Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# Toward generic control for soft robotic systems

**Yu Sun**
Harbin Institute of Technology
**Yaosheng Deng**
Nanyang Technological University
**Wenjie Mei**
Nanjing University
**Xiaogang Xiong**
xiongxg@hit.edu.cn

Harbin Institute of Technology
**Yang Bai**
Hiroshima University
**Qiyi Wang**
Wuxi University
**Masaki Ogura**
Hiroshima University
**Zeyu Zhou**
Harbin Institute of Technology
**Mir Feroskhan**
Nanyang Technological University
**Michael Yu Wang**
Great Bay University
**Qiyang Zuo**
Chinese Academy of Sciences
**Yao Li**
Harbin Institute of Technology
**Yunjiang Lou**
Harbin Institute of Technology

---

**Article**

**Keywords:** soft robotics, generic control, motor intelligence, safety-critical control

**Posted Date:** September 15th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-10492371/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** No competing interests reported.

# Toward generic control for soft robotic systems

Yu Sun[1,5,†], Yaosheng Deng[2,†], Wenjie Mei[3,†], Xiaogang Xiong[1,*], Yang Bai[4,*]
Qiyi Wang[5], Masaki Ogura[4], Zeyu Zhou[1], Mir Feroskhan[2], Michael Yu Wang[6]
Qiyang Zuo[7], Yao Li[8] and Yunjiang Lou[1,*]

[1]Shenzhen Key Lab of Advanced Motion Control Technology and Modern Automation Equipment, School of Intelligence Science and Engineering, College of Artificial Intelligence, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, China
[2]School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798, Singapore
[3]School of Robotics and Automation, Suzhou Campus, Nanjing University, Suzhou 215163, China
[4]Graduate School of Advanced Science and Engineering, Hiroshima University, Higashi-Hiroshima 739-8527, Japan
[5]School of Cyber Science and Engineering, Wuxi University, Wuxi 214105, China
[6]School of Advanced Engineering, Great Bay University, Dongguan 523000, China
[7]Center for Precision Engineering, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China
[8]School of Robotics and Advanced Manufacture, College of Artificial Intelligence, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, China

[†]These authors contributed equally: Yu Sun, Yaosheng Deng and Wenjie Mei.
[*]Correspondence: xiongxg@hit.edu.cn; yangbai@hiroshima-u.ac.jp; louyj@hit.edu.cn

**Abstract**

Conventional robotics equates motor intelligence with the generation and execution of precise actions derived from system models. In high-dimensional and strongly uncertain systems such as soft robots, this precision-centric paradigm breaks down because it tightly binds intelligence to system-specific dynamics and control structures, preventing generalization across morphologies and tasks. Here we propose an alternative view in which intelligent behavior arises from regulating the boundaries of behavioral feasibility, allowing stable and adaptive behavior to emerge through environment interaction. Guided by this perspective, we introduce a general control framework that specifies high-level intentions to constrain unsafe or infeasible behaviors rather than prescribing detailed action commands, thereby shifting control from action generation to behavior regulation. Experiments across diverse soft robotic systems demonstrate robust and transferable motor behavior without reliance on specific models, pointing to a new paradigm for motor intelligence in complex systems.

**Keywords:** soft robotics; generic control; motor intelligence; safety-critical control

# 1 Introduction

In robotics and control, intelligent systems are commonly built on a simple underlying assumption: good control means designing a controller from a system model and generating desired actions through the precise execution of control commands[1–3]. This paradigm has been highly successful for rigid robots and mechanical systems whose dynamics can be accurately modeled, and it forms the foundation of classical control theory. However, this view relies on several strong assumptions, including structural stability, separable dynamics, and the ability to suppress execution errors through feedback[4]. When these assumptions no longer hold, equating intelligent control with action-level precision becomes questionable.

Soft robots systematically violate these assumptions[5–8]. Their highly compliant structures, strong nonlinearities, and continuous physical interaction with the environment lead to dynamics that are high-dimensional, time-varying, and difficult to predict. Under such conditions, accurate modeling is often impractical, and model-based controllers tend to be highly sensitive to modeling errors and environmental changes. As a result, control performance becomes fragile, tuning costs increase, and control strategies fail to generalize across different systems[9–13].

More importantly, the difficulty of controlling soft robots is not only a consequence of complex dynamics or imperfect models, but also reflects a deeper mismatch in how the control problem is formulated. In soft robotic systems, behavior is not determined by control commands alone. Instead, it emerges from the interaction between control inputs, body mechanics, and environmental dynamics[9,14,15]. In this setting, attempting to directly specify behavior through precise low-level action commands is neither realistic nor necessary. The primary role of control is not to generate a specific action, but to constrain what the system can and cannot do. Control shapes the range of feasible behaviors, rather than prescribing behavioral details.

This view of control is consistent with natural motor systems. Humans do not explicitly solve full-body dynamics or generate precise muscle-level commands during everyday movement[16]. Instead, the nervous system produces coarse motor intentions and intervenes mainly to suppress unsafe or infeasible behaviors through reflexes and sensory feedback. Stable and flexible behavior arises not from precise action execution, but from the continuous regulation of what behaviors remain feasible. This observation suggests that, in complex and uncertain soft systems, precision is not a prerequisite for intelligent motor control.

Motivated by this perspective, we propose a general control framework for soft robots that does not rely on explicitly designing detailed controllers or action-level control laws. Instead, the framework specifies high-level control intentions that directly shape the feasibility boundaries of system behavior, allowing concrete control strategies to emerge naturally through interaction between the robot and its environment. Because this approach does not depend on system-specific models or controller structures, its core mechanism can be shared across soft robotic platforms with different morphologies and actuation principles. Moreover, by exploiting the inherent compliance and adaptive physical interaction of soft robots, the resulting emergent control strategies often exhibit greater robustness and adaptability than traditionally designed controllers.

Within this framework, the control process is organized into three cooperating layers: generation of high-level action tendencies, regulation of infeasible behaviors through safety constraints, and translation of these tendencies into executable control inputs. The framework does not require precise system models, but instead learns approximate interaction structure and explores feasible behaviors through sampling and minimal intervention.

We validate the framework on multiple soft robotic platforms with substantially different structures and actuation mechanisms, including a soft manipulator, a bio-inspired soft fish, and a biohybrid cyborg insect system. Experimental results demonstrate that the proposed approach achieves safe, robust, and transferable control performance without relying on system-specific modeling assumptions. More importantly, these results point to a paradigm shift: in soft robotic systems, the goal of control is not to generate specific action commands, but to shape the conditions under which intelligent behavior can stably emerge.

# 2 Results

This section details the proposed generic control framework to achieve reliable and transferable control across soft robotic systems. The controller design is first introduced, followed by experimental validation on three distinct platforms: a tendon-driven soft arm, a soft-bodied robotic fish, and a cyborg cockroach.

## 2.1 Controller design

This study proposes a generic control framework for soft robots, addressing the long-standing lack of unified, safe, and generalizable control approaches in this field. The control framework is inspired by how humans control their bodies. Human motor control operates effectively in dynamic and uncertain environments without relying on precise dynamic models[16,17]. Instead, it is driven by three key mechanisms: coarse internal representations of body–environment interactions[16,18], rapid sampling-based action selection grounded in perception and experience[19], and reflexive safety regulation triggered near boundaries of safe motion[20,21]. Humans update their intended motion in a step-by-step manner rather than generating full trajectories in advance, and they maintain stability by steering away from unsafe regions rather than strictly converging toward preset targets. These observations highlight a crucial insight: in complex soft systems, greater model precision does not necessarily yield better control performance; fast, adaptive, and coarse-grained strategies could sometimes be more effective.

Motivated by this principle, we propose a generic soft robot control framework that unifies learning-based modeling, sampling-based planning, and adaptive safety regulation (Fig. 1). The learning module provides a coarse yet adaptable representation of robot–environment interactions; the sampling-based planner selects feasible actions in a stepwise, human-like manner; and the adaptive safety mechanism implements reflex-like protection when approaching unsafe regions. Working together, these components preserve the foundational intent of soft robotics and enable robust, fast, and generalizable control performance in complex and uncertain environments, achieving levels of safety, adaptability, and environmental interaction that rigid-body methods inherently struggle to deliver.

