Journal of Zhejiang University-SCIENCE A 2026 27(10):1073-1094
www.jzus.zju.edu.cn; www.springer.com/journal/11582
E-mail: jzus_a@zju.edu.cn

Research Article
https://doi.org/10.1631/jzus.A2500507

Check for updates

Modeling and priority-based control of a bio-inspired hybrid actuation underwater robot with soft universal joint framework

Junhao ZHANG[1]✉, Jiancheng YU[1]✉, Yinglong CHEN[2], Yu TIAN[1], Qi ZHANG[1], Jie SUN[1], Xiaolong YU[1,3], Shilong LI[4], Yongjun GONG[2]

[1]*State Key Laboratory of Robotics and Intelligent Systems, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110169, China*
[2]*Naval Architecture and Ocean Engineering College, Dalian Maritime University, Dalian 116026, China*
[3]*State Key Lab of Precision Measurement Technology and Instruments, Tianjin University, Tianjin 300072, China*
[4]*State Key Laboratory of Underwater Acoustic Technology, Harbin Engineering University, Harbin 150001, China*

Abstract: Underwater robotic systems are increasingly expected to operate in confined, unstructured, and dynamic environments, in which conventional rigid vehicles and manipulators face limitations. To address these challenges, we develop a hybrid reconfigurable flexible underwater manipulator that integrates rigid self-propulsion modules with modular soft joint modules, enabling efficient thrust-based locomotion and highly maneuverable snake-like motions. The proposed architecture allows the robot to reconfigure its motion patterns in response to diverse operational requirements. Accurate kinematic modeling of such systems is critically dependent on the representation of the soft joint deformation. To this end, a novel soft universal joint (SUJ) model is proposed, which remains valid under non-constant-curvature (CC) bending and large-deformation conditions. The accuracy and generality of the proposed model are then evaluated through comparisons with the classical CC model. Building upon the SUJ-based kinematic framework, a priority-based control strategy is also developed to coordinate primary trajectory-tracking objectives with secondary safety-related constraints, including obstacle avoidance and joint-limit protection. Subsequently, numerical simulations and water-tank experiments, encompassing spiral motion, pitching upward, linear motion, and obstacle-avoidance tasks, are conducted to validate the proposed approach. The results demonstrate that our method achieves significantly higher accuracy than conventional CC models, while the control strategy also ensures robust task execution under multiple potentially conflicting objectives. Overall, this work presents a flexible and reconfigurable underwater robotic system with experimentally validated modeling and control methodologies, thus advancing the capabilities of underwater robots in complex and constrained environments.

Key words: Underwater robots; Underwater manipulator; Priority-based control; Swimming robot; Hybrid actuation

1 Introduction

Exploration, monitoring, and intervention within underwater environments are of increasing importance in oceanography, environmental protection, offshore engineering, and strategic defense (Liu et al., 2024; Zhao W et al., 2025). Traditional robotic platforms, such as remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs), have been widely deployed for tasks such as seabed mapping, infrastructure inspection, pipeline monitoring, and ecological observation (Sun et al., 2024). Despite their proven utility, the operational capabilities of these systems are often constrained in narrow, cluttered, or dynamic environments. In particular, rigid-body configurations and limited dexterity restrict their maneuverability in confined spaces, while their bulky structures hinder fine manipulation and adaptations in uncertain hydrodynamic conditions (Chen et al., 2022; Wu HY et al., 2024; Wang et al., 2025). These limitations highlight the need for novel underwater robotic systems that combine flexibility, reconfigurability, and multi-task execution capabilities (Vangi et al., 2025).

✉ Junhao ZHANG, zhangjunhao@sia.cn
Jiancheng YU, yjc@sia.cn
Junhao ZHANG, https://orcid.org/0000-0001-7665-7111
Jiancheng YU, https://orcid.org/0000-0003-1863-9737

Received Oct. 11, 2025; Revision accepted Jan. 15, 2026; Crosschecked Apr. 6, 2026; Online first May 18, 2026

© Zhejiang University Press 2026

In recent years, biology-inspired soft robotics have emerged as a promising solution to overcome these challenges (Qu et al., 2024). Drawing inspiration from aquatic organisms, such as eels, snakes, octopuses, and fish, researchers have developed flexible underwater robots that exhibit superior maneuverability, resilience, and environmental adaptability compared to traditional rigid designs (Kelasidi et al., 2016; Katzschmann et al., 2018; Dai et al., 2022; Zhang JH et al., 2022, 2024; Bao et al., 2023; Zhao et al., 2024; Zhao Q et al., 2025). For example, eel-like serpentine robots can perform undulatory locomotion, enabling them to traverse narrow passages and execute large-scale movements efficiently (Chen YH et al., 2021). Similarly, octopus-inspired robotic arms demonstrate strong compliance and dexterity, making them suitable for contact-rich manipulation tasks in unstructured environments (Wu MX et al., 2024). Moreover, Copebot, a copepod-like underwater soft robot, demonstrates potential for bio-mimetic locomotion in confined-space operations (He et al., 2023), while the YOLOX-DG robotic detection system provides specialized support for large-scale underwater inspection of concrete structures (Zhang CJ et al., 2024). Moreover, artificial intelligence-guided inverse design has facilitated the development of deployable thermo-metamaterial implants (Jiao et al., 2025), offering new avenues for optimizing structural adaptability in soft robotics. At the same time, combustion-based actuation technologies have expanded the boundaries of high-speed motion in soft robotic systems (Yang et al., 2024). Such advances have positioned biology-inspired soft underwater robots as a new frontier in marine robotics, with potential applications ranging from ecological monitoring to defense-oriented missions (Plum et al., 2020).

Nevertheless, developing such systems involves challenges in modeling and control aspects. The nonlinear material properties of soft actuators, coupled with complex fluid-structure interactions, make accurate dynamic modeling difficult (Du et al., 2021). Conventional approaches, such as the constant-curvature (CC) model, are mathematically tractable and have been widely adopted in continuum robot modeling (Webster III and Jones, 2010). However, CC models often fail to describe large or non-uniform bending scenarios, particularly when external forces, environmental constraints, or extreme postures are involved. To address these shortcomings, researchers have explored alternative modeling frameworks. For example, Cosserat rod theory provides a more rigorous continuum mechanics foundation capable of representing distributed strains and nonlinear deformations (Till et al., 2019; Pourghasemi Hanza and Ghafarirad, 2024; Xun et al., 2024), but it also leads to complex partial differential equations that are computationally expensive to solve and challenging for real-time control. Another approach is the piecewise constant-curvature (PCC) model (Della Santina et al., 2020), which approximates complex deformations by dividing a soft robot into multiple CC segments. Although this improves flexibility, it still exhibits CC's fundamental limitations when subjected to external loads or highly irregular shapes. More recently, machine learning-based methods such as neural networks have been applied to learn forward and inverse kinematics directly from data, enabling flexible adaptation to unmodeled dynamics (Nakajima et al., 2018; Chin et al., 2020; Shih et al., 2020). While these approaches show promise, they require extensive training datasets, which may be difficult to interpret, and often generalize poorly in dynamic underwater conditions. Thus, although significant progress has been made in soft robot modeling, existing approaches still struggle to achieve accuracy, efficiency, and robustness simultaneously in complex underwater environments. This underscores the need for new generalized modeling paradigms that can overcome these limitations while remaining tractable for control and real-time applications.

Another core challenge is the control of task execution under conflicting objectives (Wang LL et al., 2023). Underwater robots must often balance primary operational goals, such as end-effector trajectory tracking or base locomotion, with safety-critical constraints like collision avoidance, joint-limit avoidance, and energy efficiency (Mohan and Kim, 2012; Rizzini et al., 2017; Simetti and Casalino, 2017). Traditional control strategies, such as inverse kinematics or redundancy resolution methods (Iversflaten et al., 2022), often treat these tasks in parallel without clear prioritization; this can lead to instability or failure in situations where safety is a priority. Task-priority control frameworks address this by hierarchically organizing objectives into different levels of importance, ensuring that higher-priority safety constraints override lower-priority tracking tasks when conflicts occur (Simetti et al.,

2018). While this methodology has been extensively studied in terrestrial robotics, including humanoid and dual-arm manipulators (Bouyarmane and Kheddar, 2018; Hong et al., 2022), its application to biology-inspired underwater robots with hybrid soft–rigid structures has not been extensively explored.

In this work, we present a hybrid reconfigurable flexible underwater manipulator (HRFUM) that interleaves rigid self-propulsion modules with soft joint modules, enabling both snake-like locomotion and manipulator-like operations in confined and cluttered underwater spaces. A conceptual illustration of this system is shown in Fig. 1. At the modeling level, we introduce a soft universal joint (SUJ) representation for each soft module and embed it within a unified rigid–soft chain. In contrast to the classical CC model, the SUJ retains closed-form transforms while capturing non-CC deformations induced by external loads and contacts; as such, it offers a generalizable kinematic core for real-time computation. Building on this, we formulate a dynamic system of a floating-base underwater vehicle-manipulator system (UVMS), in which the thruster allocation is configuration-dependent; this is a crucial feature because the propulsors are mounted on mobile rigid modules rather than on a fixed base. Finally, we develop a priority-based trajectory tracking controller that unifies vector tasks (end-effector and base pose/trajectory) with scalar set-based tasks (obstacle avoidance and joint-limit protection) via null-space projection and an activation matrix for safe and conflict-free task reallocation. The overall framework is validated through numerical simulations and tank experiments, including spiral motion, pitching upward, linear motion, and obstacle-avoidance tasks; the results are also benchmarked against the CC baseline. The main contributions of this paper are as follows:

Hybrid reconfigurable architecture: we design an HRFUM methodology that combines rigid self-propulsion modules with modular soft joint modules, facilitating both snake-like locomotion and manipulator-like operations, and allowing adaptation to confined underwater spaces.

SUJ-based hybrid kinematics: we propose a SUJ model and integrate it with rigid modules as part of a generalized formulation. This formulation preserves compact and closed-form homogeneous transforms and adjoint/Jacobian operators, and we formally demonstrate that the CC workspace is a proper subset of the SUJ workspace.

Priority-based control unifying vector and scalar tasks: a hierarchical control strategy is established, which distinguishes between vector and scalar tasks. By leveraging null-space projection, the controller ensures safety-critical tasks are prioritized without compromising mission objectives.

2 Methods

2.1 System design

2.1.1 Hybrid rigid–soft modular architecture

The proposed architecture alternates rigid self-propulsion modules with soft joint modules, forming a hybrid serial chain (Figs. 2a and 2b) that can operate

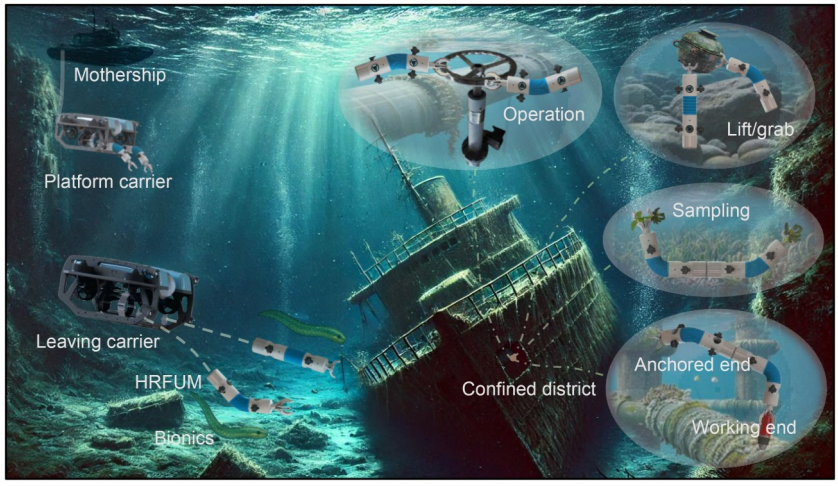


Fig. 1 Conceptual illustration of the HRFUM and its operational modes

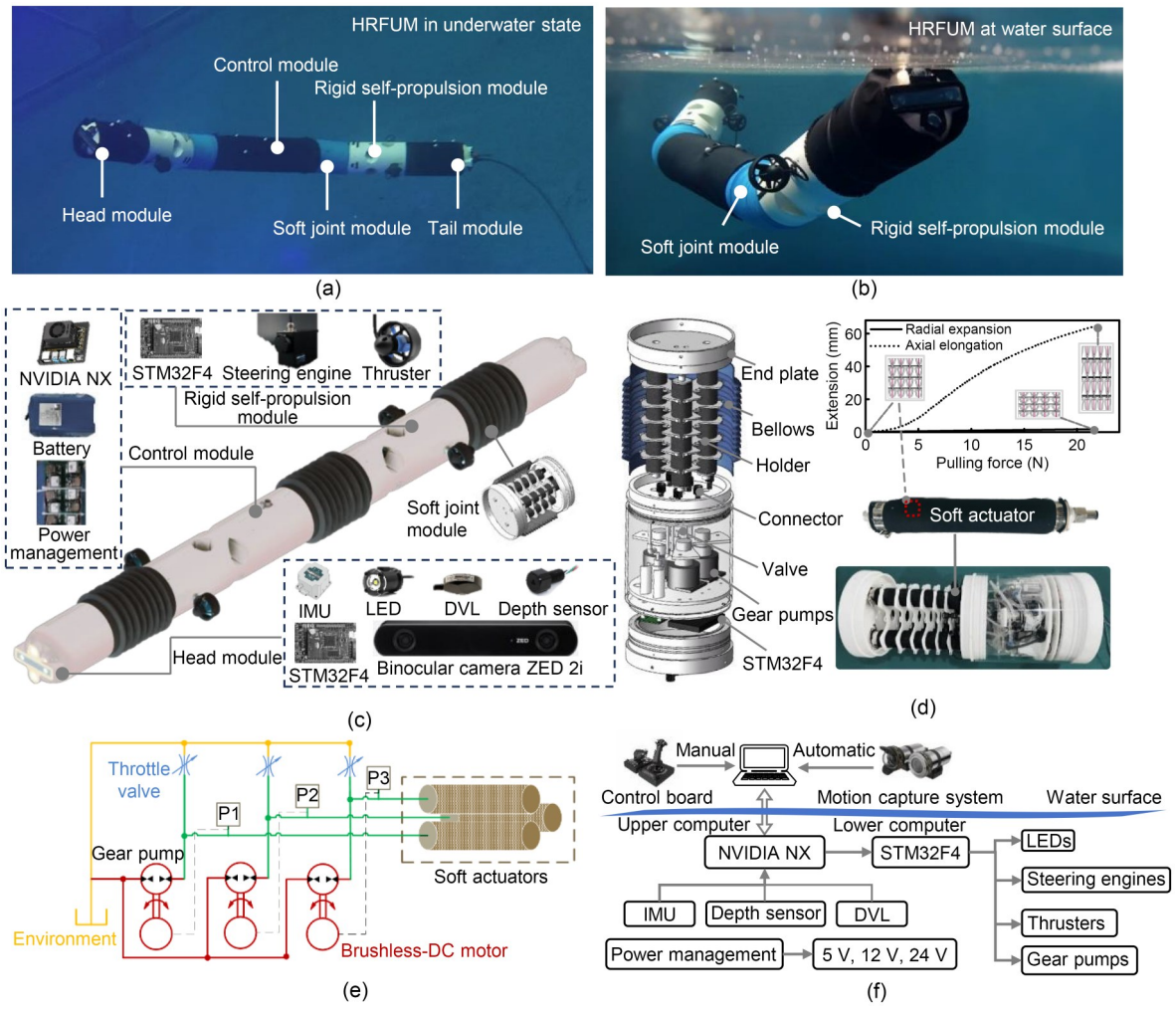


Fig. 2 Overall architecture, actuation, control, and configuration of the HRFUM: (a) underwater prototype of the HRFUM; (b) HRFUM floating on the water surface; (c) distribution of payloads and functional configurations of each module; (d) cross-sectional view of the soft joint module and anisotropic mechanical testing of the outer fabric; (e) miniaturized hydraulic actuation system for the soft joint module; (f) schematic of the electronic control system. NVIDIA NX is the NVIDIA® Jetson Xavier™ NX, STM32F4 is the STMicroelectronics STM32F407ZGT6, and both of them are controllers. IMU is the inertial measurement unit; LED is the light emitting diode; DVL is the doppler velocity log; DC represents the direct current. ZED 2i is the model of the binocular camera; P1, P2, and P3 represent the three pressure meters of the actuator

