Computational Mechanics
https://doi.org/10.1007/s00466-022-02247-x

ORIGINAL PAPER

A monolithic optimal control method for displacement tracking of Cosserat rod with application to reconstruction of *C. elegans* locomotion

Yongxing Wang[1] · Thomas Ranner[1] · Thomas P. Ilett[1] · Yan Xia[2] · Netta Cohen[1]

Received: 29 August 2022 / Accepted: 1 November 2022

Abstract
This article considers an inverse problem for a Cosserat rod where we are given only the position of the centreline of the rod and must solve for external forces and torques as well as the orientation of the cross sections of the centreline. We formulate the inverse problem as an optimal control problem using the position of the centreline as an objective function with the external force and torque as control variables, with meaningful regularisation of the orientations. A monolithic, implicit numerical scheme is proposed in the sense that primal and adjoint equations are solved in a fully-coupled manner and all the nonlinear coefficients of the governing partial differential equations are updated to the current state variables. The forward formulation, determining rod configuration from external forces and torques, is first validated by a numerical benchmark; the solvability and stability of the inverse problem are then tested using data from forward simulations. The proposed optimal control method is motivated by reconstruction of the orientations of a rod's cross sections, with its centreline being captured through imaging protocols. As a case study, we take the locomotion of the nematode, *Caenorhabditis elegans*. In this study we take laboratory data for its centreline and infer its cross-section orientation (muscle locations) with the control force and torque being interpreted as the reaction force, activated by *C. elegans*' muscles, from the surrounding fluids. This method thus combines the mathematical modelling and laboratory data to study the locomotion of *C. elegans*, which gives us insights into the potential anatomical orientation of the worm beyond what can be observed through the laboratory data. The paper is completed with several additional remarks explaining the theoretical and technical details of the model.

Keywords Optimal control · Cosserat rod · Finite element · *C. elegans* · Locomotion

1 Introduction

The beam and rod theories have been developed to model a typical three dimensional solid structure which is much longer in one dimension than the other two dimensions. The classical Euler-Bernoulli beam theory considers the extension and compression of a rod or beam and allows loads of stretching, compressing or bending [6,24], which is generally suitable for modelling a thin rod with small deformation. The Timoshenko-Ehrenfest beam theory was developed to take into account shear deformation induced by rotational bending effects, making it suitable for modelling thick beams with larger deformation [19,23]. The Kirchhoff-Love and Cosserat rod theories were developed to model rods with finite deformation, with the former allowing bending and twisting while ignoring stretching, compressing and shearing deformation [21,56]; and the latter allowing all types of loads and deformation [2,18,29,74,75]. This paper is to for-

✉ Yongxing Wang
scsywan@leeds.ac.uk

Thomas Ranner
t.ranner@leeds.ac.uk

Thomas P. Ilett
t.ilett@leeds.ac.uk

Yan Xia
y.xia@leeds.ac.uk

Netta Cohen
n.cohen@leeds.ac.uk

[1] School of Computing, University of Leeds, Leeds, UK

[2] Centre for Computational Imaging and Simulation Technologies in Biomedicine (CISTI), University of Leeds, Leeds, UK

Published online: 14 November 2022

mulate an optimal control problem based upon the special Cosserat theory of rods. Cosserat rod theory is geometrically exact for modelling bending and torsion as well as extension and shear, which is considered as a geometrically nonlinear generalisation of the Timoshenko-Ehrenfest beam (while Kirchhoff-Love is a geometrically nonlineaer generalisation of Euler-Bernoulli beam) [2,74], and has been adopted to model the locomotion of *Caenorhabditis elegans* (*C. elegans*) in recent years [31,68].

Optimal control is a branch of mathematical optimisation which seeks to optimise an objective function of a dynamical system, usually described by partial differential equations, through controlling meaningful variables of the system [78]. Optimal control theory has a wide range of applications from classical control of solid structures [45,60,64,77,84] to optimal flow control [33,43,44,61,65] including recent control formulation for fluid-structure interaction systems [13,14,62, 80–82]. The objective can be a desired deformed configuration of an elastic solid controlled by a set of load parameters [45], drag force reduction of a flow system by shape optimisation [33,61,65] or active turbulence control at the boundary layer [15,22,46,50,59,61]; it could also be velocity tracking by controlling a body force [3,36,38,40,43,44,55,57,58,66] or boundary force [3,4,26,28,37,38,41]; the objective may also be reducing vorticity [1,3,66] or matching a turbulence kinetic energy [3,57,58]. Velocity-tracking type of optimal control has a rigorous mathematical theory for its solution existence [1,26,37,39] and stability of its numerical algorithm [39,41,43,44].

In the context of optimal control of a rod or beam, the solution existence of optimal control of the longitudinal vibration of a viscoelastic rod by either a contact force or distributed force is discussed in [73], and the mean mechanical energy minimised by a boundary force is studied, using the methods of the calculus of variations [30], maximum principle [70] and Ritz method [51]; minimisation of the mean square deviation of the Timoshenko beam is investigated by controlling a distributed force [71] or by the angular acceleration [83], and singularity of its solution is discussed in [69]; optimal control of transverse vibration of Euler-Bernoulli beam is introduced in [79]. We will consider displacement tracking of the Cosserat rod in this paper, which, to the best of our knowledge, has not been studied before. Due to their long, thin nature, accurately capturing the rotation vectors of a rod is a challenging task whereas there are several well studied approaches for reconstructing the centreline [8,27,32]. To investigate the applicability of our method we will consider a case study on the nematode, *C. elegans*: We shall apply this optimal control formulation to the reconstruction of the locomotion of *C. elegans* based upon laboratory data [72].

Locomotion – the ability of an organism to move from one place to another – is achieved by animals through a variety of methods [9,16,35]. *C. elegans* is a transparent nematode of about 1 mm long [11,76], whose planar undulatory locomotion has been widely studied [17,48] by laboratory experiments [8,25,47,54] or mathematical modelling [31,63,68]. In its natural habitat, this nematode moves in three dimensional environments. However, such locomotion has only recently begun to be recorded [52,72]. Of particular interest is a recent modelling study of a roll manoeuvre modelled as a torsional turn [10]. One key challenge with interpretation of 3D video footage of locomotion is that the images (and hence the centreline reconstructions) lack information linking the body shape to the local, anatomically meaningful frame of the body (its left, right, ventral and dorsal directions (see Figure 4), and information about internal torsion or twist along the body).

In this paper, we propose a method to combine laboratory data of the motion of *C. elegans*' centreline and mathematical modelling to reconstruct the whole picture of *C. elegans* locomotion: how does the worm wriggle and wiggle locally through its body (how does its anatomical frame evolve)? The centreline data of *C. elegans* can be constructed using videos from three different perspectives[72]. However, it is challenging to construct the local frames (information about the internal torsion or twist along the body), which is the motivation to develop the proposed method in this paper. We point out that the proposed optimal control formulation is general and not limited to *C. elegans* but for simplicity we restrict the example formulation to neglect inertial terms [17] (consistent with *C. elegans* being a low Reynolds number swimmer).

The contributions of this paper are highlighted as follows: a monolithic optimal control method is developed based upon the special Cosserat theory of rods; an incompressibility condition is derived and integrated into the forward as well as the control problem; the formulation is implicit and the primal and adjoint equations are solved in a fully-coupled manner; this new optimal control method is applied to a challenging inverse problem: reconstruction of *C. elegans* locomotion based on its centreline from laboratory data; implementation in the open-source software package FreeFEM++, which is available on the public Github site.

The paper is organised as follows. The governing partial differential equations of the Cosserat rod are introduced in Section 2 with a focus on expressing these control equations in a closed component form. The optimisation problem with the corresponding primal and adjoint equations are derived in Section 3, followed by a monolithic optimal control formulation in Section 4. Numerical experiments are carried out in Section 5 to validate both the forward and the optimal control formulations, and the proposed optimal control method is applied to the reconstruction of *C. elegans* locomotion in Section 6. Finally, conclusions are drawn and future work are discussed in Section 7.

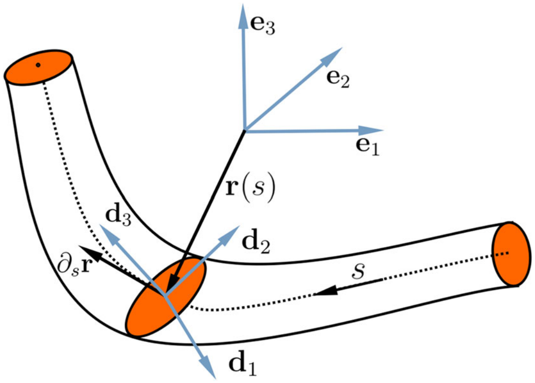


Fig. 1 A sketch of the Cosserat rod

2 Governing equations for the Cosserat rod

First, two coordinate systems or frames, as well as their relations, are introduced in order to describe the geometry of the Cosserat rod. Then, the mechanics of Cosserat rod is described by the conservation of linear momentum and angular momentum, and a set of constitutive equations is introduced to close the system. Finally, the governing equations are expressed in terms of six unknown variables: three components of the position vector (x, y, z) and three components of the rotation vector (α, β, γ). This formulation is based on the one presented in [12]: we derive the angular velocity and generalised curvature using a new method in Section 2.3 and rewrite all the control equations in a matrix-vector format; in addition, we consider dilation of the cross section of the rod by differentiation of the reference and current arc lengths and derivation of the incompressibility condition in Section 2.4.

2.1 Global and local frames

In order to describe all the types of deformation of the Cosserat rod, a global coordinate system $[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$ is first introduced as shown in Figure 1, which is assumed to form a fixed right-hand orthogonal unit basis (also called the fixed frame), to define the centreline of the rod by a three-dimensional curve: $\mathbf{r}(s, t) = x(s, t)\mathbf{e}_1 + y(s, t)\mathbf{e}_2 + z(s, t)\mathbf{e}_3$, where $s \in [a(t), b(t)]$ is the arc-length parameter of the curve and t denotes the time; a local coordinate system (the moving frame) $[\mathbf{d}_1(s, t), \mathbf{d}_2(s, t), \mathbf{d}_3(s, t)]$ (orthogonal unit basis) is also introduced everywhere at the centreline to describe the motion of the rod's cross section, and it is assumed that $\mathbf{d}_3(s, t)$ is always perpendicular (*not* necessarily coinciding with the tangential $\partial_s\mathbf{r}(s, t)$ of the centreline) to the cross section to facilitate the expressions of the moment of inertia and constitutive relations. This local coordinate system can be constructed by the following three successive rotations from the global coordinate system.

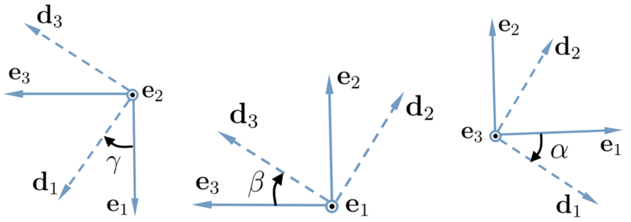


Fig. 2 Three rotations of the coordinate system

Remark 1 The arc length s is a current configuration, which is generally not a constant especially when considering the case of large extension or compression [31]. We introduce a reference or initial configuration $\tilde{s} = s_0 \in [a_0, b_0]$ to compute the strain (equations (9) and (10)), and consider $s = s(\tilde{s}, t)$ as a function of $\tilde{s}$ and time t. Let us also introduce the deformation scalar $j(\tilde{s}, t) = ds(\tilde{s}, t)/d\tilde{s}$ for the convenience of notation in the following sections.

Step 1: Rotate the $\mathbf{e}_1 - \mathbf{e}_3$ plane clockwise around the $\mathbf{e}_2$ axis by an angle γ, so that the $\mathbf{e}_3$ axis sits on the $\mathbf{d}_2 - \mathbf{d}_3$ plane as shown in Figure 2 (left), i.e.: perform a rotation operation $[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]\,\mathbf{R}_y^T(\gamma)$ with

$$\mathbf{R}_y = \begin{bmatrix} \cos\gamma & 0 & \sin\gamma \\ 0 & 1 & 0 \\ -\sin\gamma & 0 & \cos\gamma \end{bmatrix}.$$

Step 2: Rotate the $\mathbf{e}_2 - \mathbf{e}_3$ plane clockwise around the $\mathbf{e}_1$ axis by an angle β, so that the $\mathbf{e}_3$ axis overlaps with the $\mathbf{d}_3$ axis as shown in Figure 2 (middle), i.e.: perform another rotation operation $[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]\,\mathbf{R}_y^T(\gamma)\mathbf{R}_x^T(\beta)$ with

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix}.$$

Step 3: Rotate the $\mathbf{e}_1 - \mathbf{e}_2$ plane clockwise around $\mathbf{e}_3$ axis by an angle α, so that the $[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]$ overlaps with $[\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3]$ as shown in Figure 2 (right), i.e.: perform the final rotation operation $[\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]\,\mathbf{R}_y^T(\gamma)\mathbf{R}_x^T(\beta)\mathbf{R}_z^T(\alpha)$ with

$$\mathbf{R}_z = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The overall rotation matrix can be expressed as:

$$\mathbf{Q} = \mathbf{R}_z\mathbf{R}_x\mathbf{R}_y$$
$$= \begin{bmatrix} \cos\alpha\cos\gamma - \sin\alpha\sin\beta\sin\gamma & -\sin\alpha\cos\beta & \cos\alpha\sin\gamma + \sin\alpha\sin\beta\cos\gamma \\ \sin\alpha\cos\gamma + \cos\alpha\sin\beta\sin\gamma & \cos\alpha\cos\beta & \sin\alpha\sin\gamma - \cos\alpha\sin\beta\cos\gamma \\ -\cos\beta\sin\gamma & \sin\beta & \cos\beta\cos\gamma \end{bmatrix} \tag{1}$$

where all the three angles α, β and γ are functions of the arc length s and time t: $\alpha = \alpha(s, t), \beta = \beta(s, t)$ and $\gamma = \gamma(s, t)$. Therefore, the local coordinate system can be obtained by

$$[\mathbf{d}_1(s, t), \mathbf{d}_2(s, t), \mathbf{d}_3(s, t)] = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]\,\mathbf{Q}^T. \tag{2}$$

The components of any vector $\mathbf{v}$ in these two coordinates system have the following relations: if $\mathbf{v}$ is expanded in the global frame as $\mathbf{v} = \sum_{i=1}^{3} v_i^g \mathbf{e}_i$ and the local frame as $\mathbf{v} = \sum_{i=1}^{3} v_i^l \mathbf{d}_i$, then (noticing that $\mathbf{Q}$ is an orthogonal unit matrix)

$$\mathbf{v} = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3] \begin{pmatrix} v_1^g \\ v_2^g \\ v_3^g \end{pmatrix} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3]\,\mathbf{Q} \begin{pmatrix} v_1^g \\ v_2^g \\ v_3^g \end{pmatrix}, \tag{3}$$

which implies

$$\begin{pmatrix} v_1^l \\ v_2^l \\ v_3^l \end{pmatrix} = \mathbf{Q} \begin{pmatrix} v_1^g \\ v_2^g \\ v_3^g \end{pmatrix}. \tag{4}$$

In the rest of this article, we use the superscript 'g' to indicate the components of a vector expanded in the global frame and 'l' in the local frame.

