Journal of Guidance, Control, and Dynamics
Vol. 45, No. 11, November 2022

Engineering Notes

Laboratory Experimentation of Spacecraft Robotic Capture Using Deep-Reinforcement-Learning–Based Guidance

Kirk Hovell* and Steve Ulrich†
Carleton University, Ottawa, Ontario K1S 5B6, Canada

https://doi.org/10.2514/1.G006656

Nomenclature

$\mathcal{A}$	=	action space
$\boldsymbol{a}$	=	action
a, b	=	manipulator length properties, m
B	=	number of value distribution bins
$\boldsymbol{C}$	=	Coriolis matrix
$\boldsymbol{d}_t$	=	docking port position on target, m
$\mathbb{E}$	=	expectation
h	=	angular momentum, $(\mathrm{kg}\cdot\mathrm{m}^2)/\mathrm{s}$
I	=	moment of inertia, $\mathrm{kg}\cdot\mathrm{m}^2$
J	=	total expected rewards
K	=	number of actors
L	=	loss function
$\boldsymbol{l}$	=	linear momentum, $(\mathrm{kg}\cdot\mathrm{m})/\mathrm{s}$
$\boldsymbol{M}$	=	mass matrix
M	=	mini-batch size
m	=	mass, kg
N	=	N-step return length
$\mathcal{N}$	=	normal distribution
$\boldsymbol{p}$	=	position, m
R	=	replay buffer size
r	=	reward
$\boldsymbol{u}$	=	control effort
$\boldsymbol{v}$	=	velocity, m/s
$\mathcal{X}$	=	state space
$\boldsymbol{x}$	=	total state; specific state, with subscript c or t
x	=	position in the x direction, m
Y	=	target value distribution
y	=	position in the y direction, m
$Z_{\boldsymbol{\phi}}$	=	value neural network with parameters $\boldsymbol{\phi}$
α	=	policy network learning rate
β	=	value network learning rate
γ	=	discount factor for future rewards
ϵ	=	weight-smoothing parameter
$\pi_{\boldsymbol{\theta}}$	=	policy neural network with parameters $\boldsymbol{\theta}$
$\boldsymbol{\phi}'$ or $\boldsymbol{\theta}'$	=	exponentially smoothed versions of $\boldsymbol{\phi}$ or $\boldsymbol{\theta}$
ϕ, q	=	angle, deg
σ	=	exploration noise standard deviation
ω	=	angular rate, rad/s

Subscripts

c	=	denoting chaser
cm	=	denoting chaser–manipulator combination
cmt	=	denoting chaser–manipulator–target combination
e	=	end-effector
n	=	time step number
t	=	denoting target

Superscripts

d	=	desired
1, 2, 3	=	denoting the manipulator joints

I. Introduction

SINCE the first demonstrations of on-orbit servicing using a robotic manipulator with ETS-VII in 1997 [1] and Orbital Express in 2007 [2], robotic manipulator research has accelerated in hopes of enabling autonomous on-orbit servicing and space debris removal. For a chaser spacecraft to service a target, it must perform two tasks: 1) capture the target with its end-effector, and then 2) stabilize the spinning target postcapture. The first task, target capture, is typically accomplished using one of two approaches: i) the chaser spacecraft is brought to rest near the target, after which the manipulator alone executes the capture, typically in a way that minimizes the disturbances on the unactuated spacecraft base [3–6], or ii) the chaser spacecraft approaches the target and captures it in one motion (i.e., both the spacecraft base and manipulator are actuated). The latter approach is more flexible than the former, since it can handle time-varying keep-out zones, and has therefore been extensively studied with optimization methods [7–11], model predictive control methods [12], and sample-based methods [13]. The second task, stabilizing the spinning target postcapture, has been separately studied through minimum-time formulations [7,14] and various advanced control approaches [15–18].

Recently, Virgili-Llop and Romano [19] presented a method that performs both the capture and stabilization simultaneously in a planar environment. It uses an optimization-based approach to generate a real-time guidance trajectory that satisfies a number of terminal constraints such that the capture and stabilization are considered simultaneously in the optimization. This technique leads the chaser spacecraft to capture the target with some momentum of its own, effectively canceling the target's angular momentum upon capture. The authors experimentally validated their work in a planar facility while demonstrating that the approach can be executed on low-powered hardware. While this optimization-based work has shown great promise, it is accompanied by prescribed capture times and orientations, constant chaser initial conditions, and hard-coded manipulator motion during the final seconds before capture—all required for the optimization to be tractable. In addition, the approach requires significant engineering effort to design and was less effective at capturing moderately spinning targets greater than 7 deg/s [19].

This paper builds upon this previous work by a) relaxing a number of assumptions, while b) reducing the engineering effort associated with the approach (by applying machine learning to the problem), and c) improving performance by enabling the capture of high spin-rate targets. Specifically, this paper uses deep reinforcement learning (DRL) to *learn* a guidance strategy for simultaneous capture and stabilization, rather than *designing* one by hand. DRL, through trial and error, discovers the best way to solve a given problem by calculating an *action* to take from a given *observation* of the state; the

Received 29 December 2021; revision received 17 June 2022; accepted for publication 15 August 2022; published online 7 September 2022.

*Ph.D. Candidate, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive. Student Member AIAA.

†Associate Professor, Department of Mechanical and Aerospace Engineering, 1125 Colonel By Drive. Senior Member AIAA.

notion of success comes from a human-designed scalar reward function. Through analyzing trial-and-error data, DRL determines which actions maximize the rewards received—the reward function drives the learned behavior. DRL has been cited as a major influence in the future of aerospace engineering [20]. To date, it has been used with spacecraft orbit determination [21], uncrewed aerial vehicle wildfire detection [22], planetary landing [23,24], and spacecraft proximity operations [25–27] with experimental results [28]. DRL has recently been used for a dual-arm stationary spacecraft to move its two end-effectors near a target [29]. Similar dual-arm work, with a free-floating spacecraft base and additional dynamic constraints has, been presented in [30] with preliminary results (capture is considered successful when both end-effectors are within 0.2 m of the docking ports). A dual-arm truss-assembling robot used DRL to compensate for measurement errors and obstacles in experiment [31]. Experimentation has also been performed using machine-learning-based computer vision for manipulator tasks [32,33]. Unless otherwise reported, all previous work has been considered in simulation only.

This paper uses DRL to guide a manipulator-equipped chaser spacecraft to capture and simultaneously stabilize a spinning target from arbitrary initial conditions despite position, velocity, and acceleration constraints. The learned real-time closed-loop guidance technique is trained in simulation and is evaluated in planar hardware experiments at Carleton University's planar Spacecraft Proximity Operations Testbed. The novel contributions of this work are as follows:

1) An improvement in the state-of-the-art of manipulator-based spacecraft capture and simultaneous stabilization

2) The first experimental demonstration, to the best of the authors' knowledge, of using artificial intelligence techniques to perform manipulator-based capture of a spinning target

Compared to previous optimization-based work accomplishing the same task [19], this DRL-based approach allows for the following improvements to be made: 1) the chaser and manipulator initial positions are randomized; 2) the chaser and manipulator configurations at capture are not prescribed; 3) the manipulator motion is not hard-coded before capture; and 4) no prescribed maneuver time is set (i.e., the chaser is allowed to learn the best time to capture the target).

This paper is organized as follows: Sec. II presents the spacecraft capture task and how it is solved using DRL, Sec. III presents and discusses the simulated and experimental results, and Sec. IV concludes this paper.