### 2.1.1 Learning-based model

Our learning-based model was based on the observation of how humans learn to walk (Fig. 1b). In early development, a child first learns to crawl, gradually stabilizing its center of mass and coordinating limb movements. Through crawling, the body acquires a sense of balance and learns to maintain stability even under continuous motion. When transitioning from crawling to kneeling and finally to standing, this prior experience enables the body to stabilize a higher and dynamically changing center of mass, eventually achieving stable walking.

This process mirrors the function of our Neural ODE network in the learning-based model. During training, the Neural ODE fits the boundary conditions and observed reference data of the soft robotic system, learning a physically consistent representation of its dynamics. At each point along the prediction time series $t_0, \ldots, t_N$, the model not only predicts the potential outcome of an action but also takes an exploratory step into previously unseen dynamic regions. Beyond $t_N$, it can further extrapolate outside the training distribution to nearby regions, showing that what the Neural ODE "remembers" is not mere data samples but the underlying physical consistency embedded in the learned dynamics.

### 2.1.2 Sampling-based planner

Through learning-based modeling, humans learn to walk toward a goal in an open environment. However, when the surroundings become crowded, the brain begins to imagine several short trajectories a few steps ahead, evaluating each imagined path and selecting the one that appears most reliable at that moment. After

taking the first step, the brain perceives how the surroundings have changed, imagines a few new paths, evaluates them, and moves again—continuously repeating this perception and decision cycle to make its way through complex spaces (Fig. 1c).

Similarly, the sampler (Fig. 1a) first explores a large number of candidate motion sequences through importance sampling, and each sequence is propagated through the Neural ODE model to obtain the corresponding motion prediction. The predicted trajectories are then evaluated by incorporating environmental information and weighted exponentially according to their cost functions. The control sequence with the optimal weighted cost is selected as the next-step command. Only the first action of this sequence is executed before the system updates its perception, generates new samples, and repeats the process, thereby enabling rapid and adaptive decision-making for soft robots in complex environments.

### 2.1.3 Adaptive safety filter

While the sampling-based controller enables fast and efficient decision-making, potential safety risks remain due to the coarse learning of the model and environmental variations. To address these uncertainties, we introduce a safety filter that rapidly evaluates the sampled control inputs and corrects unsafe ones before execution.

This design is inspired by the way parents accompany a child learning to walk. When the child moves safely along the path, the parents simply observe from behind without interference. However, as the child approaches the edge of a dangerous area, the parents immediately intervene, gently guiding the child back onto a safe path (Fig. 1d). Similarly, the adaptive CBF defines a provably safe region for the soft robot and converts the selected optimal prediction into stable and executable control inputs. The CBF remains passive when observing motion $f$ from the soft system inside the safety region. However, it becomes active instantly when the predicted motion $f$ approaches the safety boundary. Like parents quickly correcting a child's movement, CBF formulation provides a safe motion $f_{\text{safe}}$ to the optimizer, and then calculating an optimal $f_{\text{optimal}}$ to select the corresponding candidate control signal $u$ by reciprocal mapping, which ensures robust and stable behavior even under external perturbations. This mechanism resembles how the human nervous system triggers reflexive corrections when motion approaches safety limits, much like a parent catching a falling child.

### 2.1.4 Reciprocal mapping based cross-module coordination

It is important to note that these three modules cannot be simply connected in sequence, because none of them can directly generate executable control inputs. The learning-based dynamics model is a black box, making it difficult to construct control laws on top of it. The SBMPC module can explore feasible trajectories, but searching for control inputs through it is computationally expensive and unsuitable for real-time use. More critically, traditional CBF methods require explicit and accurate system dynamics to provide formal safety guarantees, a requirement fundamentally incompatible with a black-box model.

The clue to addressing this limitation also comes from human motor control. When performing a movement, humans do not compute individual muscle torques or joint angles. Instead, they first form a high-level movement tendency: a coarse motion sketch such as “extend the arm forward" or “shift the body to the right to maintain balance." This tendency is not the final motor command but an intermediate representation that lower-level systems refine and execute. Human motion, therefore, emerges through a gradual tendency-to-execution process rather than through direct low-level input design.

Our control architecture adopts this idea. The learning module first generates surrogate actions that express movement tendencies. All interactions between the planning and safety modules then take place on this surrogate-action level rather than the control-input level: the safety module filters the surrogate actions using adaptive CBF constraints to obtain a safe subset, and the sampling-based planner retrieves or

generates executable control inputs from this safe subset. In this structure, the learning module provides the feasible dynamic space (movement-tendency generation), the safety module defines its admissible region (boundary shaping), and the planning module maps tendencies to executable inputs (final execution). The three modules communicate through the surrogate action, and a reciprocal-mapping block establishes a bidirectional correspondence between surrogate actions and real control inputs, enabling coarse movement tendencies to be reliably converted into concrete motor commands. This mapping mechanism allows the three modules to coordinate through a shared intermediate representation, achieving generalized, safe, and transferable control across diverse soft robotic platforms.

## 2.2 Deployment on diverse soft robotic systems

To evaluate the generality and applicability of our proposed controller, we deploy it across three structurally and functionally distinct soft robotic platforms: a tendon-driven soft arm, a soft-bodied robotic fish, and a cyborg cockroach. These platforms differ significantly in morphology, dynamics, actuation methods, and operational environments, ranging from structured planar settings to unstructured aquatic and biological domains. The successful deployment across all three systems confirms that the controller accommodates diverse embodiments and maintains reliable, safety-aware performance across varying tasks and constraints.

### 2.2.1 Tendon-driven soft arm

We first evaluated the proposed generic controller on a tendon-driven soft arm to assess its performance on soft manipulators with elastic compliance and actuation redundancy. The arm is commanded to follow a square trajectory in a planar workspace while avoiding a fixed obstacle.

In the trajectory tracking task, as shown in Fig. 2b, the end-effector consistently follows the square path across 10 repeated trials. The tracking safety function (TSF), defined as the minimum distance to the region boundary, remains strictly positive throughout the trials, with a minimum value of $0.70$. These results indicate robust tracking performance under safety constraints.

In the obstacle avoidance experiment, as shown in Fig. 2c, the controller successfully guides the arm around a fixed cylindrical obstacle. The avoiding safety function (ASF), defined as the minimum distance between the arm and the obstacle, remains above $2.71$ across all trials. During the obstacle avoidance phase, ASF is prioritized to ensure collision-free execution, resulting in a temporary decrease in TSF. Once the obstacle is cleared, both safety functions return to stable and safe values.

These experiments validate that the controller achieves accurate, consistent, and safety-critical motion in tendon-driven soft arms, confirming its generalizability across soft robotic platforms with high degrees of compliance.

### 2.2.2 Soft-bodied robotic fish

To further evaluate the generality of our proposed controller, we deployed it on a soft-bodied robotic fish performing autonomous obstacle avoidance in a confined aquatic environment. The fish features a compliant and multi-segment tail actuated by a single servo motor, enabling undulatory locomotion. Two fixed spherical obstacles are placed in the water tank.

As shown in Fig. 3b, the robotic fish successfully completed ten consecutive trials under the proposed controller. A trajectory density plot reveals a consistent figure-eight pattern around the obstacles, indicating reliable repeatability and adaptive maneuvering capability. Time-stamped snapshots from a representative trial (Fig. 3b-iii) further illustrate smooth and collision-free navigation.

The ability to turn (i.e., change heading direction) is achieved by modulating the bias of the input signal, defined as the angle between the fish body and the first tail segment (Fig. 3c). Varying this bias enables left

and right turns based on the planned motion. Experimental results confirm that stable directional control is achieved with minimal input modulation.

To assess safety, we tracked the Euclidean distance between the fish and each obstacle throughout the motion. In a representative trial (Fig. 3d), the minimum distances to the two obstacles were $0.33\ \mathrm{m}$ and $0.16\ \mathrm{m}$, respectively, both comfortably exceeding the predefined safety threshold of $0.10$ m. Boxplots summarizing the minimum distances across all ten trials (Fig. 3d-iii) further confirm that distances consistently remained well within the safe range, underscoring robust and reliable obstacle avoidance in a cluttered aquatic environment.

