An Experimental Validation of the Polynomial Curvature Model: Identification and Optimal Control of a Soft Underwater Tentacle

Francesco Stella , Nana Obayashi , *Graduate Student Member, IEEE*, Cosimo Della Santina*, Member, IEEE*,
and Josie Hughes *, Member, IEEE*

***Abstract*—The control possibilities for soft robots have long been hindered by the lack of accurate yet computationally treatable dynamic models of soft structures. Polynomial curvature models propose a solution to this quest for continuum slender structures. Nevertheless, the results produced with this class of models have been so far essentially theoretical. With the present work, we aim to provide a much-needed experimental validation to these recent theories. To this end, we focus on soft tentacles immersed in water. First, we propose an extension of the affine curvature model to underwater structures, considering the drag forces arising from the fluid-solid interaction. Then, we extensively test the model's capability to describe the system behavior across several shapes and working conditions. Finally, we validate model-based control policies, proposing and solving an optimal control problem for directional underwater swimming. Using the model we show an average increase of more than 3.5 times the swimming speed of a sinusoidal baseline controller, with some tentacles showing an improvement in excess of 5.5 times the baseline.**

***Index Terms*—Modeling, control, and learning for soft robots, system identification, flexible robotics.**

I. Introduction

THE last decade witnessed exponential growth in the development of soft robots with an increasing range of actuation and motion possibilities [1], [2]. Nonetheless, their capabilities are still limited due to the overwhelming complexity of the associated control problem. Model based algorithms are proving to be a promising solution to this challenge, and thus receiving an increasing amount of attention [3]. However, these strategies require control-oriented models that are manageable in terms of complexity and number of states, while accurately representing the theoretically infinite deformation stated [4].

For soft slender robots,the most commonly used method is the celebrated Piecewise Constant Curvature (PCC) model [5], which is obtained from Cosserat's rod model [6] by neglecting all strains but curvature. The latter is then discretized as a piece-wise constant function. Despite its simplicity, this model has proven to work well when dealing with lightweight robots, subject to minimal interaction with the environment, and made of homogeneous materials and structures [7], [8]. However, whenever these hypotheses are not fulfilled, the soft robot will be eventually deformed out of the constant curvature hypothesis and higher order models need to be used. Examples of such systems include tentacles, trunks, flagella and cilia interacting with the environment [9]. More sophisticated models considering piecewise discretizations of all six strains [10], [11] suffer similar limitations, even if they are more capable of representing complex systems [12].

Functional expansions of the strain functions have been recently proposed with the aim of overcoming this issue without substantially increasing the number of states. For example, [13] introduces an extension of the constant curvature model to a polynomial expansion, while [14] specializes the analysis to an affine model. The model has been further and substantially extended to generic functional expansions of all the strains in [15]. Similar strategies have also been applied to the development of kinematic and dynamic models beyond strain discretization, as discussed in [16]. Kinematic and quasi-static [17]–[19] versions of these models have been recently validated for control and sensing purposes. However, there is no experimental validation of the effectiveness of these methods in their dynamic regime, neither as modeling techniques nor as a base for generation of the specific dynamic behaviours. The lack of such results opens up the question: are these models practically valuable for their general dynamic regime, or are they only an elegant mathematical formulation?

This paper aims to answer this question by providing such an experimental validation. More specifically, we propose to use a planar soft continuum robot inspired by a biological tentacle [20] undergoing dynamic deformations in water (see Fig. 1) as a validation platform. This testbed implements a good trade-off of complexity and manageability. Indeed, hydrodynamic effects act

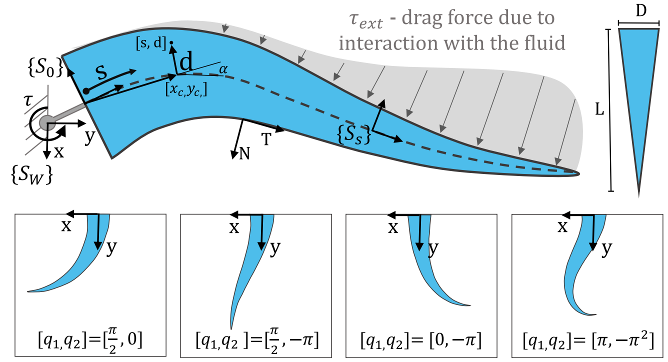
Manuscript received 24 February 2022; accepted 3 July 2022. Date of publication 21 July 2022; date of current version 31 August 2022. This letter was recommended for publication by Associate Editor C. Cao and Editor Y. L. Park upon evaluation of the reviewers' comments. *(Corresponding author: Francesco Stella.)*

Francesco Stella is with the CREATE Lab, EPFL, CH-1015 Lausanne, Switzerland, and also with the Department of Cognitive Robotics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: francesco.stella.97@live.com).

Nana Obayashi and Josie Hughes are with the CREATE Lab, EPFL, CH-1015 Lausanne, Switzerland (e-mail: nana.obayashi@epfl.ch; josie.hughes@epfl.ch).

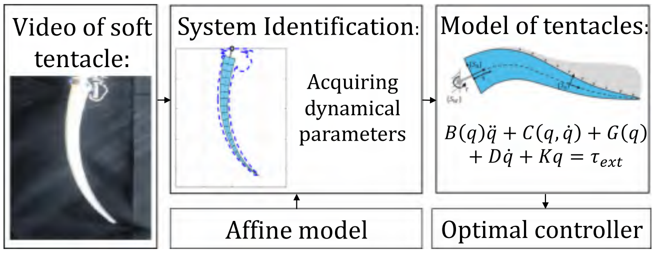
Cosimo Della Santina is with the Department of Cognitive Robotics, Delft University of Technology, 2628 CD Delft, The Netherlands, and also with the Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Wessling, Germany (e-mail: cosimodellasantina@gmail.com).

