Learning to Detumble: Adaptive Post-Capture Stabilization of Uncooperative Space Debris

Álvaro Belmonte-Baeza*[1], Celia Redondo-Verdú[2], José L. Ramón[2], Gabriel J. García[2], Miguel Cazorla[1] and Jorge Pomares[2]

***Abstract*—Active debris removal missions face a critical technical challenge in the autonomous post-capture stabilization of non-cooperative, freely tumbling targets. Traditional model-based control methods often struggle during this phase due to dynamic uncertainties and reliance on accurate state estimates. To address this gap, we propose a Reinforcement Learning (RL) framework specifically designed for the post-capture detumbling of space debris using a dual-arm robotic satellite equipped with a 6D propulsion system. By training the RL agent over a randomized distribution of target states and inertial parameters, our approach handles partial observability and state uncertainty without relying on precise analytical models. Furthermore, the reward formulation explicitly encourages energy-efficient actuation, coordinating the redundant robotic arms, base thrusters, and reaction wheels to actively dissipate the target's kinetic energy. Quantitative evaluations demonstrate that the learned policy effectively dampens multi-axis tumbling motions, achieving near-zero residual velocities across a wide spectrum of initial dynamic states.**

***Index Terms*—Reinforcement Learning, Active Debris Removal, On-Orbit Servicing, Space Robotics**

I. INTRODUCTION

The proliferation of resident space objects has made active debris removal (ADR) and on-orbit servicing (OOS) critical for the sustainable utilization of space [1], [2]. A major technical hurdle in these missions is the autonomous capture and stabilization of non-cooperative, freely tumbling targets, where the usage of space robots is one of the main considered approaches [3]. The robotic capture operation is typically divided into pre-capture approach, contact, and post-capture phases. During the post-capture phase, the servicing spacecraft must actively apply coordinated torques to dampen the residual multi-axis tumbling motion of the combined servicer-target system, preventing severe structural damage and ensuring mission safety [4].

Although extensive research has addressed the planning and control of space manipulator systems (SMSs), dealing with the post-capture detumbling of non-cooperative targets remains an open challenge. Traditional stabilization methods rely heavily on precise analytical models and optimal control theory [4]–[7]. However, these approaches inherently depend on accurate state estimation, such as knowing the exact center of mass, inertia tensor, and friction constraints of the target. In realistic unstructured environments, these parameters are highly uncertain, rendering purely model-based controllers vulnerable to modeling errors and measurement drift.

Reinforcement Learning (RL) [8] has emerged as a powerful alternative to tackle the intense dynamic coupling and uncertainties in free-floating space robots. RL-based policies have shown success for controlling multi-arm systems in combination with optimization-based techniques [9], and directly in a model-free manner [10]. Furthermore, RL-based policies have also been employed for dual-arm space robots approaching non-cooperative targets [11], [12]. However, existing RL applications in this domain primarily address the pre-capture trajectory planning phase. There is a critical gap in extending these adaptive, learning-based techniques to the highly constrained post-capture detumbling phase, where the control policy must continuously stabilize a dynamically uncertain, tumbling mass.

To bridge this gap, we propose a robust RL framework specifically tailored for the post-capture detumbling of uncooperative space debris using a dual-arm robotic platform with a 6D propulsion system. By training an RL agent over a randomized distribution of target states and inertial parameters, our approach exploits the inherent adaptability of deep RL to operate under partial observability and state uncertainty. Furthermore, we include explicit optimization objectives into the reward formulation to encourage energy-efficient actuation in order to leverage both the propulsion system and the arms redundancy to dissipate the target's kinematic energy after the capture.

II. METHODOLOGY

A. System and Task Description

In this work, we consider a robotic platform consisting of a base satellite with two robotic manipulators of ζ degrees of freedom (DoF) each attached to it. The platform's base is actuated using symmetric proportional thrusters and reaction wheels, resulting in the capability of applying a 6D wrench $\boldsymbol{F}_{base} = [\boldsymbol{f}_b, \boldsymbol{\tau}_b] \in \mathbb{R}^6$ to the platform's base center of mass (CoM).