These results demonstrate that the proposed generic controller enables autonomous, safe, and highly repeatable operation across morphologically distinct soft robotic platforms, including the soft robotic arm and the fish-like robot in constrained environments.

### 2.2.3 Cyborg cockroach

To evaluate the applicability of our generic control framework to soft bio-hybrid platforms, we conducted straight-line locomotion experiments using a cyborg cockroach. The system consists of a live Madagascar hissing cockroach equipped with a custom electronic backpack that delivers directional electrical stimulation to induce turning. The goal is to guide the cockroach to a predefined target while keeping its trajectory within a narrow safety corridor.

As shown in Fig. 4a, the platform includes a lightweight wireless stimulation module mounted on the cockroach (Fig. 4a-i). The task environment defines a straight path with a lateral safety margin of $\pm 0.05\ \mathrm{m}$ (Fig. 4a-ii). The task is considered successful if the trajectory remains entirely within this corridor after the cockroach arrived at its predefined target. Fig. 4b presents a representative trial under our control strategy. The top-view trajectory (orange-red) shows that the cockroach stays within the safety region throughout. Stimulation events for left and right turns are shown as purple and cyan dots, triggered only when the trajectory approaches the corridor boundaries. Rear-view snapshots illustrate key moments, including straight walking, initial turning, and subsequent corrections. These results confirm that the controller achieves reliable navigation with less interventions.

We further compare our proposed method with a continuous stimulation strategy in which electrical stimuli are persistently applied whenever the heading deviates from the desired direction. As shown in Fig. 4c-i, under our controller, most trajectories remain within the designated safety region even as the number of stimulations increases. In contrast, the continuous stimulation controller frequently produces trajectories that violate the safety boundaries. The boxplot in Fig. 4c-ii shows the distribution of $|Y|$ deviations under both methods. Our approach keeps lateral deviations consistently bounded across eight trials, whereas the continuous strategy results in significantly larger deviations. Fig. 4c-iii summarizes the relationship between the number of stimulations and the safety ratio, which is defined as the proportion of the trajectory that remains within the safety corridor. Under continuous stimulation, the safety ratio decreases rapidly as the number of stimulations increases. This effect is primarily due to the cockroach's habituation to repeated stimuli. In comparison, the proposed method maintains a high safety ratio with fewer interventions even when the cumulative stimulation count exceeds 120.

These results demonstrate that the proposed generic controller is effective and robust when deployed on soft bio-hybrid platforms. By triggering interventions only when necessary for safety, the method achieves reliable task performance with fewer stimulations, reducing the risk of overstimulation and slowing down habituation to stimulation in living agents.

## 2.3 Robustness and adaptability in long-term and dynamic settings

To further validate the generality and practicality of the proposed framework, we conducted comparative experiments covering long-term operation, physical constraints, and dynamic environments. The results demonstrate that the framework exhibits remarkable adaptability, stability, and real-time efficiency when facing challenges such as model degradation and input limitations, highlighting its advantages in soft robotic control.

### 2.3.1 Long-term performance maintenance

A significant challenge in deploying soft robots is their performance degradation over extended periods. This occurs because factors like material fatigue cause the robot's dynamics to change, making initial models inaccurate. We demonstrate this problem and our solution in a long-term experiment with a soft manipulator over 3000 operational cycles (Fig. 5a). Initially, the controller accurately tracks the desired trajectory (Fig. 5a-ii, red line). After 3000 cycles, the performance of the controller without our adaptive CBF module degrades substantially, causing the manipulator to violate its predefined safe region (Fig. 5a-i, 5a-iii, cyan line). In contrast, by enabling the adaptive CBF, our framework compensates for the model degradation and keeps the manipulator within the safe region, successfully maintaining its initial performance (Fig. 5a-i, 5a-iv, purple line).

The key to this sustained performance is our method's different approach to handling model uncertainty. Instead of pursuing an ever-more-perfect, but ultimately brittle, dynamic model, our framework is designed to operate effectively with an imperfect model by directly accounting for uncertainty and prediction errors. The framework has two main components. First, the Neural ODE module learns the robot's core dynamics from data to create a predictive model. Second, the CBF module provides a safety layer. It continuously measures the error between the model's predictions and the robot's actual behavior, and uses this information to update the safety constraints in real-time. This mechanism allows the controller to compensate for the gradual degradation of the learned model and enables reliable, long-duration autonomy for soft robots in unstructured environments.

### 2.3.2 Safe control under physical constraints

To operate safely in the real world, soft robots must respect physical constraints, such as actuator input limits, to prevent hardware damage. This is directly related to the need for smooth control inputs, as sudden changes can cause destructive internal stresses. Integrating these constraints is a major challenge for traditional controllers. We demonstrate this in a navigation task where a robotic fish must execute a tight turn in a confined space (Fig. 5b). As shown in the experiment, a PID controller with a simple input saturation limit is unable to complete the maneuver, causing the fish to fail the turn (Fig. 5b-i, 5b-iii, red line). In contrast, our framework successfully navigates the tight turn, keeping the fish on the desired path while respecting the same input limits (Fig. 5b-i, 5b-ii, cyan line).

This successful performance is achieved through the synergy between the optimizer and the safety module in our framework. A traditional PID controller performs poorly when its input is limited due to issues like integral windup. Our method works differently: the SBMPC module generates a diverse set of possible control actions. The adaptive CBF module then acts like a filter, ensuring that the final selected control input is not only effective for the task but also smooth and compliant with all physical constraints.

### 2.3.3 Real-time control in dynamic environments

Another feature of our framework is its computational efficiency, which enables real-time control of soft robots in dynamic environments. We demonstrated this in a dynamic target-tracking experiment, where

a robotic fish was tasked to follow a moving target with fluctuating speeds up to 0.32 m/s (Fig. 5c). The results show that the fish's speed closely matched the target's speed in real-time (Fig. 5c-i), and the controller successfully maintained a close tracking distance throughout the experiment (Fig. 5c-ii). The experimental snapshots further visualize this successful tracking performance (Fig. 5c-iii).

This real-time capability is due to the framework's sampling-based control architecture. Unlike conventional MPC, which must solve a computationally expensive optimization problem at every time step, our SBMPC module explores the solution space efficiently by sampling a limited number of candidate trajectories. This reduces the per-step computation time, making real-time implementation feasible. More importantly, this structure is inherently parallelizable: each sampled trajectory can be evaluated independently, allowing computation to be distributed across multiple cores. This means that the framework can scale gracefully: future scenarios that require increased sampling density or more complex evaluations can be handled without a prohibitive increase in latency.

## 3 Discussion

This study proposed a generic control framework for soft robotic systems, addressing a fundamental gap in soft-robot control: the lack of a unified control principle that enables generalization beyond morphology- and actuation-specific designs.

The ability of the proposed framework to generalize across morphologies stems not only from new technical design, but from a more fundamental philosophical departure from conventional soft-robot control paradigms. Existing approaches largely inherit rigid-body control assumptions, emphasizing accurate modeling and fine-grained actuation tightly coupled to specific morphologies and actuation layouts. For soft robotic systems, however, such precision is neither feasible nor necessary, as their behavior is largely shaped by compliant dynamics and environment-mediated self-regulation. By formulating control at the level of high-level tendencies rather than low-level commands, the proposed framework decouples control logic from specific morphologies and actuation layouts, thereby providing a principled explanation for its consistent performance across heterogeneous systems.

To evaluate the feasibility and generality of the proposed framework, we conducted systematic validations on multiple soft robotic systems with fundamentally different characteristics, spanning diverse morphologies and actuation mechanisms. The experimental results demonstrated that the framework was not tailored to a single platform or structure, but instead exhibited consistent control behavior across heterogeneous systems, confirming its cross-platform applicability. Crucially, this generality does not compromise performance. Instead, the proposed generic control framework surpasses specialized baselines in multiple key aspects, such as long-term robustness, effective constraint handling, and real-time computational efficiency.

Despite these results, it was not feasible to exhaustively evaluate all possible soft robotic morphologies and application scenarios, given the diversity in structure, actuation, and task requirements. Future work will extend the proposed framework to a broader range of systems and more complex tasks, while iteratively refining controller design and modeling strategies based on system-level performance. Nevertheless, the proposed generic control framework has been validated on multiple soft robotic systems with substantially different morphologies and actuation mechanisms, covering the major classes studied in soft robotics. As a result, the observed behaviors can be interpreted as control properties with cross-system relevance, rather than as isolated outcomes tied to a single platform.