both underwater and at the surface (Zhang JH et al., 2022). The rigid self-propulsion modules, being driven by propellers, supply thrust and maintain precise attitude control for the body of the HRFUM; the soft joint modules, actuated by fabric-reinforced hydraulic muscles, undergo controllable deformations that enable large-amplitude bending and continuous spatial curvature variation. This combination yields a structure capable of both high-speed swimming and agile maneuvers. Under external loads, the soft modules passively deform to absorb shocks and mitigate structural damage risk, whereas the rigid modules ensure primary thrust output and directional accuracy. Compared to wholly rigid or wholly compliant designs, this hybrid approach synchronizes rigid thrust pulsations with soft bending; in this way, the undulatory propulsion of fish or snakes and other varied motion types can be emulated, achieving greater propulsive efficiency and smoother trajectories for the same power input. Furthermore, in confined or cluttered environments, the soft joint modules enable the manipulator to conform closely to curved surfaces, while the rigid self-propulsion modules maintain thrust output and heading control. This dual capability enhances the

robot's operational reach and positioning accuracy within restricted underwater spaces. The main structural and performance parameters of the designed HRFUM are summarized in Table 1.

2.1.2 Rigid self-propulsion module

In this hybrid architecture, the rigid self-propulsion modules are not simply appended to a host ROV or AUV as in traditional methods, but are autonomously distributed along the serial chain, enabling the HRFUM body to swim freely and perform complex maneuvers. As depicted in Fig. 2c, each rigid self-propulsion module integrates a three-axis orthogonal vectored thruster array, mounted at both ends of the adjacent soft joint. An optimized thrust allocation algorithm coordinates these thrusters to deliver precise six-degree-of-freedom (DOF) actuation, granting the HRFUM complete self-propulsion capability. To maintain neutral buoyancy and stable attitude across varying depths and payloads, the upper compartment incorporates a honeycomb-structured polyurethane composite for a positive buoyancy effect; meanwhile, the lower compartment houses stainless-steel ballast, which ensures self-balancing of the attitude and steady navigation even during high-speed acceleration or abrupt turns. A modular power distribution unit and distributed controllers (control module) are compactly embedded within each propulsion module and are linked via a microsecond-scale communication bus to synchronize thrust output and status feedback; this enables real-time reconfiguration in the event of a fault or single-thruster failure. In contrast to conventional manipulators carried by an ROV or AUV, the proposed HRFUM system features its own thrust and steering mechanisms, allowing the serial-chain robot to independently cruise, localize, and manipulate without reliance on an external host. This significantly enhances deployment flexibility and mission adaptability.

Our proposed system overcomes the limitations of traditional snake-like chain robots, namely, the extremes of entirely rigid-link stacks and purely silicone-based soft structures. It accomplishes this by introducing a hybrid rigid–soft design. Rigid self-propulsion modules alternate with fabric-reinforced hydraulically driven soft joint modules so as to balance thrust efficiency and maneuverability. As shown in Fig. 2d, based on our previous research foundation on soft robotic arms (Chen YL et al., 2019, 2021), each soft joint module is comprised of three fabric-reinforced hydraulic muscle actuators arranged in a spatially symmetric fashion. With an anisotropic fiber weave in the geometric constraint layer, radial expansion is effectively suppressed, and axial elongation is amplified. As depicted in Fig. 2e, the hydraulic actuation principle dictates that when a single chamber is pressurized while the other two remain at baseline, the resulting asymmetric stress field drives directional bending, converting hydraulic energy into an adjustable bending moment; this can result in up to 90° of planar curvature. Pressurizing two chambers establishes a pressure gradient that further increases the bending angle and output torque, while coordinated activation of all three chambers enables continuous 360° rotation in 3D space. Compared to traditional rigid or soft

Table 1 Main structural and performance parameters of the HRFUM prototype

Parameter	Value	Description
Total length (mm)	3000	Overall length of assembled HRFUM
Diameter (mm)	220	Diameter of the body shell
Total mass (kg)	80	Including all modules
Number of rigid modules	3	Modular configuration
Number of soft modules	2	Modular configuration
Length of rigid module (mm)	300	–
Length of soft module (mm)	300	–
Maximum bending of soft joint (°)	88	Measured at 0.3 MPa
Forward speed (m/s)	3.02	Experimental result
Reverse speed (m/s)	2.47	Experimental result
Ascent speed (m/s)	0.63	Experimental result
Descent speed (m/s)	0.21	Experimental result
Minimum turning radius (m)	0.89	Achieved at 0.3 MPa

The types of rigid modules include the head module, control module, and propulsion module

joints, these hydraulically actuated joints achieve full-spatial rotation at only 0.3 MPa of drive pressure, exhibit high load capacity, and respond rapidly, thus providing reliable and tunable stiffness support for underwater operations. The specific derivation of stiffness control is given in Section S1 of the electronic supplementary materials (ESM).

2.1.3 Electronics system

The electronic control system of the HRFUM involves a two-layer architecture that separates real-time underwater execution from high-level decision making, as shown in Figs. 2c and 2f, ensuring both robustness and computational efficiency. Section S2 of the ESM gives more details on the design.

2.2 Mathematical model

2.2.1 Reference frame definition

The HRFUM features an inherently modular architecture, enabling flexible expansion into a scalable and multi-module structure. To describe this reconfigurability, we formulate a generalized kinematic model that applies across arbitrary configurations. As illustrated in Fig. 3b, the system comprises n rigid self-propulsion modules and $n-1$ soft joint modules arranged in a serial chain. Each rigid module is modeled as a standard cylindrical body with its center of mass located at the geometric center and is indexed from tail to head as rigid i ($i\in[1, 2, \cdots, n]$), with the first module serving as the base of the system. These modules are outfitted with both embedded and externally mounted thrusters, as shown in Fig. 3a, whose positions and thrust vectors are defined in the fixed local body frame, rendering them invariant to deformation from adjacent joint bending. These thrusters generate propulsion and hovering forces, compensating for net buoyancy, hydrodynamic drag, and environmental disturbances. The soft joints, indexed as joint i ($i\in[1, 2, \cdots, n-1]$), connect adjacent rigid modules, enabling continuous curvature adaptation and compliant

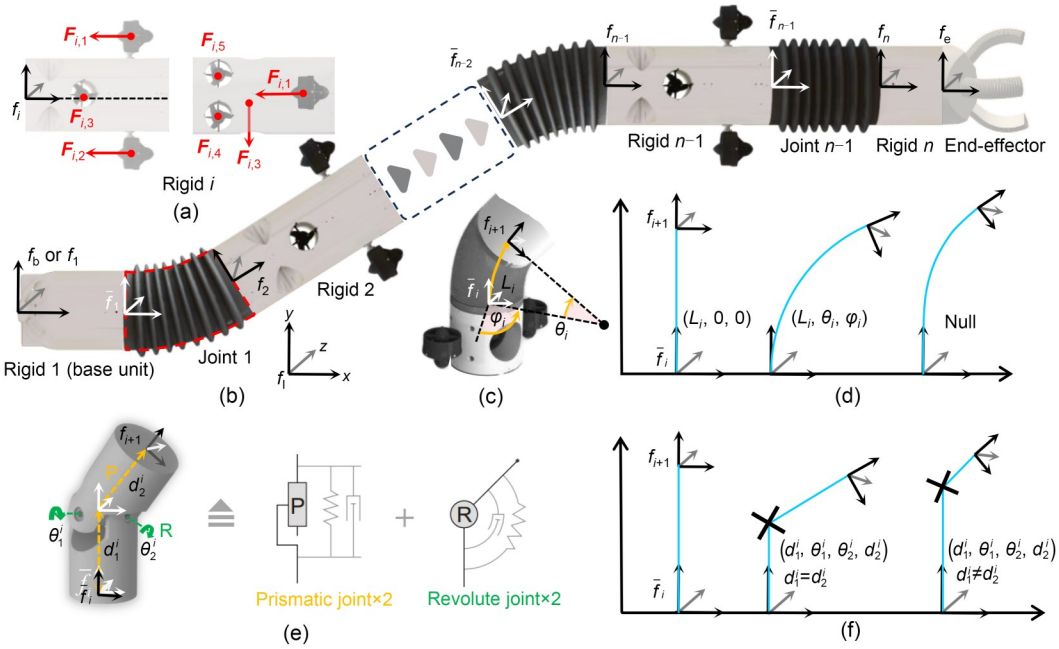


Fig. 3 Schematic illustration of the mathematical modeling of the HRFUM: (a) representation of the thrust forces generated by the rigid self-propulsion module; (b) generalized kinematic and dynamic models of HRFUM, including module indices and reference coordinate frames; (c) CC model for soft joints; (d) typical joint postures represented by the CC model, noting that the CC model becomes invalid when non-uniform curvatures occur; (e) SUJ model, which can be regarded as a combination of two soft prismatic joints (P) and two soft revolute joints (R); (f) typical postures represented by the SUJ model, which can fully describe various joint configurations. $F_{i,1}$–$F_{i,5}$ are the thrust of the five propellers of the rigid self-propulsion module in their local frame f_i, respectively. L_i, θ_i, and φ_i are the length, bending angle, and orientation angle of the joint i, respectively. d_1^i and θ_1^i are the prismatic joint variable and revolute joint variable of the first part of joint i discretized by the SUJ model, respectively; d_2^i and θ_2^i are the prismatic joint variable and revolute joint variable of the second part of joint i discretized by the SUJ model, respectively

articulation. Next, the kinematic description of the HRFUM motion relies on a hierarchy of coordinate frames and is defined as follows:

Inertial frame f_I: a global reference frame fixed to the Earth, used to describe absolute position and orientation.

Base frame f_b: a reference frame anchored at the tail end of the first rigid module, serving as the system's local reference origin.

Rigid module frames f_i, $i\in[1, 2, \cdots, n]$: each rigid self-propulsion module is associated with a body-fixed coordinate frame. The x-axis of each frame aligns with the longitudinal axis of the corresponding module.

Soft joint frames $\bar{f}_i$, $i\in[1, 2, \cdots, n-1]$: each soft joint module is assigned a coordinate frame that tracks its continuously deformable geometry.

End-effector frame f_e: a reference frame attached at the head of the last rigid module, enabling the representation of the tool pose for manipulation tasks.

All coordinate frames conform to the right-hand rule, with the z-axis oriented in-plane as per standard robotic conventions.

2.2.2 Kinematics of SUJ model for soft joint module

The forward kinematic model of the HRFUM is established via sequential coordinate transformations among the reference frames defined in Fig. 3c. Unlike traditional fixed-base manipulators, the HRFUM features a free-floating base. Its pose f_b with respect to the inertial frame f_I is described by the homogeneous transformation matrix $\boldsymbol{T}_\mathrm{I}^\mathrm{b}$:

$$\boldsymbol{T}_\mathrm{I}^\mathrm{b}=\begin{bmatrix}\boldsymbol{R}_\mathrm{I}^\mathrm{b} & \boldsymbol{p}_\mathrm{I}^\mathrm{b}\\ \boldsymbol{0}_{1\times3} & 1\end{bmatrix}, \tag{1}$$

where $\boldsymbol{R}_\mathrm{I}^\mathrm{b}$ and $\boldsymbol{p}_\mathrm{I}^\mathrm{b}$ denote the rotation matrix and position vector of the base frame f_b with respect to the inertial frame f_I, respectively. Since each rigid module maintains a straight and nondeformable configuration, the transformation matrix $\boldsymbol{T}_i^{\bar{i}}$ from rigid module frame f_i at one end of a rigid module to soft joint frame $\bar{f}_i$ at the opposite end is given by:

$$\boldsymbol{T}_i^{\bar{i}}=\begin{bmatrix}\boldsymbol{I} & [L_i\,0\,0]^\mathrm{T}\\ \boldsymbol{0}_{1\times3} & 1\end{bmatrix}, \tag{2}$$

where $\boldsymbol{I}$ denotes the identity matrix.

The kinematics of each soft joint module can initially be described using the CC model, with parameters $\boldsymbol{q}_i=[L_i, \theta_i, \varphi_i]$ ($i\in[1, 2, \cdots, n-1]$) denoting length, bending angle, and orientation angle, respectively (Webster III and Jones, 2010). While the CC model is valid in certain special cases, such as for perfectly straight or uniformly bent modules, it becomes inaccurate in most real-world scenarios. In practice, soft joints are subjected to persistent push–pull forces from the adjacent rigid modules they connect, resulting in non-uniform curvature being the dominant deformation mode. This effect is further amplified under high-end-load conditions, such as collisions or external impacts (Wang ZC et al., 2022; Zhang JH et al., 2024). As stated in Section 1, a common workaround for this is to discretize the joint into multiple CC segments or to adopt higher-order soft robot models, such as the Euler–Bernoulli beam theory or Cosserat rod theory. However, these approaches substantially increase the computational cost, compromising real-time performance. Additionally, they require additional coupling terms to account for the influence of the rigid self-propulsion modules, making the system model significantly more complex and less tractable for online control. To overcome these limitations, we propose a SUJ model, depicted in Fig. 3e, which represents each soft module as two soft prismatic joints and two soft revolute joints. Under this formulation, the configuration of the joint i is described by $\boldsymbol{q}_i=[d_1^i, \theta_1^i, d_2^i, \theta_2^i]$, and the overall joint space of the HRFUM is given by $\boldsymbol{q}=[\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_{n-1}]$. As shown in Figs. 3d and 3f, both the CC and SUJ models perform similarly under straight or uniform curvature conditions. However, under complex and non-uniform deformations caused by variable external forces, the SUJ model achieves higher geometric fidelity with substantially lower computational overhead, ensuring accuracy, robustness, and suitability for real-time underwater applications.