II. Methods

This section describes the task, system dynamics, and how DRL is used as guidance to solve the capture and stabilization problem.

A. Problem Statement

Two spacecraft exist in a planar environment: a chaser spacecraft is actively controlled and has a three-link robotic manipulator; a target spacecraft is passive, spinning at a constant angular velocity, and has a docking port. Both spacecraft are shown in Fig. 1. The goal of this work is to 1) have the chaser spacecraft learn how to capture the spinning target with its robotic manipulator, while 2) simultaneously bring the target to rest upon being captured. It is assumed that the manipulator end-effector engages rigidly with the target's docking port immediately upon contact. The planar three-link manipulator-equipped chaser has dynamics modeled according to

$$\boldsymbol{M}(\boldsymbol{x}_c)\ddot{\boldsymbol{x}}_c + \boldsymbol{C}(\boldsymbol{x}_c, \dot{\boldsymbol{x}}_c)\dot{\boldsymbol{x}}_c = \boldsymbol{u}_c \tag{1}$$

where $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{C}$ is the Coriolis matrix, $\boldsymbol{x}_c$ is the chaser state, and $\boldsymbol{u}_c$ is the control effort. The matrices are as defined in [6] with physical parameters, measured from the Spacecraft Proximity Operations Testbed, defined in Table 1. As is common in short timescale and separation distance proximity operations research, orbital effects are ignored [6,11,19,31,34–41].

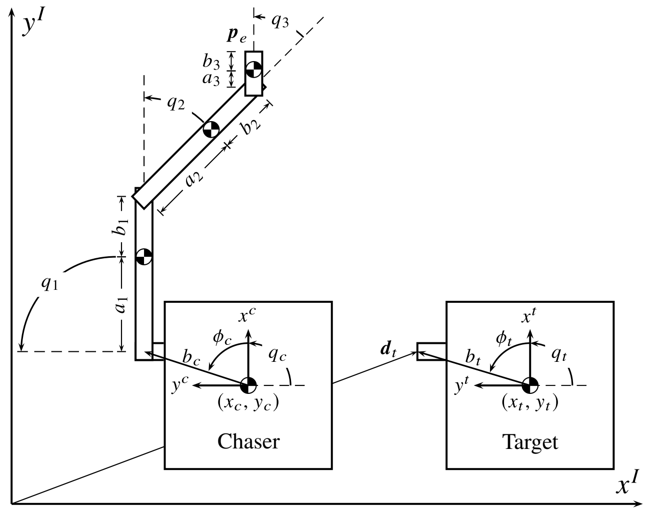


Fig. 1 Chaser and target reference frame and state notation.

Table 1 Dynamics parameters

Body	Body identifier	ϕ_i, deg	a_i, m	b_i, m	m_i, kg	I_i, kg · m^2
Chaser	C	74.31	——	0.2515	11.211	2.022×10^{-1}
Link #1	1	——	0.1968	0.1077	0.3450	3.704×10^{-3}
Link #2	2	——	0.1982	0.1063	0.3350	3.506×10^{-3}
Link #3	3	——	0.0621	0.0252	0.1110	1.060×10^{-4}
Target	t	72.00	——	0.2406	12.039	2.257×10^{-1}

B. Deep-Reinforcement-Learning–Based Guidance

DRL is used to learn the behavior required for the chaser to capture and stabilize the target. A Markov decision process (MDP) describes a discrete-time sequence of events to which DRL can be applied. For a state $\boldsymbol{x}_n \in \mathcal{X}$, at time step n, upon which a chosen action $\boldsymbol{a}_n \in \mathcal{A}$ is applied, a new state $\boldsymbol{x}_{n+1}$ is returned according to a state transition function $P_{\boldsymbol{a}}(\boldsymbol{x}_n, \boldsymbol{x}_{n+1})$, which may be stochastic, along with a corresponding scalar reward $r_n(\boldsymbol{x}_n, \boldsymbol{a}_n)$. The MDP must satisfy the Markov property, which requires the process be memoryless; i.e., $\boldsymbol{x}$ must fully describe the system. DRL performs trial and error on an MDP to learn a policy $\pi_{\boldsymbol{\theta}}$ that maps states $\boldsymbol{x}$ to actions $\boldsymbol{a}$ that maximize the rewards r accumulated over time. In contrast to optimal control, DRL only has access to the dynamics through sampling. The pursuit of rewards by the learning algorithm allows for complex behaviors to emerge according to a comparatively simple reward function; the designer does not need to know *how* to solve the problem, only *when* it is solved. This paper uses the DRL algorithm named D4PG [42] because it allows for continuous states and actions, yields deterministic behavior, can be trained across many CPUs, and has excellent performance. D4PG is an actor–critic algorithm where the critic neural network is trained using supervised learning to minimize

$$L(\boldsymbol{\phi}) = \mathbb{E}\{-Y \log(Z_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{a}))\} \tag{2}$$

where Z_ϕ is the critic with parameters ϕ, $\boldsymbol{Y}$ are the target critic values, $\mathbb{E}$ denotes the expectation, and $L(\phi)$ is the loss function to be minimized using stochastic gradient descent. The policy is trained through

$$\nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) = \mathbb{E}[\nabla_{\boldsymbol{\theta}} \pi_{\boldsymbol{\theta}}(\boldsymbol{x}) \mathbb{E}[\nabla_{\boldsymbol{a}} Z_{\boldsymbol{\phi}}(\boldsymbol{x}, \boldsymbol{a})]|_{\boldsymbol{a}=\pi_{\boldsymbol{\theta}}(\boldsymbol{x})}] \tag{3}$$

where $\nabla_{\boldsymbol{\theta}}$ denotes the gradient with respect to the policy's neural network parameters $\boldsymbol{\theta}$, $J(\boldsymbol{\theta})$ denotes the average policy performance, and all other variables have been previously defined. Algorithm and training details can be found in [28] or in the included code.[‡]

[‡]https://github.com/Carleton-SRCL/JGCD_2021.

To apply DRL to the capture and stabilization scenario considered in this paper, the DRL-based guidance technique developed by the authors [28,43] is herein used. DRL is used to learn a guidance strategy (one that issues a desired acceleration signal), while control (to track that desired acceleration signal) is performed using traditional approaches. The motivation for this architecture is twofold: 1) it brings the high-level problem solving ability of DRL into the most appropriate segment for robotics (guidance/path planning), while 2) excluding the policy from overfitting a controller to the simulated dynamics (which are not identical to reality). A block-scheme diagram of the guidance and control system is shown in Fig. 2, where $\ddot{\boldsymbol{x}}^d_{c,n}$ is the desired chaser acceleration signal and $\boldsymbol{u}_n$ is the control effort.

The input to the DRL policy is

$$\boldsymbol{x} = [\boldsymbol{x}_c \quad \dot{\boldsymbol{x}}_c \quad \dot{q}_t \quad x_t - x_c \quad y_t - y_c \quad q_t - q_c]^T \in \mathbb{R}^{16} \tag{4}$$

where $\boldsymbol{x}_c = [x_c \quad y_c \quad q_c \quad q_1 \quad q_2 \quad q_3]$, with all elements defined in Fig. 1.

