POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Deep Reinforcement Learning Aided Robotics for Uncooperative Space Asset Grasping and In-Orbit Servicing

Laurea Magistrale in Space Engineering - Ingegneria Spaziale

Author: Matteo D'Ambrosio

Advisor: Michèle Lavagna

Co-advisors: Lorenzo Capra, Andrea Brandonisio

Academic year: 2022-2023

1. Introduction

The ever-increasing exploitation of space has led to a soar in the number of artificial objects that are currently orbiting our planet, at an alarming rate. The ability to repair or dispose of these systems at their end-of-life has become a priority objective for leading space agencies, as they pose substantial threats to the longevity of operational satellites and of forthcoming space endeavors. Space robotic manipulators have emerged as promising solutions to tackle these impending problems, and with improvements towards their autonomy, reliability, and reactivity, could one day provide a universal solution for both In-Orbit Servicing (IOS) and Active Debris Removal (ADR) missions. With the purpose of achieving similar goals, the recent applications of Deep Reinforcement Learning (DRL) in the space field show that these methodologies can be used to train autonomous agents, that can provide increased robustness and adaptability when introduced into spacecraft Guidance, Navigation, and Control (GNC) systems. While the recent surges in DRL show that it is a field with vast potential, its applications to Space Robots in the literature are still at a preliminary stage, and need to be extended.

2. Related works

The application of DRL to enhance the Guidance and Control (G&C) of robotic arms in space emerges as a potential solution to overcome the main difficulties encountered in classic manipulator control strategies. Among these, there is the ill-posed inverse kinematics problem for high-Degree of Freedom (DoF) redundant manipulators, as well as the great complexities in including additional objectives such as collision avoidance and motion constraints into the robotic arm trajectory. In [1] DRL is used to plan the trajectory of only 6 of the 7-DoFs of the considered manipulator, to achieve a simple end-effector positioning and attitude alignment objective. An obstacle avoidance objective is added in [2], but is only demonstrated for the path-planning of a 3-DoF robotic arm. In [3], a 6-DoF manipulator is trained to achieve the position tracking of circular trajectories with the end-effector, despite fully neglecting its attitude.

3. Problem statement

Fig. 1 reports the main phases of an Orbital Robotics Mission (ORM) for the capture of an uncooperative Target. This is one of the most

complex scenarios, where the use of DRL strategies could be the most beneficial. The focus of this thesis is on the pre-capture phase, where the state of the Space Robot (SR), which mounts a 7-DoF manipulator, needs to be synchronized with the target. The objective is to maneuver the end-effector of the robotic arm such that it is held stationary and at a fixed offset with respect to the desired grasping position on the target, which is critical to provide greater simplicity and safety in the subsequent contact and capture operations.

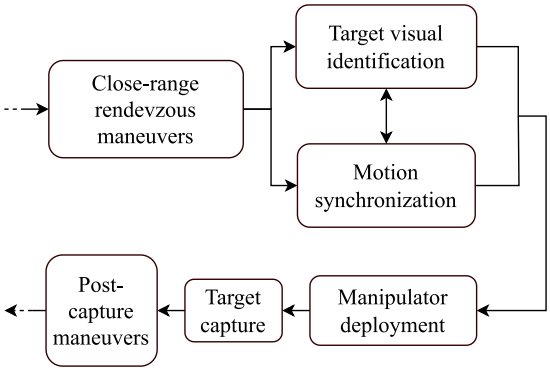

Figure 1: Mission phases.

3.1. Research objectives

The primary goal of this work is to evaluate whether an agent can be trained to autonomously generate the trajectory of a 7-DoF redundant manipulator through the state-of-the-art Proximal Policy Optimization (PPO) DRL algorithm, to guide the end-effector towards a desired goal state. The aim is to highlight what level of accuracy can be achieved in terms of the end-effector positioning and attitude alignment, as well as the overall consecutive duration that this performance can be guaranteed for, as the target spins. Autonomous DRL agents have been found to be highly adaptive and robust when applied to different G&C problems in the space field, hence an additional objective of this work is to evaluate whether these same characteristics are also found in an agent trained to guide a robotic manipulator, especially when the training and testing environments have large domain gaps. Finally, the final part of this work is aimed at providing some preliminary understanding towards the agent's tracking capabilities with the end-effector, when there is substantial relative motion between the SR and the mission target, and to see how the performance in this scenario compares to the previous results. These objectives have been formulated as a set of research questions, which are answered in the conclusions of the thesis based on the obtained results.