The target debris is chosen to be a cube with handles on two of its faces. The target has a non-zero initial inertial state (i.e., initial linear and angular velocities), and the objective of the robotic platform is to capture it and dissipate its kinetic energy, resulting in a coupled system with zero velocity. An

*Corresponding author: alvaro.belmonte@ua.es
[1] Department of Computer Science and Artificial Intelligence, University of Alicante, Spain
[2] Department of Physics, Systems Engineering and Signal Theory, University of Alicante, Spain

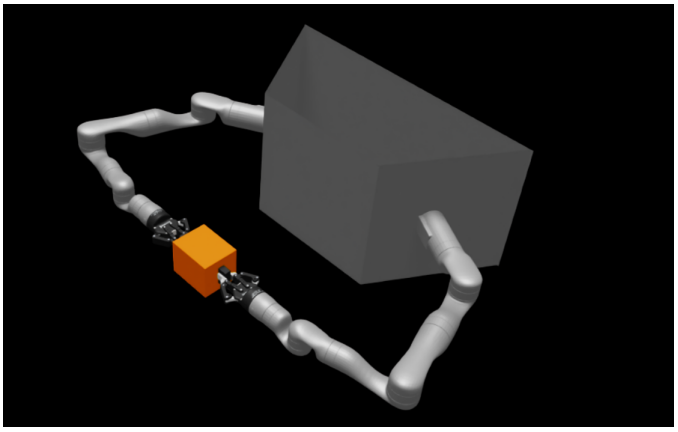

Fig. 1: System and task description: Our bi-manual robotic satellite and the target debris.

image of the described systems and initial task state is shown in Fig. 1.

B. Reinforcement Learning Problem Formulation

We formulate the post-capture detumbling task as a sequential decision-making problem using the Markov Decision Process (MDP) framework [13]. The goal of the RL problem is to find a parametrized policy π_θ that maximizes the expected discounted return. We define the observation and action spaces as described in Table I. The policy outputs two different actions: A vector $\Delta \boldsymbol{q} \in [\Delta q_{min}, \Delta q_{max}]^\zeta$ of desired position change on each arm joint, such that $\boldsymbol{q}^* = \boldsymbol{q}_t + \Delta \boldsymbol{q}$, and a vector $\boldsymbol{F}_b^* \in [-1, 1]^6$ that represents the percentage of applied 6D wrench to the system's base bounded by the maximum force and torque, $\boldsymbol{f}_{max}, \boldsymbol{\tau}_{max}$. The final desired joint positions are then fed to a low-level PD controller running at a higher frequency, and the desired base wrench is fed to a first order dynamics model to avoid instantaneous force changes, such that: $\boldsymbol{F}_{b,t} = \kappa \cdot \boldsymbol{F}_b^* + (1 - \kappa) \cdot \boldsymbol{F}_{b,t-1}$, where κ is the time constant of the dynamics model.

The observations fed to the policy consist of a noisy estimate of the target debris inertial state (linear and angular velocities), the current base velocity and acceleration to inform of the forces exerted to the system, the arm's joint positions and velocities, and the actions outputted by the policy at the previous step.

The reward function must encode the desired behavior of the policy. This is: successfully grasp and reduce the coupled system velocity to a near-zero value. To do so, we defined a modular reward function $\mathcal{R} = \mathcal{R}_{detumble} + \mathcal{R}_{reg} + \mathcal{R}_{energy} + \mathcal{R}_{safe}$, which encompasses the different priorities of the learning agent. The primary reward $\mathcal{R}_{detumble}$ (eq. (1)) aims to reduce both the target and robot velocities. To do so, we employ both an exponential kernel-based reward for the initial phases of the task, and a square root kernel-based term for fine-grained reduction of the velocities to near-zero values.

TABLE I: Definition of the observation and action spaces.

Type	Symbol	Description
Actions	$\Delta \boldsymbol{q}$	Desired joint position change.
	$\boldsymbol{F}_b^*$	Normalized 6D wrench applied to the base CoM.
Observations	$\boldsymbol{v}_{deb}, \boldsymbol{\omega}_{deb}$	Estimate of the target debris velocities.
	$\boldsymbol{v}_b, \boldsymbol{\omega}_b, \boldsymbol{a}_b, \boldsymbol{\alpha}_b$	Current base velocity and acceleration.
	$\boldsymbol{q}, \dot{\boldsymbol{q}}$	Arm's joint positions and velocities.
	$\boldsymbol{a}_{t-1}$	Policy actions at the previous step.

$$
\begin{aligned}
\mathcal{R}_{detumble} &= r_{wide} + r_{fine} \\
r_{wide} &= \sum_i \left(w_{v_i} e^{\left(-\frac{\sum \boldsymbol{v}_i^2}{\sigma_v}\right)} + w_{\omega_i} e^{\left(-\frac{\sum \boldsymbol{\omega}_i^2}{\sigma_\omega}\right)} \right) \\
r_{fine} &= \sum_i \left(w_{v_i,fine} \sum |\boldsymbol{v}_i|^{0.5} + w_{\omega_i,fine} \sum |\boldsymbol{\omega}_i|^{0.5} \right)
\end{aligned} \tag{1}
$$

where $i \in \{b, deb\}$ indicates if the velocity corresponds to the base platform or the debris, σ_v and σ_ω are sensitivity parameters, and the weighting terms w determine the relative importance of each of the reward terms.

