

# Identification of parameters for tethered satellite system to emulate net-captured debris towing

Achira Boonrath, Tarunraj Singh, Eleonora M. Botta\*

Department of Mechanical and Aerospace Engineering, University at Buffalo, Buffalo, NY, United States

## ARTICLE INFO

### Keywords:

Net-based active debris removal  
Tethered satellite system  
Parameter identification  
Optimization  
Post-capture dynamics

## ABSTRACT

Net-based debris capture systems have a high potential for success in Active Debris Removal (ADR) missions. Simulation is an important tool in the analysis of the dynamics of nets, before experiments and actual missions are put in place. However, due to the large number of degrees of freedom required to model the dynamics of a net, in addition to nonlinearities, high-fidelity net-based ADR simulations are typically very computationally costly. This work focuses on the post-capture phase of a net-based ADR mission and aims to identify parameters of a lower-order model of the towed debris system such that it best matches the high-fidelity simulations with a full net. A model of a tethered satellite system with four sub-tethers is developed, and two optimization problems are formulated – minimizing the difference in dynamical quantities of interest – to perform the parameter identification task. The proposed system identification framework is first validated on a benchmark sub-tether model with known parameters, and then employed to minimize the difference in dynamics between the full-net and sub-tether debris towing simulations. The performances of the optimized solutions obtained from two proposed cost functions are compared both qualitatively and quantitatively. Overall, the proposed lower-order modeling and parameter identification framework demonstrate satisfactory performance in approximating the dynamics of the high-fidelity system.

## 1. Introduction

The number of objects in Earth's orbit increases yearly with the proliferation of satellite launches. Approximately 35,000 objects orbiting the Earth can be tracked by current space situational awareness abilities as of July 2024 [1]. A significant portion of these objects are space debris, which threaten existing and future space missions. Debris objects range from very large artifacts, such as defunct satellites and rocket upper stages, to small fragments generated from other space vehicles or debris. The collision of space debris with other spacecraft can create thousands of additional debris fragments – which are challenging to track – as shown by the accidental collision of Cosmos 2251 and Iridium 33 in 2009 [2]. To mitigate the dangers posed by space debris, ADR is considered an effective solution to ensure future space operations safety.

ADR technologies can be classified into contactless and contact-based methods [3,4]. Examples of contactless methods include the usage of ion beams shepherd satellites and laser-based systems to change the orbital velocity of debris; however such methods can only apply very small magnitude of force on debris and thus require long deorbiting mission durations [3,4]. Contact-based debris removal methods offer much reduced mission timescales compared to contactless

methods. Four phases are involved for envisioned ADR missions employing contact-based technologies: (i) launch and rendezvous, (ii) capture, (iii) detumbling, and (iv) removal [3]. In the first phase of the mission, the active chaser spacecraft is launched and travels to the target debris of interest. In the second phase, the chaser spacecraft and debris are connected via capture mechanisms. After the capture is complete, detumbling involves reducing the angular velocity of the debris [5–7], which could be present before the capture process or caused by the capture process itself. In the last stage of contact-based ADR, the debris is displaced to either burn in Earth's atmosphere or to an unpopulated graveyard orbit [5].

Many strategies for conducting the capture phase of ADR have been proposed [3,4]. Methods of particular interest for capturing space debris include the deployment of tethered nets, tethered robots, tethered harpoons, and robotic arms [8]. Tethered nets – relatively lightweight and allowing to capture large, rotating pieces of debris from a safe distance – have a high potential for success in conducting ADR missions [3,4,9]. Net-based debris capture technology has been partially demonstrated in orbit as a part of the 2018 RemoveDEBRIS ADR technology demonstration mission [10,11]. Nonetheless, debris towing

\* Corresponding author.

E-mail addresses: [achirabo@buffalo.edu](mailto:achirabo@buffalo.edu) (A. Boonrath), [tsingh@buffalo.edu](mailto:tsingh@buffalo.edu) (T. Singh), [ebotta@buffalo.edu](mailto:ebotta@buffalo.edu) (E.M. Botta).

<https://doi.org/10.1016/j.actaastro.2024.09.022>

Received 7 August 2024; Accepted 8 September 2024

Available online 17 September 2024

0094-5765/© 2024 IAA. Published by Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

was not demonstrated, and the dynamics of the chaser-net-target system must be further studied before performing large-scale net-based missions to remove large, uncooperative objects in space.

This work focuses on the immediate post-capture phase of net-based ADR. High-fidelity net-based simulations are typically very computationally costly and may be prohibitive for detumbling and deorbiting simulations of uncooperative debris, due to the large number of degrees of freedom (DoF) required to model the dynamics of the net, and due to the nonlinearities of contact dynamics and of tension force models [12, 13]. Only a few works – such as Ref. [14] – study the in-orbit post-capture dynamics of net-based ADR, but for a single simulation case, without addressing computational effort. To reduce the computational cost, a lower-order model can be employed, in which the net-wrapped debris is replaced by a TSS model in which debris is rigidly attached to four ST that extend from the main-tether (MT). Multiple researchers have studied this system configuration in the past [15–19]; however, the dynamics of the TSS model with STs was not compared to that of a towed debris system which incorporates net-based ADR components.

The goal of this work is to determine the parameters of the lower-order system model, such that its dynamics matches that of the net-based high-fidelity model as closely as possible. In particular, the proposed system identification framework aims to determine the length, stiffness, and damping properties of the STs and the portion of the mass of the net to be lumped in the connection point between the STs and the MT. Popular system identification methods – such as the auto-regressive with external input (ARX) and auto-regressive moving average with external input (ARMAX) [20] methods along with frequency-domain based techniques [21] – are effective when the system of interest is either linear, close to linear, or can be linearized. However, due to the complicated nonlinear tether tension and contact dynamics present, such techniques are foreseen to not perform well for the parameter identification task. Therefore, approaches utilizing global optimization algorithms are employed to identify the ST parameters in order to overcome modeling challenges. Two optimization problems are formulated, in which parameters are sought that minimize the difference in dynamical quantities of interest. The parameter identification tasks are performed using data from different simulations with varying initial debris rotation rates since rotation rates of space debris vary significantly between objects [22]; these rotation rates are made further uncertain by the limitations in current sensing capabilities.

The rest of the paper is organized as follows. Section 2 discusses the components and modeling of both the high-fidelity tether-net system and of the lower-order TSS system with four STs. Section 3 details the optimization framework for the parameter identification. Section 4 discusses the debris capture and framework setup, the benchmark scenario on which the proposed framework is validated, and the results of the parameter identification tasks using data obtained from high-fidelity net-based simulations. Lastly, Section 5 summarizes the findings of this work.

## 2. Systems description and modeling

### 2.1. Net-based debris capture

A cubic chaser spacecraft is assumed to be employed to transport the tether-net system close to the target. The net component of the system consists of a square net with a corner mass (CM) attached to each corner, and the MT links the net's central node to the chaser. At a specified distance from the debris, the chaser spacecraft ejects the CMs in a symmetric manner toward the debris [9,12]. Fig. 1 displays the net-based ADR system with the net mid-deployment towards the target. The debris of interest in this work is the second stage of the Zenit-2 launch vehicle (also known as the SL-16 rocket body), which is one of the most dangerous debris in orbit because of its large size and abundance [23]. After capture, a closing mechanism interlaced with the perimeter of the net is activated and closes around the target [12].

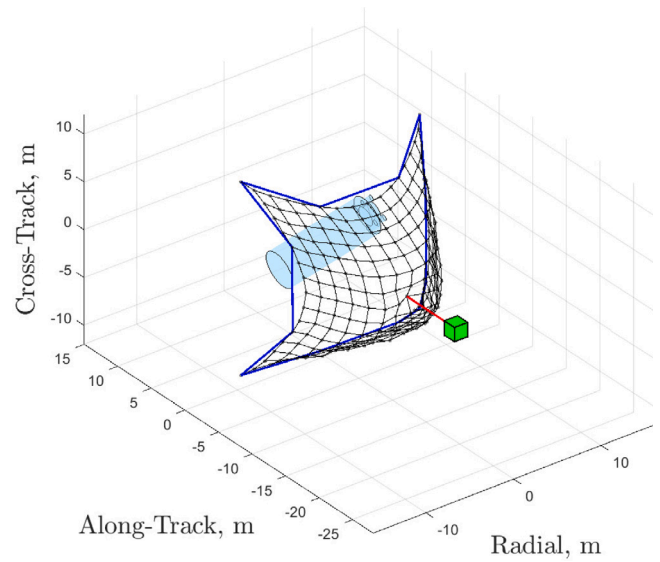


Fig. 1. Visualization of the net-based ADR system mid-deployment.

After a predetermined amount of time, the chaser's thrusters activate to change the coupled system's orbital velocity.

Net-based capture of debris is simulated employing a multi-body dynamics simulation platform as described in previous work [9,12]. Following the lumped-parameter modeling method, adjusted for implementation in a multi-body dynamics simulation framework, the mass of the net's threads is discretized into spherical rigid bodies (called *nodes*) placed at the net's thread intersections and at the CMs. The nodes (and all other rigid bodies in simulation) are given collision geometries and contact properties to represent contact dynamics with other objects. Normal contact forces and forces due to friction are computed via a modified Kelvin–Voigt model and the scaled-box friction model (i.e., an approximation based on Coulomb's friction), respectively [9,12,24]. The Kelvin–Voigt modeling method for viscoelastic materials typically represents the elasticity and energy dissipation within a structure; however, to represent the threads of the net, the nodes are joined via modified spring–damper elements that cannot withstand compression [9,12]. To ensure that the net remains around the debris after wrapping, threads laced through the four CMs and eight nodes along the net's perimeter are employed as a closing mechanism, which is modeled as in prior research on net-based capture [9,12]. The MT is modeled similarly to the net threads (i.e., via the modified Kelvin–Voigt modeling method); however, in this work, to simulate the free unreeling of a winch where the MT extends from, the stiffness and damping coefficients of the MT are set to zero throughout the flight of the net from the chaser to the debris, and are assigned their physical values at the net closing mechanism activation time.

Due to the longer simulation duration required for the study of post-capture debris towing, compared to research on capture dynamics by Botta et al. [9,12], the system is placed on orbit and subject to gravitational acceleration. For each  $\gamma$ -th body in simulation (i.e., chaser, debris, and nodes of the net), the two-body orbital model is assumed to compute the gravitational force  $F_{G,\gamma}$  applied:

$$F_{G,\gamma}(t) = -m_\gamma \mu \frac{\mathbf{r}_{\gamma/O}(t)}{\|\mathbf{r}_{\gamma/O}(t)\|^3} \quad (1)$$

where  $m_\gamma$  is the mass of each body,  $\mathbf{r}_{\gamma/O}$  is the inertial (orbital) position of the center of masses (CoM) of each body, and  $\mu = 3.986 \times 10^5 \text{ km}^3/\text{s}^2$  is the standard gravitational parameter of the Earth, and  $t$  is time. Eq. (1) is appended as an additional external force applied within the equation of motion for each node of the net, for the chaser, and for the target. While the equations of motion are not reported here for the sake of conciseness, the interested reader can refer to prior works [9,12].