4. Space Robot dynamics

The SR is modeled as a multi-body system with a 6-DoF base and a 7-DoF manipulator. Within the scope of this study, the SR is assumed to be Free-Flying, meaning that the base of the SR can be actively controlled, in contrast to the Free-Floating case [4]. By employing a direct-path kinematics approach, and a Newton-Euler dynamics formulation, the nonlinear time-variant equations of motion are retrieved (see Eq. 1) [5].

$$\boldsymbol{H}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} = \boldsymbol{\tau} \tag{1}$$

where $\boldsymbol{H} \in \mathbb{R}^{13\times13}$ is the symmetric, positive-definite Generalized Inertia Matrix (GIM), $\boldsymbol{C} \in \mathbb{R}^{13\times13}$ is the Convective Inertia Matrix (CIM) containing the Coriolis and centrifugal forces, and $\boldsymbol{\tau} \in \mathbb{R}^{13}$ is the vector of generalized forces. The selected vector of generalized variables is [5]

$$\boldsymbol{q} = [\boldsymbol{q}_0, \boldsymbol{q}_m]^\top = [\boldsymbol{r}_0, \boldsymbol{R}_0, \boldsymbol{q}_m]^\top \tag{2}$$

where $\boldsymbol{r}_0$ and $\boldsymbol{R}_0$ are the 6-DoFs corresponding respectively the Center of Mass (CoM) position and attitude of the base, employing a quaternion representation, and $\boldsymbol{q}_m$ collects the 7 joint angles of the robotic arm.
The SR is described through a file in the URDF (Unified Robotics Description Format), and the whole model is implemented in Simulink through the Simscape Multibody library, which is responsible for integrating the equations of motion. Explicit representations of the kinematic and dynamic quantities of the multi-body SR are retrieved through the MATLAB library SPART (SPAce Robotics Toolkit) [5]. The implementation is validated by cross-checking the outputs from the two libraries.

4.1. Target dynamics

Without loss of generality, the target's shape is chosen as a central cylinder, with two protruding solar panels. The target's CoM is positioned at the origin of the LVLH reference, hence its translational motion can be disregarded. Its attitude evolution is described through the Euler

Equations in principal body axes, as in Eq. 3.

$$\boldsymbol{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \boldsymbol{I}\boldsymbol{\omega} = \boldsymbol{M} \tag{3}$$

where $\boldsymbol{I}$ is the target's diagonal inertia matrix, $\boldsymbol{\omega}$ is its angular velocity in the principal axis frame, and $\boldsymbol{M}$ is the vector of external torques, assumed to be null in this work.

5. GNC implementation

The SR GNC architecture employed in this thesis is found in Fig. 2. The autonomous PPO agent generates the guidance of the manipulator and provides it to the controller. The control block is used to close the feedback loop and to actuate the system, in order to retrieve the next simulation step. Additionally, this thesis operates under the assumption of having full knowledge of the variables that enter the G&C blocks. A physical navigation block responsible for producing these quantities has been omitted since it is outside the scope, but analyses including modeled uncertainty have been carried out.

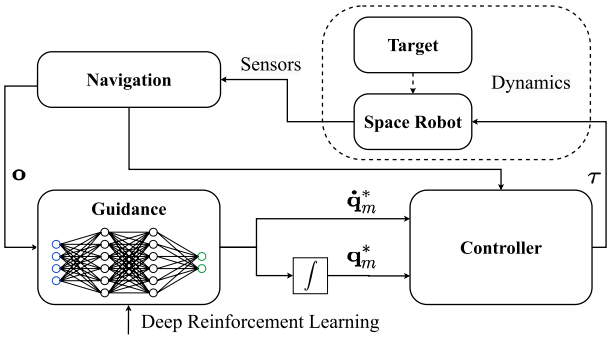

Figure 2: Space Robot GNC architecture.

