2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Learning-Aided Control of Robotic Tether-Net with Maneuverable Nodes to Capture Large Space Debris

Achira Boonrath[1†], Feng Liu[1†], Eleonora M. Botta[1], and Souma Chowdhury[1]

***Abstract*— Maneuverable tether-net systems launched from an unmanned spacecraft offer a promising solution for the active removal of large space debris. Guaranteeing the successful capture of such space debris is dependent on the ability to reliably maneuver the tether-net system – a flexible, many-DoF (thus complex) system – for a wide range of launch scenarios. Here, scenarios are defined by the relative location of the debris with respect to the chaser spacecraft. This paper represents and solves this problem as a hierarchically decentralized implementation of robotic trajectory planning and control and demonstrates the effectiveness of the approach when applied to two different tether-net systems, with 4 and 8 maneuverable units (MUs), respectively. Reinforcement learning (policy gradient) is used to design the centralized trajectory planner that, based on the relative location of the target debris at the launch of the net, computes the final aiming positions of each MU, from which their trajectory can be derived. Each MU then seeks to follow its assigned trajectory by using a decentralized PID controller that outputs the MU's thrust vector and is informed by noisy sensor feedback (for realism) of its relative location. System performance is assessed in terms of capture success and overall fuel consumption by the MUs. Reward shaping and surrogate models are used to respectively guide and speed up the RL process. Simulation-based experiments show that this approach allows the successful capture of debris at fuel costs that are notably lower than nominal baselines, including in scenarios where the debris is significantly off-centered compared to the approaching chaser spacecraft.**

***Index Terms*— PID control, reinforcement learning, robotic tether net, space debris removal**

I. Introduction

Active Debris Removal (ADR) has emerged as a promising solution to counteract the increasing threats posed by space debris to satellites and other assets orbiting the Earth. Among various proposed ADR methodologies, using a tether-net system possesses a high potential for success due to the system's relatively low weight and ability to capture massive rotating debris from a secure distance [1]–[4]. In recent years, many researchers have performed simulation-based and experiment-based analyses on net systems' deployment and capture dynamics [5]–[10]. Most past research on the topic focuses on passive tether-net systems, which function through the ejection of masses attached to the perimeter or edges of nets toward target debris. The trajectory of such systems cannot be altered once the net has been launched from the chaser. To further improve the capabilities of tether-net systems, employing maneuverable space nets enhances the mission's flexibility and dependability in capturing uncooperative space debris. Addressing this need, Huang et al. [11]–[13] introduced the concept of the Tethered Space Net Robot (TSNR) system. The TSNR configuration comprises a net connected via threads to Maneuverable Units (MUs). Compared to a purely passive net, using MUs allows the system to increase flexibility through its ability to alter its trajectory after launching from the chaser.

Within the literature, the setpoints for the TSNR system to fly towards are predetermined before the initialization of the net deployment [11], [12]. This method assumes that the chaser spacecraft is able to always perfectly approach the debris based on a pre-determined proximity approach trajectory. Not only is this unlikely in practice, but also differing debris rotational rates (that may not be known a priori) could demand different approach states. Hence, an approach that allows determining setpoints or optimal net deployment on the fly, based on the estimation of the debris' state relative to the chaser, is premised to provide a significantly more generalizable solution to debris capture. We posit that Reinforcement Learning (RL) can be used to train such generalized policies across various approach scenarios. Therefore, our work introduces a RL-guided autonomous maneuverable net system, in which MUs are guided to provide successful capture of space debris at minimal fuel costs, based on the state of the debris relative to the chaser. It is assumed that the MUs are equipped with cold-gas thrusters commanded by PID controllers.

In addition to the 4-MU TSNR system design from prior work by the authors [14], [15], this work introduces a TSNR system with 8 MUs. The alternate design explores the feasibility of using maneuvers of docking among the MUs to replace the closing mechanism (that is otherwise necessary [8]) in ensuring capture success, and the potential benefits thereof. The system components are shown in Fig. 1 for the 8-MU net, which is identical to the 4-MU net, except for the 4 additional MUs on the net sides and the lack of a separate closing mechanism.

In our previous work, RL [16] and neuroevolution [17], [18] techniques were respectively used to perform open-loop thrust control of a maneuverable net system [15] and purely mechanical net ejection and closing control [19], both with

[1]Department of Mechanical and Aerospace Engineering, University at Buffalo, Buffalo, NY 14260, USA
†These authors contributed equally
{achirabo, fliu23, ebotta, soumacho}@buffalo.edu

*This work is supported under the CMMI Award numbered 2128578 from the National Science Foundation (NSF). The author's opinions, findings, and conclusions or recommendations expressed in this material do not necessarily reflect the views of the National Science Foundation. The authors would also like to thank CM Labs Simulations for providing licenses for the Vortex Studio simulation framework.