2.2 Conservation laws

The governing equations of the Cosserat rod are based on the conservation of linear momentum and conservation of angular momentum as follows [2]:

$$\rho(s)A(s, t)\partial_{tt}\mathbf{r}(s, t) = \partial_s\mathbf{n}(s, t) + \mathbf{f}(s, t), \tag{5}$$
$$\partial_t\mathbf{h}(s, t) = \partial_s\mathbf{m}(s, t) + \partial_s\mathbf{r}(s, t) \times \mathbf{n}(s, t) + \mathbf{l}(s, t), \tag{6}$$

where $\mathbf{n}$ and $\mathbf{m}$ are the internal force and torque respectively, $\mathbf{f}$ and $\mathbf{l}$ are the external force and torque densities (per unit reference length) respectively, $\rho(s)$ and $A(s, t)$ are the density and area of the cross section respectively, and $\mathbf{h}$ is the angular momentum (per unit reference length).

In equations (5) and (6), it is convenient to express all the vectors in the local frame except $\mathbf{r}(s, t)$. Therefore, we first express these vectors in the local frame, and then transform them to the global frame using (4), which will finally be substituted into (5) and (6) in order to obtain an equation system in its component form.

The angular momentum $\mathbf{h} = \sum_{i=1}^{3} h_i^l \mathbf{d}_i$ can be expressed as:

$$\begin{pmatrix} h_1^l \\ h_2^l \\ h_3^l \end{pmatrix} = \mathbf{I}(s) \begin{pmatrix} \omega_1^l \\ \omega_2^l \\ \omega_3^l \end{pmatrix} = \begin{bmatrix} I_{11} & 0 & 0 \\ 0 & I_{22} & 0 \\ 0 & 0 & I_{33} \end{bmatrix} \begin{pmatrix} \omega_1^l \\ \omega_2^l \\ \omega_3^l \end{pmatrix} \tag{7}$$

with $\boldsymbol{\omega} = \sum_{i=1}^{3} \omega_i^l \mathbf{d}_i$ denoting the generalised angular velocity, and $\mathbf{I}(s)$ denoting the moment of inertia (per unit reference length). Let (ξ, η, ζ) denote the coordinates in the local frame, then $\mathbf{I}(s)$ can be computed as follows, noticing that $\mathbf{d}_3$ is perpendicular to the cross section:

$$I_{11} = \int_{A(s)} \rho\eta^2 d\xi d\eta, \quad I_{22} = \int_{A(s)} \rho\xi^2 d\xi d\eta,$$
$$I_{33} = \int_{A(s)} \rho\left(\xi^2 + \eta^2\right) d\xi d\eta. \tag{8}$$

In order to close the equation system (5) and (6), constitutive relations of $\mathbf{n}$ and $\mathbf{m}$ have to be established in terms of the unknown variables. In the local frame, we adopt a linear relation between the internal force $\mathbf{n}(s, t)$ and strain $\boldsymbol{\epsilon}(s, t)$, and linear relation between internal torque $\mathbf{m}$ and the curvature $\boldsymbol{\kappa}(s, t)$ [12,31] as follows. Let

$$\mathbf{n}(s, t) = \sum_{i=1}^{3} n_i^g(s, t)\mathbf{e}_i, \quad \boldsymbol{\epsilon}(s, t) := \partial_{\tilde{s}}\mathbf{r}(s, t)$$
$$= \sum_{i=1}^{3} \epsilon_i^g(s, t)\mathbf{e}_i, \tag{9}$$

and

$$\mathbf{n}(s, t) = \sum_{i=1}^{3} n_i^l(s, t)\mathbf{d}_i(s, t), \quad \boldsymbol{\epsilon}(s, t) := \partial_{\tilde{s}}\mathbf{r}(s, t)$$
$$= \sum_{i=1}^{3} \epsilon_i^l(s, t)\mathbf{d}_i(s, t), \tag{10}$$

a linear relation, in the local frame, between $\mathbf{n}(s, t)$ and $\boldsymbol{\epsilon}(s, t)$ can be expressed as

$$\begin{pmatrix} n_1^l \\ n_2^l \\ n_3^l \end{pmatrix} = \mathbf{K} \begin{pmatrix} \epsilon_1^l \\ \epsilon_2^l \\ \epsilon_3^l - 1 \end{pmatrix} = \begin{bmatrix} K_{11} & 0 & 0 \\ 0 & K_{22} & 0 \\ 0 & 0 & K_{33} \end{bmatrix} \begin{pmatrix} \epsilon_1^l \\ \epsilon_2^l \\ \epsilon_3^l - 1 \end{pmatrix}, \tag{11}$$

where

$$K_{11} = K_{22} = kGA(s,t), \quad K_{33} = EA(s,t). \tag{12}$$

E and G are the Young's and shear moduli respectively, k is a numerical factor depending on the shape of the cross section at s [34], and $A(s,t)$ is the area of the rod's cross section. We assume $A(s,t)$ is a function of space s and time t, and an incompressibility assumption will be used to determine $A(s,t)$ in Section 2.4. Using the transformation (4) between the local and global coordinates, (11) can be expressed as

$$\begin{aligned} \begin{pmatrix} n_1^g \\ n_2^g \\ n_3^g \end{pmatrix} &= \mathbf{Q}^T\mathbf{KQ} \begin{pmatrix} \epsilon_1^g \\ \epsilon_2^g \\ \epsilon_3^g \end{pmatrix} - \mathbf{Q}^T \begin{pmatrix} 0 \\ 0 \\ K_{33} \end{pmatrix} \\ &= j(\tilde{s},t)\mathbf{Q}^T\mathbf{KQ}\partial_s \begin{pmatrix} x(s,t) \\ y(s,t) \\ z(s,t) \end{pmatrix} - \mathbf{Q}^T \begin{pmatrix} 0 \\ 0 \\ K_{33} \end{pmatrix}. \end{aligned} \tag{13}$$

Similarly, let

$$\mathbf{m}(s,t) = \sum_{i=1}^{3} m_i^l(s,t)\mathbf{d}_i(s,t), \quad \boldsymbol{\kappa}(s,t) = \sum_{i=1}^{3} \kappa_i^l(s,t)\mathbf{d}_i(s,t). \tag{14}$$

Then, a linear relation, in the local frame, between $\mathbf{m}(s,t)$ and $\boldsymbol{\kappa}(s,t)$ can be expressed as

$$\begin{pmatrix} m_1^l \\ m_2^l \\ m_3^l \end{pmatrix} = \mathbf{J} \begin{pmatrix} \kappa_1^l \\ \kappa_2^l \\ \kappa_3^l \end{pmatrix} = \begin{bmatrix} J_{11} & 0 & 0 \\ 0 & J_{22} & 0 \\ 0 & 0 & J_{33} \end{bmatrix} \begin{pmatrix} \kappa_1^l \\ \kappa_2^l \\ \kappa_3^l \end{pmatrix}, \tag{15}$$

where

$$\begin{aligned} J_{11} &= \int_{A(s)} E\eta^2 d\xi d\eta, \quad J_{22} = \int_{A(s)} E\xi^2 d\xi d\eta, \\ J_{33} &= \int_{A(s)} G\left(\xi^2 + \eta^2\right) d\xi d\eta \end{aligned} \tag{16}$$

Remark 2 For the main context of this paper, we consider a circular cross section (with the exception of a rectangular cross section that is used in numerical test 5.1 for validation against a published result), and constant density ρ, Young's modulus E and shear modulus G. In which case, $I_{11} = I_{22} = \rho A^2/4\pi$, $I_{33} = \rho A^2/2\pi$, $J_{11} = J_{22} = EA^2/4\pi$ and $J_{33} = GA^2/2\pi$.

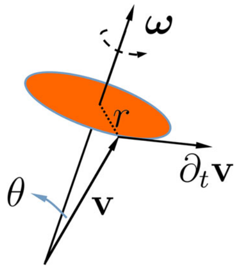


Fig. 3 Rotation of vector $\mathbf{v}$ with angular velocity $\boldsymbol{\omega}$. $r = \|\mathbf{v}\|\sin\theta$ and $\|\partial_t\mathbf{v}\| = r\|\boldsymbol{\omega}\|$

In the spirit of expressing all the unknown variables in terms of (x, y, z) and (α, β, γ), we further express the angular velocity $\boldsymbol{\omega}$ and curvature $\boldsymbol{\kappa}$ in terms of rotation angles (α, β, γ) in the following section.

2.3 Expressions of angular velocity and curvature in terms of the angles of rotation

For any fixed-length vector function, say, of t,

$$\mathbf{v}\cdot\mathbf{v} = c \Rightarrow (\partial_t\mathbf{v})\cdot\mathbf{v} = 0,$$

with constant c. This suggests that $\partial_t\mathbf{v}$ is always perpendicular to $\mathbf{v}$. If vector $\mathbf{v}$ rotates according to an angular velocity $\boldsymbol{\omega}$ as shown in Figure 3, we have

$$\partial_t\mathbf{v} = \boldsymbol{\omega}\times\mathbf{v}. \tag{17}$$

Since $\mathbf{d}_1$, $\mathbf{d}_2$ and $\mathbf{d}_3$ are all unit vectors, we can apply the above property (17) to these three vectors and have

$$\begin{aligned} &\sum_{i=1}^{i=3} \mathbf{d}_i \times \partial_t\mathbf{d}_i = \sum_{i=1}^{i=3} \mathbf{d}_i \times (\boldsymbol{\omega}\times\mathbf{d}_i) = \sum_{i=1}^{i=3} \boldsymbol{\omega}\,(\mathbf{d}_i\cdot\mathbf{d}_i) \\ &\quad - \sum_{i=1}^{i=3} \mathbf{d}_i\,(\boldsymbol{\omega}\cdot\mathbf{d}_i) = 2\boldsymbol{\omega}. \end{aligned} \tag{18}$$

Following the same argument, we also have:

$$\sum_{i=1}^{i=3} \mathbf{d}_i \times \partial_s\mathbf{d}_i = 2\boldsymbol{\kappa}. \tag{19}$$

Now, let $\mathbf{Q}^T = \left[\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3\right]$ with $\mathbf{q}_i^T = (q_{i1}, q_{i2}, q_{i3})$, $i = 1, 2, 3$, being the row vectors of $\mathbf{Q}$, then from (2) we have

$$\mathbf{d}_i = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3] \begin{pmatrix} q_{i1} \\ q_{i2} \\ q_{i3} \end{pmatrix}, \quad i = 1, 2, 3, \tag{20}$$

and

$$\partial_t \mathbf{d}_i = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3]\, \partial_t \begin{pmatrix} q_{i1} \\ q_{i2} \\ q_{i3} \end{pmatrix}$$
$$= [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3]\, \mathbf{Q} \partial_t \begin{pmatrix} q_{i1} \\ q_{i2} \\ q_{i3} \end{pmatrix}, \quad i = 1, 2, 3. \tag{21}$$

Using the fact that for any two vectors $\mathbf{u} = \sum_{i=1}^{3} u_i^l \mathbf{d}_i$ and $\mathbf{v} = \sum_{i=1}^{3} v_i^l \mathbf{d}_i$,

$$\mathbf{u} \times \mathbf{v} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3] \begin{bmatrix} 0 & -u_3^l & u_2^l \\ u_3^l & 0 & -u_1^l \\ -u_2^l & u_1^l & 0 \end{bmatrix} \begin{pmatrix} v_1^l \\ v_2^l \\ v_3^l \end{pmatrix}, \tag{22}$$

we can compute the cross product in (18):

$$\mathbf{d}_1 \times \partial_t \mathbf{q}_1 = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3] \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{Q} \partial_t \begin{pmatrix} q_{11} \\ q_{12} \\ q_{13} \end{pmatrix} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3] \begin{pmatrix} 0 \\ -\mathbf{q}_3 \cdot \partial_t \mathbf{q}_1 \\ \mathbf{q}_2 \cdot \partial_t \mathbf{q}_1 \end{pmatrix}, \tag{23}$$

$$\mathbf{d}_2 \times \partial_t \mathbf{q}_2 = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3] \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \mathbf{Q} \partial_t \begin{pmatrix} q_{21} \\ q_{22} \\ q_{23} \end{pmatrix} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3] \begin{pmatrix} \mathbf{q}_3 \cdot \partial_t \mathbf{q}_2 \\ 0 \\ -\mathbf{q}_1 \cdot \partial_t \mathbf{q}_2 \end{pmatrix}, \tag{24}$$

$$\mathbf{d}_3 \times \partial_t \mathbf{q}_3 = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3] \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \mathbf{Q} \partial_t \begin{pmatrix} q_{31} \\ q_{32} \\ q_{33} \end{pmatrix} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3] \begin{pmatrix} -\mathbf{q}_2 \cdot \partial_t \mathbf{q}_3 \\ \mathbf{q}_1 \cdot \partial_t \mathbf{q}_3 \\ 0 \end{pmatrix}. \tag{25}$$

Finally using (18) and (23) to (25), the angular velocity $\boldsymbol{\omega}$, in the local frame, can be expressed as:

$$\begin{pmatrix} \omega_1^l \\ \omega_2^l \\ \omega_3^l \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \mathbf{q}_3 \cdot \partial_t \mathbf{q}_2 - \mathbf{q}_2 \cdot \partial_t \mathbf{q}_3 \\ \mathbf{q}_1 \cdot \partial_t \mathbf{q}_3 - \mathbf{q}_3 \cdot \partial_t \mathbf{q}_1 \\ \mathbf{q}_2 \cdot \partial_t \mathbf{q}_1 - \mathbf{q}_1 \cdot \partial_t \mathbf{q}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{q}_3 \cdot \partial_t \mathbf{q}_2 \\ \mathbf{q}_1 \cdot \partial_t \mathbf{q}_3 \\ \mathbf{q}_2 \cdot \partial_t \mathbf{q}_1 \end{pmatrix}, \tag{26}$$

noticing that for $i \neq j$ $(i, j = 1, 2, 3)$

$$\mathbf{q}_i \cdot \mathbf{q}_j = 0 \Rightarrow \partial_t \mathbf{q}_i \cdot \mathbf{q}_j + \mathbf{q}_i \cdot \partial_t \mathbf{q}_j = 0.$$

A further calculation based on (1) and (26) expresses rotation angles in the local frame as follows:

$$\begin{pmatrix} \omega_1^l \\ \omega_2^l \\ \omega_3^l \end{pmatrix} = \mathbf{A} \partial_t \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}, \quad \mathbf{A} = \begin{bmatrix} 0 & -\cos\alpha & \sin\alpha\cos\beta \\ 0 & -\sin\alpha & -\cos\alpha\cos\beta \\ -1 & 0 & -\sin\beta \end{bmatrix}. \tag{27}$$

Using the same procedure, the curvature at a point along the centreline can be expressed in the local frame as:

$$\begin{pmatrix} \kappa_1^l \\ \kappa_2^l \\ \kappa_3^l \end{pmatrix} = \mathbf{A} \partial_s \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}. \tag{28}$$

Substituting equation (13) into equation (5), we express the conservation of linear momentum in its component form as follows:

$$\rho(s) A(s, t) \partial_{tt} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \partial_s \left(j(\tilde{s}, t) \mathbf{Q}^T \mathbf{K} \mathbf{Q} \partial_s \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \mathbf{Q}^T \begin{pmatrix} 0 \\ 0 \\ K_{33} \end{pmatrix} \right) + \begin{pmatrix} f_1^g \\ f_2^g \\ f_3^g \end{pmatrix}, \tag{29}$$

with $\mathbf{f}(s, t) = \sum_{i=1}^{3} f_i^g \mathbf{e}_i$.