5.1. DRL guidance

The SR G&C problem is formulated as a Partially Observable Markov Decision Process (POMDP), and is then solved through the PPO DRL algorithm, which has demonstrated its ability to solve high-dimensional continuous control problems effectively. DRL is built on the following concepts: the autonomous agent, the environment, the policy, the reward function, and the observation and action spaces. In short, the agent carries out trial-and-error interactions with the environment, and receives a higher reward the closer it gets to its objective. As the training progresses, the manipulator's guidance law (i.e. the policy) is optimized such that the overall reward obtained by the agent is maximized. The policy is embedded in a Feedforward Neural Network (FNN) that only carries out the guidance tasks; when given as input the observation vector $\boldsymbol{o}$ (see Eq. 4), it produces a set of actions $\boldsymbol{a}$ as an output (see Eq. 5).

$$\boldsymbol{o} = [\boldsymbol{q}_m, \dot{\boldsymbol{q}}_m, \tilde{\boldsymbol{r}}, \tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{v}}, \tilde{\boldsymbol{\omega}}]^\top \tag{4}$$

where the first two vectors correspond to the current joint angles and rates, and the remaining vectors represent the errors between the current and desired Cartesian end-effector states.

$$\boldsymbol{a} = \dot{\boldsymbol{q}}^*_m = [\dot{\phi}_1, \dot{\phi}_2, \dot{\phi}_3, \dot{\phi}_4, \dot{\phi}_5, \dot{\phi}_6, \dot{\phi}_7]^\top \tag{5}$$

where $\dot{\phi}_i$ correspond to the desired joint rates of the manipulator, and are integrated outside of the policy such that both the angle and rate setpoints can be provided to the controller. Since a stochastic policy is selected to circumvent local minima during optimization, both the mean and standard deviation of each $\dot{\phi}_i$ is outputted from the guidance. After optimization, only the mean action values are considered.

5.2. Reward function

The reward function has been designed with the primary goal of producing an autonomous agent that can guide the end-effector towards the desired state, in terms of both position and attitude, through the motion of the manipulator joints. The reward function in Eq. 6 is based on the concept of Artificial Potential Field, and is developed starting from the one found in [1].

$$\begin{gathered} U_k = -\tilde{r} + \frac{10}{1+\tilde{r}_{ax}} + \frac{10}{1+\tilde{r}_{tx}} + \frac{10}{1+\tilde{\theta}} \\ \Delta U = U_k - U_{k-1} \\ r_k = \begin{cases} \Delta U & \text{if } \Delta U \geq 0 \\ 1.5\Delta U & \text{if } \Delta U < 0 \end{cases} \end{gathered} \tag{6}$$

where r_k is the reward given to the agent at each timestep, $\tilde{r}$ is the error between the current and desired end-effector positions, $\tilde{r}_{ax}$ and $\tilde{r}_{tx}$ are the projections of $\tilde{r}$ parallel and transverse to the X-body axis of the SR (see Fig. 3), and $\tilde{\theta}$ is the scalar error between the end-effector's desired and current attitude in axis-angle representation. Additionally, a time minimization and a simplified collision avoidance objective have also been included in the reward, and have been found to increase the overall performance of the agent. Please refer to the full manuscript for their implementation.

5.3. Control system

The SR's control system is based on a non-linear model-based feedback linearization controller, with the goal of coupling the control systems of the base and of the manipulator, to compensate for any disturbances generated by one on the other. The resulting linearized system is controlled through two Proportional-Derivative (PD) regulators, respectively for the base and manipulator. The base is kept at the desired synchronized state with respect to the target, while the manipulator is commanded by the PPO agent. The complete control law is reported in Eq. 7.

$$\boldsymbol{\tau} = \begin{Bmatrix} \boldsymbol{\tau}_0 \\ \boldsymbol{\tau}_m \end{Bmatrix} = \boldsymbol{H} \begin{Bmatrix} PD(\tilde{\boldsymbol{q}}_0, \tilde{\dot{\boldsymbol{q}}}_0) \\ PD(\tilde{\boldsymbol{q}}_m, \tilde{\dot{\boldsymbol{q}}}_m) \end{Bmatrix} + \boldsymbol{C}\dot{\boldsymbol{q}} \quad (7)$$

where $\boldsymbol{H}$ and $\boldsymbol{C}$ are the system's GIM and CIM and $PD(\tilde{\boldsymbol{q}}, \tilde{\dot{\boldsymbol{q}}}) = K_P(\boldsymbol{q}^* - \boldsymbol{q}) + K_D(\dot{\boldsymbol{q}}^* - \dot{\boldsymbol{q}})$, with the starred quantities being the set-points. The PD's scalar gains are reported in Eq. 8.

