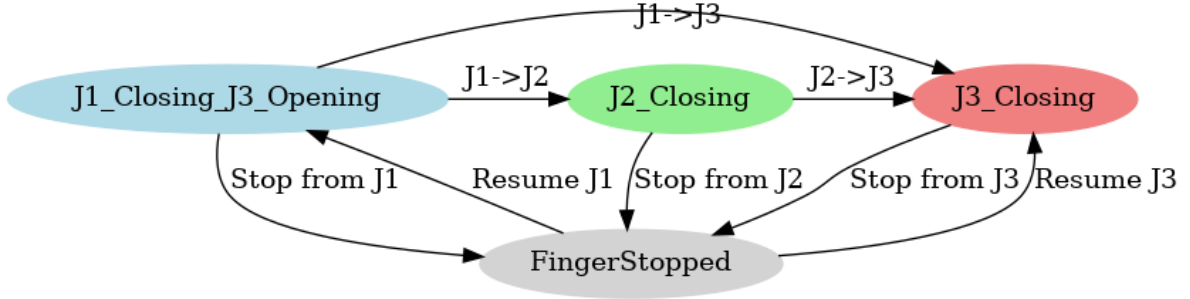


Figure 5.5: State machine model for the Robotiq 3F gripper digital twin, enabling accurate replication of mechanical linkages and contact events.

In practice, the gripper’s fingers are controlled by specifying their positions, closing speeds, and target forces. This study employs a custom Python-based API that leverages a TCP protocol to communicate with the gripper’s controller [213]. The API surpasses the default controller’s capabilities by enabling real-time adjustments of each finger’s positional setpoint, along with the force exerted on the target. Due to the multi-threaded design of the controller, simultaneous monitoring of the applied force and commanded position is achieved, which is critical for precise manipulation and collision avoidance.

The operational parameters are discretized into 255 levels for position, speed, and force, with speeds ranging from 22 mm/s to 110 mm/s, and allowable forces spanning from 15 N to 60 N. Real-time coordination of these parameters ensures that each finger can be individually tuned, facilitating synchronization among the fingers as they close around a target. By calibrating these operational settings in the simulation, I ensure that the digital twin not only mirrors but also predicts hardware performance in a range of interaction scenarios.

Simulation-to-Reality Comparisons. To validate the fidelity of the digital twin and the accuracy of the state machine model, a series of motion capture experiments, comparing the physical behavior

of the gripper with that observed in the simulation were performed. Fig. 5.6 provides an illustrative comparison of finger configurations at various stages of closure. The close alignment of finger trajectories confirms that the digital model successfully captures the mechanical coupling between joints and the progressive contact dynamics.

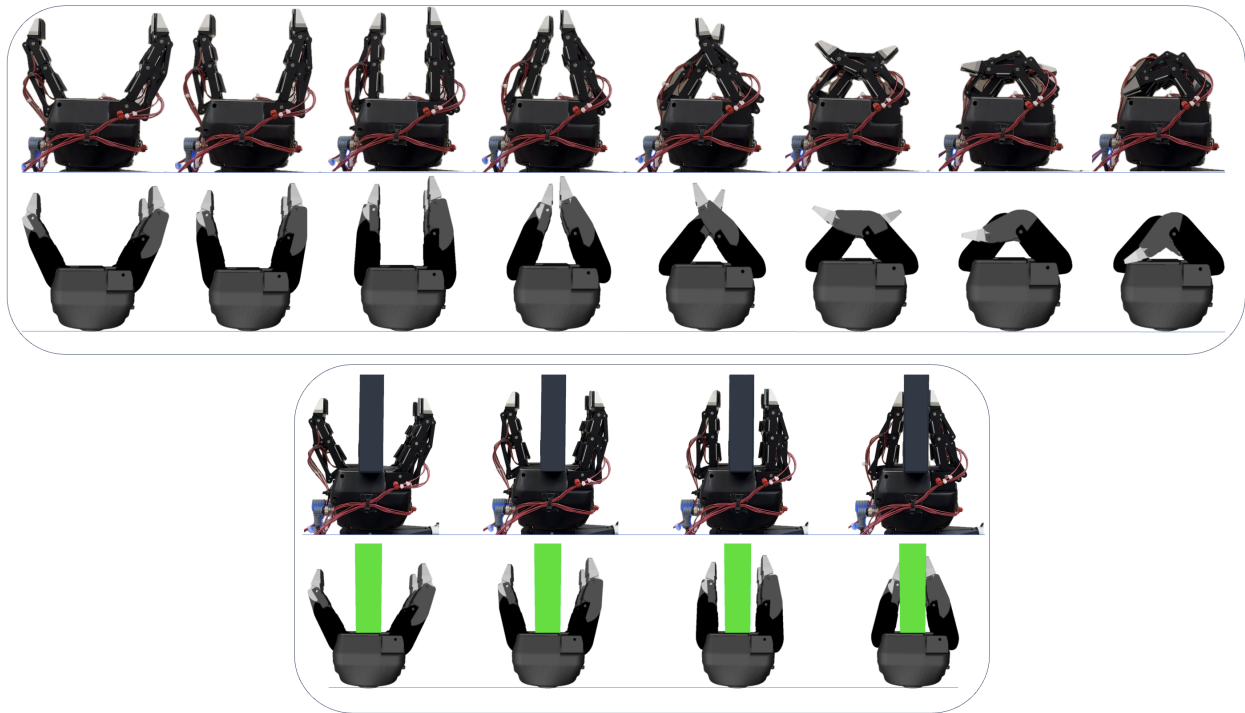


Figure 5.6: Comparison of real and simulated Robotiq 3F gripper finger motions, illustrating high-fidelity replication of mechanical linkages and contact events.

To enhance grasp reliability and ensure compliance with delicate or irregular objects, force feedback is essential. I integrated RP-C18.3-ST thin-film pressure sensors at each finger link, read by an Arduino microcontroller, to acquire real-time measurements of the applied contact forces (see Fig. 5.7). These measurements are particularly crucial during the pre-grasp phase, where small contact forces inform the agent’s grasp strategy. By feeding this sensory information into the developed control loop, the agent can dynamically modulate finger positions, speeds, and forces to maintain stable and low-impact contact with the target. This setup is especially beneficial when dealing with potential uncertainties in debris mass, inertia, and geometry.

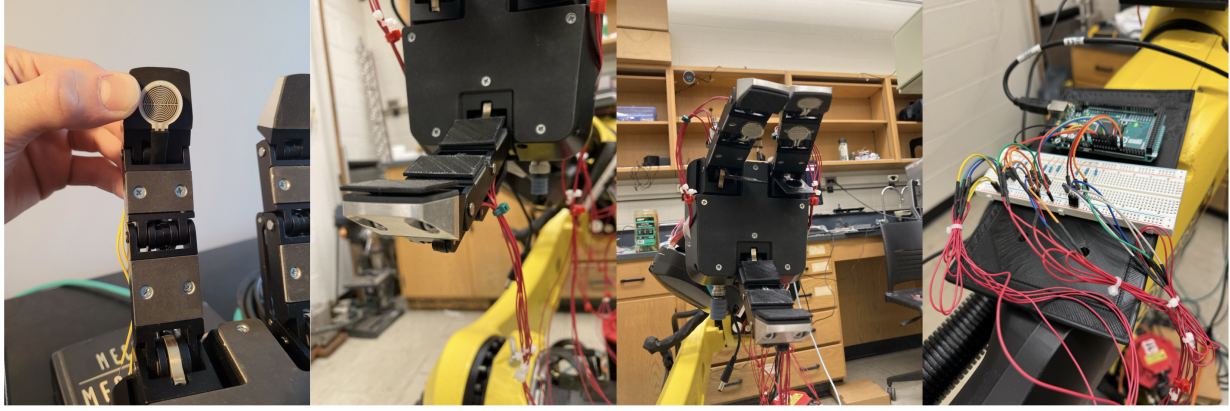


Figure 5.7: Force sensing setup with RP-C18.3-ST thin-film pressure sensors on the Robotiq 3F gripper fingertips, enabling real-time measurement of contact forces.

The integration of the Robotiq 3F gripper, its digital twin, and force-sensing instrumentation lays a robust foundation for grasping and manipulation tasks in noncooperative, microgravity-like settings. The installation of thin-film pressure sensors provided real-time feedback, completing the final component of state observation. This feedback is crucial for the RL agent to adaptively adjust its actions based on the possibility of contact with the target. The tight coupling of simulation and hardware, underpinned by state machine logic and real-time control, ensures that the developed system can respond reliably to the dynamic and uncertain conditions inherent to on-orbit debris capture.

5.1.3 Gravity Compensation

To emulate the free-floating nature of a satellite in space, the dummy target satellite is mounted on a robot end-effector that incorporates an ATI force/torque (FT) sensor. When external forces or moments act on the dummy satellite, the sensor measurements are used to drive the robot end-effector as though the satellite were entirely unconstrained in a microgravity environment.

When no external disturbance is present, the measured force and torque from the ATI sensor mainly reflect the influence of gravity. Let $\mathbf{F}_{\text{ext}}$ and $\boldsymbol{\tau}_{\text{ext}}$ be the total external force and torque measured by the sensor. These are decomposed into a gravitational component and a net component arising from any external interaction. If m denotes the payload mass, $\mathbf{g}$ the gravitational acceleration,

and $\mathbf{r}$ the vector from the sensor frame to the payload center of mass, then the gravitational force and torque can be expressed as:

$$\mathbf{F}_g = m \mathbf{g}, \quad \boldsymbol{\tau}_g = \mathbf{r} \times \mathbf{F}_g. \quad (5.3)$$

Subtracting these from the raw sensor measurements yields:

$$\mathbf{F}_{\text{net}} = \mathbf{F}_{\text{ext}} - \mathbf{F}_g, \quad \boldsymbol{\tau}_{\text{net}} = \boldsymbol{\tau}_{\text{ext}} - \boldsymbol{\tau}_g. \quad (5.4)$$

Hence, $\mathbf{F}_{\text{net}}$ and $\boldsymbol{\tau}_{\text{net}}$ represent only the external disturbance caused by collisions or other contact events. By integrating $\mathbf{F}_{\text{net}}$ and $\boldsymbol{\tau}_{\text{net}}$ into the control law, the robot end-effector is directed to move in such a way that the dummy satellite responds like a free-floating target.

Fig. 5.8 illustrates the ATI sensor's placement on the end-effector and the relevant local frames. During initialization, the satellite remains at rest to measure baseline forces and torques, which are used to calibrate offset errors. Once an external disturbance $\mathbf{F}_c$ is detected, the moment arm relative to the center of gravity ($\mathbf{r}_0 - \mathbf{r}$) is calculated. A simplified dynamic model then determines the resulting motion of the satellite, and the corresponding joint-angle commands are dispatched to the physical robot. Consequently, the dummy satellite's motion closely approximates free-flight dynamics under orbit-like conditions.

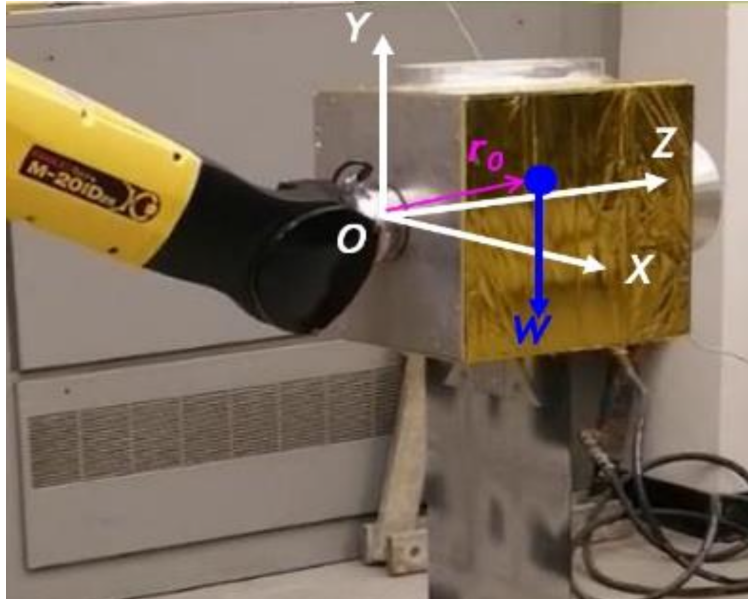


Figure 5.8: Illustration of the ATI sensor mounted on the end-effector and the local frame definitions.

Snapshots of this process are shown in Fig. 5.9, where external forces are deliberately applied to the dummy satellite. By nullifying terrestrial gravity, algorithms for satellite capture can be tested under realistic scenarios. Future enhancements will incorporate more sophisticated dynamic models—accounting for Coriolis forces, damping, and robot-environment interactions—to further improve the fidelity of microgravity emulation.