The regularization reward $\mathcal{R}_{reg}$ (eq. (2)) encourages the arms to move smoothly by penalizing the joint velocities and torques. It also prevents jitter by encouraging small changes in the actions between time steps.

$$
\begin{aligned}
\mathcal{R}_{reg} = w_{\dot{q}} \sum \dot{\boldsymbol{q}} + w_\tau \sum \boldsymbol{\tau}_{joints} + \\
w_{jitter} \sum ||\boldsymbol{a}_t - \boldsymbol{a}_{t-1}||_2
\end{aligned} \tag{2}
$$

We also include an efficiency reward $\mathcal{R}_{energy}$ (eq. (3)) that promotes low-energy maneuvers by penalizing high joint deltas and wrench commands, minimizing the energy required by the system to complete the capture.

$$
\mathcal{R}_{energy} = w_{arms} \sum \Delta \boldsymbol{q}^2 + w_{th} \sum \boldsymbol{f}_b^2 + w_{rt} \sum \boldsymbol{\tau}_b^2 \tag{3}
$$

Lastly, the safety reward $\mathcal{R}_{safe} = w_C \cdot n_C$ prevents collisions between the target debris and the robotic platform, as well as inner collision between the arms and the main body, where n_C is the number of collisions detected.

III. PRELIMINARY RESULTS

A. Implementation details

We train the policy for the post-capture detumbling task described in section II-A. The specific platform used in our simulation consists of a satellite base of mass $m_b = 300kg$, and two Kinova Gen3 7-DoF arms attached to it, implying that $\zeta = 7$ in our formulation. Regarding the base propulsion system, the thrusters generate a maximum thrust of $\boldsymbol{f}_{max} = 10N$, and the reaction wheels a maximum torque of $\boldsymbol{\tau}_{max} = 5Nm$. The target debris has a mass $m_{deb} = 30kg$,

and the initial inertial state is randomized by sampling from uniform distributions, such that $\boldsymbol{v}_{deb,0} = \mathcal{U}(-\boldsymbol{v}_{max}, \boldsymbol{v}_{max})$, and $\boldsymbol{\omega}_{deb,0} = \mathcal{U}(-\boldsymbol{\omega}_{max}, \boldsymbol{\omega}_{max})$, where $\boldsymbol{v}_{max} = 0.01\frac{m}{s}$ for the three cartesian directions, and $\boldsymbol{\omega}_{max} = 0.05\frac{rad}{s}$ for the three rotational components. We employ curriculum learning so that the sampled velocity values are increased from zero to $\boldsymbol{v}_{max}, \boldsymbol{\omega}_{max}$ linearly through training, reaching the maximum values at 70% of training completion.

B. Post-Capture Detumbling Results

To evaluate the performance of our policy, we carry out both a quantitative and a qualitative experimentation. For the quantitative experiment, we run 1000 different simulations with randomized initial inertial state of the target debris according to the uniform distributions described in section III-A. We deploy the policy for 15s and analyze the linear and angular velocities of both the target and the base platform after capture in order to assess if the detumbling task has been successfully completed. The results obtained show a mean linear velocity after completion of $\bar{\boldsymbol{v}}_{deb} = 1.7 \pm 0.7 \times 10^{-}3\frac{m}{s}$ and $\bar{\boldsymbol{v}}_{b} = 1.6 \pm 0.6 \times 10^{-}3\frac{m}{s}$, and an angular velocity residual of $\bar{\boldsymbol{\omega}}_{deb} = 1.4 \pm 0.7 \times 10^{-}3\frac{rad}{s}$ and $\bar{\boldsymbol{\omega}}_{b} = 1.1 \pm 0.4 \times 10^{-}3\frac{rad}{s}$ for the target debris and the platform, respectively. These results suggest that the policy has successfully learned to detumble the target debris in a wide spectrum of inertial conditions in a robust manner, validating the effectiveness of the proposed RL-based control.