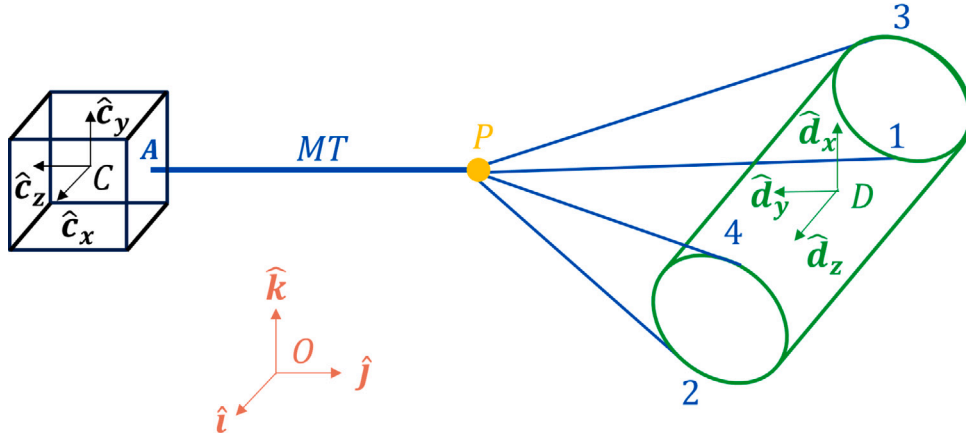


Fig. 2. TSS with four STs.

## 2.2. System modeling with sub-tethers

The lower-order TSS model consists of the components seen in Fig. 2. The chaser and debris – with their CoMs located at points  $C$  and  $D$ , respectively – are rigid bodies. In contrast, the connection point  $P$ , which connects the MT and the four STs (numbered 1 to 4), is modeled as a point mass. The MT is attached to the chaser at point  $A$  and the  $i$ th ST is attached to the debris at positions  $\mathbf{r}_{i/D}$  for  $i = 1, 2, 3, 4$  relative to the debris CoM. The three objects move in the inertial frame  $\mathcal{O} = \{O, \hat{i}, \hat{j}, \hat{k}\}$  where the origin of frame  $\mathcal{O}$  is located at the center of the Earth. Two additional, body-fixed frames can be defined:  $C = \{C, \hat{c}_x, \hat{c}_y, \hat{c}_z\}$  and  $D = \{D, \hat{d}_x, \hat{d}_y, \hat{d}_z\}$ , centered at the CoMs of the chaser and debris, respectively. In the following expressions, notation  ${}^{\mathcal{O}}(\cdot)$  denotes the inertial time derivative of the quantities of interests;  ${}^{\mathcal{O}}A^C$  represents the rotation matrix for transforming vector representation in the  $C$  frame to the  $\mathcal{O}$  frame.

Unlike the high-fidelity net simulation, the lower-order TSS model is simulated in MATLAB. As examined by Ref. [25], an identical single-tether TSS modeled and simulated in both Vortex Studio (with numerical integration performed by a semi-implicit first order solver [9,25]) and MATLAB (with numerical integration performed by ode45()) produced near-identical dynamics despite the difference in the numerical integration scheme; as such, each simulation framework is a viable substitute for the other.

**Chaser equations of motion:** For a rigid connection of the MT to the chaser, Euler's First Law is written for the chaser as:

$$m_C({}^{\mathcal{O}}\mathbf{a}_{C/O}(t)) = \mathbf{T}_{MT}(t) + \mathbf{F}_t(t) + \mathbf{F}_{G,C}(t) \quad (2)$$

where  $m_C$  is the mass of the chaser,  ${}^{\mathcal{O}}\mathbf{a}_{C/O}$  is the inertial acceleration of the CoM of the chaser,  $\mathbf{T}_{MT}$  is the MT tension applied to the chaser,  $\mathbf{F}_t$  is the applied thrust force, and  $\mathbf{F}_{G,C}$  is the gravitational force acting on the chaser (computed as per Eq. (1)). The thrust vector for the simulations in this work is computed as  $\mathbf{F}_t(t) = -F_t(t) \frac{{}^{\mathcal{O}}\mathbf{v}_{C/O}(t)}{\|{}^{\mathcal{O}}\mathbf{v}_{C/O}(t)\|}$ ; this is equivalent to applying thrust in the direction opposite to the orbital velocity. To prevent the sudden onset of high tension in the MT – which results in sudden high force applied onto the chaser and may lead to unsafe motion – the following piecewise function is utilized to define the thrust magnitude:

$$F_t(t) = \begin{cases} 0 & \text{if } t \leq t_{on} \\ F_{t,max} \left( \frac{t - t_{on}}{t_{slope}} \right) & \text{if } t_{on} < t < t_{on} + t_{slope} \\ F_{t,max} & \text{if } t_{on} + t_{slope} \leq t \end{cases}$$

with  $t_{on}$  being the time the thruster on the chaser turns on and  $t_{slope}$  being the chosen time it takes for the thrust magnitude to rise from zero

to its maximum value. With the dependency on time omitted, Euler's Second Law for the chaser is written as:

$$J_C \frac{d}{dt}({}^{\mathcal{O}}\boldsymbol{\omega}^C) = \boldsymbol{\tau}_C + \boldsymbol{\tau}_S - ({}^{\mathcal{O}}\boldsymbol{\omega}^C) \times J_C ({}^{\mathcal{O}}\boldsymbol{\omega}^C) \quad (3)$$

where  $J_C$  is the mass moment of inertia matrix of the chaser with respect to its principal axis,  $\boldsymbol{\tau}_C$  is the torque applied by the MT on the chaser,  $\boldsymbol{\tau}_S$  is the torque provided by the attitude controller, and  ${}^{\mathcal{O}}\boldsymbol{\omega}^C$  is the angular velocity of the chaser relative to the inertial frame. The following expression is used to compute the torque applied by the MT tension on the chaser :

$$\boldsymbol{\tau}_C = \mathbf{r}_{A/C} \times \mathbf{T}_{MT} \quad (4)$$

where  $\mathbf{r}_{A/C}$  is the position of the attachment point of the MT on the chaser, relative to the CoM of the chaser.

To compute the tension applied by the MT on the chaser, the following set of equations is utilized:

$$T_{MT} = \max(0, T_{MT}) \hat{\mathbf{u}}_{MT} \quad (5a)$$

$$l_{MT} = \mathbf{r}_{P/O} - (\mathbf{r}_{A/O} + \mathbf{r}_{A/C}) \quad (5b)$$

$$v_{MT,r} = {}^{\mathcal{O}}\mathbf{v}_{P/A} \cdot \hat{\mathbf{u}}_{MT} \quad (5c)$$

$${}^{\mathcal{O}}\mathbf{v}_{P/A} = {}^{\mathcal{O}}\mathbf{v}_{P/O} - ({}^{\mathcal{O}}\mathbf{v}_{C/O} + {}^{\mathcal{O}}\mathbf{v}_{A/C}) \quad (5d)$$

$${}^{\mathcal{O}}\mathbf{v}_{A/C} = {}^{\mathcal{O}}\boldsymbol{\omega}^C \times \mathbf{r}_{A/C} \quad (5e)$$

where  $T_{MT} = k_{MT}(l_{MT} - l_{MT,0}) + c_{MT}v_{MT,r}$  is the MT tension magnitude,  $k_{MT}$  is the stiffness coefficient of the MT,  $l_{MT}$  is the actual length of the MT,  $l_{MT,0}$  is the natural (unstretched) length of the MT,  $c_{MT}$  is the damping coefficient of the MT,  $\mathbf{r}_{P/O}$  is the inertial position of the connection point, and  $v_{MT,r}$  is the magnitude of the relative velocity of the connection point with respect to the MT attachment point on the chaser. To compute  $v_{MT,r}$ , the dot product is taken between the relative velocity of the connection point with respect to the MT attachment point on the chaser,  ${}^{\mathcal{O}}\mathbf{v}_{P/A}$ , and the unit vector pointing from the chaser's MT attachment point to the connection point,  $\hat{\mathbf{u}}_{MT} = l_{MT}/l_{MT}$ . It is worth noting that  ${}^{\mathcal{O}}v_{P/A} = \dot{l}_{MT}$  is the rate of change of the length of MT. The rotation matrix  ${}^{\mathcal{O}}A^C$  employs quaternions to quantify the angular state of the chaser and is expressed as:

$${}^{\mathcal{O}}A^C = \begin{bmatrix} q_{C,1}^2 + q_{C,2}^2 - q_{C,3}^2 - q_{C,4}^2 & 2(q_{C,2}q_{C,3} - q_{C,1}q_{C,4}) & 2(q_{C,2}q_{C,4} + q_{C,1}q_{C,3}) \\ 2(q_{C,2}q_{C,3} + q_{C,1}q_{C,4}) & q_{C,1}^2 - q_{C,2}^2 + q_{C,3}^2 - q_{C,4}^2 & 2(q_{C,4}q_{C,3} - q_{C,1}q_{C,2}) \\ 2(q_{C,2}q_{C,4} - q_{C,3}q_{C,1}) & 2(q_{C,3}q_{C,4} + q_{C,1}q_{C,2}) & q_{C,1}^2 - q_{C,2}^2 - q_{C,3}^2 + q_{C,4}^2 \end{bmatrix} \quad (6)$$

with  ${}^{\mathcal{O}}q^C = [q_{C,1}, q_{C,2}, q_{C,3}, q_{C,4}]^T$  representing the quaternion describing the attitude of the chaser with respect to the  $\mathcal{O}$  frame ( $q_{C,1}$  is the scalar component of the quaternion, the rest are its vector components).

To provide control over the orientation of the chaser during debris towing, the same sliding mode controller employed in Refs. [26,27] is utilized in this work. To quantify the difference between the desired and actual chaser attitude, the error quaternion  $q_e$  is computed as:

$$q_e = \begin{bmatrix} [{}^{\mathcal{O}}q_d^C]^T \cdot {}^{\mathcal{O}}q_d^C \\ Q_{C,d}^T {}^{\mathcal{O}}q_d^C \end{bmatrix} \quad (7)$$

where  ${}^{\mathcal{O}}q_d^C$  is the desired chaser quaternion and  $Q_{C,d}$  represent the quaternion kinematics matrix (taken from Ref. [28]) constructed using the values of  ${}^{\mathcal{O}}q_d^C$ :

$$Q_{C,d} = \begin{bmatrix} -q_{C,d,2} & -q_{C,d,3} & -q_{C,d,4} \\ q_{C,d,1} & -q_{C,d,4} & q_{C,d,3} \\ q_{C,d,4} & q_{C,d,1} & -q_{C,d,2} \\ -q_{C,d,3} & q_{C,d,2} & q_{C,d,1} \end{bmatrix} \quad (8)$$

For this work, the desired attitude of the chaser – and subsequently  ${}^{\mathcal{O}}q_d^C$  – is defined such that  $\hat{e}_z$  is along the  $-\hat{u}_{MT}$  direction,  $\hat{e}_x$  is along the  $-\hat{u}_{MT} \times \frac{r_{C/O}}{\|r_{C/O}\|}$  direction, and  $\hat{e}_y$  is along the  $(-\hat{u}_{MT}) \times (-\hat{u}_{MT} \times \frac{r_{C/O}}{\|r_{C/O}\|})$  direction. Next, the sliding manifold is defined as:

$$S = ({}^{\mathcal{O}}\omega^C - {}^{\mathcal{O}}\omega_d^C) + K \text{sgn}(q_{C,d,1}) Q_{C,d}^T {}^{\mathcal{O}}q_d^C \quad (9)$$

where the desired angular velocity is set as  ${}^{\mathcal{O}}\omega_d^C = \mathbf{0}$  and constant  $K > 0$ . As such, the control torque is computed as:

$$\begin{aligned} \tau_S = & ({}^{\mathcal{O}}\omega^C) \times J_C ({}^{\mathcal{O}}\omega^C) + \\ & J_C \left( 0.5 K \text{sgn}([{}^{\mathcal{O}}q_d^C]^T \cdot {}^{\mathcal{O}}q_d^C) (Q_{C,d}^T Q_{C,d} ({}^{\mathcal{O}}\omega_{C,d}^C) \right. \\ & \left. - Q_{C,d}^T Q_C ({}^{\mathcal{O}}\omega^C)) - G \text{sat}(S, \epsilon) \right) \end{aligned} \quad (10)$$

In Eq. (10), the function  $\text{sat}(S, \epsilon)$  is a saturation function which uses  $S$  and the boundary layer width  $\epsilon$  as input parameters (returning a  $3 \times 1$  vector);  $G$  is a positive definite gain matrix.