More broadly, the contribution of this work extends beyond the introduction of a generic control framework. The results suggest that soft robots do not necessarily require highly customized control designs for each individual system; instead, explicitly leveraging their inherent compliance can enable control strategies with stronger generalization capability. This perspective may help advance soft robotics from system-

specific engineering solutions toward a more unified and transferable soft-robotic science.

# 4 Methods

## 4.1 Control framework design

### 4.1.1 Learning-based model

To construct this model, we follow a three-stage learning procedure. First, we collect a set of input–output trajectories from the physical system under diverse actuation patterns and environmental conditions, forming a representative dataset that reflects the robot's deformation responses. Second, we embed a neural network inside the ODE formulation to parameterize the unknown dynamics as

$$\dot{x}(t) = f(x(t), u(t)) + \theta(t), \tag{1}$$

where state $x \in \mathcal{X} \subset \mathbb{R}^n$, input $u \in \mathbb{R}^m$. $f\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is a Lipschitz function. $\theta \in \Theta \subset \mathbb{R}^n$ represents the external uncertainty. Here, $f$ represents the desired internal dynamics of the system, which correspond to the underlying physical model without considering external disturbances $\theta(t)$.

After collecting the training data, we trained the Neural ODE parameters by minimizing the trajectory-level discrepancy between predicted and measured motion. Then, the trained Neural ODE approximates the underlying system dynamics and predicts the robot's motion response under arbitrary control inputs.

Since the Neural ODE learns only an approximate representation of the system, modeling uncertainty remains inevitable. To quantify this uncertainty and enable safe integration into control, we characterize an explicit upper bound on the modeling error. Let $f(x(t), u(t))$ denote the Neural ODE prediction and $f_{\text{des}}(t)$ a reference trajectory representing the desired system behavior. The prediction error $\varepsilon(t) = f(x(t), u(t)) - f_{\text{des}}(t)$ is bounded by a calibrated constant $\delta_{\max} \geq 0$, such that $|\delta(t)| \leq \delta_{\max}$ holds within the range of states and inputs observed in the data[22]. Hence, the learned Neural ODE is equipped with a certified upper bound on its prediction error, quantifying the worst-case discrepancy between the model and the true dynamics.

With the error bound established, the model becomes sufficiently trustworthy for control despite not being an exact physical description. This reflects the intended coarse-modeling philosophy of our Neural ODE design, where the goal is not to replicate the full analytical physics of the soft robot. Instead, it captures the essential nonlinear trends and physical regularities embedded in the observed data, providing a coarse but coherent description of robot–environment interactions.

### 4.1.2 Sampling-based planner

Building upon the learned Neural ODE dynamics, the SBMPC module makes real-time decisions by exploring multiple candidate control sequences to the Neural ODE and evaluating their predicted outcomes. The formal description of these stages is presented below.

First, at time step $t_k$, the sampler operates within a finite prediction horizon of length $H$. A set of candidate control sequences $U = \{u_1, \ldots, u_n\}$ is drawn from a parameterized probability distribution and propagated through the Neural ODE to generate the corresponding predicted motion sequences $F = \{f_1, \ldots, f_n\}$, enabling the controller to rapidly explore feasible short-horizon motions without relying on explicit analytical models. Then, the SBMPC evaluates each predicted sequence $F$ by optimizing a composite cost that reflects both motion accuracy and safety:

$$J(F, U) = \sum_{k=1}^{H} \ell(x_k, u_k) + \lambda(x_k), \tag{2}$$

where $x_k$ and $u_k$ denote the state and candidate control at time step $k$, $\ell(x_k, u_k)$ measures the tracking error with respect to the desired motion, and $\lambda(x_k)$ balances optimization performance and penalties.

Among all candidates, the prediction $f_i$ that minimizes $J(F, U)$ in (2) is selected as the provisional optimal prediction $f_{\text{optimal}}$, i.e., $f_{\text{optimal}} = \arg\min_{f_i \in F} J(F, U)$. This optimal prediction serves as a data-driven estimate of the desired next-step motion, representing the planner's best prediction under the learning-based model. This provisional prediction $f_{\text{optimal}}$ is then provided to the adaptive safety filter, which evaluates and refines it to ensure consistency with the system's safety constraints.

#### 4.1.3 Adaptive safety filter

The sampling-based planner provides optimal prediction motion $f_{\text{optimal}}$, which contains approximation errors from model uncertainty. To ensure safety under these uncertainties, we introduce an adaptive CBF module as a reflex-like safety filter. This module evaluates whether the provided optimal prediction $f_{\text{optimal}}$ is within an admissible safety region and intervenes only when necessary.

To formalize this safety region, we design $\mathcal{C} \subset \mathbb{R}^n$ to dynamically adjust its boundary according to model uncertainty. The safety condition is defined by a CBF $h(x)$, which characterizes the admissible set of system states satisfying $h(x) \geq 0$. Based on this formulation, the adaptive safety region is constructed as

$$\mathcal{C} = \{x_k \mid \Phi + \Pi \geq 0\}, \tag{3}$$

where $\Pi = f_{\text{optimal}}^{\top} \dfrac{\partial h}{\partial x_k} \left\| \dfrac{\partial h}{\partial x_k} \right\| \bar{\varepsilon}$ is a control-related term that reflects the effect of the prediction $f_{\text{optimal}}$ on safety set (3), $\Phi = \left( \hat{\theta}_k + \dfrac{\partial h}{\partial \hat{\theta}_k} \right)^{\top} \dfrac{\partial h}{\partial x_k} + h$ is an adaptive term that compensates for model uncertainty, and the adaptive term $\hat{\theta}$ is updated by $\dot{\hat{\theta}} = -\partial h / \partial x$. The CBF $h(x)$ in $\Pi$ and $\Phi$ defines the admissible state set that satisfies $h(x) \geq 0$.

With the safety region $\mathcal{C}$ defined, the adaptive CBF evaluates whether the planner's provisional prediction is admissible. If $f_{\text{optimal}} \in \mathcal{C}$ the motion is considered safe and is passed directly to the execution layer. However, if $f_{\text{optimal}} \notin \mathcal{C}$, the safety filter intervenes in a reflex-like manner. In this case, the CBF formulation first extracts all predicted motions that are within the safety region, forming the set $\mathcal{S} = \mathcal{C} \cap \{f_1, \ldots, f_n\}$. Then, among these admissible candidates, the safety filter selects a prediction $f_{\text{safe}}$ that is closest to the provisional optimal one $f_{\text{optimal}}$, i.e.,

$$f_{\text{safe}} = \arg\min_{f_i \in \mathcal{S}} \|f_i - f_{\text{optimal}}\|. \tag{4}$$

This selected $f_{\text{safe}}$ is then used in the subsequent reciprocal mapping process to determine the corresponding executable control $u_{\text{safe}}$, which ensures the executed command remains consistent with both the learned dynamics and the safety constraints.

#### 4.1.4 Reciprocal mapping based cross-module coordination

Once the adaptive safety filter provides the verified safety prediction $f_{\text{safe}}$, the reciprocal mapping searches within the candidate control sequence $\{u_1, \ldots, u_n\}$ provided by the sampler to find the corresponding control input $u_{\text{safe}}$.

This mapping explicitly encodes the correspondence between the predicted motion sequence $\{f_1, \ldots, f_n\}$ and the candidate control sequence $\{u_1, \ldots, u_n\}$ established by the sampler and Neural ODE. It serves as a bridge that allows the system to retrieve the control sequence associated with a given safe prediction. Since $f_{\text{safe}}$ is selected by the adaptive safety filter from the predicted motion sequence

$\{f_1, \ldots, f_n\}$, it follows that $f_{\text{safe}} \in \{f_1, \ldots, f_n\}$ and there exists a one-to-one correspondence mapping between the two sets $\{f_1, \ldots, f_n\}$ and $\{u_1, \ldots, u_n\}$. Therefore, the system can retrieve the corresponding control sequence $u_{\text{safe}}$ associated with the safety-verified motion $f_{\text{safe}}$ through the reciprocal mapping. Formally, we define the reciprocal mapping as

$$u_{\text{safe}} = \mathcal{M}(f_{\text{safe}}), \quad f_{\text{safe}} \in \{f_1, \ldots, f_n\}, \quad u_{\text{safe}} \in \{u_1, \ldots, u_n\}, \tag{5}$$

where $\mathcal{M}(\cdot)$ denotes a bidirectional mapping function constructed directly from the one-to-one pairs $(u_i, f_i)$ generated by the sampler and Neural ODE forward propagation.

