

# Reinforcement Learning based 6-DoF Maneuvers for Microgravity Intravehicular Docking: A Simulation Study with Int-Ball2 in ISS-JEM

Aman Arora  
Space Robotics Research Group  
University of Luxembourg  
aman.arora@uni.lu

Matteo El-Hariry  
Space Robotics Research Group  
University of Luxembourg  
matteo.elhariry@uni.lu

Miguel Olivares-Mendez  
Space Robotics Research Group  
University of Luxembourg  
miguel.olivaresmendez@uni.lu

**Abstract**—Autonomous free-flyers play a critical role in intravehicular tasks aboard the International Space Station (ISS), where their precise docking under sensing noise, small actuation mismatches, and environmental variability remains a nontrivial challenge. This work presents a reinforcement learning (RL) framework for six-degree-of-freedom (6-DoF) docking of JAXA’s Int-Ball2 robot inside a high-fidelity Isaac Sim model of the Japanese Experiment Module (JEM). Using Proximal Policy Optimization (PPO) [10], we train and evaluate controllers under domain-randomized dynamics and bounded observation noise, while explicitly modeling propeller drag-torque effects and polarity structure. This enables a controlled study of how Int-Ball2’s propulsion physics influence RL-based docking performance in constrained microgravity interiors. The learned policy achieves stable and reliable docking across varied conditions and lays the groundwork for future extensions pertaining to Int-Ball2 in collision-aware navigation, safe RL, propulsion-accurate sim-to-real transfer, and vision-based end-to-end docking.

## I. INTRODUCTION

Autonomous free-flying robots play a key role in intravehicular operations aboard crewed spacecraft such as the ISS. Platforms like SPHERES [8], Astrobees [11], CIMON [9], and JAXA’s Int-Ball series [7] support inspection, monitoring, and crew assistance. A critical capability for such platforms is autonomous navigation and docking [1, 13], enabling unassisted traversal of cluttered interiors and return to charging stations.

Microgravity environments such as the ISS JEM [7, 13] impose strict requirements: 6-DoF control without GPS, operation in confined spaces, and robustness to disturbances and sensor noise. Classical control methods (e.g. LQR, MPC), although reliable in nominal conditions, often struggle to generalize under uncertainties arising from imperfect state estimation, unmodeled propulsion effects, and variable contact-free dynamics [12], and the recent systems like Astrobees [11] and Int-Ball [7] still rely heavily on pre-programmed control logic.

Reinforcement learning (RL) offers a data-driven alternative for spacecraft guidance, with prior work showing policy resilience to model mismatch, actuator failure, and uncooperative targets [3]. Yet, most prior work focuses on open-space rendezvous or planar docking, leaving several practical challenges unexplored: (i) actuation using multi-propeller

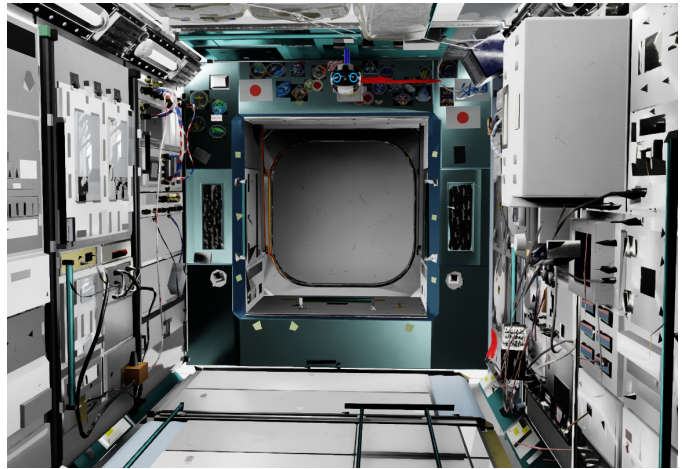


Fig. 1: Isaac Lab simulation of Int-Ball2 performing a 6-DoF docking maneuver inside the ISS-JEM.

platforms rather than thrusters, (ii) tightly constrained intravehicular environments such as the ISS JEM module, and (iii) the physical asymmetries introduced by aerodynamic drag, fan polarity, and geometric actuation coupling. While most existing research focuses on perception and state estimation for free-flyers, learning-based 6-DoF control in confined intravehicular spaces has remained largely unexplored [5].

