ME5416 Mathematics for Robotics Engineering: Mathematical Modelling of the Robot Octopus Arm

Penglin Cai

April 21, 2025

In this report, an octopus-like robot arm is modelled and simulated. To achieve efficient and precise manipulation of soft robotic structures in complex environments, this work introduces an octopus-arm simulation framework that combines the Cosserat rod model with scheduled, phase-based PID control. By integrating interpolation-driven Y-axis mid-node actuation alongside full-dimensional end-effector PID force feedback and enforcing force clipping for high-load stability, the approach preserves continuum flexibility while accurately tracking predefined motion trajectories. Leveraging Position Verlet time integration within the PyElastica framework, and imposing hinge boundary conditions at the one end of the rod, a repeated multi-target cyclic motion simulation is conducted that offers a reproducible numerical toolset and practical engineering insights for soft robotics design and control. The inspiration comes from stereotypical movement patterns in octopus arms such as reaching, the simulation task focuses on the reaching with 0.5 to 2 meters workspace in an obstacle-free underwater environment.

Bio-inspired | Soft filaments | Cosserat rods | PyElastica

1. Background and Motivation

The octopus arm represents an exemplary model for studying complex manipulation and motion control due to its extraordinary flexibility, compliance, and virtually infinite degrees of freedom. Unlike rigid articulated limbs seen in traditional robotic systems or human arms, octopus arms leverage their soft, muscular architecture to achieve sophisticated interactions with their environment, such as reaching, fetching, grasping, and manipulating objects. This remarkable capability arises from a distributed control system, where muscles along the arm independently contract, generating coordinated bending, stretching, twisting, and shearing movements. Understanding and replicating these biological strategies present significant potential for innovation in soft robotics, promising enhancements in robot dexterity, adaptability, and safety. Consequently, recent research efforts have focused on developing control methodologies inspired by the biomechanics of the octopus, such as optimal control frameworks and energy shaping techniques, to capture essential movement patterns observed experimentally and apply them to robotic applications. Investigating these approaches not only provides insight into the underlying mechanisms of biological motor control but also advances the field of soft robotic design and control.

Recent advances have further employed Cosserat rod theory to establish a comprehensive simulation method for soft filaments (1), capturing their full range of deformation including bending, twisting, stretching, and shearing. This computational approach has been validated through various forward and inverse mechanical problems, enabling accurate modeling of the dynamics of soft slender structures interacting with complex environments. By simulating active behaviors such as muscle-driven actuation, and incorporating environmental interactions such as friction, hydrodynamics, and contact mechanics, the Cosserat framework offers a robust toolset for exploring and optimizing the intricate mechanics of soft robotic systems and bio-inspired designs. Furthermore, the same research team has progressed one step further to model a more complex bio-inspired structure, they present a versatile numerical method based on Cosserat rod theory for modeling and simulating complex dynamic musculoskeletal architectures (2). The authors assemble active and passive Cosserat rods into dynamic structures capable of representing bones, tendons, ligaments, fibers, and muscle connectivity. This methodology is successfully applied across various biological and soft robotic scenarios spanning different scales and environments, including the design of millimeter-scale bio-hybrid robots, synthesis and reconstruction of intricate musculoskeletal systems, and the simulation of feathered bird wings. The paper demonstrates the generality and effectiveness of the proposed approach, providing a practical computational framework for deeper insights into biological function and aiding forward and inverse bioengineering designs.

Many biological and engineered systems consist of soft, slender elements that may act autonomously or remain passive. In nature, these include DNA helices, flagella, muscle fibers, and snakes, demonstrating the immense variety of length scales involved. In human-made devices, similar slender components appear as woven fabrics, artificial muscles, and lengthy cables. Whenever a structure's length far exceeds its cross-sectional radius (i.e. $L/r \gg 1$), it can be treated as a one-dimensional rod, which greatly simplifies its mathematical description. Cosserat rod theory provides just such a framework, capturing bending, twisting, stretching, and shearing so that all six degrees of freedom at each cross-section are represented, making it suitable for modelling the octopus arm. Individual Cosserat rods can model single strands, or they can be linked together into more intricate assemblies such as networks of muscle fibers or the interlaced twigs of a bird's nest.

In this report, the Cosserat rod theory is applied for modelling the soft robotic arm due to its ability to comprehensively capture all fundamental deformation modes such as bending, twisting, shear, and stretch that are crucial for accurately representing the mechanical behavior of soft, slender structures. Unlike traditional beam theories that often neglect shear and extensibility, the Cosserat approach allows the cross-sections to rotate, shear, and elongate freely, effectively accommodating large deformations typical in soft robotics. This makes it highly suitable for modeling realistic interactions with complex environments, including contact, friction, and muscular-like actuation, ultimately enabling robust design and control strategies for sophisticated manipulation tasks.