To better visualize the behavior of the policy, we now perform a qualitative experiment by analyzing the whole deployment of the policy in a single sample. We set the initial linear and angular velocities to their maximum values, and evaluate the evolution of both the base and target velocities across an episode of 15s. We also show the evolution of the thrusts and reaction torques generated by the policy to understand their importance in the detumbling task, as well as assessing the impact of the energy minimization reward introduced in section II-B.

Fig. 2 illustrates the evolution of the velocity components during the episode. The initial impact of the interaction of the capture forces exerted by the grippers and the initial inertia of the target causes instability at the beginning of the post-capture phase, and the policy successfully manages the chaos to dampen both the linear and angular velocities of the target (see section III-B), without barely affecting the motion of the base platform (see section III-B), resulting in a satisfactory stabilization. The usage of the thrusters and reaction wheels is critical for this stabilization, as shown in Fig. 3, where it can be seen how the policy learns to compensate the energy induced to the coupled system by the tumbling target, leveraging both the arms redundancy and the base propulsion system to counter the impact of the captured debris and complete the detumbling task.

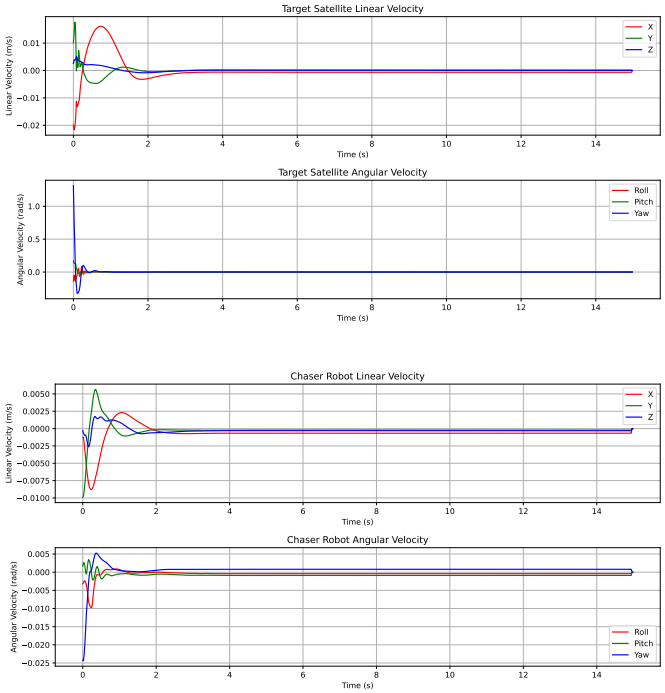

Fig. 2: Linear and angular velocities of the target debris and the base platform

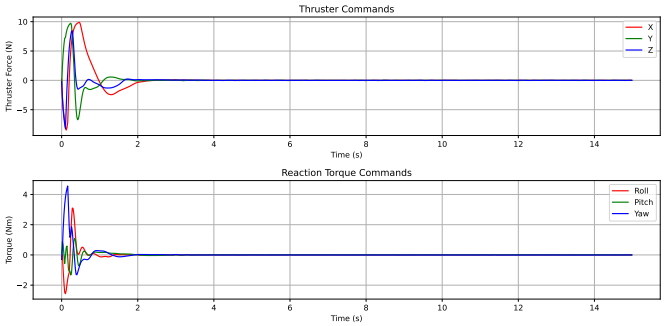

Fig. 3: Thrust forces and reaction torques applied to the base platform.

IV. CONCLUSIONS AND FUTURE WORK

This paper presents a robust Reinforcement Learning framework for the post-capture detumbling of non-cooperative targets in active debris removal. Leveraging a dual-arm robotic platform with a 6D propulsion system, the policy successfully dampens multi-axis tumbling without relying on precise analytical models or accurate state estimations. By coordinating base actuation and redundant arms, the agent actively dissipates kinetic energy while optimizing for efficiency. Evaluations demonstrate near-zero residual velocities across randomized initial states, validating the adaptability of model-free RL in uncertain orbital environments.

Future work will explore constrained RL algorithms to enhance safety, evaluate policies on variable-mass targets, and integrate this detumbling methodology into a complete end-to-end capturing maneuver.

ACKNOWLEDGMENT

This research received funding from the project PID2024-160373OB-C22 funded by MICIU /AEI /10.13039/501100011033 / FEDER, UE. Álvaro Belmonte-Baeza was supported by the Spanish Ministry of Universities under grant FPU21/02586.