C. Reward Function

The reward function incentivizes the desired behavior. A sparse reward function is used, where a reward of zero is given at all time steps except the following: 1) a reward is given for capturing the target (though this reward is reduced proportional to postcapture angular momentum and end-effector angle and velocity errors at the moment of capture); 2) a reward is given for bringing the end-effector close to the docking port to encourage docking; 3) a penalty is given if the manipulator joints reach their limits; and 4) a penalty is given for the chaser traveling too far from the target.

More concretely, the reward function is

$$r_n = \begin{cases} 0, & \text{otherwise} \\ 100 - 50|\sin(q_c + q_1 + q_2 + q_3 - q_t - \pi/2)| - \\ 50\|\dot{\boldsymbol{p}}_e - \dot{\boldsymbol{d}}_t\| - 50|\dot{q}_c + \dot{q}_1 + \dot{q}_2 + \dot{q}_3 - \dot{q}_t| - 25|h_{\text{cmt}}|, & \text{if } \|\boldsymbol{d}_t - \boldsymbol{p}_e\| \le 0.04 \text{ m(awarded once)} \\ 25, & \text{if } \|\boldsymbol{d}_t - \boldsymbol{p}_e\| \le 0.1 \text{ m(awarded once)} \\ -5, & \text{if } |q_1| \text{ or } |q_2| \text{ or } |q_3| \ge \pi/2 \\ -100, & \text{if } x_c \notin [0, 3.5] \vee y_c \notin [0, 2.4] \end{cases} \tag{5}$$

where $\boldsymbol{p}_e$ is the end-effector position, $\boldsymbol{d}_t$ is the target's docking port location, and h_{cmt} is the total postcapture angular momentum (measured about the postcapture center of mass, $\boldsymbol{p}_{\text{cmt}}$), calculated through

$$h_{\text{cmt}} = h_{\text{cm}} + \boldsymbol{S}(\boldsymbol{p}_{\text{cm}} - \boldsymbol{p}_{\text{cmt}})^{\times}\boldsymbol{l}_{\text{cm}} + h_t + \boldsymbol{S}(\boldsymbol{p}_t - \boldsymbol{p}_{\text{cmt}})^{\times}\boldsymbol{l}_t \tag{6}$$

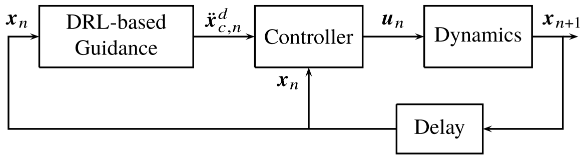


Fig. 2 DRL-based guidance strategy.

where h_{cm}, $\boldsymbol{l}_{\text{cm}}$, and $\boldsymbol{p}_{\text{cm}}$ are the angular momentum (measured about $\boldsymbol{p}_{\text{cm}}$), the linear momentum, and the position of the chaser–manipulator system, respectively, and h_t, $\boldsymbol{l}_t$, and $\boldsymbol{p}_t$ are the angular momentum (measured about $\boldsymbol{p}_t$), the linear momentum, and the position of the target, respectively. The center of mass of the chaser–manipulator–target system, $\boldsymbol{p}_{\text{cmt}}$, is calculated as the mass-weighted average of all the component positions. Finally, $\boldsymbol{S} = [0 \quad 0 \quad 1]$ to select the angular momentum component relevant to the planar motion. The skew-symmetric matrix is

$$\boldsymbol{p}^{\times} = \begin{bmatrix} 0 & -p_z & p_y \\ p_z & 0 & -p_x \\ -p_y & p_x & 0 \end{bmatrix} \tag{7}$$

The episode is terminated early, with a reward of zero, if 1) the chaser collides with the target in any fashion, or 2) the maximum time limit is reached. In seeking to maximize rewards, using Eqs. (2) and (3), the DRL-based guidance system indirectly learns to master the target capture and simultaneous stabilization task. Learning algorithm parameters are identical to the authors' previous work [28], with the exception of the value bounds being over the expected reward interval $[-100, 125]$. The theoretical maximum expected reward of 125 represents the proximity reward of $+25$ plus a perfect docking of $+100$. Although the value bound of 125 will never be reached in practice, due to future rewards being discounted by the factor γ, it is indeed a reasonable upper bound.

It should be noted that the reward function could be adjusted to yield any desired behavior. For a faster capture, time penalties could be introduced; for fuel-optimal behavior, acceleration penalties. As shown in the authors' previous work [28], the reward function may be easily modified to accommodate stationary or dynamic obstacles through penalizing chaser proximity to such obstacles.

III. Results

This section presents the training of the DRL-based guidance policy in simulation and the subsequent laboratory experimental results.

A. Training in Simulation

The capture task is trained entirely in simulation before being transferred to the spacecraft experimental laboratory. Dynamic parameters listed in Table 1 are used in the numerical simulations, along with the position, velocity, and acceleration constraints listed in Table 2. The chaser, manipulator, and target initial conditions are

Table 2 Position, velocity, and acceleration limits

Parameter	x_c	y_c	q_c	q_1	q_2	q_3
Position	[0, 3.5] m	[0, 2.4] m	No limit	$\pm\pi/2$ rad	$\pm\pi/2$ rad	$\pm\pi/2$ rad
Velocity	±0.1 m/s	±0.1 m/s	$\pm\pi/12$ rad/s	$\pm\pi/6$ rad/s	$\pm\pi/6$ rad/s	$\pm\pi/6$ rad/s
Acceleration	±0.02 m/s^2	±0.02 m/s^2	±0.05 rad/s^2	±0.1 rad/s^2	±0.1 rad/s^2	±0.1 rad/s^2

Table 3 Initial conditions and their uniform randomization

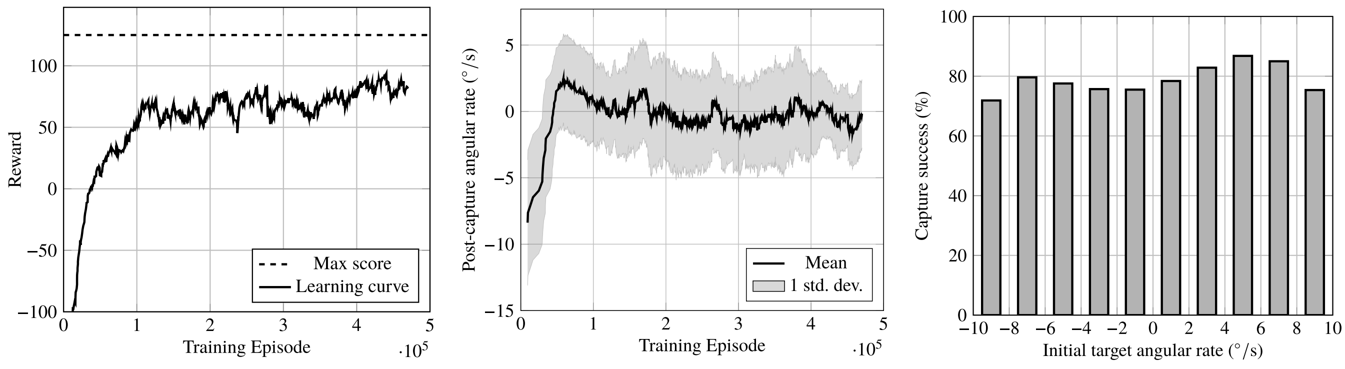
Parameter	x_c, m	y_c, m	q_c, rad	q_1, rad	q_2, rad	q_3, rad	x_t, m	y_t, m	q_t, rad
Initial position	1.2	1.2	0	0	0	0	2.3	1.2	0
Randomization	±0.05	±0.05	$\pm\pi$	$\pm\pi/2$	$\pm\pi/2$	$\pm\pi/2$	±0.05	±0.05	$\pm\pi$