Since the soft joints in the HRFUM serve as connectors between adjacent rigid modules and are consistently subjected to external forces, we adopt the SUJ model to describe their motion. By integrating this soft joint description with the rigid module models, a unified rigid–soft kinematic framework is established. This unified approach ensures structural consistency in the dynamic equations, and also facilitates the application of classical modeling tools, enhancing computational efficiency. Specifically, the transformation

from frame $\bar{f}_i$ to f_{i+1} across a soft joint can be decomposed into the following four consecutive steps: 1) translation along the x-axis by d_1^i, corresponding to matrix $\boldsymbol{T}(\boldsymbol{x}, d_1^i)$; 2) rotation around the translated y-axis by θ_1^i, corresponding to matrix $\boldsymbol{R}(\boldsymbol{y}, \theta_1^i)$; 3) rotation around the rotated z-axis by θ_2^i, corresponding to matrix $\boldsymbol{R}(\boldsymbol{z}, \theta_2^i)$; 4) translation along the x-axis by d_2^i, corresponding to matrix $\boldsymbol{T}(\boldsymbol{x}, d_2^i)$. $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ are the unit vectors along the x-, y-, and z-axes, which are [1, 0, 0], [0, 1, 0], and [0, 0, 1], respectively. The resulting homogeneous transformation matrix $\boldsymbol{T}_{\bar{i}}^{i+1}$ is then given by:

$$\boldsymbol{T}_{\bar{i}}^{i+1}=\boldsymbol{T}(\boldsymbol{x}, d_1^i)\boldsymbol{R}(\boldsymbol{y}, \theta_1^i)\boldsymbol{R}(\boldsymbol{z}, \theta_2^i)\boldsymbol{T}(\boldsymbol{x}, d_2^i)=\begin{bmatrix} c\theta_1^i c\theta_2^i & -c\theta_1^i s\theta_2^i & s\theta_1^i & d_1^i+d_2^i c\theta_1^i c\theta_2^i \\ s\theta_2^i & c\theta_2^i & 0 & d_2^i s\theta_2^i \\ -c\theta_2^i s\theta_1^i & s\theta_1^i s\theta_2^i & c\theta_1^i & -d_2^i c\theta_2^i s\theta_1^i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3}$$

where $\boldsymbol{T}\in\mathbb{R}^{4\times4}$ and $\boldsymbol{R}\in\mathbb{R}^{4\times4}$ denote the elementary homogeneous transformation matrices for translation and rotation, respectively; "c" and "s" denote the "cos" and "sin" functions for notational simplicity, respectively. By sequentially composing the transformation matrices from all rigid and soft modules, we obtain the complete forward kinematics of the HRFUM, yielding the pose of the end-effector frame f_e with respect to f_I:

$$\boldsymbol{T}_{\mathrm{I}}^{\mathrm{e}}=\boldsymbol{T}_{\mathrm{I}}^{\mathrm{b}}\boldsymbol{T}_{\mathrm{b}}^{\mathrm{e}}=\boldsymbol{T}_{\mathrm{I}}^{\mathrm{b}}\boldsymbol{T}_{\mathrm{b}}^{\bar{1}}\boldsymbol{T}_{\bar{1}}^{2}\cdots\boldsymbol{T}_{\overline{n-1}}^{n}\boldsymbol{T}_{n}^{\mathrm{e}}, \tag{4}$$

where $\boldsymbol{T}_{\mathrm{I}}^{\mathrm{e}}$ denotes the transformation matrix from f_{I} to f_{e}; $\boldsymbol{T}_{\mathrm{b}}^{\mathrm{e}}$ denotes the transformation matrix from f_{b} to f_{e}; $\boldsymbol{T}_{n}^{\mathrm{e}}$ denotes the transformation matrix from f_n to f_{e}.

A comparative analysis between the CC and SUJ models, which can be found in Section S3 of the ESM, reveals the superiority of the SUJ model.

2.2.3 Dynamics of the hybrid reconfigurable flexible underwater manipulator

The HRFUM is essentially a floating-based UVMS, and therefore its dynamics are governed not only by the rigid-body inertia of the links, but also by hydrodynamic effects such as buoyancy, added mass, and drag. To construct a compact modular dynamical model suitable for real-time control, we follow the standard matrix formulation used in UVMS form (Podder and Sarkar, 2004) and extend it to account for the SUJ representation of each soft module. The overall dynamic equation in generalized coordinates system pose $\boldsymbol{\eta}=[\boldsymbol{\eta}_{\mathrm{v}}; \boldsymbol{q}]$ (vehicle/base pose $\boldsymbol{\eta}_{\mathrm{v}}$ plus overall joint coordinates $\boldsymbol{q}$) can be written in a block-matrix form, analogous to classic UVMS models:

$$\underbrace{\begin{bmatrix} \boldsymbol{M}_1 & \boldsymbol{M}_2 \\ \boldsymbol{M}_3 & \boldsymbol{M}_4 \end{bmatrix}}_{\boldsymbol{M}(\boldsymbol{\eta})}\begin{bmatrix} \ddot{\boldsymbol{\eta}}_{\mathrm{v}} \\ \ddot{\boldsymbol{q}} \end{bmatrix}+\underbrace{\begin{bmatrix} \boldsymbol{C}_1 & \boldsymbol{C}_2 \\ \boldsymbol{C}_3 & \boldsymbol{C}_4 \end{bmatrix}}_{\boldsymbol{C}(\boldsymbol{\eta}, \dot{\boldsymbol{\eta}})}\begin{bmatrix} \dot{\boldsymbol{\eta}}_{\mathrm{v}} \\ \dot{\boldsymbol{q}} \end{bmatrix}+\underbrace{\begin{bmatrix} \boldsymbol{D}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_2 \end{bmatrix}}_{\boldsymbol{D}(\dot{\boldsymbol{\eta}})}\begin{bmatrix} \dot{\boldsymbol{\eta}}_{\mathrm{v}} \\ \dot{\boldsymbol{q}} \end{bmatrix}+\underbrace{\begin{bmatrix} \boldsymbol{G}_1(\boldsymbol{\eta}_{\mathrm{v}}) \\ \boldsymbol{G}_2(\boldsymbol{q}) \end{bmatrix}}_{\boldsymbol{G}(\boldsymbol{\eta})}=\boldsymbol{\tau}+\boldsymbol{\tau}_{\mathrm{ext}}, \tag{5}$$

where $\boldsymbol{M}$ denotes the generalized inertia matrix including both inertia and hydrodynamic added mass; $\boldsymbol{M}_1$–$\boldsymbol{M}_4$ denote the four blocks of the inertia matrix $\boldsymbol{M}$, corresponding to $\ddot{\boldsymbol{\eta}}_{\mathrm{v}}$ and $\ddot{\boldsymbol{q}}$, respectively; $\boldsymbol{C}$ consists of Coriolis and centripetal terms, including inertial coupling between the base and joints; $\boldsymbol{C}_1$–$\boldsymbol{C}_4$ follows the same rule as $\boldsymbol{M}_1$–$\boldsymbol{M}_4$, respectively; $\boldsymbol{D}$ denotes the dissipative hydrodynamic damping and internal joint viscous terms $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$; $\boldsymbol{G}$ denotes the restoring forces from gravity and buoyancy $\boldsymbol{G}_1$ and $\boldsymbol{G}_2$; $\boldsymbol{\tau}$ denotes the actuation inputs, including thrusters $\boldsymbol{\tau}_{\mathrm{T}}$ and joint actuators $\boldsymbol{\tau}_{\mathrm{J}}$; $\boldsymbol{\tau}_{\mathrm{ext}}$ denotes the model external contact/disturbance forces mapped to generalized coordinates. This matrix structure follows the well-established UVMS formalism and facilitates control-by-design.

We extend the classic UVMS matrix dynamics to include SUJ parameters by treating each soft module as two deformable sub-links with lumped inertia and added mass. This results in configuration-dependent inertia and coupling matrices $\boldsymbol{M}_3$ and $\boldsymbol{M}_4$ that capture the principal rigid–soft interactions while ensuring the model is finite-dimensional and amenable to real-time control. Compared with continuum Cosserat or multi-module CC discretization, this approach preserves modeling fidelity for non-uniform curvature deformations while drastically reducing computational complexity. We later carry out experimental validation of the model's representation of added mass and damping via tow-tank and impulse tests.

Furthermore, unlike conventional ROVs or AUVs, where thrusters are rigidly fixed relative to the vehicle body, the positions and orientations of the propellers in the HRFUM vary with the joint configuration.

Consequently, the generalized input vector $\boldsymbol{\tau}(\boldsymbol{q})$ is dependent on the configuration. The generalized input wrench acting on the system is provided by both the joint actuators and propellers, so $\boldsymbol{\tau}(\boldsymbol{q})$ can therefore be decomposed into joint torques and propeller-generated wrenches:

$$\boldsymbol{\tau}(\boldsymbol{q})=\begin{bmatrix}\boldsymbol{B} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\begin{bmatrix}\boldsymbol{\tau}_{\mathrm{T}}(\boldsymbol{q})\\ \boldsymbol{\tau}_{\mathrm{J}}(\boldsymbol{q})\end{bmatrix}, \tag{6}$$

where $\boldsymbol{B}$ denotes the configuration-dependent thrust-allocation matrix that maps propeller inputs into generalized forces/wrenches on the floating base and joints; its dependence on $\boldsymbol{q}$ arises because each propeller's position and orientation change with the SUJ configuration. Each propulsion module is associated with a configuration matrix $\boldsymbol{B}_i\in\mathbb{R}^{6\times m}$ ($i\in[1, 2, \cdots, n]$) that describes how its m propellers produce generalized forces and moments on the module. The matrix $\boldsymbol{B}_i$ is determined by the thrust directions and application points of the individual propellers. $\boldsymbol{\beta}_{i,j}$ ($i\in[1, 2, \cdots, n]$; $j\in[1, 2, \cdots, m]$) denotes the unit vector of thrust direction, and $\boldsymbol{\chi}_{i,j}$ denotes the position vector of the jth propeller measured in the local frame of:

$$\boldsymbol{B}_i=\begin{bmatrix}\boldsymbol{\beta}_{i,1} & \boldsymbol{\beta}_{i,2} & \cdots & \boldsymbol{\beta}_{i,m}\\ \boldsymbol{\chi}_{i,1}\times\boldsymbol{\beta}_{i,1} & \boldsymbol{\chi}_{i,2}\times\boldsymbol{\beta}_{i,2} & \cdots & \boldsymbol{\chi}_{i,m}\times\boldsymbol{\beta}_{i,m}\end{bmatrix}. \tag{7}$$

The derivation of the remaining matrix blocks follows the standard UVMS formalism, and therefore is omitted for brevity. For completeness, the reader is referred to the UVMS literature (Antonelli, 2014), where these matrices are derived in full detail. The modeling process of this method is depicted in Fig. 4.

2.3 Priority-based control for trajectory tasks

2.3.1 Control task setting

In robotics, control objectives can be broadly categorized into two types based on the nature of their desired variables (Kanoun et al., 2011; Moe et al., 2016): vector (equality) tasks, where a multi-dimensional task variable $\boldsymbol{\sigma}$ is driven to match a desired target $\boldsymbol{\sigma}_{\mathrm{d}}$ within a finite period of time; scalar (inequality or set-based) tasks, where a scalar task variable σ must be kept within a safe range $[\sigma_{\min}, \sigma_{\max}]$. This classification is well-recognized in task-priority control frameworks, as vector tasks often represent operational objectives, while scalar tasks typically represent safety or feasibility constraints. In contrast to traditional industrial manipulators, where the base is fixed, and only the end-effector executes motion, the proposed HRFUM features both a free-floating base and a movable end-effector. Furthermore, its slender and snake-like morphology means that the entire robot cannot be approximated as a point mass, particularly within narrow and cluttered underwater environments. Consequently, both ends of the robot should be considered as independent controllable frames.

In this work, the vector task, including controlling the position $\boldsymbol{p}_{\mathrm{e}}$ and orientation $\boldsymbol{\Theta}_{\mathrm{e}}$ of the end-effector, is denoted as $\boldsymbol{\sigma}_1$:

$$\boldsymbol{\sigma}_1=\begin{bmatrix}\boldsymbol{p}_{\mathrm{e}}\\ \boldsymbol{\Theta}_{\mathrm{e}}\end{bmatrix}\rightarrow\boldsymbol{\sigma}_{1,\mathrm{d}}=\begin{bmatrix}\boldsymbol{p}_{\mathrm{e,d}}\\ \boldsymbol{\Theta}_{\mathrm{e,d}}\end{bmatrix}, \tag{8}$$

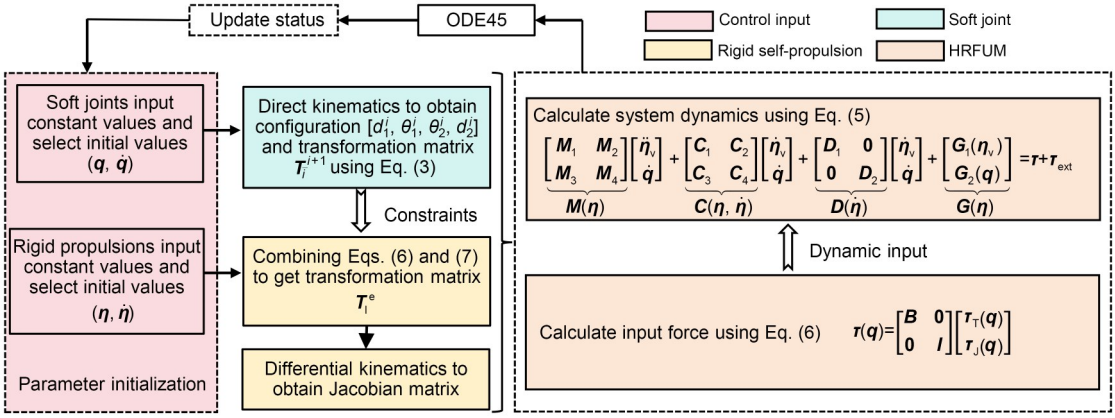


Fig. 4 Flowchart for the modeling process of the SUJ model of HRFUM. ODE45 is an adaptive step-size Runge–Kutta numerical solver for solving ordinary differential equations in MATLAB

where $\boldsymbol{\sigma}_{1,\mathrm{d}}$ denotes the desired target of $\boldsymbol{\sigma}_1$; $\boldsymbol{p}_{\mathrm{e,d}}$ and $\boldsymbol{\Theta}_{\mathrm{e,d}}$ denote the desired position and orientation of the end-effector, respectively. Similarly, the base position control task can be defined as a vector task $\boldsymbol{\sigma}_2$, which is especially relevant for operational modes where the base is required to either track a reference trajectory or maintain a fixed location:

$$\boldsymbol{\sigma}_2=\boldsymbol{p}_{\mathrm{b}}\rightarrow\boldsymbol{\sigma}_{2,\mathrm{d}}=\boldsymbol{p}_{\mathrm{b,d}}, \tag{9}$$

where $\boldsymbol{p}_{\mathrm{b}}$ denotes the position of the base; $\boldsymbol{\sigma}_{2,\mathrm{d}}$ denotes the desired target of $\boldsymbol{\sigma}_2$; $\boldsymbol{p}_{\mathrm{b,d}}$ denotes the desired position of the base.

