2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

# Validation of Space Robotics in Underwater Environments via Disturbance Robustness Equivalency

Joris Verhagen*[1], Elias Krantz[2], Chelsea Sidrane[1], David Dörner[2], Nicola De Carli[3], Pedro Roque[3,4], Huina Mao[2], Gunnar Tibert[2], Ivan Stenius[2], Christer Fuglesang[2], Dimos V. Dimarogonas[3], Jana Tumova[1]

***Abstract*— We present an experimental validation framework for space robotics that leverages underwater environments to approximate microgravity dynamics. While neutral buoyancy conditions make underwater robotics an excellent platform for space robotics validation, there are still dynamical and environmental differences that need to be overcome. Given a high-level space mission specification, expressed in terms of a Signal Temporal Logic specification, we overcome these differences via the notion of *maximal disturbance robustness* of the mission. We formulate the motion planning problem such that the original space mission and the validation mission achieve the same disturbance robustness degree. The validation platform then executes its mission plan using a near-identical control strategy to the space mission where the closed-loop controller considers the spacecraft dynamics. Evaluating our validation framework relies on estimating disturbances during execution and comparing them to the disturbance robustness degree, providing practical evidence of operation in the space environment. Our evaluation features a dual-experiment setup: an underwater robot operating under near-neutral buoyancy conditions to validate the planning and control strategy of either an experimental planar spacecraft platform or a CubeSat in a high-fidelity space dynamics simulator.**

**Code and videos can be found on the project page**

## I. INTRODUCTION

The significant costs and safety-critical nature of autonomous space missions make verification and validation essential steps in the development pipeline. However, faithfully reproducing spacecraft operation conditions on Earth remains a longstanding challenge. Several terrestrial validation platforms have been developed to address this, including granite tables [1] or epoxy resin floors [2] serving as calibrated flat surfaces where robotic systems are supported on air bearings to approximate undamped planar motion. Higher degrees-of-freedom (DoF) weightlessness has been emulated using robotic arms with specialized joints [3], [4]. While these facilities provide valuable insights, their accuracy for testing true 6-DoF motion is limited due to the difficulty of emulating weightlessness. Moreover, such platforms are impractical for evaluating human–robot collaboration tasks, which increasingly play a role in future space missions.

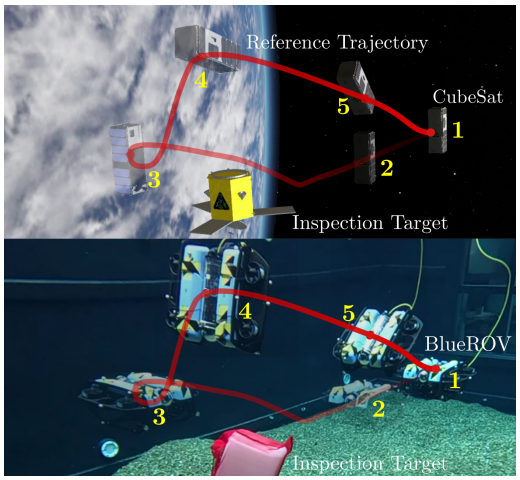


Fig. 1: A physical BlueROV underwater robot is utilized to validate a planner and controller which aim to inspect a passive target in space. While the environments are distinct, they share sufficient similarities in order to be used as a validation platform.

As an alternative, underwater environments have emerged as a promising surrogate for space. Neutral buoyancy experiments are already a cornerstone of astronaut training, with large-scale facilities such as NASA's Neutral Buoyancy Laboratory [5] and the Neutral Buoyancy Research Facility [6] (NBRF) at the University of Maryland supporting both human operations and autonomous system testing. The appeal of underwater environments lies in their ability to approximate aspects of microgravity: a vehicle close to neutral buoyancy experiences significantly reduced weight effects, and the surrounding fluid environment induces dynamics with important parallels to those of free-flying robots in space. These commonalities have motivated the use of underwater systems for prototyping, training, and system-level evaluation of space technologies.

Several underwater spacecraft analogs have been developed toward this goal. At the Space Systems Laboratory (SSL), free-flying underwater vehicles such as SCAMP [7], EUCLID [8], and the Ranger Neutral Buoyancy Vehicle (NBV) [9] were deployed in NBRF. SCAMP, finalized in 1997, was primarily teleoperated and used for inspection, extra-vehicular activity support, and situational awareness, while later work incorporated vision-based closed-loop control for semi-autonomous station keeping and position regulation. Ranger NBV and EUCLID extended this line of research toward servicing and multi-vehicle operations.

*Corresponding author: jorisv@kth.se
[1]Division of Robotics, Perception and Learning, KTH Royal Institute of Technology, Stockholm, Sweden.
[2]School of Engineering Sciences, KTH Royal Institute of Technology, Stockholm, Sweden.
[3]Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden.
[4] Department of Mechanical and Civil Engineering, California Institute of Technology, Pasadena, United States.
This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation and by the Swedish Research Council (VR) grant number 2024-05701. The authors are also affiliated with Digital Futures.

Despite their similarities, key challenges remain in making underwater experiments a systematic validation tool for space robotics. Most notably, the differences in dynamics between the two environments during mission execution introduce uncertainties about how performance in water translates to performance in space. To bridge this gap, we argue that validation requires adapting the mission specification in a principled way to produce *equivalent* behavior across robots. In particular, when a mission is specified in terms of formal requirements – such as those expressed in Signal Temporal Logic (STL) [10] – the generated trajectories must not only satisfy these requirements but also exhibit a comparable degree of robustness with respect to disturbances.

A comparable line of research addresses cross-platform validation via formal abstractions such as simulation relations [11], relative reachability [12] or maneuver automata [13]. This establishes an equivalence between systems at the level of discrete behaviors. Our goal instead is to, with minimal modification to the planning and control stack of the space platform, validate its operation in a different environment without conservativeness and with global optimality on complex task specifications. Our approach subsequently relies on equivalence through a planner (via disturbance robustness degrees) and a closed-loop controller (via feedback equivalence).

*Contributions*

The central idea of this work is that underwater testing can be used to validate spacecraft motion planners and controllers for on-orbit tasks. Our specific contributions are:

- **Robustness-matched planning.** We extend the notion of *disturbance robustness* [14] to capture the maximal disturbances that can be counteracted during the mission. We adjust the mission specification so that its robustness in underwater settings matches the robustness associated with execution in space. Thereby, we ensure comparability of plans across the two environments.
- **Feedback-equivalent control.** We validate on the underwater robot a control architecture that is nearly identical to its space counterpart through feedback equivalence and Model Predictive Control (MPC), enabling consistent execution of the same planned trajectories across environments. We additionally estimate disturbances acting on the underwater robot and assess whether they remain within the disturbance robustness degree from the planner, thereby ensuring that underwater results are representative of the intended space performance.
- **Experimental validation.** We evaluate the validation framework underwater using a BlueROV and two space robot platforms: a physical planar spacecraft analog and a CubeSat in an astrodynamics simulator. Two region-reaching scenarios are evaluated: (i) a 2D inspection-style scenario, comparing the underwater robot with the space robot; (ii) a 3D inspection-style scenario, comparing the underwater robot with a CubeSat simulation (seen in Fig. 1.)

## II. PRELIMINARIES

Let $\mathbb{R}$ denote the set of real numbers, $\mathbb{S}^n$ the $n$-dimensional unit sphere, $\mathbb{N}$ the set of natural numbers (including zero), and $\mathbb{B} := \{\top, \bot\}$ the set of binary values, representing *true* and *false*, respectively. $\mathbb{R}_{\geq 0}$ denotes the set of nonnegative real numbers. We use $\underline{\boldsymbol{x}}$ and $\overline{\boldsymbol{x}}$ to denote the lower-bound and upper-bound that vector $\boldsymbol{x} \in \mathbb{R}^n$ may take. Further, let $\hat{\boldsymbol{x}}$ denote the estimation of $\boldsymbol{x}$.