**Debris equations of motion:** For a rigid connection of the STs on the target, Euler's First Law is written for the debris as:

$$m_D ({}^{\mathcal{O}}a_{D/O}) = \sum_{i=1}^4 T_i + F_{G,D} \quad (11)$$

where  $m_D$  is the mass of the debris,  ${}^{\mathcal{O}}a_{D/O}$  is the inertial acceleration of the CoM of the debris,  $\sum_{i=1}^4 T_i$  is the sum of the tension forces applied by the STs, and  $F_{G,D}$  is the gravitational force acting on the debris. Euler's Second Law for the debris is written as:

$$J_D \frac{D}{dt} ({}^{\mathcal{O}}\omega^D) = \tau_D - ({}^{\mathcal{O}}\omega^D) \times J_D ({}^{\mathcal{O}}\omega^D) \quad (12)$$

where  $J_D$  is the mass moment of inertia matrix of the debris with respect to its principal axis,  $\tau_D$  is the net torque applied by the STs on the debris, and  ${}^{\mathcal{O}}\omega^D$  is the angular velocity of the debris relative to the inertial frame. The following expression is used to compute the torque applied by the STs on the debris:

$$\tau_D = \sum_{i=1}^4 r_{i/D} \times T_i \quad (13)$$

where  $r_{i/D}$  is the position of the  $i$ th ST attachment point relative to the CoM of the debris. To compute the tension applied by the  $i$ th ST, the following set of equations is utilized:

$$T_i = \max(0, T_i) \hat{u}_i \quad (14a)$$

$$l_i = r_{P/O} - (r_{D/O} + r_{i/D}) \quad (14b)$$

$$v_{i,r} = {}^{\mathcal{O}}v_{P/i} \cdot \hat{u}_i \quad (14c)$$

$${}^{\mathcal{O}}v_{P/i} = {}^{\mathcal{O}}v_{P/O} - ({}^{\mathcal{O}}v_{D/O} + {}^{\mathcal{O}}v_{i/D}) \quad (14d)$$

$${}^{\mathcal{O}}v_{i/D} = ({}^{\mathcal{O}}\omega^D) \times r_{i/D} \quad (14e)$$

where  $T_i = k_{ST}(l_{ST,i} - l_{ST,0}) + c_{ST}v_{i,r}$  is the tension magnitude of the  $i$ th ST,  $k_{ST}$  is the stiffness coefficient of the STs,  $l_{ST,i}$  is the actual length of the  $i$ th ST,  $l_{ST,0}$  is the natural (unstretched) length of the STs, and  $c_{ST}$  is the

damping coefficient of the STs. The variable  $v_{i,r}$  is the relative velocity magnitude of the connection point relative to the  $i$ th ST's attachment point on the debris, computed using the unit vector pointing from the  $i$ th attachment point to the connection point,  $\hat{u}_i = l_i / l_{ST,i}$ . The variable  ${}^{\mathcal{O}}A^D$  represents the rotation matrix for transforming a vector representation from the  $D$  frame to the  $\mathcal{O}$  frame. The matrix  ${}^{\mathcal{O}}A^D$  is expressed as:

$${}^{\mathcal{O}}A^D = \begin{bmatrix} q_{D,1}^2 + q_{D,2}^2 - q_{D,3}^2 - q_{D,4}^2 & 2(q_{D,2}q_{D,3} - q_{D,1}q_{D,4}) & 2(q_{D,2}q_{D,4} + q_{D,1}q_{D,3}) \\ 2(q_{D,2}q_{D,3} + q_{D,1}q_{D,4}) & q_{D,1}^2 - q_{D,2}^2 + q_{D,3}^2 - q_{D,4}^2 & 2(q_{D,4}q_{D,3} - q_{D,1}q_{D,2}) \\ 2(q_{D,2}q_{D,4} - q_{D,3}q_{D,1}) & 2(q_{D,3}q_{D,4} + q_{D,1}q_{D,2}) & q_{D,1}^2 - q_{D,2}^2 - q_{D,3}^2 + q_{D,4}^2 \end{bmatrix} \quad (15)$$

with  ${}^{\mathcal{O}}q^D = [q_{D,1}, q_{D,2}, q_{D,3}, q_{D,4}]^T$  representing the quaternion vector describing the attitude of the debris.

**Equation of motion for the connection point:** For a rigid connection of the MT and STs to the connection point, Euler's First Law for the connection point is written as:

$$m_P ({}^{\mathcal{O}}a_{P/O}) = F_{G,P} - T_{MT} - \sum_{i=1}^4 T_i \quad (16)$$

where  $m_P$  is the mass of the connection point, and all other quantities have been previously defined.

### 3. Optimization task for parameters selection

The objective of this work is to propose a framework to identify the parameters of the lower-order model such that it describes as well as possible the dynamics of the high-fidelity model in a number of possible debris capture conditions. In the following, the framework is described.

#### 3.1. Cost functions

To minimize the difference in the dynamics between the high-fidelity and lower-order models after parameter identification, the Root Mean Square Error (RMSE) between select dynamics quantities of interest of the target and chaser will be minimized. In the first cost function, the goal is to minimize the weighted sum of the RMSE of the relative position of the target with respect to the chaser –  $r_{D/C}$  –, of the debris angular velocity –  ${}^{\mathcal{O}}\omega^D$  –, and of the chaser angular velocity –  ${}^{\mathcal{O}}\omega^C$  – over  $N_s$  different simulations. These dynamics quantities are chosen for the RMSE computation because they relate to the safety and success of ADR missions, where the target is uncooperative. In addition, if these quantities are sufficiently similar between the models, it may be possible for controllers (e.g., chaser-debris relative distance controllers, debris detumbling controllers) to be designed, tuned, and tested using the lower order model, thus significantly reducing development time. The following quantities are defined which compare the dynamics of the chaser and debris between the lower-order and high-order simulations:

$$r_{D/C}^H = r_{D/O}^H - r_{C/O}^H \quad (17a)$$

$$r_{D/C}^L = r_{D/O}^L - r_{C/O}^L \quad (17b)$$

$$r_{\Delta} = r_{D/C}^L - r_{D/C}^H \quad (17c)$$

$$\omega_{C,\Delta} = ({}^{\mathcal{O}}\omega^C)^L - ({}^{\mathcal{O}}\omega^C)^H \quad (17d)$$

$$\omega_{D,\Delta} = ({}^{\mathcal{O}}\omega^D)^L - ({}^{\mathcal{O}}\omega^D)^H \quad (17e)$$

Here, superscripts  $(\cdot)^H$  and  $(\cdot)^L$  indicate that the particular quantities pertain to the high-fidelity simulation with net present and the lower-order simulation with the STs, respectively. Cost Function 1 is expressed mathematically as follows:

$$f_1(s) = \sum_{j=1}^{N_s} w_1 \text{RMSE}_{r,j} + w_2 (\text{RMSE}_{\omega_{D,j}} + \text{RMSE}_{\omega_{C,j}}) \quad (18)$$



where  $s$  is the set of parameters to be identified, and

$$\text{RMSE}_{r,j} = \sqrt{\sum_{g=1}^{N_t} \frac{\|r_{\Delta,j,g}\|^2}{N_t}} \quad (19a)$$

$$\text{RMSE}_{\omega_{D,j}} = \sqrt{\sum_{g=1}^{N_t} \frac{\|\omega_{D,\Delta,j,g}\|^2}{N_t}} \quad (19b)$$

$$\text{RMSE}_{\omega_{C,j}} = \sqrt{\sum_{g=1}^{N_t} \frac{\|\omega_{C,\Delta,j,g}\|^2}{N_t}} \quad (19c)$$

In the expressions above,  $w_1$  and  $w_2$  are the weights given to RMSE quantities regarding relative position and angular velocities, respectively, to ensure that terms contribute similarly to the cost. The RMSE quantities are summed over  $N_s$  simulations – each with  $N_t$  total timesteps – to allow the optimization task (which will be presented later in this section) to consider different debris angular velocities in towing missions, thus preventing over-fitting parameters to a single scenario. The number of timesteps in each simulation is computed as  $N_t = t_{\text{sim}}/(\Delta t_{\text{save}})$  where  $t_{\text{sim}}$  is the total simulated time and  $\Delta t_{\text{save}}$  is the output saving timestep given to the numerical integrator (set to be identical for both the lower-order and high-fidelity simulations).

In the second cost function, the goal is to minimize the RMSE of the tension in the MT between high-fidelity and lower-order models. This quantity is chosen because the tension experienced by the MT is a function of the relative states of the connection point and of the debris with respect to the chaser. Enforcing the MT tension of high-fidelity and lower-order models to be as similar as possible may indirectly enable the states of the debris and of the chaser to be as similar as possible. Additionally, the MT tension is related to mission safety, as instances of slack MT (i.e., with zero tension) during debris towing may lead to unsafe system dynamics [29]. Therefore, if the MT tension dynamics are similar between the models, safety analysis may be performed using the lower-order TSS model. Cost Function 2 is expressed mathematically as follows:

$$f_2(s) = \sum_{j=1}^{N_s} \text{RMSE}_{T,j} \quad (20)$$

with

$$\text{RMSE}_{T,j} = \sqrt{\sum_{g=1}^{N_t} \frac{(T_{\text{MT},j,g}^H - T_{\text{MT},j,g}^L)^2}{N_t}} \quad (21)$$

where  $T_{\text{MT},j}^H$  and  $T_{\text{MT},j}^L$  are the magnitudes of the tension in the MT for the  $j$ th simulation with the high-fidelity and lower-order models, respectively.