In addition to these operational vector tasks, HRFUM's redundancy allows for scalar safety tasks to be incorporated without compromising primary objectives, unless activated as such. These scalar tasks are set-based, meaning they are inactive under nominal operation but become the highest-priority tasks when safety thresholds are violated. Examples include:

Collision avoidance σ_{a}: maintaining a minimum distance between the end-effector and the center of a spherical obstacle:

$$\sigma_{\mathrm{a}}=\sqrt{(\boldsymbol{p}_{\mathrm{e}}-\boldsymbol{p}_{\mathrm{o}})^{\mathrm{T}}(\boldsymbol{p}_{\mathrm{e}}-\boldsymbol{p}_{\mathrm{o}})}, \tag{10}$$

where $\boldsymbol{p}_{\mathrm{o}}$ denotes the position of the obstacle.

Joint limit avoidance $\boldsymbol{\sigma}_{\mathrm{b}}$: keeping all joint angles within specified limits, that is, $\boldsymbol{\sigma}_{\mathrm{b}}=\boldsymbol{q}$.

When σ_{a} (or any component of $\boldsymbol{\sigma}_{\mathrm{b}}$) approaches its threshold, the corresponding task becomes active and overrides lower-priority objectives in order to ensure safe operation. The specific task settings and derivations can be found in Section S4 of the ESM.

2.3.2 Priority-based control formulation

First, the joint space dynamic model needs to be transformed into a task space representation, and so Eq. (5) is simplified to:

$$\boldsymbol{M}\ddot{\boldsymbol{\eta}}+\boldsymbol{W}=\boldsymbol{\tau}, \tag{11}$$

where $\boldsymbol{W}=\boldsymbol{C}\dot{\boldsymbol{\eta}}+\boldsymbol{D}\dot{\boldsymbol{\eta}}+\boldsymbol{G}$. For any task Δ (Δ is $\boldsymbol{\sigma}_1$, $\boldsymbol{\sigma}_2$, σ_{a}, or $\boldsymbol{\sigma}_{\mathrm{b}}$), its generalized torque $\boldsymbol{\tau}_\Delta$ and generalized force $\boldsymbol{F}_\Delta$ can be mapped through the corresponding task Jacobian matrix $\boldsymbol{\tau}_\Delta=\boldsymbol{J}_\Delta^{\mathrm{T}}\boldsymbol{F}_\Delta$. Then, we substitute this equation into Eq. (11), and the expression for acceleration can be obtained as $\ddot{\boldsymbol{\eta}}=\boldsymbol{M}^{-1}(\boldsymbol{J}_\Delta^{\mathrm{T}}\boldsymbol{F}_\Delta-\boldsymbol{W})$. Continuing to substitute into the task differential kinematics equation $\ddot{\boldsymbol{\sigma}}=\dot{\boldsymbol{J}}_\Delta\boldsymbol{\zeta}+\boldsymbol{J}_\Delta\dot{\boldsymbol{\zeta}}$, where $\boldsymbol{\zeta}$ is the system velocity, we can obtain:

$$\ddot{\boldsymbol{\sigma}}=\boldsymbol{J}_\Delta\boldsymbol{M}^{-1}\boldsymbol{J}_\Delta^{\mathrm{T}}\boldsymbol{F}_\Delta-\boldsymbol{J}_\Delta\boldsymbol{M}^{-1}\boldsymbol{W}+\dot{\boldsymbol{J}}_\Delta\boldsymbol{\zeta}. \tag{12}$$

Next, define the inertia matrix $\boldsymbol{M}_\Delta$ associated with the task Δ as: $\boldsymbol{M}_\Delta=(\boldsymbol{J}_\Delta\boldsymbol{M}^{-1}\boldsymbol{J}_\Delta^{\mathrm{T}})^{-1}$. Then, multiply both sides of Eq. (12) by the matrix $\boldsymbol{M}_\Delta$ to obtain the dynamic expression in the task space:

$$\boldsymbol{M}_\Delta\ddot{\boldsymbol{\sigma}}+\boldsymbol{W}_\Delta=\boldsymbol{F}_\Delta, \tag{13}$$

where $\boldsymbol{W}_\Delta=\boldsymbol{M}_\Delta\boldsymbol{J}_\Delta\boldsymbol{M}^{-1}\boldsymbol{W}+\boldsymbol{M}_\Delta\dot{\boldsymbol{J}}_\Delta\dot{\boldsymbol{\eta}}$.

In complex underwater missions, the HRFUM must simultaneously satisfy multiple control objectives, such as end-effector trajectory tracking, base position control, collision avoidance, and joint limit prevention. These tasks often have different priorities; for instance, maintaining end-effector trajectory tracking accuracy typically takes precedence over secondary objectives like avoiding singularities or minimizing energy consumption. To handle such multi-objective scenarios, a priority-based control scheme is adopted and formulated using null-space projection.

Considering the situation when the scalar task of the robot is not activated, the generalized torque $\boldsymbol{\tau}_\Delta$ can be split into the torque $\boldsymbol{\tau}_{\sigma_1}$ corresponding to the primary task $\boldsymbol{\sigma}_1$ and the torque $\boldsymbol{\tau}_{\sigma_2}$ required for the secondary task $\boldsymbol{\sigma}_2$. The torque required for the secondary task does not affect the implementation of the primary task. Based on the null-space operator method, the expression for the total torque $\boldsymbol{\tau}$ is:

$$\boldsymbol{\tau}=\boldsymbol{\tau}_{\sigma_1}+\boldsymbol{P}\boldsymbol{\tau}_{\sigma_2}=\boldsymbol{J}_{\sigma_1}^{\mathrm{T}}\boldsymbol{F}_{\sigma_1}+\boldsymbol{P}\boldsymbol{J}_{\sigma_2}^{\mathrm{T}}\boldsymbol{F}_{\sigma_2}, \tag{14}$$

where $\boldsymbol{J}_{\sigma_1}$ and $\boldsymbol{J}_{\sigma_2}$ denote the Jacobian matrices of the primary task $\boldsymbol{\sigma}_1$ and secondary task $\boldsymbol{\sigma}_2$, respectively; $\boldsymbol{P}$ is the null space operator related to the Jacobian matrix of the main task and can be expressed as $\boldsymbol{P}=\boldsymbol{I}-\boldsymbol{J}_{\sigma_1}^{\mathrm{T}}\overline{\boldsymbol{J}_{\sigma_1}^{\mathrm{T}}}$. $\overline{\boldsymbol{J}_{\sigma_1}}$ is the weighted pseudoinverse matrix of $\boldsymbol{J}_{\sigma_1}$ and can be expressed as:

$$\overline{\boldsymbol{J}_{\sigma_1}^{\mathrm{T}}}=\boldsymbol{M}^{-1}\boldsymbol{J}_{\sigma_1}^{\mathrm{T}}\boldsymbol{M}_{\sigma_1}=\boldsymbol{M}^{-1}\boldsymbol{J}_{\sigma_1}^{\mathrm{T}}(\boldsymbol{J}_{\sigma_1}\boldsymbol{M}^{-1}\boldsymbol{J}_{\sigma_1}^{\mathrm{T}})^{-1}, \tag{15}$$

where $\boldsymbol{M}_{\sigma_1}$ denotes the inertia matrix associated with the primary task $\boldsymbol{\sigma}_1$.

This ensures that the secondary task operates strictly in the null space of the primary task, avoiding interference. Next, we can set the point-to-point trajectory at the robot end as $\boldsymbol{r}_1$, and the trajectory of the robot base as $\boldsymbol{r}_2$. The desired primary task $\boldsymbol{\sigma}_{1,\mathrm{d}}$ and desired secondary task $\boldsymbol{\sigma}_{2,\mathrm{d}}$ of the smooth trajectory are generated through a second-order damping equation. Introducing the task error $\tilde{\boldsymbol{\sigma}}_i=\boldsymbol{\sigma}_{i,\mathrm{d}}-\boldsymbol{\sigma}_i$, ($i$=1, 2), and the generalized force $\boldsymbol{F}_{\sigma_i}$ associated with the task $\boldsymbol{\sigma}_i$ results in:

$$\boldsymbol{F}_{\sigma_i}=\boldsymbol{M}_{\sigma_i}\left(\ddot{\boldsymbol{\sigma}}_{i,\mathrm{d}}-K_{\mathrm{p},i}\tilde{\boldsymbol{\sigma}}_i-K_{\mathrm{i},i}\int\tilde{\boldsymbol{\sigma}}_i\,\mathrm{d}t-K_{\mathrm{d},i}\dot{\tilde{\boldsymbol{\sigma}}}_i-\dot{\boldsymbol{J}}_{\sigma_i}\dot{\boldsymbol{\eta}}\right)+\boldsymbol{W}_{\sigma_i},\tag{16}$$

where $\boldsymbol{M}_{\sigma_i}$ denotes the inertia matrix associated with the vector task $\boldsymbol{\sigma}_i$; $\boldsymbol{J}_{\sigma_i}$ denotes the Jacobian matrices of the vector task $\boldsymbol{\sigma}_i$; $K_{\mathrm{p},i}$, $K_{\mathrm{i},i}$, and $K_{\mathrm{d},i}$ denote the control gains of the vector tasks; t is the time.

When the scalar task is activated, collision avoidance σ_a and joint-limit avoidance $\boldsymbol{\sigma}_\mathrm{b}$ are incorporated with higher priority than vector tasks. Task priority is assigned in the hierarchy of $\sigma_\mathrm{a}>\boldsymbol{\sigma}_\mathrm{b}>\boldsymbol{\sigma}_1>\boldsymbol{\sigma}_2$. Next, the activation of the scalar task can be determined according to the following conditions. For a scalar task σ and its effective interval range $[\sigma_{\min},\ \sigma_{\max}]$, if $\sigma_{\min}<\sigma<\sigma_{\max}$, $\sigma\leqslant\sigma_{\min}$ and $\dot{\sigma}>0$, as well as $\sigma\geqslant\sigma_{\max}$ and $\dot{\sigma}<0$, σ is not activated. Except for the above restrictions, all other conditions are in the activated state. Subsequently, when one or more scalar tasks are in the activated state, an activation matrix $\boldsymbol{A}$ needs to be constructed. Initially, the activation matrix $\boldsymbol{A}\in\mathbb{R}^{M\times M}$ is taken as the identity matrix, where M is the total number of scalar tasks. Then, each scalar task is checked one by one: if the task is within the safe range, or if it reaches the lower limit but is still moving outwards, or if it reaches the upper limit but is still moving outwards, then the corresponding row is removed from $\boldsymbol{A}$. Finally, the returned $\boldsymbol{A}\in\mathbb{R}^{P\times M}$ ($P\leqslant M$, and P is the number of scalar tasks in an active state) retains only the rows of scalar tasks that are in the active state, thus dynamically updating the scalar task set in priority control. To simplify the representation, the scalar tasks are constructed as a set $\boldsymbol{\sigma}^*$. Let $\boldsymbol{\sigma}^*=[\sigma_\mathrm{a}\ \boldsymbol{\sigma}_\mathrm{b}^\mathrm{T}]^\mathrm{T}$, along with $\dot{\boldsymbol{\sigma}}^*=[\dot{\sigma}_\mathrm{a}\ \dot{\boldsymbol{\sigma}}_\mathrm{b}^\mathrm{T}]^\mathrm{T}$, $\boldsymbol{\sigma}^*_{\min}=[\sigma_{\mathrm{a},\min}\ \boldsymbol{\sigma}^\mathrm{T}_{\mathrm{b},\min}]^\mathrm{T}$, $\boldsymbol{\sigma}^*_{\max}=[\sigma_{\mathrm{a},\max}\ \boldsymbol{\sigma}^\mathrm{T}_{\mathrm{b},\max}]^\mathrm{T}$, and $\boldsymbol{J}_{\sigma^*}=[\boldsymbol{J}_{\sigma_\mathrm{a}}\boldsymbol{J}_{\sigma_\mathrm{b}}]^\mathrm{T}$. $[\boldsymbol{\sigma}^*_{\min},\ \boldsymbol{\sigma}^*_{\max}]$ is the safe range of $\boldsymbol{\sigma}^*$; $[\sigma_{\mathrm{a},\min},\ \sigma_{\mathrm{a},\max}]$ and $[\boldsymbol{\sigma}_{\mathrm{b},\min},\ \boldsymbol{\sigma}_{\mathrm{b},\max}]$ are the safe ranges of σ_a and $\boldsymbol{\sigma}_\mathrm{b}$, respectively. The trajectory tracking controller based on task priority is shown in Fig. 5. When there is only input of vector tasks or the scalar task is not activated, the controller takes the form of Eq. (16). Furthermore, based on the set effective intervals $D(\sigma_\mathrm{a})$ and $D(\boldsymbol{\sigma}_\mathrm{b})$ of the scalar task, if a scalar task is activated, the activation matrix $\boldsymbol{A}$ is calculated, and then, the generalized torque $\boldsymbol{\tau}_{\sigma^*}$ of the scalar task set $\boldsymbol{\sigma}^*$ is determined. Finally, according to the calculated $\boldsymbol{\tau}_{\sigma_1}$, $\boldsymbol{\tau}_{\sigma_2}$, and $\boldsymbol{\tau}_{\sigma^*}$, control will be performed in the order of priority. The generalized driving torque $\boldsymbol{\tau}$ can be set as:

$$\begin{aligned}\boldsymbol{\tau}=\boldsymbol{\tau}_{\sigma^*}+\boldsymbol{P}_1\boldsymbol{\tau}_{\sigma_1}+\boldsymbol{P}_2\boldsymbol{\tau}_{\sigma_2}=\underbrace{(\boldsymbol{A}\boldsymbol{J}_{\sigma^*})^\mathrm{T}}_{\boldsymbol{J}_A}+\boldsymbol{F}_{\sigma^*}+\\ \boldsymbol{P}_1\boldsymbol{J}_{\sigma_1}^\mathrm{T}\boldsymbol{F}_{\sigma_1}+\boldsymbol{P}_2\boldsymbol{J}_{\sigma_2}^\mathrm{T}\boldsymbol{F}_{\sigma_2},\end{aligned}\tag{17}$$

where $\boldsymbol{P}_1$ and $\boldsymbol{P}_2$ denote the null-space operators, which can be expressed as $\boldsymbol{P}_1=\boldsymbol{I}-\boldsymbol{J}_A^\mathrm{T}\overline{\boldsymbol{J}_A^\mathrm{T}}$ and $\boldsymbol{P}_2=(\boldsymbol{I}-\boldsymbol{J}_{\sigma_1}^\mathrm{T}\overline{\boldsymbol{J}_{\sigma_1}^\mathrm{T}})\boldsymbol{P}_1$, respectively; $\boldsymbol{J}_{\sigma^*}$ denotes the Jacobian matrix of the scalar task set $\boldsymbol{\sigma}^*$; $\overline{\boldsymbol{J}_A}$ and $\overline{\boldsymbol{J}_{\sigma_1}}$ are the weighted pseudo-inverse matrices of $\boldsymbol{J}_A=\boldsymbol{A}\boldsymbol{J}_{\sigma^*}$ and $\boldsymbol{J}_{\sigma_1}$, respectively. $\boldsymbol{F}_{\sigma^*}$ is the generalized force required for the scalar task set $\boldsymbol{\sigma}^*$ and can be designed by referring to $\boldsymbol{F}_{\sigma_1}$ and $\boldsymbol{F}_{\sigma_2}$ as:

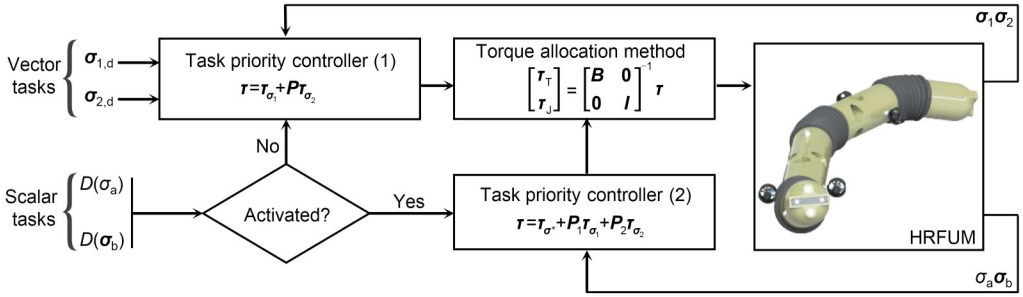


Fig. 5 Priority-based control framework for trajectory tracking tasks

$$\boldsymbol{F}_{\sigma^*}=\underbrace{\left(\boldsymbol{A}\boldsymbol{J}_{\sigma^*}\boldsymbol{M}^{-1}\boldsymbol{J}_{\sigma^*}^{\mathrm{T}}\boldsymbol{A}^{\mathrm{T}}\right)^{-1}}_{\boldsymbol{M}_{\sigma^*}}\left(\ddot{\boldsymbol{\sigma}}^*-\boldsymbol{A}\boldsymbol{K}_{\mathrm{p},*}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\tilde{\boldsymbol{\sigma}}^*-\boldsymbol{A}\boldsymbol{K}_{\mathrm{i},*}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\int\tilde{\boldsymbol{\sigma}}_i\,\mathrm{d}t-\boldsymbol{A}\boldsymbol{K}_{\mathrm{d},*}\boldsymbol{A}^{\mathrm{T}}\boldsymbol{A}\dot{\tilde{\boldsymbol{\sigma}}}^*-\boldsymbol{A}\dot{\boldsymbol{J}}_{\sigma^*}\boldsymbol{\zeta}\right),\tag{18}$$

where $K_{\mathrm{p},*}$, $K_{\mathrm{i},*}$, and $K_{\mathrm{d},*}$ denote the control gains of the scalar task set; $\tilde{\boldsymbol{\sigma}}^*$ denotes the task error of the scalar task set $\boldsymbol{\sigma}^*$.

The proposed framework ensures closed-loop stability through two mechanisms: (1) hierarchical task prioritization via null-space projection, where high-priority scalar tasks and vector tasks are decoupled, preventing conflicting control inputs that could induce instability; (2) incorporation of proportional–integral–derivative (PID) damping terms in both the vector and scalar task controllers, which suppresses oscillatory responses and enhances system damping. For scenarios with activated scalar tasks, the dynamic activation matrix $\boldsymbol{A}$ ensures smooth switching between task sets without discontinuous control signals, further promoting stability.

3 Results

3.1 Test platform and system performance

The experimental validation was carried out using a deep-water integrated testing platform at the Key Laboratory of Rescue and Salvage Engineering, Dalian Maritime University, Dalian, China. The main tank measures 50 m×30 m×10 m and is equipped with a Qualisys underwater 3D motion-capture system. As shown in Fig. 6a, the capturing system has a symmetrical dual-bracket arrangement, with three high-sensitivity underwater optical cameras mounted on each side. Each camera is supported by a spherical hinge allowing ±60° continuous adjustment in all directions, and all three are installed on motorized rails to aid dynamic

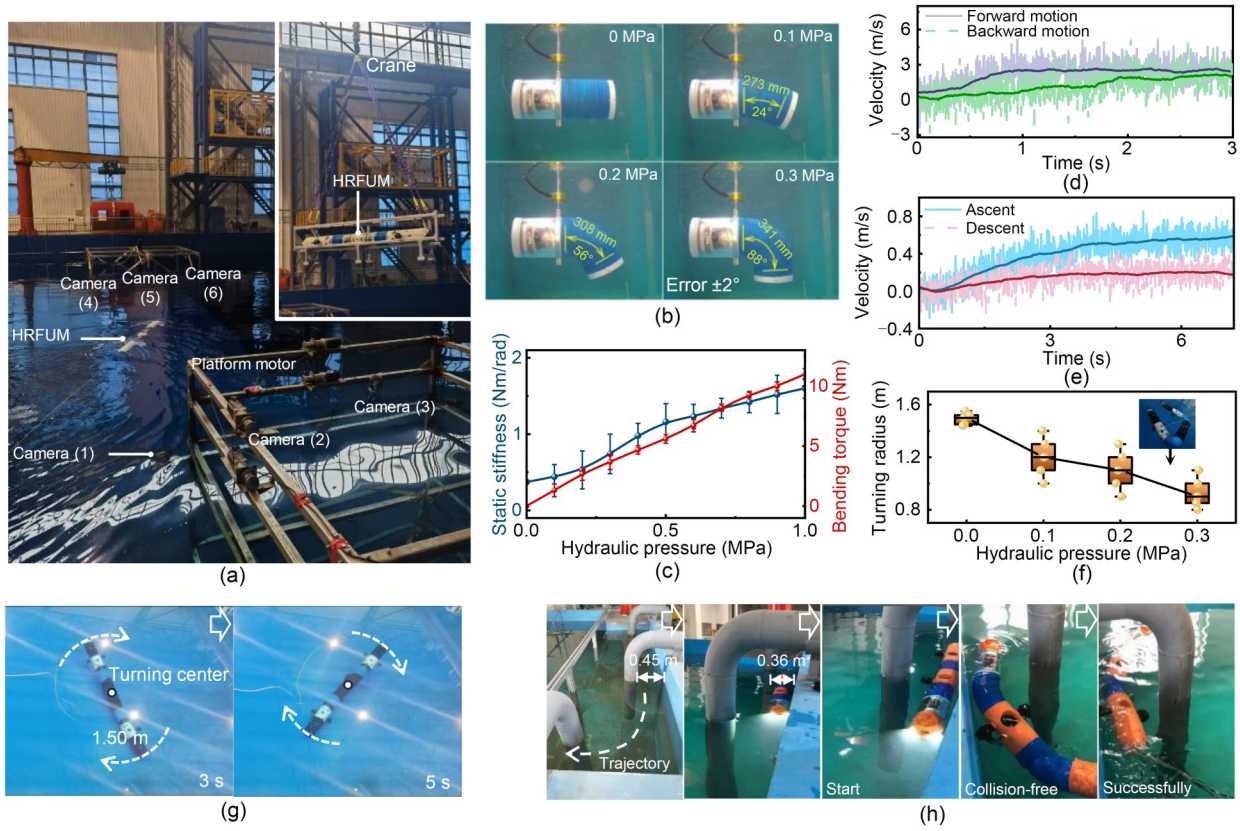


Fig. 6 Experimental platform, testing facility, and performance evaluation of the HRFUM: (a) experimental setup showing the Qualisys motion capture system and deployment device; (b) workspace characterization of soft joint bending motions; (c) measurement of bending torque and static stiffness of the soft joint; (d) maximum speed test during forward and backward straight-line motions; (e) maximum speed test during vertical ascent and descent motions; (f) turning radius test under different hydraulic pressures to the soft joint; (g) turning performance of the HRFUM in AUV mode; (h) narrow-passage traversal test, demonstrating the robot's maneuverability in confined environments

spatial positioning. This configuration provides a capture volume of approximately 12 m×10 m×5 m. The system operates at 500 Hz with a spatial resolution of 0.5 mm, enabling real-time reconstruction of 3D underwater trajectories. To reduce optical interference, the reflective internal components and metallic skeleton of the HRFUM were enclosed in a light-absorbing and drag-reducing fabric, with reflective markers affixed at equidistant key points on the head, trunk, and tail using waterproof adhesives and reinforcement structures. Moreover, a crane and mobile storage frame facilitated rapid deployment and recovery of the test prototype.

The fully assembled HRFUM has a total length of 3000 mm, a maximum diameter of 220 mm, and a mass of approximately 80 kg. The robot integrates multi-DOF soft joints with rigid self-propulsion modules, overcoming the limitations of traditional rigid single-DOF joints. Our motion performance evaluations highlight the versatility of the soft joint system. As shown in Fig. 6b, uniform pressurization of all three actuator chambers to 0.3 MPa yields an axial elongation of 50.2 mm. Under a single-chamber pressure gradient of 0.1–0.3 MPa, the joint achieves a maximum planar bending angle of 88°±2°. Additionally, asymmetric pressurization enables continuous adjustment of the bending direction. The static stiffness modulation capacity is showcased in Fig. 6c: under the condition of a hydraulic pressure of 1.0 MPa, the bending torque can reach 10 Nm, and the static stiffness can be tuned up to 3:1 across the pressure range. For high-precision manipulation or grasping tasks, pressurizing two chambers to 0.5 MPa approximates a rigid state and maximizes the resistance to external disturbances. Conversely, reducing the pressure to 0.1 MPa or lower restores compliance, allowing navigation through confined spaces or improved impact absorption. Integrated micro proportional valves and distributed control enable rapid static stiffness switching.

The propulsion performance is summarized in Figs. 6d and 6e. The axial propulsion system achieves forward and backward speeds of 3.02 and 2.47 m/s, respectively, at the maximum duty cycle, while the vertical propulsion system enables ascent and descent speeds of 0.63 and 0.21 m/s, respectively. Dynamic flexibility tests demonstrate exceptional agility: at a drive pressure of 0.3 MPa, the HRFUM achieves a minimum turning radius of 0.89 m (approximately one-third of its body length), as depicted in Fig. 6f. When the soft joints are unpressurized, the robot remains straight and turns overall like a conventional AUV, with a turning radius of 1.50 m (about half of its body length, Fig. 6g). Its narrow-space maneuvering capability was verified further (Fig. 6h): the widest cross-section of the robot, including its propellers, is 0.36 m, yet it successfully navigated through a gap of only 0.45 m between the water pipes and the tank wall without collisions.

Overall, the results demonstrate that the biology-inspired actuation framework of the HRFUM enables multi-DOF coordinated motions, rapid torque/stiffness modulation, tight-radius maneuvering, and adaptive posture adjustments. These capabilities provide a new technological pathway for reliable robotic operations in complex and constrained underwater environments.

3.2 Dynamic simulations and experiments

Differing from conventional rigid chain-type underwater robots, the proposed HRFUM adopts a hybrid architecture that interleaves rigid self-propulsion modules with soft joint modules. This design enables a rich motion repertoire, including movements such as straight horizontal translation, surface turning, pitching/surging (or diving and climbing), and spiral ascent/descent, without relying on body undulation as the primary means of propulsion. The key enabler of this behavior is the soft joint multi-DOF deformation, where the robot achieves axial extension and full spatial bending by independently pressurizing the three actuators in each soft joint; as such, it can reconfigure its shape to adapt to confined underwater environments, unlike rigid single-DOF hinge designs. Moreover, the addition of actively driven propellers further decouples translation from serpentine gait, improving locomotion efficiency and maneuverability.

This section focuses on validating the SUJ-based model against the classical CC and PCC representation under identical conditions. The main structural and environmental parameters are as follows: rigid self-propulsion module: mass 18.6 kg, length 300 mm, radius 220 mm, center of buoyancy in the body frame [100 mm, 0 mm, 0 mm], center of mass [100 mm, 0 mm, −100 mm], and distance from the propeller center to the rigid-body center 76 mm; soft joint: mass 9.2 kg, joint length 300 mm, radius 220 mm, and actuator Young's modulus 2.048×10^5 Pa; hydrodynamics: cross-sectional added-mass coefficient 1.28,

cross-sectional drag coefficient 10.26, lateral drag coefficient 12.83, fluid density 1000 kg/m^3, water speed 0.5 m/s, gravity vector [0 m/s^2, 0 m/s^2, −9.81 m/s^2], cross-sectional drag coefficient 10.26, lateral drag coefficient 12.83, and cross-sectional added-mass coefficient 1.28.

To evaluate the effectiveness of the proposed SUJ model, static simulations were conducted to compare with the conventional CC and PCC models under a 0.5 m/s flow disturbance. Three representative robot configurations were considered: (I) a straight state, (II) one end bent at π/6, and (III) both ends bent at π/6, as illustrated in Fig. 7. *X*, *Y*, and *Z* are the transformed *X*-, *Y*-, and *Z*-coordinates relative to the f_I frame. Results show that in the straight state (Fig. 7a), the CC, PCC, and SUJ models predict similar end-effector displacements, indicating that the classical model remains valid under small deformation conditions. However, when the robot is subjected to asymmetric or compounded bending (Figs. 7b and 7c), clear discrepancies emerge. Specifically, the CC or PCC model significantly underestimates the end-effector displacement, while the SUJ model maintains consistent accuracy across all configurations. This demonstrates that the SUJ model captures the nonlinear coupling between bending and axial deformation more effectively, providing a reliable description even under large curvatures and complex loading conditions.