### A. Signal Temporal Logic

STL is a formal specification language widely used to express temporal properties of dynamical systems. An STL formula can be interpreted either qualitatively (the signal satisfies or violates the specification) or quantitatively (the extent to which the signal satisfies or violates it).

Time-bounded STL over $n$-dimensional continuous-time signals $x : \mathbb{R}_{\geq 0} \to X \subseteq \mathbb{R}^n$ is defined as follows.

*Definition 1 (Time-bounded STL):* Let $I = [t_1, t_2]$ be a closed bounded time interval, with $t_1, t_2 \in \mathbb{R}_{\geq 0}$ and $t_1 \leq t_2$. Let $\mu : X \to \mathbb{R}$ be a real-valued function, and define the Boolean predicate $p$ as

$$p = \begin{cases} \top, & \mu(x) \geq 0 \\ \bot, & \mu(x) < 0 \end{cases} . \tag{1}$$

STL formulas are then recursively constructed as

$$\phi := \top \mid p \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \mathcal{U}_I \phi_2, \tag{2}$$

where $\neg$ and $\wedge$ denote Boolean negation and conjunction, and $\mathcal{U}_I$ is the time-bounded Until operator.

Intuitively, $\phi_1 \mathcal{U}_I \phi_2$ requires $\phi_1$ to hold until, within interval $I$, $\phi_2$ becomes true. From this, additional temporal operators can be derived, such as the Eventually operator $\diamondsuit_I \phi = \top \mathcal{U}_I \phi$ (stating that $\phi$ eventually holds within $I$), and the Always operator $\square_I \phi = \neg \diamondsuit_I \neg \phi$ (stating that $\phi$ holds at all times within $I$).

Beyond Boolean satisfaction, STL admits a quantitative semantics known as *spatial robustness*, which evaluates to what *degree* a signal satisfies or violates a specification at time $t$. Spatial robustness is defined recursively as:

$$\rho_p(t, x) = \mu(x(t)), \tag{3a}$$

$$\rho_{\neg\phi}(t, x) = -\rho_\phi(t, x), \tag{3b}$$

$$\rho_{\phi_1 \wedge \phi_2}(t, x) = \min(\rho_{\phi_1}(t, x), \rho_{\phi_2}(t, x)), \tag{3c}$$

$$\rho_{\phi_1 \mathcal{U}_I \phi_2}(t, x) = \max_{\tau \in t+I} \big( \min(\rho_{\phi_2}(\tau, x), \min_{s \in [t, \tau]} \rho_{\phi_1}(s, x)) \big). \tag{3d}$$

The reader is referred to [10], [15] for a detailed treatment of STL semantics.

## III. PROBLEM STATEMENT

Consider the dynamics of a robot operating in space and underwater, denoted with subscripts $sp$ and $uw$, respectively. We denote the dynamics with $\circ \in \{sp, uw\}$ as,

$$\mathcal{M}_\circ(\boldsymbol{x}_\circ, \boldsymbol{u}_\circ) = \dot{\boldsymbol{x}}_\circ = f_\circ(\boldsymbol{x}_\circ) + g_\circ(\boldsymbol{x}_\circ)\boldsymbol{u}_\circ + C_\circ(\boldsymbol{x}_\circ)\boldsymbol{d}_\circ, \tag{4}$$

where we model the full 6-DoF dynamics with actuation in force and torque. The state is given by $\boldsymbol{x}_\circ \in \mathcal{X}_\circ \subseteq \mathbb{R}^9 \times \mathbb{S}^3$, the control input by $\boldsymbol{u}_\circ \in \mathcal{U}_\circ \subseteq \mathbb{R}^6$, and the disturbance terms by $\boldsymbol{d}_\circ \in \mathcal{D}_\circ \subseteq \mathbb{R}^6$ which represent bounded additive uncertainties and external perturbations. $f_\circ : \mathcal{X}_\circ \to \mathcal{X}_\circ$ represents the drift term, $g_\circ : \mathcal{X}_\circ \to \mathbb{R}^{12\times 6}$ the the actuation matrix and $C_\circ : \mathcal{X}_\circ \to \mathbb{R}^{12\times 6}$ the disturbance mapping. We use $\mathcal{M}_\circ(\boldsymbol{x}_\circ, \boldsymbol{u}_\circ)$ as shorthand for the dynamics of a platform.

Although underwater platforms can approximate space-like conditions through near-neutral buoyancy, the two domains differ significantly in their dynamics, parameter uncertainties, and disturbance profiles. We require that satisfaction of a specification in the underwater robot, $\phi_{uw}$, under disturbances from set $\mathcal{D}_{uw}$ implies satisfaction of the space mission $\phi_{sp}$ under disturbances from set $\mathcal{D}_{sp}$. In other words,

$$\hat{\boldsymbol{x}}_{uw} \models \phi_{uw}, \forall \boldsymbol{d}_{uw} \in \mathcal{D}_{uw} \implies \hat{\boldsymbol{x}}_{sp} \models \phi_{sp}, \forall \boldsymbol{d}_{sp} \in \mathcal{D}_{sp}, \tag{5}$$

where $\boldsymbol{x}_\circ \models \phi_\circ$ denotes that the trajectory $\boldsymbol{x}_\circ$ satisfies STL specification $\phi_\circ$. We tackle this challenge by introducing the *disturbance-robustness degree* for a mission, defined as the maximal scaling factor of the disturbance set that may affect the robot while still satisfying $\phi_\circ$. Motion planning for $\boldsymbol{x}_{sp}$ is first performed to satisfy $\phi_{sp}$, and the disturbance-robustness degree is quantified. The specification is then transformed into an underwater counterpart $\phi_{uw}$ such that executing $\phi_{uw}$ has an equivalent disturbance-robustness degree for the underwater robot. In this way, the two environments can be compared fairly, and the underwater execution provides systematic validation of the space mission.

## IV. Models

We consider three dynamics models: a linear space robot model $\mathcal{M}_{spL}$, a nonlinear space robot model $\mathcal{M}_{sp}$, and a nonlinear underwater robot model $\mathcal{M}_{uw}$. The nonlinear models are modeled as a rigid body with state vector $\boldsymbol{x}_\circ = [\boldsymbol{p},\, \mathbf{q},\, \boldsymbol{v},\, \boldsymbol{\omega}]$, where $\circ \in \{sp, uw\}$. Here, $\boldsymbol{p} \in \mathbb{R}^3$ is the robot position in the inertial frame, $\mathbf{q} \in \mathbb{S}^3$ is the body attitude represented as a quaternion relative to the inertial frame, and $\boldsymbol{v}, \boldsymbol{\omega} \in \mathbb{R}^3$ are the linear and angular velocities in the body frame, respectively. The robots are actuated through body-frame force and torque inputs $\mathbf{w} = [\boldsymbol{F}_{\text{act}},\, \boldsymbol{\tau}_{\text{act}}]$, generated by the robot's thrusters, each applying a force $\mu_i \in [\mu_{\min}, \mu_{\max}]$ along its axis. The total wrench is $\mathbf{w} = \boldsymbol{G}\boldsymbol{\mu}$, with $\boldsymbol{\mu} = [\mu_1 \ \dots \ \mu_m]^\mathsf{T} \in \mathbb{R}^m$ being the vector of thruster forces, and $\boldsymbol{G} = \begin{bmatrix}\boldsymbol{G}_F^\mathsf{T} & \boldsymbol{G}_\tau^\mathsf{T}\end{bmatrix}^\mathsf{T}$ the control allocation matrix mapping them to body-frame wrenches. The robots are also subject to external disturbances, $\boldsymbol{d} = [\boldsymbol{F}_d,\, \boldsymbol{\tau}_d]$, where $\boldsymbol{F}_d$ and $\boldsymbol{\tau}_d$ are in inertial and body-frame, respectively. We define the robot configuration $\boldsymbol{\eta} = [\boldsymbol{p}, \mathbf{q}]^T$ and body-frame velocity twist $\boldsymbol{\nu} = [\boldsymbol{v}, \boldsymbol{\omega}]^T$. The full nonlinear model of both the space and underwater robots can then be expressed in the standard form

