

In the above equation, the variables m_D , ${}^O\mathbf{a}_{D/O}$, $\sum_{i=1}^{N_{\text{nodes}}} \mathbf{F}_{\text{contact},i}$, and $\mathbf{F}_{G,D}$ are the mass of the debris, the inertial acceleration of the CoM of the debris, the sum of the forces resulting from contact between the debris and a net possessing a total of N_{nodes} nodes, and the force due to gravity acting on the debris (computed using Eq. (1) with the debris's mass and position substituted). For each i -th node of the net, $\mathbf{F}_{\text{contact},i}$ is non-zero only when a collision between the node and the surface of the debris is detected [31, 32], with $\mathbf{F}_{\text{contact},i}$ being computed per the methodologies described earlier in this section.

Euler's Second Law is then expressed for the debris as:

$$J_D \frac{d}{dt}({}^O\boldsymbol{\omega}^D) = \boldsymbol{\tau}_D - ({}^O\boldsymbol{\omega}^D) \times J_D({}^O\boldsymbol{\omega}^D) \quad (7)$$

with J_D , $\frac{d}{dt}({}^O\boldsymbol{\omega}^D)$, $\boldsymbol{\tau}_D$, and ${}^O\boldsymbol{\omega}^D$ representing the debris's principal mass moment of inertia matrix, the time derivative of the chaser's angular velocity with respect to its body-fixed frame, the torque applied onto the debris through net-debris contact, and the angular velocity of the debris relative to the inertial frame, respectively. To compute the torque applied onto the debris, the following expression is used (assuming each node has a very small radius):

$$\boldsymbol{\tau}_D = \sum_{i=1}^{N_{\text{nodes}}} \mathbf{r}_{i/D} \times \mathbf{F}_{\text{contact},i} \quad (8)$$

with $\mathbf{r}_{i/D}$ representing the position of each i -th node relative to the CoM of the debris. Like the chaser, the attitude of the debris with respect to the O frame is quantified through the definition of a rotation matrix ${}^O A^D$ and a quaternion ${}^O \mathbf{q}^D$.

III. Model Reference Adaptive Control for Net-Based Debris Towing

During the post-capture phase of a net-based ADR mission (i.e., when the chaser is relocating the debris to a graveyard orbit or deorbiting it into the Earth's atmosphere), the elongation of MT (i.e., z) should be ideally positive for safety and stability related reasons, as mentioned in Section I. In this work, two formulations of MRAC algorithms for MT elongation control will be presented, labeled MRAC-1 and MRAC-2 for the rest of the work. In general, MRAC algorithms are designed to modify the dynamics of a given system so that it behaves similarly to an ideal *reference model* [28, 29], exhibiting a desired stable behavior. Although both of the formulations presented in this work are considered *direct* MRAC algorithms (i.e., algorithms that directly estimate the ideal controller parameters by evaluating the tracking errors between the actual system's states and those of the reference model), there are significant differences in the mathematical structure of the control and parameter adaptation laws employed for each. As will be seen in the following, MRAC-1 has a smaller total number of control parameters compared to MRAC-2, resulting in less effort required for the initial selection of parameters, but also less adaptability.

A. MRAC-1 MT Elongation Control

Here, the first MRAC algorithm is presented. The algorithm has been modified from the one presented in [28], where the original algorithm was derived for controlling single-input, single-output, control-affine nonlinear systems with a single-degree-of-freedom (DoF) and with an approximately known mathematical model. The original algorithm assumed that all system states could be observed (i.e., assumed the knowledge of the state variable to be controlled and its time derivatives). Accordingly, for all algorithms developed and employed in this work, the states $\mathbf{z} = [z, \dot{z}]^T$ and their time derivatives $\dot{\mathbf{z}} = [\ddot{z}, \ddot{\dot{z}}]^T$ are used to determine the control force. In this work, the algorithm proposed in [28] is modified to increase control robustness against large possible errors between the actual system's states and those of the reference model and to handle actuation saturation; the details of the modifications are shown near the end of this section.