Transforming the local coordinates in (27), (28), (7) and (15) into global coordinates by (4), then substituting them into equation (6), we express the conservation of angular momentum equation in its component form as follows:

$$\partial_t \left(\mathbf{Q}^T \mathbf{I} \mathbf{A} \partial_t \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} \right) = \partial_s \left(\mathbf{Q}^T \mathbf{J} \mathbf{A} \partial_s \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix} \right) + \begin{bmatrix} 0 & -\partial_s z & \partial_s y \\ \partial_s z & 0 & -\partial_s x \\ -\partial_s y & \partial_s x & 0 \end{bmatrix} \left(j(\tilde{s}, t) \mathbf{Q}^T \mathbf{K} \mathbf{Q} \partial_s \begin{pmatrix} x \\ y \\ z \end{pmatrix} - \mathbf{Q}^T \begin{pmatrix} 0 \\ 0 \\ K_{33} \end{pmatrix} \right) + \begin{pmatrix} l_1^g \\ l_2^g \\ l_3^g \end{pmatrix}, \tag{30}$$

with $\mathbf{l}(s, t) = \sum_{i=1}^{3} l_i^g \mathbf{e}_i$.

Remark 3 It can be seen from (2) that $\mathbf{d}_i \equiv \mathbf{q}_i$ $(i = 1, 2, 3)$ if we choose $\mathbf{e}_1 = (1, 0, 0)^T$, $\mathbf{e}_2 = (0, 1, 0)^T$ and $\mathbf{e}_3 = (0, 0, 1)^T$. This observation will be adopted in Section 5 for numerical implementation.

2.4 Incompressibility assumption

We assume the rod is incompressible and derive a condition for its cross section $A(s, t)$ in this section. An incompressible material requires the total volume to be constant, i.e.:

$$\frac{d}{dt}\int_{a(t)}^{b(t)} A(s,t)ds = \frac{d}{dt}\int_{a_0}^{b_0} A(s(\tilde{s},t),t)j(\tilde{s},t)d\tilde{s} = 0, \tag{31}$$

from which we get

$$\frac{dA(s,t)}{dt}j(\tilde{s},t) + A(s,t)\frac{dj(\tilde{s},t)}{dt} = 0. \tag{32}$$

This equation can be solved by separation of variables as follows:

$$\frac{dA}{A} = -\frac{dj}{j} = 0. \tag{33}$$

Considering the initial condition $j(\tilde{s},0) = 1$ and $A(\tilde{s},0) = A_0$, and noticing that A and j are both positive, the solution of (33) can be expressed as:

$$ln(A) = -ln(j) + ln(A_0) \Rightarrow A(s,t) = \frac{A_0}{j(\tilde{s},t)}. \tag{34}$$

2.5 Finite element weak form

Equations (29) and (30) can be solved either on the reference configuration $\tilde{s}$ (total Lagrangian formulation) or the current configuration s (updated Lagrangian formulation). These two formulations can be transformed from one to another using equation (34), and we introduce these two formulations in this section. Let

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}, \quad \boldsymbol{\alpha} = \begin{pmatrix} \alpha \\ \beta \\ \gamma \end{pmatrix}, \quad \mathbf{B}(\mathbf{x}) = \begin{bmatrix} 0 & -\partial_s z & \partial_s y \\ \partial_s z & 0 & -\partial_s x \\ -\partial_s y & \partial_s x & 0 \end{bmatrix}, \tag{35}$$

then the weak form of (29) and (30) on the current configuration $[a(t), b(t)]$ can be expressed as

$$\begin{aligned} &\int_a^b \rho A(s,t)\delta\mathbf{x}^T\partial_{tt}\mathbf{x}ds + \int_a^b \partial_s\delta\mathbf{x}^T\left[j\mathbf{Q}^T\mathbf{K}\mathbf{Q}\partial_s\mathbf{x} - K_{33}\mathbf{q}_3\right]ds \\ &\quad + \int_a^b \delta\boldsymbol{\alpha}^T\partial_t\left[\mathbf{Q}^T\mathbf{I}\mathbf{A}\partial_t\boldsymbol{\alpha}\right]ds + \int_a^b \partial_s\delta\boldsymbol{\alpha}^T\left[\mathbf{Q}^T\mathbf{J}\mathbf{A}\partial_s\boldsymbol{\alpha}\right]ds \\ &= \int_a^b \delta\boldsymbol{\alpha}^T\mathbf{B}\left[j\mathbf{Q}^T\mathbf{K}\mathbf{Q}\partial_s\mathbf{x} - K_{33}\mathbf{q}_3\right]ds \\ &\quad + \int_a^b \delta\boldsymbol{\alpha}^T\mathbf{l}ds + \int_a^b \delta\mathbf{x}^T\mathbf{f}ds \end{aligned} \tag{36}$$

with $\delta\mathbf{x}$ and $\delta\boldsymbol{\alpha}$ denoting the test functions corresponding to $\mathbf{x}$ and $\boldsymbol{\alpha}$ respectively. Let

$$\begin{aligned} &K_{11}^0 = K_{22}^0 = kGA_0, \quad K_{33}^0 = EA_0, \\ &\mathbf{K}_0 = \text{diag}\left(K_{11}^0, K_{22}^0, K_{33}^0\right), \end{aligned} \tag{37}$$

$$\begin{aligned} &I_{11}^0 = I_{22}^0 = \rho A_0^2/4\pi, \quad I_{33}^0 = \rho A_0^2/2\pi, \\ &\mathbf{I}_0 = \text{diag}\left(I_{11}^0, I_{22}^0, I_{33}^0\right), \end{aligned} \tag{38}$$

and

$$\begin{aligned} &J_{11}^0 = J_{22}^0 = EA_0^2/4\pi, \quad J_{33}^0 = GA_0^2/2\pi, \\ &\mathbf{J}_0 = \text{diag}\left(J_{11}^0, J_{22}^0, J_{33}^0\right). \end{aligned} \tag{39}$$

Then, (36) can rewritten, in the reference configuration $\tilde{s}$, as:

$$\begin{aligned} &\int_{a_0}^{b_0} \rho A_0\delta\mathbf{x}^T\partial_{tt}\mathbf{x}d\tilde{s} + \int_{a_0}^{b_0} \partial_{\tilde{s}}\delta\mathbf{x}^T\left[\mathbf{Q}^T\mathbf{K}_0\mathbf{Q}\partial_{\tilde{s}}\mathbf{x} - K_{33}^0\mathbf{q}_3\right]d\tilde{s} \\ &\quad + \int_{a_0}^{b_0} j\delta\boldsymbol{\alpha}^T\partial_t\left[j^{-2}\mathbf{Q}^T\mathbf{I}_0\mathbf{A}\partial_t\boldsymbol{\alpha}\right]d\tilde{s} \\ &\quad + \int_{a_0}^{b_0} j^{-3}\partial_{\tilde{s}}\delta\boldsymbol{\alpha}^T\left[\mathbf{Q}^T\mathbf{J}_0\mathbf{A}\partial_{\tilde{s}}\boldsymbol{\alpha}\right]d\tilde{s} \\ &= \int_{a_0}^{b_0} j^{-1}\delta\boldsymbol{\alpha}^T\mathbf{B}\left[\mathbf{Q}^T\mathbf{K}_0\mathbf{Q}\partial_{\tilde{s}}\mathbf{x} - K_{33}\mathbf{q}_3\right]d\tilde{s} \\ &\quad + \int_{a_0}^{b_0} j\delta\boldsymbol{\alpha}^T\mathbf{l}d\tilde{s} + \int_{a_0}^{b_0} j\delta\mathbf{x}^T\mathbf{f}d\tilde{s}. \end{aligned} \tag{40}$$

Remark 4 It is convenient to solve a forward problem on $\tilde{s}$ with updating the deformation scaler j based on a fixed-point iteration for example, while it is convenient to solve a backward problem on s (see Section 3) because we already have the current mesh s and j can be computed directly.

3 The optimal control problem

In this section, we formulate an optimal control problem based on the Cosserat rod model described in the previous section. The motivation is to reconstruct the full rod configuration by computing (α, β, γ) from observed data (x_g, y_g, z_g). This is an inverse problem which we formulate as a control problem. In case of low Reynolds number rods, we neglect the inertia terms and consider the following optimisation problem: reducing the discrepancy between the centreline $\mathbf{x}$ and an objective position given by the observed data $\mathbf{x}_g = (x_g, y_g, z_g)$, by optimisation of the external force $\mathbf{f}$ and torque $\mathbf{l}$ in (29) and (30).

Problem 1 (piecewise-in-time control) Given the state variables $\mathbf{x}_{n-1}$ and $\boldsymbol{\alpha}_{n-1}$ at the previous time t_{n-1} ($n = 1, 2, \ldots$), and an objective position vector $\mathbf{x}_g(t_n)$ of the worm's centreline at current time t_n,

$$\begin{aligned}\underset{\mathbf{f}_n, \mathbf{l}_n \in L^2([a,b])}{\text{minimise}} \quad & J(\mathbf{x}_n, \boldsymbol{\alpha}_n, \mathbf{f}_n, \mathbf{l}_n) = \frac{\lambda_g}{2} \int_a^b \left|\mathbf{x}_n - \mathbf{x}_g(t_n)\right|^2 \\ & + \frac{\lambda_f}{2} \int_a^b |\mathbf{f}_n|^2 + \frac{\lambda_l}{2} \int_a^b |\mathbf{l}_n|^2 \\ & + \frac{\lambda_d}{2} \int_a^b |\partial_s \mathbf{x}_n - \mathbf{d}_3(\boldsymbol{\alpha}_n)|^2 + \frac{1}{2} \int_a^b \left|\boldsymbol{\Lambda}_\kappa^{1/2} \mathbf{A} \partial_s \boldsymbol{\alpha}_n\right|^2 \\ & + \frac{1}{2} \int_a^b \left|\boldsymbol{\Lambda}_\omega^{1/2} \mathbf{A} \frac{(\boldsymbol{\alpha}_n - \boldsymbol{\alpha}_{n-1})}{\Delta t}\right|^2, \end{aligned} \tag{41}$$

subject to

$$\partial_s \left[\mathbf{Q}^T(\boldsymbol{\alpha}_n) \mathbf{K}_0 \mathbf{Q}(\boldsymbol{\alpha}_n) \partial_s \mathbf{x}_n - j^{-1} K_{33} \mathbf{q}_3(\boldsymbol{\alpha}_n)\right] + \mathbf{f}_n = 0, \tag{42}$$

and

$$\begin{aligned}& \partial_s \left[\mathbf{Q}^T(\boldsymbol{\alpha}_n) \mathbf{J} \mathbf{A}(\boldsymbol{\alpha}_n) \partial_s \boldsymbol{\alpha}_n\right] + \mathbf{B}(\mathbf{x}_n) \left[\mathbf{Q}^T(\boldsymbol{\alpha}_n) \mathbf{K}_0 \mathbf{Q}(\boldsymbol{\alpha}_n) \partial_s \mathbf{x}_n \right. \\ & \left. - j^{-1} K_{33} \mathbf{q}_3(\boldsymbol{\alpha}_n)\right] + \mathbf{l}_n = 0. \end{aligned} \tag{43}$$

In the above, $\boldsymbol{\Lambda}_\kappa = \operatorname{diag}\left(\lambda_{\kappa_1}, \lambda_{\kappa_2}, \lambda_{\kappa_3}\right)$ and $\boldsymbol{\Lambda}_\omega = \operatorname{diag}\left(\lambda_{\omega_1}, \lambda_{\omega_2}, \lambda_{\omega_3}\right)$ are two diagonal matrices. The first term in (41) is the real objective to be minimised, and we choose $\lambda_g = 10^9 \sim 1/\int_a^b |\mathbf{x}_g|^2$, so that the first term would not become infinitely small during the process of minimisation. All the other terms are regularisation terms with regularisation parameters λ_f, λ_l, λ_d, $\boldsymbol{\Lambda}_\kappa$ and $\boldsymbol{\Lambda}_\omega$. Too large regularisation parameters could make it difficult to achieve the real objective, while too small ones may cause convergence issues for the numerical scheme. These regularisation terms have different mathematical and computational purposes (see Section 3.4 for biological meanings of each term): generally speaking, we want to add reasonable constraints so that the control problem is solvable and a unique solution can be obtained; the second (λ_f-term) and the third term (λ_l-term) are constraints of the control variables so that they would not go to infinity; the term λ_d is used to disallow arbitrary rotation angles and ensure the problem is solvable; the regularisation parameter of the last two terms $\boldsymbol{\Lambda}_\kappa-$term and $\boldsymbol{\Lambda}_\omega-$term in (41) are diagonal matrices, whose elements are parameters to control the three components of generalised curvature and angular velocity correspondingly.

Remark 5 Because we lack data to control the rotation angle $\boldsymbol{\alpha}$, we set a control of the variation of $\boldsymbol{\alpha}$, with respect to space ($\boldsymbol{\Lambda}_\kappa-$term) and time ($\boldsymbol{\Lambda}_\omega-$term), in the last two terms in (41), which is necessary for the solution existence and for the convergence of our numerical method as formulated in Problem 2 in Section 4.