Compared to other modeling approaches, such as mass-spring systems, pseudo-rigid body models, or even classical finite element methods, the Cosserat rod formulation offers a unique balance between computational efficiency and physical fidelity. While mass-spring models are easy to implement, they lack physical accuracy in capturing coupled deformation modes. Finite element methods, on the other hand, can be highly accurate but are computationally intensive and less suitable for real-time control. The Cosserat rod model, by contrast, provides a geometrically exact and dynamically consistent representation that is lightweight enough for real-time simulation and control, while still preserving essential nonlinear behaviors. This makes it particularly advantageous for modeling bioinspired continuum structures like octopus arms, where both flexibility and responsiveness are vital.

2. Problem Formulation

In this section, the mathematical model used to simulate the octopus arm will be discussed briefly, the detailed mathematics behind the simulation will be explained.

A. Cosserat Rod Theory (3). The Cosserat rod is the generalized version of the Kirchhoff rod (4), the key assumption to apply the theory is $L \gg r$, and for the Kirchhoff rod model, it incorporates the constraint that the rod experiences neither axial stretching nor compression, and excludes shear deformation. The soft body is treated as filament which can be formulated as a slender cylindrical structures deforming in three dimensions, therefore, the filament can be simplified to a one-dimensional geometric model, with its dynamic response approximated by integrating the balance laws over each cross-sectional area.

For a rod described by a centerline $\mathbf{r}(s,t)$ (where $s \in [0, L]$ is the arc-length of the rod and t is time) we begin by defining two reference frames within which we can express a vector $\mathbf{x}$:

$$\text{The laboratory (Eulerian) frame:} \quad \bar{\mathbf{x}} = x_1\mathbf{i} + x_2\mathbf{j} + x_3\mathbf{k} \qquad [1]$$

$$\text{The local (Lagrangian) frame:} \quad \mathbf{x} = x_1\mathbf{d}_1 + x_2\mathbf{d}_2 + x_3\mathbf{d}_3 \qquad [2]$$

where $\{\mathbf{i}, \mathbf{j}, \mathbf{k}\}$ is the laboratory canonical basis, $\{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3\}$ is the body-convected director basis.

The below equations from Eq. (3) to Eq. (6) can fully describe the kinematics and dynamics of the Kirchhoff rod (1):

$$\frac{\partial \mathbf{r}}{\partial t} = \mathbf{v} \qquad [3]$$

$$\frac{\partial \mathbf{d}_j}{\partial t} = \left(Q^T \boldsymbol{\omega}_{\mathcal{L}}\right) \times \mathbf{d}_j, \quad j = 1, 2, 3 \qquad [4]$$

$$\frac{\partial \left(\rho A\, \mathbf{v}\right)}{\partial t} = \frac{\partial \left(Q^T \mathbf{n}_{\mathcal{L}}\right)}{\partial s} + \mathbf{f} \qquad [5]$$

$$\frac{\partial (\rho I\, \boldsymbol{\omega}_{\mathcal{L}})}{\partial t} = \frac{\partial \boldsymbol{\tau}_{\mathcal{L}}}{\partial s} + \boldsymbol{\kappa}_{\mathcal{L}} \times \boldsymbol{\tau}_{\mathcal{L}} + Q\,\frac{\partial \mathbf{r}}{\partial s} \times \mathbf{n}_{\mathcal{L}} + (\rho I\, \boldsymbol{\omega}_{\mathcal{L}}) \times \boldsymbol{\omega}_{\mathcal{L}} + \mathbf{c}_{\mathcal{L}} \qquad [6]$$

where $\mathbf{Q}$ is the rotation matrix between two reference frames Eq. (1) and Eq. (2), ρ is the constant material density, A is the cross-sectional area, $\mathbf{v}$ is the velocity, $\mathbf{n}_{\mathcal{L}}$ and $\boldsymbol{\tau}_{\mathcal{L}}$ are, respectively, the internal force and couple resultants, $\mathbf{f}$ and $\mathbf{c}$ are external body-force and torque line densities, and the tensor $\mathbf{I}$ is the second area moment of inertia. This full set of equations establish the complete kinematic and dynamic framework for a Kirchhoff rod, firstly relate the time evolution of the centerline and local director frame to the rod's linear and angular velocities, and then enforce conservation of linear and angular momentum by balancing inertial contributions against spatial gradients of internal forces and moments as well as external forces and torques.