$$\begin{aligned} \dot{\boldsymbol{\eta}} &= \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu}, \\ \boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) &= \mathbf{w} + \boldsymbol{F}(\boldsymbol{\eta})\boldsymbol{d}, \end{aligned} \tag{6}$$

where

$$\boldsymbol{J}(\boldsymbol{\eta}) = \begin{bmatrix}\boldsymbol{R}(\mathbf{q}) & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \frac{1}{2}\boldsymbol{E}(\mathbf{q})\end{bmatrix}, \boldsymbol{F}(\boldsymbol{\eta}) = \begin{bmatrix}\boldsymbol{R}(\mathbf{q})^\mathsf{T} & \mathbf{0}_{3\times 3} \\ \mathbf{0}_{3\times 3} & \boldsymbol{I}_3\end{bmatrix}, \tag{7}$$

with $\boldsymbol{J}(\boldsymbol{\eta})$ mapping body-frame velocities to inertial-frame rates, with $\boldsymbol{E}(\mathbf{q})$ being a standard matrix for quaternion kinematics, and $\boldsymbol{F}(\boldsymbol{\eta})$ mapping external disturbances to body-frame rates. $\boldsymbol{M}$ is the generalized mass–inertia matrix; $\boldsymbol{C}(\boldsymbol{\nu})$ the Coriolis and centripetal terms; $\boldsymbol{D}(\boldsymbol{\nu})$ the hydrodynamic damping; $\boldsymbol{g}(\boldsymbol{\eta})$ the restoring forces such as gravity and buoyancy; and $\boldsymbol{R}(\mathbf{q})$ the rotation matrix. For detailed expressions of these terms for $\mathcal{M}_{uw}$, we refer the reader to [16].

To align with the model description in (4), we can express the dynamics functions for the full nonlinear model of the two robots as below

$$\begin{aligned} f_\circ(\boldsymbol{x}_\circ) &= \begin{bmatrix}\boldsymbol{J}(\boldsymbol{\eta}) \\ \boldsymbol{M}^{-1}\left(-\boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{g}(\boldsymbol{\eta})\right)\end{bmatrix}, \\ g_\circ(\boldsymbol{x}_\circ) &= \begin{bmatrix}\mathbf{0}_{7\times 6} \\ \boldsymbol{M}^{-1}\end{bmatrix}, \quad C_\circ(\boldsymbol{x}_\circ) = \begin{bmatrix}\mathbf{0}_{7\times 6} \\ \boldsymbol{M}^{-1}\boldsymbol{F}(\boldsymbol{\eta})\end{bmatrix}. \end{aligned} \tag{8}$$

The $\mathcal{M}_{sp}$–model reduces to a classical rigid-body model with only the inertia and Coriolis terms. In contrast, the $\mathcal{M}_{uw}$–model also includes added mass terms and the corresponding hydrodynamic coupling in the inertia and Coriolis matrices, as well as dissipative drag and restoring forces from gravity and buoyancy. Finally, the $\mathcal{M}_{spL}$–model, used for high-level planning, is a simplified linearized version of $\mathcal{M}_{sp}$, expressed with Euler angles and assuming decoupled translational and rotational dynamics.

## V. Planning

We consider an STL specification that prescribes the space mission via desired behavior over state and time. We enable underwater validation of the space mission by generating motion plans for both the space and underwater robots that satisfy the specification with an equivalent disturbance-robustness degree. This approach reconciles the fundamental differences in their dynamics.

At the high-level planning, we consider a linear model of the space robot that enables globally optimal solutions w.r.t. the STL specification. We then reason about how this solution is sound when executed on the true nonlinear space robot system and can be locally adapted to the nonlinear underwater robot system. We assume that all predicate functions in $\phi_{sp}$ are piecewise-linear functions of the position and orientation.

### A. Disturbance Robustness as a Metric

We utilize the size of the maximal disturbance set (via scalar multiplication) as an optimization metric in the planner to promote robustness against unforeseen disturbances. The maximal disturbance set is then the largest set from which disturbances may be drawn that the platform can counteract while satisfying the specification $\phi_\circ$.

At the planning stage, we assume that the controller can track the planned trajectory with reasonably small error

despite external disturbances. To ensure this, we impose the following assumptions:

*Assumption 1 (Matched disturbances):* The disturbance is *matched* [17], meaning it acts through the same channels as the control input and there exists some $\boldsymbol{K}$ such that for any $d$ we can write $g(\boldsymbol{x})\boldsymbol{K}d = C(\boldsymbol{x})d$, or equivalently:

$$\forall \boldsymbol{x}_\circ \in \mathcal{X}_\circ, \ \ \exists \boldsymbol{K} \ \text{ s.t. } \ g_\circ(\boldsymbol{x}_\circ) = C_\circ(\boldsymbol{x}_\circ)\boldsymbol{K}^{-1}. \tag{9}$$

*Assumption 2 (Sufficient control authority):* The available control authority is large enough to counteract any worst-case disturbances:

$$-\mathcal{U}_\circ \supseteq \boldsymbol{K}\mathcal{D}_\circ. \tag{10}$$

From Assumptions 1–2, disturbances effectively reduce the usable control authority of the robot. Intuitively, part of the control input must always be "reserved" to counteract disturbances, so only a smaller portion of the nominal control set can be used for planning. We capture this via the *effective* (or *robust*) control set

$$\overline{\mathcal{U}}_\circ = \mathcal{U}_\circ \ominus \boldsymbol{K}\mathcal{D}_\circ, \tag{11}$$

where $\ominus$ denotes the Minkowski difference. This set represents all control inputs that remain available after accounting for the worst-case disturbances.

The maximal permissible disturbance set (equivalently, the minimal effective control set) that still allows the robot to satisfy its mission can be found by, following the notion of *maximal disturbance robustness* [14], finding the maximal scalar disturbance robustness degree $\alpha^*$.

$$\alpha^* = \max \alpha, \ \ \text{s.t.} \ \ \ \boldsymbol{u} \in \mathcal{U}_\circ \ominus \alpha\boldsymbol{K}\mathcal{D}_\circ, \ \ \boldsymbol{x} \models \phi_\circ \tag{12}$$

with the maximal admissible disturbance set $\mathcal{D}^* = \alpha^*\mathcal{D}$.

This metric (i) identifies the largest disturbance set that preserves feasibility and (ii) enables transfer of the specification to a different platform—such as an underwater platform—by ensuring an equivalent *degree of difficulty*. If online estimates of disturbances remain within $\mathcal{D}^*_{uw}$, successful operation in the underwater domain provides evidence of feasibility in the space scenario under $\mathcal{D}^*_{sp}$ (see Sec. V-C).

### B. Space Plan Synthesis

First, we consider $\mathcal{M}_{spL}$ with rotational dynamics in Euler angles $\boldsymbol{\theta}$, with $\ddot{\boldsymbol{\theta}} = \boldsymbol{\alpha}_\omega$ as planning-level inputs, and neglecting nonlinear coupling. Translational dynamics are expressed in the world frame, and are hence linear.