Furthermore, to capture the robot's motion for quantitative comparisons with simulations, the Qualisys motion-capture system was employed. In the spiral motion experiment, two soft joints were instructed to bend upward-right by 30°, while the propellers applied a forward thrust of 10 N. The recorded trajectory is shown in Fig. 8a, with simulation–experiment comparisons provided in Fig. 8b. The simulated end-effector position and orientation corresponding to this maneuver are illustrated in Figs. 8c and 8d, respectively. To further examine the generality of the model, a pitch-based upward maneuver was tested. The experimental motion record of this is given in Fig. 8e, and the corresponding end-effector position and orientation profiles are shown in Figs. 8f and 8g, respectively. In addition, a linear motion was performed under propeller thrusts of 10 and 20 N (Fig. 8h), with trajectory comparisons presented in Fig. 8i. Finally, Fig. 8j presents quantitative model error analysis between the experimental data and simulations using the SUJ and CC models across all three motion types. The results show that the error increases as the motion becomes more complex. Nevertheless, the SUJ model consistently yields higher accuracy, particularly under conditions

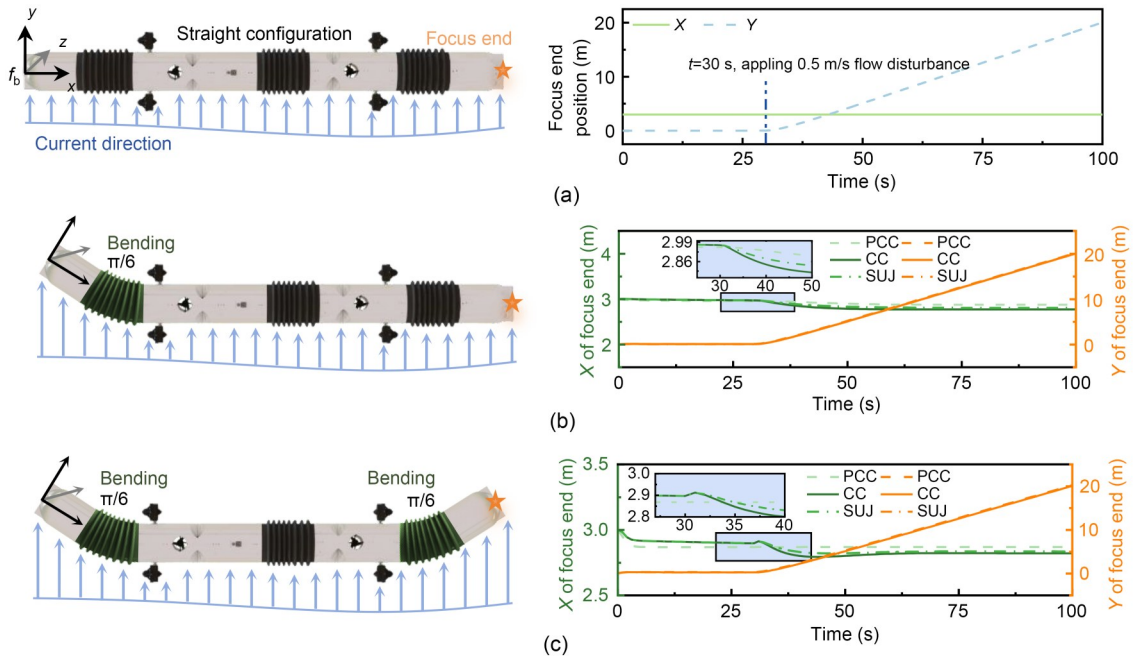


Fig. 7 Static simulation comparison of the HRFUM under CC, PCC, and SUJ models with 0.5 m/s flow disturbance: (a) end-effector displacement in a straight configuration; (b) end-effector displacement with one end bent at π/6; (c) end-effector displacement with both ends bent at π/6

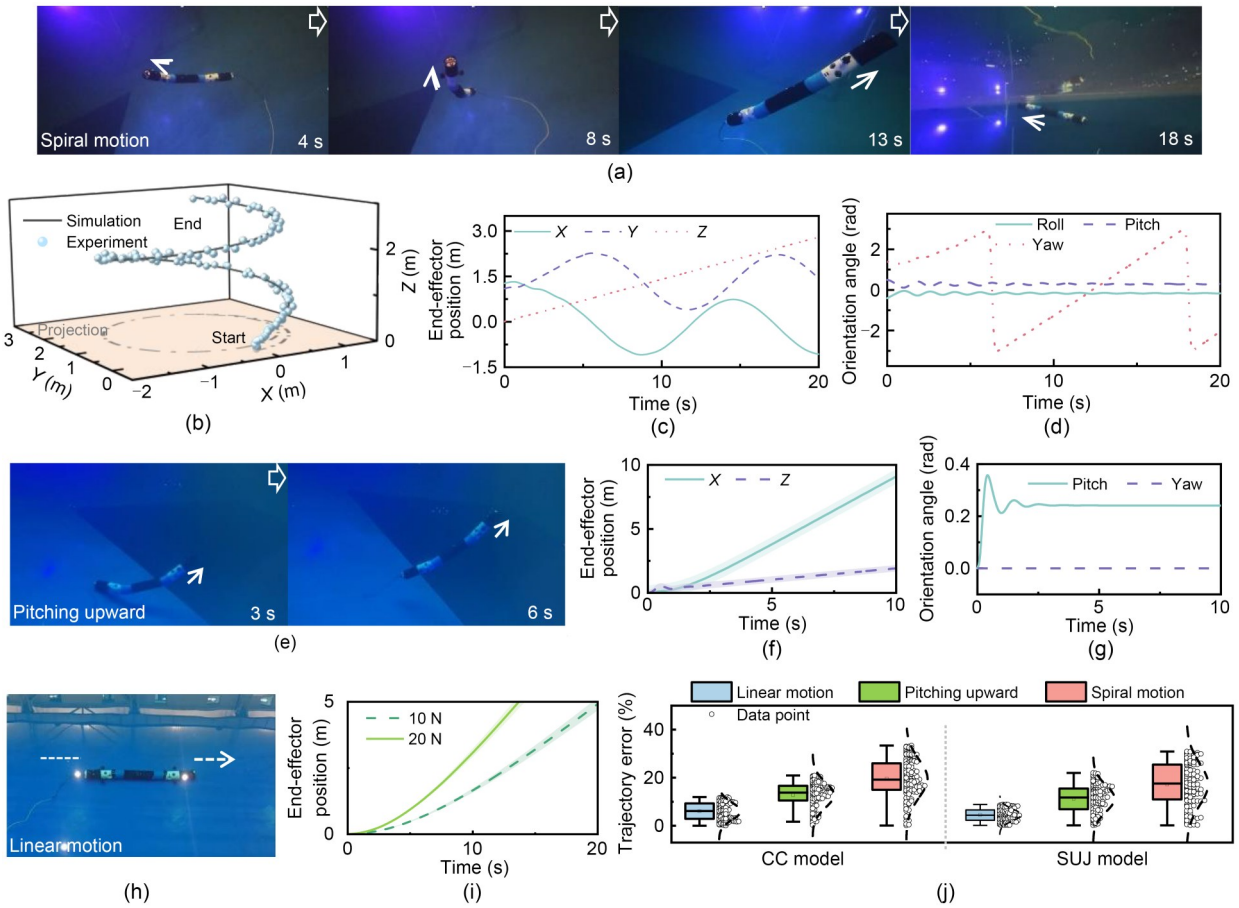


Fig. 8 Dynamic motion simulations and experimental validation of HRFUM: (a) experimental recording of spiral motion; (b) comparison between simulation and experimental results of the spiral motion trajectory; (c and d) positions and orientation angles of the end-effector during spiral motion; (e) experimental recording of pitching upward; (f and g) positions and orientation angles of the end-effector for pitching upward; (h) experimental recording of linear motion; (i) linear forward motion trajectories corresponding to 10 and 20 N propeller thrusts; (j) statistical comparison of trajectory errors between the SUJ and CC models across different motion modes

where external loads induce non-CC deformation. This highlights SUJ's ability to capture load-dependent shape changes and motion characteristics, whereas the CC model is fundamentally limited to CC deformations, and therefore exhibits reduced fidelity in scenarios involving payload handling or environmental contact.

Beyond the specific motions validated in this study, the hybrid rigid–soft design inherently supports a variety of locomotion patterns such as undulatory swimming, sidewinding, and pivoting rolling turns. These modes extend the versatility of the HRFUM and will be explored in future work.

3.3 Controller validation

As a baseline evaluation, we first conducted simulations of end-effector trajectory tracking, considering only the primary vector task without any task priority control mechanism. The results of this analysis can be found in Section S5 of the ESM. Next, to validate the proposed priority-based trajectory tracking controller, three representative scenarios were simulated under the scalar task not activated. These modes highlight the versatility of the control framework in simultaneously managing end-effector and base behaviors.

Mode 1: end-effector trajectory tracking with a fixed base.

In the first scenario, the base remained fixed, while the end-effector tracked a predefined trajectory (Fig. 9a). This situation enables the robot to imitate an industrial robotic arm in order to perform end-tasks in water, such as the scene of one end being anchored while the other end being operated, as shown in Fig. 1. Starting from the initial state with the base at [0, 0, 0] and end at [3, 0, 0], the target trajectory was defined

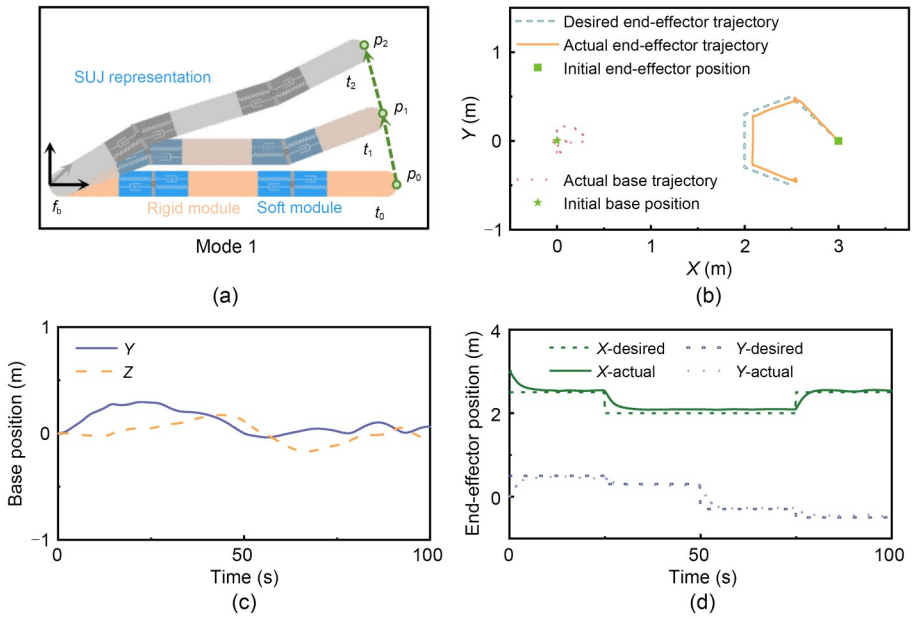


Fig. 9 Simulation results of HRFUM trajectory tracking under mode 1: (a) schematic diagram of end-effector trajectory tracking with a fixed base; (b) overall trajectory; (c) base trajectory tracking; (d) end-effector trajectory tracking. t_0 is the initial time; t_1 and t_2 are the two time points after the start; p_1, p_2, and p_3 are the positions corresponding to t_0, t_1, and t_2, respectively

as follows: at t=25 s, $\boldsymbol{\sigma}_{1,d}$=[2.5, 0]; at t=50 s, $\boldsymbol{\sigma}_{1,d}$=[2.0, 0.3]; at t=75 s, $\boldsymbol{\sigma}_{1,d}$=[2.5, −0.3]; at t=100 s, $\boldsymbol{\sigma}_{1,d}$=[2.5, −1.0]; $\boldsymbol{\sigma}_{2,d}\equiv$[0, 0, 0]. The PID parameters were tuned as follows: $K_{p,1}$=0.2, $K_{i,1}$=0.4, and $K_{d,1}$=0.8 for the end-effector vector task $\boldsymbol{\sigma}_1$; $K_{p,2}$=0.4, $K_{i,2}$=0.3, and $K_{d,2}$=0.8 for the base vector task $\boldsymbol{\sigma}_2$. The simulation results (Figs. 9b–9d) show that while minor deviations occurred in the base, the controller effectively constrained its motion, ensuring precise end-effector tracking. This showcases the feasibility of using the proposed robot in anchored operations or confined hovering tasks, where conventional ROV manipulators face limitations.

Mode 2: coordinated trajectory tracking of both the base and end-effector.

The second scenario examined simultaneous motion of the base and end-effector along predefined trajectories (Fig. 10a). This configuration simulates wide-range locomotion and turning. The initial base and end-effector positions were [0, 0, 0] and [3, 0, 0], respectively. The desired trajectories were set as follows: at t=50 s, $\boldsymbol{\sigma}_{1,d}$=[3.0, 0.0, 0.0, 0.0, 0.0, 0.0] and $\boldsymbol{\sigma}_{2,d}$=[0, 0, 0]; at t=100 s, $\boldsymbol{\sigma}_{1,d}$=[2.5, −0.5, 0.0, 0.0, 0.0, 0.0] and $\boldsymbol{\sigma}_{2,d}$=[1, −1, 0]. With the PID parameters tuned to $K_{p,1}$=0.22, $K_{i,1}$=0.01, $K_{d,1}$=0.02, $K_{p,2}$=0.04, $K_{i,2}$=0.03, and $K_{d,2}$=0.08, the tracking accuracy was improved. The overall trajectory is shown in Fig. 10b, along with detailed comparisons of the base trajectories in Fig. 10c and the end-effector trajectories in Fig. 10d. The results confirm that the robot can coordinate the base and end-effector simultaneously, thereby enhancing its maneuverability in complex underwater environments.

Mode 3: end-effector orientation tracking with a fixed base position.

The third scenario focused on controlling the end-effector's orientation while keeping the base position fixed (Fig. 11a). This task reflects the need for enhanced visual sensing in narrow underwater spaces. The desired trajectories were defined as: at t=50 s, $\boldsymbol{\sigma}_{1,d}$=[3, 0, 0, 0, 0, 0] and $\boldsymbol{\sigma}_{2,d}$=[0, 0, 0]; at t=100 s, $\boldsymbol{\sigma}_{1,d}$=[3, 0, 0, 0, π/3, π/6] and $\boldsymbol{\sigma}_{2,d}$=[0, 0, 0]. The controller parameters were set as follows: $K_{p,1}$=0.002, $K_{i,1}$=0.040, $K_{d,1}$=0.040, $K_{p,2}$=0.520, $K_{i,2}$=0.002, and $K_{d,2}$=0.010. The simulation results are presented in Figs. 11b and 11c. Although the orientation tracking was successful, a positional deviation of approximately 0.3 m occurred. This offset is inevitable, as changes in orientation induce soft joint deformation. Nevertheless, these results demonstrate the robot's robust capability for attitude control, which is critical for perception and exploration in constrained environments.