uniformly randomized at the beginning of each episode according to Table 3. A target-centered reference frame is used such that the chaser's velocity is reported relative to the target; the chaser starts each episode with zero relative velocity. The target's angular velocity is uniformly randomized across the range $\dot{q}_t = \pm 10$ deg /s. In other words, the target's angular velocity is randomized in direction and magnitude between episodes, but is constant within each episode. It is assumed that the full system state is known (as did [19] among others [6,10,11,28,34–36,40,41]). The chaser and its manipulator are actively controlled to track the desired acceleration signal calculated by the policy, $\ddot{\boldsymbol{x}}_c^d$, using the following feedforward open-loop controller:

$$\boldsymbol{u}_c = \boldsymbol{M}(\boldsymbol{x}_c^d)\ddot{\boldsymbol{x}}_c^d + \boldsymbol{C}(\boldsymbol{x}_c^d, \dot{\boldsymbol{x}}_c^d)\dot{\boldsymbol{x}}_c^d \tag{8}$$

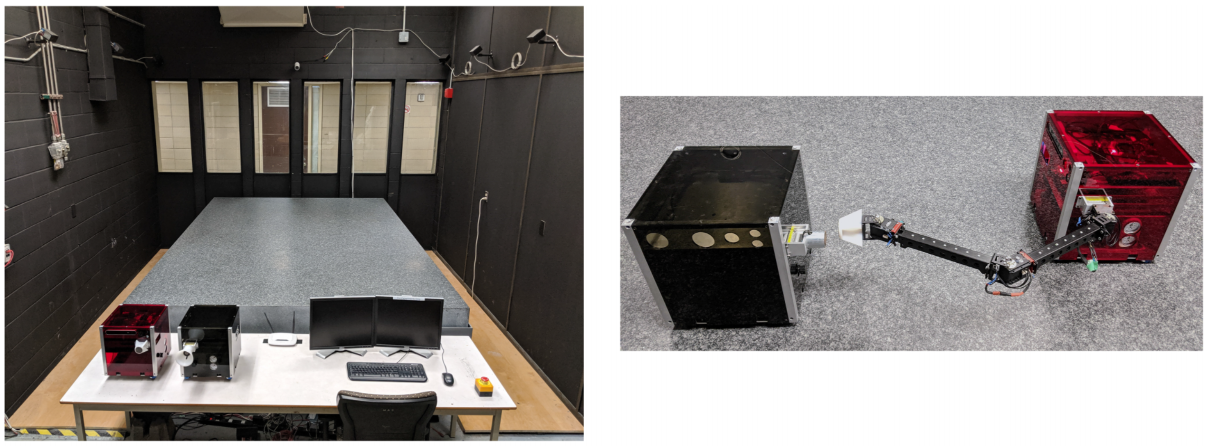
Training is performed on Carleton University's Research Computing and Development Cloud (32-Intel Xenon x86 CPUs) over a period of 15 days.

The learning curve in Fig. 3a, which is exponentially smoothed for clarity with a factor of 0.95, shows the total rewards accumulated over each episode as a function of the number of training episodes. The accumulated reward in each episode increases, on average, as training progresses. The policy's initially random behavior leads to poor initial performance; penalties for exceeding the table boundaries are often encountered along with manipulator joint limit penalties. However, better performance, as judged by the accumulated rewards received, is quickly learned within the first $\sim 10^5$ episodes. As training continues, the maximum average reward received after 4.5×10^5 episodes is 90, which is near the theoretical maximum rewards of 125 obtainable for this task. Figure 3b shows the exponentially smoothed postcapture angular velocity of the combined chaser–manipulator–target system along with one standard deviation. The postcapture angular velocity is only reported when successful capture occurs, and is why the curve does not start from episode 0. During training, the average postcapture angular velocity of the combined system approaches zero and its variance decreases (final values are −0.17°/s average residual angular velocity with a standard deviation of 2.56°/s). Figure 3c shows the capture success rate of the final 10,000 training episodes as a function of the target angular velocity. A capture is deemed successful if >25 rewards are received on the episode, which is only possible when a capture occurs [see Eq. (5)]. A near-constant success rate is observed across all target angular rates, in contrast to the decreasing success with increasing angular rate reported in [19]. These three plots demonstrate that, starting from purely random behavior, the



a) Learning curve b) Post-capture angular velocity c) Capture success rate

Fig. 3 Training progression in simulation.

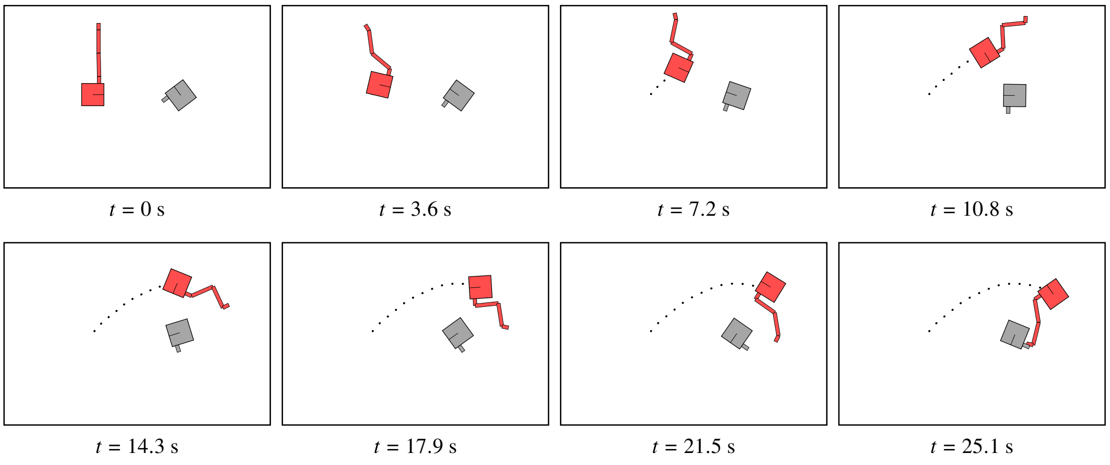


a) An overview of the SPOT facility b) SPOT platforms with a robotic manipulator (right) and docking port (left)

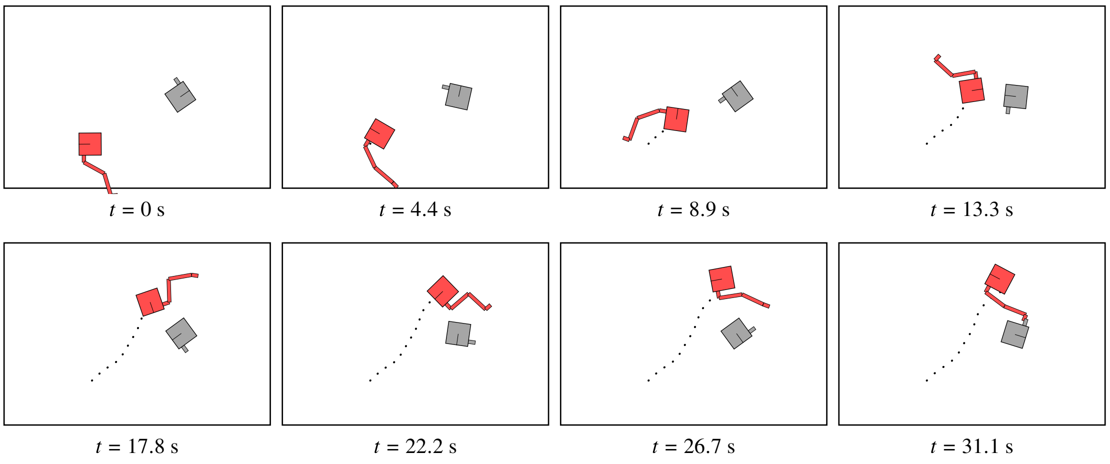
Fig. 4 The Spacecraft Proximity Operations Testbed.