We introduce the Lagrange multipliers (or adjoint variables) $\hat{\mathbf{x}}$, $\hat{\boldsymbol{\alpha}}$ to eliminate the constraints of Problem 1 (dropping the subscript 'n' for simplicity).

$$\begin{aligned} L\left(\mathbf{x}, \boldsymbol{\alpha}, \mathbf{f}, \mathbf{l}, \hat{\mathbf{x}}, \hat{\boldsymbol{\alpha}}\right) &= J(\mathbf{x}, \mathbf{f}, \mathbf{l}) \\ & - \int_a^b \hat{\mathbf{x}}^T \left[\partial_s \left(\mathbf{Q}^T \mathbf{K}_0 \mathbf{Q} \partial_s \mathbf{x} - j^{-1} K_{33} \mathbf{q}_3\right) + \mathbf{f}\right] \\ & - \int_a^b \hat{\boldsymbol{\alpha}}^T \left[\partial_s \left(\mathbf{Q}^T \mathbf{J} \mathbf{A} \partial_s \boldsymbol{\alpha}\right)\right. \\ & \left. + \mathbf{B} \left(\mathbf{Q}^T \mathbf{K}_0 \mathbf{Q} \partial_s \mathbf{x} - j^{-1} K_{33} \mathbf{q}_3\right) + \mathbf{l}\right] \\ & + \hat{\mathbf{x}}(b)^T \left[\mathbf{x}(b) - \mathbf{x}_g(b)\right] - \hat{\mathbf{x}}(a)^T \left[\mathbf{x}(a) - \mathbf{x}_g(a)\right] \\ & + \hat{\boldsymbol{\alpha}}(b)^T \mathbf{m}(b) - \hat{\boldsymbol{\alpha}}(a)^T \mathbf{m}(a). \end{aligned} \tag{44}$$

The following boundary conditions are also included in the above functional L:

$$\mathbf{x}(a) - \mathbf{x}_g(a) = \mathbf{x}(b) - \mathbf{x}_g(b) = 0, \tag{45}$$

and

$$\mathbf{m}(a) = \mathbf{m}(b) = 0. \tag{46}$$

Remark 6 We find that the proposed optimal control formulation is solvable with either a Dirichlet boundary condition $\alpha(a) = \alpha_a$ (first component of rotation $\boldsymbol{\alpha}$) or the regularisation $\boldsymbol{\Lambda}_\omega-$term in (41). We do not have the Dirichlet data for all frames unfortunately, but we notice from the last term in (41) that $\boldsymbol{\alpha}_0$ must be given. Therefore, we solve the first frame using Dirichlet boundary condition $\alpha(a) = 0$ instead of $\boldsymbol{\Lambda}_\omega-$term, and from the second frame we use the $\boldsymbol{\Lambda}_\omega-$term.

Remark 7 The solvability of Problem 1 is generally a difficult question, and there is one case we are sure is unsolvable: suppose the rod undergoes a pure twist induced only by the third component of the torque $\mathbf{l}$ in (43), in which case there is no way to determine the frames only using the data from the centreline. In all other cases ($l_3^l = 0$), the deformation of the centreline is coupled with the frames and hopefully we can detect the frames through this coupling by solving Problem 1. We shall validate this idea in numerical test 5.2. Luckily, it is parsimonious to assume $l_3^l = 0$ when modelling *C. elegans* because its longitudinal body wall muscles which may not generate l_3^l torque.

After integration by parts, L can be further expressed as follows:

$$\begin{aligned} L\left(\mathbf{x}, \boldsymbol{\alpha}, \mathbf{f}, \mathbf{l}, \hat{\mathbf{x}}, \hat{\boldsymbol{\alpha}}\right) &= J(\mathbf{x}, \mathbf{f}, \mathbf{l}) \\ &+ \int_a^b \partial_s \hat{\mathbf{x}}^T \left(\mathbf{Q}^T \mathbf{K}_0 \mathbf{Q} \partial_s \mathbf{x} - j^{-1} K_{33} \mathbf{q}_3\right) - \int_a^b \hat{\mathbf{x}}^T \mathbf{f} \\ &+ \int_a^b \partial_s \hat{\boldsymbol{\alpha}}^T \left(\mathbf{Q}^T \mathbf{J} \mathbf{A}\right) \partial_s \boldsymbol{\alpha} \\ &\quad - \int_a^b \hat{\boldsymbol{\alpha}}^T \mathbf{B} \left(\mathbf{Q}^T \mathbf{K}_0 \mathbf{Q} \partial_s \mathbf{x} - j^{-1} K_{33} \mathbf{q}_3\right) \\ &- \int_a^b \hat{\boldsymbol{\alpha}}^T \mathbf{l} \\ &+ \hat{\mathbf{x}}(b)^T \left[\mathbf{x}(b) - \mathbf{x}_g(b)\right] - \hat{\mathbf{x}}(a)^T \left[\mathbf{x}(a) - \mathbf{x}_g(a)\right] \\ &- \hat{\mathbf{x}}(b)^T \mathbf{n}(b) + \hat{\mathbf{x}}(a)^T \mathbf{n}(a). \end{aligned} \tag{47}$$

The following Karush-Kuhn-Tucker (KKT) conditions are the first-order necessary conditions to minimise (47).

$$\delta L(\cdot)\left[\left(\hat{\mathbf{x}}, \hat{\boldsymbol{\alpha}}\right); \left(\delta\hat{\mathbf{x}}, \delta\hat{\boldsymbol{\alpha}}\right)\right] = 0, \tag{48}$$

$$\delta L(\cdot)\left[(\mathbf{x}, \boldsymbol{\alpha}); (\delta\mathbf{x}, \delta\boldsymbol{\alpha})\right] = 0, \tag{49}$$

$$\delta L(\cdot)\left[(\mathbf{f}, \mathbf{l}); (\delta\mathbf{f}, \delta\mathbf{l})\right] = 0, \tag{50}$$

with

$$\delta L(\cdot)[\mathbf{p}; \mathbf{q}] = \frac{d}{d\epsilon} L\left(\mathbf{p} + \epsilon\mathbf{q}\right)\Big|_{\epsilon=0} \tag{51}$$

being the GOateaux derivative with respect to variable $\mathbf{p}$ along the direction $\mathbf{q}$ [67]. If $\mathbf{q}$ is an arbitrary direction from $\mathbf{p}$, it is usually expressed as $\mathbf{q} = \delta\mathbf{p}$ (variation of $\mathbf{p}$) [7], in which case it is convenient to abbreviate $\delta L(\mathbf{p})[\mathbf{p}; \delta\mathbf{p}]$ as $\delta L(\mathbf{p})$.

Let $L^2([a, b])$ be the square integrable functions in domain $[a, b]$ with inner product $(u, v) = \int_a^b uv$ and the induced norm $\|u\| = (u, u)^{1/2}$, $\forall u, v \in L^2([a, b])$. For vector function $\mathbf{u} \in L^2([a, b])^d$ ($d = 6$ for three components of the position vector and three components of the rotation vector), the norm is defined component-wise as $\|\mathbf{u}\|^2 = \sum_{i=1}^d \|u_i\|^2$. Let $H^1([a, b]) = \left\{\mathbf{u} : \mathbf{u}, \partial_s\mathbf{u} \in L^2([a, b])^d\right\}$, and $H^1_D([a, b])$ be the subspace of $H^1([a, b])$ whose functions satisfy the Dirichlet boundary condition in (45), in particular $H^1_0([a, b])$, the homogeneous Dirichlet boundary conditions. The above optimality conditions: (48) to (50), lead to the following partial differential equations (in weak forms).

3.1 Primal equation

The optimality condition (48) gives the primal equation in its weak form as follows. Find $(\mathbf{x}, \boldsymbol{\alpha}) \in H^1_D$, such that $\forall \left(\delta\hat{\mathbf{x}}, \delta\hat{\boldsymbol{\alpha}}\right) \in H^1_0$:

$$\begin{aligned} &\int_a^b \partial_s \delta\hat{\mathbf{x}}^T \left(\mathbf{Q}^T \mathbf{K}_0 \mathbf{Q}\right) \partial_s \mathbf{x} + \int_a^b \partial_s \delta\hat{\boldsymbol{\alpha}}^T \left(\mathbf{Q}^T \mathbf{J} \mathbf{A}\right) \partial_s \boldsymbol{\alpha} \\ &\quad - \int_a^b \delta\hat{\boldsymbol{\alpha}}^T \left(\mathbf{B} \mathbf{Q}^T \mathbf{K}_0 \mathbf{Q}\right) \partial_s \mathbf{x} \\ &\quad = \int_a^b \delta\hat{\mathbf{x}}^T \mathbf{f} + \int_a^b \delta\hat{\boldsymbol{\alpha}}^T \mathbf{l} + \int_a^b j^{-1} K_{33} \partial_s \delta\hat{\mathbf{x}}^T \mathbf{q}_3 \\ &\quad - \int_a^b j^{-1} K_{33} \delta\hat{\boldsymbol{\alpha}}^T \mathbf{B} \mathbf{q}_3. \end{aligned} \tag{52}$$

3.2 Adjoint equation

The optimality condition (49) gives the adjoint equation in its weak form as follows (neglecting the variation of the matrix $\mathbf{Q}$ $\mathbf{A}$, $\mathbf{B}$ and vector $\mathbf{q}_3$). Find $\left(\hat{\mathbf{x}}, \hat{\boldsymbol{\alpha}}\right) \in H^1_0$, such that $\forall (\delta\mathbf{x}, \delta\boldsymbol{\alpha}) \in H^1_0$:

$$\begin{aligned} &\int_a^b \partial_s \hat{\mathbf{x}}^T \left(\mathbf{Q}^T \mathbf{K}_0 \mathbf{Q}\right) \partial_s \delta\mathbf{x} + \int_a^b \partial_s \hat{\boldsymbol{\alpha}}^T \left(\mathbf{Q}^T \mathbf{J} \mathbf{A}\right) \partial_s \delta\boldsymbol{\alpha} \\ &\quad - \int_a^b \hat{\boldsymbol{\alpha}}^T \left(\mathbf{B} \mathbf{Q}^T \mathbf{K}_0 \mathbf{Q}\right) \partial_s \delta\mathbf{x} + \lambda_g \int_a^b \delta\mathbf{x}^T \left(\mathbf{x} - \mathbf{x}_g\right) \\ &\quad + \int_a^b \delta\boldsymbol{\alpha}^T \mathbf{A}^T \boldsymbol{\Lambda}_\kappa \mathbf{A} \partial_s \boldsymbol{\alpha} \\ &\quad + \frac{1}{\Delta t^2} \int_a^b \delta\boldsymbol{\alpha}^T \mathbf{A}^T \boldsymbol{\Lambda}_\omega \mathbf{A} \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}_{n-1}\right) \\ &\quad + \lambda_d \int_a^b \partial_s \delta\mathbf{x}^T \left(\partial_s \mathbf{x} - \mathbf{d}_3(\boldsymbol{\alpha})\right) \\ &\quad + \lambda_d \int_a^b \delta\mathbf{d}_3^T \left(\partial_s \mathbf{x} - \mathbf{d}_3(\boldsymbol{\alpha})\right) = 0. \end{aligned} \tag{53}$$

Remark 8 We have implemented the adjoint equation with consideration of variation of $\mathbf{q}_3$, and we found that our optimal control algorithm in Section 4 struggled to converge. It is worth investigating the reason for this convergence issue, and testing the case in which the variation of all these terms is included in the future.

3.3 Optimality equation

The optimality condition (50) gives the relation between the control force and adjoint variable:

$$\lambda_f \int_a^b \delta\mathbf{f}^T \mathbf{f} + \lambda_l \int_a^b \delta\mathbf{l}^T \mathbf{l} = \int_a^b \delta\mathbf{f}^T \hat{\mathbf{x}} + \int_a^b \delta\mathbf{l}^T \hat{\boldsymbol{\alpha}}. \tag{54}$$

3.4 Relation to *C. elegans* locomotion problem

Our locomotion dataset, obtained from a 3D microscopic set up [72] contains many different trajectories of centreline

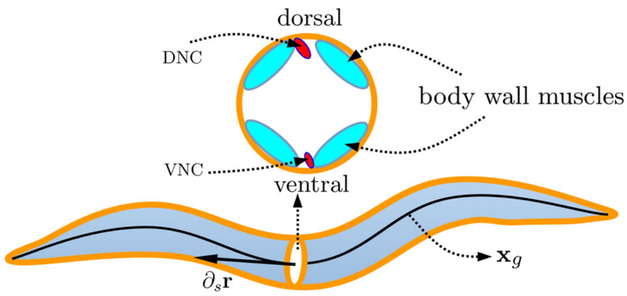


Fig. 4 A sketch of *C. elegans*. DNC: dorsal nerve cord; VNC: ventral nerve cord

positions $\mathbf{x}_g$ over time which will be used in Problem 1. The controls, the forces and the torques, can be interpreted as the reaction force from the surrounding fluids, which is initially activated by the worm's muscles.

The muscles generating locomotion in *C. elegans*, are called body wall muscles (see Figure 4) because they are tethered to the 'wall' (or cuticle) of the animal, acting longitudinally to contract or relax the local side of the body. As *C. elegans* contains 95 body wall muscles that span the entire body length, we consider the action of the muscles continuously along the body. The directionality of the muscle contraction at every point along the animal is determined by α, β and γ which themselves are unknowns as part of the control problem. To represent this muscle action, we only allow $\mathbf{d}_3$ close to the tangential direction of the body and also restrict the twisting movement of the worm as captured by the λ_d-term.

The regularisations given by $\boldsymbol{\Lambda}_\kappa-$term and $\boldsymbol{\Lambda}_\omega-$term are also biologically motivated. For example, it is known from the anatomy that the left and right muscle quadrants do not receive distinct neural connections along the body and tail (i.e. the posterior two thirds of the body which lie beyond the neck of the animal). This results in the majority of bending occurring in the dorsal-ventral directions and less in the left-right directions with the exception of the head and tail. We therefore adjust the magnitude of λ_{κ_1} and λ_{κ_2} to favour solutions that have more bending around $\mathbf{d}_1$ than $\mathbf{d}_2$. The longitudinal muscles also restrict the twisting motion of the body which can be considered by setting a larger λ_{κ_3}. These additional adjustments of constraints may result in a frame that more closely matches the anatomically meaningful frame of the animal as it moves around in 3D. $\boldsymbol{\Lambda}_\omega-$term models the internal friction, which can also be different in three local directions based on the worm's anatomy.

For the forward simulations of biological worms, the Neumann boundary condition is usually adopted at both ends of the rod. For the backward simulations, we can fully use the information from the data and adopt appropriate Dirichlet boundary conditions as considered above in (45).

Remark 9 We have the data $\mathbf{x}_g$ of the worm's centreline for every time frame, which means the mesh (arc length) $s(\tilde{s}, t_n)$ is known at the current time frame t_n. Therefore, we can directly compute the deformation scaler $j = \partial_{\tilde{s}} s(\tilde{s}, t_n)$. This is generally not true for a forward problem, which usually requires j to be iteratively computed.