Here, given the fact that the ADR system has multiple DoFs (i.e., including the states of multiple rigid and particle bodies moving in three-dimensional space per the equations of Section II.B), the following assumptions are made regarding the dynamics for the purpose of defining the mathematical structure of the MRAC-1 algorithm per [28]: (i) the attitude motion of the chaser is not considered, as the sliding model attitude control can stabilize its rotational motion (i.e., in practice, for the translational control, the chaser can be treated as a particle), (ii) $\cos^{-1}(-\hat{\mathbf{e}}_{MT} \cdot \hat{\mathbf{e}}_z)$ is assumed to be near-zero, resulting in the MT being approximately aligned with the $\hat{\mathbf{e}}_z$ direction, (iii) the gravitational acceleration has a small effect on the chaser-debris relative dynamics (i.e., it is ignored for control derivations), and (iv) $m_C \ll m_D$ and ${}^O \mathbf{v}_{C/P} \approx {}^O \mathbf{v}_{C/D}$ under the assumptions that the chaser behaves, in relation to the towed debris, like a

nonlinear single-DoF mass-spring-damper system that is attached to a rigid wall. The equation of motion for the chaser (as observed along the $-\hat{\mathbf{e}}_{MT}$ direction from the debris) is abstracted to the following for control law design:

$$\ddot{z}^* + b(c_{MT}\dot{z}^* + k_{MT}z^*)g(z^*, \dot{z}^*) = bF_t \quad (9)$$

where $b = 1/m_C$ and $F_t = \text{sat}(u) = \min(\max(u, -u_{\max}), u_{\max})$, with u and u_{\max} representing the pre-saturation control actuation and the maximum actuation magnitudes, respectively. The variables $\mathbf{z}^* = [z^*, \dot{z}^*]^T$ represent the states of the *abstracted model* that employs the above four simplifications; it is notably different from the true dynamics of the full-net system. The above single-DoF equation can also be written in a state-space form as follows:

$$\dot{\mathbf{z}}^* = \mathbf{f}^*(\mathbf{z}^*) = \begin{bmatrix} 0 & 1 \\ -bk_{MT}g(z^*, \dot{z}^*) & -bc_{MT}g(z^*, \dot{z}^*) \end{bmatrix} \mathbf{z}^* + [0, b]^T F_t \quad (10)$$

where $\mathbf{f}^*(\mathbf{z}^*)$ denotes the abstracted model.

Based on the abstracted model $\mathbf{f}^*(\mathbf{z}^*)$, let the pre-saturation scalar control force be defined as [28]:

$$u = \hat{h}\eta_r + g(z, \dot{z})\hat{\mathbf{a}}^T \mathbf{z} - k_s s \quad (11)$$

where \hat{h} and $\hat{\mathbf{a}} = [\hat{a}_1, \hat{a}_2]^T$ are adaptive control parameters that evolve over time. The parameter k_s is a constant gain that is multiplied by the sliding variable s . The intermediate variable η_r is defined as:

$$\eta_r = \ddot{z}_m - \lambda_0(\dot{z} - \dot{z}_m) \quad (12)$$

with z_m being the *reference model elongation* (i.e., the elongation of a dynamics model that the actual system with the full-net model should match). The parameter λ_0 is a gain-like constant for the difference between the actual and reference states. For this work, the reference model is chosen as:

$$\dot{\mathbf{z}}_m = \mathbf{f}_m(\mathbf{z}_m, z_d) = \begin{bmatrix} 0 & 1 \\ -bk_{MT}g(z, \dot{z}) & -bc_{MT}g(z, \dot{z}) \end{bmatrix} \mathbf{z} + [0, bsat(K_{P,m}(z_d - z_m) + K_{D,m}(-\dot{z}_m))]^T \quad (13)$$