Digital Object Identifier 10.1109/LRA.2022.3192887



τ_{ext} - drag force due to interaction with the fluid
{S_0}
[s, d]
d
s
α
[x_c,y_c]
τ
y
x
{S_W}
{S_s}
T
N
D
L
x
y
$[q_1,q_2]=[\frac{\pi}{2},0]$
x
y
$[q_1,q_2]=[\frac{\pi}{2},-\pi]$
x
y
$[q_1,q_2]=[0,-\pi]$
x
y
$[q_1,q_2]=[\pi,-\pi^2]$

Fig. 1. The soft tentacle with main modeling elements highlighted. The central axis is shown as a dashed gray line. The system is actuated through a motor connected rigidly to the base of the tentacle. Its curvature at any point is an affine function s, which identifies the position along the main axis of the robot. The weights of this function are the Lagrangian variables of the system. The terms D and L define the diameter and length of the tentacle.



Video of soft tentacle:
System Identification:
Acquiring dynamical parameters
Model of tentacles:
$B(q)\ddot{q} + C(q,\dot{q}) + G(q) + D\dot{q} + Kq = \tau_{ext}$
Affine model
Optimal controller

Fig. 2. Schematic representation of the methodological steps performed. First the model is learnt from video by using system identification. The model can then be used to efficiently optimize the controller of the tentacle.

along the whole tentacle, making it bend in complex ways. Similar problems have fascinated the biomechanics community [21] and even inspired learning strategies [22]. The simplicity of the setup allows to directly measure all information necessary for proper validation of the theory and to test several variations on the same concept. Although we apply the model to the complex case of interactions between water and the soft body, the approach is applicable in any other scenario where a structure is subject to dynamical loads.

After extending the theory to include hydrodynamic forces, we perform a thorough identification and validation across 7 different geometries, 3 material properties, and an extensive class of excitation conditions. The proposed model can achieve accurate predictions in all datasets with only two configuration variables. Finally, we show that we can benefit from such a compact yet accurate description of the robot dynamics to do open-loop control. We formulate and numerically solve an optimal thrust problem, resulting in superior swimming capabilities compared to benchmark excitation patterns.

To summarize, this paper contributes to the state of the art in control-oriented models for soft robots with:

- a minimalist analytical model of a tentacle dynamics based on the affine-curvature model,
- a regression method to match the dynamical model to the real world tentacle through computer vision,
- an in-depth analysis of the performance of this variable strain model, providing a much-needed experimental validation to these theories,
- an optimal controller pattern that maximizes the thrust of the tentacle generated with evolutionary algorithms.

II. Modelling & System Identification

Polynomial curvature models offer an efficient method to represent the kinematic and dynamic properties of continuum soft slender structures with a finite approximation. In the following we present the kinematical and dynamical model when assuming that the polynomium describing the curvature is limited to the first order term. We believe that such truncation represents the best trade-off between accuracy and computational complexity for the tentacle-like structure under examination. As shown in Fig. 2 we will then use this model for system identification and optimization.

A. *Affine Curvature Model*

1) Kinematics: In this section we want to describe the kinematic model of a tentacle-like structure, represented in Fig. 1, starting from the description of the curvature of its central axis. We assume that the curvature of the central axis can be described by the affine function:

$$c(t) = q_1(t) + q_2(t)s \tag{1}$$

where q are the Lagrangian coordinates of the state and $s \in [0,1]$ parameterizes the position along the main axis of the tentacle, so that Ls is the arc length of the path connecting the base to the point s through the main axis. A reference frame S_s is connected at each point s, such that S_0 is the base frame and S_1 the tip frame. Within these frames the relative orientation of all frames w.r.t. the base frame is zero when the system in straight. The posture of S_s w.r.t. S_0 is $(x(s), y(s), \alpha(s)) \in R^3$ with $x(s)$ and $y(s)$ being the Cartesian coordinates of the frame's origin, and $\alpha(s)$ the local orientation. A further coordinate d is introduced to parametrize the points in the tentacle normal to the central axis. The Cartesian coordinates of a generic point (s, d) in the global frame are called (x_s, y_s, d), while in the local frame S_s they are $(d, 0)$. Therefore the overall shape of the robot at each time t is completely specified by the configuration of its central axis. The angle of the central axis α can be found by integration of the curvature:

$$\alpha(s) = \int_0^s c(v)\mathrm{d}v = q_1 s + \frac{q_2}{2}s^2, \tag{2}$$

where v is an auxiliary variable with the same meaning as s. Hence, the Cartesian position of point s on the central axis can be derived by:

$$x_\mathrm{c}(s) = \int_0^s -L\sin(\alpha(v))\mathrm{d}v, \quad y_\mathrm{c}(s) = \int_0^s L\cos(\alpha(v))\mathrm{d}v, \tag{3}$$

similarly, v is an auxiliary variable with the same meaning as s. We can then compute the position of a generic point defined by $[s, d]$ as

$$x(s,d) = x_\mathrm{c}(s) + d\cos(\alpha(s)),\ y(s,d) = y_\mathrm{c}(s) + d\sin(\alpha(s)).$$

From the definition of the central axis of the tentacle we can obtain the location of the perimeter and hence its velocity.

Moreover, we can define the normal and tangential vectors to the perimeter. These are required to determine the direction of the forces arising from the interaction with the environment. The 2D perimeter for the conical cross section of the tentacle can be given as:

$$x_{\mathrm{p}}(s) = x_c(s) \pm r(1-s)\cos(\alpha(s)),$$

$$y_{\mathrm{p}}(s) = y_c(s) \pm r(1-s)\sin(\alpha(s)), \tag{4}$$

where r is the radius at the base of the tentacle and $[x_c, y_c]$ represent the position of the central axis, as in (3). Hence, the velocity of a perimetral point can be given as:

$$v(s,q) = J_{s,q}\dot{q}, \tag{5}$$

where $J_{s,d}$ is the Jacobian of the point within the structure in $[s,d]$, i.e. $J_{s,d} = \frac{\mathrm{d}f(q,s,d)}{\mathrm{d}q}$, where f represents the forward kinematic mapping of a general point defined by $[s,d]$. Finally, for any given point on the perimeter, it is possible to define a normal, N, and tangential, P, vector respect to the perimeter as:

$$P = \frac{\mathrm{d}\begin{bmatrix} x_{\mathrm{p}}(s) \\ y_{\mathrm{p}}(s) \end{bmatrix}}{\mathrm{d}s}, \quad N = \frac{\mathrm{d}T}{\mathrm{d}s}. \tag{6}$$

Given P and N it is possible to write a transformation matrix from the base reference system to the perimeter reference system as:

$$M = \begin{bmatrix} P^\top \\ N^\top \end{bmatrix}. \tag{7}$$

2) Dynamical Model: Given the kinematic description of the structure, the dynamical model should take the form:

$$B(q)\ddot{q} + C(q,\dot{q}) + G(q) + D\dot{q} + Kq = \tau_{\text{ext}}, \tag{8}$$

where $B(q)$ and $C(q,\dot{q})$ represent the inertia matrix and the Coriolis terms respectively, $G(q)$ is the gravitational force, and D and K are the damping and stiffness matrices. τ_{ext} groups together the external forces which includes external disturbances, the forces that arise from the interaction between the tentacle and the water and the torque exerted by the motor at the base of the tentacle.