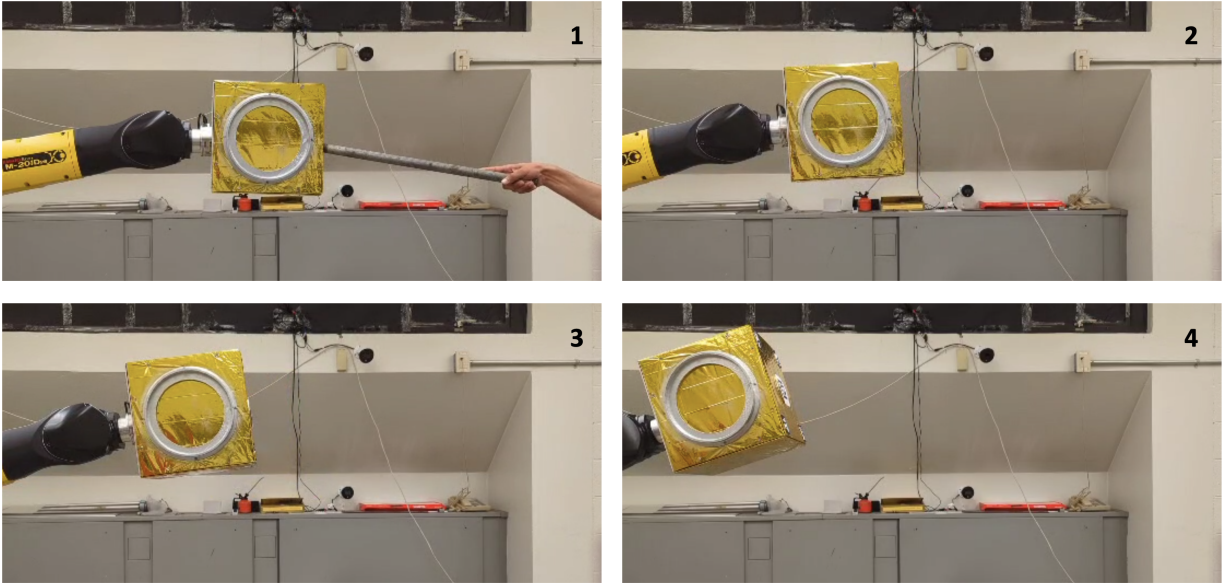


Figure 5.9: Snapshots demonstrating the dummy satellite’s motion in response to external forces. The robot applies gravity compensation, enabling the satellite to behave as if in a microgravity environment.

5.1.4 Grasping Task Demonstration

To validate the effectiveness of the York University Testing Facility in replicating space debris capture scenarios, a series of grasping tasks were conducted. The objective was to demonstrate the laboratory’s capability to emulate the dynamics of a free-floating satellite and to showcase the integration of various system components, including the Robotiq 3F gripper, ATI force/torque sensor, and vision system. The experiments aimed to highlight the seamless coordination within the HIL setup.

In this demonstration, the target satellite motion was prescribed based on the tumbling rates of the GOES 8 geostationary satellite, as reported in [214, 215], and those of the Envisat satellite [216]. Specifically, a drift rate of -0.01 m/s along the x -axis was employed, while tumbling rates of 0.28, 2.8, and 0.28 deg/s were imposed about the roll, pitch, and yaw axes, respectively. To transfer these motion profiles to the FANUC M20iD industrial robot and the dummy satellite, the controller—designed to accept next-goal joint positions—relied on IK to generate appropriate joint commands. The resulting trajectories in both joint and Cartesian spaces, depicted in Fig. 5.10,

illustrate how the robot precisely executes each intermediate goal to mimic the satellite's free-floating motion.

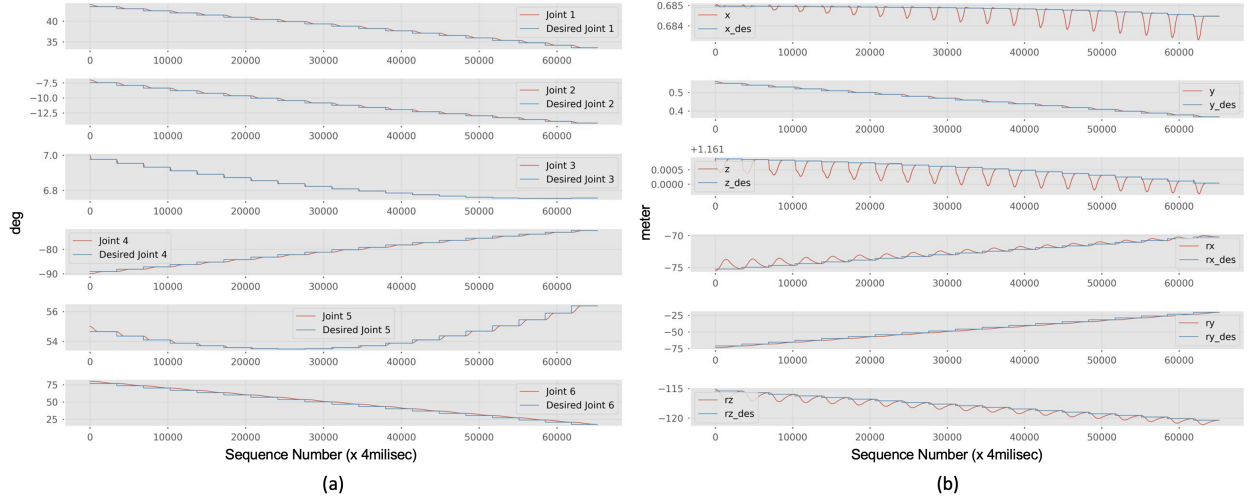


Figure 5.10: Trajectories of the dummy satellite in (a) joint space and (b) Cartesian space, demonstrating the robot's ability to replicate the target's tumbling motion.

To orchestrate the grasping sequence, the second robot arm carrying the Robotiq 3F gripper was synchronized with the satellite's simulated tumbling behavior. Under open-loop control, it received joint-space commands guiding the gripper into the final grasping pose, as shown in Fig. 5.11. Once properly positioned, the gripper was signaled to close its fingers, thereby completing a successful open-loop demonstration of capturing a freely tumbling target. This procedure not only underscores the integration and responsiveness of the York University Testing Facility but also serves as a proof of concept for on-orbit servicing operations. A series of snapshots depicting the entire demonstration is presented in Fig. 5.12.

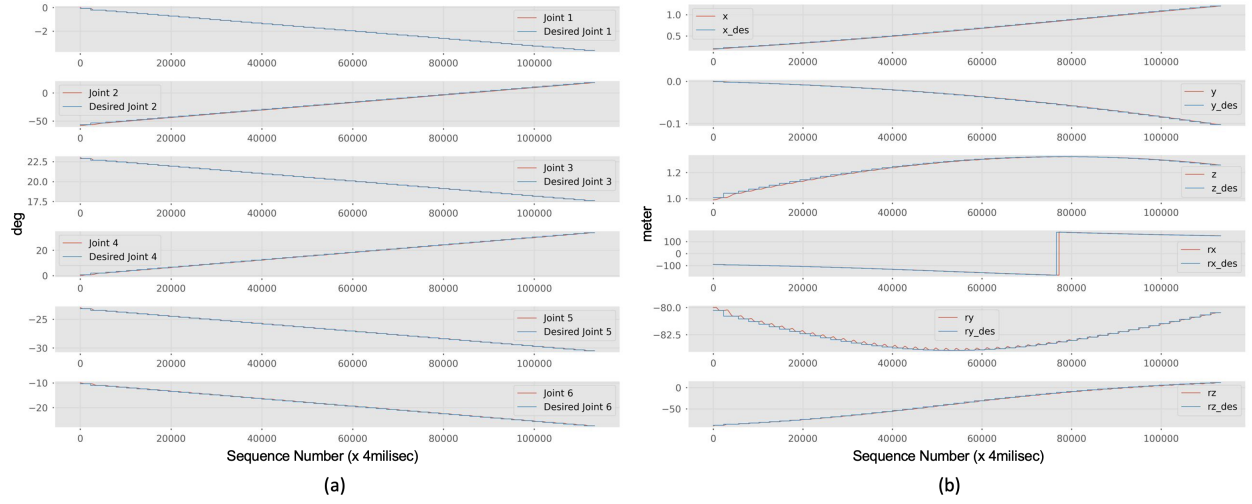


Figure 5.11: Chaser robot arm trajectories in (a) joint space and (b) Cartesian space, aligning the Robotiq 3F gripper with the target satellite for grasping.

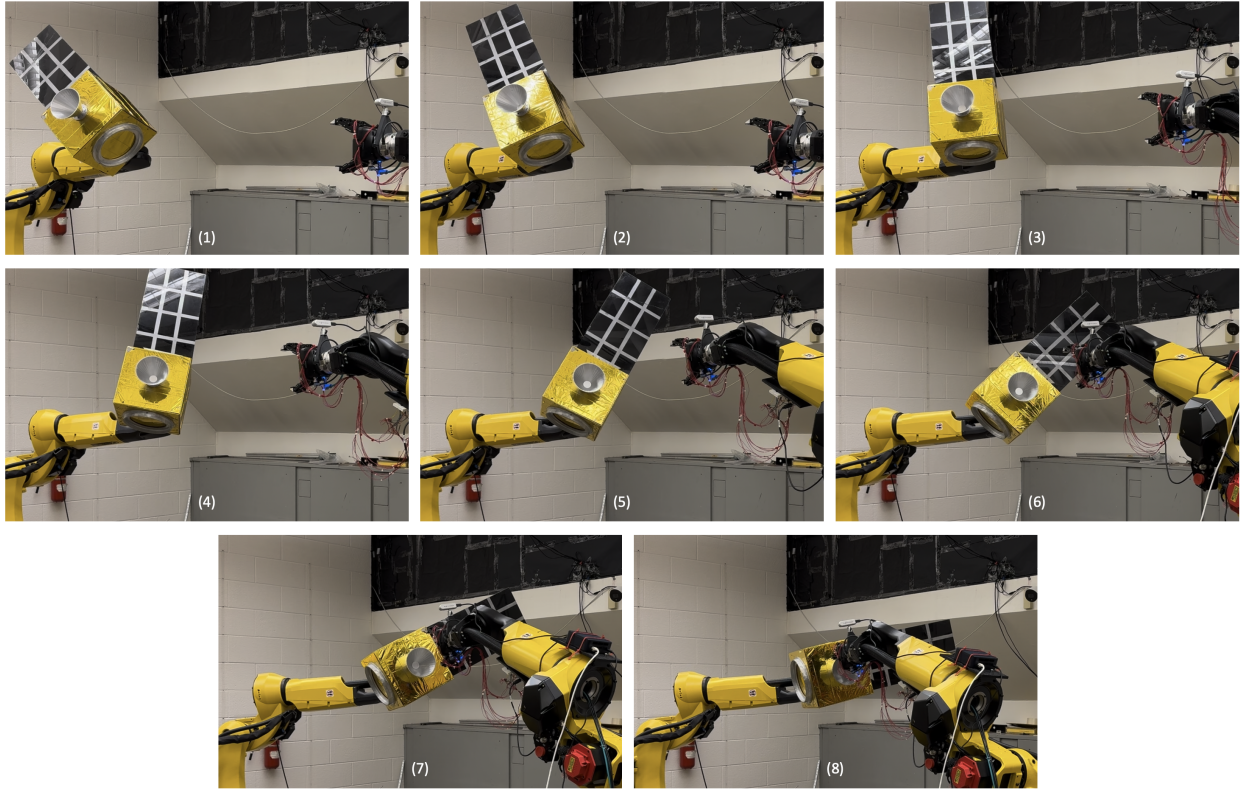


Figure 5.12: Snapshots of the grasping task demonstration, showcasing the coordinated motion of the chaser robot and the target satellite.

5.1.5 Trained Agent Implementation

In this work, the implementation of the trained agent involves several critical steps to transition from simulation to real-world deployment. After training a robust agent in the simulation environment with an acceptable success rate, the agent is deployed on the physical robots to evaluate its effectiveness in addressing real-world challenges.

The process of transferring the trained agent for real-world testing necessitates several considerations:

- With the gripper now affixed to the real manipulator robot, it is essential to utilize the robot's IK to determine the subsequent position and orientation of the gripper when executing the most recent action predicted by the trained agent.
- Unlike the simulation environment, where all observations can be seamlessly obtained from readily available data (e.g., target position or the minimum distance between the target and the gripper's palm), in a real-world setting, these observations must be derived using available sensors.
- Coordination must be adjusted to mirror that of the simulation environment. Additionally, ascertaining the ground truth position and orientation with respect to an inertial frame for each component becomes paramount when navigating real-world scenarios.

Fig. 5.13 illustrates the comprehensive HIL connections that facilitate real-world testing. The setup comprises two primary computer systems, each tasked with controlling different robots. These computers are programmed to communicate with Fanuc M-20iD robots using UDP (User Datagram Protocol) connections. The primary system, which governs the gripper via a trained agent, randomly selects a position for the target. Subsequently, it dispatches a command to the secondary system, instructing it to relocate the target. Once the system initiates this movement, the primary setup remains in a waiting state until the target attains its predetermined final position.

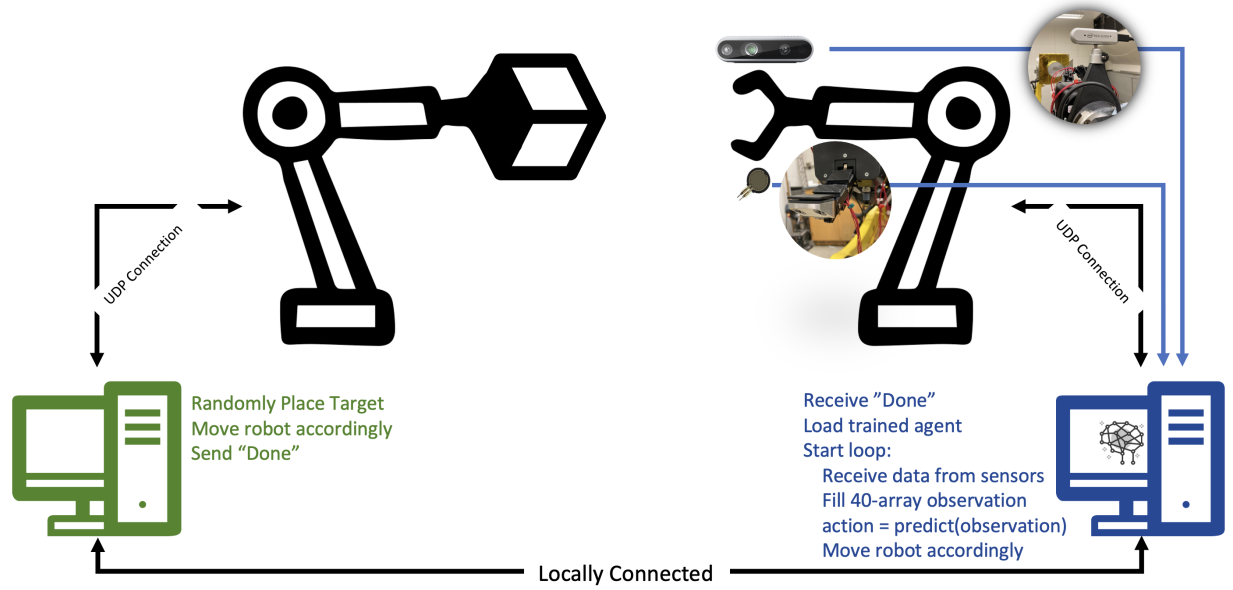


Figure 5.13: Overall structure of the HIL connection. Each component serves a distinct role in achieving precise control over the gripper's motion.