Table 4 Experiment initial conditions

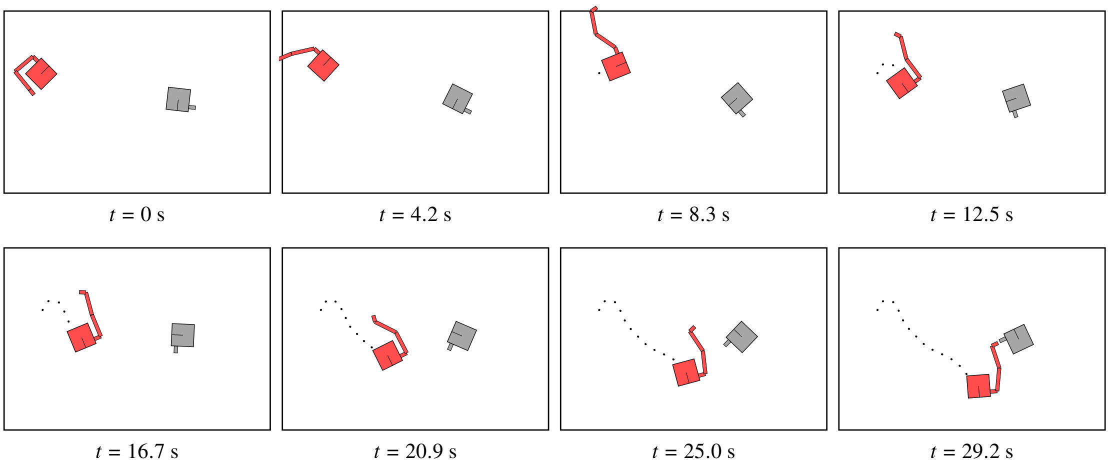
Initial condition	x_c, m	y_c, m	q_c, °	q_1, °	q_2, °	q_3, °	x_t, m	y_t, m	q_t, °	$\dot{q}_t$, °
1	1.2	1.2	0	0	0	0	2.3	1.2	125	5
2	1.2	0.6	180	60	−45	80	2.3	1.2	40	10
3	0.5	1.6	45	90	90	0	2.3	1.2	−100	−5



a) Initial condition 1



b) Initial condition 2



c) Initial condition 3

Fig. 5 Snapshots of three experimental trajectories.

DRL-based guidance algorithm has successfully learned to maneuver the chaser and its manipulator, through calculating desired acceleration signals, to capture the target spacecraft in a way that the combined system is stabilized.

B. Experimental Results

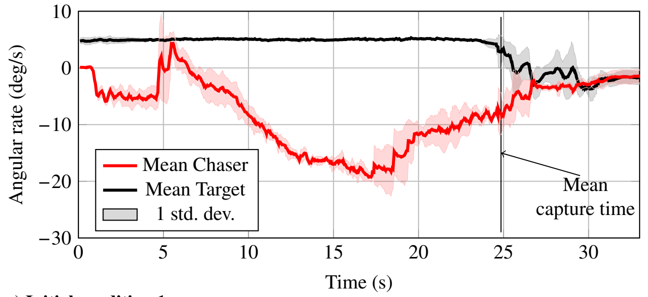
Experiments are conducted at Carleton University's Spacecraft Proximity Operations Testbed (SPOT). SPOT consists of two air-bearing spacecraft platforms operating in a three-degree-of-freedom and near-friction-free environment in close proximity on a 2.4 m × 3.5 m granite surface. Each spacecraft platform has eight miniature air thrusters, with a 1.76 mN · s minimum impulse bit, which provide full planar control authority. Additional facility details are presented in [28]. Figure 4a shows the laboratory facility and Fig. 4b shows the two platforms: one equipped with a robotic manipulator (the chaser) and the other with a magnetic docking port (the target). The target's electromagnetic docking port creates a rigid connection with the chaser's end-effector when the two are brought within approximately 1 cm of each other.

The final version of the trained-in-simulation guidance policy is exported for use in the SPOT facility; no additional training or fine-tuning is performed during the experiment. Both platforms maneuver to their initial conditions. The chaser then performs inference on the trained guidance policy and uses its on-board feedforward controller to track the guided acceleration signal it receives from the policy.

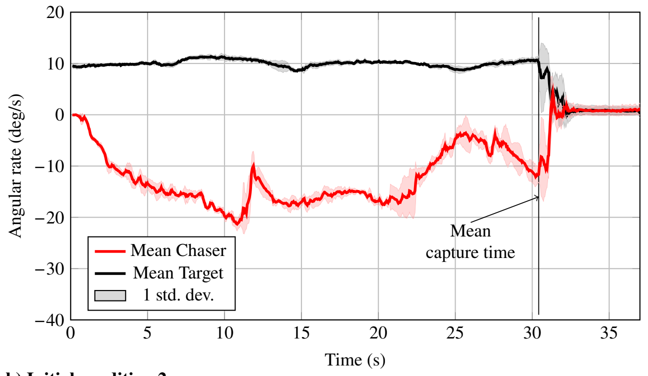
Initial conditions for three experimental scenarios are shown in Table 4. Each experiment is repeated five times, and snapshots of a select experiment for each initial condition are shown in Fig. 5. The average angular rates across all five trials are shown in Fig. 6. Post-capture angular velocities for the three initial conditions are $-1.5^\circ/\mathrm{s}$, $0.5^\circ/\mathrm{s}$, and $1.0^\circ/\mathrm{s}$, respectively. The target was successfully captured at a $10^\circ/\mathrm{s}$, higher than attempted in previous work, and with fewer assumptions and reduced engineering effort than previous work [19].

Due to experimental limitations that are not accounted for in simulation (control allocation problems and chaser–target thruster plume interactions), initial conditions have to be chosen that minimize these effects; the chosen initial conditions are listed in Table 4. The control allocation strategy is a function that converts desired forces and torques into thruster firings for the eight miniature air thrusters on the SPOT platforms; these thruster firings must be accurate in order for the guided accelerations to be properly tracked. It was discovered that the thruster air pressure decreases as a function of the number of thrusters firing simultaneously, leading to control allocation problems and the desired accelerations not being accurately tracked. In addition, the chaser's thruster plumes often dramatically affect the motion of the target immediately before capture, causing capture to fail. Initial conditions that lead to these situations are excluded, as these experiment-related problems are deemed outside the scope of this work. However, plume interactions were still present in some successful experiments, as observed in Fig. 6c around 25 s. There, the target's angular rate slows over 5 s just before capture due to the chaser's thruster plumes interacting with the target's structure.