The Cosserat theory expands the Kirchhoff's method into a more general case where the stretch and shear are considered. Every cross section of the soft rod or filament is expected to take transverse shear and axial strains in addition to bending and twisting. To expand the Kirchhoff theory, three more constitutive laws must be added to define the stress resultants. The shear and axial deformations arise from the mismatch between the normal vector to the cross section and the tangent direction of the centerline. This is done by introducing a local stretch factor that adjusts the filament's length, cross-sectional area, moment of inertia, and bending and shear stiffness according to how much it has elongated or sheared. Linear constitutive relations then tie the resulting internal force and moment to these deformations, effectively adding three new stiffness, strain laws on top of the existing bending and twisting behavior. This unified framework captures all six deformation modes: bending, twisting, shear and stretch in a form that is straightforward to discretize and simulate numerically, enabling robust modeling of complex soft filament dynamics (1). Eventually, the kinematics and dynamics equations are given as:

$$\frac{\partial \mathbf{r}}{\partial t} = \mathbf{v} \qquad [7]$$

$$\frac{\partial \mathbf{d}_j}{\partial t} = \left(Q^T \boldsymbol{\omega}_{\mathcal{L}}\right) \times \mathbf{d}_j, \quad j = 1, 2, 3 \qquad [8]$$

$$dm\,\frac{\partial^2 \mathbf{r}}{\partial t^2} = \underbrace{\frac{\partial}{\partial \hat{s}}\left(\frac{Q^T \hat{\mathbf{S}}\,\sigma_{\mathcal{L}}}{e}\right) d\hat{s}}_{\text{shear/stretch internal force}} + \underbrace{\mathbf{F}}_{\text{external force}} \qquad [9]$$

$$\frac{d\hat{J}}{e}\,\frac{\partial \boldsymbol{\omega}_{\mathcal{L}}}{\partial t} = \underbrace{\frac{\partial}{\partial \hat{s}}\left(\frac{\hat{B}\,\hat{\boldsymbol{\kappa}}_{\mathcal{L}}}{e^3}\right) d\hat{s} + \frac{\hat{\boldsymbol{\kappa}}_{\mathcal{L}} \times \hat{B}\,\hat{\boldsymbol{\kappa}}_{\mathcal{L}}}{e^3}\, d\hat{s}}_{\text{bend/twist internal couple}} + \underbrace{\left(Q\,\mathbf{t} \times \hat{\mathbf{S}}\,\sigma_{\mathcal{L}}\right) d\hat{s}}_{\text{shear/stretch internal couple}} + \underbrace{\left(\frac{d\hat{J}\,\boldsymbol{\omega}_{\mathcal{L}}}{e} \times \boldsymbol{\omega}_{\mathcal{L}}\right)}_{\text{Lagrangian transport}} + \underbrace{\frac{d\hat{J}\,\boldsymbol{\omega}_{\mathcal{L}}}{e^2}\,\frac{\partial e}{\partial t}}_{\text{unsteady dilatation}} + \underbrace{\mathbf{C}_{\mathcal{L}}}_{\text{external couple}} \qquad [10]$$

where $dm = \rho\hat{A}\,d\hat{s} = \rho A\,ds$ is the infinitesimal mass element, and $d\hat{J} = \rho\hat{I}\,d\hat{s}$ is the infinitesimal mass second moment of inertia. By providing the appropriate initial and boundary conditions, Eq. (7) to Eq. (10) can express the precise kinematics and dynamics of a configured soft rod from its initial state, these equations represent **Linear Velocity**, **Angular Velocity**, **Linear Momentum** and **Angular Momentum** respectively. Solving these equations allows one to model the dynamics of a single Cosserat rod, since there is not always an analytical solution due to the complex dynamics, the numerical methods are used and will be discussed in the following section. The multiple rods cases will not be covered in this project as it does not align with the task requirements which will be a much more complicated scenario to simulate.

B. Numerical Solutions of the Cosserat rod Dynamics. Since the kinematics and dynamics of the single Cosserat rod are provided, to model any filament as desire, there are number of parameters need to be defined before solving the equations. A complete simulation requires the following parameters:

The **geometrical parameters** of the rod include:

- Number of elements: N
- Rest length of each element: Δs
- Radius of cross-section: r
- Cross-sectional area: $A = \dfrac{\hat{A}}{e}$
- Second area moment of inertia: $\mathbf{I} = \dfrac{\hat{\mathbf{I}}}{e^2}$

The **material parameters** that define the rod's stiffness and density are:

- Mass per unit length: ρ
- Young's modulus: E
- Shear modulus: G
- Bending stiffness matrix: $\mathbf{B} = \dfrac{\hat{\mathbf{B}}}{e^2}$
- Shearing stiffness matrix: $\mathbf{S} = \dfrac{\hat{\mathbf{S}}}{e}$

The **initial conditions** necessary to describe the rod's configuration at time $t = 0$ include:

- Initial position vector: $\mathbf{r}_0(s)$
- Initial velocity vector: $\mathbf{v}_0(s)$
- Initial director frame: $\{\mathbf{d}_1(s), \mathbf{d}_2(s), \mathbf{d}_3(s)\}$

The **external forces and constraints** that act on the rod are given by:

- Gravitational acceleration: $\mathbf{g}$
- External body forces: $\mathbf{f}_{\text{ext}}(s,t)$
- Damping coefficients or drag terms
- Boundary conditions (e.g., fixed, prescribed motion, free)

To solve the Cosserat rod governing equations, there are three steps in sequences: spatial discretizations, time discretization and implementation of the boundary conditions and interaction forces. Details will be given the subsequent sections

B.1. Spatial Discretization. The quantities in the Cosserat model are associated with nodes and with segments. The nodes are the discretized points along the centre line of the rod where the segments are the element connecting two adjacent nodes, they represent the finite portions of the rod and are used to calculate.