### 3.2. Decision variables and optimization problem formulation

In this work, four parameters for the ST model were selected to be identified by the framework (i.e.,  $s = [s_1, s_2, s_3, s_4]^T$ ). The first three parameters relate to the physical properties of the STs:  $k_{\text{ST}}$ ,  $l_{\text{ST},0}$ , and  $c_{\text{ST}}$ , which are assumed to be the same of all STs. Additionally, the allocation of the mass of the net must also be determined, as the mass of the net is distributed in the high-fidelity model and can be entirely lumped with the target, the connection point, or somewhere in between in the lower-order model. The decision variables take on the following relationship with the physical parameters of the STs and connection point mass:

$$s_1 = \log_{10}(k_{\text{ST}}) \quad (22a)$$

$$s_2 = l_{\text{ST},0} \quad (22b)$$

$$s_3 = \log_{10}(c_{\text{ST}}) \quad (22c)$$

$$s_4 = m_P/m_{\text{net}} \quad (22d)$$

where  $m_{\text{net}}$  is the total mass of the net in the high-fidelity simulation. The base-10 logarithm of the quantity of interest for Eq. (22a) and (22c) are taken to allow the optimization algorithm to search for the best-optimized solution over a large range of  $k_{\text{ST}}$  and  $c_{\text{ST}}$ , since it is foreseen these the ranges far exceed those of the other two parameters.

With these stated decision variables, the optimization problem to determine the ST and connection point parameters is mathematically expressed as:

$$\min_s f_k(s)$$

$$\text{where : } s = [s_1, s_2, s_3, s_4]^T \quad (23)$$

$$s_{\xi,\text{LB}} \leq s_{\xi} \leq s_{\xi,\text{UB}}$$

where  $k = 1 \vee 2$  indicates which cost function is being optimized for. The shapes of the cost functions are unknown but are possibly nonlinear, nonconvex, and multimodal with respect to the decision variables. Due to this, the optimization problem can be classified as a nonlinear programming problem. Additionally, the gradients of the cost functions are unknown and expensive to accurately estimate via finite difference approximations. Therefore, global optimization algorithms that do not rely on gradients are preferred for solving the problem.

Two global optimization solvers are chosen for this work. The first solver is the Mesh Adaptive Direct Search (MADS) algorithm, which is a derivative-free optimization algorithm, thus not requiring the gradient of the cost function to be known; it is available in MATLAB via the `patternsearch()` function with the `Algorithm=nups-mads` option. The algorithm has been successfully used in many black-box optimization problems with scientific and engineering applications [30]. However, it requires an Initial Guess (IG) of the decision variables to be given at the initialization of the optimization. Therefore, a second derivative-free global optimization solver that does not require initial design guesses is also utilized to cross-validate the solutions obtained from the MADS algorithm. The second optimization algorithm employed is the Particle Swarm Optimization (PSO) algorithm, which also has been used successfully to solve many scientific and engineering optimization problems [31]. MATLAB's implementation of the PSO algorithm via the `particleswarm()` function is employed for this work. Both the MADS and PSO algorithms in MATLAB can be used with parallel CPU computation to speed up the optimization task.

## 4. Simulation and optimization setup and parameter identification results

This section will first describe the parameters of the net-based ADR system, simulation initial conditions, and the initialization of the optimization framework. Then, the proposed system identification framework will be validated on a benchmark TSS model with known STs parameters. Lastly, the results of parameter identification utilizing high-fidelity simulation data will be discussed.

### 4.1. Net-based debris capture scenario and lower- and higher-fidelity post-capture simulations

Data from 20 unique high-fidelity simulations were acquired for parameter identification and subsequent validation on unseen simulations. The collected data is split into a training dataset (utilized for parameter identification) and a validation dataset (utilized for parameter evaluation on unseen data); for both datasets  $N_s = 10$ . The system parameters for all 20 simulations are identical and can be seen in Table 1, where quantities with  $[\cdot]_C$  and  $[\cdot]_D$  are representations of vectors in coordinates of the  $C$  and  $D$  frames, respectively. Except for the natural length of each net thread section,  $l_{\text{net},0} = 1.5714$  m, and for the direction of deployment of the net, set to be along the  ${}^0v_{C/O}$  direction, the net present in this work – with a total mass of  $m_{\text{net}} = 2.7326$  kg – possesses the same physical properties and launch parameters as that of the net utilized to capture the second stage of the

**Table 1**  
System parameters for High-Fidelity and Lower-Order simulations.

Variables	Value
Chaser Mass $m_C$ , kg	1600
Nominal Debris Mass $m_{D,nom}$ , kg	9000
Chaser Side Length $L_C$ , m	1
Debris Length $L_D$ , m	11
Debris Diameter $d_D$ , m	3.9
Chaser Inertia Matrix $J_C$ , kg m <sup>2</sup>	diag(266.67, 266.67, 266.67)
Debris Inertia Matrix $J_D$ , kg m <sup>2</sup>	diag(94880, 94880, 46295.5)
Chaser Attachment Point $[r_{A/C}]_C$ , m	$[0, 0, -0.5]^T$
Debris Attachment Point 1 $[r_{1/D}]_D$ , m	$[-1.9500, 0, -5.7170]^T$
Debris Attachment Point 2 $[r_{2/D}]_D$ , m	$[-1.9500, 0, 5.2830]^T$
Debris Attachment Point 3 $[r_{3/D}]_D$ , m	$[1.9500, 0, -5.7170]^T$
Debris Attachment Point 4 $[r_{4/D}]_D$ , m	$[1.9500, 0, 5.2830]^T$
Main-Tether Stiffness $k_{MT}$ , N/m	3000
Main-Tether Damping $c_{MT}$ , Ns/m	300
Main-Tether Natural Length $l_{MT,0}$ , m	15

**Table 2**  
High fidelity simulations initial conditions.

Variables	Value
$[r_{C/O}]_O$ , m	$[5.199371, -1.107437, -4.882531]^T \times 10^6$
$[v_{C/O}]_O$ , m/s	$[4.65748, 4.166098, 4.021629]^T \times 10^3$
$q^C$ , -	$[0.50637, 0.6281, -0.5865, -0.0709]^T$
$[\omega^C]_C$ , deg/s	$[0.0, 0.0, 0.0]^T$
$[r_{D/O}]_O$ , m	$[5.199384, -1.107427, -4.882521]^T \times 10^6$
$[v_{D/O}]_O$ , m/s	$[4.65748, 4.166098, 4.021629]^T \times 10^3$
$q^D$ , -	$[0.38344, 0.4526, 0.7958, 0.1218]^T$
$[\omega^D]_D$ , deg/s	$  \omega^D(0)   \cdot [0.0, -1.0, 0.0]^T$

Zenit-2 rocket in previous works by the authors [12,32]. The closing mechanism activation time is fixed at  $t = 15$  s, and the thrust-on time is set as  $t_{on} = 25$  s. The chaser thrust maximum magnitude is  $F_{t,max} = 200$  N and  $t_{slope} = 50$  s. The sliding mode control parameters are  $\epsilon = 0.001$ ,  $K = 24$ , and  $G = \text{diag}(1,1,1)$ . The total time in-simulation for all high-fidelity simulations is 300 s (i.e., 25 s of net deployment and target capture, and 275 s of debris towing).

The initial conditions for all high-fidelity simulations can be seen in Table 2. The net capture of the Zenit-2 upper stage happens on an 840 km altitude circular orbit, approximately where one such debris is located [23]. The chaser is initially positioned such that  $r_{D/C}$  is along the  ${}^O v_{C/O}$  direction with its magnitude  $\|r_{D/C}\| = 19.75$  m. For this work, it is assumed that the chaser is able to rendezvous with the debris and achieve  ${}^O v_{D/C} = 0$  at the instance of net ejection. It is assumed that the  $\hat{d}_z$  and  $\hat{d}_y$  axes of the target are initially aligned with the  $r_{D/O}$  and  $-{}^O v_{D/O}$  directions, respectively. In accordance with previous works by the authors [12,32], the upper stage of Zenit-2 is set to initially rotate about its major axis of inertia corresponding to  $\hat{d}_y$ , as seen in Fig. 2. The magnitude of the initial debris angular velocity varies in each simulation; its values are chosen to be in the range of observed debris rotation rates [33] and are listed in Table 3. The values in Table 3 were uniformly sampled throughout the considered interval of debris rotation rates (i.e., 9.60 to 29.01 deg/s). Note that, due to the difference in the initial target angular velocity, the orientation of the target relative to the net at the time of closing mechanism activation is different between all high-fidelity simulations; this can be seen in Fig. 3, where the system is visualized at  $t = 15$  s for simulations with  $||\omega^D(0)|| = 9.60$  deg/s and  $||\omega^D(0)|| = 29.01$  deg/s.

The comparisons of the dynamics between the high-fidelity and lower-order models are set to begin at  $t = 25$  s (i.e., chaser thrust activation time) until the end of the high-fidelity simulations. Correspondingly, the total simulation time for all lower-order simulations is  $t_{sim} = 275$  s. The states of the debris and chaser at  $t = 25$  s of the high-fidelity simulations are used as initial conditions for the lower-order model simulations. The initial position and velocity of the connection point are set to be the same as the initial position and velocity of the node that connects the MT to the net at 25 s for the corresponding

**Table 3**  
Initial Debris Angular Velocity Magnitude for each Dataset.

Simulation #	Training $  \omega^D(0)  $ , deg/s	Validation $  \omega^D(0)  $ , deg/s
1	10.71	9.60
2	11.69	12.67
3	14.80	13.63
4	15.57	14.90
5	16.55	17.52
6	19.77	18.64
7	20.44	21.42
8	23.38	22.40
9	28.25	26.29
10	29.01	27.26

high-fidelity simulation. With the exception of the mass of the debris, in the lower-order model, both chaser and debris are given the same parameters as the high-fidelity simulations. The mass of the debris utilized by the lower-order TSS simulations is computed as  $m_D = m_{D,nom} + m_{net}(1 - s_4)$  to account for the proportion of the net mass that is allocated to the debris, where  $s_4$  is uniquely assigned to each lower-order simulation by the optimization solver. As the mass of the net is much smaller than that of the debris, it is assumed that the portion of the net mass that is allocated to the debris does not affect the mass moment of inertia (i.e.,  $J_D$  is the same for all simulations). The mass of the connection point is computed as  $m_P = m_{net}s_4$ . Due to the rocket's nozzles, the CoM of the Zenit-2 s stage is not located at the geometric center of the debris [12]. This is reflected via the asymmetry in the  $\hat{d}_z$ -components of the ST attachment locations, indicated in Table 1. Nonetheless, it is chosen that the  $\hat{d}_x$ -components of the ST attachment locations are symmetric to each other, and the  $\hat{d}_y$ -components are all set to be equal to zero.

#### 4.2. Optimization setup

To determine the weights  $w_1$  and  $w_2$  for  $f_1(s)$ , a sample simulation with the STs model with  $s = [3, 7.5, 1, 1]^T$  was performed in MATLAB with the same parameters as the high-fidelity Validation Simulation 5 (see Table 3). The lower-order simulation took approx. 1 min to complete while the high-fidelity simulation took approx. 4.5 h (both utilizing a computer with Intel(R) Core(TM) i9-9900 CPU @ 3.10 GHz processor). For this single simulation, it was found that  $\text{RMSE}_{r,j} = 1.6411$  m and  $\text{RMSE}_{\omega_{C,j}} = 0.04212$  rad/s,  $\text{RMSE}_{\omega_{D,j}} = 0.04699$  rad/s. Therefore, to enable the relative position and angular velocities RMSE to have a similar impact on the overall computed cost, it was chosen that  $w_1 = 1$  and  $w_2 = 40$ .