$$\begin{aligned} K_{P,0} &= 0.4 \qquad K_{D,0} = 0.3 \\ K_{P,m} &= 2.5 \qquad K_{D,m} = 1.25 \end{aligned} \quad (8)$$

6. Baseline agent performance

The first scenario treated is that of an uncooperative target tumbling solely around its major-inertia axis, corresponding to the axis of the cylinder in Fig. 3. The angular rate of the SR $\boldsymbol{\omega}_0$ is synchronized to that of the Target $\boldsymbol{\omega}_T$ to minimize any relative motion. The CoM of the SR's base is positioned along the target's angular momentum $\boldsymbol{L}_T$, at a nominal distance of 5 m.

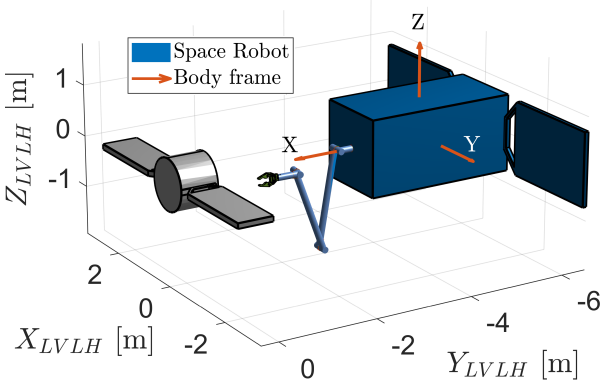

Figure 3: IOS motion synchronization scenario.

To stress the agent's abilities and avoid that it is subjected to the same scenario more than once, the initial conditions of each simulation are randomly selected within a vast pool:

- Target tumbling rate $\boldsymbol{\omega}_T \in [-3, 3]$ deg/s.
- A perturbation $\delta\phi_i \in [-15, 15]$ deg is added to each initial joint angle.
- Grasping position on the Target is selected anywhere on the cylinder's circular face, where the end-effector needs to be positioned at a 35 cm offset from this location.
- A value $\in [-25, 25]$ cm is added to the nominal distance between the two spacecraft.

The FNN hyperparameters reported in Tab. 1 are selected through a sensitivity analysis. The agent is trained for 7500 episodes of 420 s duration, and the reward evolution is shown in Fig. 4.

Table 1: Neural Network hyperparameters.

Layers	Actor neurons	Critic neurons
1st hidden	300	300
2nd hidden	300	300
3rd hidden	300	300
Learning Rate	1e-5	1e-5
Activation	*tanh*	*tanh*

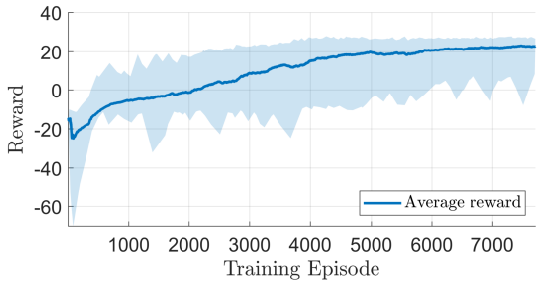

Figure 4: Baseline agent training.

The trained agent is tested through a Monte Carlo analysis of 500 testing episodes, to evaluate the performance that can be achieved. With respect to the current literature, more stringent conditions need to be satisfied to consider an episode successful: the agent must keep the end-effector within {5 cm, 5 deg} tolerances around the desired state, for at least 30 s consecutively. Fig. 5 shows that regardless of where the grasping position is located on the target's circular face, the agent can successfully achieve its objective with a 100% success rate. Additionally, the end-effector error evolution in Fig. 6 shows that once the errors enter inside the desired thresholds, they can stay within them for consecutive periods much longer than 30 s.