4 A monolithic optimal control formulation

Substituting the optimality condition (54), specifically its strong form $\mathbf{f} = \hat{\mathbf{x}}/\lambda_f$ and $\mathbf{f} = \hat{\boldsymbol{\alpha}}/\lambda_l$, into equation (52), we have a monolithic scheme to solve the optimisation Problem 1 as follows.

$$\begin{aligned}
&\int_a^b \partial_s \delta\hat{\mathbf{x}}^T \left(\mathbf{Q}^T \mathbf{K}_0 \mathbf{Q}\right) \partial_s \mathbf{x} + \int_a^b \partial_s \delta\hat{\boldsymbol{\alpha}}^T \left(\mathbf{Q}^T \mathbf{J}\mathbf{A}\right) \partial_s \boldsymbol{\alpha} \\
&\quad - \int_a^b \delta\hat{\boldsymbol{\alpha}}^T \left(\mathbf{B}\mathbf{Q}^T \mathbf{K}_0 \mathbf{Q}\right) \partial_s \mathbf{x} \\
&\quad + \int_a^b \partial_s \hat{\mathbf{x}}^T \left(\mathbf{Q}^T \mathbf{K}_0 \mathbf{Q}\right) \partial_s \delta\mathbf{x} + \int_a^b \partial_s \hat{\boldsymbol{\alpha}}^T \left(\mathbf{Q}^T \mathbf{J}\mathbf{A}\right) \partial_s \delta\boldsymbol{\alpha} \\
&\quad - \int_a^b \hat{\boldsymbol{\alpha}}^T \left(\mathbf{B}\mathbf{Q}^T \mathbf{K}_0 \mathbf{Q}\right) \partial_s \delta\mathbf{x} + \lambda_g \int_a^b \delta\mathbf{x}^T \left(\mathbf{x} - \mathbf{x}_g\right) \\
&\quad + \int_a^b \delta\boldsymbol{\alpha}^T \mathbf{A}^T \boldsymbol{\Lambda}_\kappa \mathbf{A} \partial_s \boldsymbol{\alpha} + \frac{1}{\Delta t^2} \int_a^b \delta\boldsymbol{\alpha}^T \mathbf{A}^T \boldsymbol{\Lambda}_\omega \mathbf{A} \left(\boldsymbol{\alpha} - \boldsymbol{\alpha}_{n-1}\right) \\
&\quad + \lambda_d \int_a^b \partial_s \delta\mathbf{x}^T \left(\partial_s \mathbf{x} - \mathbf{d}_3(\boldsymbol{\alpha})\right) + \lambda_d \int_a^b \delta\mathbf{d}_3^T \left(\partial_s \mathbf{x} - \mathbf{d}_3(\boldsymbol{\alpha})\right) \\
&= \int_a^b \frac{1}{\lambda_f} \delta\hat{\mathbf{x}}^T \hat{\mathbf{x}} + \int_a^b \frac{1}{\lambda_l} \delta\hat{\boldsymbol{\alpha}}^T \hat{\boldsymbol{\alpha}} + \int_a^b j^{-1} K_{33} \partial_s \delta\hat{\mathbf{x}}^T \mathbf{q}_3 \\
&\quad - \int_a^b j^{-1} K_{33} \delta\hat{\boldsymbol{\alpha}}^T \mathbf{B}\mathbf{q}_3.
\end{aligned} \tag{55}$$

The above equation is highly non-linear and coupled between the state variables $\mathbf{x}, \boldsymbol{\alpha}$ and adjoint variables $\hat{\mathbf{x}}, \hat{\boldsymbol{\alpha}}$. For the coefficient matrices $\mathbf{Q}$, $\mathbf{A}$ and $\mathbf{B}$, we use the fixed-point iterations to compute $\mathbf{Q}(\boldsymbol{\alpha}_n^k) \to \mathbf{Q}(\boldsymbol{\alpha}_n)$, $\mathbf{A}(\boldsymbol{\alpha}_n^k) \to \mathbf{A}(\boldsymbol{\alpha}_n)$ and $\mathbf{B}(\mathbf{x}_n^k) \to \mathbf{B}(\mathbf{x}_n)$ as $k \to +\infty$, starting from $\boldsymbol{\alpha}_n^0 = \boldsymbol{\alpha}_{n-1}$ and $\mathbf{x}_n^0 = \mathbf{x}_{n-1}$. In addition, $\delta\mathbf{d}_3(\boldsymbol{\alpha})$ is also linearised as $\delta\mathbf{d}_3(\boldsymbol{\alpha}^k)$ based on the fixed-point iteration. We use Newton's method to linearise the non-linear term $\mathbf{d}_3(\boldsymbol{\alpha})$ as follows. From (1) and Remark 3, $\mathbf{d}_3 = \left[-\cos\beta \sin\gamma, \sin\beta, \cos\beta\cos\gamma\right]^T$, we compute the variation of $\mathbf{d}_3$ with respect to $\boldsymbol{\alpha}$ along $\delta\boldsymbol{\alpha}$:

$$\delta\mathbf{d}_3\left(\boldsymbol{\alpha}\right) = \delta\beta \mathbf{d}_\beta(\beta, \gamma) + \delta\gamma \mathbf{d}_\gamma(\beta, \gamma), \tag{56}$$

with

$$\mathbf{d}_\beta(\beta, \gamma) = \left[\sin\beta \sin\gamma, \cos\beta, -\sin\beta\cos\gamma\right]^T, \tag{57}$$

and

$$\mathbf{d}_\gamma(\beta,\gamma) = \left[-\cos\beta\cos\gamma, 0, -\cos\beta\sin\gamma\right]^T. \tag{58}$$

The first order Taylor approximation of $\mathbf{d}_3(\boldsymbol{\alpha})$ at $\boldsymbol{\alpha}^k$ is expressed as:

$$\mathbf{d}_3(\boldsymbol{\alpha}) \approx \mathbf{d}_3(\boldsymbol{\alpha}^k) + \delta\mathbf{d}_3\left[\boldsymbol{\alpha}^k; \boldsymbol{\alpha} - \boldsymbol{\alpha}^k\right]. \tag{59}$$

Substituting (57) and (58) into (59), we then can linearise $\mathbf{d}_3(\boldsymbol{\alpha})$ as follows:

$$\begin{aligned}\mathbf{d}_3(\boldsymbol{\alpha}) \approx\ & \mathbf{d}_3(\boldsymbol{\alpha}^k) + \left(\beta - \beta^k\right)\mathbf{d}_\beta(\beta^k, \gamma^k) \\ & + \left(\gamma - \gamma^k\right)\mathbf{d}_\gamma(\beta^k, \gamma^k).\end{aligned} \tag{60}$$

Finally, by substituting (56) with $\beta = \beta^k$ and $\gamma = \gamma^k$ and (60) into equation (55), and denoting $\mathbf{Q}\left(\boldsymbol{\alpha}_n^k\right) = \mathbf{Q}_k$, $\mathbf{A}\left(\boldsymbol{\alpha}_n^k\right) = \mathbf{A}_k$, $\mathbf{B}(\mathbf{x}_n^k) = \mathbf{B}_k$, $\mathbf{d}_\beta(\beta^k, \gamma^k) = \mathbf{d}_\beta^k$ and $\mathbf{d}_\gamma(\beta^k, \gamma^k) = \mathbf{d}_\gamma^k$, we have the following (Problem 2) monolithic formulation to solve Problem 1.

Problem 2 Given the state variables $(\mathbf{x}_{n-1}, \boldsymbol{\alpha}_{n-1})$ at the previous time t_{n-1} $(n = 1, 2, \ldots)$, and an objective position vector $\mathbf{x}_g(t_n)$ at the current time t_n, compute $\left(\mathbf{x}^k, \boldsymbol{\alpha}^k\right) \to (\mathbf{x}, \boldsymbol{\alpha}) \in H_D^1$, $\left(\hat{\mathbf{x}}, \hat{\boldsymbol{\alpha}}\right) \in H_0^1$ iteratively from $\left(\mathbf{x}^0, \boldsymbol{\alpha}^0\right) = (\mathbf{x}_{n-1}, \boldsymbol{\alpha}_{n-1})$, such that $\forall\,(\delta\mathbf{x}, \delta\boldsymbol{\alpha}) \in H_0^1$ and $\forall\left(\delta\hat{\mathbf{x}}, \delta\hat{\boldsymbol{\alpha}}\right) \in H_0^1$:

$$\begin{aligned}
& \int_a^b \partial_s\delta\hat{\mathbf{x}}^T\left(\mathbf{Q}_k^T\mathbf{K}_0\mathbf{Q}_k\right)\partial_s\mathbf{x} + \partial_s\delta\hat{\boldsymbol{\alpha}}^T\left(\mathbf{Q}_k^T\mathbf{J}\mathbf{A}_k\right)\partial_s\boldsymbol{\alpha} \\
& \quad + \int_a^b \partial_s\hat{\mathbf{x}}^T\left(\mathbf{Q}_k^T\mathbf{K}_0\mathbf{Q}_k\right)\partial_s\delta\mathbf{x} + \partial_s\hat{\boldsymbol{\alpha}}^T\left(\mathbf{Q}_k^T\mathbf{J}\mathbf{A}_k\right)\partial_s\delta\boldsymbol{\alpha} \\
& \quad - \int_a^b \hat{\boldsymbol{\alpha}}^T\left(\mathbf{B}_k\mathbf{Q}_k^T\mathbf{K}_0\mathbf{Q}_k\right)\partial_s\delta\mathbf{x} + \delta\hat{\boldsymbol{\alpha}}^T\left(\mathbf{B}_k\mathbf{Q}_k^T\mathbf{K}_0\mathbf{Q}_k\right)\partial_s\mathbf{x} \\
& \quad + \lambda_g\int_a^b \delta\mathbf{x}^T\mathbf{x} - \frac{1}{\lambda_f}\int_a^b \delta\hat{\mathbf{x}}^T\hat{\mathbf{x}} - \frac{1}{\lambda_l}\int_a^b \delta\hat{\boldsymbol{\alpha}}^T\hat{\boldsymbol{\alpha}} \\
& \quad + \int_a^b \partial_s\delta\boldsymbol{\alpha}^T\mathbf{A}_k^T\boldsymbol{\Lambda}_\kappa\mathbf{A}_k\partial_s\boldsymbol{\alpha} + \frac{1}{\Delta t^2}\int_a^b \delta\boldsymbol{\alpha}^T\mathbf{A}_k^T\boldsymbol{\Lambda}_\omega\mathbf{A}_k\boldsymbol{\alpha} \\
& \quad + \lambda_d\int_a^b \partial_s\delta\mathbf{x}^T\partial_s\mathbf{x} + \lambda_d\int_a^b \left(\delta\beta\mathbf{d}_\beta^k + \delta\gamma\mathbf{d}_\gamma^k\right)^T\partial_s\mathbf{x} \\
& \quad - \partial_s\delta\mathbf{x}^T\left(\beta\mathbf{d}_\beta^k + \gamma\mathbf{d}_\gamma^k\right) \\
& \quad - \lambda_d\int_a^b \left(\delta\beta\mathbf{d}_\beta^k + \delta\gamma\mathbf{d}_\gamma^k\right)^T\left(\beta\mathbf{d}_\beta^k + \gamma\mathbf{d}_\gamma^k\right) \\
& = \lambda_g\int_a^b \delta\mathbf{x}^T\mathbf{x}_g + \int_a^b j^{-1}K_{33}\partial_s\delta\hat{\mathbf{x}}^T\mathbf{q}_3 \\
& \quad - \int_a^b j^{-1}K_{33}\delta\hat{\boldsymbol{\alpha}}^T\mathbf{B}_k\mathbf{q}_3^k \\
& \quad + \lambda_d\int_a^b \partial_s\delta\mathbf{x}^T\left(\mathbf{d}_3^k - \beta^k\mathbf{d}_\beta^k - \gamma^k\mathbf{d}_\gamma^k\right) \\
& \quad + \lambda_d\int_a^b \left(\delta\beta\mathbf{d}_\beta^k + \delta\gamma\mathbf{d}_\gamma^k\right)\mathbf{d}_3^k \\
& \quad - \lambda_d\int_a^b \left(\delta\beta\mathbf{d}_\beta^k + \delta\gamma\mathbf{d}_\gamma^k\right)^T\left(\beta^k\mathbf{d}_\beta^k + \gamma^k\mathbf{d}_\gamma^k\right) \\
& \quad + \frac{1}{\Delta t^2}\int_a^b \delta\boldsymbol{\alpha}^T\mathbf{A}_k^T\boldsymbol{\Lambda}_\omega\mathbf{A}_k\boldsymbol{\alpha}_{n-1}.
\end{aligned} \tag{61}$$

Remark 10 For the above fixed-point iteration, a relaxation parameter $0 \le w \le 1$ is introduced to stabilise the algorithm: instead of directly updating $\left(\mathbf{x}^k, \boldsymbol{\alpha}^k\right)$ after solving (61), a weighted $w\left(\mathbf{x}^k, \boldsymbol{\alpha}^k\right) + (1 - w)\left(\mathbf{x}^{k-1}, \boldsymbol{\alpha}^{k-1}\right)$ is adopted. We use $w = 0.5$ for all our simulations.

5 Numerical tests

We first validate the formulation (40) for simulation of a forward problem with a time discretisation scheme introduced in Appendix A, and then apply the optimal control formulation (61) to data from a forward simulation, in which case we have the ground truth rotations of the local frames and a quantitative comparison can be performed. Finally, we apply the optimal control method to data from laboratory experiments and infer the frames of rotation. All the numerical tests are implemented using open-source library FreeFem++ [42]. For code and results, see Data Availability in declarations section below.

5.1 Forward simulation of a cantilever beam

We consider a cantilever beam with a dynamic load and reproduce the result presented in [12]. The beam's length $L = 1m$ (which is a constant for this test due to a small deformation), with density $\rho = 2.73 \times 10^3 kg/m^3$, Young's modulus $E = 7.10 \times 10^{10} Pa$ and shear modulus $G = 2.69 \times 10^{10} Pa$. The cross section of the beam is a rectangle with width $a = 0.06m$ and height $h = 0.04m$, and the numerical shear correction factor for this cross section is set to be $k = 0.833$. The moment of inertia is $I_{11} = \rho ab^3/12$, $I_{22} = \rho a^3 b/12$ and $I_{33} = I_{11} + I_{22}$. The stiffness for the torque in (15) is $J_{11} = Eab^3/12$, $J_{22} = Ea^3b/12$ and $J_{33} = G(ab^3 + a^3b)/12$. The external force, corresponding to $\mathbf{f}$ in equation (40), is expressed as:

$$\begin{aligned}& f_1^g(s,t) = f_2^g(s,t) = 2\sin(\pi s)\sin(8\omega_0 t)kNm^{-1}, \\ & f_3^g(s,t) = 0,\end{aligned} \tag{62}$$

with the natural frequency of the system $\omega_0 = 207.0236s^{-1}$.