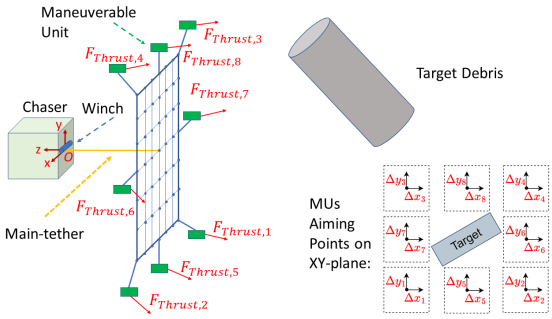


Fig. 1: Representation of the 8-MU Tether-net System

the goal to maximize the success rate of debris capture. Instead, in this paper, we assume that each MU is able to fly a trajectory using a PID controller to reach a desired end location, which we call the *aiming point*, as shown in Fig. 1. This aiming point is determined by a policy model trained using RL, given the debris' relative location with respect to the chaser. To update the policy, the Capture Quality Index (CQI), the number of locked pairs of closing nodes of the net, the total fuel consumption of the MUs, and the net mouth area are used as the simulated evaluation metrics to compute a reward function. To circumvent the cost of simulating the net/debris contact mechanics, and therefore speed up the RL process, a surrogate model is trained to predict the capture quality based on the position and velocities of selected points on the net right before the net mouth starts closing.

The primary contributions of this paper and its outline can be summarized as: **1)** We present a new concept of operation for tether-net systems based on autonomously maneuverable MUs in tandem with or without a closing mechanism, depending on the number of MUs (Sec. II). **2)** We present a new closed-loop (PID) controller for trajectory tracking by each MU under sensing uncertainties (Sec. II). **3)** We present a simulation environment that combines the net's deployment dynamics model with a surrogate model of the capture success, leading to an 8-fold speed-up of the RL process; **4)** We formulate the trajectory planning problem for each MU as a time-encoded linear path that is defined by its initial position and aiming point (on the plane parallel to the inertial X-Y plane and intersecting the center of mass of the debris or target), with the aiming point computed by solving a Markov Decision Process (MDP) using the policy gradient method (Sec. III). Performance of the resulting tether-net maneuver process is compared to nominal cases in Sec. IV.

II. Tether-Net: ConOps and Simulation

A. Concept of Operation (ConOps)

In the first stage of the net-based ADR operation, the chaser spacecraft is launched and rendezvous with the target debris in orbit. Once in close proximity, the relative displacement between the target debris and the chaser is determined through a LiDAR system installed on the chaser; this concept was demonstrated as a part of the RemoveDEBRIS in-orbit demonstration mission [20], [21]. Before the launch of the TNSR, considering the debris state, a semi-decentralized RL-based trajectory planner computes and relays information regarding the optimal desired position at the end of deployment for each MU (aiming points). To launch the net, the MUs utilize their propulsion capabilities; this eliminates the necessity for an ejection mechanism, thus removing either gas-based or spring-based launchers required for passive net designs [2], [8]. Employing closed-loop control, each MU applies thrusts to follow a designated trajectory until a start-of-closing condition is reached (see Sec. II-B). To secure the debris, the net mouth is then closed by a closing mechanism in the case of the 4-MU system, or purely by the thrusters in the case of the 8-MU system. No human intervention is involved from the moment of debris position determination to the end of the capture. After the debris is secured, the chaser satellite is assumed to bring it to a disposal orbit; the post-capture phase is not simulated in this work, but is the topic of concurrent work [22].

B. Simulation Description

The modeling of the deployment of the TSNR and the closing of the net around the debris are performed using a formerly assembled simulator in the multibody dynamics simulation framework Vortex Studio. The chaser satellite is modeled as a cubic rigid body with a side length of 1.5 m and a mass of 1600 kg. During the capture process, the chaser spacecraft is assumed to be floating without control to enable the examination of motion induced by tension in the main tether. The main tether links the net's central knot to a winch on the chaser, which is designed to spool freely during deployment and to remain locked during the capture phase [2], [8]. The target debris used here is the ∼9000 kg second stage of the Zenit-2 rocket, which is considered to be one of the most dangerous debris currently orbiting the Earth [23], [24].

The lumped-parameter modeling method represents a flexible net within the simulation framework [2], [8]. The process lumps the mass of the net into many small spherical rigid bodies (called *nodes*); the nodes of the net are linked together into a mesh via spring-damper elements that are unable to withstand compression [2], [8]. The MUs are modeled as small rectangular prism-shaped rigid bodies, each possessing a dimension of 0.1 m × 0.1 m × 0.2 m and a total weight of 2.5 kg. The simulator leverages a scaled-box friction model to compute the frictional forces during contact (approximating Coulomb's friction model); the normal contact forces are calculated by continuous compliant contact forces modeling. The parameters of the nets studied within this paper are the same as in [15]. For more comprehensive insights into the modeling of the net and of contact dynamics, readers are encouraged to refer to prior work by Botta et al. [2], [8], [25].

Two system variants are considered for the TNSR. The first is characterized by a 4-MU layout, also studied in previous work [15]. In this case, a closing mechanism is utilized to close the mouth of the net around debris after contact; it consists of threads that are interlaced around the net's perimeter and of winches placed in each MU. The winches reel the threads into the MUs at the net

closing activation time, thus pulling the nodes together. The maximum number of locked pairs, i.e., the number of pairs of adjacent nodes on the closing mechanism that are locked together, is $\max(N_{\mathrm{L}}) = 12$.