Upon reaching this state, the closed-loop control of the gripper is activated. The feedback loop sources its positional data from an Intel RealSense camera affixed to the gripper, creating an eye-in-hand configuration. This camera is adept at pinpointing the desired grasping section of the target using the pre-trained YOLOX algorithm [210]. The YOLOX algorithm is trained to identify a grasping point on an object by creating a bounding box and analyzing it with image processing and depth sensor data. It integrates this information with data from robot and tactile sensors, allowing the trained agent to decide the next best action based on its simulation training.

The system updates the robot's position and orientation by adding the predicted gripper movement each cycle. This update is performed using the robot's Denavit–Hartenberg table and the Levenberg-Marquardt method [217] to calculate the optimal joint angles for the desired motion. To simplify the process and avoid potential damage when controlling the gripper with a trained agent, the agent was tested exclusively on a stationary free-floating target. The free-floating capability is achieved through the robot's gravity compensation, utilizing ATI force-torque (F/T) sensors attached to each robot's end-effector. These sensors calculate external forces applied to the robot and adjust its movements accordingly.

The target is first randomly positioned in space. Then, the robot equipped with the Robotiq 3F gripper, controlled by the trained agent, positions itself in a grasping pose relative to the free-floating dummy satellite. Fig. 5.14 presents snapshots of the two different scenarios where the target placed randomly in space and the gripper approaching to grasp the target after it has been randomly positioned in space. Videos of all grasp attempts, both simulation and real-world can be found on the GitHub page [218].

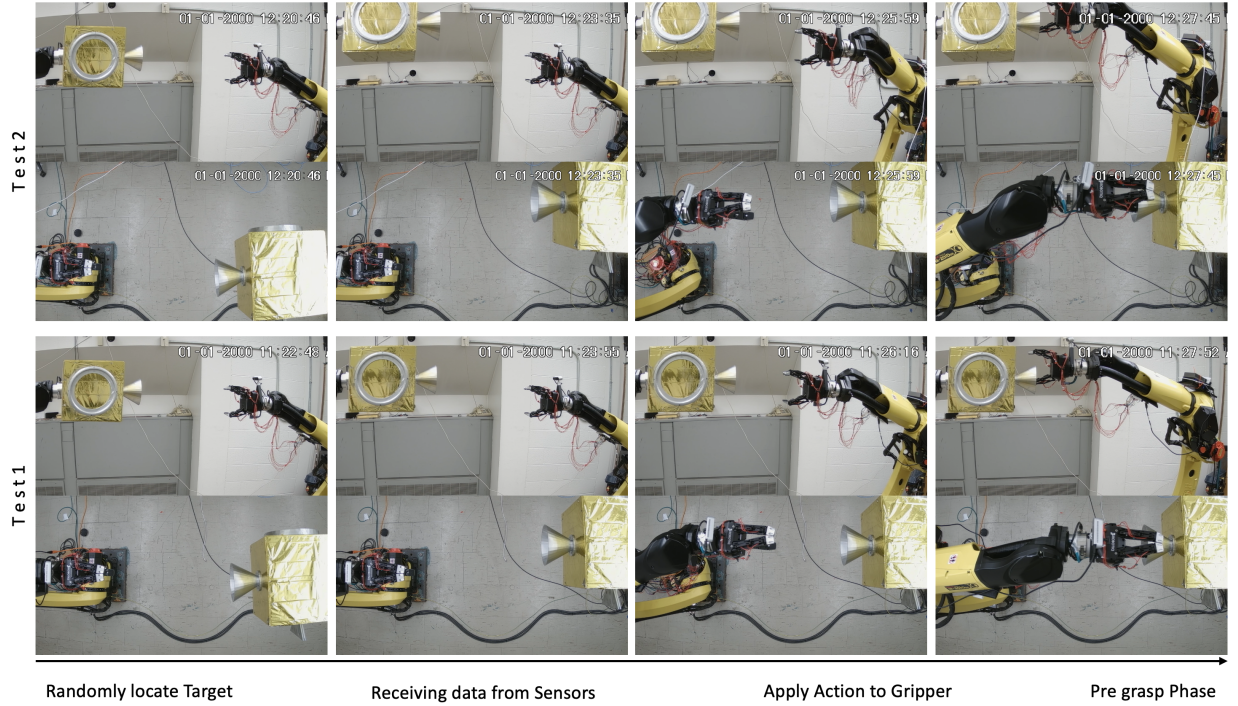


Figure 5.14: Movement of the Target and Gripper in a Real-world Setting: Upon positioning the target at random location, the gripper aligns itself to a pre-grasp position.

Fig. 5.15 illustrates the camera's field of view, strategically positioned to encompass the nozzle of the target satellite. Mounted directly on the gripper, the camera offers an unobstructed and high-resolution visual perspective of the target, facilitating precise alignment and real-time monitoring during the grasping maneuver. This optimal placement ensures that critical features of the target, such as the nozzle, remain within the camera's visual range, thereby enhancing the RL agent's ability to accurately interpret the target's spatial orientation and movement dynamics.

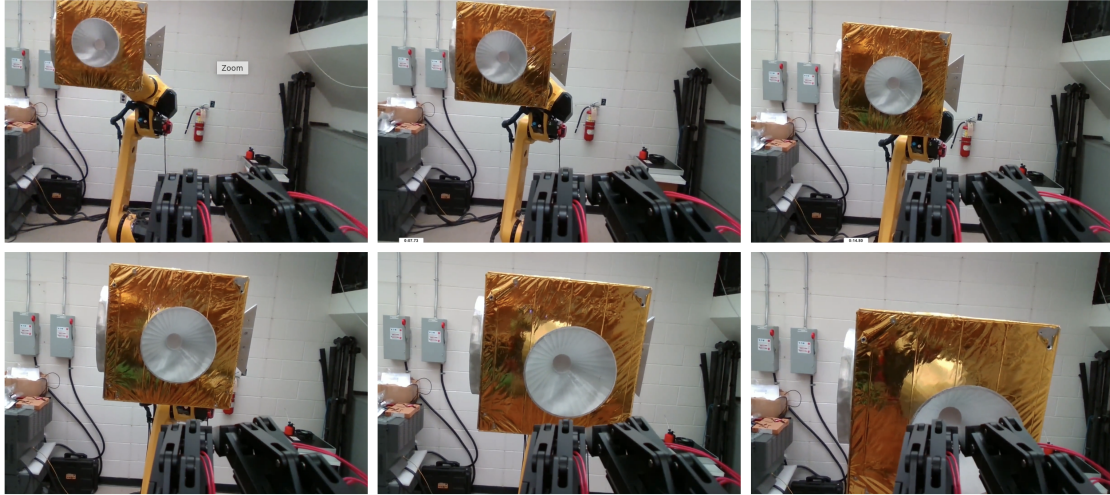


Figure 5.15: Camera field of view illustrating the nozzle of the target satellite during the grasping task.

The experimental results demonstrate the system’s ability to effectively translate the trained agent’s actions from simulation to a real-world environment. Challenges such as sensor integration, real-time communication, and accurate kinematic calculations were addressed to ensure smooth and reliable operation of the robotic manipulators. The successful implementation of the trained agent underscores the potential of the York University Testing Facility as a robust platform for advancing space robotics research.

5.2 University of Luxembourg Testing Facility

The University of Luxembourg hosts a multi-purpose facility designed to emulate diverse space operations and validate robotic control strategies in a terrestrial setting [65]. This “Zero-G Lab”, shown in Fig. 5.16 aims to replicate microgravity-like conditions for on-orbit servicing and debris capture tasks. The facility is equipped with multiple components that collectively enable comprehensive 6-DoF motion emulation and real-time sensing.

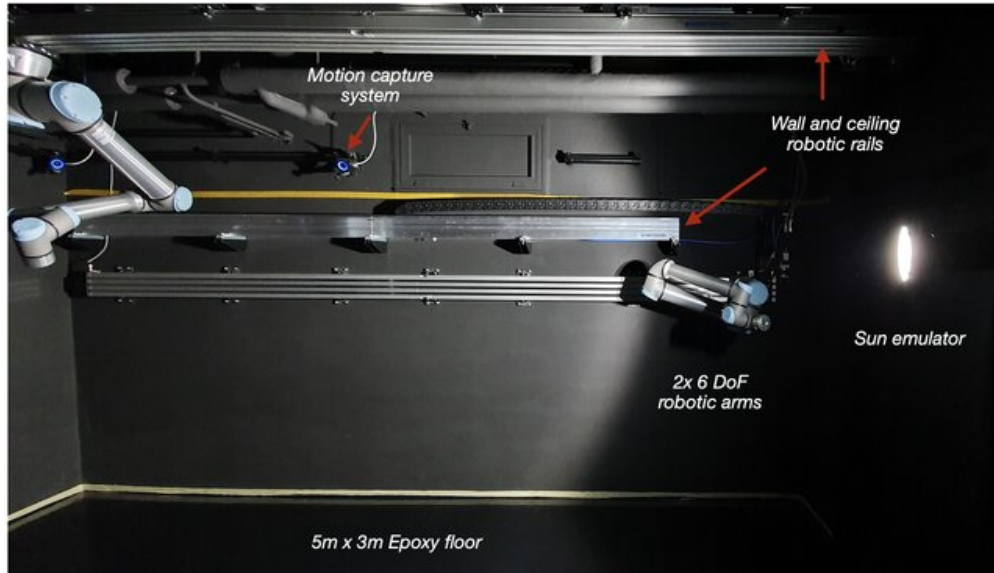


Figure 5.16: University of Luxembourg Zero-G Lab facility, showcasing the rail-mounted robotic arms, motion capture system, and illumination setup.

Central to the Zero-G Lab are two *Universal Robots* UR10e robotic arms, each featuring 6 revolute joints. These arms provide a 1300 mm working radius and can accommodate modular end-effectors such as grippers, cameras, or additional sensors. Each UR10e arm is mounted on a separate rail system furnished by Cobotrails, one fixed to the wall and the other to the ceiling. The motorized rails introduce a seventh DoF to each arm, significantly expanding the workspace and enhancing the capability to simulate two free-flying objects in close-proximity operations. Each rail supports a stroke length of up to 3204 mm (ceiling-mounted) or 4330 mm (wall-mounted), with a repeatability of ± 0.1 mm and a maximum linear velocity of 500 mm/s. Both the ceiling-mounted and wall-mounted arm systems incorporate integrated force sensors at the robot flange, and external force sensors can be mounted if additional precision is required. Each robot-rail system is equipped with an individual teach pendant for localized monitoring and basic control tasks.

Precise target tracking and localization within the Zero-G Lab are enabled by an OptiTrack Motion Capture System comprising eight PrimeX 13W cameras. These cameras operate at up to 240 Hz and use invisible 850 nm infrared illumination with standard 3.5 mm lenses. Reflective markers (either passive or active) are attached to the test objects, and the MCS processes the camera data using triangulation within the *Motive* software. This system provides submillimeter accuracy,

with positional errors below 0.30 mm and rotational errors within $\pm 0.5^\circ$. To establish a local coordinate frame for each tracked object, a minimum of four reflective markers must be attached in a non-coplanar configuration.

To replicate a broad range of orbital lighting conditions, the Zero-G Lab employs a Godox SL-60 Video Light, which operates at 60 W and emits a color temperature of 5600 K. This temperature closely simulates daytime sunlight. Collimators and reflectors can be utilized to modify the light beam, either focusing it to mimic direct sunlight or diffusing it for broader illumination. The light source can be mounted on the robotic arms to create dynamically changing lighting conditions or fixed on a tripod for a static illumination setup.

The Zero-G Lab supports diverse end-effectors for specialized tasks. Notably, a *Robotiq* three-finger adaptive gripper weighing approximately 2.3 kg is available for experiments requiring object capture and manipulation. This gripper can apply forces between 15–60 N, supporting payloads up to 2.5 kg. The adaptive design, in conjunction with the force-feedback sensors, enables delicate handling and robust grasping of noncooperative targets, closely aligning with real on-orbit servicing requirements.

To control the robotic arms and end-effectors, a real-time control loop is established with a communication frequency of 40 Hz. This loop ensures that sensor data are gathered, processed, and transmitted to the robots at a high rate, enabling rapid decision-making and precise motion execution. The control system is implemented using the Robot Operating System (ROS) framework, which provides a modular and extensible architecture for integrating various components. The predicted actions from the trained agent are transmitted to the robot using ROS servoing, which converts the desired Cartesian positions into joint-space commands. This approach ensures that the robot can accurately replicate the agent's actions in real-world scenarios. The Zero-G Lab's framework is shown in Fig. 5.17.