This work presents a PPO-trained 6-DoF docking controller for JAXA’s Int-Ball2, evaluated in a high-fidelity Isaac Lab simulation of the JEM. Our hypothesis is that an attempt to accurately model the Int-Ball2 propeller drag-torque dynamics and polarity structure materially improves docking stability and final alignment, and that RL can leverage this additional torque channel to achieve robust performance under uncertainty.

To test this, we incorporate task- and robot-level reward shaping, including a physically motivated drag-torque penalty [6], safe-region-aware navigation, and domain-randomized dynamics. The resulting framework enables systematic ablations on drag dynamics, polarity structure, and regularization, and establishes a foundation for future research

extensions pertaining to learned-controller development for Int-Ball2.

## II. METHOD

### A. Problem Formulation.

We formulate the docking approach task as a 6-DoF goal-reaching problem, where the Int-Ball2 robot must align its pose with that of a fixed docking port inside the ISS Kibo module. This is modeled as a Markov Decision Process  $(\mathcal{S}, \mathcal{A}, \mathcal{P}, r, \gamma)$  and trained using actor-critic PPO [10] within our modular simulation framework RoboRAN [2], built on top of NVIDIA Isaac Lab. Robots and tasks are configured independently, allowing flexible robot-task pairing.

### B. DockToStation Environment

#### Observation, Action Spaces and Rewards

The agent has access to the complete state information available in the simulation. The observed quantities are described in Table I. The observation vector consists of calculated errors from the goal, and the velocities in the agent's body frame, as well as the policy's continuous thrust command for the respective propeller, from the preceding timestep. The continuous action space is defined as  $\mathcal{A} = [-1, 1]^{N_{\text{thr}}}$ , representing normalized thrust commands for each of the eight propellers, which are linearly mapped to  $[0, 1]$  prior to actuation. The rewards used for the *DockToStation* task are described in Table II.

The Int-Ball2 is modeled as a rigid body with eight thrusters at positions  $\mathbf{r}_i$  and unit thrust directions  $\mathbf{n}_i$ . For a normalized propeller command  $u_{i,t} \in [0, 1]$ , the aerodynamic force and resulting body torque are

$$\mathbf{f}_i = -u_{i,t} f_{\max,i} \mathbf{n}_i, \quad (1)$$

$$\boldsymbol{\tau}_i = \mathbf{r}_i \times \mathbf{f}_i, \quad (2)$$

where  $f_{\max,i}$  includes the per-propeller thrust-scaling factors provided in the Int-Ball2 propulsion characterization [6]. As demonstrated experimentally in [6], each propeller produces a small reaction torque due to blade-tip aerodynamic drag. The propulsion system intentionally alternates CW/CCW spin polarity so that, when operated symmetrically, these drag torques cancel. Unmodeled or uncompensated drag torque leads to significant yaw drift and undesirable cumulative attitude errors, and is therefore essential to study in both training and simulation.

*a) Penalty for residual drag torque.:* Let  $s_j \in \{-1, +1\}$  denote the spin polarity of the  $j$ -th propeller, and  $D_j$  its thrust magnitude after scaling. Following [6], the simplified net residual drag torque is modeled as

$$\tau_{\text{drag}} = \sum_{j=1}^N s_j k_{\text{drag}} |D_j|, \quad (3)$$

where  $k_{\text{drag}}$  is an empirical coefficient derived from Int-Ball2 torque-response measurements. We have taken the coefficient for 1 propeller from [6], and used the same for all propellers, to calculate the reward  $R_{\text{drag}}$ , in Table II, for this study.

*b) Drag-torque dynamics in simulation.:* We inject the same simplified model into the dynamics: in addition to the standard thrust-induced torque  $\mathbf{r}_j \times \mathbf{f}_j$ , each propeller contributes a drag-torque vector

$$\boldsymbol{\tau}_j^{\text{world}} = s_j k_{\text{drag}} |D_j| \mathbf{a}_j, \quad (4)$$

where  $\mathbf{a}_j$  is the rotor spin axis expressed in the world frame. The total torque applied at each propeller is the sum of these two terms, enabling controlled ablations on (i) the presence of drag-torque dynamics, (ii) polarity structure, and (iii) the drag-penalty term during learning.