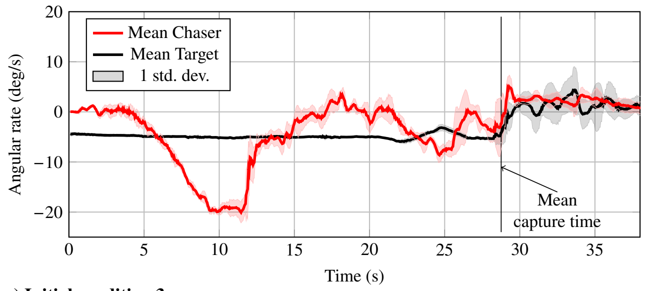
Two perturbation experiments are performed (using initial condition 2 in Table 4), where 1) the chaser is pushed off course, and 2) the target is stopped from spinning midway through the experiment. Both perturbations are applied by manually interfering with the platforms. In Fig. 7a, the chaser is abruptly moved to a new location and orientation midway through the experiment. Because the guidance policy calculates desired acceleration signals in real-time according to the current state of the chaser, manipulator, and target, in Eq. (4), it immediately adapts to the perturbation and adjusts its approach to successfully capture the target. Similarly, in Fig. 7b, the target is perturbed to a new attitude and its angular rate is brought to zero midexperiment. The guidance policy again adapts in real-time to the new scenario and captures the target regardless. These two perturbation experiments highlight the real-time nature of the DRL-based guidance technique and its ability to handle perturbations not seen during training.



a) Initial condition 1



b) Initial condition 2



c) Initial condition 3

Fig. 6 Angular rate over time for three sets of initial conditions, each with five trials.

C. Simulation-to-Reality, Safety, and Limitations

Successful manipulator-based capture was achieved despite a significant number of discrepancies between the experimental facility and the simulated environment within which the policy was trained. Namely, the simulated environment dynamics did not model friction (between the air bearings and the table, nor at the manipulator joints), air resistance, signal noise, system delays, center of mass offsets, discrete thrusters and the thrust actuation strategy, thruster plume interaction, or the table slope. The learned system transfers well from simulation to reality because DRL was implemented to exclusively learn guidance, which is less sensitive to dynamic discrepancies than control. A feedforward controller ensured proper guidance tracking regardless of modeling errors.

System safety was considered in the crafting of the reward function and early ending conditions to avoid chaser–target collisions. Safety is an important topic whenever learning-based methods are used, as traditional stability proofs are not yet available for neural networks.

Learning-based methods have limitations. Designing the reward function to elicit the desired behavior can be time-consuming, the offline training process can be compute-intensive, and the resulting behaviors may be unintuitive to humans. However, the advantages of learning-based techniques (ease of use, state-of-the-art behavior, real-time performance) demand their consideration for aerospace applications.

A video summarizing the simulations and experiments can be found in the Supplemental Material (or online at https://youtu.be/Ft6WKnHFhcY). All code used to train and evaluate the reinforcement

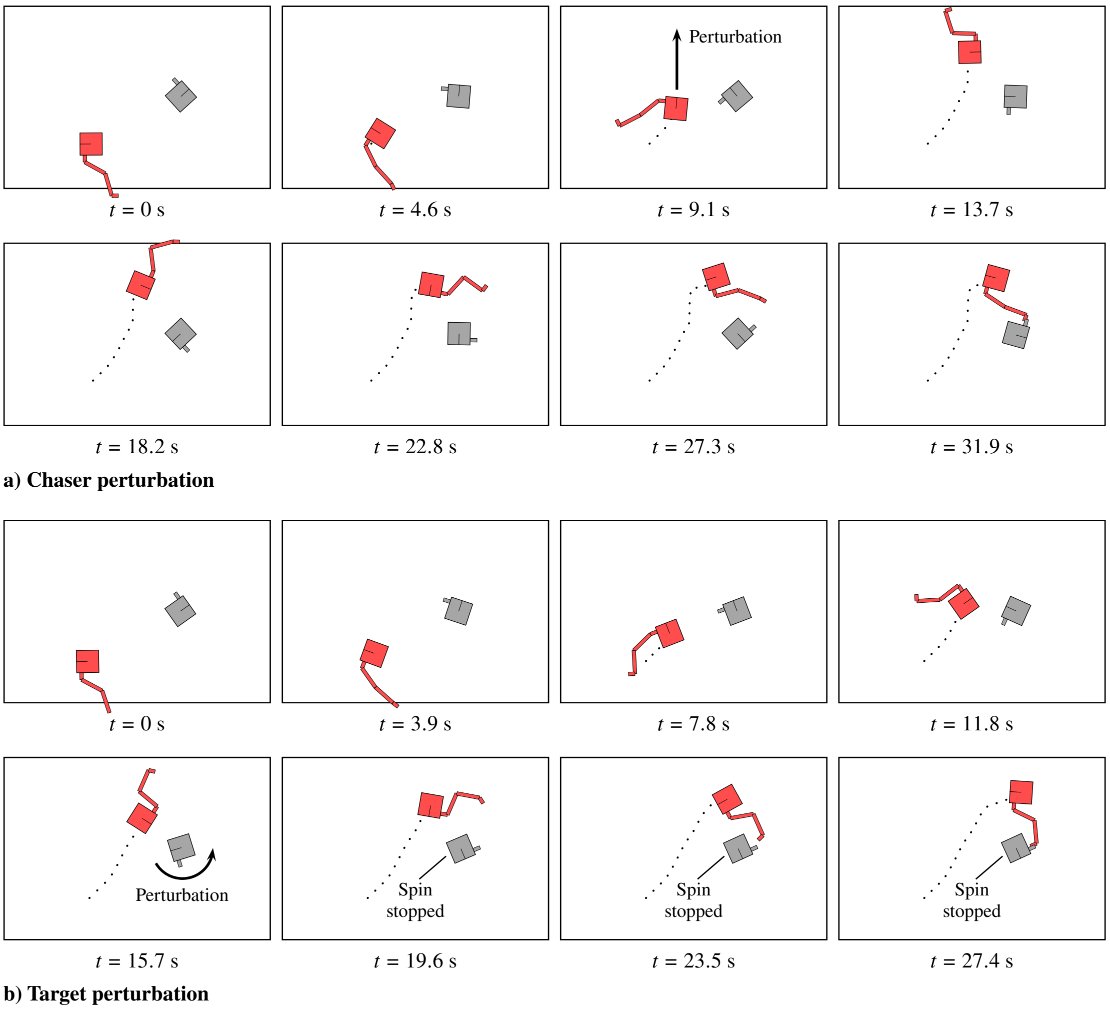


Fig. 7 Snapshots of two perturbation experiments.

learning algorithm on the capture task can be found at https://github.com/Carleton-SRCL/JGCD_2021.

IV. Conclusions

Using a robotic manipulator to capture and stabilize an uncooperative target is a promising solution to the growing space debris problem and to enable on-orbit satellite servicing. The motion required for a manipulator-equipped spacecraft to capture and simultaneously stabilize a spinning target is complex, leading traditional approaches to require significant engineering effort and assumptions to solve. DRL, on the other hand, allows for complex behaviors to be learned, rather than designed, according to a simple reward function. DRL was used to learn a guidance strategy, and a conventional controller tracked its desired acceleration signals. The guidance system was successfully trained in simulation despite position, velocity, and acceleration constraints, randomized initial conditions, and target spin rates from $-10^\circ/\mathrm{s}$ to $10^\circ/\mathrm{s}$; the chaser properly learned to capture the target and simultaneously stabilize it. The trained guidance policy was then transferred to the Spacecraft Proximity Operations Testbed facility at Carleton University, where the learned behavior was successfully executed on a manipulator-equipped spacecraft hardware platform that was able to capture and simultaneously stabilize an uncooperative spinning target despite being subjected to initial conditions and perturbations not seen during training. This work improves the state-of-the-art in manipulator-based capture and stabilization of spacecraft and represents the first, to the best of the authors' knowledge, experimental demonstration of using artificial intelligence techniques to enable the capture of a spinning spacecraft.