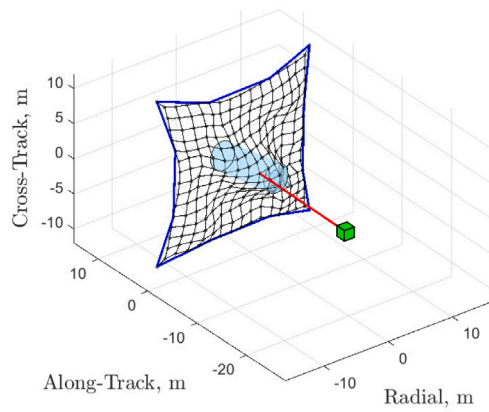
The bounds of the decision variables concerning the STs are chosen based on heuristic estimates of where the values are expected to be, while for the portion of net mass lumped in the connection point, the value ranges from 1 (i.e., the entire mass of the net is lumped into the connection point) to 0.1 (i.e., 10% of the mass of the net is lumped into the connection point, the rest with the debris). A non-zero minimum value is chosen for  $s_4$  since  $s_4 = 0$  is equivalent to the connection point possessing no mass, which is not physical. The bounds for the decision variables are as follows:

$$1 \leq s_1 \leq 3.5 \quad (24a)$$

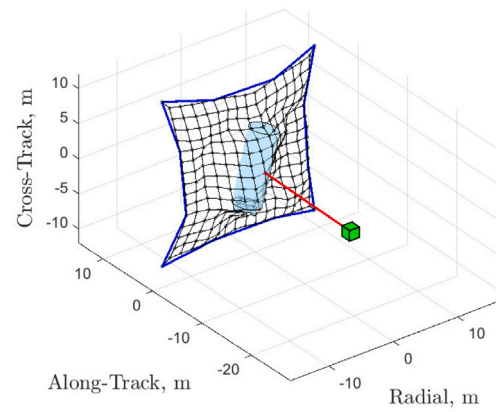
$$5 \leq s_2 \leq 11 \quad (24b)$$

$$0 \leq s_3 \leq 3.5 \quad (24c)$$

$$0.1 \leq s_4 \leq 1 \quad (24d)$$



a Full-Net Simulation with  $\|\omega^D(0)\| = 9.60$  deg/s,  $t = 15$  s



b Full-Net Simulation with  $\|\omega^D(0)\| = 29.01$  deg/s,  $t = 15$  s

Fig. 3. Full-net systems at the moment of closing mechanism activation for (a) Slowest debris rotation case and (b) Fastest debris rotation case.

#### 4.3. Parameter identification benchmark scenario

To evaluate the performance of the proposed setup for parameter identification at converging to the optimal solution, a benchmark scenarios was evaluated using  $f_1(s)$  and  $f_2(s)$ . In the benchmark scenarios, 10 lower-order TSS simulations with STs present were run utilizing  $s = [2.0, 7.0, 2.0, 1.0]^T$ , the ST model's parameters and initial conditions in Section 4.1, and the training set initial angular speeds of the debris (see Table 3). For the proposed setup for parameter identification to show satisfactory performance, both cost functions should converge to solutions close to the chosen decision variables. Each cost function was optimized with both MADS and PSO solvers in the benchmark scenario. The MADS solver termination criteria were set to be 250 optimization iterations or MeshTolerance=1e-6. The IG  $s_0 = [1, 6, 3, 0.25]^T$  was heuristically chosen to initialize the MADS-based optimization. The particleswarm() function settings were chosen to be SwarmSize=40, FunctionTolerance=1e-06, and MaxStallIterations=20. For both solvers, all other settings were left to their default values present within MATLAB 2023a.

The convergence history plots, converged cost function values, and converged decision variables for the benchmark scenario are seen in Fig. 4, Table 4, and Table 5, respectively. As per Fig. 4 and Table 4, it can be seen that, at their final optimization iterations, both cost functions converged to small values relative to their first iterations, with both optimizers. For example, MADS- and PSO-based optimization of  $f_1(s)$  had cost values of approx. 47 and 11 in their first solver iterations, respectively (see Fig. 4(a)); in the end, both converged to cost values of less than 1. Similar statements regarding the convergence histories can be made about  $f_2(s)$  in Fig. 4(b). In Table 5, it is also observed that all of the converged decision variables are very close to the truth, except for  $s_4$  in the MADS optimization cases; this demonstrates the ability of the framework to converge to near the true solutions.

For both cost functions, the converged PSO  $s_1$  to  $s_3$  decision variables are slightly closer to the truth than the corresponding values from MADS. At the same time, PSO was able to converge very close to the true value of  $s_4 = 1$ . Thus, it is expected that, in parameter identification tasks utilizing data obtained from high-fidelity net simulations, PSO-based solutions will be more favorable compared to MADS-based solutions.

#### 4.4. Results of ST TSS parameter identification utilizing high-fidelity simulation data

This section presents the results of parameter identification performed via minimizing  $f_1(s)$  and  $f_2(s)$  per the optimization problem

Table 4

Converged cost function values for the benchmark scenario.

Optimization #	Converged $f_1(s)$	Converged $f_2(s)$
MADS	0.4960	3.4684
PSO	$6.3958 \cdot 10^{-5}$	0.0217

Table 5

Converged decision variables for the benchmark scenario.

Optimization #	$s_1$	$s_2$	$s_3$	$s_4$
MADS $f_1(s)$	1.9959287	6.9976071	1.9901418	0.4832296
PSO $f_1(s)$	2.0000003	7.0000015	1.9999996	0.9999957
MADS $f_2(s)$	1.9971665	7.0066112	1.9987416	0.8285154
PSO $f_2(s)$	2.0000035	7.0000049	2.0000158	0.9999974

Table 6

Initial guesses given to the MADS optimization solver.

Optimization #	$s_1$	$s_2$	$s_3$	$s_4$
MADS 1	3	9	1	0.5
MADS 2	2	9	1	0.5
MADS 3	3	8	1	0.5
MADS 4	3	9	2	0.5
MADS 5	3	9	1	0.75

defined in Eq. (23). The high-fidelity simulation training dataset (see Table 3) is used with the defined identification framework. Then, through the use of the validation dataset, the difference in the dynamics of the high-fidelity model and the lower-order model with the optimized parameters are evaluated for unseen scenarios.

Since the choice of initial decision variables may affect the solution to which the MADS optimization algorithm will converge, five different IGs (seen in Table 6) were employed for the MADS-based parameter identification. The IGs were heuristically chosen such that different locations within the decision space were utilized to start the optimization. Similarly, as PSO is stochastic (i.e., each particleswarm() function call may return different results), PSO-based optimizations were performed 3 times. Both solvers used the same settings as described in Section 4.3.

The optimization convergence history for  $f_1(s)$  and  $f_2(s)$  optimizations (5 MADS and 3 PSO runs each) is provided in Fig. 5(a) and 5(b), respectively. From Fig. 5(a), it can be seen that the MADS-based solutions started the optimization with higher  $f_1(s)$  values compared to PSO-based solutions (i.e., between approx. 65–77 compared to between approx. 56–58, respectively). This can be explained by the difference in how the two algorithms start their solution search processes: In the

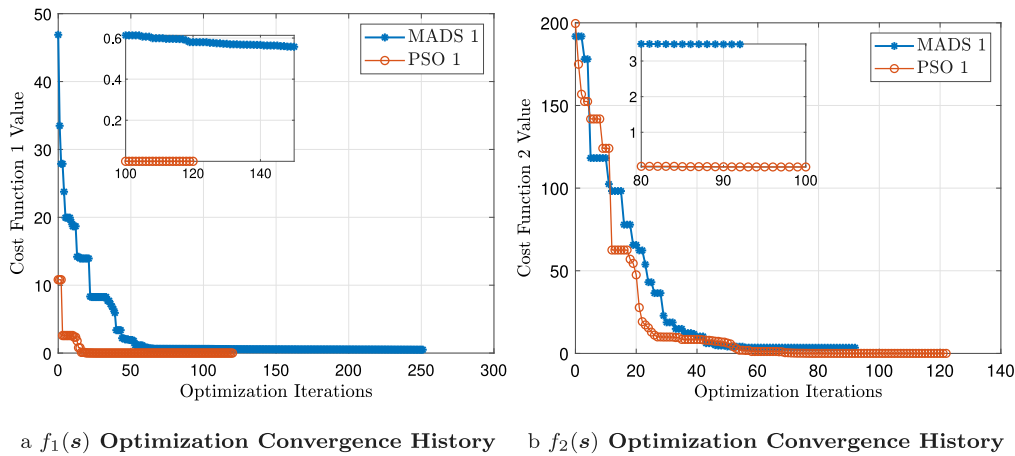


Fig. 4. Convergence history for the benchmark scenario.

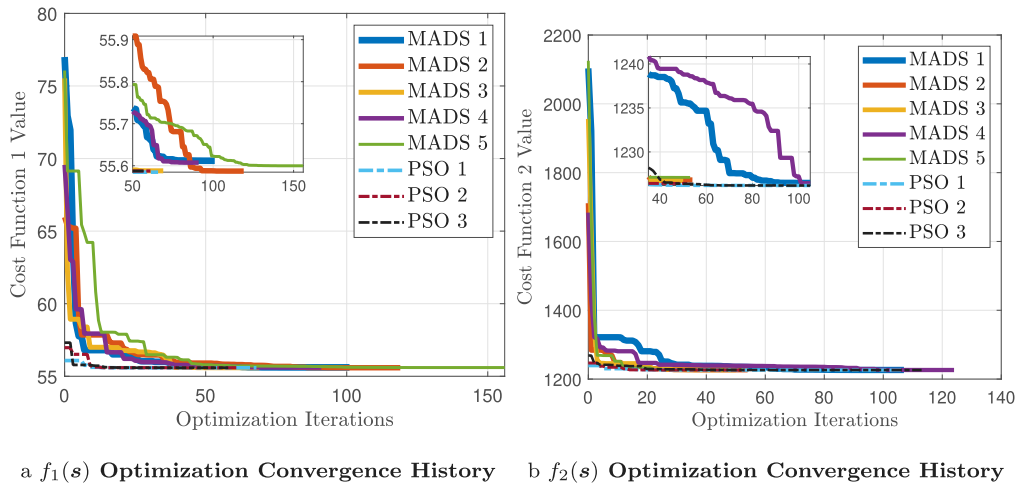


Fig. 5. Convergence history of parameter identification using optimization with the training dataset.