For a node, it is associated with the position vector $\mathbf{r}_i$, the velocity $\mathbf{v}_i = \partial\mathbf{r}_i/\partial t$, the pointwise mass m_i, and any external forces applied to it $\bar{\mathbf{f}}_i$. For a segment, it is associated with both geometric and dynamical parameters. Geometrically, a segment is defined by its edge vector $\boldsymbol{\ell}_j = \mathbf{r}_{j+1} - \mathbf{r}_j$, its current length $\ell_j = |\boldsymbol{\ell}_j|$, reference length $\hat{\ell}_j$, stretch ratio $e_j = \ell_j/\hat{\ell}_j$, and unit tangent vector $\mathbf{t}_j = \boldsymbol{\ell}_j/\ell_j$. The local orientation of each segment is described using a rotation matrix $\mathbf{Q}_j = \{\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3\}$. Dynamically, each segment is assigned a shear/axial strain vector $\boldsymbol{\sigma}_j = \mathbf{Q}_j(e_j\mathbf{t}_j) - \mathbf{d}_{3,j}$, an angular velocity $\boldsymbol{\omega}_j$, a reference cross-sectional area $\hat{A}_j$, a mass second moment of inertia $\hat{\mathbf{J}}_j$, a bend/twist stiffness matrix $\hat{\mathbf{B}}_j$, and a shear/strain stiffness matrix $\hat{\mathbf{S}}_j$. Additionally, any external couples applied to the segment are represented by $\mathbf{c}_j$.

In the continuum case, all physical quantities are defined point-wise. However, due to the discretization used in numerical simulations, it is often more appropriate to represent quantities as integrated values over a small domain along the rod. By dividing the integrated quantity by the length of the domain, we obtain a point-wise average that approximates the value continuum cases.

To make the kinematics and dynamics of Cosserat rod discretized, the rod is divided into a series of discrete nodes and segments, the equations from Eq. (7) to Eq. (10) are indexed by

$$\begin{aligned}
&i \in [1, n+1] &&\text{for Eq. (7) and Eq. (9)}\\
&i \in [1, n] &&\text{for Eq. (8) and Eq. (10)}\\
&j = 1, 2, 3 &&\text{for Eq. (8)}
\end{aligned}$$

Now the equations are discretized spatially, the next step is to implement a time stepper to evolve the equations. The purpose of the time stepper is to numerically integrated along the time direction and gradually advance the state of the system as result, the system status at each time step is simulated.

B.2. Time Discretizations. The time stepper is a numerical integrator used to evolve the Cosserat rod equations forward in time. The method used is energy conserving, second-order Verlet scheme. The second-order position Verlet scheme proceeds in three stages: it begins with a half-step update of both translational and rotational coordinates (Eq. (11) and Eq. (12)), then computes the local linear and angular accelerations (Eq. (13) and Eq. (14)), and finally applies a second half-step to update both translational and rotational coordinates once more (Eq. (15) and Eq. (16)).

$$\mathbf{r}_i\left(t + \tfrac{\delta t}{2}\right) = \mathbf{r}_i(t) + \tfrac{\delta t}{2}\,\mathbf{v}_i(t), \quad i = 1, \ldots, n+1 \qquad [11]$$

$$\mathbf{Q}_i\left(t + \tfrac{\delta t}{2}\right) = \exp\left[\tfrac{\delta t}{2}\,\boldsymbol{\omega}^i_{\mathcal{L}}(t)\right]\mathbf{Q}_i(t), \quad i = 1, \ldots, n \qquad [12]$$

$$\mathbf{v}_i(t + \delta t) = \mathbf{v}_i(t) + \delta t\,\frac{d\mathbf{v}_i}{dt}\left(t + \tfrac{\delta t}{2}\right), \quad i = 1, \ldots, n+1 \qquad [13]$$

$$\boldsymbol{\omega}^i_{\mathcal{L}}(t + \delta t) = \boldsymbol{\omega}^i_{\mathcal{L}}(t) + \delta t\,\frac{d\boldsymbol{\omega}^i_{\mathcal{L}}}{dt}\left(t + \tfrac{\delta t}{2}\right), \quad i = 1, \ldots, n \qquad [14]$$

$$\mathbf{r}_i(t + \delta t) = \mathbf{r}_i\left(t + \tfrac{\delta t}{2}\right) + \tfrac{\delta t}{2}\,\mathbf{v}_i\left(t + \tfrac{\delta t}{2}\right), \quad i = 1, \ldots, n+1 \qquad [15]$$

$$\mathbf{Q}_i(t + \delta t) = \exp\left[\tfrac{\delta t}{2}\,\boldsymbol{\omega}^i_{\mathcal{L}}\left(t + \tfrac{\delta t}{2}\right)\right]\mathbf{Q}_i\left(t + \tfrac{\delta t}{2}\right), \quad i = 1, \ldots, n \qquad [16]$$

The figure below illustrates the spatial discretizations of the Cosserat rod.