As an alternative design, a net that utilizes the maneuvering capabilities of the MUs to close the net mouth is introduced. Since in previous work by Botta et al. [8] it was observed that solely relying on the four corner masses to close the net's mouth might fail to maintain a secure hold on the targeted debris, four additional MUs are introduced to assist in closing the net perimeter. Each of the four additional MUs is attached to the center of a side of the net via a thread. At the start-of-closing time, the MUs fly towards a set closing position and dock with each other. To simulate the docking process, when the adjacent MUs are within a distance of 0.5 m, *distance joint* constraints with a 0 m nominal length are engaged within the simulation: this ensures that the net mouth remains closed and capture is maintained. In this 8-MU design, the maximum value of locked pairs $\max(N_{\mathrm{L}}) = 8$ should be achieved for an ideal capture. In the 8-MU design, each of the MUs could be simpler and smaller due to the absence of a winch, compensating for the increase in total system mass. Additionally, the new design benefits from increased redundancy since the failure of 1 MU is foreseen to have a smaller overall effect on net deployment.

C. Thrust Control

To control the motion of the MUs, thrust forces, produced by cold-gas thrusters and defined as $\mathbf{F}_{\mathrm{T},i}$, are employed. For state estimation, each MU is assumed to be equipped with an inertial measurement unit with a sampling rate of 20 Hz (based on sensors currently employed within CubeSats [26]–[28]). To represent sensing or measurement errors, random noises within 3σ bounds of $\pm$ 0.1 m and $\pm$ 0.1 m/s are sampled from a Gaussian distribution and respectively added to the positions and velocities obtained from the simulation. These slightly noisy state values are utilized by each controller, with the actuation assumed to be perfect.

During deployment, PID controllers compute the thrust needed to bring each MU to its desired position. At each command timestep, the desired position of i-th MU is:

$$\mathbf{r}_{d,i}(t) = \mathbf{r}_{0,i} + \frac{t}{t_{\text{final}}}(\mathbf{r}_{\text{final},i} - \mathbf{r}_{0,i}) \tag{1}$$

where $\mathbf{r}_{0,i}$, t_{final}, and $\mathbf{r}_{\text{final},i}$ are the initial position of the i-th MU, the desired duration of deployment, and the desired position of the MU at the end of the deployment, respectively. The position $\mathbf{r}_{\text{final},i}$ is the position of the aiming point (located on the plane parallel to the inertial X-Y plane and passing through the debris' center of mass), which is the point the i-th MU must aim for. Note that in practice, the closing mechanism is usually triggered slightly before the MUs reach their aiming points since the trigger event is defined as the time when the net's center of mass is 2.5 m away from the debris' center of mass [15]. The RL-trained policy determines these aiming positions $\mathbf{r}_{\text{final},i}$ for each MU based on the state of the debris with respect to the point of net-launch at the instant when the tether net is launched. In this work, t_{final} is a user-prescribed net's deployment time value, set as 25 s.

The PID controller assumes that thrust can be independently controlled in the x, y, and z directions. As such, the control force for the i-th MU is computed as:

$$\mathbf{F}_{\mathrm{T},i} = K_P(\mathbf{r}_{d,i} - \mathbf{r}_i) - K_D\dot{\mathbf{r}}_i + K_I\int_0^t (\mathbf{r}_{d,i} - \mathbf{r}_i)\, d\tau \tag{2}$$

where K_P, K_D, and K_I are the proportional, derivative, and integral gains, respectively. Variables $\mathbf{r}_i$ and $\dot{\mathbf{r}}_i$ are the position and inertial velocity of the i-th MU at any time. Activation of the thrusters takes place at $t = 0$ s. For the 4-MU net, the thrusters deactivate once the predetermined distance between the center of mass of the net and the debris is achieved. The magnitude of the thrust in each direction is limited to 5.1 N, chosen so that the total maximum thrust available for each MU is (approx. 8.9 N) the same as in previous work [15]. Each thruster has a command frequency of 20 Hz [29]. In between each command signal, a zero-order hold is utilized. The structure of the PID controller is shown in the dashed box in Fig. 2.

To compute the fuel consumption of the thruster on each MU, a specific impulse $I_{sp} = 60$ s is chosen, based on existing cold-gas thruster technology [30]. For the i-th MU, the total fuel utilized throughout the maneuver is:

$$m_{\mathrm{f},i} = \int_0^{t_{\text{end}}} \frac{\bar{F}_{\mathrm{T},i}}{(g_0 I_{sp})}\, dt \tag{3}$$

where g_0 is the gravitational acceleration at Earth's sea level (i.e., 9.81 m/s^2), and $\bar{F}_{\mathrm{T},i}$ is the magnitude of the saturated thrust force. Variable t_{end} indicates the final simulation time, set to be 15.0 s after the start of the closing of the net's mouth (achieved using a winch in the case of the 4-MU system or the MUs themselves in the case of the 8-MU system). Therefore, Eq. (3) accounts for both deployment and closure.