**Table 7**  
Converged cost function values and decision variables using  $f_1(s)$ .

Optimization #	Training $f_1(s)$	$s_1$	$s_2$	$s_3$	$s_4$
MADS 1	55.6119	1.4417	6.8448	2.1176	0.5616
MADS 2	55.5872	1.4440	6.8627	2.1140	0.1030
MADS 3	55.5879	1.4150	6.7240	2.0274	0.1111
MADS 4	55.6074	1.4401	6.8374	2.1231	0.5000
MADS 5	55.5998	1.4417	6.8457	2.1204	0.4277
PSO 1	55.5847	1.4422	6.8510	2.1060	0.1000
PSO 2	55.5875	1.4143	6.7198	2.0252	0.1000
PSO 3	55.5875	1.4141	6.7198	2.0254	0.1000

**Table 8**  
Converged cost function values and decision variables using  $f_2(s)$ .

Optimization #	Training $f_2(s)$	$s_1$	$s_2$	$s_3$	$s_4$
MADS 1	1226.48	2.6998	8.8563	2.8286	0.1731
MADS 2	1226.81	2.6719	8.8024	2.7712	0.4018
MADS 3	1226.73	2.6914	8.8299	2.8088	0.3957
MADS 4	1226.35	2.6931	8.8568	2.8059	0.1477
MADS 5	1227.10	2.7157	8.7958	2.8151	0.6106
PSO 1	1226.18	2.6822	8.8304	2.7865	0.1533
PSO 2	1226.42	2.6960	8.8551	2.8143	0.2521
PSO 3	1226.17	2.6786	8.8301	2.7821	0.1561

first iteration of each MADS-based optimization, only the given IG is evaluated. In contrast, 40 sets of decision variables sampled throughout the search space (i.e., one per each PSO particle) were evaluated in the first iteration of each PSO-based optimization. As a result, it is highly probable that the initial value in the PSO-based cases (i.e., the best  $f_1(s)$  found from evaluating the 40 sampled decision variables) is lower than the initial value of MADS-based cases (each value is the  $f_1(s)$  value found using the corresponding IG). MADS-based optimizations also took more iterations to reach the termination condition compared to PSO-based optimizations (see the zoomed-in plot of Fig. 5(a)). Similar observations are made regarding the convergence history of  $f_2(s)$  in Fig. 5(b), where MADS-based solutions start the optimization with higher cost compared to PSO-based solutions (i.e., approx. 1650–2150 compared to approx. 1250–1270). However, with  $f_2(s)$ , PSO-based optimization does not necessarily converge using less iterations than MADS-based optimization. For example, the PSO 1 optimization of  $f_2(s)$  took over 100 iterations to finish, while its MADS 5 optimization only took approx. 50 iterations.

Tables 7 and 8 display the training dataset’s converged optimized cost function values and decision variables with  $f_1(s)$  and  $f_2(s)$ , respectively. It can be seen that  $f_1(s)$  and  $f_2(s)$  return their smallest values of 55.5847 and 1226.1, respectively, with the first PSO run. Reflecting the observations already made in the benchmark scenario in Section 4.3, optimized decision variables obtained from the PSO solver result in smaller cost function values than MADS-based results. The difference between the minimum and maximum converged cost function values



**Table 9**

Best ST and connection point parameters for each cost function.

Parameter	From $f_1(s)$ PSO 1	From $f_2(s)$ PSO 1
$k_{ST}$ , N/m	27.6822	481.0608
$l_{ST,0}$ , m	6.8510	8.8304
$c_{ST}$ , Ns/m	127.643	611.6458
$m_p$ , kg	0.27326	0.4189

**Table 10**

Cost function values for validation dataset.

Optimization #	Validation $f_1(s)$	Validation $f_2(s)$
MADS 1	50.2055	1301.75
MADS 2	50.2248	1296.49
MADS 3	49.9488	1300.29
MADS 4	50.2130	1301.70
MADS 5	50.2141	1299.75
PSO 1	50.1917	1299.46
PSO 2	49.9409	1302.19
PSO 3	49.9430	1299.21

is 0.0272 (i.e., 0.0489% difference) for  $f_1(s)$  and 1.0 (i.e., 0.0816% difference) for  $f_2(s)$ . This indicates that, for each cost function, all found solutions resulted in very similar converged values.

Comparing the different converged decision obtained with  $f_1(s)$  in Table 7, it can be seen that all solutions converged to values close to  $s_1 = 1.4$ ,  $s_2 = 6.8$ , and  $s_3 = 2.1$ ; however,  $s_4$  either converged to approx. 0.1 or to approx. 0.5. Comparing the different converged decision obtained with  $f_2(s)$  in Table 8, it can be seen that all solutions converged to values around  $s_1 = 2.7$ ,  $s_2 = 8.8$ , and  $s_3 = 2.8$ ; however, unlike the  $f_1(s)$  optimization cases,  $s_4$  of  $f_2(s)$  converged to many values between approx. 0.15 and 0.61. The overall small influence of varying  $s_4$  on the converged cost function values indicates that the particular choice of the connection point mass has a minor effect on the overall dynamics of the system. Nonetheless, Tables 7 and 8 demonstrate that for the parameter identification task, similar solutions can be found irrespective of the choice of the optimization solver and of IGs for the MADS solver, with either cost function. Using Eq. (22) to convert the decision variables to physical parameters of the STs and connection point, the best cost function values are displayed in Table 9.

Next, the validation dataset in Table 3 is utilized to examine if the identified parameters can be generalized to simulations unseen by the identification framework. Table 10 reports the cost function values obtained utilizing the parameters identified from the training dataset together in the simulations defined as the validation dataset. It is worth noting that all the values for  $f_1(s)$  (of approx. 50.0) are lower than their corresponding values in the training data set (of approx. 55.6). At the same time, for  $f_2(s)$ , all the values of the validation dataset (of approx. 1300) are overall greater than their values in the training data set (of approx. 1226). Nonetheless, the cost values are overall similar between training and validation datasets for both cost functions. This demonstrates that the obtained solutions can be generalized to unseen debris towing scenarios, with  $f_1(s)$  parameters yielding better results in the validation task than  $f_2(s)$  ones. Additionally, the similar cost values between training and validation datasets indicate that the employment of 10 training simulations is sufficient to perform TSS parameter identification within the ADR scenarios considered for this work.

To first qualitatively compare the pose of the high-fidelity and lower-order systems at different instances in a sample simulation, Fig. 6 displays both systems as viewed from the local vertical, local horizontal (LVLH) frame of the debris at  $t = 90.0$  s,  $t = 180.0$  s and  $t = 275.0$  s for Validation Simulation 5. The parameters used for the lower-order model were taken from the optimization runs using  $f_1(s)$  with its PSO 1 and  $f_2(s)$  also with its PSO 1 (see Table 9). Here, it should be noted that the lower-order simulations with the best parameters from  $f_1(s)$  and  $f_2(s)$  took approx. 1.75 and 3.45 min to compute,

respectively; these simulations are approx. 135 and 78 times faster, respectively, to compute than the corresponding high-fidelity simulation. From Fig. 6(a)–(c), it can be seen that the positions and orientations of the chaser and target are similar for all simulations soon after the simulations begin. At  $t = 180$  s, while the positions and orientations of the chaser and the positions of the target are similar among the simulations (see Fig. 6(d)–(f)), there are noticeable target attitude differences: the major axis of the debris in high-fidelity simulation is closer to aligning with the LVLH radial direction (see Fig. 6(d)) compared to those in the lower order simulations (see Fig. 6(e)–(f)). Similar comparisons can be made regarding the positions and orientations of the chaser and target in Fig. 6(g)–(i), which display the systems at the end of the simulations. It is interesting to note that the difference between the debris attitude of the high-fidelity and lower-order systems grew over time, as can be seen from comparing debris orientation within Fig. 6(a)–(c) and within Fig. 6(g)–(i).

Figs. 7 and 8 – utilizing the same datasets – display the variations of the dynamics quantities in time for the high-fidelity and lower-order systems utilizing  $f_1(s)$  and  $f_2(s)$ , respectively. In both lower-order simulations, the MT tensions approximately match the quasi-steady-state values of the MT tension of the high-fidelity simulation (see Fig. 7(a) and 8(a)). Furthermore, the lower-order model MT tension from the simulation using values obtained via  $f_2(s)$  demonstrates small oscillations towards the end of the simulation similar to those observed in the high-fidelity simulation. The same lower-order simulation also demonstrates a tension spike at the same time (approx. 25 s) and with a similar magnitude as the high-fidelity model. This matching is a consequence of  $f_2(s)$  minimizing the difference between the MT tension values of the two models. In contrast, it is observed that the MT tension from the lower-order simulation using values obtained via  $f_1(s)$  displays highly damped oscillation without any spikes; this is a product of the high  $c_{ST}/k_{ST}$  value obtained with  $f_1(s)$  (see Table 9).

Within the lower plots of Fig. 7(a) and 8(a), it can be observed that the maximum value of the displacement magnitude between chaser and target CoM (i.e.,  $\|r_{D/C}^L - r_{D/C}^H\|$ ) is approx. 0.75 m for  $f_1(s)$  and approx. 1.2 m for  $f_2(s)$ . The better performance of  $f_1(s)$  at matching the positions is to be expected as  $f_1(s)$  directly penalizes larger differences in the chaser-debris relative positions across the two models. Nonetheless, the displacement magnitude between chaser and target for both  $f_1(s)$  and  $f_2(s)$  remains relatively small compared to their initial relative distance of 19.75 m. From Fig. 7(b) and 8(b), it can be seen that the maximum difference in the chaser position throughout simulations is approx. 1.1 m and 1.7 m, and that the maximum difference in the target debris inertial position throughout simulations is approx. 0.2 m and 0.3 m for  $f_1(s)$  and  $f_2(s)$ , respectively. Compared to orbital position magnitudes, in the order of  $10^6$  m, these differences in position are minimal.

The effect of system modeling is more important on the attitude motion, as can be observed in Fig. 7(c)–(d) and 8(c)–(d). In the high-fidelity simulation, the chaser's sliding mode attitude controller – which was initially designed for a single satellite [27] – struggles to maintain the desired chaser attitude and angular velocity (reported in the upper plots of Fig. 7(c) and 8(c)). The angular velocity components of the chaser oscillate about zero with significant amplitude throughout the high-fidelity simulation, albeit with higher frequency up to approx. 70 s. On the other hand, with both sets of identified parameters, the chaser's sliding mode attitude controller can maintain near-zero chaser angular velocity in the lower-order simulations after approx. 20 s (see the zoomed-in plots of Fig. 7(c) and 8(c)). It is noted, however, that the controller in the lower-order simulation utilizing parameters obtained from  $f_1(s)$  demonstrates better performance than in simulations with parameters obtained from  $f_2(s)$ . For the high-fidelity simulations, the control parameters – or possibly the controller itself – may need to be adjusted if the attitude of the chaser is to be stabilized. This is expected, as the controller was originally derived and employed in situations where net-wrapped debris towing was not considered [26,27]. The