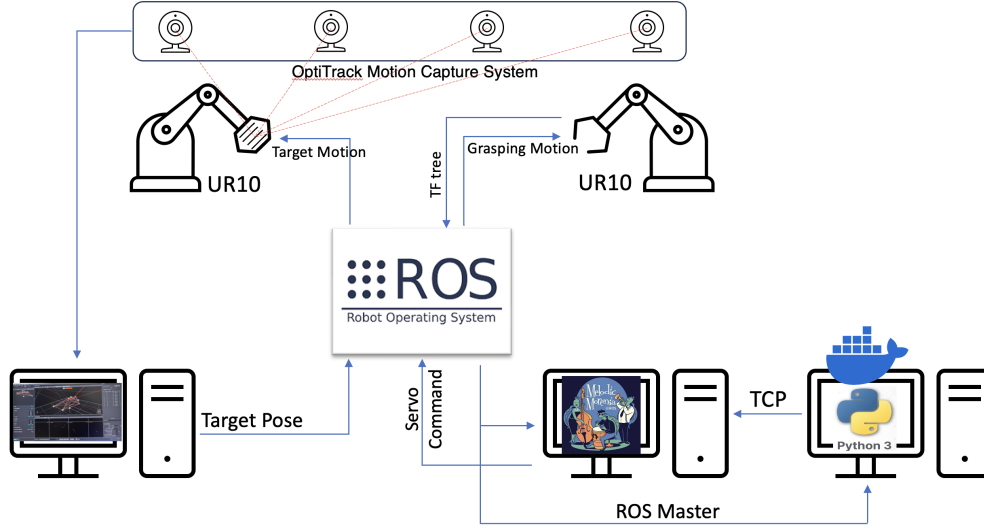


Figure 5.17: Zero-G Lab control framework, illustrating the real-time communication and control loop for the robotic arms and end-effectors.

5.2.1 Curriculum-Based Agent Implementation

In this section, I present the real-world deployment of my curriculum-based agent within the Zero-G Lab. Building upon a robust training phase in simulation, where the agent achieved a success rate exceeding 90%, I ported the trained policy onto physical robots to evaluate its effectiveness in addressing the complexities of real-world scenarios.

Autonomous capture of tumbling or noncooperative targets in space is inherently risky, expensive, and technologically demanding [22, 219]. Consequently, the Zero-G Lab at the University of Luxembourg offers a terrestrial testbed to approximate the conditions of on-orbit servicing [65]. Two rail-mounted UR10e arms were leveraged to emulate the motion of a target satellite and a servicing robot, each having full 6-DoF mobility in position and orientation. One robot carried the target mockup, while the other held the Robotiq three-finger gripper equipped with onboard force sensors at the flange.

Unlike the simulation environment where observations (e.g., target position, minimum distance to gripper palm) can be extracted directly from the simulator state, real-world measurements must be derived from physical sensors. Specifically, the gripper’s pose is computed via joint encoders and

FK, while the OptiTrack system provides the position and orientation of the target with typical errors of $0.30 \pm \text{mm}$ and $\pm 0.5^\circ$. These non-negligible sensor inaccuracies mimic real-mission scenarios more closely, rendering the experimental setup particularly relevant. Fig. 5.18 depicted the target equipped with markers for motion capture and Motive software used for tracking and localization.

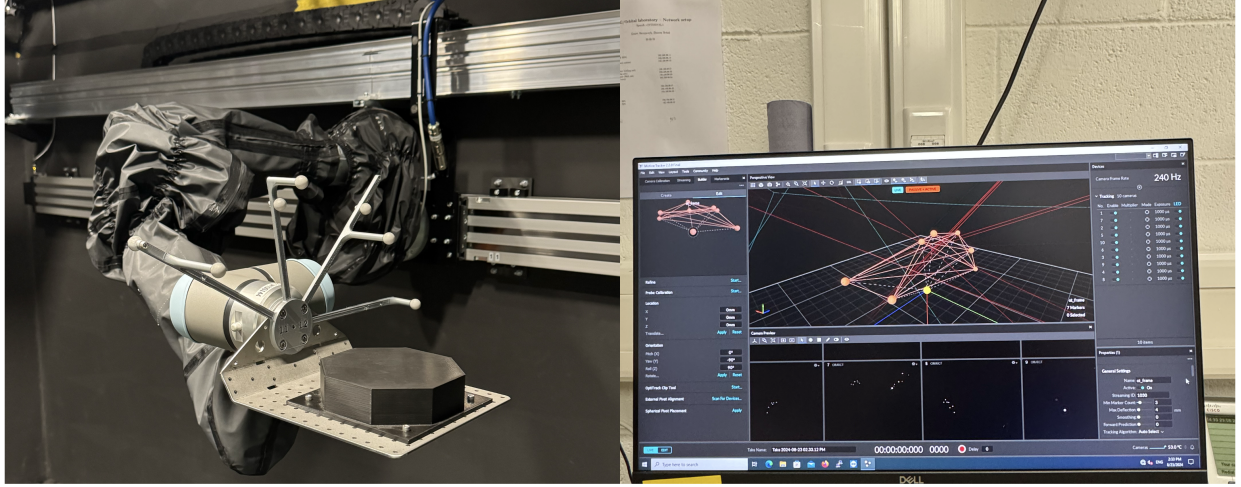


Figure 5.18: OptiTrack motion capture system setup for tracking the target satellite in the Zero-G Lab.

To ensure smooth and timely command execution, the control and communication frequency is set to 40 Hz. Every 0.025 s, sensor data are gathered, fused into an observation vector, and fed into the trained agent’s policy to compute the best deterministic action. This action is then transmitted to the robot carrying the gripper for motion execution. The target’s motion is randomly generated to satisfy two constraints:

1. Maintaining a feasible relative distance and velocity representative of orbital approach maneuvers.
2. Remaining within the workspace limits of the UR10e arm carrying the target mockup.

Fig. 5.19 illustrates the real-world positioning performance of the trained agent. Initially, the target exhibits random linear and angular velocities, such as $[-2.59, 0.37, 2.26] \times 10^{-3} \text{ m/s}$ and $[-0.16, 8.98, 0.10] \times 10^{-3} \text{ rad/s}$, respectively. In response, the gripper maneuvers to synchronize with the target’s motion, guided by the agent’s control outputs. Specifically, subplot (a) displays

the positional control actions applied to the gripper, while subplot (e) shows the orientational control actions. Subplot (b) illustrates the minimum distance between the gripper and the target, demonstrating the agent's effectiveness in reducing this distance. The quaternion difference between the gripper and target orientations is depicted in subplot (f), indicating the agent's ability to achieve precise alignment. Subplots (c) and (g) present the positional and orientational trajectories of both the gripper and the target, respectively, highlighting the dynamic synchronization process. Additionally, subplots (d) and (h) plot the linear and angular velocities of both the gripper and the target, showcasing the agent's adaptive response to changing environmental conditions. Notably, subplot (e) reveals that maximum allowable orientational control actions is required about the y-axis to compensate for the target's rotation and maintain alignment, meaning that the agent dealing with the maximum possible target's angular velocity effectively.

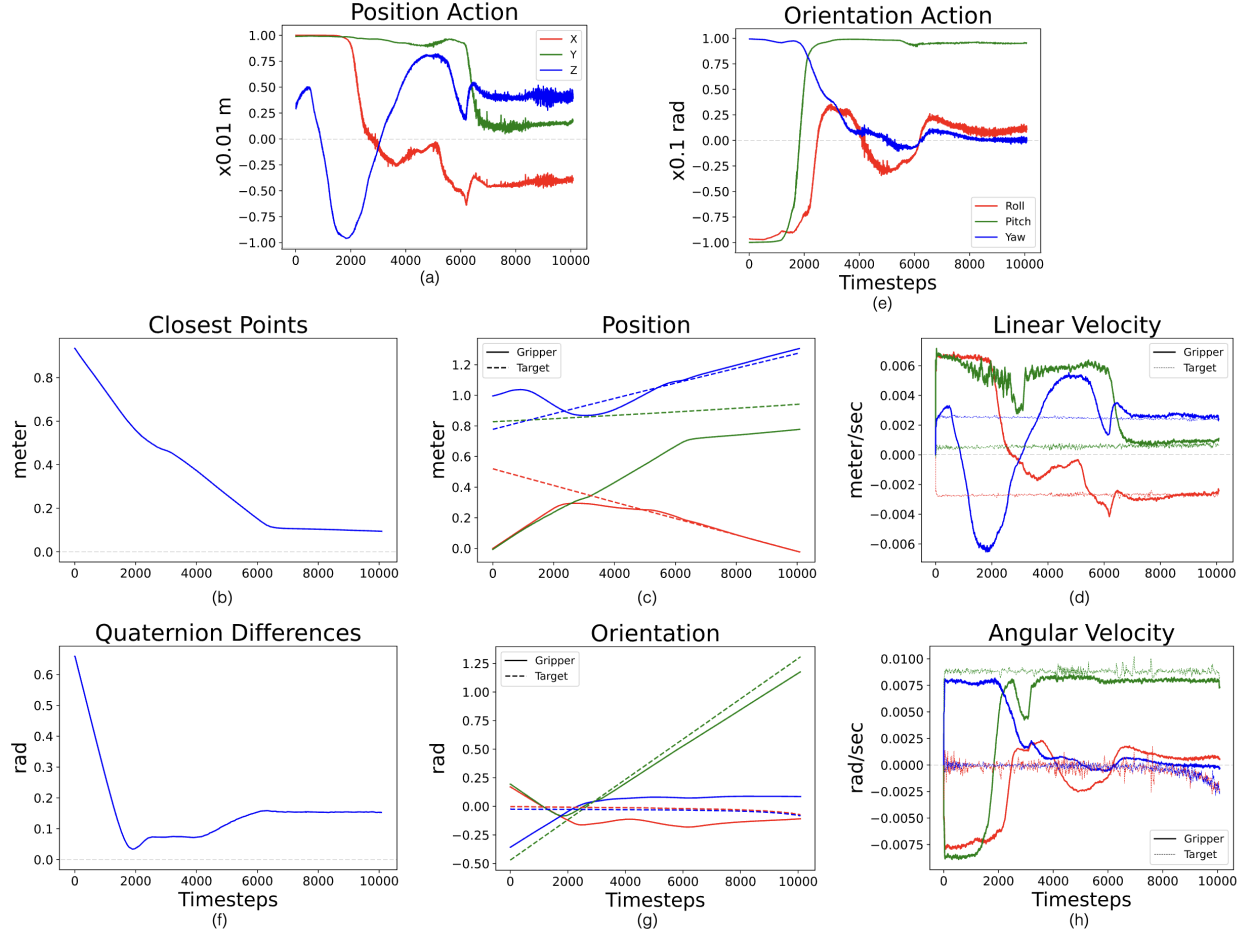


Figure 5.19: Performance analysis of gripper control during a positioning task in a dynamic real-world scenario.

These experiments confirm the agent's robustness in real-world conditions, even when confronted with sensor inaccuracies that deviate from ideal Gaussian noise. As the gripper nears the target, the sensor-induced noise becomes more pronounced, producing high-frequency control adjustments. Such noisy actions can be attributed to the reduced tolerance for alignment errors during the synchronization phase. Nonetheless, the agent effectively mitigates these discrepancies and maintains a stable capture trajectory.

A series of snapshots of two random episodes depicting the real-world experimentation is presented in Fig. 5.20. From an initially random orientation and position, the gripper approaches the moving target, aligns itself with the target's orientation, and transitions into a stable pre-grasp configuration. Videos demonstrating these grasping maneuvers—encompassing both simulation

and real-world trials—are publicly accessible on the project’s webpage

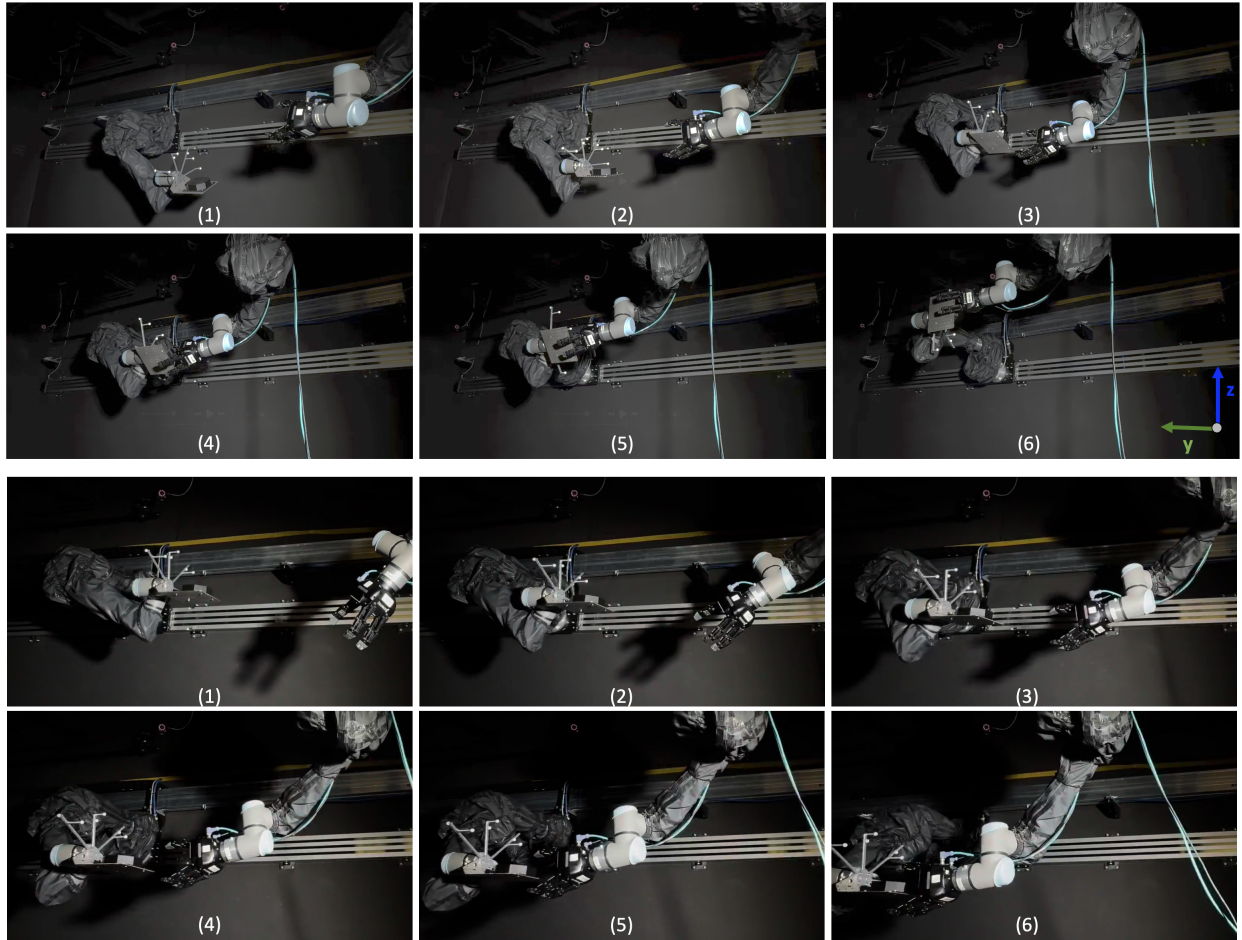


Figure 5.20: Sequential snapshots of two episodes in real-world experiment, showing its controlled approach from a random orientation to a moving, randomly positioned target. The final alignment showcases the agent’s precision in achieving a reliable pre-grasp pose under realistic conditions.