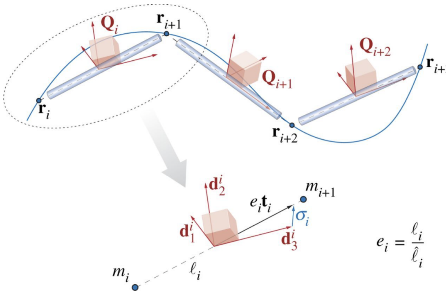


Fig. 1. Illustration of the Spatial Discretizations for the Cosserat Rod (1).

As shown in Fig. 1, a flexible filament is modeled as a series of time-dependent points, accompanied by material frames associated with each connecting segment. Each material frame consists of three mutually perpendicular directions that define the local orientation of the segment.

The line segment formed between any two adjacent points is assigned a length and a direction representing its tangent. The stretch of this segment is characterized by the ratio between its current length and its original, unstressed length. Based on this stretch, a deformation measure is introduced to capture both axial and shear strains relative to the segment's material frame.

The filament's mass is not distributed continuously, but rather concentrated at discrete points located at the vertices. These point masses govern translational motion. To account for rotational dynamics, each segment is also assigned a second moment of inertia, which reflects the rotational resistance of a cylindrical element conceptually representing the segment's geometry.

3. Simulation Procedures

In this section, the simulation pipelines using python will be explained in details, various physical phenomena have been characterized in PyElastica, such as internal energy dissipation, muscle-generated forces, self-contact events, interactions with rigid bodies, static and kinetic friction, and hydrodynamic effects. The octopus arm simulation is built based on PyElastica repository that models the arm as a Cosserat rod and applies both active control and passive hydrodynamics on it. First, a *Flexible_Octopus_Arm_Simulator* class is instantiated by combining *Elastica's Base_system_Collection* with constraint, forcing, and callback mixins which allows to record and store simulation data. Two specialized forcing modules are then defined: *Scheduled_PID_Control_Forces*, which applies full three dimensional PID control at the distal tip and interpolated Y-axis PID forces at intermediate nodes to simulate bionic kinematics like a real octopus arm movements. And Underwater Hydro Forces, which computes segment-wise anisotropic quadratic drag plus buoyancy and weight based on local velocities and fluid density. In the setup phase, a straight Cosserat rod is created with specified element count, geometry, and material properties, a hinge boundary condition fixes the base for one end of the rod, and both forcing modules are applied to the rod. Diagnostics callbacks sample position, director, and internal stress data at regular intervals, storing them in a history dictionary. Finally, the simulator is finalized and integrated over time using a Position-Verlet scheme for the total duration determined by the PID schedule, and the resulting trajectory data is saved for later analysis or rendering. .

A. Basic Parameter Settings. The arm is modeled as a Cosserat rod, some common and critical parameters include geometry, material, discretizations, and simulation related settings are defined by:

- $n\_\mathrm{elem} = 100$ (number of nodes on rod)
- $L = 2.3$ m (total rod length)
- $R_1 = 5 \times 10^{-2}$ m (base radius)
- $R_2 = 7 \times 10^{-3}$ m (tip radius)
- $\rho = 1100$ kg/m^3 (rod material density)
- $E = 2 \times 10^4$ Pa (Young's modulus)
- $G = E/(1 + \nu)$ with $\nu = 0.5$ (shear modulus)

To build a soft arm that is identical to real octopus arm. the rod's cross section is made to taper smoothly by first specifying a larger radius at its base and a much smaller radius at its tip. A one-dimensional array of nodal radius is then generated so that each entry interpolates linearly from the base value down to the tip value. To assign a single radius to each finite element, the radius values of the two nodes that define that element are averaged. This array of averaged radius is passed to the CosseratRod constructor, ensuring that the simulated rod gradually narrows along its length. During each callback, these element-wise radius are recorded so that any subsequent visualization accurately reflects the rod's continuously varying geometry. To make the arm feasible to finish the task, it is assigned a length of 2.3 meters which provides an extra redundancy.

B. Boundary Constraint. A hinge-like condition is imposed at the base of the rod to fix its position and eliminate base motion. Specifically, the base node maintains a constant position $\mathbf{r}_0$ and zero velocity throughout the simulation. This ensures the arm pivots about the root without translation.