The beam is discretised by 100 segments and the total computational time $T = 0.06$ is divided into 1000 steps. The displacement and rotation at the end of the beam are plotted in

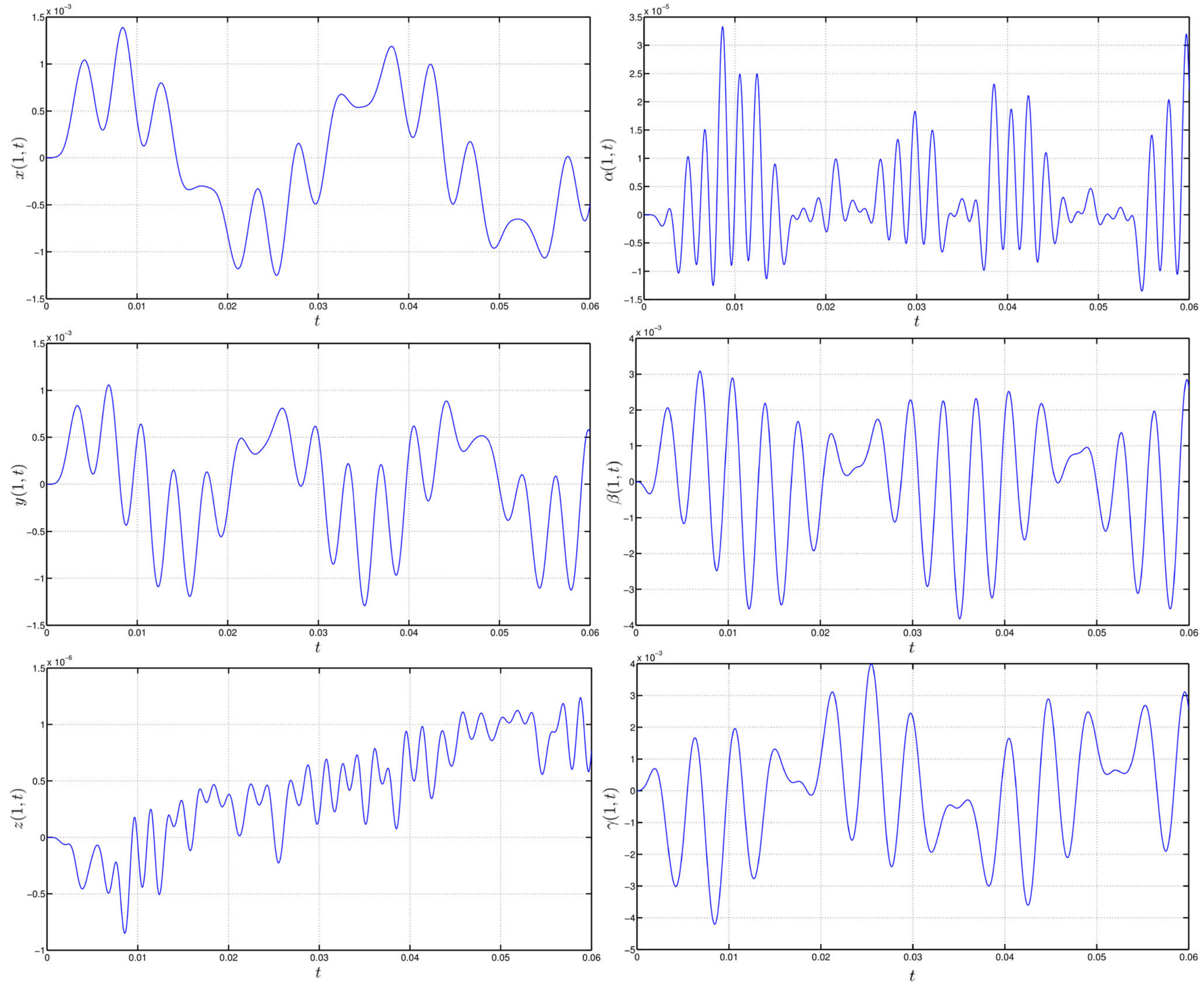


Fig. 5 Displacement and rotation at the tip of the cantilever beam

Figure 5, which quantitatively reproduces the results (Fig. 6 and Fig. 7) in [12].

5.2 Optimal control using data from a forward simulation

In this example, we modify the previous test of the cantilever beam so that it has similar material properties to *C. elegans* and undergoes a large deformation. As discussed in Section 3, we also neglect the inertia terms in equation (40) to model *C. elegans* locomotion. Our motivation is to generate a dataset to validate the proposed optimal control method. The new beam now has an initial length $L_0 = 10^{-3}m$ and circular cross section with radius $r_0 = L_0/40m$ [5]. The numerical correction factor for a circular cross section is taken to be $k = 4/3$ [34]. We adopt values for its Young's modulus $E = 1.1 \times 10^5 Pa$ and shear modulus $G = 5.0 \times 10^4 Pa$ [5].

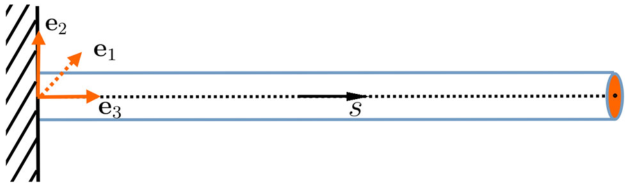


Fig. 6 Diagram of a cantilever beam

Before generating the dataset by applying a complicated external force and torque, we first apply a simple force F at the end of the beam, as shown in Figure 6, to validate the approach against the analytical solution based on the Timoshenko beam theory [34]:

$$y = -\frac{F}{kAG}s - \frac{FL_0}{2J_{11}}s^2 + \frac{F}{6J_{11}}s^3. \tag{63}$$

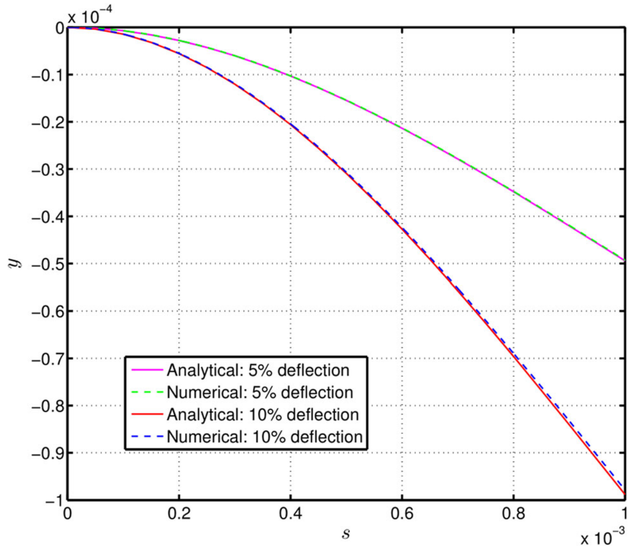


Fig. 7 Deflection of a cantilever beam under a concentrated force at the end of the beam. $F = 5 \times 10^{-9}N$ for the case of 5% deflection and $F = 10^{-8}N$ for 10% deflection

The rod is discretised by 100 segments (for which the mesh has converged), and we compute the deflection of the rod by solving the primal equation (52). It can be seen from Figure 7 that the result of the Cosserat model agrees very well with the prediction of the Timoshenko theory for up to 10% deflection of the rod.

The first test of the proposed control algorithm is to use a dataset generated by a distributed force along the rod:

$$f_2^l = -F_{\max}\left(e^{4z/L_0} - 1\right)/3, \quad f_1^l = f_3^l = 0, \tag{64}$$

with $F_{\max} = 10^{-4}$. The beam undergoes a large deformation as shown in Figure 8 (left); meanwhile a curvature κ_1^l (see formula (28)) is also generated along the rod. Using the proposed control formulation in Section 4 and control parameters of $\lambda_f = 1$, $\lambda_l = 10^{-10}$, $\lambda_d = 10^{-6}$, $\lambda_{\kappa_3} = 10^{-10}$ ($\lambda_{\kappa_1} = \lambda_{\kappa_2} = 0$) and $\boldsymbol{\Lambda}_\omega = \mathbf{0}$, both the position and the curvature can be recovered accurately as shown in Figure 8. Note that all the other components of the position vector and generalised curvature are zero although they are not presented here. Before moving to other test cases, let us test the convergence of the objective $\|\mathbf{x} - \mathbf{x}_g\|$, the output curvature $\|\boldsymbol{\kappa}^l - \boldsymbol{\kappa}_f^l\|$ ($\boldsymbol{\kappa}_f^l$ is from the forward simulation), the tangential direction $\|\partial_s\mathbf{x} - \mathbf{d}_3\|$ as well as the algorithm itself measured by the relative error of $(\mathbf{x}, \boldsymbol{\alpha})$ between the current and previous fixed-point iterations, with regards to the control parameters λ_f, λ_l and λ_d.

The main findings are summarised as follows: (1) If we only use $\mathbf{f}$ (notice that this does *not* mean $\lambda_l = 0$; we have to remove $\mathbf{l}$ term in (61)) as the control, the proposed algorithm struggles to converge no matter how we play with the other parameters. Therefore, the regularization λ_l- term in (41) does play an important stabilisation role, although $\mathbf{l} = \mathbf{0}$ when we generate the dataset; (2) If we plot the above convergence measures as shown in Figure 9 ($\lambda_f = 1$, $\lambda_d = 10^{-6}$ and $\lambda_{\kappa_3} = 10^{-20}$) and vary λ_l from magnitude 10^{-20} to 10^3, we find that these convergence curves are exactly the same. The only difference we notice is that the magnitude of the adjoint variable $\delta\boldsymbol{\alpha}$ varies correspondingly from 10^{-13} to 10^{10} so that the control torque $\mathbf{l} = \delta\boldsymbol{\alpha}/\lambda_l$ always has a magnitude of 10^{-7}. Therefore, the algorithm (at least for this test) is not sensitive to the regularisation parameter λ_l, although it is required to stabilise the algorithm as pointed out above; (3) The proposed algorithm can converge stably with the regularisation parameter λ_d varying from 10^1 to 10^{-10}, and the convergence of $\|\boldsymbol{\kappa}^l - \boldsymbol{\kappa}_f^l\|$ and $\|\partial_s\mathbf{x} - \mathbf{d}_3\|$ is plotted in Figure 10 with $\lambda_f = 1$, $\lambda_l = 10^{-10}$ and $\lambda_{\kappa_3} = 10^{-20}$; (4) The proposed algorithm converges for a range of parameters λ_f from 10^{-10} to 10^6. Despite the steady convergence of the algorithm, the value of λ_f must be sufficiently small for the objectives to be sufficiently reduced. To demonstrate this, convergence plots for the relevant quantities given two extreme values (10^{-10} and 10^6) of λ_f, are compared in Figure 11; (5) The purpose of the λ_{κ_3} parameter is to control twist along the rod. In this test, the algorithm can converge stably and the curvature error can be reduced sufficiently with λ_{κ_3} from 1 to 10^{-15}.

We next consider a dataset generated by the following force and torque, which creates both curvature and torsion along the rod as shown in Figure 12.

$$f_1^l = -2.2F_{\max}, \quad f_2^l = f_3^l = 0, \quad l_1^l = 10^{-7}, \quad l_2^l = l_3^l = 0. \tag{65}$$

Again we test all the regularisation parameters systematically, and our findings are summarised as follows: (1) λ_l is necessary for stability, and it can be chosen from a magnitude of 10^{-30} to 10^{-2} (based on a test with $\lambda_f = 1$, $\lambda_d = 10^{-6}$ and $\lambda_{\kappa_3} = 10^{-20}$); (2) the recommended value for λ_d ranges between 10^{-10} and 10^3, otherwise the objective cannot be sufficiently reduced when $\lambda_d > 10^3$, or the algorithm cannot converge when $\lambda_d < 10^{-10}$ (based on a test with $\lambda_f = 1$, $\lambda_l = 10^{-6}$ and $\lambda_{\kappa_3} = 10^{-20}$); (3) the proposed algorithm can converge steadily for a range of λ_f from 10^{-30} to 10^2, and the suggested values are $\lambda_f < 1$, otherwise the objective cannot be reduced sufficiently (based on test of $\lambda_l = 10^{-6}$, $\lambda_d = 10^{-6}$ and $\lambda_{\kappa_3} = 10^{-20}$); (4) The algorithm converges stably with λ_{κ_3} for magnitude of 10^{-15} to 1, but the recommended value is $> 10^{-25}$ otherwise a too large κ_3^l could be generated. A convergence of relevant quantities with a specific parameter set is plotted in Figure 13. The comparison of the position and curvature between the forward and backward computations are displayed in Figure 12 and 14 respectively.

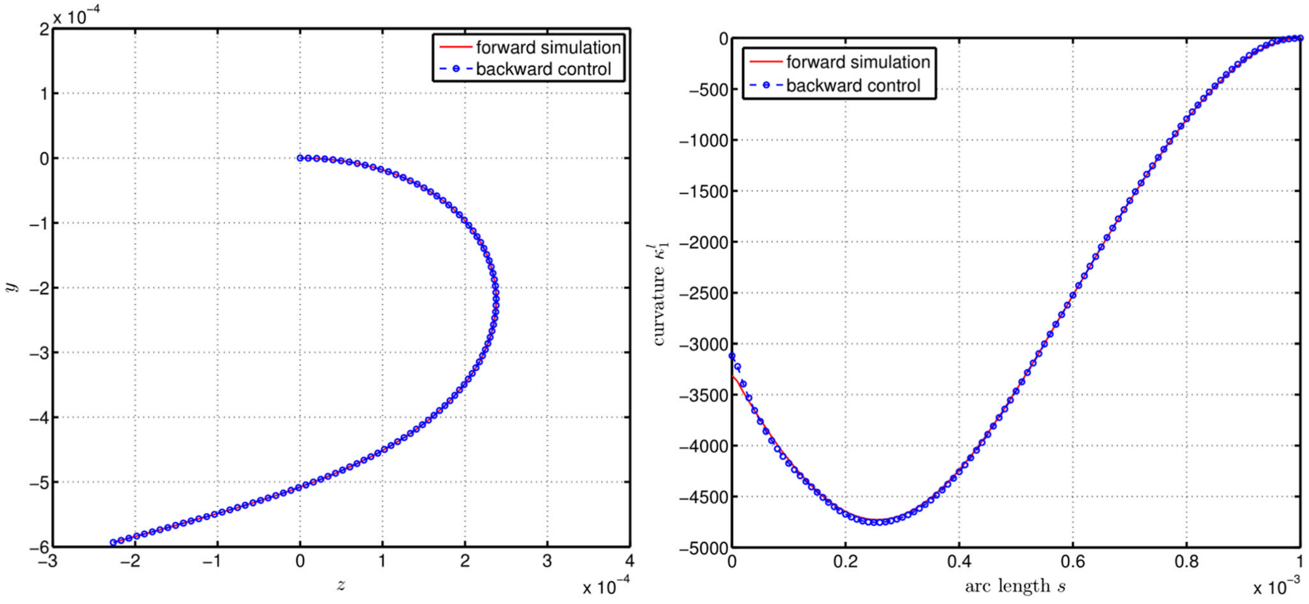


Fig. 8 Comparison of the position vector (left) and curvature (right) between forward and backward simulations for the first test case in 5.2

