Electronic supplementary materials

For https://doi.org/10.1631/jzus.A2500507

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 Laboratory 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*

Section S1

The tunable‐stiffness capability is achieved by incorporating closed‐loop pressure regulation within the hydraulic circuit combined with coordinated actuation of the three actuator chambers. Starting from the angular velocity ω_i $i \in [1\ 2\ 3]$ of each brushless‐DC motor driving gear pump and the gear pump delivers flow rate proportional to speed

$$Q_{p,i} = \alpha_p \omega_i \tag{S1}$$

with volumetric displacement coefficient α_p. This flow is metered by the proportional throttle valve, whose orifice equation is

$$Q_{v,i} = C_d A_{v,i}(u_i)\sqrt{\frac{2P_i}{\rho}} \tag{S2}$$

where C_d is the discharge coefficient, $A_{v,i}(u_i)$ is the valve opening area as a function of electrical command u_i, P_i is the soft actuator chamber pressure, ρ the fluid density. Under steady flow continuity $Q_{p,i}=Q_{v,i,}$ solve for P_i

$$P_i = \frac{\rho}{2}\left(\frac{\alpha_p \omega_i}{C_d A_{v,i}}\right)^2 \tag{S3}$$

Under quasi‐static conditions, the axial tensile force generated by a single muscle is approximated by $F_i=P_i A_{eff}$, where A_{eff} denotes the actuator's effective cross‐sectional area. Since the three actuators are equally spaced on a circle of radius r, their polar angular positions in the plane are φ_i=0°, 120°, 240°. When the soft segment bends to a curvature κ (i.e., radius $1/\kappa$), the axial length change of the i-th actuator is approximated by $\Delta l_i \approx -d\kappa l\cos\varphi_i$, where the negative sign indicates compression of the inside actuator

and tension of the outside actuator. If modeled as a linear spring, its axial stiffness can be expressed as

$$K_a(P_i) = \frac{F_i}{\Delta l_i(P_i)} = \frac{P_i A_{\text{eff}}}{\Delta l_i} \tag{S4}$$

For a material with intrinsic Young's modulus E and an undeformed length l, one may further approximate $K_a(P_i) \approx EA/l$. The bending stiffness can then be defined as $K_b(P_i) = F_i r/\kappa$. Bending stiffness is the primary mechanical parameter governing HRFUM performance, and this property was characterized in the subsequent performance analysis. All key parameters for the hydraulic actuation model can be seen in Table S1.

Table S1 Key parameters for the hydraulic actuation model.

Symbol	Parameter Description	Value	Source
C_d	Discharge coefficient of the proportional throttle valve	0.64	From proportional valve datasheet (iQ Coral valve)
A_v	Effective orifice area as a function of control signal	0–7.5×10^{-6} m^2	From manufacturer calibration data
α_p	Volumetric displacement coefficient of the pump	0.31 $cm^3 rev^{-1}$	From specification of micro gear pump (Jier A500-4N)
A_{eff}	Effective cross-sectional area of hydraulic actuator	4.2×10^{-4} m^2	Determined from static pressure–elongation tests
r	Radial distance between actuator chambers	24 mm	From prototype geometry
E	Effective Young's modulus	2.0485×10^5 Pa	From tensile test of the actuator
l	Undeformed length of the soft joint	300 mm	From prototype geometry

Section S2

Underwater execution layer: The execution layer is built around an STM32 microcontroller that integrates multiple onboard sensors, including an inertial measurement unit (IMU), Doppler Velocity Log (DVL), and depth sensor. These sensors provide real-time measurements of navigation attitude, velocity, and depth. The STM32's abundant PWM and GPIO interfaces directly drive the electronic speed controllers of the thrusters and the proportional valves of the soft joints, thereby decoupling propulsion and articulation control. Local closed-loop controllers are implemented on subordinate STM32 nodes, operating at a 1 ms cycle time, which guarantees low-latency response

and high robustness against hydrodynamic disturbances. A modular power management board that further supplies regulated outputs to thrusters, hydraulic pumps, and sensors, supporting uninterrupted underwater operation for more than 8 hours.