## III. EXPERIMENTS

We evaluate four actuation configurations to isolate the role of Int-Ball2's drag-torque mechanism [6] and rotor-polarity structure in RL-based docking performance. For all configurations, the robot is initialized at varying, dynamically reachable poses inside the JEM volume. The details are included in Table III:

Each configuration is trained with three seeds and evaluated in headless deterministic mode over 300 parallel environments (900+ docking attempts per condition). A bounded observation noise is injected during all training runs, where uniform perturbations  $\epsilon \sim \mathcal{U}(-\delta, \delta)$  are independently applied to the four slices of the 15-dimensional observation vector: position error ( $\delta = 0.03$  m), 6D orientation representation ( $\delta = 0.01$ ), linear velocity ( $\delta = 0.03$  m/s), and angular velocity ( $\delta = 0.03$  rad/s). This noise models sensor uncertainty but is not treated as an ablated experimental factor.

### A. Success Definition

We deem a docking attempt as successful if the agent achieves and maintains, for at least 5 consecutive timesteps, both the position and orientation errors below their respective thresholds as defined below:

- a relative position error below 2 cm, and
- an orientation error below  $2^\circ$

with respect to a set target pre-docking pose (i.e., the pose immediately prior to magnetic capture). Orientation error is computed from the trace of the relative rotation matrix reconstructed from the 6D continuous representation used in the observations [14]. For each episode, evaluation metrics are taken at the timestep of minimum positional error close to the end of the episode, which offers a physically meaningful assessment of docking accuracy.

### B. Results

The study shows a clear effect of physically accurate drag-torque modeling and polarity structure on the modelled docking performance. Including drag-torque dynamics with the intended alternating CW/CCW polarity yields consistently stable final alignment and the most reliable docking behavior, which shows that the additional torque channel is both useful and effectively exploited by the learned controller. Removing the drag-torque penalty (Config D) leads to an ideally sparse, however, unstructured actuation pattern (see Figure 2), where

TABLE I: Observation structure for the *DockToStation* task ( $\dim(o_t) = 23$ ; 15 task-specific state variables + 8 thruster commands). All quantities are expressed in the robot body frame.  $\Delta \mathbf{p}$  and  $\Delta \mathbf{q}$  denote the position and orientation errors,  $\mathbf{v}_{\text{lin}}$ ,  $\mathbf{v}_{\text{ang}}$  are body-frame velocities, and  $\mathbf{u}_t$  represents the applied thruster command vector.

Dim	Component	Included Variables
3	Position error	$\Delta \mathbf{p} = [\Delta p_x, \Delta p_y, \Delta p_z]$
6	Orientation error (6D)	$\text{Rot6D}(\Delta \mathbf{q}) = \text{first two columns of the rotation matrix derived from } \mathbf{q}_{\text{cur}}^{-1} \otimes \mathbf{q}_{\text{tgt}}$
3	Linear velocity	$\mathbf{v}_{\text{lin}} = [v_x, v_y, v_z]$
3	Angular velocity	$\mathbf{v}_{\text{ang}} = [\omega_x, \omega_y, \omega_z]$
8	Thruster commands	$\mathbf{u}_{t-1} = [u_{1,t-1}, \dots, u_{8,t-1}]$

TABLE II: Reward components used for training in the *DockToStation* task. The total reward is computed as a weighted sum of all terms  $R_i$ , each scaled by a weight in the task configuration.

Reward Term	Description	Formulation
$R_{\text{pose}}$	Exponential shaping on position and orientation errors ( $e_\theta$ ). Encourages precise 6-DoF alignment with the docking port pose.	$e^{-\ \Delta \mathbf{p}\ _2 / \kappa_p} + e^{-e_\theta / \kappa_o}$
$R_{\text{vel}}$	Encourages bounded linear and angular speeds within nominal limits.	$[\ \mathbf{v}_{\text{lin}}\ _2 - v_{\min}]_0^{v_{\max} - v_{\min}} + [\ \mathbf{v}_{\text{ang}}\ _2 - \omega_{\min}]_0^{\omega_{\max} - \omega_{\min}}$
$R_{\text{boundary}}$	Penalizes deviation from the operational region around the docking port through an exponential boundary envelope ( $d_b$ being the boundary distance).	$e^{-d_b / \kappa_b}$
$R_{\text{prog}}$	Rewards forward progress toward the docking goal between successive timesteps.	$(d_{\text{prev}} - d)(d_{\text{max}} - d)$
$R_{\text{cuboid}}$	Magnitude-proportional penalty applied when the agent exits the defined safe docking envelope ( $\delta_i$ being the cuboid violation offsets).	$-\sqrt{\sum_i \delta_i^2}$
$R_{\text{drag}}$	Penalizes residual torque imbalance arising from CW/CCW drag-torque polarity mismatch.	$-\left  \sum_j s_j k_{\text{drag}} D_j \right $
$R_{\text{act}}$	Action penalty penalizing abrupt variations in thruster commands to encourage smooth control transitions.	$\ \mathbf{u}_t - \mathbf{u}_{t-1}\ _2^2$
$R_{\text{torque}}$	Regularizes the net body torque magnitude to encourage energy-efficient actuation.	$\sum_i \ \boldsymbol{\tau}_i\ _1$