where z_d is the desired elongation of the MT, $\dot{\mathbf{z}}_m = [\dot{z}_m, \ddot{z}_m]^T$, and $\mathbf{z}_m = [z_m, \dot{z}_m]^T$. This model is equivalent to a single-DoF nonlinear mass-spring-damper system that incorporates a proportional-derivative (PD) controller to achieve its desired elongation. It resembles the system analyzed in [23], which demonstrated the ability to achieve the elongation control objectives in various scenarios with varying initial conditions and system parameters. The variables K_P and K_D represent the proportional and derivative gains for the reference model, respectively; these gain values must be positive, and should be adjusted to achieve the desired transient and steady state behaviors [23]. Here, it should be noted that other choices of reference model and control are possible (e.g., using PID instead of PD control for $\mathbf{f}_m(\mathbf{z}_m, z_d)$), where as long as the references exhibit stable convergence of \mathbf{z}_m towards a desired value, the overall functionality of the adaptive algorithm proposed in this work would remain the same. Finally, the sliding variable s is defined as:

$$s = (\dot{z} - \dot{z}_m) + \lambda_0(z - z_m) \quad (14)$$

To update the control parameters, the following differential equations are defined:

$$\dot{\hat{h}} = (-s\gamma\eta_r - s^2\xi_h\hat{h})H(u_{\max} - |u|) \quad (15a)$$

$$\dot{\hat{\mathbf{a}}} = (-s\Gamma\mathbf{z} - s^2\Xi_a\hat{\mathbf{a}})H(u_{\max} - |u|), \quad (15b)$$

where $\Gamma = \text{diag}(\gamma, \gamma)$ and γ is the parameter used to adjust the rate of adaptation. The above adaptation laws are modified from [28] through the incorporation of terms $H(u_{\max} - |u|)$, $s^2\xi_h\hat{h}$, and $s^2\Xi_a\hat{\mathbf{a}}$. The $H(u_{\max} - |u|)$ term is present to pause the adaptation of the parameters when the control force is saturated (i.e., when $|u| > u_{\max}$); this is similar in concept to an integrator anti-windup algorithm for a PID controller [33]. The $s^2\xi_h\hat{h}$ and $s^2\Xi_a\hat{\mathbf{a}}$ terms, with damping gain constant ξ_h and diagonal gain matrix $\Xi_a \in \mathbb{R}^{2 \times 2}$, are included to improve the robustness of the algorithm against large errors between the reference and actual models; this is done by dampening the adaptation rate when the errors between the actual and desired system dynamics are relatively large, thus preventing an excessive change in the control parameters [29]. With the pre-saturation control force determined as per Eq. (11), the thrust (control) force applied onto the chaser within the debris removal system with the full-net model is defined as:

$$\mathbf{F}_t = F_t \hat{\mathbf{e}}_z = \text{sat}(u) \hat{\mathbf{e}}_z \quad (16)$$

The control framework is summarized in Fig. 2, which displays the block diagram for MRAC-1. The "Plant" block on the diagram represents the actual system dynamics (i.e., \mathbf{z} as computed via the equations of Section II.B). For MRAC-1, the desired elongation z_d is only fed to the reference model (and not directly to the control law); this results in the control algorithm relying only on the error between the actual and reference dynamics for computing u . The selection of parameters for MRAC-1, for both the constant and adaptive parameters, is addressed in Section IV.A.