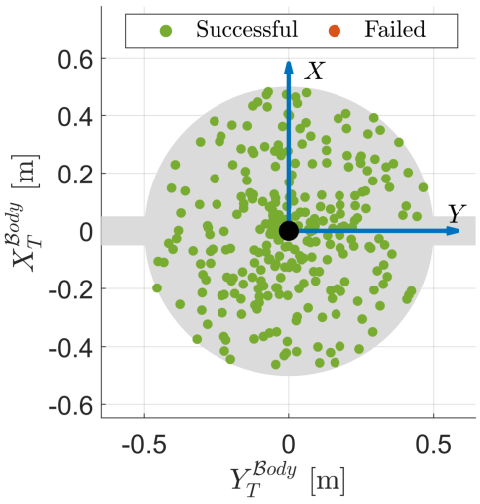

Figure 5: Correlation between randomized grasp point location on target's face and agent success.

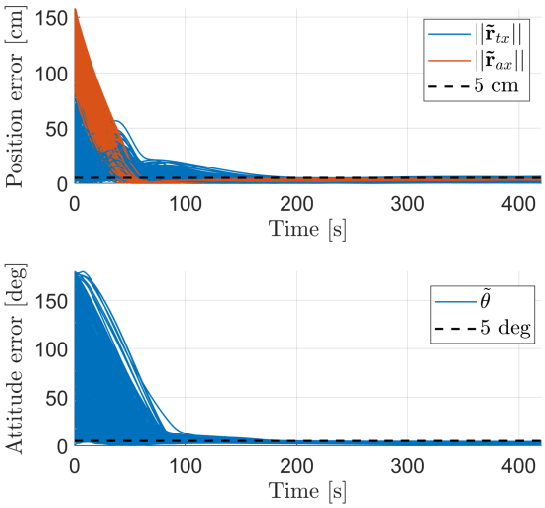

Figure 6: End-effector error evolution.

The thresholds and consecutive time used to compute episode success are heavily mission-dependent, hence a sensitivity analysis has been conducted to further analyze the agent's performance when these conditions are changed. Overall, when the thresholds are reduced to {4 cm, 4 deg} and {3.5 cm, 3.5 deg}, the success rate respectively drops to 95% and 84%, showing that the agent can still achieve its objective in the majority of cases.

6.1. Robustness & generalization

The past literature shows that using DRL for GNC problems can be beneficial in terms of system autonomy, robustness, generalization, and adaptability. To understand whether the agent possesses these characteristics, it is tested in four environments that have a large domain gap with respect to training. The results obtained in these conditions are briefly described.

Spin rate synchronization errors. In these conditions, the main difference with respect to training is that a relative rotation between the SR and the grasping point emerges, of maximum 0.5 deg/s [6]. The agent's performance drops by about 6% (see Tab. 2), showing that in the majority of cases, it can achieve its objective.

Table 2: Success rate comparison.

Baseline	Synchronization errors
100%	94%

Larger targets. The agent is trained on a target of 50 cm radius. The tests show that as long as the target's radius does not exceed 64.5 cm (29% increase), the agent successfully achieves its goal with no changes in performance.

Uncertain navigation. By adding varying levels of modeled uncertainty (offset from mean + Gaussian noise) to the observation vector entering the policy, the agent is found to be insensitive to its knowledge on the manipulator's joint angles and rates. Instead, it is sensitive to uncertainty on the end-effector's errors with respect to the desired state (i.e. on its knowledge of the real position of the grasping position). Despite this, conducting the agent's training with uncertainty is found to increase its performance.

Single-joint failures. By conducting 200×7 testing episodes with each joint locked in place (see Tab. 3), the agent is found to not generalize well to degraded manipulator configurations, and fails quite arbitrarily, which would not allow for a mission to be conducted. By having the agent encounter random joint failures during training, improvements are not seen in terms of success rate alone, but rather in terms of where the failures are located on the target's face, which end up only being encountered in specific locations. In these conditions, a 100% success rate could be achieved by selecting an opportune attitude synchronization phasing, in function of the failed joint.

Table 3: Baseline agent success.