In the nonlinear model, the feasible wrench set rotates with attitude since each thruster is fixed in the body frame. Feasibility is ensured by imposing conservative control bounds: the largest symmetric origin-centered polytope (within a chosen class) is always contained in the rotated wrench sets. Our procedure (for forces, but analogous for torques) is as follows: i) Represent the feasible forces as a zonotope,$\mathcal{F} = \{R(\mathbf{q})\boldsymbol{G}_f\boldsymbol{\mu} | \boldsymbol{\mu}_{\min} \le \boldsymbol{\mu} \le \boldsymbol{\mu}_{\max}\}$, where $\boldsymbol{G}_f = [\boldsymbol{d}_1 \dots \boldsymbol{d}_m]$; ii) Obtain the vertices of $\mathcal{F}$ by mapping each vertex of the input hypercube $\boldsymbol{\mu}_{\min} \le \boldsymbol{\mu} \le \boldsymbol{\mu}_{\max}$ and from those compute its polytope $\mathcal{H}$-representation [18] $\boldsymbol{H}_f\boldsymbol{\mu} \le \boldsymbol{b}_f$ with $\boldsymbol{H}_f = \begin{bmatrix}\boldsymbol{h}_{f1} & \dots & \boldsymbol{h}_{fm}\end{bmatrix}^\mathsf{T}$; iii) Find the largest inscribed sphere (a simple variant of the Chebyshev center problem [19]), whose radius can be efficiently computed as $\min_i(b_{fi}/\|\boldsymbol{h}_{fi}\|)$; iv) Approximate this sphere with an inscribed polytope (e.g., cube, icosahedron), yielding tractable linear constraints for planning.

In what follows, $\mathcal{U}_{spL}$ is such an inscribed polytope. This linearized formulation enables us to cast the planning problem—maximizing disturbance robustness under STL specifications for the space robot—as a Mixed-Integer Linear Program.

$$\min_{\boldsymbol{x}_{spL},\boldsymbol{u}_{spL},\alpha} \ -\alpha - c_1\rho_{\phi_{sp}}(t,\boldsymbol{x}_{spL}) + c_2 J(\boldsymbol{u}_{spL}), \tag{13a}$$

$$\text{s.t.} \quad \boldsymbol{x}^{k+1}_{spL} = \boldsymbol{x}^k_{spL} + \mathcal{M}_{spL}(\boldsymbol{x}^k_{spL}, \boldsymbol{u}^k_{spL})\Delta t, \tag{13b}$$

$$\boldsymbol{u}_{spL} \in \mathcal{U}_{spL} \ominus \alpha\boldsymbol{K}\mathcal{D}_{spL}, \tag{13c}$$

With the linearized space robot dynamics we obtain a trajectory, $\boldsymbol{x}^*_{sp}$, that, with the maximal disturbance-robustness degree $\alpha$, satisfies $\phi_{sp}$ with maximal spatial robustness $\rho_{\phi_{sp}}(t,\boldsymbol{x})$. The cost term $J(\boldsymbol{u}_{sp}) = \sum_{i,j}|u^j_i|$ approximates total propellant consumption which encourages efficiency[1] and is of particular interest in space applications where fuel is limited. The coefficients $c_1$ and $c_2$ are defined such that $c_1 \gg c_2$, and $c_1 \gg \alpha$ to ensure that satisfying the specification with maximal robustness is the primary objective of the optimization problem. The constraint (13c) ensures the robot respects the effective control bounds. Given that $\mathcal{U}_{spL}$ is a convex polytope in the form $\{u : Hu \le b\}$ and assuming $\mathcal{D}$ is a zero-centered hyperrectangle, this constraint can be embedded via $\{u \mid Hu \le \check{b}\}$ with $\check{b}_i = b_i - \alpha\sum(|\boldsymbol{K}^{-T}H_{i,:}|\overline{\mathcal{D}})$ which is linear in its decision variable $\alpha$.

### C. Underwater Plan Synthesis

To achieve a plan, $\boldsymbol{x}_{uw}$ with an equal degree of difficult to $\boldsymbol{x}_{sp}$, we adjust the mission duration of $\boldsymbol{x}^*_{sp}$ such that $\boldsymbol{x}^*_{uw}$ has a matching disturbance robustness degree $\alpha^*$ when executed on $\mathcal{M}_{uw}$. We adjust the time scaling to find a minimally deviating plan by solving

$$\begin{aligned}\min_{\boldsymbol{x}_{uw},\boldsymbol{u}_{uw},\Delta t} \ & \Delta t, \\ \text{s.t.} \quad & \boldsymbol{x}^{k+1}_{uw} = \boldsymbol{x}^k_{uw} + \mathcal{M}_{uw}(\boldsymbol{x}^k_{uw}, \boldsymbol{u}^k_{uw})\Delta t \\ & (\boldsymbol{p}^k_{uw}, \mathbf{q}^k_{uw}) = (\boldsymbol{p}^k_{sp}, \mathbf{q}^k_{sp}), \\ & \boldsymbol{u}_{uw} \in \mathcal{U}_{uw} \ominus \alpha^*\boldsymbol{K}\mathcal{D}_{uw},\end{aligned} \tag{14}$$

where $(\boldsymbol{p}^k_{sp}, \mathbf{q}^k_{sp})$ and $\alpha^*$ are solutions of Eq. (13). The resulting trajectory satisfies $\phi_{uw}$, which is a time-scaled version of $\phi_{sp}$ (scaling all intervals in $\phi_{uw}$ with $\frac{\Delta t^*}{\Delta t}$).

This plan has an equal disturbance robustness degree which means that regardless of whether $\mathcal{M}_{uw}$ is a *faster* or *slower* platform, a motion plan is available for it which can be used to verify operation in space upon observation of these disturbances.

[1]This comes from the fact that the thrust from thruster $i$ is given by $\boldsymbol{F}_i = u_{\mathrm{dc},i}\,\dot{m}_i\boldsymbol{v}_{e,i}$, where $u_{\mathrm{dc},i} \in [0,1]$ is the duty cycle, $\dot{m}_i$ the propellant mass flow, and $\boldsymbol{v}_{e,i}$ the exhaust velocity. Thus, total propellant consumption is proportional to $\sum_i u_{\mathrm{dc},i}$.

## VI. CONTROL

The planned trajectories include expected bounds on external disturbances which an Extended Kalman Filter (EKF) estimates online for feed-forward control and analysis. For underwater validation, we use an MPC with a feedback equivalence law that enforces closed-loop space-robot dynamics. The MPC for the underwater validation is then defined with respect to the space dynamics, incorporating space-EKF disturbance estimates and tightened actuator limits accounting for the closed-loop model compensation. Underwater validation then provides practical evidence of feasibility on the space robot, provided that estimated disturbances remain within the assumed bounds.

### A. Disturbance Estimation

The EKF uses $\mathcal{M}_\circ$ with a reduced state augmented with the disturbance force and torque $\boldsymbol{x}^k_{\text{EKF}} = \begin{bmatrix}\boldsymbol{v}^k, & \boldsymbol{\omega}^k, & \boldsymbol{F}^k_d, & \boldsymbol{\tau}^k_d\end{bmatrix}$, and measurements $\boldsymbol{z}^k = \begin{bmatrix}\boldsymbol{v}^k, & \boldsymbol{\omega}^k\end{bmatrix}$. Actuator wrenches are assumed equal to commanded values. Disturbances $\boldsymbol{d} = [\boldsymbol{F}_d, \boldsymbol{\tau}_d]$ are modeled as constant over the MPC horizon and as random-walk states in the EKF with zero-mean Gaussian process and measurement noise. The disturbance estimates are fed forward to the MPC via $\boldsymbol{u}^k_{\text{ref}} = -\hat{\boldsymbol{d}}, \forall k \in [0; N-1]$ so the input cost is minimized when the control cancels the estimated disturbance. See [2] for details.