D. Capture Quality Index and Locked Pairs

A quantitative measure of the quality of debris capture, called CQI, is utilized for this work. This provides a measure of the ability of the TSNR to wrap around and hold onto the debris without examining simulation graphics [31], [32]. The CQI compares the geometries of the net's Convex Hull (CH) and of the debris, in addition to accounting for the distance between their centers of mass. The CQI is thus defined as:

$$I_{\mathrm{CQI},n} = 0.1\frac{|V_n - V_D|}{V_{\mathrm{D}}} + 0.1\frac{|S_n - S_{\mathrm{D}}|}{S_{\mathrm{D}}} + 0.8\frac{|q_n|}{L_c} \tag{4}$$

At the n-th time-step, V_n, V_{D}, S_n, and S_{D} respectively represent the net's CH volume, the debris's volume, the net's CH surface area, and the debris's surface area; q_n and L_c respectively represent the distance between the net's and debris's centers of mass and the debris's characteristic length. The CQI can distinguish between successful and unsuccessful captures, as shown in [32]. In this work, however, the CQI is complemented by the number of locked pairs, which provides information on the fact that capture will not be lost.

In the current framework, a successful capture is defined to have a settled CQI (i.e., CQI value 15 s after closing activation) lower than 2.5 and a number of locked pairs greater than a certain threshold (8 for the 4-MU net and 6 for the 8-MU net). For the second stage of the Zenit-2 rocket, the geometric propertiesare $V_{\mathrm{D}} = 159.9$ m^3, $S_{\mathrm{D}} = 59.9$ m^2, and $L_c = 1.95$ m.

E. Surrogate Modeling to Predict Capture Status

With the high-fidelity simulator, computing time could become excessive for the simulation to be directly used in optimization or learning processes. Over a set of 50 random successful capture scenarios (running in parallel on a 16-Core Processor Windows workstation with 64 GB RAM, where on average, deployment took 25.2 s of physical time and capture was simulated for 15 s of physical time), the average computing time taken to simulate the deployment phase and the capture phase were observed to be respectively 97.1 s and 206.7 s. Hence, the computing time is dominated by the simulation of the capture process, which involves net closing and net/debris contact. This motivated the use of surrogate models to substitute the capture simulation. Through numerical experiments, recurrent neural network (RNN) modeling was observed to outperform other multi-variate regression or surrogate models (e.g., multi-layer perceptron, random forest regression). Hence, RNN is used in this work as a surrogate model to predict the CQI and the number of locked pairs in the capture phase.

The inputs to the RNN model are the positions and velocities of MUs and of 165 nodes on the net, relative to the debris's center of mass. The 165 net nodes (out of 533 total) form three loops on the net, which are used as a reference to represent the net's configuration in space. High-fidelity simulation stops at the start-of-closing time. Subsequently, the RNN reads the position and velocity (input) data of the MUs and of selected nodes and predicts the final CQI and number of locked pairs at time t_{end}. The latter are used in the RL framework to compute the reward function as will be explained in Eq. (5).

The RNN network for the 4-MU case study consists of 990 neurons in the input layer, 500 and 300 neurons in two consecutive hidden layers, and two neurons in the output layer, which provide the values of the CQI and number of locked pairs. The RNN network for the 8-MU case study has a similar structure, except that the input layer comprises 1014 neurons to account for the states of more MUs. The RNN model for the 4-MU system was trained with 2540 scenarios, and that for the 8-MU system was trained with 2790 scenarios, both using a 0.00001 learning rate and 500 epochs. The models were then tested over 200 unseen scenarios, where the accuracy of computing whether a capture is successful (refer to constraints in Eq. (5) given by the RNN predictions) was found to be over 98%. To compensate for the RNN's prediction error (which was observed to be high in failed capture cases) and promote robust learning for scenarios with CQI<20, the prediction error is represented by a Gaussian distribution. This serves as the noise model from which the CQI is sampled during the RL process to compute rewards.

III. Formulation of the Learning Task

The objective of RL in this paper is to find the aiming points in different launch scenarios, where scenarios (state space) are defined in terms of the x, y, and z coordinates of the debris' center of mass. These aiming points define the optimal positions to be achieved by the MUs when the closing process is initiated. Optimal aiming points should minimize the fuel cost (due to the MUs thrusting to maneuver the net) while ensuring that debris is successfully captured, with the latter determined in terms of settled CQI and number of locked pairs.

Proximal Policy Optimization (PPO) [33], provided by stable-baselines3 [34], is used to perform the RL process. The learning framework is shown in Fig. 2, comprising the following major elements: 1) the policy model being trained by PPO, 2) the high-fidelity simulation, including the PID controller, to model the net deployment, and 3) the RNN based surrogate model to predict the measures of capture success.

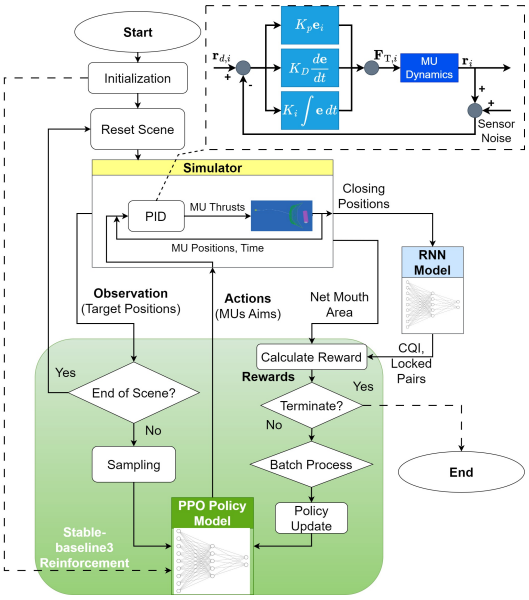


Fig. 2: Computational Framework for Reinforcement Learning of Aiming and Control of the MUs

The Markov Decision Process over which this RL process operates can be described in terms of the state space, the action space, the state transition model, and the reward function.