The mass matrix, expressed in joint space, can be written as:

$$B = \int_0^1 \int_{-\frac{1}{2}(1-s)}^{\frac{1}{2}(1-s)} \rho_s \, J_{s,d}(v,d)^\top \, J_{s,d}(v,d) \, \mathrm{d}v \, \mathrm{d}d, \tag{9}$$

where ρ_s represents the density of silicone. Note that, with this model, the volume is not preserved locally when the tentacle deforms, but the overall change in volume is limited to 2% for extreme deformations. The Coriolis terms are computed thanks to Christoffel symbols [23]. The gravitational field, together with the buoyancy force field can be expressed as:

$$G = \int_0^1 \int_{-\frac{1}{2}(1-s)}^{\frac{1}{2}(1-s)} -(\rho_s - \rho_w) g y_{s,d}(v,d) \ \mathrm{d}v \, \mathrm{d}d, \tag{10}$$

where ρ_{w} represent the specific mass of water. The elastic field can be determined from the potential elastic energy:

$$U_{\mathrm{E}}(q,k) = \frac{1}{2}\int_0^1 k(s)\alpha^2(s,t)\mathrm{d}s, \tag{11}$$

where $k(s) : [0,1] \to \mathbb{R}^+$ associates a local flexural stiffness to any point along the central axis. Evaluating the $j-$th partial derivative of U_{E} with respect to q_j yiels to the elastic force:

$$G_{E,j}(q) = \frac{1}{2}\int_0^1 k(s)\frac{\partial \alpha^2(s,t)}{\partial q_j}\mathrm{d}s = \tag{12}$$

$$= \int_0^1 k(s)\alpha(s,t)\frac{\partial \alpha(s,t)}{\partial q_j}\mathrm{d}s = \int_0^1 k(s)\left(\sum_{i=0}^{1} q_i(t)s^i\right)s^j\mathrm{d}s,$$

therefore the elastic force field can be rewritten as:

$$G_E(q) = Kq(t), \quad K_{i,j} = \int_0^1 k(s)s^{i+j}\mathrm{d}s. \tag{13}$$

Given the structure of the tentacle, we assume the stiffness function $k(s)$ to be an affine function $k(s) = k_0 + k_1 s$. Similarly, the damping force field S_D can be evaluated by:

$$S_D(q) = D\dot{q}(t), \quad D_{i,j} = \int_0^1 d(s)s^{i+j}\mathrm{d}s, \tag{14}$$

where the damping function is also assumed to be affine, i.e. $d(s) = d_0 + d_1 s$. Finally, the drag forces can be written as [24]:

$$f_{\mathrm{d}} = \left(-\frac{1}{2}\rho_{\text{water}} v^\top v C_\perp A \frac{v}{|v|} \cdot N\right) N, \tag{15}$$

where A represents the area of the tentacle perpendicular to the plane in which the motion takes place and $C_\perp$ represents a linear coefficient on the drag forces. The total Lagrangian force τ_{ext} that acts on the tentacle is then:

$$\tau_{\text{ext}}(q,\dot{q}) = \int_0^1 J_s^\top f_{\mathrm{d}}(s)\mathrm{d}s, \tag{16}$$

which is evaluated numerically. Similarly, the total thrust produced in the x and y direction can be evaluated as:

$$T(q,\dot{q}) = \int_0^1 f_{\mathrm{d}}(s)\mathrm{d}s. \tag{17}$$

B. System Identification

In order to apply the model to the physical tentacle, we first need to identify the relevant dynamical parameters.

1) Data Acquisition: In order to do so, the tentacle is rigidly connected at the base to a motor driven with a sinusoidal wave (amplitude $\pi/6$, frequency 3 Hz) for 24 cycles which equates to 39 s of motion. A camera is placed manually so to be parallel to the plane of motion of the tentacle, at a distance of 30 cm, capturing the motion at 25 frame/s. As the actuation force and motion of the tentacle are both in a plane parallel to the camera, we restrict the analysis at a 2D system. For more information about the experimental setup, please refer to Section IV-A. The video is analysed through a computer vision algorithm written in Matlab using the Computer Vision toolbox to extract the curvature variables $[q_1, q_2]$ over time.

The image analysis steps are depicted in Fig. 3. For each frame (subfigure a), the image is binarized (subfigure b-c) and the largest contour is detected (subfigure d). The perimeter is then averaged to obtain the central axis curve profile (subfigure e). Finally, the affine-curvature kinematic model variables $[q_1, q_2]$ are fitted to the backbone curve profile using a least-square fitting (subfigure e). To implement such fitting, for each frame, we compute the least-square error with respect to the boundary found with computer vision for each point of a grid that covers the whole workspace of $[q_1, q_2]$. Hence, the evolution of q is

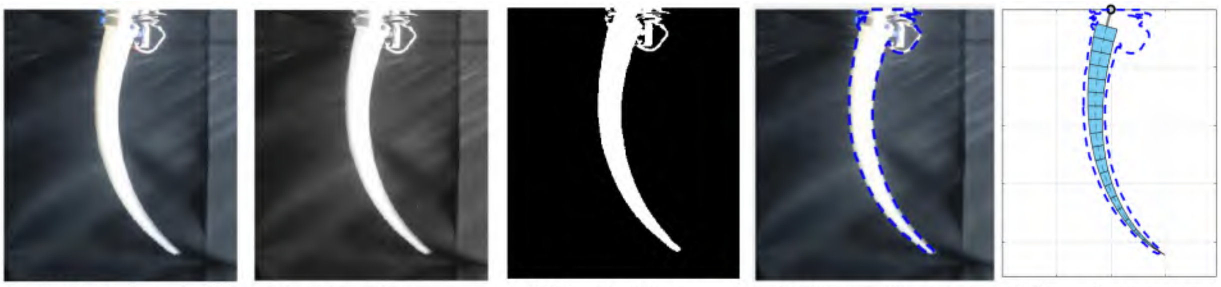

(a) Initial frame (b) Gray scale image (c) Binarized image (d) Contour detected (e) Kinematic regression

Fig. 3. Image processing pipeline, applied to each frame of the raw video file to obtain the contour of the tentacle (a-d). The last step shows the fitted model super-imposed to the extracted contour.