### B. Space robot MPC

The MPC for the space robot is formulated as

$$\begin{aligned}\min_{\boldsymbol{x}^{0:N}, \boldsymbol{u}^{0:N},} \; & \sum_{i=0}^{N-1} ||\boldsymbol{x}^k_{sp} - \boldsymbol{x}^k||_Q + \sum_{i=0}^{N-1} ||\boldsymbol{u}^k_{ref} - \boldsymbol{u}^k||_R + \\ & ||\boldsymbol{x}^N_{sp} - \boldsymbol{x}^N||_P, \\ \text{s.t.} \quad & \boldsymbol{x}^{k+1} = \text{RK4}(\mathcal{M}_{sp}(\boldsymbol{x}^k, \boldsymbol{u}^k, \hat{\boldsymbol{d}})), \\ & \boldsymbol{x}^k \in \mathcal{W}_{\text{free}} \\ & \boldsymbol{u}^k \in \mathcal{U}_{sp}\end{aligned} \tag{15}$$

where $Q \succeq 0$ and $R \succ 0$ are stage cost weight matrices and $P \succeq 0$ is the terminal weight matrix, RK4 denotes the Runge-Kutta 4 integration scheme [20], and $\mathcal{W}_{\text{free}}$ denotes the workspace. Obstacle avoidance is captured in $\phi_\circ$, and $\boldsymbol{x}_{sp}$ is obtained from the planner.

### C. Dynamical Equivalency

We apply feedback equivalence control to $\mathcal{M}_{uw}$ to behave as $\mathcal{M}_{sp}$ in closed-loop. This compensates for effects such as hydrodynamics and may even introduce orbital dynamics to $\mathcal{M}_{uw}$. The transformation is defined as

$$\boldsymbol{u}_{sp\to uw}(\boldsymbol{x}, \boldsymbol{u}) = g_{uw}(\boldsymbol{x})^\dagger (f_{sp}(\boldsymbol{x}) + g_{sp}(\boldsymbol{x})\boldsymbol{u} - f_{uw}(\boldsymbol{x})) \tag{16}$$

where $g^\dagger_{uw}$ denotes the pseudo-inverse. This controller ensures $\mathcal{M}_{uw}$ internally behaves like $\mathcal{M}_{sp}$: the output of $\boldsymbol{u}_{sp\to uw}$ is the input that needs to be given to the underwater robot such that it behaves as if $\boldsymbol{u}$ is applied to the space robot.

### D. Underwater robot MPC

The MPC for the underwater robot is defined as

$$\begin{aligned}\min_{\boldsymbol{x}^{0:N}, \boldsymbol{u}^{0:N},} \; & \sum_{i=0}^{N-1} ||\boldsymbol{x}^k_{uw} - \boldsymbol{x}^k||_Q + \sum_{i=0}^{N-1} ||\boldsymbol{u}^k_{uw} - \boldsymbol{u}^k||_R + \\ & ||\boldsymbol{x}^N_{uw} - \boldsymbol{x}^N||_P, \\ \text{s.t.} \quad & \boldsymbol{x}^{k+1} = \text{RK4}(\mathcal{M}_{sp}(\boldsymbol{x}^k, \boldsymbol{u}^k, \hat{\boldsymbol{d}})), \\ & \boldsymbol{x}^k \in \mathcal{W}_{\text{free}} \\ & \boldsymbol{u}_{sp\to uw}(\boldsymbol{x}^k, \boldsymbol{u}^k) \in \mathcal{U}_{uw}\end{aligned} \tag{17}$$

which utilizes the dynamics of the space robot with the tightened constraints from the feedback-equivalence control law in Eq. (16), similar to [21]. The final control input to the underwater robot is then obtained as $\boldsymbol{u}_{uw} = \boldsymbol{u}_{sp\to uw}(\boldsymbol{x}^0, \boldsymbol{u}^0)$. While Eq. (17) utilizes a disturbance estimator using $\mathcal{M}_{sp}$, the requirement of $\boldsymbol{d} \in \mathcal{D}_{uw}$ is obtained from observing a parallel disturbance estimator using $\mathcal{M}_{uw}$.

## VII. VALIDATION

### A. Experimental Setup

Experiments use an underwater robot as the validation of a space-analog platform, all running an identical software stack: a Pixhawk 6X–mini microcontroller, running PX4 [22], performs sensor fusion of onboard IMU and external motion-capture data, and computes low-level control by allocating commanded forces and torques to PWM signals for the thrusters. Offboard computers solve Eq. (15) or Eq. (17) with ACADOS [23], providing wrench commands that are then transmitted to the microcontroller. For all our experiments the timestep and horizon length are identical but weight matrices $Q$, $R$, and $P$ are altered between Eq. (15) and Eq. (17) to improve stability of the controllers.

*Underwater Setup:* For underwater validation, we use a BlueRobotics BlueROV2 Heavy in the KTH Marinarium testing facilities [24]. The dynamics model is parameterized according to [25], with actuator limits reduced to avoid cavitation: $\overline{\mathcal{U}}_{uw} = -\underline{\mathcal{U}}_{uw} = [21, 21, 30, 20, 11, 17]$, and disturbance bounds $\overline{\mathcal{D}}_{uw} = -\underline{\mathcal{D}}_{uw} = [10, 10, 10, 5, 5, 5]$.

*Space Setup:* Validation involves (i) The ATMOS air-bearing platform [2] for 3-DoF frictionless 2D dynamics with compressed-air thrusters and (ii) A Basilisk simulation [26] with a CubeSat parameterized to match the space robot's inertia and actuator configuration, placed in a 700 km circular equatorial orbit and propagated in the rotating Hill frame. A bridge interface exposes Basilisk's internal message system and provides a PX4-compatible communication layer [27]. Both environments share actuator limits $\overline{\mathcal{U}}_{sp} = -\underline{\mathcal{U}}_{sp} = [1.42, 1.42, 1.42, 0.24, 0.24, 0.24]$, which is below the hardware maximum to allow simultaneous force and torque saturation, and $\overline{\mathcal{D}}_{sp} = -\underline{\mathcal{D}}_{sp} = [0.64, 0.64, 0.64, 0.01, 0.01, 0.01]$, where forces are assumed from floor imperfections, and torques are assumed to be small.

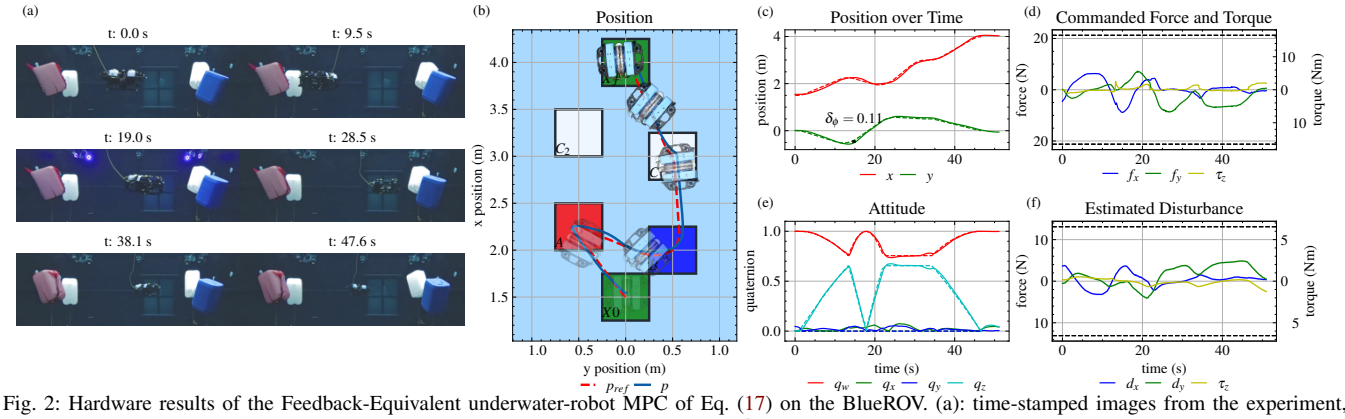


Fig. 2: Hardware results of the Feedback-Equivalent underwater-robot MPC of Eq. (17) on the BlueROV. (a): time-stamped images from the experiment, (b) planned and executed trajectory, (c) position and the maximal spatial deviation $\delta_{\phi_{uw}} = 0.11 \leq \rho_{\phi_{uw}} = 0.25$ (indicating satisfaction of $\phi_{uw}$), (d) commanded force and torque with its actuation limits in dotted lines, (e) attitude, (f) estimated force and torque disturbance $\hat{\boldsymbol{d}}_{uw}$ with the admissible disturbance bounds $\alpha^* \mathcal{D}_{uw}$. The observed disturbances again being bounded by the disturbance robustness degree $\alpha^* \mathcal{D}_{uw}$ with a satisfied specification provides experimental evidence of operation in the space environment, shown in Fig. 3.