This reciprocal mapping preserves the internal consistency between the learned dynamics and control inputs, ensuring that safety-verified prediction can be transferred into executable control input without additional computation.

## 4.2 Robotic systems

### 4.2.1 Tendon-driven soft arm

The soft arm shown in Fig. 2 was fabricated as a single monolithic structure via 3D printing with thermoplastic polyurethane (TPU), which confers high axial compliance while preserving sufficient torsional stiffness for precise control. The arm measures 300 mm in length and 48 mm in diameter, thereby optimizing the balance between structural integrity and flexibility. Four tendons are symmetrically routed through the arm and anchored at the distal tip, enabling omnidirectional bending in three-dimensional space. Each tendon is independently actuated by a NEMA 17 stepper motor via a low-backlash capstan-drive mechanism. The end-effector bears a visual fiducial marker that is tracked in real time by an external Intel RealSense D435i depth camera, yielding 3D pose feedback for performance evaluation and closed-loop control.

### 4.2.2 Soft-bodied robotic fish

The soft robotic fish depicted in Fig. 3 incorporates an undulatory propulsion system actuated by a flexible caudal tail. The robot measures approximately 450 mm $\times$ 90 mm $\times$ 94 mm and weighs 0.88 kg. The rigid head houses all onboard electronics, including a wireless communication module (APC220), an inertial measurement unit (JY901B), a 32-bit microcontroller (STM32F407ZET6), and a 14.8 V lithium-polymer battery.

The caudal tail comprises two serially connected passive joints fabricated from spring steel, which facilitate smooth and continuous bending under actuation. A single waterproof servo motor (GDW-BLS896) imparts periodic oscillations at the tail base, generating a propagating traveling wave that yields forward thrust. The body and tail were molded from soft silicone-based elastomers via a multi-part casting process, ensuring watertightness and compliant deformation under hydrodynamic loads.

This tail-driven propulsion system endows the fish robot with exceptional maneuverability and seamless navigation within confined aquatic environments, such as narrow corridors and bottleneck spaces.

### 4.2.3 Cyborg cockroach

**Experimental species and ethics.** The adult *Gromphadorhina portentosa* shown in Fig. 4 were housed at approximately 25 °C and 60% relative humidity, with food and water provided *ad libitum*. All experimental procedures were approved by the Institutional Animal Care and Use Committee (IACUC) of Harbin Institute of Technology, Shenzhen (approval ID: IACUC-2020026). Prior to surgery, animals were briefly anesthetized with $CO_2$ for approximately one minute. Platinum microelectrodes (bare diameter: 76.2 µm; insulated: 139.7 µm) were implanted bilaterally into the optic lobes at a depth of 2 mm and secured using biocompatible adhesive.

**Electronic backpack and stimulation system.** A lightweight backpack was mounted on the insect's thorax, shown in Fig. 4a, integrating power supply, wireless control, and stimulation modules. The system includes a Wi-Fi-enabled microcontroller, a DC–DC voltage regulator, and a constant-current stimulator (Intan V063) communicating via SPI. The stimulator supports both unidirectional (0–2550 µA) and bidirectional (0–5100 µA) modes, with resolution of 1 µA below 255 µA and 10 µA above. Stimulation protocols included forward-drive pulses (50 Hz, 40–160 µA) and turning pulses (50 Hz, 80–110 µA), applied to either the abdomen or optic lobe depending on the task. The total backpack weight, including battery, is approximately 7.1 g.

## Data availability

All study data needed to evaluate the conclusions are included in the Article and the Supplementary Information. Additional raw data are available from the corresponding authors upon reasonable request.

## Code availability

The custom code used for controller implementation and analysis is available from the corresponding authors upon reasonable request.

## Acknowledgements

This work was partially supported by the National Natural Science Foundation of China under grant no. 62573159, the Guangdong Provincial Key Laboratory of Intelligent Morphing Mechanisms and Adaptive Robotics under grant no. 2023B1212010005, and the Shenzhen Science and Technology Program under grant nos. KJZD20240903100802004 and GXWD 20231130153844002.

## Author contributions

Y.B. and X.X. conceived and designed the research. Y.B., Y.D. and Y.S. developed the control framework. Y.S., Z.Z., Q.Z. and Y.L. built the experimental platform and robots. Y.S. and Z.Z. conducted the experiments and collected the raw data. Y.S., Z.Z., Y.B., Y.D., X.X., Q.W. and W.M. conducted the data analysis. Y.B., Y.D., Y.S., Q.W., X.X., W.M., Y.J.L., M.Y.W., M.F. and M.O. contributed to the preparation of the manuscript. X.X., Y.B. and Y.J.L. supervised the research.

## Competing interests

The authors declare no competing interests.

## References

[1] Federico Renda, Frédéric Boyer, Jorge Dias, and Lakmal Seneviratne. Discrete cosserat approach for multisection soft manipulator dynamics. *IEEE Transactions on Robotics*, 34(6):1518–1533, 2018. doi: 10.1109/TRO.2018.2868815.

[2] Olivier Goury and Christian Duriez. Fast, generic, and reliable control and simulation of soft robots using model order reduction. *IEEE Transactions on Robotics*, 34(6):1565–1576, 2018. doi: 10.1109/TRO.2018.2861900.

[3] Weicheng Huang, Xiaonan Huang, Carmel Majidi, and M. Khalid Jawed. Dynamic simulation of articulated soft robots. *Nature Communications*, 11:2233, 2020. doi: 10.1038/s41467-020-15651-9.

[4] Takahiro Miki, Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62): eabk2822, 2022. doi: 10.1126/scirobotics.abk2822.

[5] O. Goury and C. Duriez. Fast, generic, and reliable control and simulation of soft robots using model order reduction. *IEEE Transactions on Robotics*, 34(6):1565–1576, 2018. doi: 10.1109/TRO.2018.2861900.

[6] Wenzhong Huang, Xiaotian Huang, Carmel Majidi, and Mohammed K Jawed. Dynamic simulation of articulated soft robots. *Nature Communications*, 11(1):2872, 2020. doi: 10.1038/s41467-020-15651-9.

[7] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi. Model-based reinforcement learning for closed-loop dynamic control of soft robotic manipulators. *IEEE Transactions on Robotics*, 35(1):127–134, 2019. doi: 10.1109/TRO.2018.2878318.

[8] Yaosheng Deng, Masaki Ogura, Yang Bai, Yujie Wang, and Mir Feroskhan. Safety-critical control for nonlinear systems with complex input constraints. *IEEE Transactions on Automatic Control*, 70(10): 7016–7023, 2025. doi: 10.1109/TAC.2025.3570826.

[9] Daniela Rus and Michael T. Tolley. Design, fabrication and control of soft robots. *Nature*, 521(7553): 467–475, 2015. doi: 10.1038/nature14543.

[10] Cecilia Laschi, Barbara Mazzolai, and Matteo Cianchetti. Soft robotics: Technologies and systems pushing the boundaries of robot abilities. *Science Robotics*, 1(1):eaah3690, 2016. doi: 10.1126/scirobotics.aah3690.

[11] Robert K. Katzschmann, Joseph DelPreto, Robert MacCurdy, and Daniela Rus. Exploration of underwater life with an acoustically controlled soft robotic fish. *Science Robotics*, 3(16):eaar3449, 2018. doi: 10.1126/scirobotics.aar3449.

[12] Tiefeng Li, Gurui Li, Yiming Liang, T Cheng, J Dai, X Yang, B Liu, Z Zeng, Z Huang, Y Luo, et al. Fast-moving soft electronic fish. *Science Advances*, 3(4):e1602045, 2017.