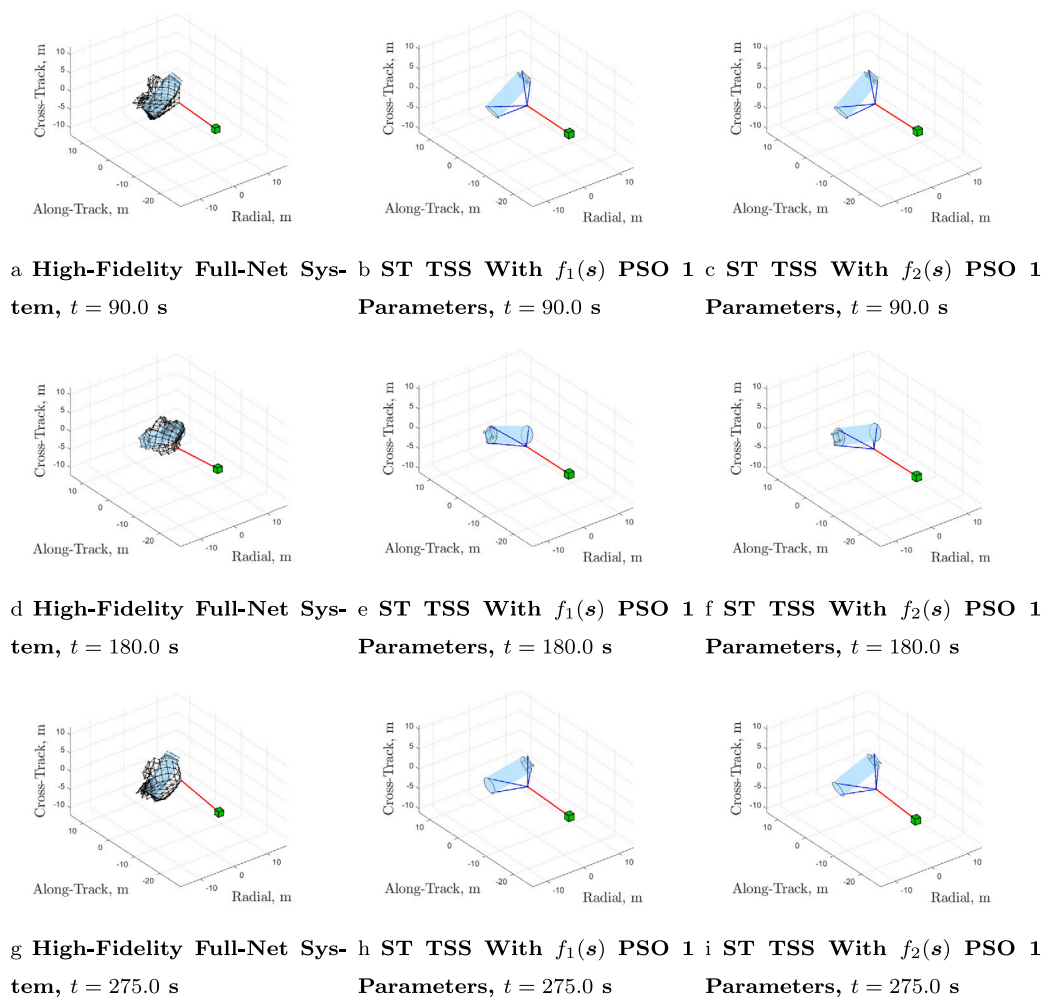


Fig. 6. Comparison between high-fidelity and lower-order model system dynamics vs. Time using best decision variables obtained from  $f_1(s)$  and  $f_2(s)$ .

increased control difficulty is explained by the fact that the net is not rigidly attached to the target in the high-fidelity net capture, while in the lower-order model, the STs are rigidly attached to the target. The difference between the angular velocity of the chaser with the two models can be seen in the lower plots of Fig. 7(c) and 8(c), and is characterized by maximum values of approx. 3.5 and 6.0 deg/s, respectively.

Reflecting the observations made in Fig. 6, the attitude motion of the debris in the lower-order models is overall relatively different from that in the high-fidelity model, as seen in the upper plots of Fig. 7(d) and 8(d). The components of the angular velocities of the debris have maximum differences between the models of approx. 1.6 and 2.8 deg/s for  $f_1(s)$  and  $f_2(s)$ , respectively (see lower plots of Fig. 7(d) and 8(d)). Fewer oscillations in the angular velocity components of the debris – and therefore higher similarity to the corresponding values of the high-fidelity model – are also observed for  $f_1(s)$  compared to  $f_2(s)$ .

## 5. Conclusion

While many works exist in literature that focus either on the net-based debris capture process or the post-capture debris towing via TSS without a net, few studies incorporate net-wrapped debris in the post-capture phase due to the expensive computational cost. However, it is important to bridge these sets of works and improve the efficiency of post-capture simulations. This study explored the use of global optimization algorithms to perform parameter identification of a

lower-order TSS model to approximate a high-fidelity net-based debris towing simulation. To quantitatively match the dynamics of the high-fidelity and lower-order models, two cost functions were formulated that penalize the difference in dynamics quantities of interest: (i) angular velocities of the chaser and debris along with their relative positions in  $f_1(s)$  and (ii) the tension in the MT via  $f_2(s)$ . The parameter identification framework consists of finding the STs stiffness, damping, and length, along with the mass of the ST and MT connection point, that minimize the cost function.

The MADS and PSO global optimization solvers were used to perform the optimization tasks (i) in a benchmark scenario with a known solution and (ii) to determine the STs parameters utilizing a set of high-fidelity simulation data. In the benchmark case, both cost functions converged to solutions that were close to the truth, thereby validating the framework. Using data obtained from the high-fidelity simulations, both of the proposed cost functions consistently converged to similar optimized solutions despite the choice of solvers and initial guesses, the PSO optimizer resulting in slightly better results. While simulations with parameters determined with  $f_2(s)$  show sufficiently similar dynamics between the two models, parameter identification with  $f_1(s)$  produces a better matching of the dynamics to the high-fidelity simulations, except for the tension in the main tether. This is not surprising since  $f_1(s)$  explicitly requires matching a larger set of states. In general, translational dynamics is more accurately represented than attitude dynamics, which is explained by the presence of rigid attachments on the target in the lower-order model that do not exist in the net-based model.

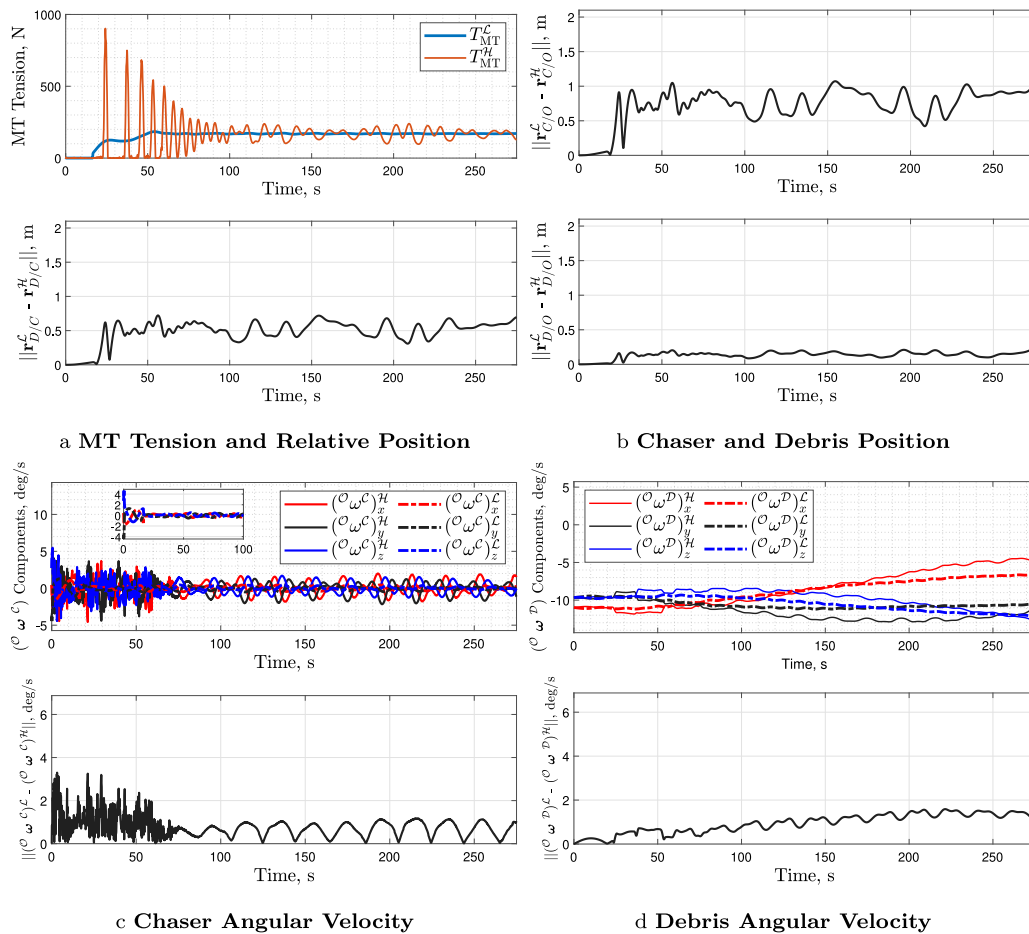


Fig. 7. Comparison between high-fidelity and lower-order model system dynamics using best decision variables obtained from  $f_1(s)$ .

Overall, the proposed framework demonstrated good performance in determining unknown parameters of the STs and connection point in the lower-order TSS model. With the determined parameters, the lower-order TSS model can approximate the dynamics of the high-fidelity model with a much lower computational cost (of less than 4 min vs 4.5 h). The lower-order TSS model performs especially well at the matching of the dynamics after the MT tension reaches a quasi-steady-state. In future studies concerning net-based ADR, simulating the debris towing with the high-fidelity model may be beneficial until the quasi-steady-state is reached. Afterward, the lower-order TSS model can be used as a substitute to reduce the computational cost. This work is limited to the initial (i.e., at the beginning of towing) variations of debris rotation rates and orientations between simulations. Future work should investigate additional aspects of the ADR mission – such as the initial chaser-debris displacement and the thrust applied on the chaser – for the parameter identification training and validation datasets. More varied initial rotation rates and debris attitudes (along additional axes relative to the chaser) could also be considered for future TSS parameter identification studies.

#### Funding sources

This work was funded by the National Science Foundation (NSF) under CRII Award No. 2105011. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Achira Boonrath acknowledges the University at Buffalo Presidential Fellowship.

#### CRedit authorship contribution statement

**Achira Boonrath:** Conceptualization, Formal analysis, Methodology, Software, Validation, Visualization, Writing – original draft, Data curation, Investigation, Writing – review & editing. **Tarunraj Singh:** Writing – review & editing, Conceptualization, Formal analysis, Investigation, Methodology. **Eleonora M. Botta:** Conceptualization, Formal analysis, Funding acquisition, Resources, Supervision, Writing – review & editing, Project administration.

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgments

The authors express gratitude toward CM Labs Simulations for providing licenses for the Vortex Studio simulation framework. The authors would also like to thank Liam Field for providing the LVLH frame system visualization code.