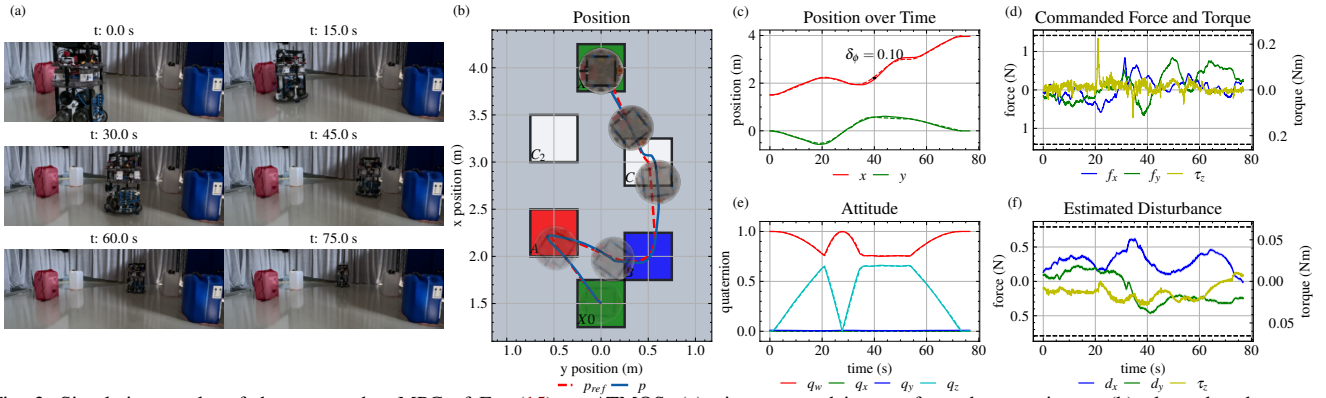


Fig. 3: Simulation results of the space-robot MPC of Eq. (15) on ATMOS. (a): time-stamped images from the experiment, (b) planned and executed trajectory in the 2D plane, (c) position and the maximal spatial deviation $\delta_{\phi_{sp}} = 0.10 \leq \rho_{\phi_{sp}} = 0.25$ (indicating satisfaction of $\phi_{sp}$), (d) commanded force and torque with its actuation limits in dotted lines, (e) attitude, (f) estimated force and torque disturbances $\hat{d}_{sp}$ with the admissible disturbance bounds $\alpha^* \mathcal{D}_{sp}$. The experimental evidence of the underwater validation experiment is translated to the space platform as long as the estimated force and torque disturbances acting on the CubeSat are within the admissible disturbance bounds $\alpha^* \mathcal{D}_{sp}$ which is shown in (f).

### B. Results

We perform validation in two inspection-style experiments: 1) a 2D mission, comparing the BlueROV with the physical space robot, and 2) a 3D mission, comparing the BlueROV with the simulated CubeSat.

*1) Planar Specification:* Consider an intra-vehicular space robot (e.g., the Astrobee [28] in the ISS) tasked with visually inspecting certain areas of interest within certain time intervals. The specification for the space robot is

$$\phi_{sp} = \diamondsuit_{[20,25]}(\boldsymbol{\eta} \in A) \wedge \diamondsuit_{[0,37]}(\boldsymbol{\eta} \in B) \wedge (\square_{[50,55]}(\boldsymbol{\eta} \in C_1) \vee \square_{[50,55]}(\boldsymbol{\eta} \in C_2)) \quad (18)$$

with $\boldsymbol{\eta}(0) \in X_0$ and $\boldsymbol{\eta}(75) \in X_f$. The regions of interest $X_0$, $X_f$, $A$, $B$, $C_1$, and $C_2$ (including desired orientations) are defined in Tab. I and shown in Fig. 3. This specification captures that both the red and blue region should be visited within a certain interval while one of the white regions has to be investigated for at least 5 seconds.

We first obtain the nominal motion plan for the space robot by solving Eq. (13). The result is shown in Fig. 3 with the disturbance-robustness degree of $\alpha^* = 1.31$, indicating that $\exists \boldsymbol{u} \in \mathcal{U}, s.t. \forall \boldsymbol{d} \in 1.31 \cdot \mathcal{D}, \boldsymbol{x}_{sp} \models \phi_{sp}$. We then obtain the scaled motion plan for the underwater robot according to Eq. (14) shortening the mission duration $t_f$ from 75 to 65.5 seconds (a 1.15× speedup as the control limits of the BlueROV model are higher than the space robot), making the underwater plan have an identical disturbance-robustness degree of $\alpha^* = 1.31$, shown in Fig. 2. During execution on the space robot, we solve Eq. (15) with the estimated disturbance from the space robot EKF. As this control strategy is to be validated on the different underwater robot, we solve the feedback-equivalent form in Eq. (17), which considers the space robot model and a space robot EKF. The experimental results on the BlueROV in Fig. 2. Fig. 2(f) indicate that the estimated disturbances on the BlueROV are within the $\alpha^* \mathcal{D}_{uw}$ bounds. Fig. 2d shows the control inputs and the $\overline{\mathcal{U}}_{uw} = \mathcal{U}_{uw} \ominus \alpha^* \mathcal{D}_{uw}$ bounds. The results, together with the assumptions from the planner, imply operation on the space robot as $\hat{\boldsymbol{d}}_{sp} \in \mathcal{D}^*_{sp}$, as shown in Fig. 3.

*2) Complex Inspection Task:* Consider a task where an extra-vehicular CubeSat needs to inspect the sides of an unknown satellite. The specification captures that each axis of the satellite is to be observed, e.g., the $x$ axis of the spacecraft has to be observed from the front or the left side.