State Space: The state space comprises the position vector of the center of mass of the debris at the instant of net launch, and is expressed as $S = [x_{\mathrm{D}}, y_{\mathrm{D}}, z_{\mathrm{D}}]$. Note that it is a one step experience here, that is, a single action is taken given this state in every episode simulating a net launch scenario. The state transition model is stochastic since noise is added to the simulation to mimic sensor imperfections and to the RNN model to account for uncertainty in the predicted CQI and locked pairs.

Action Space: The actions of the RL policy are the offsets of the aiming points for each MU relative to nominal aiming points (defined as a baseline) Δx_i and Δy_i, as shown in Fig. 1. The position of the nominal point for the i-th MU can be expressed as $(x_{\text{nom},i}, y_{\text{nom},i}, z_{\text{D}})$; the nominal positions are defined in Table II. The nominal aiming point allows the net to be flattened out at the approximate time the closing of the net mouth starts. Both the 4-MU and 8-MU designs use the same nominal coordinates for MU 1 to MU 4. The RL-trained policy model is tasked to compute where the optimal aiming points must be placed, compared to the nominal points. The actual aiming point for any i-th MU is thus given by $\mathbf{r}_{\text{final},i} = (x_{\text{nom},i} + \Delta x_i)\hat{\mathbf{i}} + (y_{\text{nom},i} + \Delta y_i)\hat{\mathbf{j}} + z_{\text{D},i}\hat{\mathbf{k}}$. Hence, the actions computed by the RL policy for each MU are $A_i = [\Delta x_i, \Delta y_i]$. Both nominal and policy-based z-coordinates for all the aiming points are set to be the same to prevent the net configuration from becoming too asymmetrical.

Reward: The reward function is shown in Eq. (5), which considers multiple factors. Firstly, it rewards the policy if the MUs spend less fuel in total and are able to expand the net mouth wider before the closing mechanism is triggered. The reward function penalizes the policy if the MUs fail to capture the target debris, which is determined based on the value of settled CQI and the number of locked pairs. In Eq. (5), Φ is the RL policy. Here, $b_{\text{fuel}} = 1 - \frac{m_{\text{f}}}{m_{\text{fmax}}}$ to promote higher reward when the fuel consumption is lower, where m_{f} is the total fuel consumption of all MUs and m_{fmax} is the reference maximum total fuel consumption, which is set based on prior observations; w is a positive weighting parameter, set at 1 in the 4-MU case and 1.5 in the 8-MU case; A_{c} is the net mouth area when the closing mechanism activates, and A_{max} is the maximum net mouth area; I^*_{CQI} and N_{L} are respectively the CQI and number of locked pairs 15 s after the net mouth closing is triggered; N_{D} is the minimum number of locked pairs for successful capture (8 for the 4-MU case and 6 for the 8-MU case).

TABLE I: Bounds of RL States and Actions

	Variables	Bounds	Step Size	Unit
State	x_{D}	-9 to 9	0.1	m
	y_{D}	-9 to 9	0.1	m
	z_{D}	-60 to -40	0.1	m
Action	$\Delta x_{i,i=1,2,\ldots,N}$	-5 to 5	0.1	m
	$\Delta y_{i,i=1,2,\ldots,N}$	-5 to 5	0.1	m

* N is the number of MUs.

$$\max_{\Phi} \quad R = b_{\text{mouth}} + p_{\text{CQI}} + p_{\text{NL}} + b_{\text{end}} \tag{5}$$

where: $b_{\text{mouth}} = A_c / A_{\text{max}}$

$$p_{\text{CQI}} = \begin{cases} -\ln((I^*_{\text{CQI}} - 2.5)^2 + 1), & \text{if } I^*_{\text{CQI}} > 2.5 \\ 0, & \text{otherwise} \end{cases}$$

$$p_{\text{NL}} = \begin{cases} -\ln((N_{\text{L}} - N_{\text{D}})^2 + 1), & \text{if } N_{\text{L}} < N_{\text{L,max}} \\ 0, & \text{otherwise} \end{cases}$$

$$b_{\text{end}} = \begin{cases} w\, b_{\text{fuel}}, & \text{if } I^*_{\text{CQI}} \leq 2.5 \wedge N_{\text{L}} \geq N_{\text{L,max}} \\ 0, & \text{otherwise} \end{cases}$$

IV. Simulation Results and Analysis

A. Net Deployment PID Control: Results

A Routh table was constructed to determine the stable range of gain values to select controller gains. Values of $K_P = 10.0$, $K_I = 6.0$, and $K_D = 6.0$ were picked from the determined range and used for each MU. The performance of the controller at tracking a given deployment trajectory is demonstrated for the 8-MU case in Fig. 3, which displays the magnitude of the position error over time, computed as:

$$e(t)_i = ||\mathbf{r}_{d,i}(t) - \mathbf{r}_i(t)|| \tag{6}$$

It can be seen that the error for all MUs is minimal after 10 seconds post-launch, demonstrating the ability of the controller to track the given position input most of the time throughout the deployment. A slight increase in the position error is observed starting at approximately 24 s, which corresponds to when the net becomes almost flattened out. To improve the controller performance in future work, tuning the PID gains may reduce the time required for the MUs to reach a near-zero position error and the tracking ability towards the end of the deployment. In Fig. 3, the net-to-debris's centers of mass distance is also shown to steadily decrease during the deployment, confirming the approach of net towards the debris.