TABLE III: Propulsion-model and reward-shaping configurations used for ablation. DT = drag-torque dynamics in physics; Polarity = rotor spin pattern [6]; Penalty = drag-torque penalty  $R_{\text{drag}}$ .

Config	DT Dynamics	Polarity	Penalty
A (Baseline)	Disabled	Alternating	Enabled
B	Enabled	Alternating	Enabled
C	Enabled	Same (all +1)	Enabled
D	Enabled	Alternating	Disabled

only a subset of propellers is heavily used while others remain largely inactive. This indicates that the penalty is necessary to regularize thrust allocation and maintain the balanced, polarity-consistent propeller usage.

In contrast, enforcing identical spin polarity across all propellers severely degrades fine attitude control, leading to poor convergence and frequent orientation failures. This directly reflects the physical importance of polarity symmetry for passive drag-torque cancellation in Int-Ball2. The thrust-only baseline achieves reasonable positional approach but struggles with precise orientation, highlighting that accurate modeling of the

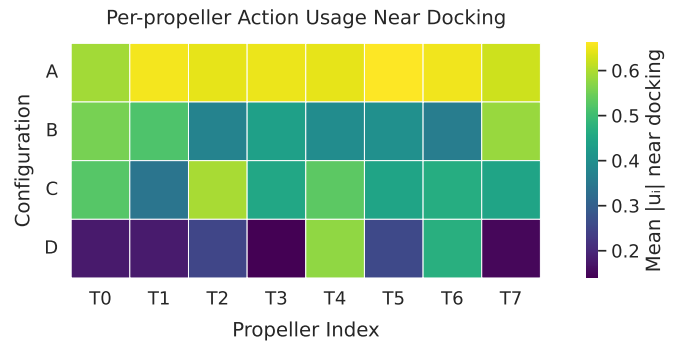


Fig. 2: Mean per-propeller action magnitude  $|u_i|$  over a  $\pm 5$ -timestep window around closest approach (“near docking”), averaged across all seeds and 300 environments for configurations A to D.

drag-torque mechanism materially improves the controllability and stability of the final docking phase.

#### IV. CONCLUSION

This work shows that physically accurate modeling of Int-Ball2’s propeller drag-torque dynamics and polarity structure

TABLE IV: Docking performance across the four configurations described in Section III. Docking Success requires maintaining both position and orientation errors within threshold for at least five consecutive timesteps. Momentary Achievement reports the percentage of episodes that satisfy both thresholds at the evaluated docking point (timestep of minimum distance-to-dock, with orientation evaluated over a  $\pm 5$ -step window).

Config	Final Pos Err (m)	Final Ori Err (deg)	% Pos < thresh	% Ori < thresh	% Momentary Achievement	% Stable Docking Success	% Time Pos < thresh	% Time Ori < thresh
A	0.009	3.90	96.4	38.6	37.6	65.6	6.7	4.6
B	0.008	0.67	98.1	<b>99.1</b>	<b>97.3</b>	97.2	<b>10.7</b>	35.1
C	0.128	7.03	3.0	14.4	0.8	0.1	0.0	2.1
D	<b>0.007</b>	<b>0.60</b>	<b>99.7</b>	97.7	<b>97.3</b>	<b>99.3</b>	9.1	<b>35.4</b>

improves docking stability in constrained intravehicular environments, and that PPO can exploit these effects to achieve reliable 6-DoF docking inside the ISS JEM. The ablations highlight how symmetry and residual-torque regularization shape stable attitude control, underscoring the interaction between actuation physics and learned policies. Looking forward, safety-aware maneuvering of Int-Ball2 inside the JEM remains an open challenge, motivating extensions toward collision-aware navigation and constrained RL methods that enforce safety and orientation limits directly. Real-world experiments using our modular sim-to-real framework [2] will further enable hardware validation and pave the way for deployable autonomy in next-generation intravehicular free-flyers.