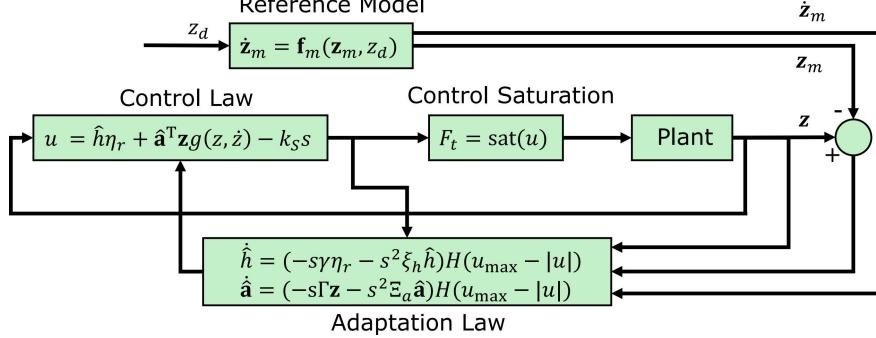


Fig. 2 Control Block Diagram for MRAC-1

B. MRAC-2 MT Elongation Control

Here, an alternate version of the MRAC algorithm is proposed for the control task, modified from the *e-modification* algorithm presented in [29]. This second formulation is inspired by the fact that the abstracted system (see Eq. (10)) is linear when $g(z, \dot{z}) = 1$ (i.e., when the system is elongated and its tension is positive); in such a case, a controller designed for linear systems with affine nonlinear components (e.g., the *e-modification* algorithm in [29]) is potentially suited for the control task. Compared to MRAC-1, this formulation benefits from greater adaptability, as more parameters evolve to minimize the error; however, with a larger total number of control parameters, more effort is required for their initial selection. As with the previous algorithm, the system model is first abstracted into Eq. (10) for the purpose of formulating the mathematical structure of the control law (i.e., abstracting the system model into a single-DoF system with a single scalar input), with all of its associated assumptions. Then, Eq. (10) is algebraically reorganized into the following for the MRAC-2 algorithm:

$$\dot{\mathbf{z}}^* = \mathbf{f}^*(\mathbf{z}^*) = A\mathbf{z}^* + [0, b]^T ((-c_{MT}\dot{z}^* - k_{MT}z^*)(g(z^*, \dot{z}^*) - 1) + F_t), \quad A = \begin{bmatrix} 0 & 1 \\ -bk_{MT} & -bc_{MT} \end{bmatrix} \quad (17)$$

where $A\mathbf{z}^*$ is the linear component of the dynamics equation, while $[0, b]^T((-c_{MT}\dot{z} - k_{MT}z)(g(z^*, \dot{z}^*) - 1))$ is the nonlinear component. Given that $\mathbf{B} = [0, b]^T$, the state-space equation for the abstracted model is then transformed into:

$$\dot{\mathbf{z}}^* = A\mathbf{z}^* + \mathbf{B}F_t + \mathbf{B}\Theta^T \phi(\mathbf{z}^*) \quad (18a)$$

$$\Theta = [-k_{MT}, -c_{MT}]^T \quad (18b)$$

$$\phi(\mathbf{z}^*) = (g(z^*, \dot{z}^*) - 1)\mathbf{z}^* \quad (18c)$$

where Θ and $\phi(\mathbf{z})$, respectively, represent the MT properties and the nonlinear component of the dynamics equation, which is equal to zero when the MT is under tension. Given the above set of equations, the pre-saturation MRAC-2 control force is defined as [29]:

$$u = \hat{\mathbf{K}}_z^T \mathbf{z} + \hat{k}_r z_d - \hat{\Theta}^T \phi(\mathbf{z}) \quad (19)$$

where $\hat{\mathbf{K}}_z \in \mathbb{R}^{2 \times 1}$ and $\hat{\Theta} \in \mathbb{R}^{2 \times 1}$ are arrays of adaptive parameters and \hat{k}_r is an adaptive scalar parameter. The first term on the right-hand side of Eq. (19), $\hat{\mathbf{K}}_z^T \mathbf{z}$, is equivalent to a full-state feedback controller for a second-order system, while the second term, $\hat{k}_r z_d$, represents a feed-forward component. The last term in the above equation represents an approximate feedback linearization term that is intended to counteract the nonlinearities of the system [29]. Here, it should be noted that in Eq. (19), the time-varying, estimated $\hat{\Theta}$ term is used for the control, rather than the true Θ .

term. The estimated term is used to account for the difference between the actual system dynamics (i.e., dynamics as described in Section II.B) compared to the abstracted model (i.e., $\mathbf{f}^*(\mathbf{z}^*)$ of Eq. (10)), the latter only employed to define the algorithm's mathematical formulation. This formulation is based on five evolving scalar parameters in total, two more than that of MRAC-1.