Surface Decision Layer: The decision layer employs an NVIDIA Jetson Xavier NX as its high-performance computing core. Physically embedded in the HRFUM body but connected to a topside workstation via Gigabit Ethernet, this layer executes global trajectory planning and mission-level decision making. Using the ROS middleware, it dispatches control parameters to the underwater execution layer while simultaneously collecting state feedback from the IMU, DVL, and onboard vision sensors. This architecture forms a complete closed-loop control system spanning perception, planning, and actuation. In addition, an integrated operator console—comprising a display and joystick—enables smooth switching between manual and semi-autonomous modes, offering both real-time adaptability and mission-level flexibility.

Section S3

A comparative analysis between the CC and SUJ models reveals the following properties:

Property 1. *For any position and orientation described by the CC model configuration variables* $[\varphi_i, \theta_i, L_i]$*, there always exists a corresponding set of SUJ model parameters* $[d_1^i, \theta_1^i, \theta_2^i, d_2^i]$*that produces an equivalent transformation, assuming* $d_1^i = d_2^i$.

Proof. In the CC model, the orientation of the x_{i+1}-axis of frame f_{i+1} is determined by first rotating frame $\overline{f_i}$ about the x_i-axis by angle θ_i, followed by a rotation about the new y-axis by φ_i. The resulting rotation is represented by the vector $(\cos\theta_i \sin\varphi_i, \sin\theta_i \sin\varphi_i, \cos\varphi_i)^{\mathrm{T}}$. Next, for the SUJ model, the orientation is obtained by first rotating around the z-axis by θ_1^i, and then around the x-axis by θ_2^i, yielding $(\sin\theta_1^i \cos\theta_2^i, -\sin\theta_2^i, \cos\theta_1^i \cos\theta_2^i)^{\mathrm{T}}$. Equating the two expressions and solving yields $\theta_1^i = \tan^{-1}(\cos\theta \tan\varphi), \theta_2^i = -\sin^{-1}(\sin\theta \sin\varphi)$. For position consistency, when $d_1^i = d_2^i = \frac{1}{\varphi}\tan(\frac{\varphi}{2})$, and the segment length L_i is matched accordingly, both models produce the same end-frame position and orientation. This confirms the theoretical equivalence of the CC and SUJ model representations under this mapping.

Property 2. *The set of positions and orientations representable by the CC model is a strict subset of those representable by the SUJ model.*

Proof. The subset relationship is directly followed from **Property 1**. To establish strict inclusion, we show that there exists a configuration represented by frame f_i, that can be realized by the SUJ model but cannot be described using the constant-curvature formulation. Specifically, consider a soft joint deformation confined to a plane, where the curvature along the segment is non-uniform (i.e., not constant). Under such conditions, it becomes impossible to fit an arc that simultaneously connects the base and tip positions while satisfying the orientation constraints at both ends. However, the SUJ model, due to its additional degrees of freedom, that is $d_1^i \neq d_2^i$can accommodate this configuration. This demonstrates that the CC model lacks the expressive capacity required for

general soft joint deformations, thereby confirming that it is a proper subset of the SUJ model space.

Property 3. *A minimum of four parameters is required to fully describe the position and orientation of the end frame of a soft joint that undergoes arbitrary planar deformation.*

Proof. For a body undergoing deformation in a two-dimensional plane, three parameters are necessary to describe its pose: two for position and one for orientation. When a soft joint deforms in three-dimensional space, one additional degree of freedom is needed to capture the azimuth of the deformation plane. Therefore, a total of four independent coordinates is required to represent arbitrary planar bending within 3D space.

Remark. The configuration space defined by the SUJ model forms a smooth manifold, enabling a one-to-one mapping with the Euclidean state space by simply setting $d_1^i=d_2^i$ and assuming $\theta_1^i=\theta_2^i=0$ in the undeformed state. In contrast, CC model relies on angular parameters to represent bending, which introduces artificial redundancies and limits model fidelity. These redundancies can lead to numerical instability in both the dynamic model and control computations, especially under perturbations or transitions between deformation modes.

To sum up, both CC and SUJ models perform similarly under straight or uniform curvature conditions. However, under complex, non-uniform deformations caused by variable external forces, the SUJ model achieves higher geometric fidelity while maintaining low computational overhead. This ensures accurate representation of soft joint kinematics without sacrificing real-time performance, making it highly suitable for underwater applications where rigid–soft coupling and rapid online computation are critical.