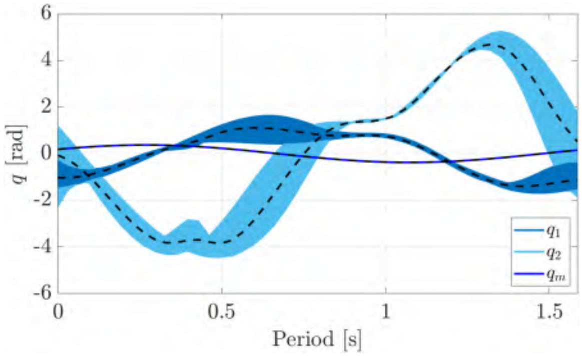


Fig. 4. Evolution of the tentacle extracted with the computer vision algorithm. The dotted line shows the average evolution of each coordinate over one period, while the colored areas represent the standard deviation of each coordinate over the whole recording.

averaged over multiple periods, so to obtain the average motion per period, shown in Fig. 4. The overall curvature trends are consistent and coherent with the recording.

2) Dynamical Parameter Estimation: Thanks to the evolution of q over time, we can infer the parameters of the dynamics. In this work we consider the following as parameters to be estimated with a regression on experimental data: $k_0, k_1, d_0, d_1, C_\perp$. We are therefore infering simultaneously the physics of the tentacle and of the body-water interaction. We call π the vector collecting these parameters. Instead, we measure experimentally all the parameters that can be easily done so with good accuracy, which are lengths, masses and inertias. This step allows us to reduce the parameter space to only those that can not be easily evaluated experimentally. Thus we can re-write the dynamics as:

$$\delta(q, \dot{q}, \ddot{q}) = Y(q, \dot{q}, \tau)\pi, \tag{18}$$

where δ collects the first three terms in (8) - the dynamic forces - and Y is such that $Y\pi$ collects all the remaining terms. We call $\tilde{q}_1, \tilde{q}_2$ $\tilde{\tau}$ the measurements of curvature and motor torque gathered in the above discussed experiments. In particular, $\tilde{\tau}$ is computed based on a dynamical model of the motor. We assume that the servomotor low level controller is a PID control that, given an error on the motor orientation, generates the torque $\tilde{\tau}$ to bring the base of the tentacle to the angle experimentally measured. We call $t_1, \ldots, t_m$ the instances during which the measurements are done. We evaluate the set of parameters that better explain the real data by minimizing the least mean square error of $||\delta - Y\pi||_2^2$ across all data. This is achieved by:

$$\hat{\pi} = \begin{bmatrix} Y(\tilde{q}_1(t_1), \ldots, \tilde{\tau}_2(t_1)) \\ \vdots \\ Y(\tilde{q}_1(t_m), \ldots, \tilde{\tau}_2(t_m)) \end{bmatrix}^+ \begin{bmatrix} \delta(\tilde{q}_1(t_1), \ldots, \ddot{\tilde{q}}_2(t_1)) \\ \vdots \\ \delta(\tilde{q}_1(t_m), \ldots, \ddot{\tilde{q}}_2(t_m)) \end{bmatrix}, \tag{19}$$

where $\cdot^+$ is the Moore-Penrose pseudo-inverse. Time derivatives are calculated through numerical differentiation. Note that such method is applicable only if the relation between the known terms and the parameters is linear. Optimization methods need to be used to perform the identification for cases in which the relation is non-linear. The resulting parameters are $\pi = [6.20\frac{\text{Nm}}{\text{rad}}, 4.92\frac{\text{Nm}}{\text{rad}}, 14.55\frac{\text{Nm}s}{\text{rad}}, 12.24\frac{\text{Nm}s}{\text{rad}}, 0.60]$. The accuracy of the model is then evaluated by comparing the evolution computed by integrating the acquired model with respect with the motion of the real tentacle. The evaluation is performed on a sinusoidal trajectory while a chirp signal is used to generate the training data used to extract the parameters. As shown in Fig. 5, the motion of the tentacle matches well with the evolution. The initial conditions for the integration are based on the numerical differentiation of the evolution extracted through computer vision. Fig. 6 shows on the top row the tentacle motion overlaid with the perimeter extracted through computer vision. On the bottom, instead it shows the resulting evolution of the acquired model overlaid on the real tentacle motion.

III. Optimization Method

In previous works [25], we reported on the fluid propulsion generated by an oscillating tentacle structure. Similar designs have been proposed in different scales, ranging from micro-robotics [9], to large scale robots [12]. In all of these works, the oscillating structure was actuated with very simple, hard-coded, periodic patterns, such as sinusoidal waves. The resulting desired oscillatory behavior is therefore an emergent characteristic of the soft body [26]. Yet, solving the problem by completely relying on physical intelligence has two limitations. First, the robot is limited to the emergent behavior, restricting the possibilities of applications and the optimality of the performance. Secondly, the unavoidable differences that exist between the actual physical robot and the ideal case may change drastically the resulting