Future work should compare learned DRL behaviors to state-of-the-art hand-crafted methods such as model predictive control on the basis of performance, fuel consumption, and real-time compute requirements. It should also investigate increasing the success rate of these preliminary results higher to render the technique closer to production quality.

Acknowledgments

This research was financially supported in part by the Natural Sciences and Engineering Research Council of Canada under the Postgraduate Scholarship-Doctoral PGSD3-503919-2017 and the Ontario Graduate Scholarship awards. This research was enabled in part by support provided by Research Computing Services (https://carleton.ca/rcs) at Carleton University; specific thanks to Ryan Taylor for his help. Computations were also made on the supercomputer Béluga from Université du Québec, managed by Calcul Québec and Compute Canada. The operation of this supercomputer is funded by the Canada Foundation for Innovation, Minisètre de l'Économie et de l'Innovation du Québec and le Fonds de recherche du Québec–Nature et technologies.

References

[1] Yoshida, K., "Achievements in Space Robotics," *IEEE Robotics & Automation Magazine*, Vol. 16, No. 4, 2009, pp. 20–28. https://doi.org/10.1109/MRA.2009.934818
[2] Friend, R. B., "Orbital Express Program Summary and Mission Overview," *Sensors and Systems for Space Applications II*, Vol. 6958, SPIE, Bellingham, WA, 2008, Paper 695803. https://doi.org/10.1117/12.783792

[3] Torres, M. A., and Dubowsky, S., "Minimizing Spacecraft Attitude Disturbances in Space Manipulator Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 1010–1017.
https://doi.org/10.2514/3.20936

[4] Yoshida, K., Dimitrov, D., and Nakanishi, H., "On the Capture of Tumbling Satellite by a Space Robot," *IEEE International Conference on Intelligent Robots and Systems*, Inst. of Electrical and Electronics Engineers, New York, 2006, pp. 4127–4132.
https://doi.org/10.1109/IROS.2006.281900

[5] Flores-Abad, A., Wei, Z., Ma, O., and Pham, K. D., "Optimal Control of Space Robots for Capturing a Tumbling Object with Uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 2014–2017.
https://doi.org/10.2514/1.G000003

[6] Crain, A., and Ulrich, S., "Experimental Validation of Pseudospectral-Based Optimal Trajectory Planning for Free-Floating Robots," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 8, 2019, pp. 1726–1742.
https://doi.org/10.2514/1.G003528

[7] Aghili, F., "Optimal Control for Robotic Capturing and Passivation of a Tumbling Satellite with Unknown Dynamics," *AIAA Guidance, Navigation and Control Conference*, AIAA Paper 2008-7274, 2008.
https://doi.org/10.2514/6.2008-7274

[8] Boyarko, G., Yakimenko, O., and Romano, M., "Optimal Rendezvous Trajectories of a Controlled Spacecraft and a Tumbling Object," *Journal of Guidance, Control, and Dynamics*, Vol. 34, No. 4, 2011, pp. 1239–1252.
https://doi.org/10.2514/1.47645

[9] Lampariello, R., and Hirzinger, G., "Generating Feasible Trajectories for Autonomous On-Orbit Grasping of Spinning Debris in a Useful Time," *IEEE International Conference on Intelligent Robots and Systems*, Inst. of Electrical and Electronics Engineers, New York, 2013, pp. 5652–5659.
https://doi.org/10.1109/IROS.2013.6697175

[10] Virgili-Llop, J., Drew, J. V., Zappulla, R., and Romano, M., "Laboratory Experiments of Resident Space Object Capture by a Spacecraft–Manipulator System," *Aerospace Science and Technology*, Vol. 71, Dec. 2017, pp. 530–545.
https://doi.org/10.1016/j.ast.2017.09.043

[11] Virgili-Llop, J., Zagaris, C., Zappulla, R., Bradstreet, A., and Romano, M., "A Convex-Programming-Based Guidance Algorithm to Capture a Tumbling Object on Orbit Using a Spacecraft Equipped with a Robotic Manipulator," *International Journal of Robotics Research*, Vol. 38, No. 1, 2019, pp. 40–72.
https://doi.org/10.1177/0278364918804660

[12] Rybus, T., Seweryn, K., and Sasiadek, J. Z., "Control System for Free-Floating Space Manipulator Based on Nonlinear Model Predictive Control (NMPC)," *Journal of Intelligent and Robotic Systems*, Vol. 85, Sept. 2017, pp. 491–509.
https://doi.org/10.1007/s10846-016-0396-2

[13] Persson, S. M., and Sharf, I., "Ground-Based Experiments Towards the Interception of Non-Cooperative Space Debris with a Robotic Manipulator," *IEEE International Conference on Intelligent Robots and Systems*, IEEE Publ., Piscataway, NJ, 2015, pp. 5441–5446.
https://doi.org/10.1109/IROS.2015.7354147

[14] Aghili, F., "Time-Optimal Detumbling Control of Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 5, 2009, pp. 1671–1675.
https://doi.org/10.2514/1.43189

[15] Zhang, B., Liang, B., Wang, Z., Mi, Y., Zhang, Y., and Chen, Z., "Coordinated Stabilization for Space Robot After Capturing a Noncooperative Target with Large Inertia," *Acta Astronautica*, Vol. 134, May 2017, pp. 75–84.
https://doi.org/10.1016/j.actaastro.2017.01.041

[16] Gangapersaud, R. A., Liu, G., and De Ruiter, A. H., "Detumbling a Non-Cooperative Space Target with Model Uncertainties Using a Space Manipulator," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 4, 2019, pp. 910–918.
https://doi.org/10.2514/1.G003111

[17] Aghili, F., "Pre- and Post-Grasping Robot Motion Planning to Capture and Stabilize a Tumbling/Drifting Free-Floater with Uncertain Dynamics," *IEEE International Conference on Robotics and Automation*, Inst. of Electrical and Electronics Engineers, New York, 2013, pp. 5461–5468.
https://doi.org/10.1109/ICRA.2013.6631360

[18] Liu, X.-F., Zhang, X.-Y., Cai, G.-P., and Wang, M.-M., "A Collision Control Strategy for Detumbling a Non-Cooperative Spacecraft by a Robotic Arm," *Multibody System Dynamics*, Vol. 53, May 2021, pp. 225–255.
https://doi.org/10.1007/s11044-021-09793-x

[19] Virgili-Llop, J., and Romano, M., "Simultaneous Capture and Detumble of a Resident Space Object by a Free-Flying Spacecraft-Manipulator System," *Frontiers Robotics AI*, Vol. 6, No. 14, 2019, pp. 1–24.
https://doi.org/10.3389/frobt.2019.00014

[20] Izzo, D., Märtens, M., and Pan, B., "A Survey on Artificial Intelligence Trends in Spacecraft Guidance Dynamics and Control," *Astrodynamics*, Vol. 3, No. 4, 2019, pp. 287–299.
https://doi.org/10.1007/s42064-018-0053-6