Finally, the tracking errors of all three modes were statistically analyzed, with results shown in Fig. 11d. As illustrated, the reported errors only concern the primary task variables (e.g., the end-effector trajectory in

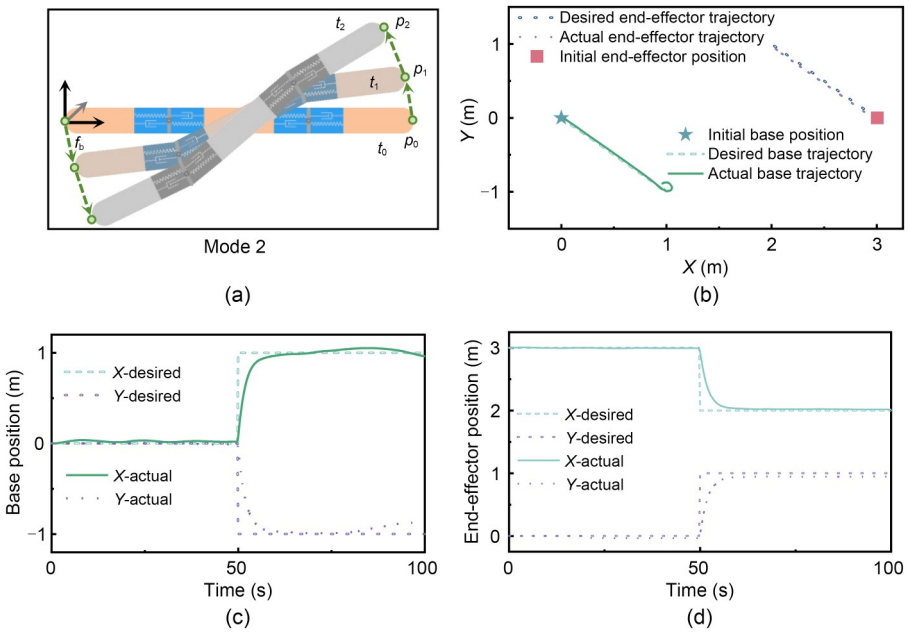


Fig. 10 Simulation results of HRFUM trajectory tracking under mode 2: (a) schematic diagram of coordinated trajectory tracking of both the base and end-effector; (b) overall trajectory; (c) base trajectory tracking; (d) end-effector trajectory tracking

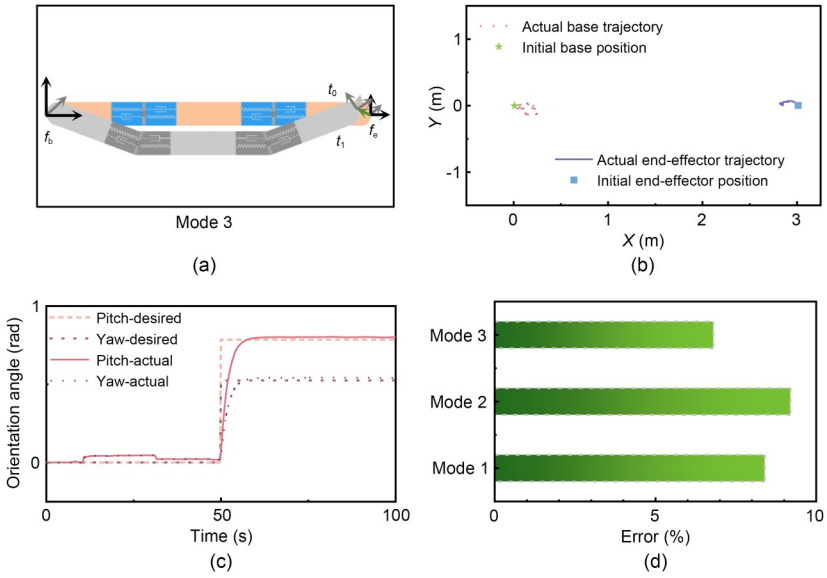


Fig. 11 (a–c) Simulation results of HRFUM trajectory tracking under mode 3: (a) schematic diagram of end-effector orientation tracking with a fixed base position; (b) overall trajectory; (c) orientation tracking; (d) statistical comparison of the tracking errors across modes 1–3

mode 1). These findings highlight that while error tends to increase with task complexity, the priority-based control framework ensures reliable tracking performance across different operational scenarios, validating its effectiveness for real underwater applications.

The three representative scenarios validate the effectiveness of the proposed priority-based control framework when only vector tasks are activated. Mode 1 demonstrates that the robot can achieve precise end-effector trajectory tracking while constraining the base; this enables anchored operations and confined hovering tasks that conventional underwater manipulators cannot easily perform. Mode 2 highlights the robot's ability to coordinate base and end-effector

motions simultaneously, thereby improving maneuverability and adaptability in terms of wide-range locomotion, turning, and obstacle-avoidance maneuvers. Mode 3 further confirms the controller's abilities in orientation tracking, ensuring robust attitude regulation that is critical for sensing and exploration in constrained spaces. Collectively, these scenarios establish that the vector-task-based priority control framework provides versatile and reliable motion regulation across diverse underwater tasks. Even as the task complexity increases, the framework preserves tracking accuracy for the primary objectives while flexibly accommodating secondary tasks. This validates the feasibility of the proposed approach for real-world underwater operations, ranging from precise manipulations to large-scale navigation and environmentally aware perceptions.

To further evaluate the proposed priority-based trajectory tracking controller, we considered scenarios in which scalar tasks were activated. In such cases, the controller needed to not only satisfy vector tasks but also handle scalar constraints, such as obstacle avoidance and joint limit enforcement. In particular, two representative cases were studied.

Case 1: obstacle avoidance during large-scale trajectory tracking.

The first case simulated the robot tracking a large-scale trajectory in a cluttered environment. Initially, the robot was fully extended in a straight configuration, with the base at [0, 0, −4] and the end-effector at [3, 0, −4], i.e., at a depth of 4 m. The simulation time was set to 100 s, during which the desired end-effector positions were sequentially updated every 20 s as follows: [5, 2, 4]→[7, 0, −4]→[9, −2, −4]→[11, 0, 4]. Two spherical obstacles were placed at [5, 0, 4] and [9, 0, −4], each with a radius of r_o=1.0 m. A safety margin of r_s=0.1 m was defined, resulting in a collision-free region $D(\sigma_a)=[r_o+r_s, \infty]$. The joint angle limits were set to ±100°, which means $D(\boldsymbol{\sigma}_b)=[-100, 100]$. After PID parameter tuning, the controller achieved satisfactory performance with the following gains: scalar task: $K_{p,*}$=0.0002, $K_{i,*}$=0.0060, and $K_{d,*}$=0.0090; vector task for end-effector $\boldsymbol{\sigma}_1$: $K_{p,1}$=0.0340, $K_{i,1}$=0.0068, and $K_{d,1}$=0.0075; vector task for base $\boldsymbol{\sigma}_2$: $K_{p,2}$=0.0520, $K_{i,2}$=0.0020, $K_{d,2}$=0.0030. The 3D simulation results are depicted in Fig. 12a, where the black points indicate the target trajectory, and the yellow and blue curves represent the actual trajectories of the end-effector and base, respectively. The top view (Fig. 12b) illustrates that the robot successfully followed the planned path while avoiding collisions. For clarity, the trajectories of the intermediate joints were not plotted in the global view; instead, their angular variations are analyzed separately in Fig. 12c. It is clear that at t= 40 and 80 s, joint angles exceeded the 100° threshold, which triggered the joint-limit scalar task. Once activated, the controller immediately redistributed thrust and torque to prevent further violations. The tracking errors of the end-effector and base are shown in

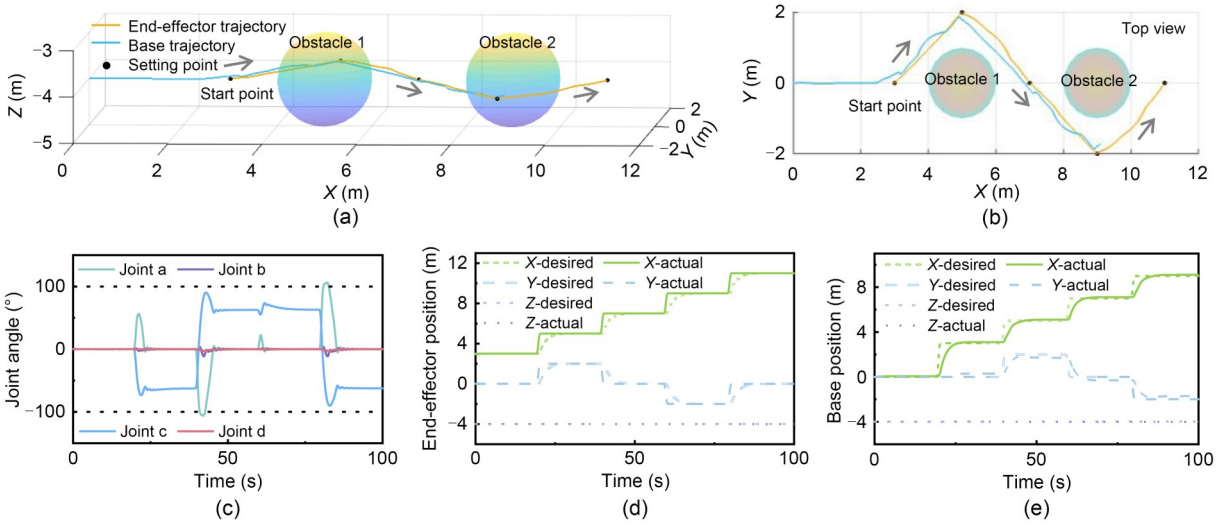


Fig. 12 Simulation results of HRFUM trajectory tracking under case 1: (a) end-effector trajectory tracking with obstacle avoidance during wide-range locomotion; (b) top view of (a); (c) joint angle variations; (d) end-effector tracking; (e) base tracking. Joints a–d are the four virtual joints based on SUJ model corresponding to Fig. 3e, which are the two prismatic joints and two revolute joints

Figs. 12d and 12e, respectively. As expected, the end-effector exhibited smaller errors compared to the base. This reflects the task prioritization strategy: the end-effector was required to strictly follow the desired trajectory, whereas the base was allowed to deviate to ensure avoidance of potential obstacles. The relatively larger average error was mainly due to the need to balance trajectory accuracy with collision safety. Notably, in scenarios without obstacles, the tracking precision would be significantly improved.

Case 2: end-effector trajectory tracking with a fixed base under obstacle constraints.

The second case extends mode 1 (e.g., end-effector trajectory tracking with a fixed base) by introducing obstacles into the environment. In this scenario, the robot was required to follow the desired end-effector trajectory while the base remained stationary, simulating task execution in cluttered underwater environments. The simulation was conducted under planar motion conditions. The robot was initially in a straight configuration, with the base and end-effector located at [0.0, 0.0] and [3.0, 0.0], respectively; the simulation time was set to 100 s. The desired end-effector trajectory was defined as follows: at t=33 s, the end-effector moved to [1.5, 1.0]; at t=66 s, it moved further to [0.0, 2.0]. The base position was fixed throughout the experiment, and the flow velocity was set to 0 m/s. Two circular obstacles of different sizes were introduced, with radii of 0.25 and 0.45 m, centered at [2.25, 0.65] and [0.90, 1.70], respectively. Notably, these obstacles overlapped with the desired end-effector trajectory, creating potential collisions. A safety margin of 0.1 m was imposed to ensure collision-free motion. After controller tuning, satisfactory performance was obtained with the following gains: scalar task: $K_{p,*}$= 0.550, $K_{i,*}$=0.043, and $K_{d,*}$=0.078; vector task for the end-effector $\boldsymbol{\sigma}_1$: $K_{p,1}$=0.850, $K_{i,1}$=0.260, and $K_{d,1}$=0.370; vector task for base $\boldsymbol{\sigma}_2$: $K_{p,2}$=0.036, $K_{i,2}$=0.530, and $K_{d,2}$= 0.014. The simulation results for this are presented in Fig. 13a. The two circles denote the obstacles; the yellow dashed line is the desired end-effector trajectory; the yellow solid line represents the actual end-effector trajectory; and the blue line shows the base trajectory. The results indicate that the end-effector successfully followed the prescribed trajectory while avoiding obstacles, with minimal displacement of the base, thus mimicking the operational characteristics of an industrial manipulator. In terms of end-effector tracking accuracy (Fig. 13b): the mean errors were 8.32% in the x-direction and 7.24% in the y-direction, respectively. Distance to the obstacle boundaries (Fig. 13c): at t=38 s and t=72 s, the distance reached or slightly exceeded the safety threshold, which triggered the obstacle avoidance scalar task. Joint angle evolution (Fig. 13d) confirms that scalar constraints were respected throughout most of the trajectory. Activation status of scalar tasks (Fig. 13e): labels “N”, “A1”, and “A2” represent no scalar task, joint-limit task, and

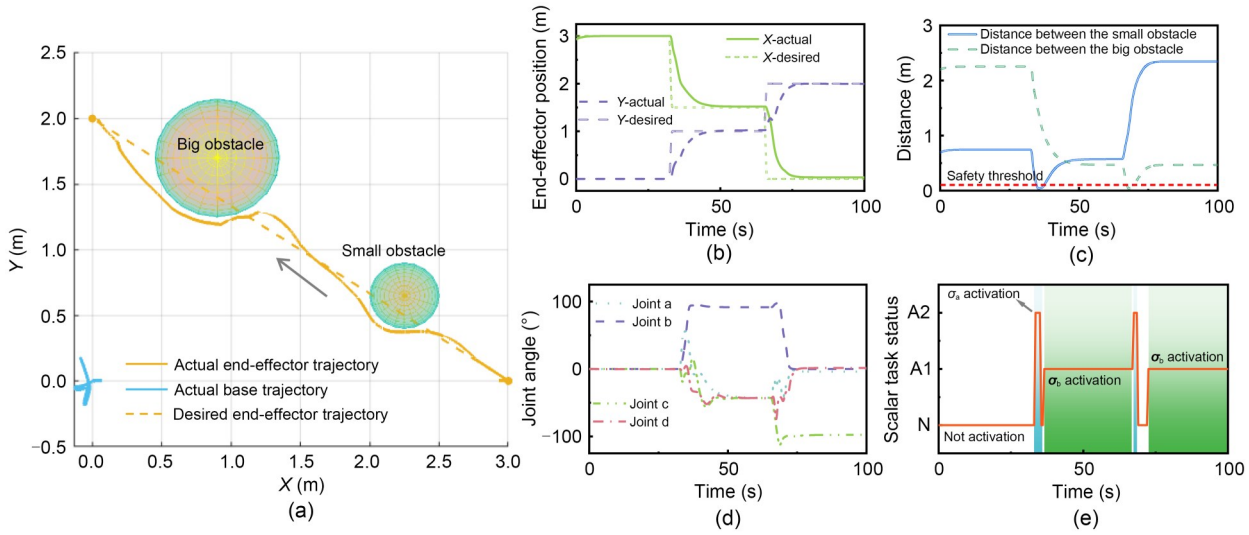


Fig. 13 Simulation results of HRFUM trajectory tracking under case 2: (a) end-effector trajectory tracking with a fixed base in the presence of obstacles; (b) end-effector tracking; (c) real-time distances between the end-effector and obstacle boundaries; (d) joint angle variations; (e) scalar task activation status

obstacle-avoidance task, respectively. At the identified times, A2 was activated, and the controller redistributed the forces and torques to gradually steer the end-effector away from the obstacles. Overall, this case demonstrates that the priority-based framework enables seamless integration of scalar tasks with vector tasks, ensuring that the primary trajectory tracking objective is preserved while the avoidance of collisions is simultaneously guaranteed.