[13] David Howard, Agoston E Eiben, Dunstan F Kennedy, Jean-Baptiste Mouret, Philip Valencia, and David Winkler. Evolving embodied intelligence from materials to machines. *Nature Machine Intelligence*, 1(1):12–19, 2019. doi: 10.1038/s42256-018-0009-2.

[14] Thomas George Thuruthel, Yasmin Ansari, Egidio Falotico, and Cecilia Laschi. Control strategies for soft robotic manipulators: A survey. *Soft Robotics*, 5(2):149–163, 2018. doi: 10.1089/soro.2017.0007.

[15] Benjamin Shih, Dylan Shah, Chung-Wen Li, and et al. Electronic skins and machine learning for intelligent soft robots. *Science Robotics*, 5(41):eaaz9239, 2020. doi: 10.1126/scirobotics.aaz9239.

[16] Daniel M. Wolpert, Zoubin Ghahramani, and Michael I. Jordan. An internal model for sensorimotor integration. *Science*, 269(5232):1880–1882, 1995. doi: 10.1126/science.7569931.

[17] Emanuel Todorov and Michael I. Jordan. Optimal feedback control as a theory of motor coordination. *Nature Neuroscience*, 5(11):1226–1235, 2002. doi: 10.1038/nn963.

[18] Konrad P. Körding and Daniel M. Wolpert. Bayesian integration in sensorimotor learning. *Nature*, 427 (6971):244–247, 2004. doi: 10.1038/nature02169.

[19] Stephen H. Scott. Optimal feedback control and the neural basis of volitional motor control. *Nature Reviews Neuroscience*, 5(7):532–545, 2004. doi: 10.1038/nrn1427.

[20] Kathleen E. Cullen. Vestibular processing during natural self-motion: Implications for perception and motor control. *Nature Reviews Neuroscience*, 20(6):346–363, 2019. doi: 10.1038/s41583-019-0153-1.

[21] Mohsen Jamali, Donald E. Mitchell, Amanda Dale, Jerome Carriot, Soroush G. Sadeghi, and Kathleen E. Cullen. Self-motion evokes precise spike timing in the primate vestibular system. *Nature Communications*, 7:13229, 2016. doi: 10.1038/ncomms13229.

[22] Wenjie Mei, Dongzhe Zheng, and Shihua Li. Controlsynth neural ODEs: Modeling dynamical systems with guaranteed convergence. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

# Figure legends

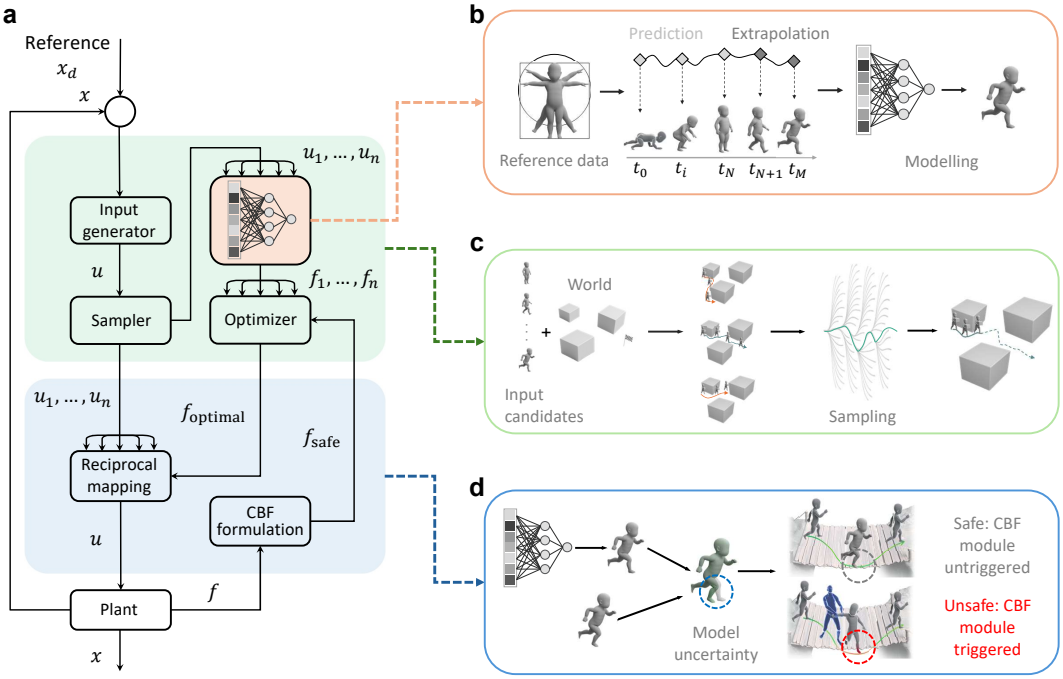


**Fig. 1. Overview of the proposed control framework inspired by human motor intelligence.** **a** Overall architecture. The control framework consists of three core modules: a learning-based model, a sampling-based planner and an adaptive safety filter. The learning-based model employs a neural ordinary differential equation (Neural ODE) to approximate the intrinsic robot–environment dynamics; the sampling-based planner generates candidate control sequences via a sampling-based model predictive control (SBMPC) formulation and predicts corresponding outputs; and the adaptive safety filter, grounded on a control barrier function (CBF), selects the optimal predicted motion within the provably safe region and maps it back to real-time control signals through a reciprocal mapping. This structure mirrors the human motor control process (coarse modeling, rapid planning and reflex-like safety regulation) to maintain both flexibility and stability under uncertainty. **b** Learning-based model. The Neural ODE captures dynamics of the soft robot and provides bounded prediction errors for motion forecasting, similar to how the human brain forms and refines internal representations of body–environment interactions through perception and experience. **c** Sampling-based planner. The SBMPC-based formulation explores and optimizes candidate motion sequences using importance sampling, achieving efficient planning of soft robotic motion, much as the human nervous system performs rapid sample-based motion planning in complicated environments. **d** Adaptive safety filter. The adaptive CBF defines a provably safe region for the soft robot and converts the selected optimal prediction into stable, executable control inputs, ensuring robust and safe behavior under external perturbations.