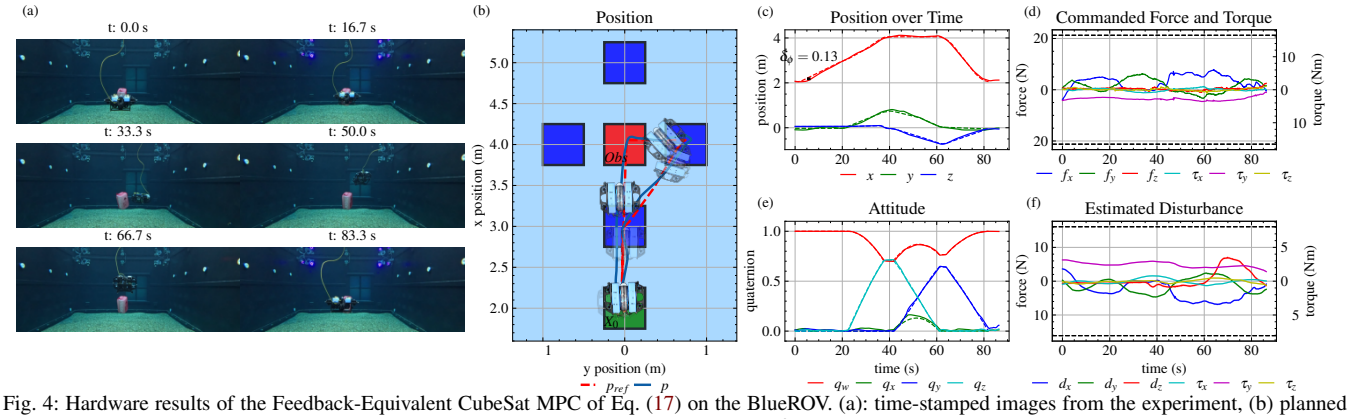


Fig. 4: Hardware results of the Feedback-Equivalent CubeSat MPC of Eq. (17) on the BlueROV. (a): time-stamped images from the experiment, (b) planned and executed trajectory in the 2D plane, (c) position and the maximal spatial deviation $\delta_{\phi_{uw}} = 0.13 \leq \rho_{\phi_{uw}} = 0.19$ (indicating satisfaction of $\phi_{uw}$), (d) commanded force and torque with its actuation limits in dotted lines, (e) attitude, (f) estimated force and torque disturbances $\hat{d}_{uw}$ with the admissible disturbance bounds $\alpha^* \mathcal{D}_{uw}$. The observed disturbances again being bounded by the disturbance robustness degree $\alpha^* \mathcal{D}_{uw}$ with a satisfied specification provides experimental evidence of operation in the space environment, shown in Fig. 5.

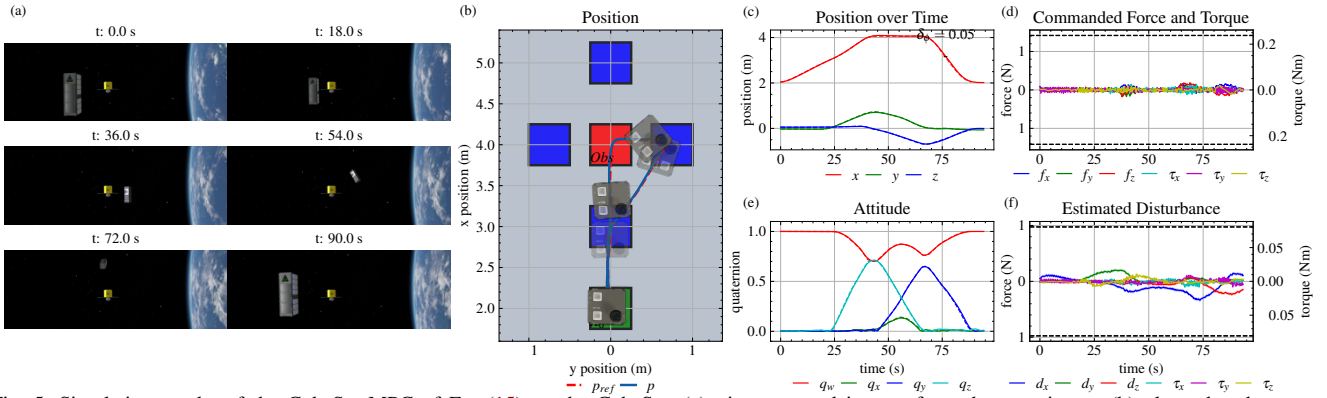


Fig. 5: Simulation results of the CubeSat MPC of Eq. (15) on the CubeSat. (a): time-stamped images from the experiment, (b) planned and executed trajectory in the 2D plane, (c) position and the maximal spatial deviation $\delta_{\phi_{sp}} = 0.05 \leq \rho_{\phi_{sp}} = 0.19$ (indicating satisfaction of $\phi_{sp}$), (d) commanded force and torque with its actuation limits in dotted lines, (e) attitude, (f) estimated force and torque disturbances $\hat{d}_{sp}$ with the admissible disturbance bounds $\alpha^* \mathcal{D}_{sp}$. The experimental evidence of the underwater validation experiment is translated to the space platform as long as the estimated force and torque disturbances acting on the CubeSat are within the admissible disturbance bounds $\alpha^* \mathcal{D}_{sp}$, shown in (f).

The specification for the space robot is

$$\phi_{sp} = \bigwedge_{k=x,y,z} \Diamond_{[\underline{t}_k, \overline{t}_k]} (\boldsymbol{\eta} \in \underline{A}_k \vee \boldsymbol{\eta} \in \overline{A}_k) \wedge \Box_{[0,90]} (\boldsymbol{\eta} \notin \text{Obj}) \tag{19}$$

with $\boldsymbol{\eta}(0) \in X_0$ and $\boldsymbol{\eta}(90) \in X_0$. The regions of interest $\underline{A}_k$ and $\overline{A}_k$ (including required rotation) and their time intervals are specified in Tab. I. The nominal motion plan has a disturbance-robustness degree of $\alpha^* = 1.61$. The scaled motion plan for the underwater robot reduces the mission duration $t_f$ from 90 to 80.4 seconds (a speed-up of $1.11\times$, less significant due to the torque requirements on the BlueROV model).

The results on the BlueROV in Fig. 4 show that the torque disturbance is particularly large around $y$ (likely due to a model mismatch of the center of buoyancy and center of mass) yet still within the permissible disturbance bounds of $\alpha^* \mathcal{D}_{uw}$. The satisfaction $\phi_{uw}$ with disturbances within this scaled set shows experimental evidence that the planning and control architecture will operate successfully in space, under the same disturbance robustness bounds. Using the EKF, we are able to quantify this modeling error (plus any additional disturbances from, for example, the tether). In the validation of the CubeSat mission in Basilisk in Fig. 5, the disturbances are significantly below $\alpha^* \mathcal{D}$.

### C. Discussion

The results demonstrate that space planning and control can be validated on dynamically equivalent underwater platforms, by quantifying permissible disturbances via the disturbance robustness degree while the estimator verifies online whether this assumption holds.

There are some limitations however. First, the guarantees hold with the assumptions in Sec. V, of which instantaneously detecting and counteracting the disturbance are the most significant. Additionally, space missions can only be validated with underwater robots if the underwater model is sufficiently accurate, such that its errors can be represented as additive disturbances within the maximal disturbance robustness degree. However, the permissible level of mismatch is not identified. Lastly, uncertainty and different characteristics of estimation pipelines are currently not considered as ground-truth state estimation is assumed.

## VIII. CONCLUSIONS

We presented an experimental framework for the validation of space robotics in underwater environments. Using disturbance robustness as an optimization metric, we

TABLE I: Details on the STL specifications from Eq. (18) and Eq. (19).

| | center | widths | dimensions* | interval |
|---|---|---|---|---|
| A | $[2.25, -0.5, -\frac{\pi}{2}]$ | $[0.5, 0.5, \frac{\pi}{4}]$ | $[x,y,\psi]$ | [20,25] |
| B | $[2, 0.5, -\frac{\pi}{2}]$ | $[0.5, 0.5, \frac{\pi}{4}]$ | $[x,y,\psi]$ | [0,37] |
| $C_1$ | $[3, 0.5, \frac{\pi}{2}]$ | $[0.5, 0.5, \frac{\pi}{4}]$ | $[x,y,\psi]$ | [50,55] |
| $C_2$ | $[3.25, -0.5, \frac{\pi}{2}]$ | $[0.5, 0.5, \frac{\pi}{4}]$ | $[x,y,\psi]$ | [50,55] |
| $\underline{A}_x$ | $[3, 0, 0, 0, 0]$ | $[0.5, 0.5, \frac{\pi}{8}, \frac{\pi}{8}]$ | $[x,y,z,\theta,\psi]$ | [20,25] |
| $\overline{A}_x$ | $[5, 0, 0, 0, \pi]$ | $[0.5, 0.5, \frac{\pi}{8}, \frac{\pi}{8}]$ | $[x,y,z,\theta,\psi]$ | [20,25] |
| $\underline{A}_y$ | $[4, -0.75, 0, 0, \frac{\pi}{2}]$ | $[0.5, 0.5, \frac{\pi}{8}, \frac{\pi}{8}]$ | $[x,y,z,\theta,\psi]$ | [40,45] |
| $\overline{A}_y$ | $[4, 0.75, 0, 0, \frac{\pi}{2}]$ | $[0.5, 0.5, \frac{\pi}{8}, \frac{\pi}{8}]$ | $[x,y,z,\theta,\psi]$ | [40,45] |
| $\overline{A}_z$ | $[4, 0, 0.75, 0, -\frac{\pi}{2}, 0]$ | $[0.5, 0.5, \frac{\pi}{8}, \frac{\pi}{8}]$ | $[x,y,z,\theta,\psi]$ | [65,70] |