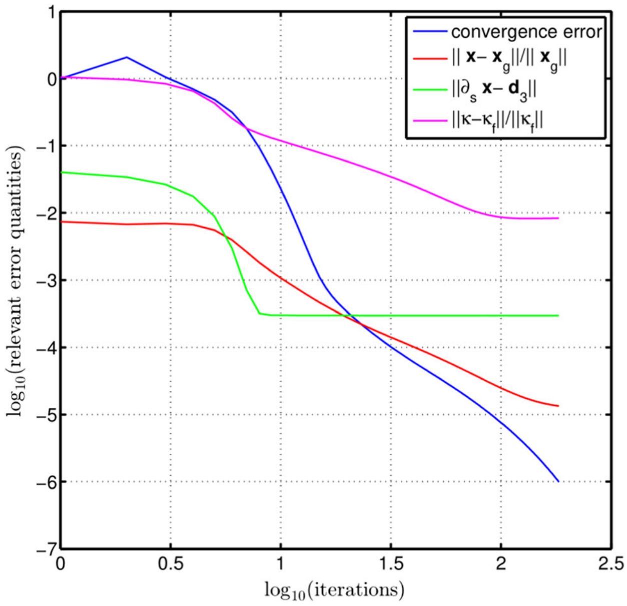


Fig. 9 Log-log plot of the convergence of relevant quantities as functions of the number of iterations for the first test case in 5.2

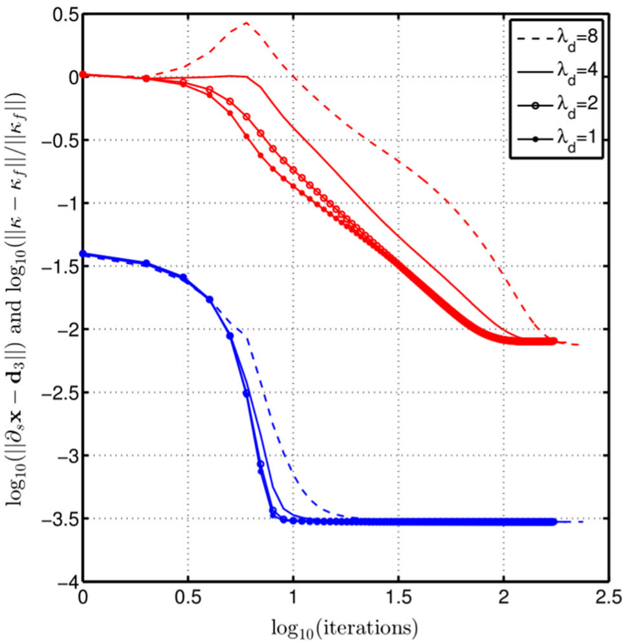


Fig. 10 Convergence of $\|\boldsymbol{\kappa}^l - \boldsymbol{\kappa}^l_f\|$ (red) and $\|\partial_s \mathbf{x} - \mathbf{d}_3\|$ (blue) in terms of regularisation parameter λ_d for the first test case in 5.2

It can be seen that the positions between the forward and backward simulations match very well along the rod, and the curvatures also match well except the end where the Dirichlet boundary condition is applied.

Remark 11 We find that non-trivial forward data, involving large bend and torsion for example, is not easy to generate, because it is not straightforward to provide or design a force **f** or torque **l** so that the forward problem can converge easily. While once a dataset is given, the control problem is easy to converge – converging to the same position vector **x** and rotation $\boldsymbol{\alpha}$ (as the forward simulation results) even with a different control force and torque. This can be understood and by noting that we do not expect that the force and torque (producing the same **x** and $\boldsymbol{\alpha}$) are unique. As an example, we show the control force and torque for the previous test in Figure 15, from which it can be seen that the magnitude

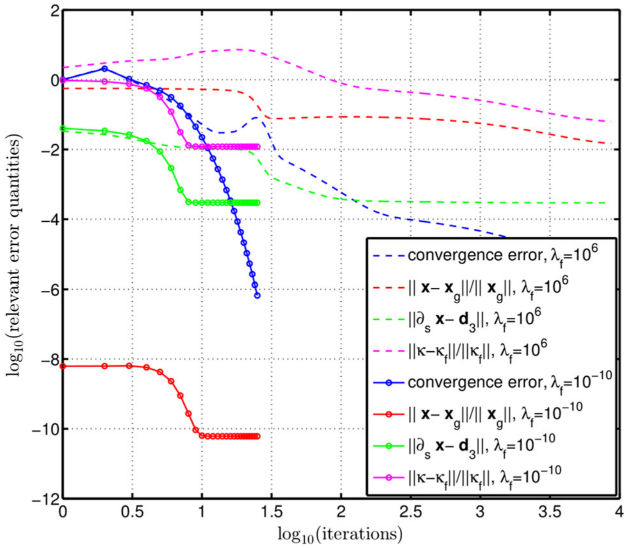


Fig. 11 Comparison of convergence between two extreme values of λ_f for the first test case in 5.2

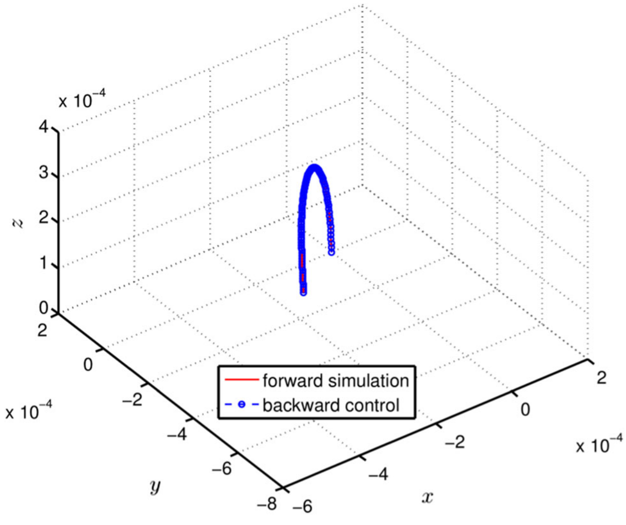


Fig. 12 Comparison of the position vector between forward and backward simulations using (65) for the second test case in 5.2

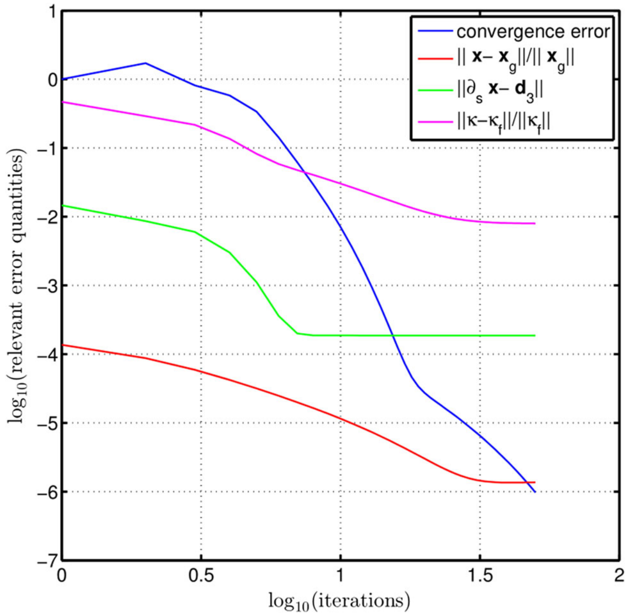


Fig. 13 Convergence of different measures with $\lambda_f = 10^{-2}$, $\lambda_l = 10^{-6}$, $\lambda_d = 10^{-6}$ and $\lambda_\kappa = 10^{-20}$ for the second test case in 5.2

is similar to the designed one in (65), but the distribution is different.

6 Reconstruction of *C. elegans* locomotion based on experimental data

We tested the proposed method on three examples of *C. elegans* locomotion in 3D volumes [72]. The data represent the body-midlines that were reconstructed from microscopy-video footage of freely and spontaneously moving worms that were immersed in different fluids. We present one test case in this section and all the three tests (including the dataset, FreeFem++ code and simulation results) can be found from public GitHub repository: https://github.com/yongxingwang. The first test case consists of a sequence of 1160 time frames with a sampling interval of $0.04s$ (as inertia is not considered in our model and the time step only appears in the regularisation term in (41)), and each reconstructed body centerline consists of 128 discrete three-dimensional spatial points. These examples are of interest due to the three-dimensional postures and motion of the swimmer: in this clip, the worm exhibits large bend, large torsion and moves forward and backward in a three-dimensional space for about $46s$. The physical body of the worm is modelled by a cylindrical Cosserat rod with a circular cross section of initial radius $2 \times 10^{-5}m$, Young's modulus $E = 1.1 \times 10^5 Pa$ and shear modulus $E/(1+\nu)/2Pa$ with $\nu = 0.4$ [5,20]. Four typical postures of the worm are shown in Figure 16: the worm initially moves from the right to the left and starts a manoeuvre to reverse its motion at around the 400^{th} time frame; after another 350 steps, the worm suddenly bends to resemble a capital Ω (left-bottom in Figure 16) and moves to the right.

To construct the first frame, we set $\alpha_a = 0$ and $\boldsymbol{\Lambda}_\omega = 0$; and from the second frame, we use a non-zero $\boldsymbol{\Lambda}_\omega$ (at least non-zero $\lambda_{\omega 3}$ for the sake of convergence) without any Dirichlet data. Three components of generalised curvature are plotted along the worm's body for all the time frames in a two-dimensional plane as shown in Figure 17, from which a bending (κ_1^l and κ_2^l) wave can be seen propagating from the worm's head to the tail; the twisting (κ_3^l) wave is not obvious but some twisting can still be observed. The propagated wave is consistent with the moving direction of the worm as shown in Figure 16 and analysed in the above.

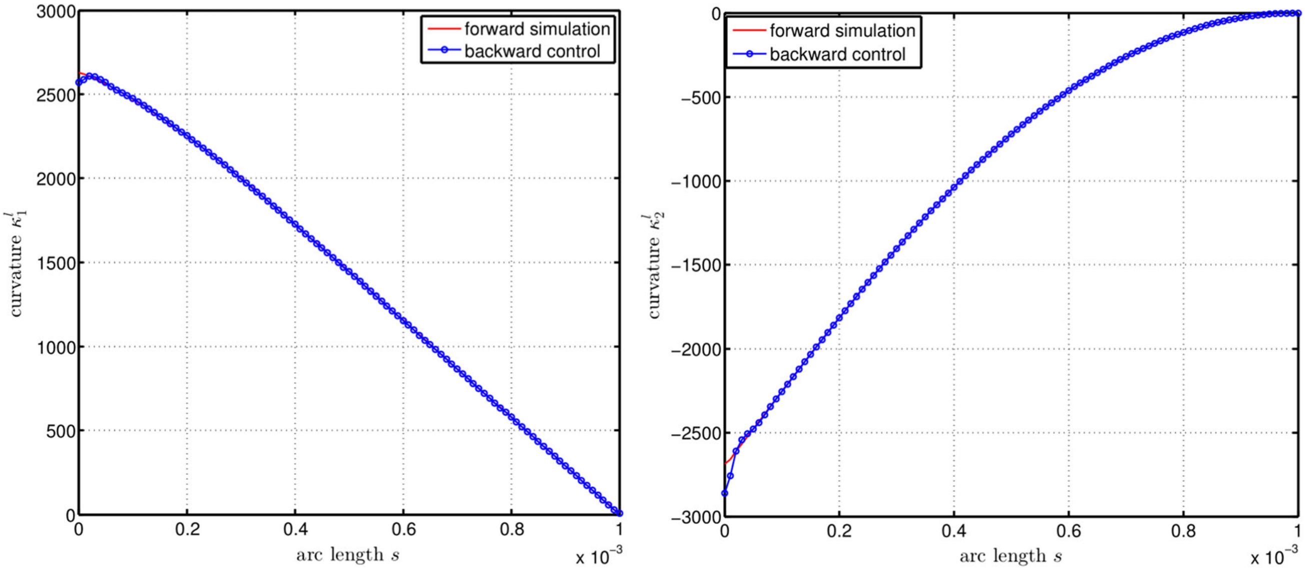
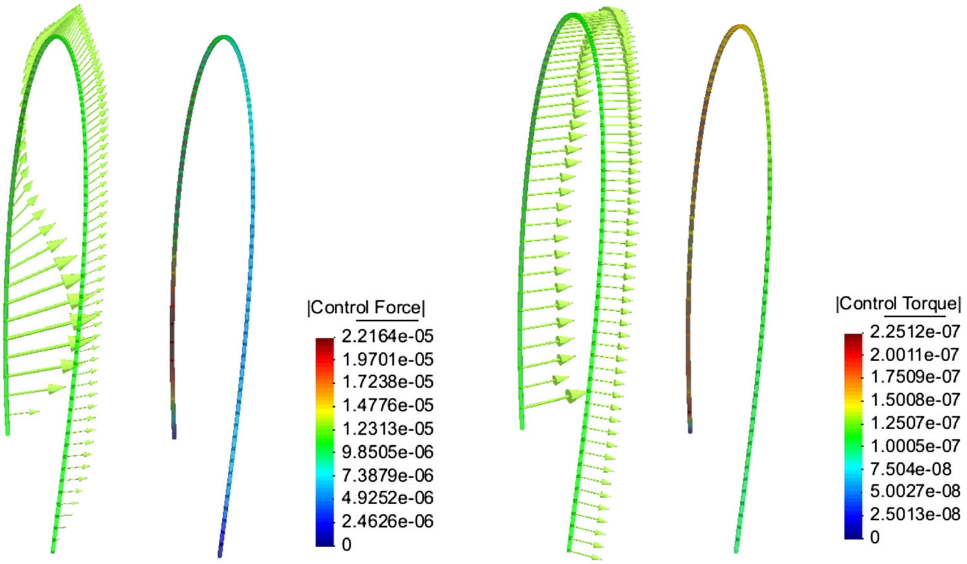


Fig. 14 Comparison of the curvature between forward and backward simulations for the second test case in 5.2

Fig. 15 Control force (left) and torque (right) for the second test case in 5.2



Starting from a converged parameter set as shown in Table 1 (for the results in Figures 16 and 17), with the converged objectives in (41) being shown in Figure 18, we vary these parameters, study the convergence of the algorithm and compare corresponding results in the following. Notice that this set of parameters has the minimal non-zero parameters to make sure the proposed algorithm can converge (please refer to Remark 5 and 6 and Section 3.4 for explanations).

Parameter λ_f: with other parameters frozen, the proposed algorithm converges stably for λ_f from 10^{-20} to 10; the fixed-point iteration becomes slower for larger λ_f. We find that all the objectives stay the same except the control $\mathbf{f}$ (see Figure 19).

Parameter λ_l: we then keep $\lambda_f = 10^{-20}$ and the proposed algorithm still converges stably with the magnitude of λ_l from 10^{-10} to 1. We observe that all the objectives are still almost the same except the control $\mathbf{l}$ as shown in Figure 19.