Section S4

In this work, vector tasks include controlling the position $\boldsymbol{p}_e$ and $\boldsymbol{\Theta}_e$ orientation of the end-effector, denoted as $\boldsymbol{\sigma}_1$

$$\boldsymbol{\sigma}_1=\begin{bmatrix}\boldsymbol{p}_e\\ \boldsymbol{\Theta}_e\end{bmatrix}\rightarrow\boldsymbol{\sigma}_{1,d}=\begin{bmatrix}\boldsymbol{p}_{e,d}\\ \boldsymbol{\Theta}_{e,d}\end{bmatrix} \tag{S5}$$

where $\boldsymbol{p}_{e,d}$ and $\boldsymbol{\Theta}_{e,d}$ represent desired position and orientation. To build the subsequent task priority control algorithm, it is necessary to further derive the first-order derivative and second-order derivative of the task $\boldsymbol{\sigma}_i$

$$\begin{aligned}\dot{\boldsymbol{\sigma}}_i&=\boldsymbol{J}_{\sigma i}\dot{\boldsymbol{\eta}}\\ \ddot{\boldsymbol{\sigma}}_i&=\dot{\boldsymbol{J}}_{\sigma i}\dot{\boldsymbol{\eta}}+\boldsymbol{J}_{\sigma i}\ddot{\boldsymbol{\eta}}\end{aligned} \tag{S6}$$

where $\boldsymbol{J}_{\sigma i}$ is the task Jacobian matrix corresponding to the vector task $\boldsymbol{\sigma}_i$, so the $\boldsymbol{J}_{\sigma 1}$ can be written as

$$\begin{aligned}\boldsymbol{J}_{\sigma 1} &= \begin{bmatrix} \boldsymbol{R}_I^b & \mathbf{0} \\ \mathbf{0} & \boldsymbol{H}_I^b \end{bmatrix} \boldsymbol{J}_e \\ \dot{\boldsymbol{J}}_{\sigma 1} &= \begin{bmatrix} \boldsymbol{R}_I^b S(\boldsymbol{w}_e) & \mathbf{0} \\ \mathbf{0} & \dot{\boldsymbol{H}}_I^b \end{bmatrix} \boldsymbol{J}_e + \begin{bmatrix} \boldsymbol{R}_I^b & \mathbf{0} \\ \mathbf{0} & \boldsymbol{H}_I^b \end{bmatrix} \dot{\boldsymbol{J}}_e \end{aligned} \tag{S7}$$

where matrices $\boldsymbol{R}_I^b$ and $\boldsymbol{H}_I^b$ map the linear velocity and angular velocity respectively, $\boldsymbol{J}_{\mathrm{e}}$ is the end-effector Jacobian matrix, which can be calculated iteratively, $S(\boldsymbol{w}_e)$ is the skew-symmetric matrix form of the angular velocity vector.

Similarly, base position control can also 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,

$$\boldsymbol{\sigma}_2 = \boldsymbol{p}_b \rightarrow \boldsymbol{\sigma}_{2,d} = \boldsymbol{p}_{b,d} \tag{S8}$$

The corresponding Jacobian $\boldsymbol{J}_{\sigma 2}$ only maps the base's translational motion and is given by

$$\begin{aligned}\boldsymbol{J}_{\sigma 2} &= \begin{bmatrix} \boldsymbol{R}_I^b & \mathbf{0} \end{bmatrix} \\ \dot{\boldsymbol{J}}_{\sigma 2} &= \begin{bmatrix} \boldsymbol{R}_I^b S(\boldsymbol{w}_e) & \mathbf{0} \end{bmatrix} \end{aligned} \tag{S9}$$

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

Collision avoidance (σ_a): maintaining a minimum distance between the end-effector and a spherical obstacle center $\boldsymbol{p}_o$,

$$\sigma_a = \sqrt{(\boldsymbol{p}_e - \boldsymbol{p}_o)^{\mathrm{T}} (\boldsymbol{p}_e - \boldsymbol{p}_o)} \tag{S10}$$