*Here, $\theta$ and $\psi$ denote pitch and yaw.

generate motion plans that satisfy a high-level specification while being of equal disturbance robustness degree for the space robot and the underwater analogue. This ensures that, regardless of the dynamical and environmental differences, a motion plan for the validation platform can be obtained. Estimating the overall disturbances to the platform in the underwater analogue platform then provides practical evidence that planning and control methods are able to operate in the inaccessible space environment as long as these disturbances are bounded by the maximal disturbance set.

Our theoretical contribution relies on instantaneous disturbance estimation and rejection. Future work will relax these assumptions. Additionally, we will extend our method with time-varying disturbance robustness metrics, complex and multi-robot scenarios such as docking and transportation, and enable human-in-the-loop validation of space missions.

## REFERENCES


[1] N. Uyama, H. Lund, K. Asakimori, Y. Ikeda, D. Hirano, H. Nakanishi, and K. Yoshida, "Integrated experimental environment for orbital robotic systems, using ground-based and free-floating manipulators," in *2010 IEEE/SICE International Symposium on System Integration*. IEEE, 2010, pp. 108–113.

[2] P. Roque, S. Phodapol, E. Krantz, J. Lim, J. Verhagen, F. J. Jiang, D. Dörner, H. Mao, G. Tibert, R. Siegwart, I. Stenius, J. Tumova, C. Fuglesang, and D. V. Dimarogonas, "Toward open-source and modular space systems with atmos," *IEEE Transactions on Field Robotics*, vol. 3, pp. 141–161, 2026.

[3] K. Saulnier, D. Pérez, R. Huang, D. Gallardo, G. Tilton, and R. Bevilacqua, "A six-degree-of-freedom hardware-in-the-loop simulator for small spacecraft," *Acta Astronautica*, vol. 105, no. 2, pp. 444–462, 2014.

[4] P. Tsiotras *et al.*, "Astros: A 5dof experimental facility for research in space proximity operations," *Advances in the Astronautical Sciences*, vol. 151, no. 1235, pp. 717–730, 2014.

[5] J. C. Jairala, R. Durkin, R. J. Marak, S. A. Sipila, Z. A. Ney, S. E. Parazynski, and A. H. Thomason, "Eva development and verification testing at nasa's neutral buoyancy laboratory," in *42nd International Conference on Environmental Systems (ICES)*, no. JSC-CN-26179, 2012.

[6] D. Akin, R. Roberts, J. Lane, and S. Weisman, "Robotic capabilities for complex space operations," in *AIAA Space 2001 Conference and Exposition*. American Institute of Aeronautics and Astronautics (AIAA), 2001.

[7] D. P. Miller, A. Wright, Y. Sargent, R. Cohen, and T. Hunt, "Attitude and Position Control Using Real-Time Color Tracking," *ResearchGate*, Mar. 2000.

[8] NASA, "Vertical Habitability Layout Studies and Neutral Buoyancy/Parabolic Flight Habitat Studies," NASA Technical Reports Server., Tech. Rep. Document ID: 20230008022, 2014.

[9] C. Carignan and D. Akin, "The reaction stabilization of on-orbit robots," *IEEE Control Systems Magazine*, vol. 20, no. 6, pp. 19–33, 2000.

[10] A. Donzé and O. Maler, "Robust satisfaction of temporal logic over real-valued signals," in *International conference on formal modeling and analysis of timed systems*. Springer, 2010, pp. 92–106.

[11] P. Tabuada, *Verification and control of hybrid systems: a symbolic approach*. Springer Science & Business Media, 2009.

[12] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "Fastrack: A modular framework for fast and guaranteed safe motion planning," in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2017, pp. 1517–1522.

[13] D. Heß, M. Althoff, and T. Sattel, "Formal verification of maneuver automata for parameterized motion primitives," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 1474–1481.

[14] J. Verhagen, L. Lindemann, and J. Tumova, "Robust stl control synthesis under maximal disturbance sets," in *2024 IEEE 63rd Conference on Decision and Control (CDC)*. IEEE, 2024, pp. 315–321.

[15] L. Lindemann and D. V. Dimarogonas, *Formal Methods for Multi-Agent Feedback Control Systems*. MIT Press, 2025.

[16] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, 2nd ed. John Wiley & Sons, 2021.

[17] M. Krstic, P. V. Kokotovic, and I. Kanellakopoulos, *Nonlinear and adaptive control design*. John Wiley & Sons, Inc., 1995.

[18] K. Fukuda, "Exact algorithms and software in optimization and polyhedral computation," in *Proceedings of the twenty-first international symposium on Symbolic and algebraic computation*, 2008, pp. 333–334.

[19] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[20] A. L. Schwartz, *Theory and implementation of numerical methods based on Runge-Kutta integration for solving optimal control problems*. University of California, Berkeley, 1996.

[21] D. Simon, J. Löfberg, and T. Glad, "Nonlinear model predictive control using feedback linearization and local inner convex constraint approximations," in *2013 European Control Conference (ECC)*. IEEE, 2013, pp. 2056–2061.

[22] L. Meier, D. Honegger, and M. Pollefeys, "PX4: A node-based multithreaded open source robotics framework for deeply embedded platforms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 6235–6240.

[23] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados – a modular open-source framework for fast embedded optimal control," *Mathematical Programming Computation*, 2021.

[24] I. Torroba, D. Dorner, V. N. Fernandez-Ayala, M. Kartasev, J. Verhagen, E. Krantz, G. Marchesini, C. Ljung, P. Roque, C. Sidrane, L. V. der Spaa, N. D. Carli, P. Ogren, C. Fuglesang, J. Tumova, D. V. Dimarogonas, and I. Stenius, "Marinarium: a new arena to bring maritime robotics closer to shore," 2026. [Online]. Available: https://arxiv.org/abs/2602.23053

[25] M. von Benzon, F. F. Sørensen, E. Uth, J. Jouffroy, J. Liniger, and S. Pedersen, "An open-source benchmark simulator: Control of a bluerov2 underwater robot," *Journal of marine science and engineering*, vol. 10, no. 12, p. 1898, 2022.

[26] P. W. Kenneally, S. Piggott, and H. Schaub, "Basilisk: A flexible, scalable and modular astrodynamics simulation framework," *Journal of aerospace information systems*, vol. 17, no. 9, pp. 496–507, 2020.

[27] E. Krantz, N. N. Chan, G. Tibert, H. Mao, and C. Fuglesang, "Bridging the Basilisk astrodynamics framework with ROS 2 for modular spacecraft simulation and hardware integration," in *Proceedings of the International Conference on Space Robotics (iSpaRo)*, Sendai, Japan, Dec. 2025, to appear. arXiv:2512.09833.

[28] L. Flückiger, K. Browne, B. Coltin, J. Fusco, T. Morse, and A. Symington, "Astrobee robot software: A modern software system for space," NASA Ames Research Center, Moffett Field, CA, USA, Tech. Rep. ARC-E-DAA-TN55483, Jun. 2018.