Joint	1	2	3	4	5	6	7
Success rate	62%	9%	13%	0%	42%	58%	54%

7. Complex trajectory tracking

The final chapter of this work extends the agent's abilities to achieve a concurrent end-effector position and attitude tracking objective. The end-effector's desired time-varying state is generated by perturbing the target's initial spin rates, specifically along the intermediate inertia axis (see Eq. 9). This results in an unstable rotational equilibrium, and the target starts tumbling around all three of its body axes.

$$\boldsymbol{\omega}_T = [0, \pm\delta\omega_y, \omega_z]^\top \tag{9}$$

where $\omega_z \in \pm[1,3]$ deg/s is the randomized major-axis spin rate, and $\delta\omega_y$ is the spin rate perturbation taken between 10-15% of ω_z. The agent is trained for 16000 episodes (see Fig. 7) with the same hyperparameters found in Tab. 1; the episode randomizations introduced previously persist both during training and testing.

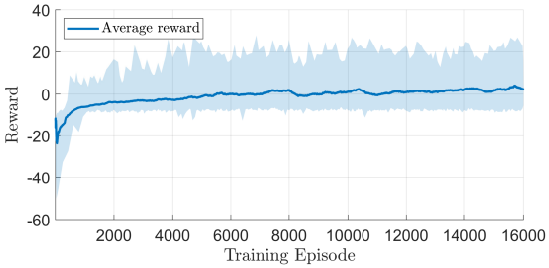

Figure 7: Trajectory tracking training.

Again, a 500 episode Monte Carlo analysis is used to evaluate the agent's performance. After training, the agent's success rate reaches 83% on the {5 cm, 5 deg} thresholds, due to the greater complexity of the tracking objective. If instead the {5 cm, 10 deg} thresholds from [7] are used, the success rate increases to 97%. In this scenario, Fig. 8 shows that the main bottleneck in performance is the agent's attitude tracking capabilities, since the attitude error has large oscillations around the 5 deg limit. A final comparison is made between the two agents from the nominal and perturbed target environments. Fig. 9 shows the distribution of the maximum consecutive time in an episode, that the

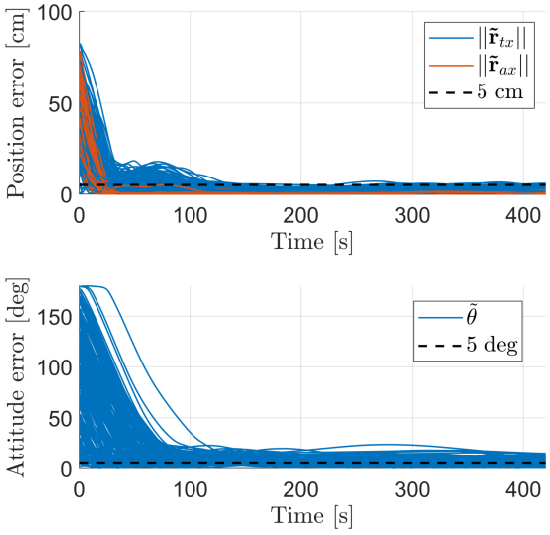

Figure 8: End-effector error evolution.

end-effector satisfies the imposed thresholds. In the unperturbed scenario, the baseline agent can keep the end-effector's errors low for long consecutive time periods, resulting in a distribution that is shifted to higher values. By instead looking at the performance of the agent trained on the perturbed target, the distribution spans over a wider range of values, specifically towards lower ones. This is due to the attitude error oscillations that occasionally lead to the loss of the desired pointing, and explain the overall lower performance achieved by the agent.

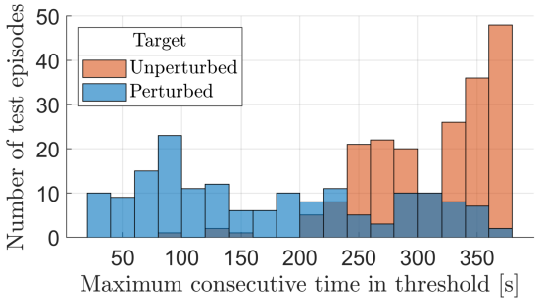

Figure 9: Agent performance in mono-axial and multi-axial target tumbling conditions.