C. Underwater Environment. In the simulation, the underwater environment is modeled by combining anisotropic quadratic drag with buoyancy and weight effects on each Cosserat rod element. Denoting the fluid density by ρ_f, the normal and tangential drag coefficients by C_n and C_t, and the gravitational acceleration by g, each element e of length Δs_e and radius r_e has volume:

$$V_e = \pi\, r_e^2\, \Delta s_e,$$

and projected area:

$$A_e = 2\, r_e\, \Delta s_e.$$

The local segment velocity $\mathbf{v}_e$ is decomposed into components parallel and perpendicular to the unit tangent $\hat{\mathbf{t}}_e$:

$$\mathbf{v}_\parallel = (\mathbf{v}_e \cdot \hat{\mathbf{t}}_e)\, \hat{\mathbf{t}}_e, \quad \mathbf{v}_\perp = \mathbf{v}_e - \mathbf{v}_\parallel.$$

The corresponding drag forces are:

$$\mathbf{F}_{\parallel,e} = -\tfrac{1}{2}\, \rho_f\, C_t\, A_e\, \|\mathbf{v}_\parallel\|\, \mathbf{v}_\parallel, \quad \mathbf{F}_{\perp,e} = -\tfrac{1}{2}\, \rho_f\, C_n\, A_e\, \|\mathbf{v}_\perp\|\, \mathbf{v}_\perp$$

so that the total drag is $\mathbf{F}_{d,e} = \mathbf{F}_{\parallel,e} + \mathbf{F}_{\perp,e}$. Buoyancy and net weight from the rod density ρ_{rod} produce a vertical force:

$$\mathbf{F}_{b,e} = \big(\rho_f - \rho_{\mathrm{rod}}\big)\, V_e\, g\, \mathbf{e}_y.$$

Finally, each element's forces are evenly distributed to its two end nodes, ensuring conservation of momentum along the rod.

This formulation captures both the directional drag and buoyant support experienced by a slender flexible body in a viscous fluid. The underwater environment is modeled under several simplifying assumptions: the fluid is treated as an incompressible, homogeneous medium with constant density ρ_f, hydrodynamic forces on each rod element are represented solely by anisotropic quadratic drag proportional to the square of parallel and perpendicular velocity components, without accounting for viscous shear effects, turbulence-induced additional drag, or added mass, the buoyant force is computed as the net weight difference $(\rho_f - \rho_{\text{rod}})V_e g$ and is assumed to act uniformly and instantaneously over each element's volume, element radius are averaged when calculating projected area and volume, neglecting fine-scale variations along the segment. Finally, fluid–structure interaction is limited to these drag and buoyancy terms, omitting fluid elasticity or flow coupling. These assumptions enable efficient computation while capturing the primary drag and buoyant support experienced by a slender, flexible body in a viscous fluid.

D. Control Strategy. The controller is a time-scheduled PID scheme that applies full 3D PID control to the end effector and one-dimensional PID control in the y-direction to intermediate nodes every five elements. Let:

$$\begin{aligned} P &= \text{number of phases} \\ T_s &= \text{segment time} \\ C &= \text{number of cycles} \end{aligned}$$

so that the total duration is:

$$T_{\text{total}} = P\,T_s\,C.$$

At time t, define the phase index:

$$p = \left\lfloor \frac{(t \bmod (P\,T_s))}{T_s} \right\rfloor, \quad \mathbf{x}_{\text{tgt}}(p) \in \mathbb{R}^3$$

as the current target position. **End-Effector PID Control**, denote the end-effector position and velocity by:

$$\mathbf{x}(t) = \mathbf{x}_{\text{end}}(t), \quad \dot{\mathbf{x}}(t) = \mathbf{v}_{\text{end}}(t).$$

Define the position error:

$$\mathbf{e}(t) = \mathbf{x}_{\text{tgt}}(p) - \mathbf{x}(t).$$

The integral and derivative terms evolve as

$$\begin{aligned} \mathbf{I}(t) &= \int_0^t \mathbf{e}(\tau)\,\mathrm{d}\tau, \\ \mathbf{D}(t) &= \dot{\mathbf{x}}(t). \end{aligned}$$

The PID force is then:

$$\mathbf{F}_{\text{end}}(t) = K_p\,\mathbf{e}(t) + K_i\,\mathbf{I}(t) - K_d\,\mathbf{D}(t),$$

with clipping to a specified maximum $\mathbf{F}_{\max}$ component wise. **Intermediate-Node y-Directional PID** is introduced to provide a octopus-like movements, for each rod node indexed by $i = 0, 1, \ldots, N$, only those with $i \equiv 0 \pmod 5$ are controlled in the y-direction. Let:

$$y_i(t) = \text{position of node } i \text{ in } y, \quad v_i(t) = \dot{y}_i(t),$$

and define the interpolation parameter:

$$\alpha_i = \frac{i}{N},$$

so that the desired y-position is:

$$y_{\text{des}}^{(i)}(t) = (1 - \alpha_i)\,y_{\text{base}} + \alpha_i\,[\mathbf{x}_{\text{tgt}}(p)]_y.$$