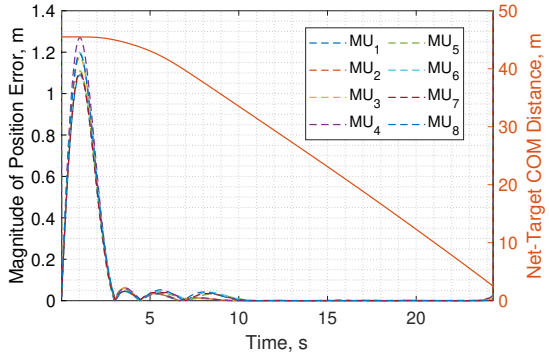


Fig. 3: L2-Norm of MUs position error (left y-axis) and distance of net from the debris (right y-axis) over time

B. Aiming Point Planning via RL: Results

A Windows workstation with an AMD Ryzen 9 5950X 16-core Processor and 64 GB RAM was utilized for the RL training for both the 4-MU and 8-MU designs. The training was performed with 32 episodes in parallel, with a mini-batch size of 64 and a learning rate of 0.001. For the 4-MU design, 11360 episodes were used in the learning process, which took 14 hours. For the 8-MU case, 11360 episodes were used, which took 10.5 hours.

The corresponding reward averaged over 32 consecutive episodes is reported to show the training history in Fig. 4. For the chosen reward function, the reward for each episode takes values between -2.0 and 2.5. For each episode, a negative reward corresponds to a failed capture caused by completely missing the debris or by failure to activate the closing procedure. A reward between 0 and 1 corresponds to a capture where the CQI threshold or number of locked pairs

TABLE II: Nominal End-of-Deployment Coordinates for Each MU

Coordinates, m	MU 1	MU 2	MU 3	MU 4	MU 5	MU 6	MU 7	MU 8
x_{nom}	x_{D}-12.00	x_{D}+12.00	x_{D}-12.00	x_{D}-12.00	x_{D}	x_{D}+11.70	x_{D}-11.71	x_{D}
y_{nom}	y_{D}-12.00	y_{D}-12.00	y_{D}+12.00	y_{D}+12.00	y_{D}-11.71	y_{D}	y_{D}	y_{D}-11.71

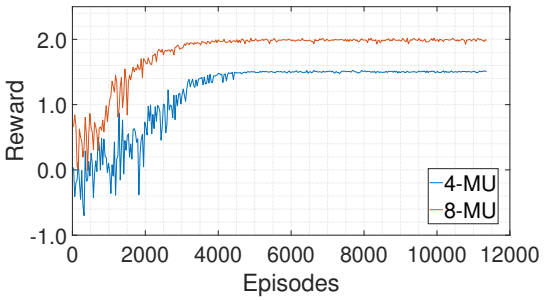
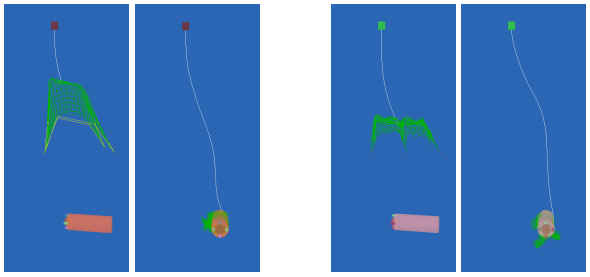


Fig. 4: Training History: Reward Averaged Over 32 Episodes

threshold is not reached. A reward greater than 1 corresponds to a successful capture. From the training history in Fig. 4, it can be seen that in the 4-MU and 8-MU cases, the reward converges to a value of approximately 1.5 and 2.0, respectively. This observation confirms the effectiveness of the designed RL framework and settings.

Figure 5 shows snapshots of key moments in an example scenario, with a notable offset of the debris of $x_{\text{D}} = 8.9$ m, $y_{\text{D}} = 6.2$ m, and $z_{\text{D}} = -44.0$ m, for (a) the 4-MU and (b) 8-MU systems. The left images display the systems in an early stage of the deployment, while the right images display the systems approximately 10 s after the closing of net mouth was initiated, demonstrating successful capture.



(a) 4-MU at 15 s and 35 s (b) 8-MU at 15 s and 35 s

Fig. 5: Simulation Screenshots Utilizing RL Guidance

To evaluate the performance of the RL policy model, 100 random unseen test scenarios (with different locations of the debris with respect to the chaser) are used. Evaluation is performed using the high fidelity simulation from net launch to the end of capture. The capture success rate over the testing samples is observed to be 100%. This is an important improvement over the Case 1 RL policy in our earlier work [15], which considered a direct open-loop control of the thrusts of each MU in the 4-MU case and reported 88% success over a more limited range of debris/chaser offsets.