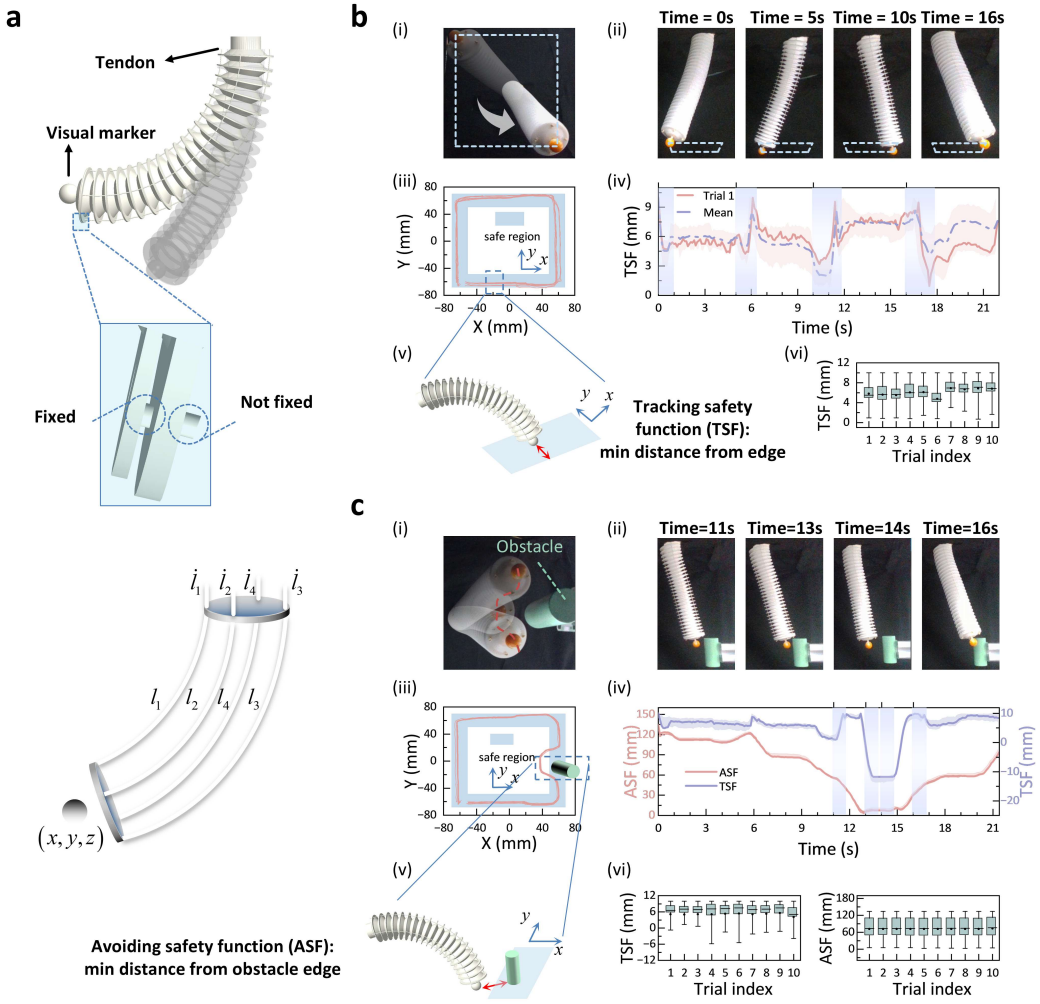


**Fig. 2. Experimental validation on the soft robotic arm. The experiments include trajectory tracking and obstacle avoidance of the arm's end-effector.** **a** Structure and actuation of the tendon-driven soft arm. **b** Square trajectory tracking with safety constraints. Top and side views are shown in (i) and (ii). The end-effector follows a square path within the defined safety margin (blue band) (iii). (iv) shows the temporal evolution of the tracking safety function (TSF), defined as the minimum distance to the boundary (see (v)). It indicates safe operation when within 0–10. As shown in (iv), TSF remains stable over time, with slight dips at corners but no violations. Boxplots in (vi) show consistent TSF distributions across ten trials, with all trajectories remaining within the safe region. **c** Obstacle-avoidance performance. Top and side views in (i) and (ii) show the arm adapting its motion to avoid a green obstacle, with corresponding trajectories in (iii). The temporal evolution of TSF and the avoidance safety function (ASF) is shown in (iv), where ASF is defined as the minimum distance to the obstacle (see (v)). It remains positive throughout, ensuring safe obstacle avoidance, when TSF is relaxed to satisfy ASF constraints. Boxplots in (vi) show stable TSF and ASF distributions, confirming the controller's consistency and reliability.

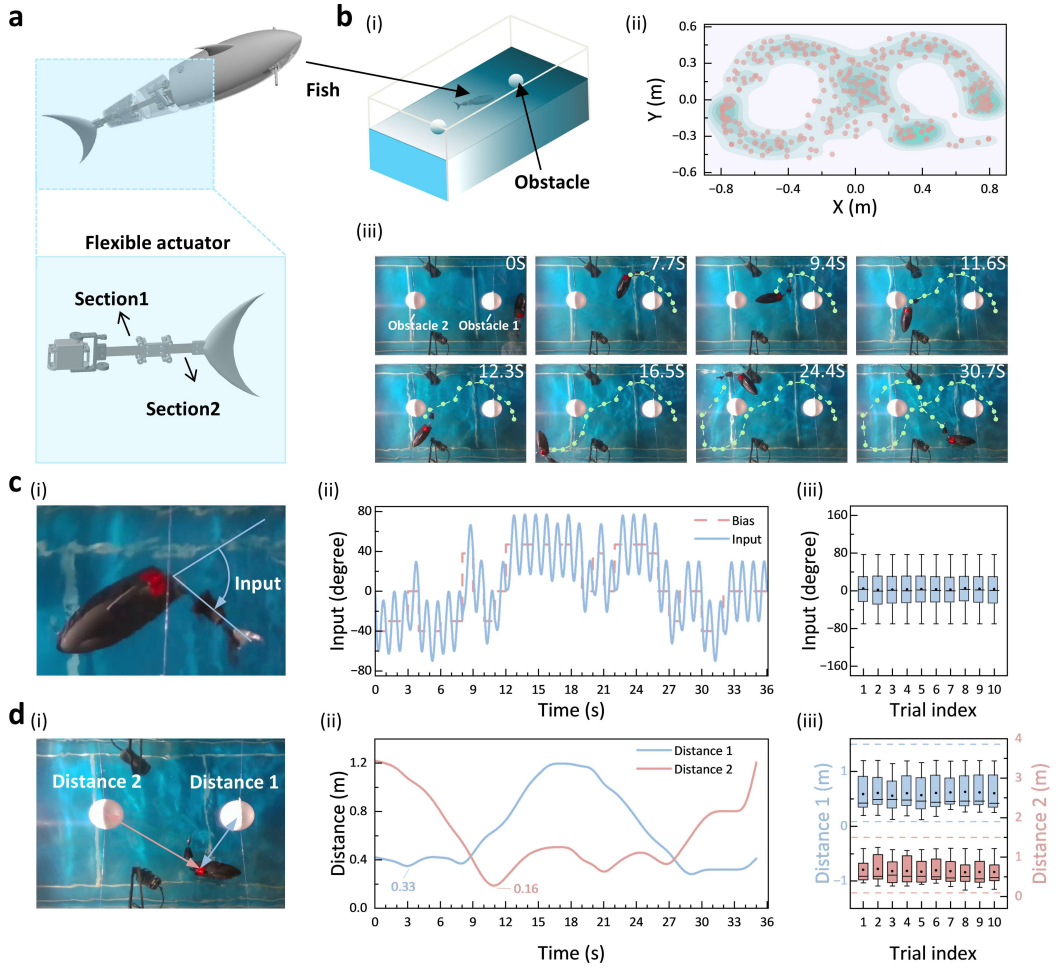


**Fig. 3. Experimental validation on a soft-bodied robotic fish navigating a confined aquatic environment.** **a** Structure of the soft-bodied robotic fish propelled by a multi-segment flexible tail. **b** Navigation in an obstacle-filled environment. The setup with two obstacles is shown in (i). The robot followed a figure-eight trajectory over ten trials, as seen in (ii), successfully avoiding collisions. Snapshots in (iii) illustrated smooth swimming of a representative experimental trial. **c** Turning control through input modulation. The control input is defined as the angle between the body and first tail segment (i). A sinusoidal signal governs tail motion, and turning is achieved by modulating its bias; the input trajectory for trial 1 is shown in (ii). Boxplots in (iii) confirm that the required tail oscillation amplitudes remain below the designed safety limit. **d** Evaluation of obstacle distances. Distances to obstacle 1 (blue) and 2 (red) are defined in (i). In trial 1 (ii), both remain above the safety threshold with minima of 0.33 m and 0.16 m. Boxplots in (iii) confirm safety over ten trials.