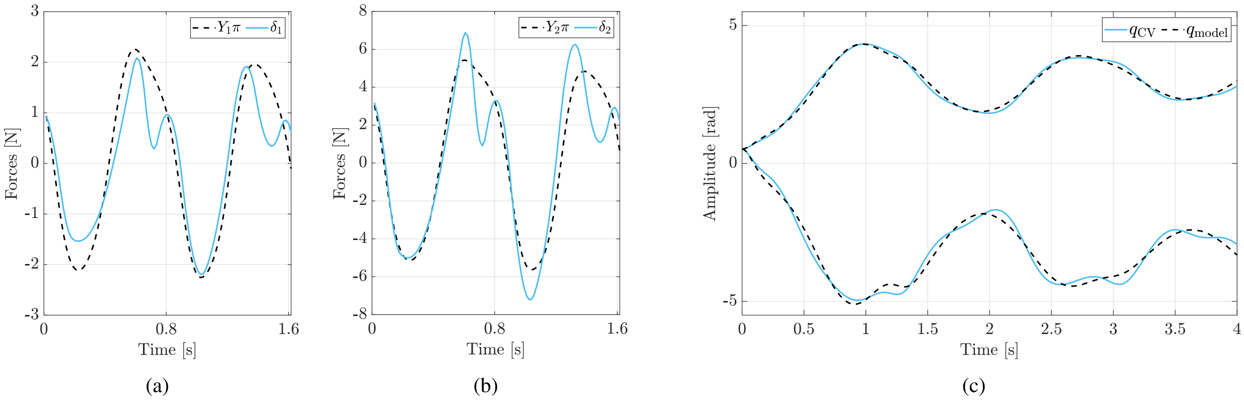
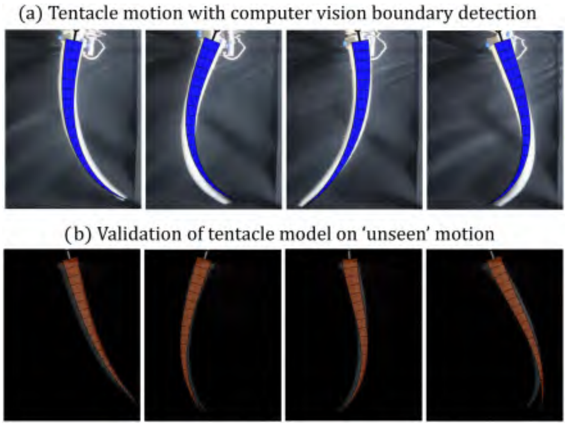


Fig. 5. For one example tentacle of dimensions d = 25 mm and l = 140 mm the measured parameters in comparison to those found with the model, corresponding to the inputs found utilizing the computer vision in Fig. 4. The two left panels show the difference between the rows of δ and $Y\pi$ (i.e. what we are actively optimizing for). The right panel shows the simulated evolution compared to the real data.



(a) Tentacle motion with computer vision boundary detection
(b) Validation of tentacle model on 'unseen' motion

Fig. 6. a) Stills from the motion of the tentacle overlaid with the perimeter extracted using computer vision and b) Generated model overalaid over the corresponding real world tentacle motion. It is worth stressing that predictions in (b) are obtained as open loop simulation of the identified model.

open loop response, especially for the micro-structures. In turn, these two factors may result in sub-optimal behaviors of the soft robot. Both these limitations can be addressed by augmenting the intelligent body with a suitable soft robotic brain. Indeed, thanks to the accuracy of the affine curvature model, we are then able to efficiently optimize the open-loop performance of these oscillating structures in simulation.

A. Controller Optimization

The model allows us to efficiently optimize the performance of the tentacle for a specific task. Specifically we can evaluate the possibilities of improving the thrust generation of a specific tentacle. The trajectory input to the motor has been defined to span the space of sinusoidal, triangular and square waves with varying amplitude, frequency and phase:

$$q_{\mathrm{m}}(t) = \alpha_1 \underbrace{\sin\left(\alpha_4 t + \alpha_7\right)}_{\text{Sinusoidal term}} + \alpha_2 \underbrace{\sum_{n=1}^{+\infty} \frac{\sin\left((\alpha_5 (2n-1) t + \alpha_8\right)}{2n-1}}_{\text{Square wave term}} + \alpha_3 \underbrace{\sum_{n=1}^{+\infty} \frac{\sin\left(\alpha_6 n t + \alpha_9\right).}{n}}_{\text{Saw-tooth wave term}} \tag{20}$$

Relying on the affine curvature model, we can then find optimal actuation policies that maximize the thrust generated by the tentacle in simulation. In order to achieve this result, we have developed an evolutionary algorithm based approach that optimizes the thrust produced by the tentacle using the acquired model. The bounds for the optimizable parameters α have been set based on the physical constraints of the motor dynamics. We can define this optimization problem as:

$$\max_{\alpha} \int_0^{t_{end}} T_{\mathrm{y}}(q_{\mathrm{m}}(\alpha))\mathrm{d}t, \quad \text{s.t.}\ l_{\mathrm{bound}} \leq \alpha \leq u_{\mathrm{bound}} \tag{21}$$

where T_{y} represents the thrust produced by the tentacle in the vertical direction and the upper and lower bounds $u_{\mathrm{bound}}, l_{\mathrm{bound}}$ were defined based on the physical limitations on amplitude and frequency imposed by the motor dynamics. Since the optimization problem is highly non-convex due to the numerical integration of the EOMs, we can not rely on gradient-based optimization methods, and a heuristic based algorithm, such as genetic algorithms, should be used. The evolutionary algorithm has been integrated using the Matlab Global Optimization toolbox, while the integration of the EOMs has been implemented in Simulink. The parameters used in the genetic algorithm are: population size $= 50$, mutation rate$= 0.02$, maximum number of generations $= 100$

IV. Experimental Investigation

A. Experimental Setup

The tentacles are manufactured by casting silicones of different stiffnesses into 3D printed molds. The molds, represented in Fig. 7, are composed from two half-cones to allow the cured tentacle to be released from the mould. In order to experimentally

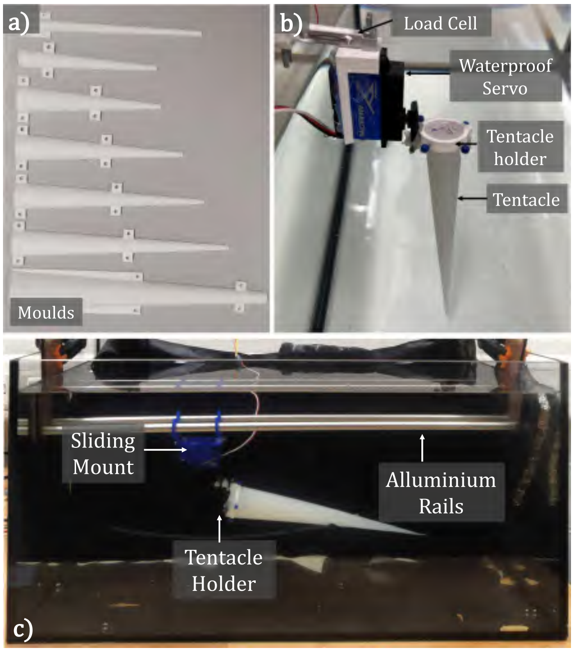


Fig. 7. a) Moulds used to fabricate the tentacles, b) Experimental setup for data collection and c) Experimental setup for speed evaluation.