Figure 6 shows the comparison of the fuel consumption by the RL-trained systems vs. using the nominal aiming points (see Table II). In particular, the reduction in fuel mass from the nominal setting to the RL cases (i.e., $m_{\text{f,nom}} - m_{\text{f,RL}}$, the greater, the better) is plotted as boxplots for the same scenarios. On average, the RL-based policies reduce fuel

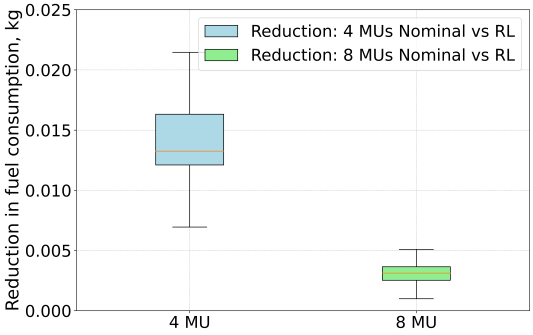


Fig. 6: Comparison of Total Fuel Consumption Reduction for 4- and 8-MU Cases

consumption by 11% in the 4-MU case, and by 1.8% in the 8-MU case, in comparison to their respective nominal aiming baselines. In addition, we observed that, across the test scenarios, the average fuel consumption of each MU by the 4-MU system was higher (at 0.028 kg) than that by the 8-MU system (at 0.021 kg). The rate of capture success remained 100% across the 100 test scenarios for both the RL-trained 4-MU and 8-MU cases. Collectively, these observations provide evidence of the feasibility of performing effective closing directly with MUs without the use of a winch mechanism.

V. Conclusion

This paper proposed learning-aided PID control of the maneuverable units (MUs) of a tether-net launched from a chaser spacecraft for the capture of large space debris. More specifically, RL is used to learn the aiming points for the MUs as a function of the location of the debris relative to the chaser. Based on that, a trajectory is derived and flown thanks to PID controllers, which decide the thrusts on the MUs. To accelerate the RL process, an RNN-based surrogate model is used to replace the more time-consuming portion of the simulation, and predict capture quality metrics. Tested on two different net designs (i.e., a 4-MU system with winch-based closing, and a 8-MU system that uses the MUs to close), the proposed framework is found to provide significant reduction in total fuel costs compared to the baseline, while ensuring 100% capture success over unseen test scenarios. At present, the control framework utilized in this work ignores the attitude dynamics of the MUs, which is an important concern in actual spaceflight. Consideration of position and attitude control in the future will allow us to also reliably expand the action space to include both aiming position and velocity. This, along with increased realism in considering vision-based sensing of the debris state and accounting for its rotation rates, will enable us to generate further insights regarding the effectiveness of autonomous tether-net systems for space debris capture.

REFERENCES

[1] A. Ledkov and V. Aslanov, "Review of contact and contactless active space debris removal approaches," *Progress in Aerospace Sciences*, vol. 134, p. 100858, 2022.

[2] E. M. Botta, "Deployment and capture dynamics of tether-nets for active space debris removal," Ph.D. dissertation, McGill University, Montreal, QC, 2017.

[3] M. Shan, J. Guo, and E. Gill, "A review and comparison of active space debris capturing and removal methods," *Progress in Aerospace Sciences*, vol. 80, pp. 18–32, 2015.

[4] S. Chen, C. T. Woods, A. Boonrath, and E. M. Botta, "Analysis of the robustness and safety of net-based debris capture," in *AIAA SCITECH 2022 Forum*, 2022, p. 1001.

[5] R. Benvenuto, S. Salvi, and M. Lavagna, "Dynamics analysis and gnc design of flexible systems for space debris active removal," *Acta Astronautica*, vol. 110, pp. 247–265, 2015.

[6] A. Medina, L. Cercós, R. M. Stefanescu, R. Benvenuto, V. Pesce, M. Marcon, M. Lavagna, I. González, N. R. López, and K. Wormnes, "Validation results of satellite mock-up capturing experiment using nets," *Acta Astronautica*, vol. 134, pp. 314–332, 2017.

[7] W. Golebiowski, R. Michalczyk, M. Dyrek, U. Battista, and K. Wormnes, "Validated simulator for space debris removal with nets and other flexible tethers applications," *Acta Astronautica*, vol. 129, pp. 229–240, 2016.

[8] E. M. Botta, I. Sharf, and A. K. Misra, "Simulation of tether-nets for capture of space debris and small asteroids," *Acta Astronautica*, vol. 155, pp. 448–461, 2019.

[9] Y. Endo, H. Kojima, and P. M. Trivailo, "Study on acceptable offsets of ejected nets from debris center for successful capture of debris," *Advances in Space Research*, vol. 66, no. 2, pp. 450–461, 2020.

[10] Y. Hou, C. Liu, H. Hu, W. Yang, and J. Shi, "Dynamic computation of a tether-net system capturing a space target via discrete elastic rods and an energy-conserving integrator," *Acta Astronautica*, vol. 186, pp. 118–134, 2021.

[11] P. Huang, F. Zhang, J. Ma, Z. Meng, and Z. Liu, "Dynamics and configuration control of the maneuvering-net space robot system," *Advances in Space Research*, vol. 55, no. 4, pp. 1004–1014, 2015.