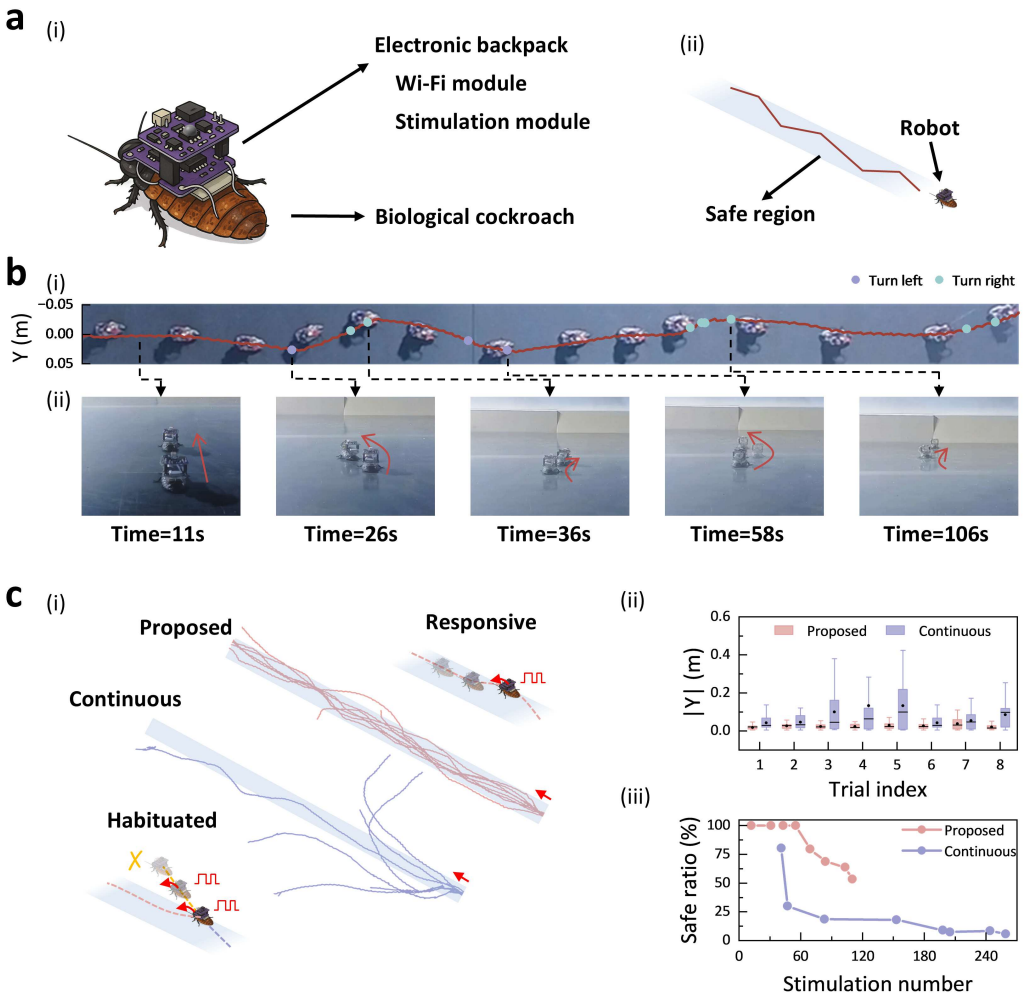


**Fig. 4. Experimental validation on a cyborg cockroach performing straight-line locomotion.** **a** Structure and task setup of the cyborg cockroach. The composition of the cyborg cockroach is shown in (i). The cockroach follows a straight-line path within the defined safety margin (blue band) (ii). **b** Straight-line locomotion with safety constraints. Top and rear views are shown in (i) and (ii), with stimulation events labeled for left turns (purple) and right turns (cyan). The cockroach remains within the safety region throughout. **c** Comparison with continuous stimulation strategy. (i) shows trajectories under both algorithms. With the proposed event-triggered strategy, most trajectories stay within the safety region (blue band). Under continuous stimulation, trajectories often exceed this region due to habituation. Boxplots in (ii) show the $|Y|$ distribution under both algorithms. The proposed method yields a stable distribution with lower variance and a smaller mean. As the number of stimulations increases, habituation emerges in both cases, leading to a decline in safety ratio, as shown in (iii).

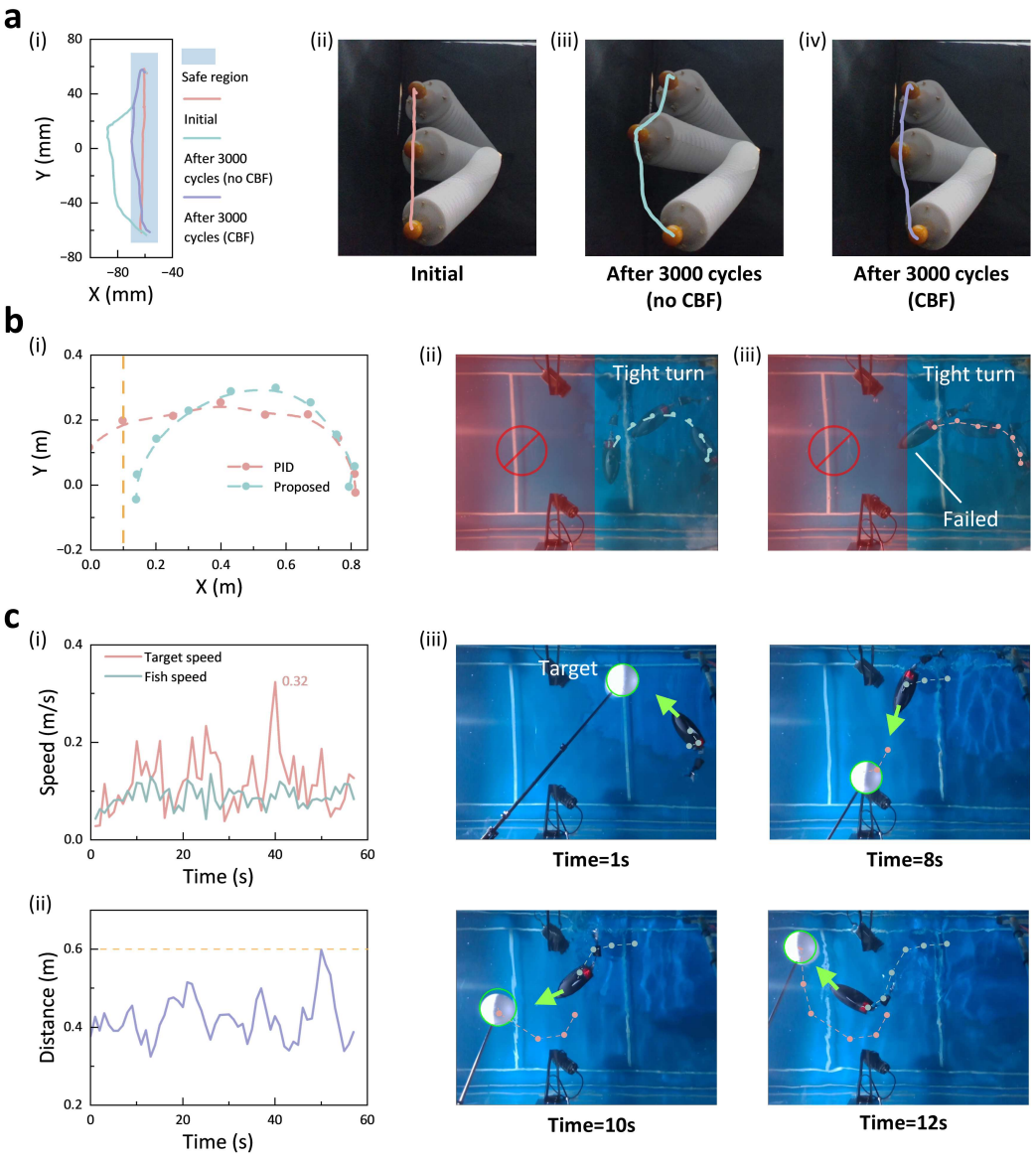


**Fig. 5. Experimental validation of the framework's robustness, safety and real-time performance.** **a** Robustness against model degradation in long-term operation. The purpose was to test our algorithm's ability to maintain performance despite material fatigue. Initially, the controller tracks the desired trajectory accurately within the safe region (i, red line; ii). After 3000 cycles, the controller without the adaptive CBF fails, causing the manipulator to violate the predefined safe region (i, cyan line; iii). In contrast, our framework with the adaptive CBF enabled compensates for the model degradation, successfully maintaining safe and accurate performance (i, purple line; iv). **b** Safe navigation under physical constraints. This experiment aimed to demonstrate the algorithm's capacity to handle actuator input limits during challenging maneuvers. A robotic fish was tasked with executing a tight turn in a confined space, comparing our method against a standard PID controller. The PID controller fails the maneuver due to input saturation (i, red dashed line; iii). Our framework successfully navigates the tight turn by generating constraint-compliant control actions, keeping the fish on its intended path (i, cyan dashed line; ii). **c** Real-time control in a dynamic environment. This experiment validated the algorithm's efficiency in real-time tracking of a speed-varying target using a robotic fish. The results demonstrate successful real-time tracking, as the fish's speed

closely matched the target's speed, which reached a maximum of $0.32$ m/s (i, iii). The controller consistently maintained a close tracking distance throughout the 60-second trial (ii).

## Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- movie1.mp4
- movie1.mp4
- movie1.mp4
- movie1.mp4
- movie2.mp4
- movie2.mp4
- movie2.mp4
- movie2.mp4
- movie3.mp4
- movie3.mp4
- movie3.mp4
- movie3.mp4
- movie4.mp4
- movie4.mp4
- movie4.mp4
- movie4.mp4