APPENDIX

A. Simulation and Training details

We use NVIDIA *Isaac Sim* [14] as the physics simulator, and *Isaac Lab* [15] as the robot learning framework. We use PPO [16] as the RL algorithm, and define the policy π_θ as a Multi-Layer Perceptron (MLP) with three hidden layers of 256 neurons each with hyperbolic activation functions. The policy runs at 50Hz, while the low-level physics engine runs at 200Hz. The training lasts 2000 iterations with a maximum of 750 policy steps per episode (15s), taking about 50 minutes in a desktop machine with a single NVIDIA GeForce RTX3090Ti GPU.

B. Reward weights

TABLE II: Reward Function Weight Parameters

Reward Term	**Weight Symbol**	**Value**
Detumbling Reward ($\mathcal{R}_{detumble}$)		
Linear velocity (wide)	w_{v_i}	5.0
Angular velocity (wide)	w_{ω_i}	5.0
Linear velocity (fine)	$w_{v_i,fine}$	-0.5
Angular velocity (fine)	$w_{\omega_i,fine}$	-0.5
Regularization Reward ($\mathcal{R}_{reg}$)		
Joint velocities	$w_{\dot{q}}$	-0.1
Joint torques	w_τ	-0.001
Action jitter	w_{jitter}	-1.0
Efficiency Reward ($\mathcal{R}_{energy}$)		
Arm joint displacement	w_{arms}	-0.1
Base thruster force	w_{th}	-0.25
Reaction wheel torque	w_{rt}	-0.25
Safety Reward ($\mathcal{R}_{safe}$)		
Collision penalty	w_C	-100.0

REFERENCES

[1] E. Papadopoulos, F. Aghili, O. Ma, and R. Lampariello, "Robotic manipulation and capture in space: A survey," *Frontiers in Robotics and AI*, vol. Volume 8 - 2021, 2021. [Online]. Available: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2021.686723

[2] W.-J. Li, D.-Y. Cheng, X.-G. Liu, Y.-B. Wang, W.-H. Shi, Z.-X. Tang, F. Gao, F.-M. Zeng, H.-Y. Chai, W.-B. Luo *et al.*, "On-orbit service (oos) of spacecraft: A review of engineering developments," *Progress in Aerospace Sciences*, vol. 108, pp. 32–120, 2019.

[3] H. Dong, D. Gangqi, M. Zhiqing *et al.*, "Capture and detumbling control for active debris removal by a dual-arm space robot," *Chinese Journal of Aeronautics*, vol. 35, no. 9, pp. 342–353, 2022.

[4] F. Aghili, "Spinning-base space robot for seamless capture and stabilization of rotating objects," *IEEE Robotics and Automation Letters*, vol. 9, no. 12, pp. 11 593–11 600, 2024.

[5] ——, "Optimal trajectories and robot control for detumbling a non-cooperative satellite," *Journal of Guidance, Control, and Dynamics*, vol. 43, no. 5, pp. 981–988, 2020.

[6] J. Liu, J. Guo, and E. Gill, "Online policy iteration adp-based control of post-capture combined spacecraft without inertia identifications," *Proceedings of the 73rd International Astronautical Congress, IAC*, vol. 2022-September, 2022.

[7] F. Aghili, "Autonomous sequential submanoeuvres in pre- and post-capturing space objects using obstructed 3-d vision data," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 6, pp. 7626–7639, 2023.

[8] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, 2018.

[9] Álvaro Belmonte-Baeza, J. L. Ramón, L. Felicetti, M. Cazorla, and J. Pomares, "Path planning and reinforcement learning-driven control of on-orbit free-flying multi-arm robots," 2026. [Online]. Available: https://arxiv.org/abs/2603.23182

[10] A. Belmonte-Baeza, C. Redondo-Verdú, J. L. Ramón, M. Cazorla, and J. Pomares, "Trajectory optimization with reinforcement learning-driven control of multi-arm robots in on-orbit servicing operations," in *Proceedings of the 1st ESA SPAICE Conference on AI in and for Space*, 9 2024.

[11] Z. Peng and C. Wang, "Reinforcement learning-based pose coordination planning capture strategy for space non-cooperative targets," *Aerospace*, vol. 11, no. 9, 2024. [Online]. Available: https://www.mdpi.com/2226-4310/11/9/706

[12] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963822007726

[13] M. Lauri, D. Hsu, and J. Pajarinen, "Partially observable markov decision processes in robotics: A survey," *IEEE Transactions on Robotics*, vol. 39, no. 1, pp. 21–40, 2023.

[14] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," 2021.

[15] M. Mittal *et al.*, "Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning," *arXiv preprint arXiv:2511.04831*, 2025.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.