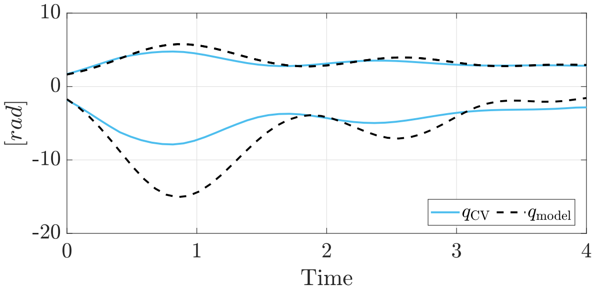


Fig. 8. To illustrate the range of error observed in the two curvature terms we show the reconstruction with the maximum error, $e = [5.7, 22.2]$, i.e. the case $L = 25$ cm. The best reconstruction is shown in Fig. 4.

demonstrate the analysis presented above we have developed two experimental setups. The first is for data collection where the tentacles actuator is fixed for system identification, and the second 'free swimming' setup to allow for experimental comparison of different controllers.

1) Tentacle Data-Collection Setup: To gather the necessary data for the methods presented above, the data collection system has been designed to simultaneously collect videos of the tentacle deformation, the generated thrust force, and the motor position (Fig. 7(b)). To perform the water-based experiments a 30 × 60 × 30 cm tank is used. The tentacle is actuated by a waterproof servo motor (5821LV Xam racing) which is attached to the base, via a 3D printed holder. The motor is connected to a load cell (TAL221, 500 g) which is fixed rigidly to the side of the tank. The orientation of the load-cell has been set such that it measures the thrust produced in the vertical direction - i.e. the forward swimming direction.

The load cell data is obtained using an amplifier board and an Arduino DUE microcontroller. A camera (Logitech 4 K pro) is placed in front of the tank to record the deformation of the tentacle. The motor, load-cell and camera are centrally controlled by a computer which is able to send and receive data through serial connections and to allow the synchronization of the generated data.

TABLE I
ERROR BETWEEN THE RECONSTRUCTED MODEL AND THE REAL EVOLUTION OF THE TENTACLE, DETECTED THROUGH COMPUTER VISION

	Error (rad)					
Length	**Eco-Flex**		**Dragon Skin 10**		**Dragon Skin 20**	
L (cm)	**q1**	**q2**	**q1**	**q2**	**q1**	**q2**
12	2.2	5.1	3.1	4.3	1.1	6.2
15	4.2	6.7	3.8	5.8	2.8	9.5
18	5.1	8.9	4.2	15.4	5.4	8.2
22	6.7	20.1	4.2	9.8	5.0	6.3
25	5.7	22.2	5.2	6.8	6.7	15.6

This setup allows us to perform dynamic experiments where the tentacle is actuated at its base with different input motions, while recording the visual displacement and the thrust generated. Due to the small volume of the tank, standing waves could potentially disturb the measurements. Therefore, between experiments, we wait for the standard deviation of the load-cell readings to fall under an hard-coded threshold, which indicates that the water has settled, before starting the next experiment. However, such disturbances are captured to some extent thanks to the estimation of the fluid-solid interaction parameter $C_{\perp}$.

2) Swimming Setup: To evaluate the swimming performance of various tentacles a second setup (Fig. 7(c)) is used. In this second setup the tentacles are attached to the actuator in the same configuration, but to allow forwards free swimming are guided by long low-friction aluminium rails. The rails have been designed to exert minimal effects on the swimming behavior but are used to allow for easy and meaningful comparison of swimming speeds between different tentacles.

B. System Identification

In order to validate the robustness of theidentification method, we performed the procedure described in Section II-B2 on multiple tentacles of varying geometries and materials. In Table I the average error across all frames $e = \frac{1}{t_{\text{end}}} \sum_{\text{t}=0}^{t_{\text{end}}} (q_{\text{model}} - q_{\text{CV}})^2$ between the reconstructed evolution q_{CV} and model q_{model} for each experiment is given. To visually represent the error, we report a time series of the real and predicted curvature values for the worst and best case errors in Fig. 8.

The results shows that with an affine curvature model and with the described identification method, we can accurately model structure with different geometries and stiffnesses. Interestingly, structures with higher stiffness (Dragon Skin 20) and lower lengths show less error. In these regimes the lower stiffness results in the hydrodynamic effects having a more significant impact on the behavior of the tentacle which may be less well captured by the model. In addition, EcoFlex, which has the highest error, also has the most non-linear material properties of the three silicones tested. This may again lead to the model capturing the true behaviour with a poorer performance.