Since the agent is trained in a simulated environment with domain randomization, it is expected to fail in some real-world scenarios. As shown in Fig. 5.21, the agent occasionally fails to predict the correct actions needed to place the gripper in the pre-grasp position and it gets in contact with the moving object.

The agent might be able to recover from the failure, and continue the task, like in simulation environment, but because of the safety reasons, the task is stopped right after the contact happens. The main reason for the failure is the agent is not trained with the exact same conditions as the real-world or simply the agent success rate was not 100% to ensure all possible scenarios are

covered.

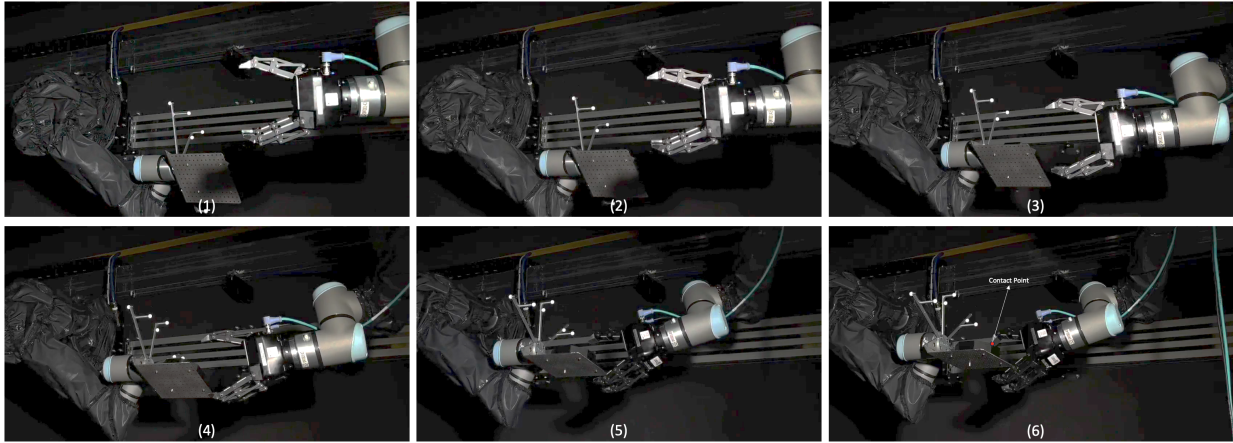


Figure 5.21: An example of the agent’s failure in a real-world scenario. The gripper fails to align with the moving target and makes contact with it.

In summary, the Zero-G Lab’s rail-enabled UR10e robotic arms, high-frequency motion capture system, and dynamic lighting environment provide a compelling experimental framework for verifying the robustness of the curriculum-based agent under realistic constraints. Despite unavoidable sensor errors and modeling discrepancies, the agent demonstrates consistent performance and efficient capture strategies, underscoring the viability of applying learning-based control in near-Earth object manipulation and on-orbit servicing scenarios.

Chapter 6 Conclusion

This dissertation investigated the application of Deep Reinforcement Learning (DRL) to grasp free-floating, moving space debris in microgravity. Unlike traditional robotic manipulation tasks anchored to rigid environments, debris removal missions must cope with floating, tumbling objects that can be easily perturbed. The research addressed these unique challenges by designing and validating learning-based control policies capable of precise, robust, and adaptive grasping under unpredictable dynamics.

6.1 Summary of Key Conclusions

- **Microgravity constraints demand adaptive control.** Through extensive simulations and experiments, I confirmed that even minor contact forces can significantly alter the trajectory of a free-floating target. The developed DRL policies learned to mitigate these effects by actively controlling both the positioning (pre-grasp phase) and the closure dynamics (grasp phase), thereby reducing inadvertent velocity transfer to the target.
- **Domain randomization accelerates sim-to-real transfer.** By systematically randomizing physical properties (e.g., friction, mass, contact stiction) and initial states (poses, velocities) in simulation, the RL agents generalized well to real-world HIL experiments. This approach significantly narrowed the gap between simulation performance and hardware tests.
- **Hierarchical control improves grasp reliability.** Separating the problem into distinct phases (pre-grasp alignment and final finger closure) allowed specialized DRL agents to tackle each sub-task efficiently. This hierarchical design reduced failure modes—particularly collisions during approach—and resulted in high success rates, even with objects of unknown or time-varying properties.

- **Tactile sensing is crucial for precise grasping.** Incorporating real-time force feedback (via thin-film pressure sensors) enabled the agent to modulate closing forces on the fly, avoiding both under-grasp (slippage) and over-grasp (damaging impacts). The ability to detect incipient contact greatly enhanced stability and success rates across diverse target shapes.
- **Hardware-in-the-loop (HIL) validation is essential.** Validating learned policies on two separate HIL platforms (York University’s dual-robot system and the University of Luxembourg’s Zero-G Lab) confirmed that the trained agents handle real sensor noise, latency, and mechanical compliance. The physical tests underscored the maturity and practicality of DRL-based approaches in microgravity emulation environments.

Overall, this work demonstrates that learning-based control can address the inherent uncertainties of space debris manipulation. The proposed strategies achieve robust grasping by blending real-time perception, adaptive control, and hardware integration—paving a path toward fully autonomous on-orbit servicing and debris removal missions.

6.2 Original Contributions to the Field

The following *original contributions* distinguish this research from prior studies and advance the state of the art in space robotics:

1. **High-Fidelity Microgravity Simulation for RL:** A custom microgravity simulation environment was developed—integrating PyBullet with advanced contact models and domain randomization. This environment not only supports standard DRL algorithms (SAC, TD3, DDPG) but also captures the intricate contact dynamics crucial for accurate modeling of free-floating objects. This contribution enables robust, large-scale training without incurring real hardware risks.
2. **Hierarchical DRL Framework for Soft and Hard Capture:** A novel two-stage reinforcement learning approach was formulated, decoupling the *soft capture* (pre-grasp alignment)

from the *hard capture* (finger closure). By training specialized RL agents and fusing them into one hierarchical policy, the system achieved high success rates with minimal debris disturbance—an advance over single-stage methods that often fail in microgravity if contact forces are poorly managed.

3. **Integration of Tactile Feedback into RL Policies:** Although tactile sensing is common in terrestrial manipulation, its use in microgravity tasks is less explored. This research proposed force-awareness in RL by incorporating real-time sensor data from a digitally twinned 3F gripper. Experimental evidence showed significantly improved alignment accuracy and slip prevention—thereby introducing a practical, sensor-based path to safer, more reliable debris captures.
4. **Comprehensive HIL System Enabling Realistic Validation:** Beyond simulation, a hardware-in-the-loop testbed was designed and demonstrated. Featuring two industrial robots (to emulate a free-floater and a chaser), active gravity compensation, and advanced sensor fusion, this system validated RL policies under near-real orbital dynamics. The smooth interface between the industrial controllers and the external PC facilitated repeatable 6-DoF manipulations, broadening the scope for future experimental evaluations in space robotics.

These contributions push the boundaries of debris removal research, offering robust autonomy methods that bridge simulation and physical platforms. In particular, the hierarchical RL schemes and tactile-based feedback strategies lay a foundation for sophisticated multi-phase captures in real mission scenarios.

6.3 Future Research Directions

While this dissertation establishes a promising framework for RL-driven debris capture, there remain open challenges that warrant deeper investigation:

- **Deformable or Irregular Debris:** Future studies can extend the approach to objects with

highly irregular or deformable surfaces, incorporating more complex geometry and partial tumbling rates to reflect real upper-stage rocket bodies or broken satellite fragments.

- **Multi-Manipulator and Multi-Agent Systems:** Investigating cooperative robotic teams for simultaneously capturing multiple debris pieces could drastically reduce total mission time. Approaches such as multi-agent RL might yield synergy in planning intercept trajectories and distributing tasks among multiple end-effectors.
- **Adaptive Navigation and Full Autonomy:** Integrating vision-based orbital navigation with RL-based manipulation remains a key milestone. Incorporating advanced computer vision or large language model (LLM) planners could yield holistic systems capable of *detecting, tracking, and capturing* multiple uncooperative debris objects with minimal ground intervention.
- **Performance under Failures and Uncertainties:** Future work should incorporate partial sensor failures or mechanical malfunctions to develop resilient control policies. By systematically injecting anomalies—akin to real orbital contingencies—agents can learn fault-tolerant behaviors that maintain safe capture under adverse conditions.
- **Scalability to Higher Mass and Larger Structures:** Finally, exploring larger-scale experiments—where target mass and inertia significantly exceed the chaser manipulator’s capabilities—would refine the approach’s boundaries. This may involve advanced momentum-exchange or tether-based methods to detumble massive debris.

In conclusion, this dissertation demonstrates the feasibility and efficacy of DRL-based strategies for on-orbit debris capture, from fundamental simulation designs to real robotic validation. By tackling the distinctive challenges of microgravity operations—instability, partial observability, and dynamic uncertainties—this work lays a robust framework for future missions aimed at making near-Earth space more sustainable and safer for all stakeholders.

Bibliography

- [1] Union of Concerned Scientists, “Ucs usa satellite database.” <https://www.ucsusa.org/resources/satellite-database>, 2023. Accessed: 2023-10-09.
- [2] G. Muntoni, G. Montisci, T. Pisanu, P. Andronico, and G. Valente, “Crowded space: a review on radar measurements for space debris monitoring and tracking,” *Applied Sciences*, vol. 11, no. 4, p. 1364, 2021.
- [3] P. Anz-Meador and J.-C. Liou, “Analysis and consequences of the iridium 33-cosmos 2251 collision,” *National Aeronautics and Space Administration,[Online]. Available: <http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20100008433.pdf>*. [Accessed 18 October 2014], 2010.
- [4] S.-I. Nishida, S. Kawamoto, Y. Okawa, F. Terui, and S. Kitamura, “Space debris removal system using a small satellite,” *Acta Astronautica*, vol. 65, no. 1, pp. 95–102, 2009.
- [5] W. Cheng, L. Tianxi, and Z. Yang, “Grasping strategy in space robot capturing floating target,” *Chinese Journal of Aeronautics*, vol. 23, no. 5, pp. 591–598, 2010. Publisher: Elsevier.
- [6] H. Schaub, L. E. Jasper, P. V. Anderson, and D. S. McKnight, “Cost and risk assessment for spacecraft operation decisions caused by the space debris environment,” *Acta Astronautica*, vol. 113, pp. 66–79, 2015.
- [7] W. Nozawa, K. Kurita, T. Tamaki, and S. Managi, “To what extent will space debris impact the economy?,” *Space Policy*, vol. 66, p. 101580, 2023.
- [8] C. P. Mark and S. Kamath, “Review of active space debris removal methods,” *Space Policy*, vol. 47, pp. 194–206, 2019.

- [9] J. Peng, W. Xu, and H. Yuan, "An efficient pose measurement method of a space non-cooperative target based on stereo vision," *IEEE Access*, vol. 5, pp. 22344–22362, 2017. Conference Name: IEEE Access.
- [10] R. Lampariello, "On grasping a tumbling debris object with a free-flying robot," *IFAC Proceedings Volumes*, vol. 46, no. 19, pp. 161–166, 2013. Publisher: Elsevier.
- [11] J. Luo, L. Zong, M. Wang, and J. Yuan, "Optimal capture occasion determination and trajectory generation for space robots grasping tumbling objects," *Acta Astronautica*, vol. 136, pp. 380–386, 2017.
- [12] J. Yang, X. Hou, Y. H. Hu, Y. Liu, and Q. Pan, "A reinforcement learning scheme for active multi-debris removal mission planning with modified upper confidence bound tree search," *IEEE Access*, vol. 8, pp. 108461–108473, 2020. Conference Name: IEEE Access.
- [13] Y.-H. Wu, Z.-C. Yu, C.-Y. Li, M.-J. He, B. Hua, and Z.-M. Chen, "Reinforcement learning in dual-arm trajectory planning for a free-floating space robot," *Aerospace Science and Technology*, vol. 98, p. 105657, 2020.
- [14] S. Liu, S. Wu, Y. Liu, Z. Wu, and Z. Mao, "Autonomous reinforcement learning control for space robot to capture non-cooperative targets," *Sci. Sin.-Phys. Mech. Astron.*, vol. 49, no. 2, p. 024512, 2019. Publisher: Science China Press.
- [15] G. S. Aglietti, B. Taylor, S. Fellowes, S. Ainley, D. Tye, C. Cox, A. Zarkesh, A. Mafficini, N. Vinkoff, K. Bashford, *et al.*, "Removedebris: An in-orbit demonstration of technologies for the removal of space debris," *The Aeronautical Journal*, vol. 124, no. 1271, pp. 1–23, 2020.
- [16] European Space Agency, "Annual space environment report," 2023.