#### ACKNOWLEDGMENTS

The Int-Ball2 visual model and the JEM module used in our simulator were adapted from the open-source `int-ball2_isaac_sim` package by SpaceData Inc. [4]. The dynamics, control, and reward logic are our own.

#### REFERENCES

- [1] Roberto Carlino, Andres E Mora Vargas, J Benavides, J Barlow, H Orosco, A Katterhagen, and S Kanis. Lessons learned from astrobbee operations on the international space station. In *International Space Station Research and Development Conference*, 2024.
- [2] Matteo El-Hariry, Antoine Richard, Ricard Marsal, Luis Felipe Wolf Batista, Matthieu Geist, Cédric Pradalier, and Miguel Olivares-Mendez. Roboran: A unified robotics framework for reinforcement learning-based autonomous navigation. *Transactions on Machine Learning Research*.
- [3] Kirk Hovell and Steve Ulrich. Deep reinforcement learning for spacecraft proximity operations guidance. *Journal of spacecraft and rockets*, 58(2):254–264, 2021.
- [4] SpaceData Inc. `int-ball2_isaac_sim` (version v1.0.0) [source code]. GitHub, 2025. Available at: [https://github.com/sd-robotics/int-ball2\\_isaac\\_sim](https://github.com/sd-robotics/int-ball2_isaac_sim).
- [5] Suyoung Kang, Ryan Soussan, Daekyeong Lee, Brian Coltin, Andres Mora Vargas, Marina Moreira, Katie Browne, Ruben Garcia, Maria Bualat, Trey Smith, et al. Astrobbee iss free-flyer datasets for space intra-vehicular robot navigation research. *IEEE Robotics and Automation Letters*, 9(4):3307–3314, 2024.
- [6] S. Mitani, T. Nishishita, and D. Hirano. Int-ball2: Compact high-torque propulsion system actively utilizes propeller air drag polarity. *Proceedings of the 33rd Astrodynamics Symposium (in Japanese)*, 2023.
- [7] Shinji Mitani, Masayuki Goto, Ryo Konomura, Yasushi Shoji, Keiji Hagiwara, Shuhei Shigeto, and Nobutaka Tanishima. Int-ball: Crew-supportive autonomous mobile camera robot on iss/jem. In *2019 IEEE Aerospace Conference*, pages 1–15. IEEE, 2019.
- [8] Swati Mohan, Alvar Saenz-Otero, Simon Nolet, David W Miller, and Steven Sell. Spheres flight operations testing and execution. *Acta Astronautica*, 65(7-8):1121–1132, 2009.
- [9] Hans-Christian Schmitz, Frank Kurth, Kevin Wilkinghoff, Uwe Müllerschowski, Christian Karrasch, and Volker Schmid. Towards robust speech interfaces for the iss. In *Companion Proceedings of the 25th International Conference on Intelligent User Interfaces*, pages 110–111, 2020.
- [10] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [11] Trey Smith, Jonathan Barlow, Maria Bualat, Terrence Fong, Christopher Provencher, Hugo Sanchez, and Ernest Smith. Astrobbee: A new platform for free-flying robotics on the international space station. In *International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS)*, number ARC-E-DAA-TN31584, 2016.
- [12] Massimo Tipaldi, Raffaele Iervolino, and Paolo Roberto Massenio. Reinforcement learning in spacecraft control applications: Advances, prospects, and challenges. *Annual Reviews in Control*, 54:1–23, 2022.
- [13] Seiko P Yamaguchi, Tatsuya Yamamoto, Hideyuki Watanabe, Riichi Itakura, Masaru Wada, Shinji Mitani, Daichi Hirano, Keisuke Watanabe, Taisei Nishishita, Yuta Kawai, et al. Int-ball2: Iss jem internal camera robot with increased degree of autonomy—design and initial checkout. In *2024 International Conference on Space Robotics (iSpaRo)*, pages 328–333. IEEE, 2024.
- [14] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5745–5753, 2019.