The error, integral, and derivative are:

$$\begin{aligned} e_i(t) &= y_{\text{des}}^{(i)}(t) - y_i(t), \\ I_i(t) &= \int_0^t e_i(\tau)\,\mathrm{d}\tau, \\ D_i(t) &= v_i(t). \end{aligned}$$

Thus the applied force in y at node i is:

$$F_{y,i}(t) = K_p\,e_i(t) + K_i\,I_i(t) - K_d\,D_i(t),$$

again clipped to the scalar limit $F_{y,\max}$. The scheduling Logic is given as following:

- Compute $p = \lfloor (t \bmod PT_s)/T_s \rfloor$, select target $\mathbf{x}_{\text{tgt}}(p)$.
- Apply $\mathbf{F}_{\text{end}}$ to the last node in all three directions.
- For $i = 0, \ldots, N$, if $i \bmod 5 = 0$, compute $F_{y,i}$ and apply in the y-direction.
- If $t > T_{\text{total}}$, no forces are applied.

In terms of the parameter settings, the end-effector PID gains are $K_p = 5$, $K_i = 0.05$, and $K_d = 4.0$, the segment time is $T_s = 4$ s, the number of phases is $P = 2$ (waypoints $(2.0, 0.5, 0)$ and $(0.5, 0.5, 0)$), and the number of cycles is $C = 2$, yielding a total duration $T_{\text{total}} = 16$ s, the end-effector force is clipped component-wise to $\pm(4.0, 4.0, 4.0)$ N, and the same PID gains with a ± 4.0 N limit are applied in the y-direction to every fifth intermediate node.

This scheme ensures that the end effector tracks a sequence of waypoints in full 3D, while intermediate nodes smoothly transition their y-positions via PID interpolation.

E. Time Stepper. In the simulation, the integrator loop proceeds by first advancing the velocity by half of `dt` using the current acceleration, then updating the positions by a full `dt` with this intermediate velocity. After moving the system forward, all forces are recomputed at the new positions to obtain an updated acceleration, and the velocity is finally advanced by the remaining half of `dt`. This sequence repeats for `N` steps, providing a time-reversible and energy-conserving update of the rod's dynamics. The input parameters are:

- $\Delta t = 0.0007 \times \dfrac{L}{n_{\text{elem}}} = 0.0007 \times \dfrac{2.3}{100} \approx 1.61 \times 10^{-5}\,\text{s}$
- T = segment_time × phases × cycles $= 4 \times 2 \times 3 = 16\,\text{s}$
- $N = \left\lceil \dfrac{T}{\Delta t} \right\rceil = \left\lceil \dfrac{24}{1.61 \times 10^{-5}} \right\rceil = 993\,790$
- step_skip $= \max\bigl(1, \lfloor N/200 \rfloor\bigr) = \left\lfloor \dfrac{1\,490\,684}{200} \right\rfloor = 4\,968$

By doing so, it delivers time-reversibility, excellent energy conservation, and long-term numerical stability, ensuring accurate simulation of the flexible rod in a complex underwater environment.

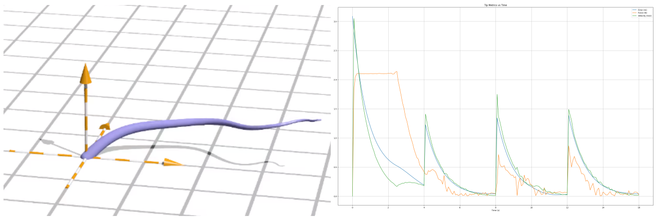
4. Results and Evaluations

In this section, the simulation results will be discussed. In the visualization stage, the time-resolved data $\mathbf{x}(t)$ and $r(t)$ from the simulation are loaded, smoothly interpolate both to match the frame rate f, convert the geometry at each time t into a continuous swept surface $\mathcal{S}(t)$, place multiple virtual cameras and lights around the scene, render each viewpoint into an image sequence using ray tracing, and finally stitch those sequences into a cohesive video capturing the soft arm's dynamic motion. Two videos of simulation are created in top view and side view respectively.

A. Qualitative and Quantitative Results. In the simulation, the array time stores the timestamp of each sample point (in seconds) for alignment with other variables. The tensor $\mathbf{force\_end}$ represents the three-dimensional force vector (in newtons) applied to the end effector at each time step, with its norm reflecting the magnitude of the applied force. The tensor $\mathbf{error\_end}$ denotes the three-dimensional error vector (in meters) between the actual end effector position and the target position, whose norm measures the positional deviation. The tensor $\mathbf{tip\_velocity}$ gives the three-dimensional velocity vector (in meters per second) of the end effector, with its norm describing the rate of motion. We define the sequences $\mathrm{force\_norm}$, $\mathrm{err\_norm}$, and $\mathrm{vel\_norm}$ as the Euclidean norms of these vectors, and compute their Root Mean Square Error (RMSE), maximum, and minimum values to quantitatively assess the force levels, positioning accuracy, and motion speed of the end effector, visualizing these metrics over time using line plots. The computed results are:

$$\begin{aligned}
&\text{End-Force Norm: } \mathrm{RMSE} = 0.9258\,\mathrm{N}, \quad \mathrm{Max} = 2.1416\,\mathrm{N}, \\
&\quad \mathrm{Min} = 0.0000\,\mathrm{N} \\
&\text{Tip-Error Norm: } \mathrm{RMSE} = 0.7140\,\mathrm{m}, \quad \mathrm{Max} = 3.0887\,\mathrm{m}, \\
&\quad \mathrm{Min} = 0.0263\,\mathrm{m} \\
&\text{Tip-Velocity Norm: } \mathrm{RMSE} = 0.0263\,\mathrm{m/s}, \quad \mathrm{Max} = 3.0506\,\mathrm{m/s}, \\
&\quad \mathrm{Min} = 0.0000\,\mathrm{m/s}
\end{aligned}$$

The simulation environment and the plot are given in the Fig. 2, each grid in the rendered virtual environment represents 0.5 meters in distance, the animation illustrates the simulated arm can perform a perfect reaching movements. In Tip metrics against time graph, the horizontal axis represents simulation time (seconds) and the vertical axis shows the Euclidean norms of three quantities: the end-effector position error $\|\mathbf{error\_end}\|$ (m), the applied force $\|\mathbf{force\_end}\|$ (N), and the tip velocity $\|\mathbf{tip\_velocity}\|$ (m/s). Plotting these norms together allows one to evaluate the controller's positioning accuracy, force levels, and motion speed simultaneously over the course of the simulation. As the trajectories show, each reaching maneuver exhibits a pronounced initial peak in tip error, velocity, and control force, followed by an approximately exponential decay toward zero. During the first reach (around $t \approx 0.3$ s), the tip error peaks at ~ 3.1 m, the velocity at ~ 3.0 m/s, and the control force remains near 2.1 N before declining; it takes approximately 4 s for the error and velocity to fall below 0.1 m and 0.05 m/s, respectively. For the subsequent reaches at $t \approx 4$ s, 8 s, and 12 s, the peak error and velocity stabilize in the ranges 1.2–1.4 m and 1.2–1.8 m/s, with control force peaks of 0.4–0.9 N and noticeably faster decay. Overall, the controller demonstrates consistent rapid multiple reaching tasks within 20 seconds, indicates a robust performance.



(a) Simulation Environment

(b) Tip Metrics against Time

Fig. 2. Simulation Illustration

5. Conclusion

The present simulation successfully completed multiple reaching movements within a total elapsed time of 16 seconds, guiding the octopus-arm end effector sequentially between target distances of 0.5 m and 2.0 m from the base. The time-scheduled PID controller delivered accurate full-dimensional control at the tip while enforcing one-dimensional Y–axis interpolation control at intermediate nodes every five elements. Meanwhile, the anisotropic hydrodynamic model incorporating quadratic drag based on a linearly tapered radius profile and buoyancy effects—yielded stable underwater dynamics throughout the maneuver. Nevertheless, several limitations emerged. First, the use of fixed PID gains may not generalize across varying trajectories or in the presence of external disturbances, potentially resulting in tracking errors under more demanding scenarios. Second, the hydrodynamic formulation omits added-mass and vortex–induced forces, limiting fidelity for high-speed motions. Third, the uniform spatial discretizations and constant time step, while sufficient for 20 seconds of motion, may hinder scalability to longer simulations or finer deformation details and impose significant computational cost.

Future work will address these shortcomings by integrating adaptive control schemes (e.g., gain scheduling or model-based tuning) to enhance robustness, extending the fluid–structure interaction model to include added mass and wake effects, and employing non-uniform mesh refinement or higher-order integration to improve accuracy and efficiency. Additionally, exploring a broader set of target trajectories, environmental disturbances, and real-time parameter adaptation will further validate and broaden the applicability of this octopus arm reaching framework.

Acknowledgments

I would like to express my sincere gratitude to Professor Gianmarco Mengaldo and his teaching assistant Wang Zhan for delivering an exceptionally inspiring course and for their dedicated hard work. Their efforts in creating engaging lectures, detailed explanations, and continuous support have significantly enhanced our learning experience and deepened our understanding of the subject.

1 M Gazzola, LH Dudte, AG McCormick, L Mahadevan, Forward and inverse problems in the mechanics of soft filaments. *Royal Soc. Open Sci.* **5**, 171628 (2018).
2 X Zhang, FK Chan, T Parthasarathy, M Gazzola, Modeling and simulation of complex dynamic musculoskeletal architectures. *Nat. Commun.* **10**, 4825 (2019).
3 E Cosserat, F Cosserat, *Theorie des corps deformables.* (Cornell University Library, Ithaca, NY), (1909).
4 G Kirchhoff, Ueber das gleichgewicht und die bewegung eines unendlich dünnen elastischen stabes. *J. für die Reine und Angewandte Math.* **56**, 285–313 (1859).