- [17] R. Biesbroek, L. Innocenti, A. Wolahan, and S. M. Serrano, “e. deorbit-esa’s active debris removal mission,” in *Proceedings of the 7th European Conference on Space Debris*, vol. 10, ESA Space Debris Office, 2017.
- [18] N. L. Johnson and E. G. Stansbery, “The new nasa orbital debris mitigation procedural requirements and standards,” *Acta Astronautica*, vol. 66, no. 3-4, pp. 362–367, 2010.
- [19] J.-C. Liou, “An active debris removal parametric study for leo environment remediation,” *Advances in space research*, vol. 47, no. 11, pp. 1865–1876, 2011.
- [20] M. Matney, A. Manis, P. Anz-Meador, D. Gates, J. Seago, A. Vavrin, and Y.-L. Xu, “The NASA orbital debris engineering model 3.1: development, verification, and validation,” in *International Orbital Debris Conference (IOC)*, 2019. Issue: JSC-E-DAA-TN73945.
- [21] B. Tarran, “Prepare for impact: Space debris and statistics,” *Significance*, vol. 18, no. 3, pp. 18–23, 2021.
- [22] M. Shan, J. Guo, and E. Gill, “Review and comparison of active space debris capturing and removal methods,” *Progress in aerospace sciences*, vol. 80, pp. 18–32, 2016.
- [23] P. Zhao, J. Liu, and C. Wu, “Survey on research and development of on-orbit active debris removal methods,” *Sci. China Technol. Sci.*, vol. 63, no. 11, pp. 2188–2210, 2020.
- [24] C. Pardini and L. Anselmo, “Review of past on-orbit collisions among cataloged objects and examination of the catastrophic fragmentation concept,” *Acta Astronautica*, vol. 100, pp. 30–39, 2014.
- [25] C. Bonnal, J.-M. Ruault, and M.-C. Desjean, “Active debris removal: Recent progress and current trends,” *Acta Astronautica*, vol. 85, pp. 51–60, 2013.
- [26] K. Ota and M. Takahashi, “Release of space debris from flexible nets for active debris removal,” in *AIAA SCITECH 2025 Forum*, p. 1368, 2025.

- [27] M.-H. Jin, C. Zhou, Y.-C. Liu, Z.-Q. Liu, and H. Liu, “Reaction torque control of redundant free-floating space robot,” *International Journal of Automation and Computing*, vol. 14, pp. 295–306, 2017. Publisher: Springer.
- [28] V. Dubanchet, D. Saussié, D. Alazard, C. Bérard, and C. L. Peuvédic, “Modeling and control of a space robot for active debris removal,” *CEAS Space Journal*, vol. 7, no. 2, pp. 203–218, 2015.
- [29] N. Mavrakis, Z. Hao, and Y. Gao, “On-orbit robotic grasping of a spent rocket stage: Grasp stability analysis and experimental results,” *Frontiers in Robotics and AI*, vol. 8, 2021.
- [30] Y. Sun, J. Falco, M. A. Roa, and B. Calli, “Research challenges and progress in robotic grasping and manipulation competitions,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 874–881, 2022. Conference Name: IEEE Robotics and Automation Letters.
- [31] E. Papadopoulos, F. Aghili, O. Ma, and R. Lampariello, “Robotic manipulation and capture in space: A survey,” *Front. Robot. AI*, vol. 0, 2021. Publisher: Frontiers.
- [32] M. M. Castronuovo, “Active space debris removal—a preliminary mission analysis and design,” *Acta Astronautica*, vol. 69, no. 9, pp. 848–859, 2011.
- [33] J. Silha, J.-N. Pittet, M. Hamara, and T. Schildknecht, “Apparent rotation properties of space debris extracted from photometric measurements,” *Advances in space research*, vol. 61, no. 3, pp. 844–861, 2018.
- [34] Y. Yu, F. Yang, H. Yue, Y. Lu, S. Li, and H. Zhao, “Prospects of de-tumbling large space debris using a two-satellite electromagnetic formation,” *Advances in Space Research*, vol. 67, no. 6, pp. 1816–1829, 2021.
- [35] S.-I. Nishida and S. Kawamoto, “Strategy for capturing of a tumbling space debris,” *Acta Astronautica*, vol. 68, no. 1, pp. 113–120, 2011.

- [36] G. Chen, Y. Wang, Y. Wang, J. Liang, L. Zhang, and G. Pan, “Detumbling strategy based on friction control of dual-arm space robot for capturing tumbling target,” *Chinese Journal of Aeronautics*, vol. 33, no. 3, pp. 1093–1106, 2020.
- [37] F. Aghili, “Optimal control of a space manipulator for detumbling of a target satellite,” in *2009 IEEE International Conference on Robotics and Automation*, pp. 3019–3024, 2009. ISSN: 1050-4729.
- [38] W. Cheng, Z. Li, and Y. He, “Strategy and control for robotic detumbling of space debris by using flexible brush,” in *2019 3rd International Conference on Robotics and Automation Sciences (ICRAS)*, pp. 41–47, IEEE, 2019.
- [39] J. Luo, R. Xu, and M. Wang, “Detumbling and stabilization of a tumbling target using a space manipulator with joint-velocity limits,” *Advances in Space Research*, vol. 66, no. 7, pp. 1689–1699, 2020.
- [40] S. Wu, F. Mou, Q. Liu, and J. Cheng, “Contact dynamics and control of a space robot capturing a tumbling object,” *Acta Astronautica*, vol. 151, pp. 532–542, 2018.
- [41] A. Ellery, “Tutorial review on space manipulators for space debris mitigation,” *Robotics*, vol. 8, no. 2, p. 34, 2019.
- [42] S. Peters, C. Pirzkall, H. Fiedler, and R. Förstner, “Mission concept and autonomy considerations for active debris removal,” *Acta Astronautica*, vol. 129, pp. 410–418, 2016.
- [43] J. Deloo and E. Mooij, “Active debris removal: Aspects of trajectories, communication and illumination during final approach,” *Acta Astronautica*, vol. 117, pp. 277–295, 2015.
- [44] C. R. Phipps, “A laser-optical system to re-enter or lower low earth orbit space debris,” *Acta Astronautica*, vol. 93, pp. 418–429, 2014.
- [45] C. Pardini, T. Hanada, and P. H. Krisko, “Benefits and risks of using electrodynamic tethers to de-orbit spacecraft,” *Acta Astronautica*, vol. 64, no. 5-6, pp. 571–588, 2009.

- [46] M. Andrenucci, P. Pergola, A. Ruggiero, J. Olympio, and L. Summerer, “Active removal of space debris-expanding foam application for active debris removal,” *ESA Final Report*, vol. 132, 2011.
- [47] A. Flores-Abad, O. Ma, K. Pham, and S. Ulrich, “A review of space robotics technologies for on-orbit servicing,” *Progress in aerospace sciences*, vol. 68, pp. 1–26, 2014.
- [48] D. J. Kessler, N. L. Johnson, J. Liou, and M. Matney, “The kessler syndrome: implications to future space operations,” *Advances in the Astronautical Sciences*, vol. 137, no. 8, p. 2010, 2010.
- [49] F. Yang, Z. Dong, and X. Ye, “Research on equivalent tests of dynamics of on-orbit soft contact technology based on on-orbit experiment data,” in *Journal of Physics: Conference Series*, vol. 1016, p. 012013, IOP Publishing, 2018.
- [50] L. Santaguida and Z. H. Zhu, “Development of air-bearing microgravity testbed for autonomous spacecraft rendezvous and robotic capture control of a free-floating target,” *Acta Astronautica*, vol. 203, pp. 319–328, 2023.
- [51] T. Rybus, “Obstacle avoidance in space robotics: Review of major challenges and proposed solutions,” *Progress in Aerospace Sciences*, vol. 101, pp. 31–48, 2018.
- [52] K. Seweryn, T. Rybus, J. Lisowski, T. Barciński, M. Ciesielska, K. Grassmann, J. Grygorczuk, M. Krzewski, T. Kuciński, J. Nicolau-Kukliński, *et al.*, “The laboratory model of the manipulator arm (wms1 lemur) dedicated for on-orbit operation,” in *Proc of the I-SAIRAS conference*, 2014.
- [53] W. Xu, B. Liang, Y. Xu, C. Li, and W. Qiang, “A ground experiment system of free-floating robot for capturing space target,” *Journal of intelligent and robotic systems*, vol. 48, pp. 187–208, 2007.

- [54] T. Boge and O. Ma, “Using advanced industrial robotics for spacecraft rendezvous and docking simulation,” in *2011 IEEE international conference on robotics and automation*, pp. 1–4, IEEE, 2011.
- [55] J. Jia, Y. Jia, and S. Sun, “Preliminary design and development of an active suspension gravity compensation system for ground verification,” *Mechanism and Machine Theory*, vol. 128, pp. 492–507, 2018.
- [56] C. R. Carignan and D. L. Akin, “The reaction stabilization of on-orbit robots,” *IEEE Control Systems Magazine*, vol. 20, no. 6, pp. 19–33, 2000.
- [57] M. Shan, J. Guo, E. Gill, and W. Golebiowski, “Validation of space net deployment modeling methods using parabolic flight experiment,” *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 12, pp. 3319–3327, 2017.
- [58] Z. H. Zhu, J. Kang, and U. Bindra, “Validation of cubesat tether deployment system by ground and parabolic flight testing,” *Acta Astronautica*, vol. 185, pp. 299–307, 2021.
- [59] H. Selig, H. Dittus, and C. Lämmerzahl, “Drop tower microgravity improvement towards the nano-g level for the microscope payload tests,” *Microgravity Science and Technology*, vol. 22, pp. 539–549, 2010.
- [60] W. Xu, B. Liang, and Y. Xu, “Survey of modeling, planning, and ground verification of space robotic systems,” *Acta Astronautica*, vol. 68, no. 11-12, pp. 1629–1649, 2011.
- [61] I. Rekleitis, E. Martin, G. Rouleau, R. L’Archevêque, K. Parsa, and E. Dupuis, “Autonomous capture of a tumbling satellite,” *Journal of Field Robotics*, vol. 24, no. 4, pp. 275–296, 2007.
- [62] W. Xu, Y. Liu, B. Liang, Y. Xu, and W. Qiang, “Autonomous path planning and experiment study of free-floating space robot for target capturing,” *Journal of Intelligent and Robotic Systems*, vol. 51, pp. 303–331, 2008.

- [63] Q. Liu, X. Xiao, F. Mou, S. Wu, W. Ma, and C. Hu, “Study on a numerical simulation of a manipulator task verification facility system,” in *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 2132–2137, IEEE, 2018.
- [64] F. Mou, X. Xiao, T. Zhang, Q. Liu, D. Li, C. Hu, and W. Ma, “A hil simulation facility for task verification of the chinese space station manipulator,” in *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 2138–2144, IEEE, 2018.
- [65] M. Olivares-Mendez, M. R. Makhdoomi, B. C. Yalcin, Z. Bokal, V. Muralidharan, M. O. Del Castillo, V. Gaudilliere, L. Pauly, O. Borgue, M. Alandihallaj, *et al.*, “Zero-g lab: A multi-purpose facility for emulating space operations,” *Journal of Space Safety Engineering*, vol. 10, no. 4, pp. 509–521, 2023.
- [66] A. Bicchi and V. Kumar, “Robotic grasping and contact: A review,” in *Proceedings 2000 ICRA. Millennium conference. IEEE international conference on robotics and automation. Symposia proceedings (Cat. No. 00CH37065)*, vol. 1, pp. 348–353, IEEE, 2000.
- [67] K. Kleeberger, R. Bormann, W. Kraus, and M. F. Huber, “A survey on learning-based robotic grasping,” *Current Robotics Reports*, vol. 1, pp. 239–249, 2020.
- [68] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, “Mastering the game of go without human knowledge,” *nature*, vol. 550, no. 7676, pp. 354–359, 2017.
- [69] H. Moravec, *Mind children: The future of robot and human intelligence*. Harvard University Press, 1988.
- [70] J. Falco, K. V. Wyk, S. Liu, and S. Carpin, “Grasping the performance: Facilitating replicable performance measures via benchmarking and standardized methodologies,” *IEEE Robotics & Automation Magazine*, vol. 22, no. 4, p. 125–136, 2015.