[21] Chan, D. M., and Agha-Mohammadi, A. A., "Autonomous Imaging and Mapping of Small Bodies Using Deep Reinforcement Learning," *IEEE Aerospace Conference*, IEEE Publ., Piscataway, NJ, 2019, pp. 1–12.
https://doi.org/10.1109/AERO.2019.8742147

[22] Julian, K. D., and Kochenderfer, M. J., "Distributed Wildfire Surveillance with Autonomous Aircraft Using Deep Reinforcement Learning," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 8, 2019, pp. 1768–1778.
https://doi.org/10.2514/1.G004106

[23] Gaudet, B., Linares, R., and Furfaro, R., "Adaptive Guidance and Integrated Navigation with Reinforcement Meta-Learning," *Acta Astronautica*, Vol. 169, April 2020, pp. 180–190.
https://doi.org/10.1016/j.actaastro.2020.01.007

[24] Gaudet, B., Linares, R., and Furfaro, R., "Deep Reinforcement Learning for Six Degree-of-Freedom Planetary Landing," *Advances in Space Research*, Vol. 65, No. 7, 2020, pp. 1723–1741.
https://doi.org/10.1016/j.asr.2019.12.030

[25] Federici, L., Benedikter, B., and Zavoli, A., "Deep Learning Techniques for Autonomous Spacecraft Guidance During Proximity Operations," *Journal of Spacecraft and Rockets*, Vol. 58, No. 6, 2021, pp. 1774–1785.
https://doi.org/10.2514/1.A35076

[26] Oestreich, C. E., Linares, R., and Gondhalekar, R., "Autonomous Six-Degree-of-Freedom Spacecraft Docking with Rotating Targets via Reinforcement Learning," *Journal of Aerospace Information Systems*, Vol. 18, No. 7, 2021, pp. 417–428.
https://doi.org/10.2514/1.I010914

[27] Broida, J., and Linares, R., "Spacecraft Rendezvous Guidance in Cluttered Environments via Reinforcement Learning," *Advances in the Astronautical Sciences*, Vol. 168, Jan. 2019, pp. 1777–1788.

[28] Hovell, K., and Ulrich, S., "Deep Reinforcement Learning for Spacecraft Proximity Operations Guidance," *Journal of Spacecraft and Rockets*, Vol. 58, No. 2, 2021, pp. 254–264.
https://doi.org/10.2514/1.A34838

[29] Wu, Y.-H., Yu, Z.-C., Li, C.-Y., He, M.-J., Hua, B., and Chen, Z.-M., "Reinforcement Learning in Dual-Arm Trajectory Planning for a Free-floating Space Robot," *Aerospace Science and Technology*, Vol. 98, March 2020, Paper 105657.
https://doi.org/10.1016/j.ast.2019.105657

[30] Li, Y., Hao, X., She, Y., Li, S., and Yu, M., "Constrained Motion Planning of Free-Float Dual-Arm Space Manipulator via Deep Reinforcement Learning," *Aerospace Science and Technology*, Vol. 109, April 2021, Paper 106446.
https://doi.org/10.1016/j.ast.2020.106446

[31] Senda, K., Murotsu, Y., Mitsuya, A., Adachi, H., Ito, S., Shitakubo, J., and Matsumoto, T., "Hardware Experiments of a Truss Assembly by an Autonomous Space Learning Robot," *Journal of Spacecraft and Rockets*, Vol. 39, No. 2, 2002, pp. 267–273.
https://doi.org/10.2514/2.3808

[32] Zeng, A., Song, S., Lee, J., Rodriguez, A., and Funkhouser, T., "TossingBot: Learning to Throw Arbitrary Objects with Residual Physics," *Robotics: Science and Systems*, Vol. 36, No. 4, Aug. 2020.
https://doi.org/10.1109/TRO.2020.2988642

[33] Dasari, S., Ebert, F., Tian, S., Nair, S., Bucher, B., Schmeckpeper, K., Singh, S., Levine, S., and Finn, C., "RoboNet: Large-Scale Multi-Robot Learning," *Conference on Robotic Learning*, Brookline, MA, 2019, pp. 885–897.
https://doi.org/arXiv:1910.11215

[34] Wilde, M., Ciarcià, M., Grompone, A., and Romano, M., "Experimental Characterization of Inverse Dynamics Guidance in Docking with a Rotating Target," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 6, 2016, pp. 1173–1187.
https://doi.org/10.2514/1.G001631

[35] Zappulla, R., Park, H., Virgili-Llop, J., and Romano, M., "Real-Time Autonomous Spacecraft Proximity Maneuvers and Docking Using an Adaptive Artificial Potential Field Approach," *IEEE Transactions on Control Systems Technology*, Vol. 27, No. 6, 2019, pp. 2598–2605.
https://doi.org/10.1109/TCST.2018.2866963

[36] Saulnier, K., Pérez, D., Huang, R. C., Gallardo, D., Tilton, G., and Bevilacqua, R., "A Six-Degree-of-Freedom Hardware-in-the-Loop Simulator for Small Spacecraft," *Acta Astronautica*, Vol. 105, No. 2,

2014, pp. 444–462.
https://doi.org/10.1016/j.actaastro.2014.10.027

[37] Romano, M., Friedman, D. A., and Shay, T. J., "Laboratory Experimentation of Autonomous Spacecraft Approach and Docking to a Collaborative Target," *Journal of Spacecraft and Rockets*, Vol. 44, No. 1, 2007, pp. 164–173.
https://doi.org/10.2514/1.22092

[38] Romano, M., "Laboratory Experimentation of Autonomous Spacecraft Proximity-Navigation Using Vision and Inertia Sensors," *AIAA Guidance, Navigation, and Control Conference*, AIAA Paper 2005-6461, 2005.
https://doi.org/10.2514/6.2005-6461

[39] Romano, M., and Hall, J., "A Test Bed for Proximity Navigation and Control of Spacecraft for On-Orbit Assembly and Reconfiguration," *Space Conference*, AIAA Paper 2006-7519, 2006.
https://doi.org/10.2514/6.2006-7519

[40] Park, H., Zagaris, C., Virgili-Llop, J., Zappulla, R., Kolmanovsky, I., and Romano, M., "Analysis and Experimentation of Model Predictive Control for Spacecraft Rendezvous and Proximity Operations with Multiple Obstacle Avoidance," *AIAA/AAS Astrodynamics Specialist Conference*, AIAA Paper 2016-5273, 2016.
https://doi.org/10.2514/6.2016-5273

[41] Park, H., Zappulla, R., Zagaris, C., Virgili-Llop, J., and Romano, M., "Nonlinear Model Predictive Control for Spacecraft Rendezvous and Docking with a Rotating Target," *Advances in the Astronautical Sciences*, AAS Paper 2017–496, 2017.

[42] Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., Dhurva, T. B., Muldal, A., Heess, N., and Lillicrap, T., "Distributed Distributional Deterministic Policy Gradients," *International Conference on Learning Representations*, 2018.
https://doi.org/arXiv:1804.08617

[43] Hovell, K., Ulrich, S., and Bronz, M., "Acceleration-Based Quadrotor Guidance Under Time Delays Using Deep Reinforcement Learning," *AIAA Guidance, Navigation and Control Conference*, AIAA Paper 2021-1751, 2021.
https://doi.org/10.2514/6.2021-1751