8. Conclusions

In this thesis, DRL is successfully employed to train an autonomous agent in providing the guidance of a 7-DoF manipulator mounted on a SR, to achieve an end-effector positioning and attitude alignment objective, chosen within a

vast pool of randomized initial conditions. The baseline agent achieves a 100% success rate in keeping the end-effector's state errors below thresholds of {5 cm, 5 deg} for at least 30 s consecutively, with respect to a goal state; additionally, if the requested errors are reduced to {4 cm, 4 deg} and {3.5 cm, 3.5 deg}, the success rate respectively drops to 95% and 84%. The system adapts to errors in the angular rate synchronization between SR and target, achieving a 94% success rate, and can also achieve its objective without changes in performance, when trying to fulfill the objective on targets that are at most 29% larger with respect to training. The agent is then subjected to increasing levels of uncertainty in the navigation outputs, and is found to be most sensitive to errors in its knowledge on the end-effector's goal state. When random joint failures are introduced, the highly different manipulator dynamics make the objective almost unachievable for the agent. The addition of random actuator failures during training is provided as an effective solution to increase the agent's overall performance and robustness. Finally, the agent is trained to synchronize the end-effector with a time-varying state, generated by perturbing the target's rotational dynamics. While seeing an overall decrease in performance with respect to the nominal case, the agent achieves a 97% success rate within thresholds of {5 cm, 10 deg}.

8.1. Future developments

Overall, using a DRL-based manipulator guidance system makes studying similar approaches in the future very enticing, in view of the benefits they provide even at such a preliminary design stage. A few methods to extend the current state-of-the-art on these topics are proposed:

- The main push should be towards increasing the achievable accuracy in the end-effector's state, overcoming the current {5 cm, 5 deg} limit that provides a 100% success rate. This could be done by defining a new reward function, or by using alternative architectures to implement the policy, such as Recurrent Neural Networks.
- To exploit the advantages of both classic and DRL-based guidance architectures, a strategy worth looking into could be a hybrid guidance system, where the main guidance tasks are done in a classic framework, while tasks such as collision avoidance are achieved in real-time by an autonomous agent that perceives its surroundings.

References

[1] Y. Li, D. Li, W. Zhu, J. Sun, X. Zhang, and S. Li, "Constrained Motion Planning of 7-DOF Space Manipulator via Deep Reinforcement Learning Combined with Artificial Potential Field," *Aerospace*, vol. 9, 3 2022.

[2] Z. Huang, G. Chen, Y. Shen, R. Wang, C. Liu, and L. Zhang, "An Obstacle-Avoidance Motion Planning Method for Redundant Space Robot via Reinforcement Learning," *Actuators*, vol. 12, 2 2023.

[3] S. Wang, X. Zheng, Y. Cao, and T. Zhang, "A Multi-Target Trajectory Planning of a 6-DoF Free-Floating Space Robot via Reinforcement Learning," in *IEEE International Conference on Intelligent Robots and Systems*, pp. 3724–3730, Institute of Electrical and Electronics Engineers Inc., 2021.

[4] E. Papadopoulos, F. Aghili, O. Ma, and R. Lampariello, "Robotic Manipulation and Capture in Space: A Survey," *Frontiers in Robotics and AI*, vol. 8, 2021.

[5] M. Romano, J. Virgili-Llop, and J. V. Drew Ii, "SPART SPAcecraft Robotics Toolkit: an Open-Source Simulator for Spacecraft Robotic Arm Dynamic Modeling And Control," 2016.

[6] P. Colmenarejo, J. Branco, N. Santos, P. Serra, J. Telaar, H. Strauch, M. Fruhnert, A. M. Giordano, M. De Stefano, C. Ott, M. Reiner, D. Henry, J. Jaworski, E. Papadopoulos, G. Visentin, F. Ankersen, and J. Gil-Fernandez, "Methods and outcomes of the COMRADE project - Design of robust Combined control for robotic spacecraft and manipulator in servicing missions," in *69th International Astronautical Congress (IAC)*, pp. 1–5, 2018.

[7] L. Capra and M. Lavagna, "Adaptive Space Robot Motion Synchronization Towards Tumbling Uncooperative Target Grasping," in *74th International Astronautical Congress*, (Baku, Azerbaijan), 10 2023.