Through the two representative cases, one can conclude that the activation of scalar tasks plays a critical role in enhancing safety and adaptability. The results verify that the proposed priority-based control framework can seamlessly integrate scalar and vector tasks, enabling the robot to maintain precise trajectory tracking while dynamically handling joint-limit violations and avoiding collisions. This capability is particularly important for underwater environments, where unpredictable obstacles and physical constraints are common; it therefore underscores the applicability of the proposed control scheme.

4 Conclusions

This study presented the design, modeling, and control of an HRFUM system that integrates rigid self-propulsion modules with soft joint modules inspired by biological locomotion. To accurately capture the robot's kinematic and dynamic behaviors, especially under large deformations, a novel SUJ model was established and validated against simulations and experiments. Comparative analysis with the conventional CC model demonstrated that the SUJ model maintains high accuracy in both normal and extreme curvature cases, highlighting its advantages in representing real underwater operational conditions. Regarding control, a priority-based trajectory tracking framework was developed to handle the coexistence of vector and scalar tasks. The controller was validated through simulations under multiple scenarios, including fixed-base manipulation, coordinated base–end-effector motion, attitude control, and scalar-task activation for collision-free operations. The results confirmed that the proposed framework can effectively balance conflicting task demands while ensuring robustness and safety.

Acknowledgments

This work is financially supported in part by the National Natural Science Foundation of China (No. 52271353), the Fundamental Research Program of Shenyang Institute of Automation, Chinese Academy of Sciences (Nos. 2025JC1K13, 2023JC3G03, and 2022JC3K05), China, and the State Key Laboratory of Robotics and Intelligent Systems, Shenyang Institute of Automation, Chinese Academy of Sciences (Nos. 2025-Z08 and 2024-Z15), China.

Author contributions

Junhao ZHANG and Yinglong CHEN designed the research. Junhao ZHANG wrote the first draft of the manuscript. Jiancheng YU helped to organize the manuscript. Jiancheng YU and Yongjun GONG revised and edited the final version. Yu TIAN, Qi ZHANG, Jie SUN, Xiaolong YU, and Shilong LI processed the corresponding data.

Conflict of interest

Junhao ZHANG, Jiancheng YU, Yinglong CHEN, Yu TIAN, Qi ZHANG, Jie SUN, Xiaolong YU, Shilong LI, and Yongjun GONG declare that they have no conflict of interest.

References

Antonelli G, 2014. Modelling of underwater robots. *In*: Antonelli G (Ed.), Underwater Robots. Springer, Cham, Switzerland, p.23-63. https://doi.org/10.1007/978-3-319-02877-4_2

Bao PX, Shi LW, Duan LJ, et al., 2023. A review: from aquatic lives locomotion to bio-inspired robot mechanical designations. *Journal of Bionic Engineering*, 20(6):2487-2511. https://doi.org/10.1007/s42235-023-00421-2

Bouyarmane K, Kheddar A, 2018. On weight-prioritized multitask control of humanoid robots. *IEEE Transactions on Automatic Control*, 63(6):1632-1647. https://doi.org/10.1109/TAC.2017.2752085

Chen G, Ti X, Shi LC, et al., 2022. Design of beaver-like hind limb and analysis of two swimming gaits for underwater narrow space exploration. *Journal of Intelligent & Robotic Systems*, 104(4):65. https://doi.org/10.1007/s10846-022-01610-7

Chen YH, Wang T, Wu CF, et al., 2021. Design, control, and experiments of a fluidic soft robotic eel. *Smart Materials and Structures*, 30(6):065001. https://doi.org/10.1088/1361-665X/abf5ef

Chen YL, Zhang JH, Gong YJ, 2019. Utilizing anisotropic fabrics composites for high-strength soft manipulator integrating soft gripper. *IEEE Access*, 7:127416-127426. https://doi.org/10.1109/ACCESS.2019.2940499

Chen YL, Li WS, Gong YJ, 2021. Static modeling and analysis of soft manipulator considering environment contact based on segmented constant curvature method. *Industrial Robot*, 48(2):233-246. https://doi.org/10.1108/IR-07-2020-0131

Chin K, Hellebrekers T, Majidi C, 2020. Machine learning for

soft robotic sensing and control. *Advanced Intelligent Systems*, 2(6):1900171. https://doi.org/10.1002/aisy.201900171

Dai HL, He YX, Zhou K, et al., 2022. Utilization of nonlinear vibrations of soft pipe conveying fluid for driving underwater bio-inspired robot. *Applied Mathematics and Mechanics*, 43(7):1109-1124. https://doi.org/10.1007/s10483-022-2866-7

Della Santina C, Bicchi A, Rus D, 2020. On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control. *IEEE Robotics and Automation Letters*, 5(2):1001-1008. https://doi.org/10.1109/LRA.2020.2967269

Du T, Hughes J, Wah S, et al., 2021. Underwater soft robot modeling and control with differentiable simulation. *IEEE Robotics and Automation Letters*, 6(3):4994-5001. https://doi.org/10.1109/LRA.2021.3070305

He ZG, Yang Y, Jiao PC, et al., 2023. Copebot: underwater soft robot with copepod-like locomotion. *Soft Robotics*, 10(2):314-325. https://doi.org/10.1089/soro.2021.0158

Hong MQ, Zhou H, Liu LX, et al., 2022. Coordinated control of free-floating dual-arm space robots based on hybrid task-priority approach. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 236(11):2206-2217. https://doi.org/10.1177/09544100211053537

Iversflaten MH, Haraldsen A, Pettersen KY, 2022. Kinematic and dynamic control of cooperating underwater vehicle-manipulator systems. *IFAC-PapersOnLine*, 55(31):110-117. https://doi.org/10.1016/j.ifacol.2022.10.417

Jiao PC, Zhang CJ, Meng WX, et al., 2025. Artificial intelligence-guided inverse design of deployable thermo-metamaterial implants. *ACS Applied Materials & Interfaces*, 17(2):2991-3001. https://doi.org/10.1021/acsami.4c17625

Kanoun O, Lamiraux F, Wieber PB, 2011. Kinematic control of redundant manipulators: generalizing the task-priority framework to inequality task. *IEEE Transactions on Robotics*, 27(4):785-792. https://doi.org/10.1109/TRO.2011.2142450

Katzschmann RK, Delpreto J, Maccurdy R, et al., 2018. Exploration of underwater life with an acoustically controlled soft robotic fish. *Science Robotics*, 3(16):eaar3449. https://doi.org/10.1126/scirobotics.aar3449

Kelasidi E, Liljeback P, Pettersen KY, et al., 2016. Innovation in underwater robots: biologically inspired swimming snake robots. *IEEE Robotics & Automation Magazine*, 23(1):44-62. https://doi.org/10.1109/MRA.2015.2506121

Liu KX, Ding MX, Pan BY, et al., 2024. A maneuverable underwater vehicle for near-seabed observation. *Nature Communications*, 15(1):10284. https://doi.org/10.1038/s41467-024-54600-8

Moe S, Antonelli G, Teel AR, et al., 2016. Set-based tasks within the singularity-robust multiple task-priority inverse kinematics framework: general formulation, stability analysis, and experimental results. *Frontiers in Robotics and AI*, 3:16. https://doi.org/10.3389/frobt.2016.00016

Mohan S, Kim J, 2012. Indirect adaptive control of an autonomous underwater vehicle-manipulator system for underwater manipulation tasks. *Ocean Engineering*, 54:233-243. https://doi.org/10.1016/j.oceaneng.2012.07.022

Nakajima K, Hauser H, Li T, et al., 2018. Exploiting the dynamics of soft materials for machine learning. *Soft Robotics*, 5(3):339-347. https://doi.org/10.1089/soro.2017.0075

Plum F, Labisch S, Dirks JH, 2020. SAUV—a bio-inspired soft-robotic autonomous underwater vehicle. *Frontiers in Neurorobotics*, 14:8. https://doi.org/10.3389/fnbot.2020.00008

Podder TK, Sarkar N, 2004. A unified dynamics-based motion planning algorithm for autonomous underwater vehicle-manipulator systems (UVMS). *Robotica*, 22(1):117-128. https://doi.org/10.1017/S0263574703005368

Pourghasemi Hanza S, Ghafarirad H, 2024. Mechanics of fiber-reinforced soft manipulators based on inhomogeneous cosserat rod theory. *Mechanics of Advanced Materials and Structures*, 31(15):3161-3173. https://doi.org/10.1080/15376494.2023.2170498

Qu JT, Xu YN, Li ZK, et al., 2024. Recent advances on underwater soft robots. *Advanced Intelligent Systems*, 6(2):2300299. https://doi.org/10.1002/aisy.202300299

Rizzini DL, Kallasi F, Aleotti J, et al., 2017. Integration of a stereo vision system into an autonomous underwater vehicle for pipe manipulation tasks. *Computers & Electrical Engineering*, 58:560-571. https://doi.org/10.1016/j.compeleceng.2016.08.023

Shih B, Shah D, Li JX, et al., 2020. Electronic skins and machine learning for intelligent soft robots. *Science Robotics*, 5(41):eaaz9239. https://doi.org/10.1126/scirobotics.aaz9239

Simetti E, Casalino G, 2017. Manipulation and transportation with cooperative underwater vehicle manipulator systems. *IEEE Journal of Oceanic Engineering*, 42(4):782-799. https://doi.org/10.1109/JOE.2016.2618182

Simetti E, Casalino G, Wanderlingh F, et al., 2018. Task priority control of underwater intervention systems: theory and applications. *Ocean Engineering*, 164:40-54. https://doi.org/10.1016/j.oceaneng.2018.06.026

Sun LX, Wang Y, Hui XL, et al., 2024. Underwater robots and key technologies for operation control. *Cyborg and Bionic Systems*, 5:0089. https://doi.org/10.34133/cbsystems.0089

Till J, Aloi V, Rucker C, 2019. Real-time dynamics of soft and continuum robots based on cosserat rod models. *The International Journal of Robotics Research*, 38(6):723-746. https://doi.org/10.1177/0278364919842269

Vangi M, Topini E, Liverani G, et al., 2025. Design, development, and testing of an innovative autonomous underwater reconfigurable vehicle for versatile applications. *IEEE*

Journal of Oceanic Engineering, 50(2):509-526. https://doi.org/10.1109/JOE.2024.3511709

Wang LL, Zhu DQ, Pang W, et al., 2023. A survey of underwater search for multi-target using multi-AUV: task allocation, path planning, and formation control. *Ocean Engineering*, 278:114393. https://doi.org/10.1016/j.oceaneng.2023.114393

Wang XR, Sha ZY, Zhang FT, 2025. Adaptive integral sliding mode control for attitude tracking of underwater robots with large range pitch variations in confined spaces. *IEEE Robotics and Automation Letters*, 10(2):979-986. https://doi.org/10.1109/LRA.2024.3515733

Wang ZC, Wang GT, Chen XP, et al., 2023. Dynamic modeling and control of a soft robotic arm using a piecewise universal joint model. IEEE International Conference on Robotics and Biomimetics, p.1-6. https://doi.org/10.1109/ROBIO58561.2023.10354732

Webster III RJ, Jones BA, 2010. Design and kinematic modeling of constant curvature continuum robots: a review. *The International Journal of Robotics Research*, 29(13):1661-1683. https://doi.org/10.1177/0278364910368147

Wu HY, Chen YL, Yang QM, et al., 2024. A review of underwater robot localization in confined spaces. *Journal of Marine Science and Engineering*, 12(3):428. https://doi.org/10.20944/preprints202312.2175.v1

Wu MX, Afridi WH, Wu JX, et al., 2024. Octopus-inspired underwater soft robotic gripper with crawling and swimming capabilities. *Research*, 7:0456. https://doi.org/10.34133/research.0456

Xun LX, Zheng G, Kruszewski A, 2024. Cosserat-rod-based dynamic modeling of soft slender robot interacting with environment. *IEEE Transactions on Robotics*, 40:2811-2830. https://doi.org/10.1109/TRO.2024.3386393

Yang Y, Ren HL, Jiao PC, et al., 2024. How do combustions actuate high-speed soft robots? *Soft Robotics*, 11(6):911-923. https://doi.org/10.1089/soro.2023.0168

Zhang CJ, Ma HK, Chen ZC, et al., 2024. YOLOX-DG robotic detection systems for large-scale underwater concrete structures. *iScience*, 27(4):109337. https://doi.org/10.1016/j.isci.2024.109337

Zhang JH, Chen YL, Liu Y, et al., 2022. Dynamic modeling of underwater snake robot by hybrid rigid-soft actuation. *Journal of Marine Science and Engineering*, 10(12):1914. https://doi.org/10.3390/jmse10121914

Zhang JH, Chen YL, Gong YJ, 2024. Kinematics, dynamics, and control of underwater robotic snake based on a rigid–soft unified model. *Ships and Offshore Structures*, 19(9):1408-1423. https://doi.org/10.1080/17445302.2023.2247131

Zhao JX, Gao XJ, Xiao YM, et al., 2024. A fish-like underwater miniature robot capable of high-speed and controllable locomotion. *IEEE Robotics and Automation Letters*, 9(4):3910-3917. https://doi.org/10.1109/LRA.2024.3372443

Zhao Q, Yang TF, Tang GQ, et al., 2025. Bio-inspired swarm of underwater robots: a review. *Bioinspiration & Biomimetics*, 20(4):041002. https://doi.org/10.1088/1748-3190/ade215

Zhao W, Zhou C, Zhang ZW, et al., 2025. The South China Sea mooring array and its applications in exploring oceanic multiscale dynamics. *Science Bulletin*, 70(5):609-623. https://doi.org/10.1016/j.scib.2024.12.008

Electronic supplementary materials

Sections S1–S5