The construction of $\boldsymbol{J}_{\sigma a}$ and derivative can be obtained by taking the first-order and second-order derivative of σ_a,

$$\begin{aligned}\boldsymbol{J}_{\sigma a} &= -\frac{(\boldsymbol{p}_e - \boldsymbol{p}_o)^{\mathrm{T}}}{\sigma_a} \begin{bmatrix} \boldsymbol{R}_I^e & \mathbf{0}_{3\times 3} \end{bmatrix} \boldsymbol{J}_e \\ \dot{\boldsymbol{J}}_{\sigma a} &= (\boldsymbol{v}_e)^{\mathrm{T}} \left[\frac{\mathbf{I}}{\sigma_a} - \frac{(\boldsymbol{p}_e - \boldsymbol{p}_o)(\boldsymbol{p}_e - \boldsymbol{p}_o)^{\mathrm{T}}}{\sigma_a^3} \right] \begin{bmatrix} \boldsymbol{R}_I^e & \mathbf{0} \end{bmatrix} \boldsymbol{J}_e + \frac{(\boldsymbol{p}_I^e - \boldsymbol{p}_o)^{\mathrm{T}}}{\sigma_a} \left\{ \begin{bmatrix} \boldsymbol{R}_I^e S(\boldsymbol{w}_e) & \mathbf{0}_{3\times 3} \end{bmatrix} \boldsymbol{J}_e + \begin{bmatrix} \boldsymbol{R}_I^e & \mathbf{0} \end{bmatrix} \dot{\boldsymbol{J}}_e \right\} \end{aligned} \tag{S11}$$

Joint limit avoidance ($\boldsymbol{\sigma}_b$): keeping all joint angles within specified limits, that is $\boldsymbol{\sigma}_b = \boldsymbol{q}$, and $\boldsymbol{J}_{\sigma b} = [\mathbf{0}\ \mathbf{I}]$.

When σ_a or any component of $\boldsymbol{\sigma}_b$ approaches its threshold, the corresponding task becomes active and overrides lower-priority objectives to ensure safe operation.

Section S5

Three representative reference trajectories were selected to cover different levels of geometric complexity: a matrix-shaped trajectory (Fig. S1A), a pentagon-shaped trajectory (Fig. S1B), and a spiral trajectory (Fig. S1C). The results show that the HRFUM was able to follow all three trajectories with high fidelity. The matrix trajectory validates the controller's performance in handling sharp turns and discontinuities, while the pentagon trajectory demonstrates its capability in tracking non-convex paths. The spiral trajectory highlights robustness under continuously varying curvature and directional changes. In all cases, the actual end-effector path closely matched the predefined reference trajectory, confirming that the proposed controller ensures reliable performance when only the end-effector tracking task is considered. The significance of these simulations lies in establishing a performance baseline. By first validating trajectory tracking under ideal conditions, we can later quantify the effect of task priority control mechanism on tracking accuracy. This comparison not only highlights the necessity of the proposed priority-based control framework in more complex and constrained environments but also emphasizes the controller's inherent effectiveness in managing diverse trajectory types.

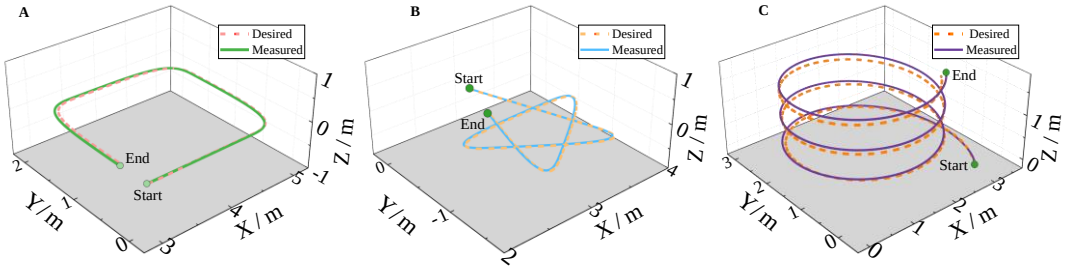


Fig. S1 Simulation results of end-effector trajectory tracking without task-priority control. (A) Matrix-shaped trajectory tracking. (B) Pentagon-shaped trajectory tracking. (C) Spiral trajectory tracking.