- [71] X. Wang and J. Katupitiya, “Robust control of a dual-arm space robot to capture a non-cooperative target in 3d space,” *Aerospace Science and Technology*, vol. 141, p. 108538, 2023.
- [72] S. S. M. Salehian, M. Khoramshahi, and A. Billard, “A dynamical system approach for softly catching a flying object: Theory and experiment,” *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 462–471, 2016.
- [73] H. Xi, B. Chen, R. Zhang, X. Zhang, and M. Luo, “Adaptive optimal concurrent control for detumbling space non-cooperative target via multipoint repeated contact,” *Aerospace Science and Technology*, vol. 154, p. 109473, 2024.
- [74] C. Goldfeder, M. Ciocarlie, J. Peretzman, H. Dang, and P. K. Allen, “Data-driven grasping with partial sensor data,” in *2009 IEEE/RSJ international conference on intelligent robots and systems*, pp. 1278–1283, IEEE, 2009.
- [75] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, “Close-range scene segmentation and reconstruction of 3d point cloud maps for mobile manipulation in domestic environments,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1–6, IEEE, 2009.
- [76] A. Sahbani, S. El-Khoury, and P. Bidaud, “An overview of 3d object grasp synthesis algorithms,” *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, 2012. Publisher: Elsevier.
- [77] R. Ala, D. H. Kim, S. Y. Shin, C. Kim, and S.-K. Park, “A 3d-grasp synthesis algorithm to grasp unknown objects based on graspable boundary and convex segments,” *Information Sciences*, vol. 295, pp. 91–106, 2015. Publisher: Elsevier.
- [78] M. Kallmann, A. Aubel, T. Abaci, and D. Thalmann, “Planning collision-free reaching motions for interactive object manipulation and grasping,” in *ACM SIGGRAPH 2008 classes*, pp. 1–11, Association for Computing Machinery, 2008.

- [79] P. Huang, D. Wang, Z. Meng, F. Zhang, and Z. Liu, “Impact dynamic modeling and adaptive target capturing control for tethered space robots with uncertainties,” *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 5, pp. 2260–2271, 2016.
- [80] A. Mohebbi, M. Keshmiri, and W. F. Xie, “An eye-in-hand stereo visual servoing for tracking and catching moving objects,” in *Proceedings of the 33rd Chinese Control Conference*, pp. 8570–8577, 2014.
- [81] G. Gilardi and I. Sharf, “Literature survey of contact dynamics modelling,” *Mechanism and Machine Theory*, vol. 37, no. 10, pp. 1213–1239, 2002.
- [82] P. Flores and H. M. Lankarani, *Contact force models for multibody dynamics*, vol. 226. Springer, 2016.
- [83] D. Piskorz and K. L. Jones, “On-orbit assembly of space assets: A path to affordable and adaptable space infrastructure,” *The Aerospace Corporation*, pp. 12–13, 2018.
- [84] S. Natsiavas, “Analytical modeling of discrete mechanical systems involving contact, impact, and friction,” *Applied Mechanics Reviews*, vol. 71, no. 5, 2019.
- [85] S. Wu, F. Mou, and O. Ma, “Contact dynamics and control of a space manipulator capturing a rotating object,” in *AIAA Guidance, Navigation, and Control Conference*, p. 1048, 2017.
- [86] M. Sabatini, P. Gasbarri, and G. B. Palmerini, “Coordinated control of a space manipulator tested by means of an air bearing free floating platform,” *Acta Astronautica*, vol. 139, pp. 296–305, 2017.
- [87] M. De Stefano, R. Balachandran, and C. Secchi, “A passivity-based approach for simulating satellite dynamics with robots: Discrete-time integration and time-delay compensation,” *IEEE Transactions on Robotics*, vol. 36, no. 1, pp. 189–203, 2019.

- [88] M. Shibli, F. Aghili, and C.-Y. Su, "Modeling of a free-flying space robot manipulator in contact with a target satellite," in *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.*, pp. 559–564, IEEE, 2005.
- [89] D. Nenchev and K. Yoshida, "Impact analysis and post-impact motion control issues of a free-floating space robot subject to a force impulse," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 548–557, 1999.
- [90] O. Ma, A. Flores-Abad, and T. Boge, "Use of industrial robots for hardware-in-the-loop simulation of satellite rendezvous and docking," *Acta Astronautica*, vol. 81, p. 335–347, 2012.
- [91] Y.-H. Liu, M.-L. Lam, and D. Ding, "A complete and efficient algorithm for searching 3-d form-closure grasps in the discrete domain," *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 805–816, 2004.
- [92] A. Miller, S. Knoop, H. Christensen, and P. Allen, "Automatic grasp planning using shape primitives," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, pp. 1824–1829 vol.2, 2003.
- [93] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp planning via decomposition trees," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 4679–4684, 2007.
- [94] J. Aleotti and S. Caselli, "A 3d shape segmentation approach for robot grasping by parts," *Robotics and Autonomous Systems*, vol. 60, no. 3, p. 358–366, 2011.
- [95] V. Rakesh, U. Sharma, S. Murugan, S. Venugopal, and A. Thondiyath, "Optimizing force closure grasps on 3d objects using a modified genetic algorithm," *Soft Computing*, vol. 22, no. 3, p. 759–772, 2016.

- [96] M. Ciocarlie and P. K. Allen, “Hand posture subspaces for dexterous robotic grasping,” *The International Journal of Robotics Research*, vol. 28, no. 7, p. 851–867, 2009.
- [97] B. Wu, I. Akinola, J. Varley, and P. Allen, “Mat: Multi-fingered adaptive tactile grasping via deep reinforcement learning,” *arXiv preprint arXiv:1909.04787*, 2019.
- [98] I. Lenz, H. Lee, and A. Saxena, “Deep learning for detecting robotic grasps,” *The International Journal of Robotics Research*, vol. 34, no. 4, pp. 705–724, 2015. Publisher: SAGE Publications Ltd STM.
- [99] S. Muravyov and A. Filchenkov, “A cloud-based network of 3d objects for robust grasp planning,” in *Proceedings of the 2020 1st International Conference on Control, Robotics and Intelligent System*, pp. 99–105, 2020.
- [100] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, “Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics,” *arXiv:1703.09312 [cs]*, 2017.
- [101] S. Yang, W. Zhang, W. Lu, H. Wang, and Y. Li, “Learning actions from human demonstration video for robotic manipulation,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1805–1811, IEEE, 2019.
- [102] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
- [103] T. Lampe and M. Riedmiller, “Acquiring visual servoing reaching and grasping skills using neural reinforcement learning,” in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2013. ISSN: 2161-4407.

- [104] D. Quillen, E. Jang, O. Nachum, C. Finn, J. Ibarz, and S. Levine, “Deep reinforcement learning for vision-based robotic grasping: A simulated comparative evaluation of off-policy methods,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6284–6291, 2018. ISSN: 2577-087X.
- [105] Z. Hu, Y. Zheng, and J. Pan, “Grasping living objects with adversarial behaviors using inverse reinforcement learning,” *IEEE Transactions on Robotics*, pp. 1–13, 2023. Conference Name: IEEE Transactions on Robotics.
- [106] B. Mehta, M. Diaz, F. Golemo, C. J. Pal, and L. Paull, “Active domain randomization,” in *Proceedings of the Conference on Robot Learning*, pp. 1162–1176, PMLR, 2020. ISSN: 2640-3498.
- [107] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, “Sim-to-real transfer of robotic control with dynamics randomization,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3803–3810, 2018. ISSN: 2577-087X.
- [108] I. Exarchos, Y. Jiang, W. Yu, and C. Karen Liu, “Policy transfer via kinematic domain randomization and adaptation,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 45–51, 2021. ISSN: 2577-087X.
- [109] Y. Suzuki, K. Koyama, A. Ming, and M. Shimojo, “Grasping strategy for moving object using net-structure proximity sensor and vision sensor,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1403–1409, 2015. ISSN: 1050-4729.
- [110] P. Chen and W. Lu, “Deep reinforcement learning based moving object grasping,” *Information Sciences*, vol. 565, pp. 62–76, 2021.
- [111] O.-M. Pedersen, E. Misimi, and F. Chaumette, “Grasping unknown objects by coupling deep reinforcement learning, generative adversarial networks, and visual servoing,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5655–5662, 2020. ISSN: 2577-087X.

- [112] S. Joshi, S. Kumra, and F. Sahin, “Robotic grasping using deep reinforcement learning,” in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pp. 1461–1466, 2020. ISSN: 2161-8089.
- [113] Y. Hou and J. Li, “Learning 6-dof grasping with dual-agent deep reinforcement learning,” *Robotics and Autonomous Systems*, vol. 166, p. 104451, 2023.
- [114] E. Johns, S. Leutenegger, and A. J. Davison, “Deep learning a grasp function for grasping under gripper pose uncertainty,” in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4461–4468, IEEE, 2016.
- [115] J. Zhang, W. Zhang, R. Song, L. Ma, and Y. Li, “Grasp for stacking via deep reinforcement learning,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2543–2549, IEEE, 2020.
- [116] A. Mousavian, C. Eppner, and D. Fox, “6-dof graspnet: Variational grasp generation for object manipulation,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2901–2910, 2019.
- [117] S. Herland, K. Bach, and E. Misimi, “6-dof closed-loop grasping with reinforcement learning,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7812–7818, 2024.
- [118] A. Murali, A. Mousavian, C. Eppner, C. Paxton, and D. Fox, “6-dof grasping for target-driven object manipulation in clutter,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6232–6238, 2020.
- [119] S. Song, A. Zeng, J. Lee, and T. Funkhouser, “Grasping in the wild: Learning 6dof closed-loop grasping from low-cost demonstrations,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4978–4985, 2020.

- [120] A. Koenig, Z. Liu, L. Janson, and R. Howe, “Tactile grasp refinement using deep reinforcement learning and analytic grasp stability metrics,” *arXiv:2109.11234 [cs, eess]*, 2021.
- [121] A. Saxena, J. Driemeyer, and A. Y. Ng, “Robotic grasping of novel objects using vision,” *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.
- [122] H.-S. Fang, C. Wang, M. Gou, and C. Lu, “GraspNet: A large-scale clustered and densely annotated dataset for object grasping,” *arXiv:1912.13470 [cs]*, 2019.
- [123] C. de Farias, N. Marturi, R. Stolkin, and Y. Bekiroglu, “Simultaneous tactile exploration and grasp refinement for unknown objects,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3349–3356, 2021. Conference Name: IEEE Robotics and Automation Letters.
- [124] V. Kumar, T. Hermans, D. Fox, S. Birchfield, and J. Tremblay, “Contextual reinforcement learning of visuo-tactile multi-fingered grasping policies,” *arXiv preprint arXiv:1911.09233*, 2019.
- [125] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, “High-dimensional continuous control using generalized advantage estimation.”
- [126] S. James, A. J. Davison, and E. Johns, “Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task,” in *Proceedings of the 1st Annual Conference on Robot Learning*, pp. 334–343, PMLR, 2017. ISSN: 2640-3498.
- [127] R. Martin-Martin, M. A. Lee, R. Gardner, S. Savarese, J. Bohg, and A. Garg, “Variable impedance control in end-effector space: An action space for reinforcement learning in contact-rich tasks,” in *2019 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 1010–1017, IEEE, 2019.
- [128] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Bochoon, and S. Birchfield, “Training deep networks with synthetic data: Bridging the

- reality gap by domain randomization,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 969–977, 2018.
- [129] M. Kaspar, J. D. Muñoz Osorio, and J. Bock, “Sim2real transfer for reinforcement learning without dynamics randomization,” in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4383–4388, 2020. ISSN: 2153-0866.
- [130] J. Tobin, L. Biewald, R. Duan, M. Andrychowicz, A. Handa, V. Kumar, B. McGrew, A. Ray, J. Schneider, and P. Welinder, “Domain randomization and generative models for robotic grasping,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3482–3489, IEEE, 2018.
- [131] X. Deng, Y. Xiang, A. Mousavian, C. Eppner, T. Bretl, and D. Fox, “Self-supervised 6d object pose estimation for robot manipulation,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3665–3671, 2020. ISSN: 2577-087X.
- [132] J. Watson, J. Hughes, and F. Iida, “Real-world, real-time robotic grasping with convolutional neural networks,” in *Towards Autonomous Robotic Systems* (Y. Gao, S. Fallah, Y. Jin, and C. Lekakou, eds.), Lecture Notes in Computer Science, pp. 617–626, Springer International Publishing, 2017.
- [133] A. Billard and D. Kragic, “Trends and challenges in robot manipulation,” *Science*, vol. 364, no. 6446, p. eaat8414, 2019.
- [134] M. Q. Mohammed, K. L. Chung, and C. S. Chyi, “Review of deep reinforcement learning-based object grasping: Techniques, open challenges, and recommendations,” *IEEE Access*, vol. 8, pp. 178450–178481, 2020. Conference Name: IEEE Access.
- [135] B. Luo, Y. Yang, and D. Liu, “Adaptive Q -learning for data-based optimal output regulation with experience replay,” *IEEE Transactions on Cybernetics*, vol. 48, no. 12, pp. 3337–3348, 2018. Conference Name: IEEE Transactions on Cybernetics.