[12] F. Zhang and P. Huang, "Stability control of a flexible maneuverable tethered space net robot," *Acta astronautica*, vol. 145, pp. 385–395, 2018.

[13] Y. Zhao, F. Zhang, and P. Huang, "Dynamic closing point determination for space debris capturing via tethered space net robot," *IEEE transactions on aerospace and electronic systems*, vol. 58, no. 5, pp. 1–1, 2022.

[14] C. Zeng, G. R. Hecht, S. Chowdhury, and E. M. Botta, "Concurrent design optimization of tether-net system and actions for reliable space-debris capture," *Journal of Spacecraft and Rockets*, vol. 0, no. 0, pp. 1–11, 0. [Online]. Available: https://doi.org/10.2514/1.A35812

[15] F. Liu, A. Boonrath, P. KrisshnaKumar, E. M. Botta, and S. Chowdhury, "Learning constrained corner node trajectories of a tether net system for space debris capture," in *AIAA AVIATION 2023 Forum*, 2023, p. 3920.

[16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017. [Online]. Available: http://arxiv.org/abs/1707.06347

[17] A. Behjat, S. Chidambaran, and S. Chowdhury, "Adaptive genomic evolution of neural network topologies (agent) for state-to-action mapping in autonomous agents," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 9638–9644.

[18] A. Behjat, N. Maurer, S. Chidambaran, and S. Chowdhury, "Adaptive neuroevolution with genetic operator control and two-way complexity variation," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 6, pp. 1627–1641, 2023.

[19] C. Zeng, G. R. Hecht, P. K. Kumar, R. K. Shah, E. M. Botta, and S. Chowdhury, "Learning robust policies for generalized debris capture with an automated tether-net system," in *AIAA SCITECH 2022 Forum*. American Institute of Aeronautics and Astronautics, jan 2022. [Online]. Available: https://doi.org/10.2514%2F6.2022-2379

[20] G. S. Aglietti, B. Taylor, S. Fellowes, S. Ainley, D. Tye, C. Cox, A. Zarkesh, A. Mafficini, N. Vinkoff, K. Bashford *et al.*, "Removedebris: An in-orbit demonstration of technologies for the removal of space debris," *The Aeronautical Journal*, vol. 124, no. 1271, pp. 1–23, 2020.

[21] T. Chabot, K. Kanani, A. Pollini, F. Chaumette, E. Marchand, and J. Forshaw, "Vision-based navigation experiment onboard the removedebris mission," in *GNC 2017-10th International ESA Conference on Guidance, Navigation & Control Systems*, 2017, pp. 1–23.

[22] L. Field and E. M. Botta, "Relative distance control of uncooperative tethered debris," *The Journal of the Astronautical Sciences*, vol. 70, no. 6, p. 55, 2023.

[23] D. McKnight, R. Witner, F. Letizia, S. Lemmens, L. Anselmo, C. Pardini, A. Rossi, C. Kunstadter, S. Kawamoto, V. Aslanov *et al.*, "Identifying the 50 statistically-most-concerning derelict objects in leo," *Acta Astronautica*, vol. 181, pp. 282–291, 2021.

[24] C. Wiedemann, S. Flegel, M. Möckel, J. Gelhaus, V. Braun, C. Kebschull, J. Kreisel, M. Metz, and P. Vörsmann, "Cost estimation of active debris removal," in *63rd International Astronautical Congress*. International Astronautical Federation Naples, Italy, 2012.

[25] E. M. Botta, I. Sharf, A. K. Misra, and M. Teichmann, "On the simulation of tether-nets for space debris capture with Vortex Dynamics," *Acta Astronautica*, vol. 123, pp. 91–102, 2016.

[26] A. Colagrossi, M. Lavagna, and R. Bertacin, "An effective sensor architecture for full-attitude determination in the hermes nano-satellites," *Sensors*, vol. 23, no. 5, p. 2393, 2023.

[27] D. S. Sanders, D. L. Heater, S. R. Peeples, P.-H. A. Huang *et al.*, "Pushing the limits of cubesat attitude control: a ground demonstration," in *Small Satellite Conference*, no. SSC13-III-10, 2013.

[28] S. K. Tullino and E. D. Swenson, "Testing and evaluating deployment profiles of the canisterized satellite dispenser (csd)," in *55th AIAA Aerospace Sciences Meeting*, 2017, p. 0850.

[29] Y. Silik and U. Yaman, "Control of rotary inverted pendulum by using on–off type of cold gas thrusters," in *Actuators*, vol. 9, no. 4. MDPI, 2020, p. 95.

[30] B. Yost and S. Weston, "State-of-the-art: Small spacecraft technology," 2023.

[31] N. Ravichandra and E. M. Botta, "Output space mapping for net-based debris capture," in *AIAA Scitech 2020 Forum*, 2020, p. 0717.

[32] C. M. Barnes and E. M. Botta, "A quality index for net-based capture of space debris," *Acta Astronautica*, vol. 176, pp. 455–463, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0094576520304100

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[34] A. Hill, A. Raffin, M. Ernestus, A. Gleave, A. Kanervisto, R. Traore, P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu, "Stable baselines," https://github.com/hill-a/stable-baselines, 2018.