Let us define with $\mathbf{e} = \mathbf{z} - \mathbf{z}_m$, the error between the states of the actual model and of the reference models (described again by Eq. (13)). The parameter adaptation laws for MRAC-2 are defined as follows:

$$\dot{\hat{\mathbf{K}}}_z = (-\Gamma_z \mathbf{e}^T P \mathbf{B} - \xi_2 ||\mathbf{e}^T P \mathbf{B}|| \Gamma_z \hat{\mathbf{K}}_z) H(u_{\max} - |u|) \quad (20a)$$

$$\dot{\hat{k}}_r = (-\gamma_r z_d \mathbf{e}^T P \mathbf{B} - \xi_2 z_d ||\mathbf{e}^T P \mathbf{B}|| \gamma_r \hat{k}_r) H(u_{\max} - |u|) \quad (20b)$$

$$\dot{\hat{\Theta}} = (-\Gamma_\Theta \hat{\Theta}^T \mathbf{e}^T P \mathbf{B} - \xi_2 ||\mathbf{e}^T P \mathbf{B}|| \Gamma_\Theta \hat{\Theta}) H(u_{\max} - |u|) \quad (20c)$$

Except for the multiplications by $H(u_{\max} - |u|)$ – again present to pause the adaptation of the parameters when the control force is saturated – the structures of the above adaptation laws are identical to those of the *e-modification* algorithm in [29]. The variables $\Gamma_z \in \mathbb{R}^{2 \times 2}$ and $\Gamma_\Theta \in \mathbb{R}^{2 \times 2}$ are constant gain matrices used to adjust the adaptation rates, and γ_r represents a parameter update rate scalar. The positive definite matrix $P \in \mathbb{R}^{2 \times 2}$ is the solution to a linear Lyapunov equation, defined as the following for this work:

$$P \begin{bmatrix} 0 & 1 \\ -b(k_{MT} + K_P) & -b(c_{MT} + K_D) \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ -b(k_{MT} + K_P) & -b(c_{MT} + K_D) \end{bmatrix}^T P = -Q \quad (21a)$$

where the matrix $Q \in \mathbb{R}^{2 \times 2}$ is a constant symmetric, positive definite matrix. With u determined by Eq. (19), the thrust (control) force applied onto the chaser within the debris removal system with the full-net model is again defined per Eq. (16).

The control framework for MRAC-2 is summarized in Fig. 3. Different from MRAC-1, the desired elongation z_d is used by both the reference model and the actual controller, while the error between the reference and actual states is only used for the parameter adaptation law. It should also be noted that only \mathbf{z}_m (not $\dot{\mathbf{z}}_m$) is used to compute the pre-saturation control u , resulting in a slightly less complex structure for the computation of u compared to MRAC-1. The selection of parameters for MRAC-2, for both the constant and adaptive parameters, is addressed in Section IV.A.

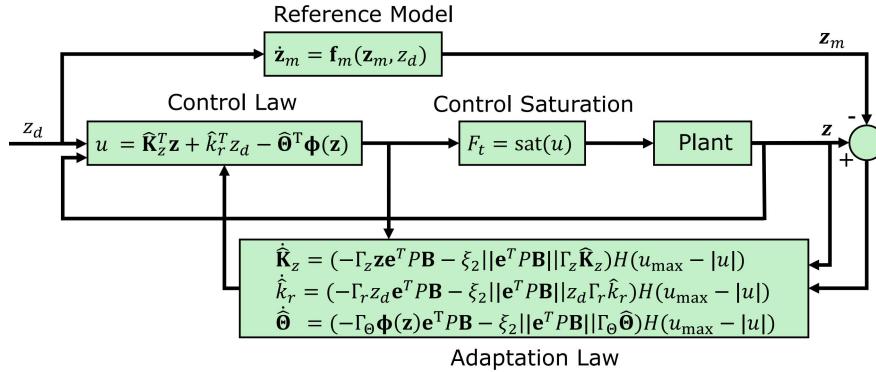


Fig. 3 Control Block Diagram for MRAC-2

IV. Simulation Results and Analysis

In the following, results are presented for scenarios in which the proposed MRAC algorithms were applied to control the MT elongation throughout net-based debris towing missions. Section IV.A provides the simulation settings and initial conditions used for this work. In Section IV.B, the time histories of the system's dynamical quantities and of the controls will be examined for a nominal scenario; the results shown will demonstrate the transient and steady state behaviors of the system with the applied controls. Then in Section IV.C, the controller performances will be compared using simulations where the captured debris has varying initial conditions (i.e., varying rotation rates and initial orientation with respect to the chaser); these simulations are expected to demonstrate the robustness of the proposed MRAC algorithm in situations with uncertain knowledge of the pose of the debris.