- [136] H. v. Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double q-learning,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016. Number: 1.
- [137] A. Ghadirzadeh, A. Maki, D. Kragic, and M. Björkman, “Deep predictive policy training using reinforcement learning,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2351–2358, 2017. ISSN: 2153-0866.
- [138] B. Ma, Z. Jiang, Y. Liu, and Z. Xie, “Advances in space robots for on-orbit servicing: A comprehensive review,” *Advanced Intelligent Systems*, vol. 5, no. 8, p. 2200397, 2023.
- [139] A. Melnik, L. Lach, M. Plappert, T. Korthals, R. Haschke, and H. Ritter, “Tactile sensing and deep reinforcement learning for in-hand manipulation tasks,” in *IROS Workshop on Autonomous Object Manipulation*, 2019.
- [140] Z. Kappassov, J.-A. Corrales, and V. Perdereau, “Tactile sensing in dexterous robot hands — review,” *Robotics and Autonomous Systems*, vol. 74, pp. 195–220, 2015.
- [141] R. Calandra, A. Owens, M. Upadhyaya, W. Yuan, J. Lin, E. H. Adelson, and S. Levine, “The feeling of success: Does touch sensing help predict grasp outcomes?,” *arXiv:1710.05512 [cs, stat]*, 2017.
- [142] P. Chen and W. Lu, “Deep reinforcement learning based moving object grasping,” *Information Sciences*, vol. 565, pp. 62–76, 2021.
- [143] I. Akinola, J. Xu, S. Song, and P. K. Allen, “Dynamic grasping with reachability and motion awareness,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9422–9429, IEEE, 2021.
- [144] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

- [145] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, “Curriculum learning for reinforcement learning domains: A framework and survey,” *Journal of Machine Learning Research*, vol. 21, no. 181, pp. 1–50, 2020.
- [146] G. Hachohen and D. Weinshall, “On the power of curriculum learning in training deep networks,” in *International conference on machine learning*, pp. 2535–2544, PMLR, 2019.
- [147] X. Wang, Y. Chen, and W. Zhu, “A survey on curriculum learning,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 4555–4576, 2021.
- [148] R. Portelas, C. Colas, K. Hofmann, and P.-Y. Oudeyer, “Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments,” in *Conference on Robot Learning*, pp. 835–853, PMLR, 2020.
- [149] Y. Zhang, P. Abbeel, and L. Pinto, “Automatic curriculum learning through value disagreement,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 7648–7659, 2020.
- [150] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, “Learning to walk in minutes using massively parallel deep reinforcement learning,” in *Conference on Robot Learning*, pp. 91–100, PMLR, 2022.
- [151] P. Fournier, O. Sigaud, M. Chetouani, and P.-Y. Oudeyer, “Accuracy-based curriculum learning in deep reinforcement learning,” *arXiv preprint arXiv:1806.09614*, 2018.
- [152] Y. Cao, S. Wang, X. Zheng, W. Ma, X. Xie, and L. Liu, “Reinforcement learning with prior policy guidance for motion planning of dual-arm free-floating space robot,” *Aerospace Science and Technology*, vol. 136, p. 108098, 2023.
- [153] T. Osa, J. Peters, and G. Neumann, “Hierarchical reinforcement learning of multiple grasping strategies with human instructions,” *Advanced Robotics*, vol. 32, no. 18, pp. 955–968, 2018.
- [154] M. Li, K. Hang, D. Kragic, and A. Billard, “Dexterous grasping under shape uncertainty,” *Robotics and Autonomous Systems*, vol. 75, pp. 352–364, 2016.

- [155] Y. Fan, T. Tang, H.-C. Lin, and M. Tomizuka, “Real-time grasp planning for multi-fingered hands by finger splitting,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4045–4052, IEEE, 2018.
- [156] L. Shao, F. Ferreira, M. Jorda, V. Nambiar, J. Luo, E. Solowjow, J. A. Ojea, O. Khatib, and J. Bohg, “Unigrasp: Learning a unified model to grasp with multifingered robotic hands,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2286–2293, 2020.
- [157] H. Liang, L. Cong, N. Hendrich, S. Li, F. Sun, and J. Zhang, “Multifingered grasping based on multimodal reinforcement learning,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1174–1181, 2022.
- [158] I. Akinola, J. Xu, S. Song, and P. K. Allen, “Dynamic grasping with reachability and motion awareness,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 9422–9429, 2021.
- [159] Y. Huang, D. Liu, Z. Liu, K. Wang, Q. Wang, and J. Tan, “A novel robotic grasping method for moving objects based on multi-agent deep reinforcement learning,” *Robotics and Computer-Integrated Manufacturing*, vol. 86, p. 102644, 2024.
- [160] Z. Hu, Y. Zheng, and J. Pan, “Living object grasping using two-stage graph reinforcement learning,” *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1950–1957, 2021.
- [161] F. Zeng, C. Wang, and S. S. Ge, “A survey on visual navigation for artificial agents with deep reinforcement learning,” *IEEE Access*, vol. 8, pp. 135426–135442, 2020.
- [162] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [163] X. Wang, S. Wang, X. Liang, D. Zhao, J. Huang, X. Xu, B. Dai, and Q. Miao, “Deep reinforcement learning: A survey,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

- [164] Y. Li, “Deep reinforcement learning: An overview,” *arXiv preprint arXiv:1701.07274*, 2017.
- [165] H. Nguyen and H. La, “Review of deep reinforcement learning for robot manipulation,” in *2019 Third IEEE international conference on robotic computing (IRC)*, pp. 590–595, IEEE, 2019.
- [166] O. Kroemer, S. Niekum, and G. Konidaris, “A review of robot learning for manipulation: Challenges, representations, and algorithms,” *Journal of machine learning research*, vol. 22, no. 30, pp. 1–82, 2021.
- [167] P. Ladosz, L. Weng, M. Kim, and H. Oh, “Exploration in deep reinforcement learning: A survey,” *Information Fusion*, vol. 85, pp. 1–22, 2022.
- [168] M. Coggan, “Exploration and exploitation in reinforcement learning,” *Research supervised by Prof. Doina Precup, CRA-W DMP Project at McGill University*, 2004.
- [169] O. Nachum, M. Norouzi, K. Xu, and D. Schuurmans, “Bridging the gap between value and policy based reinforcement learning,” in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.
- [170] S. Gu, T. Lillicrap, Z. Ghahramani, R. E. Turner, and S. Levine, “Q-prop: Sample-efficient policy gradient with an off-policy critic,” *arXiv preprint arXiv:1611.02247*, 2016.
- [171] D. Bahdanau, P. Brakel, K. Xu, A. Goyal, R. Lowe, J. Pineau, A. Courville, and Y. Bengio, “An actor-critic algorithm for sequence prediction.”
- [172] Q. Huang, “Model-based or model-free, a review of approaches in reinforcement learning,” in *2020 International Conference on Computing and Data Science (CDS)*, pp. 219–221, IEEE, 2020.
- [173] P. Swazinna, S. Udluft, D. Hein, and T. Runkler, “Comparing model-free and model-based algorithms for offline reinforcement learning,” *IFAC-PapersOnLine*, vol. 55, no. 15, pp. 19–26, 2022.

- [174] S. Fujimoto, H. Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” in *Proceedings of the 35th International Conference on Machine Learning*, pp. 1587–1596, PMLR, 2018. ISSN: 2640-3498.
- [175] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, “Continuous control with deep reinforcement learning.”
- [176] S. Ivanov and A. D’yakonov, “Modern deep reinforcement learning algorithms.”
- [177] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *Proceedings of the 35th International Conference on Machine Learning*, pp. 1861–1870, PMLR, 2018. ISSN: 2640-3498.
- [178] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.
- [179] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, “Soft actor-critic algorithms and applications.”
- [180] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization.”
- [181] 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.
- [182] M. Turchetta, A. Kolobov, S. Shah, A. Krause, and A. Agarwal, “Safe reinforcement learning via curriculum induction,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 12151–12162, 2020.
- [183] S. Luo, H. Kasaei, and L. Schomaker, “Accelerating reinforcement learning for reaching using continuous curriculum learning,” in *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2020.

- [184] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, “A survey of robot learning from demonstration,” *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.
- [185] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, “Recent advances in robot learning from demonstration,” *Annual review of control, robotics, and autonomous systems*, vol. 3, no. 1, pp. 297–330, 2020.
- [186] E. Coumans and Y. Bai, “Pybullet, a python module for physics simulation for games, robotics and machine learning,” 2016.
- [187] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, “Stable-baselines3: Reliable reinforcement learning implementations,” *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.
- [188] G. Brockman, “Openai gym,” *arXiv preprint arXiv:1606.01540*, 2016.
- [189] C. Mower, T. Stouraitis, J. Moura, C. Rauch, L. Yan, N. Z. Behabadi, M. Gienger, T. Vercauteren, C. Bergeles, and S. Vijayakumar, “Ros-pybullet interface: A framework for reliable contact simulation and human-robot interaction,” in *Conference on Robot Learning*, pp. 1411–1423, PMLR, 2023.
- [190] J. Tremblay, T. To, B. Sundaralingam, Y. Xiang, D. Fox, and S. Birchfield, “Deep object pose estimation for semantic robotic grasping of household objects.”
- [191] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler, “Detailed 3d representations for object recognition and modeling,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2608–2623, 2013.
- [192] Y. Yao, G. Cheng, G. Wang, S. Li, P. Zhou, X. Xie, and J. Han, “On improving bounding box representations for oriented object detection,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–11, 2022.

- [193] H. Jahanshahi and Z. H. Zhu, “Review of machine learning in robotic grasping control in space application,” *Acta Astronautica*, 2024.
- [194] C. Rubert, B. León, A. Morales, and J. Sancho-Bru, “Characterisation of grasp quality metrics,” *J Intell Robot Syst*, vol. 89, no. 3, pp. 319–342, 2018.
- [195] M. A. Roa and R. Suárez, “Grasp quality measures: review and performance,” *Autonomous robots*, vol. 38, pp. 65–88, 2015.
- [196] J. P. C. de Souza, L. F. Rocha, P. M. Oliveira, A. P. Moreira, and J. Boaventura-Cunha, “Robotic grasping: from wrench space heuristics to deep learning policies,” *Robotics and Computer-Integrated Manufacturing*, vol. 71, p. 102176, 2021.
- [197] V.-D. Nguyen, “Constructing force-closure grasps,” *The International Journal of Robotics Research*, vol. 7, no. 3, pp. 3–16, 1988.
- [198] C. Ferrari, J. F. Canny, *et al.*, “Planning optimal grasps.,” in *ICRA*, vol. 3, p. 6, 1992.
- [199] Y. Lin and Y. Sun, “Grasp planning to maximize task coverage,” *The International Journal of Robotics Research*, vol. 34, no. 9, pp. 1195–1210, 2015.
- [200] J. Bohg, A. Morales, T. Asfour, and D. Kragic, “Data-driven grasp synthesis—a survey,” *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014. Conference Name: IEEE Transactions on Robotics.
- [201] W. Xu, C. Li, X. Wang, Y. Liu, B. Liang, and Y. Xu, “Study on non-holonomic cartesian path planning of a free-floating space robotic system,” *Advanced Robotics*, vol. 23, pp. 113–143, 2009. Publisher: Taylor & Francis.
- [202] Y. Umetani and K. Yoshida, “Resolved motion rate control of space manipulators with generalized jacobian matrix,” *IEEE Transactions on robotics and automation*, vol. 5, no. 3, pp. 303–314, 1989.

- [203] Y. Xu and T. Kanade, *Space robotics: dynamics and control*, vol. 188. Springer Science & Business Media, 1992.
- [204] K. Yoshida and Y. Umetani, “Control of space manipulators with generalized jacobian matrix,” in *Space Robotics: Dynamics and Control* (Y. Xu and T. Kanade, eds.), The Kluwer International Series in Engineering and Computer Science, pp. 165–204, Springer US, 1993.
- [205] R. T. Icarte, T. Q. Klassen, R. Valenzano, and S. A. McIlraith, “Reward machines: Exploiting reward function structure in reinforcement learning,” *Journal of Artificial Intelligence Research*, vol. 73, pp. 173–208, 2022.
- [206] H. Zhu, J. Yu, A. Gupta, D. Shah, K. Hartikainen, A. Singh, V. Kumar, and S. Levine, “The ingredients of real-world robotic reinforcement learning.”
- [207] Q. Vuong, S. Vikram, H. Su, S. Gao, and H. I. Christensen, “How to pick the domain randomization parameters for sim-to-real transfer of reinforcement learning policies?,” *arXiv preprint arXiv:1903.11774*, 2019.
- [208] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, “Domain randomization for transferring deep neural networks from simulation to the real world,” in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 23–30, IEEE, 2017.
- [209] B. Beigomi, “Fanuc m20id control framework using python and udp connection.” https://github.com/baha2r/Fanuc_M20iD_Control, 2024.
- [210] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: Exceeding YOLO series in 2021.”
- [211] A. S. Sadun, J. Jalani, and F. Jamil, “Grasping analysis for a 3-finger adaptive robot gripper,” in *2016 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA)*, pp. 1–6, IEEE, 2016.
- [212] F. Tao and M. Zhang, “Digital twin shop-floor: A new shop-floor paradigm towards smart manufacturing,” *IEEE Access*, vol. 5, pp. 20418–20427, 2017.

- [213] B. Beigomi, “robotiq3f.py,” May 2023.
- [214] R. L. Cognion, “Rotation rates of inactive satellites near geosynchronous earth orbit,” in *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies Conference*, Maui Economic Development Board Kihei, HI, 2014.
- [215] C. Benson and D. Scheeres, “Averaged rotational dynamics of geo debris,” in *Proceedings of the Eighth European Conference on Space Debris* (T. Flohrer, S. Lemmens, and F. Schmitz, eds.), vol. 8, 2021.
- [216] D. Kucharski, G. Kirchner, F. Koidl, C. Fan, R. Carman, C. Moore, A. Dmytrotsa, M. Ploner, G. Bianco, M. Medvedskij, *et al.*, “Attitude and spin period of space debris envisat measured by satellite laser ranging,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 12, pp. 7651–7657, 2014.
- [217] P. Corke and J. Haviland, “Not your grandmother’s toolbox – the robotics toolbox reinvented for python,” in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11357–11363, 2021. ISSN: 2577-087X.
- [218] B. Beigomi, “Simulation and real-world results, figures, and videos for york university testing facility.” https://github.com/baha2r/Fanuc_Robotiq_Grasp, 2024.
- [219] A. A. Ali, B. Beigomi, and Z. H. Zhu, “Development of 6dof hardware-in-the-loop ground testbed for autonomous robotic space debris removal,” *Aerospace*, vol. 11, no. 11, p. 877, 2024.