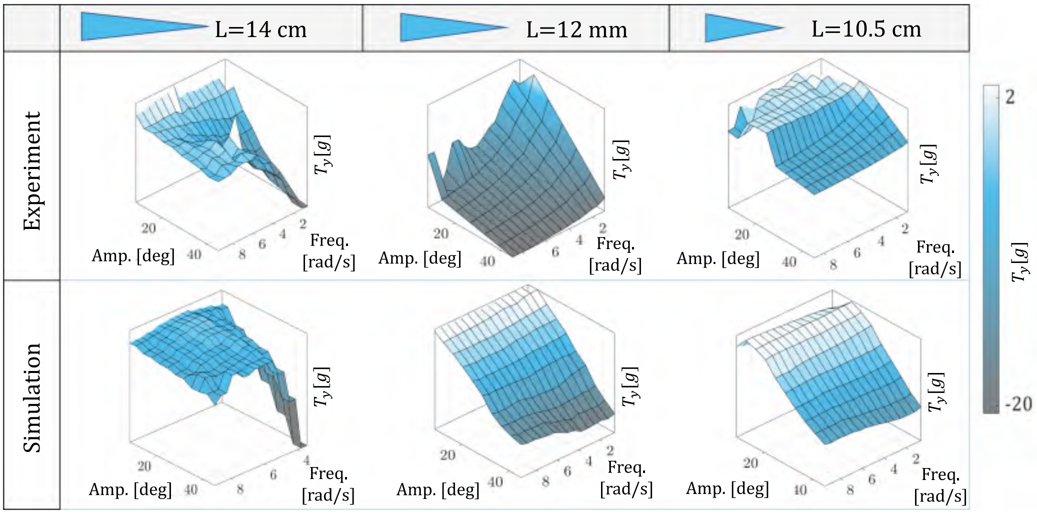


Fig. 9. Baseline performance achieved with sinusoidal inputs only. In the top row, we can observe the mean thrust produced experimentally by three tentacles. In the bottom row, it is instead shown the thrust computed in simulation. Interestingly, similar trends can be observed between simulation and real-experiment results. Therefore, thanks to the model we can predict the thrust performance of the tentacle as a function of the input motion of the motor.

The error on the second term of the curvature polynomial is generally higher than the first term. This is related to the definition and formulation of this second term in the curvature (2), where the angle of curvature is given by the square of this curvature term and hence is numerically more sensitive than lower order terms.

C. Estimation of Thrust

Thanks to the acquired model we can estimate the thrust generated by the tentacle by integrating the EOM over 10 seconds and evaluating the average thrust τ_{ext} in the x and y direction over this period. The thrust predicted by the model can be compared to that measured experimentally. This experiment has been repeated for 3 tentacles across a range of sinusoidal control inputs with varying amplitude and frequency. This allows the trends across the controller design space to be observed and compared for the modelled and experimentally tested tentacles. The results are shown in Fig. 9. Firstly it can be seen that with only the visual information that is obtained during system identification the generated models have absolute values similar to the experimental ones. Secondly, although there is some discrepancy in the absolute values the model captures and reflects the trends seen in the experimental results. For the longest tentacle (L = 140 mm) we see the largest amplitude, lowest frequency controllers performing best with this trend reflected in the model. For the mid-length tentacle (L = 120 mm) the model reflects the gradient towards higher performing controllers occurring at higher frequencies and amplitudes. Finally the shortest tentacle (L = 105 mm) shows the weakest controllers to be in the middle range, with the best towards low amplitude low frequency.

D. Controller Optimization

To assess the performance increase that can be achieved using the models, an optimized controller has been found and evaluated on three tentacles of different geometries. The three different tentacles were chosen to span the design space and to showcase the ability to optimize for multiple tentacles. For each tentacle the optimized controller was evaluated and benchmarked against the best sinusoidal input and the average sinusoidal input, as extracted from the experiments shown in Fig. 9. Each experimental combination was repeated five times in the experimental swimming setup shown in Fig. 7(c). As shown in Fig.4 10 for all three tentacles the optimized controller shows a increase in performance compared to the sinusoidal baselines. The quantitative results given in Table II show an improvement of 5.5 times that of the fastest baseline speed for the shortest tentacles, over 2 times for the middle length, and over 3 times for the longest tentacle. Thus, the controller optimization leverages the model to achieve a significant increase in performance. This highlights that the model captures the dynamic behaviour of the tentacles with good accuracy whilst providing a reduced order form that allow for computationally efficient controller optimization. Note that this experiment is not intended to propose novel control techniques for tentacles, but to demonstrate that, thanks to an accurate modeling, the performance can be improved with simple, open-loop, control.

TABLE II
AVERAGE SWIMMING VELOCITY AND THE STD REPORTED FOR 3 TENTACLES WITH EACH SWIMMING EXPERIMENT REPEATED FIVE TIMES

	Swimming Speed ($\frac{cm}{s}$)					
Length	**Low Freq. Sinusoid**		**High Freq. Sinusoid**		**Optimized controller**	
L (cm)	**mean**	**std**	**mean**	**std**	**mean**	**std**
14	0.06	0.01	0.12	0.02	0.66	0.01
21	0.52	0.15	1.04	0.12	2.27	0.31
25	0.49	0.02	0.67	0.07	2.14	0.46

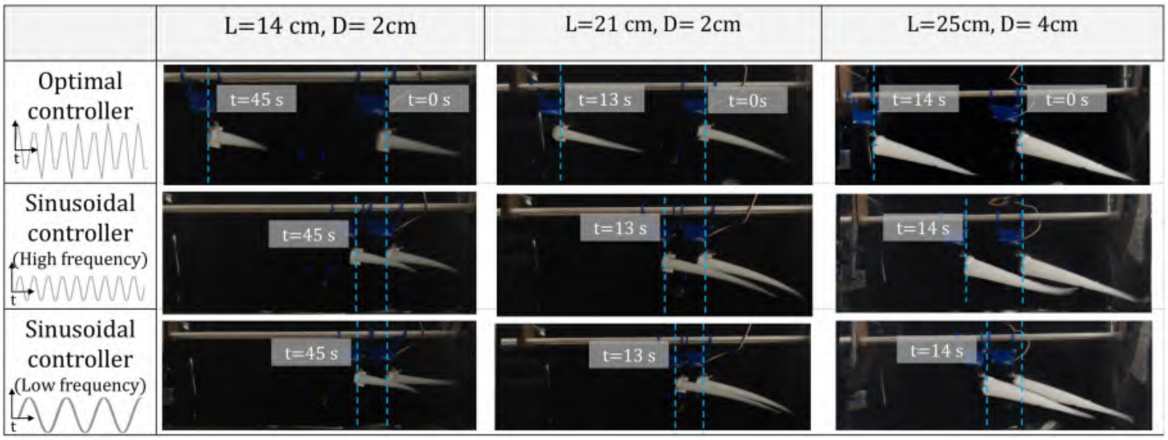


Fig. 10. Velocity comparison of three tentacles actuated with three different controllers, shown in the left column. The difference in speed demonstrates the improvement in performance achieved with accurate modeling of soft structures.