During the preparation of this work the author(s) used Grammarly in order to correct accidental syntax and grammar mistakes during the writing process. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

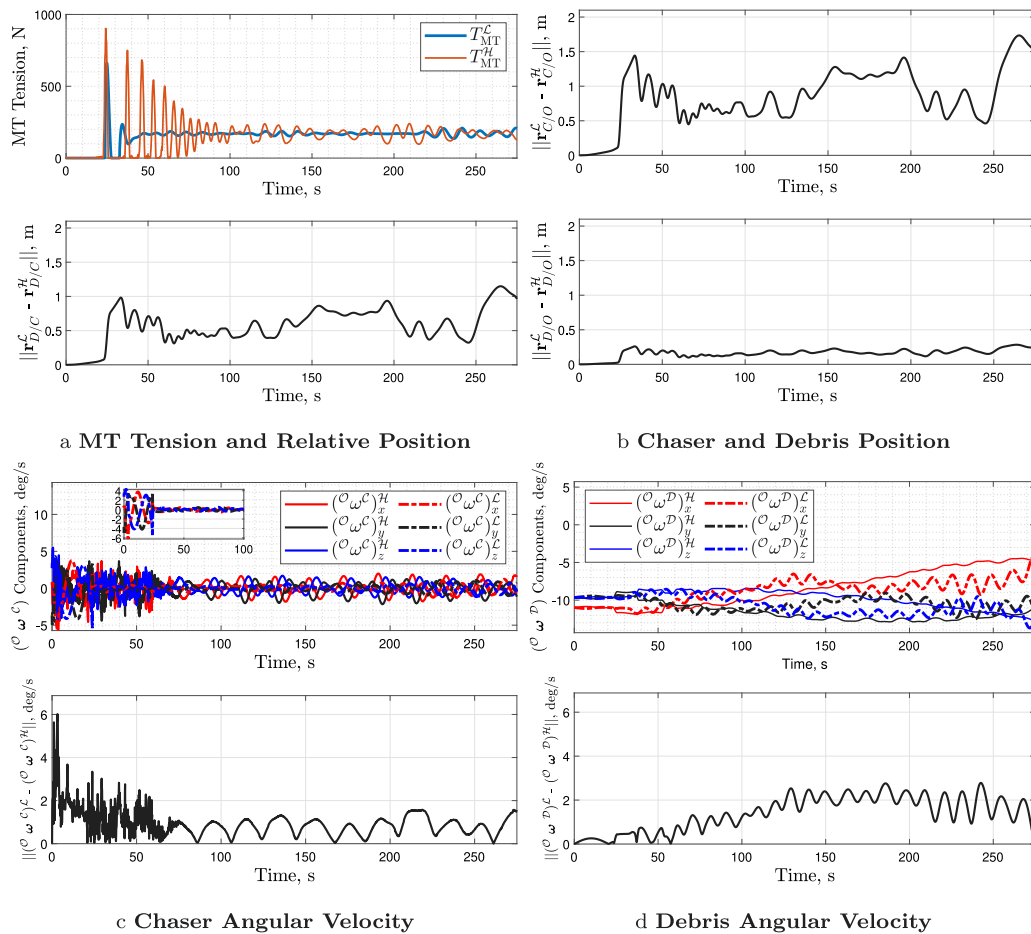


Fig. 8. Comparison between high-fidelity and lower-order model system dynamics using best decision variables obtained from  $f_2(s)$ .

## References

- [1] Esa's Annual Space Environment Report, ESA Space Debris Office, 2024.
- [2] C. Pardini, L. Anselmo, Physical properties and long-term evolution of the debris clouds produced by two catastrophic collisions in Earth orbit, *Adv. Space Res.* 48 (3) (2011) 557–569, <http://dx.doi.org/10.1016/j.asr.2011.04.006>.
- [3] A. Ledkov, V. Aslanov, Review of contact and contactless active space debris removal approaches, *Prog. Aerosp. Sci.* 134 (2022) 100858, <http://dx.doi.org/10.1016/j.paerosci.2022.100858>.
- [4] M. Shan, J. Guo, E. Gill, A review and comparison of active space debris capturing and removal methods, *Prog. Aerosp. Sci.* 80 (2015) 18–32, <http://dx.doi.org/10.1016/j.paerosci.2015.11.001>.
- [5] Z. Zhang, Z. Yu, Q. Zhang, M. Zeng, S. Li, Dynamics and control of a tethered space-tug system using Takagi-Sugeno fuzzy methods, *Aerosp. Sci. Technol.* 87 (2019) 289–299, <http://dx.doi.org/10.1016/j.ast.2019.02.033>.
- [6] B. Wang, Z. Meng, P. Huang, Attitude control of towed space debris using only tether, *Acta Astronaut.* 138 (2017) 152–167, <http://dx.doi.org/10.1016/j.actaastro.2017.05.012>.
- [7] X. Sun, R. Zhong, Nutation damping and spin orientation control of tethered space debris, *Acta Astronaut.* 160 (2019) 683–693, <http://dx.doi.org/10.1016/j.actaastro.2019.03.019>.
- [8] A.S. Ledkov, V.S. Aslanov, Active space debris removal by ion multi-beam shepherd spacecraft, *Acta Astronaut.* 205 (2023) 247–257, <http://dx.doi.org/10.1016/j.actaastro.2023.02.003>.
- [9] E.M. Botta, Deployment and Capture Dynamics of Tether-Nets for Active Space Debris Removal (Ph.D. thesis), McGill University, Montreal, QC, 2017.
- [10] 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, *Aeronaut. J.* 124 (1271) (2020) 1–23, <http://dx.doi.org/10.1017/aer.2019.136>.
- [11] R. Axthelm, B. Klotz, I. Retat, U. Schlossstein, W. Tritsch, S. Vahsen, Net capture system for debris removal demonstration mission, in: 7th European Conference on Space Debris, ESOC, Darmstadt, Germany, 2017.
- [12] E.M. Botta, I. Sharf, A.K. Misra, Simulation of tether-nets for capture of space debris and small asteroids, *Acta Astronaut.* 155 (2019) 448–461, <http://dx.doi.org/10.1016/j.actaastro.2018.07.046>.
- [13] A. Boonrath, E.M. Botta, Validation of models for net deployment and capture simulation with experimental data, *J. Spacecr. Rockets* (2023) 1–19, <http://dx.doi.org/10.2514/1.A35798>.
- [14] R. Benvenuto, M. Lavagna, S. Salvi, Multibody dynamics driving GNC and system design in tethered nets for active debris removal, *Adv. Space Res.* 58 (1) (2016) 45–63, <http://dx.doi.org/10.1016/j.asr.2016.04.015>.
- [15] R. Benvenuto, M. Lavagna, A. Cingoli, C. Yabar, M. Casasco, et al., MUST: Multi-body dynamics simulation tool to support the GNC design for active debris removal with flexible elements, in: Proceedings of 9th International ESA Conference on Guidance, Navigation & Control Systems, Porto, Portugal, 2014.
- [16] K. Hovell, S. Ulrich, Attitude stabilization of an uncooperative spacecraft in an orbital environment using visco-elastic tethers, in: AIAA Guidance, Navigation, and Control Conference, 2016, p. 0641, <http://dx.doi.org/10.2514/6.2016-0641>.
- [17] M. Shan, L. Shi, Comparison of tethered post-capture system models for space debris removal, *Aerospace* 9 (1) (2022) 33, <http://dx.doi.org/10.3390/aerospace9010033>.
- [18] K. Yang, A.K. Misra, J. Zhang, R. Qi, S. Lu, Y. Liu, Dynamics of a debris towing system with hierarchical tether architecture, *Acta Astronaut.* 177 (2020) 891–905, <http://dx.doi.org/10.1016/j.actaastro.2019.10.048>.
- [19] K. Stadnyk, S. Ulrich, Validating the deployment of a novel tether design for orbital debris removal, *J. Spacecr. Rockets* 57 (6) (2020) 1335–1349, <http://dx.doi.org/10.2514/1.A34781>.
- [20] L. Ljung, System identification: theory for the user, second ed., in: Prentice Hall Information and System Sciences Series, Prentice Hall PTR, Upper Saddle River, NJ, 1999.
- [21] R. Isermann, M. Münchhof, Identification of Dynamic Systems: An Introduction with Applications, vol. 85, Springer, 2011.
- [22] J. Silha, T. Schildknecht, T. Flohrer, Optical observations of BRIZ-M fragments in GEO, in: Proceedings of AMOS Conference, Maui, Hawaii, 2016.
- [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 Astronaut.* 181 (2021) 282–291, <http://dx.doi.org/10.1016/j.actaastro.2021.01.021>.
- [24] E.M. Botta, I. Sharf, A.K. Misra, Contact dynamics modeling and simulation of tether nets for space-debris capture, *J. Guid. Control Dyn.* 40 (1) (2017) 110–123.



- [25] L. Field, *Modeling, simulation, and control of tethered space debris*, (Master's thesis), State University of New York at Buffalo, 2022.
- [26] L. Field, E.M. Botta, Relative distance control of uncooperative tethered debris, *J. Astronaut. Sci.* 70 (6) (2023) 55, <http://dx.doi.org/10.1007/s40295-023-00422-7>.
- [27] J.L. Crassidis, S.R. Vadali, F.L. Markley, Optimal variable-structure control tracking of spacecraft maneuvers, *J. Guid. Control Dyn.* 23 (3) (2000) 564–566, <http://dx.doi.org/10.2514/2.4568>.
- [28] H. Schaub, J.L. Junkins, *Analytical Mechanics of Space Systems*, AIAA, 2003.
- [29] V.S. Aslanov, V.V. Yudinsev, Dynamics of large debris connected to space tug by a tether, *J. Guid. Control Dyn.* 36 (6) (2013) 1654–1660, <http://dx.doi.org/10.2514/1.60976>.
- [30] S. Alarie, C. Audet, A.E. Gheribi, M. Kokkolaras, S. Le Digabel, Two decades of blackbox optimization applications, *EURO J. Comput. Optim.* 9 (2021) 100011, <http://dx.doi.org/10.1016/j.ejco.2021.100011>.
- [31] Y. Zhang, S. Wang, G. Ji, A comprehensive survey on particle swarm optimization algorithm and its applications, *Math. Probl. Eng.* 2015 (1) (2015) 931256, <http://dx.doi.org/10.1155/2015/931256>.
- [32] S. Chen, C.T. Woods, A. Boonrath, E.M. Botta, Analysis of the robustness and safety of net-based debris capture, in: *AIAA SCITECH 2022 Forum*, 2022, p. 1001, <http://dx.doi.org/10.2514/6.2022-1001>.
- [33] J. Šilha, J.-N. Pittet, M. Hamara, T. Schildknecht, Apparent rotation properties of space debris extracted from photometric measurements, *Adv. Space Res.* 61 (3) (2018) 844–861, <http://dx.doi.org/10.1016/j.asr.2017.10.048>.



**Achira Boonrath** is an Aerospace Engineering Ph.D. student in the Department of Mechanical and Aerospace Engineering at the University at Buffalo (UB). Achira received his Bachelor of Science and Masters of Science in Aerospace Engineering from UB in 2022 and 2024, respectively. His primary research interest is in the simulation and modeling of flexible space systems designed for active debris removal.



**Tarunraj Singh** earned his Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in mechanical engineering in 1991. He was a postdoctoral fellow in Aerospace Engineering at Texas A & M University from 1991–1993. Since 1993, he has been with the University at Buffalo, Buffalo, NY, where he is currently a Professor with the Department of Mechanical and Aerospace Engineering. His research interests are in robust vibration control, optimal control, estimation, and uncertainty quantification. He is a fellow ASME and AAAS and has served as Associate Editor of *IEEE Mechatronics*, *Control Engineering Practice* and *ASME Journal for Dynamic Systems, Measurement and Control*.



**Eleonora M. Botta** is an assistant professor in the Department of Mechanical and Aerospace Engineering at the University at Buffalo, NY. She received her Ph.D. in Mechanical Engineering from McGill University, QC, Canada in 2017, and holds degrees from Politecnico di Milano and Politecnico di Torino, Italy. Her main research interests are the dynamics and control of tethered space systems, spacecraft trajectory optimization, and space situational awareness. She is a senior member of AIAA and a member of AAS and IEEE. She is also the Chair of the AIAA Space Tethers Technical Committee, a member of the AAS Space Flight Mechanics Committee, and an Associate Editor for the *AIAA Journal of Spacecraft and Rockets* and for *Advances in Space Research*.