V. Conclusion

Modeling the soft body dynamics is essential to enable soft robot structures to be designed, controlled, and deployed for real-world applications. This work explored how a simple affine curvature model can capture complex interactions between a soft tentacle and the environment and represent these in a computationally-efficient manner. This is the first experimental validation of variable strain models of soft robots. To this end, we first have proposed an extension of the dynamic affine curvature model that includes a simple description of hydrodynamic effects. Second, we have discussed a method for identifying the relevant dynamical parameters from only a single video. Third, we extensively evaluate the performance of this model over several designs. Finally, we show that this compact model allows optimizing the control input to maximize the thrust in a swimming task.

In future work, we will further leverage the reduced-order complexity of the model to perform co-optimization of the geometrical structure, material properties, and controller within a complete soft octopus system.

References

[1] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, May 2015.

[2] C. Della Santina, M. G. Catalano, and A. Bicchi, "Soft robots," *Encyclopedia Robot.*, vol. 489, pp. 1–15, 2020.

[3] C. Della Santina, C. Duriez, and D. Rus, "Model based control of soft robots: A survey of the state of the art and open challenges," 2021, *arXiv:2110.01358*.

[4] C. Armanini, C. Messer, A. T. Mathew, F. Boyer, C. Duriez, and F. Renda, "Soft robots modeling: A literature unwinding," 2021, *arXiv:2112.03645*.

[5] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1661–1683, Nov. 2010.

[6] J. Till, V. Aloi, and C. Rucker, "Real-time dynamics of soft and continuum robots based on cosserat rod models," *Int. J. Robot. Res.*, vol. 38, no. 6, pp. 723–746, May 2019.

[7] I. S. Godage, R. Wirz, I. D. Walker, and R. J. Webster III, "Accurate and efficient dynamics for variable-length continuum arms: A center of gravity approach," *Soft Robot.*, vol. 2, no. 3, pp. 96–106, Sep. 2015.

[8] B. J. Caasenbrood, A. Y. Pogromsky, and H. Nijmeijer, "Dynamic modeling of hyper-elastic soft robots using spatial curves," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9238–9243, 2020.

[9] E. Milana, F. Stella, B. Gorissen, D. Reynaerts, and C. Della Santina, "Model-based control can improve the performance of artificial cilia," in *Proc. IEEE 4th Int. Conf. Soft Robot.*, 2021, pp. 527–530.

[10] S. Grazioso, G. Di Gironimo, and B. Siciliano, "A geometrically exact model for soft continuum robots: The finite element deformation space formulation," *Soft Robot.*, vol. 6, no. 6, pp. 790–811, 2019.

[11] F. Renda, F. Boyer, J. Dias, and L. Seneviratne, "Discrete cosserat approach for multisection soft manipulator dynamics," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1518–1533, Dec. 2018.

[12] C. Armanini, M. Farman, M. Calisti, F. Giorgio-Serchi, C. Stefanini, and F. Renda, "Flagellate underwater robotics at macroscale: Design, modeling, and characterization," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 731–747, Apr. 2022.

[13] C. D. Santina and D. Rus, "Control oriented modeling of soft robots: The polynomial curvature case," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 290–298, Apr. 2020.

[14] C. D. Santina, "The soft inverted pendulum with affine curvature," in *Proc. 59th IEEE Conf. Decis. Control*, 2020, pp. 4135–4142.

[15] F. Boyer, V. Lebastard, F. Candelier, and F. Renda, "Dynamics of continuum and soft robots: A strain parameterization based approach," *IEEE Trans. Robot.*, vol. 37, no. 3, pp. 847–863, Jun. 2021.

[16] S. Sadati, S. E. Naghibi, L. Da Cruz, and C. Bergeles, "Reduced-order modeling and model order reduction for soft robots," 2021.

[17] F. Xu, H. Wang, W. Chen, and Y. Miao, "Visual servoing of a cable-driven soft robot manipulator with shape feature," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4281–4288, Jul. 2021.

[18] P. Rao, Q. Peyron, and J. Burgner-Kahrs, "Using Euler curves to model continuum robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 1402–1408.

[19] L. Besselaar and C. Della Santina, "One-shot learning closed-loop manipulation of soft slender objects based on a planar polynomial curvature model," in *Proc. IEEE 5th Int. Conf. Soft Robot.*, 2021, pp. 518–524.

[20] Y. Yekutieli, R. Sagiv-Zohar, R. Aharonov, Y. Engel, B. Hochner, and T. Flash, "Dynamic model of the octopus arm. I. Biomechanics of the octopus reaching movement," *J. Neuriophysiol.*, vol. 94, no. 2, pp. 1443–1458, 2005.

[21] J. Elgeti, R. G. Winkler, and G. Gompper, "Physics of microswimmers–single particle motion and collective behavior: A review," *Reports Prog. Phys.*, vol. 78, no. 5, 2015, Art. no. 056601.

[22] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, "Learning dynamic models for open loop predictive control of soft robotic manipulators," *Bioinspiration biomimetics*, vol. 12, no. 6, 2017, Art. no. 066003.

[23] M. Safeea, P. Neto, and R. Bearee, "Robot dynamics: A recursive algorithm for efficient calculation of Christoffel symbols," *Mechanism Mach. Theory*, vol. 142, 2019, Art. no. 103589.

[24] F. Renda, M. Giorelli, M. Calisti, M. Cianchetti, and C. Laschi, "Dynamic model of a multibending soft robot arm driven by cables," *IEEE Trans. Robot.*, vol. 30, no. 5, pp. 1109–1122, Oct. 2014.

[25] N. Obayashi, C. Bosio, and J. Hughes, "Soft passive swimmer optimization: From simulation to reality using data-driven transformation," in *Proc. IEEE 5th Int. Conf. Soft Robot.*, 2022, pp. 328–333.

[26] H. Hauser, A. J. Ijspeert, R. M. Füchslin, R. Pfeifer, and W. Maass, "Towards a theoretical foundation for morphological computation with compliant bodies," *Biol. Cybern.*, vol. 105, no. 5, pp. 355–370, Dec. 2011.