Parameter λ_d: based on the above two tests, we realise that the total objective function (41) is dominated by the λ_d-term. With other parameters frozen, we find that the convergence range for λ_d is approximately between 10^{-2} and 10^3. A comparison of converged objectives between Parameter-0 in Table 1 and its variation case with $\lambda_d = 10^{-2}$ (reduced from $\lambda_d = 10^2$) is plotted in Figure 20, from which it can be seen that (i) the real objective $\|\mathbf{x} - \mathbf{x}_g\|/\|\mathbf{x}_g\|$ is reduced by two orders of magnitude, with oscillations for some frames which is expected for such a small regularisation parameter; (ii) the

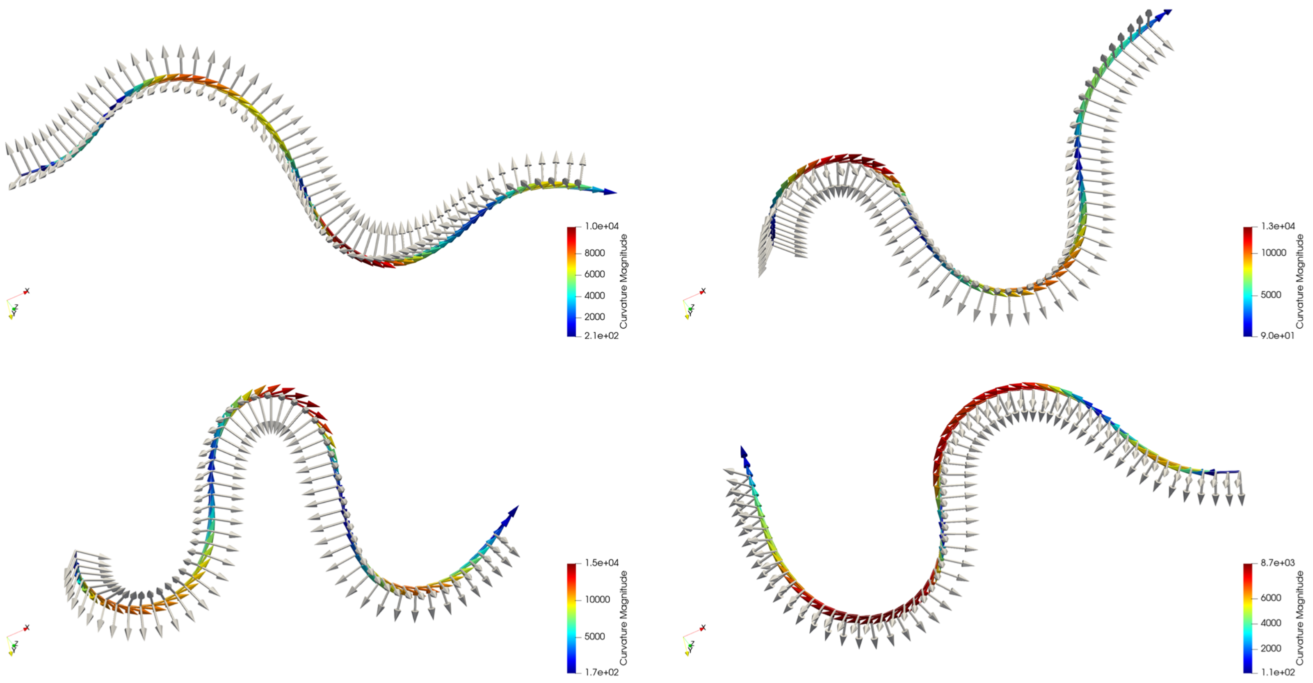


Fig. 16 Four typical postures of *C. elegans*. The arrows show the local frames and the colourful one is $\mathbf{d}_3$ pointing from the worm's head to tail, with colour showing the magnitude of generalised curvature. (Color figure online)

Table 1 Parameter-0: minimal non-zero parameters for the sake of convergence of the proposed algorithm

λ_f	λ_l	λ_d	$\boldsymbol{\Lambda}_\kappa$	$\boldsymbol{\Lambda}_\omega$
10^{-3}	10^{-6}	10^{2}	$\operatorname{diag}(0, 0, 0)$	$\operatorname{diag}\left(0, 0, 10^{-20}\right)$

λ_d−term increases as its regularisation parameter decreases from $\lambda_d = 10^2$ to $\lambda_d = 10^{-2}$. We notice that the magnitude of $\|\partial_s \mathbf{x} - \mathbf{d}_3\|/\|\mathbf{d}_3\|$ increases to 10^{-1} for some frames, which means $\mathbf{d}_3$ detaches from the tangential $\partial_s \mathbf{x}$ of the centreline 10%. For example, Figure 21 shows frame number 700 where the normal direction $\mathbf{d}_3$ of the cross section detaches from the tangential direction $\partial_s \mathbf{x}$ of the centreline; (iii) none of the other objective terms show a significant change except the control $\mathbf{f}$ which varies according to $\|\mathbf{x} - \mathbf{x}_g\|/\|\mathbf{x}_g\|$ as expected.

Remark 12 The detachment of $\mathbf{d}_3$ from $\partial_s \mathbf{x}$ is an important feature of Cosserat rods. Otherwise, the Cosserat rod approaches to the Kirchhoff rod (if the deformation scaler $j = 1$, which is true for our case study system of *C. elegans*: we observe that $|j - 1| < 10^{-3}$ always holds numerically), in which case it is assumed that $\mathbf{d}_3 = \partial_s \mathbf{x}$ [31].

Remark 13 In keeping $\lambda_d = 10^{-2}$ (which now cannot dominate the total objective function of (41)) and varying λ_f and λ_l, we again can observe a variation of other objective terms although we would not present all these tests here. However, too small regularisation parameters can cause stability issues as we have already seen when reducing λ_d from 10^2 to 10^{-2} although the algorithm still converged.

The parameter $\boldsymbol{\Lambda}_\kappa$ provides a constraint of the curvature along the worm's body, which allows us to consider the anatomical muscle structure of *C. elegans* as explained in Section 3.4. Based on Parameter-0, we now simply choose $\boldsymbol{\Lambda}_\kappa = \operatorname{diag}\left(0, 0, 10^{-20}\right)$ to restrict the twist motion of the worm, and the generalised curvature is plotted in Figure 23. Comparing with Figure 17, we can see that not only does the magnitude of κ_3^l dramatically decrease, but a clear twisting wave also appears along the worm. In addition, there is also an influence on the first two components κ_1^l and κ_2^l: the wave propagation is clearer although the curvature magnitudes are almost the same as the case of $\boldsymbol{\Lambda}_\kappa = \mathbf{0}$. We also notice that the reversing manoeuvre (starting at around frame 700) becomes more distinct: κ_3^l is much larger near the worm's head at frame 700 than elsewhere or any other frames, as can be observed from Figure 22 and 23. Similarly, using non-zero λ_{κ_1} or λ_{κ_2} would allow us to favour bending in the dorsal-ventral directions which is consistent with the *C. elegans* neuromusculature.

$\boldsymbol{\Lambda}_\omega$ is used to model the internal friction of the worm which is necessary from the second time frame. For the sake of convergence of our algorithm, only λ_{ω_3} is required. If we increase $\boldsymbol{\Lambda}_\omega$ in Parameter-0 from $\operatorname{diag}\left(0, 0, 10^{-20}\right)$ to some value less than $\operatorname{diag}(1, 1, 1)$, the proposed algorithm converges stably and the angular velocity stays almost the same as shown in Figure 24. However, there is a big change in the

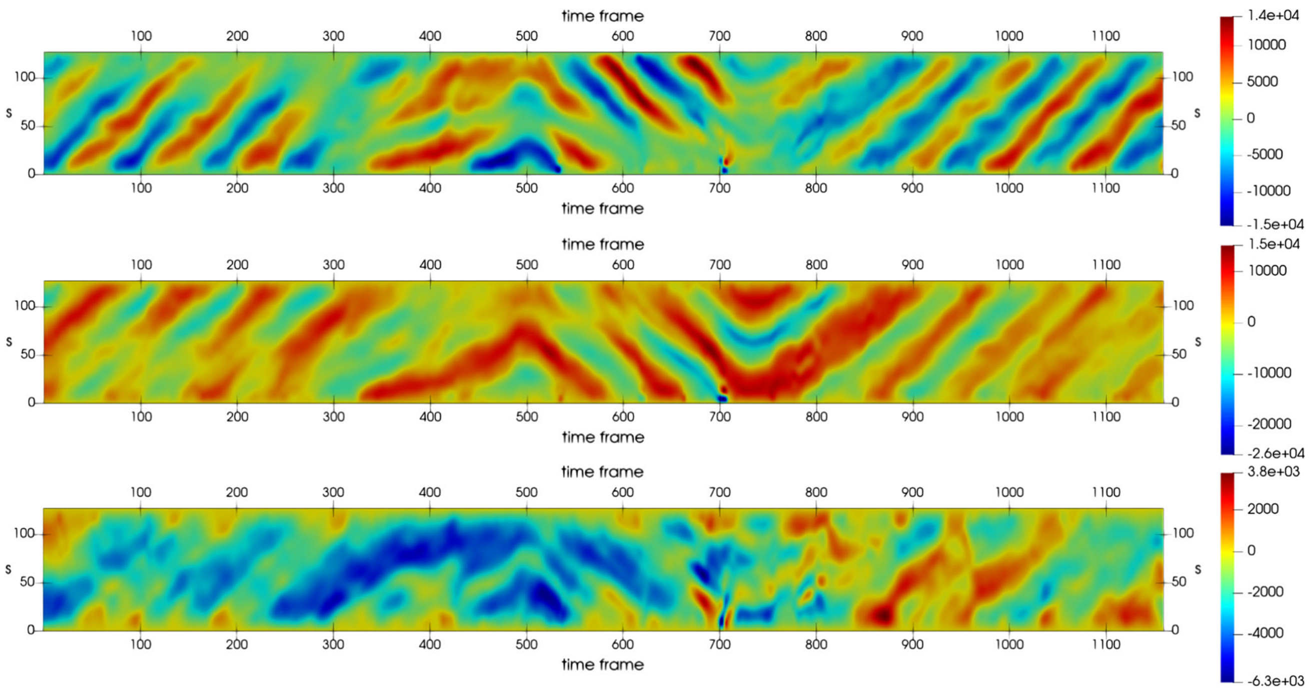
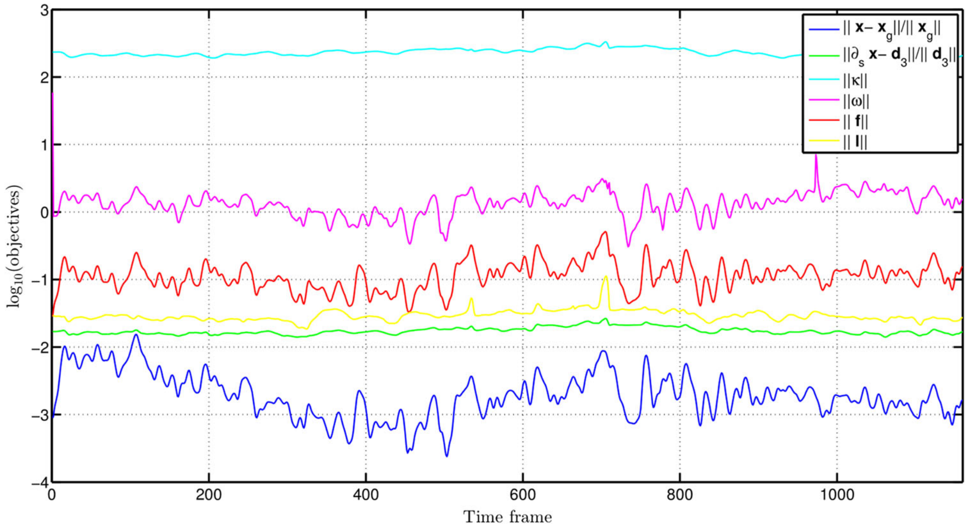


Fig. 17 Local curvature κ_1^l, κ_2^l, κ_3^l (from top to bottom) along the *C. elegans* body centerline (vertical axis: 0 = head, 127 = tail) as a function of time (horizontal axis), as computed from data (clip 1) for all the time frames. $\lambda_f = 10^{-3}$, $\lambda_l = 10^{-6}$, $\lambda_d = 10^2$, $\boldsymbol{\Lambda}_\kappa = \mathbf{0}$ and $\boldsymbol{\Lambda}_\omega = \text{diag}\left(0, 0, 10^{-20}\right)$

Fig. 18 Objective terms in (41) as a function of time, using $\lambda_f = 10^{-3}$, $\lambda_l = 10^{-6}$, $\lambda_d = 10^2$, $\boldsymbol{\Lambda}_\kappa = \mathbf{0}$ and $\boldsymbol{\Lambda}_\omega = \text{diag}\left(0, 0, 10^{-20}\right)$



curvature (see Figure 24 and Figure 25): wave propagation is no longer apparent because larger $\boldsymbol{\Lambda}_\omega$ tends to keep the rotation angles (along the body and overtime) the same in time, consequently the space derivative (curvature) along the worm's body does not change significantly in time.

7 Conclusion and discussion

This paper presents three contributions: the forward formulation of a Cosserat rod, the optimal control method and the reconstruction of *C. elegans* locomotion.

The forward formulation of Cosserat rod is developed from [12], in which the Cosserat rod is described by three components of the position vector (x, y, z) and three components of the rotation vector (α, β, γ). We derive the angular velocity and generalised curvature using a new method in Section 2.3 and rewrite all the control equations in a matrix-

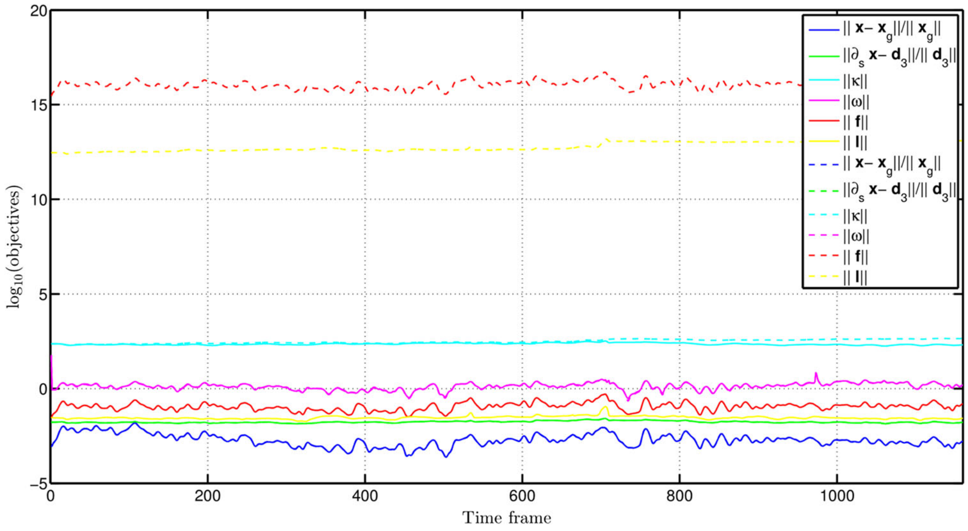


Fig. 19 Objective terms in (41) as a function of time; solid lines (—) are the initial parameter set as shown in Figure 18 and dashed lines (- - -) are a variation of λ_f and λ_d in the initial parameter set with $\lambda_f = 10^{-20}$ and $\lambda_l = 10^{-10}$

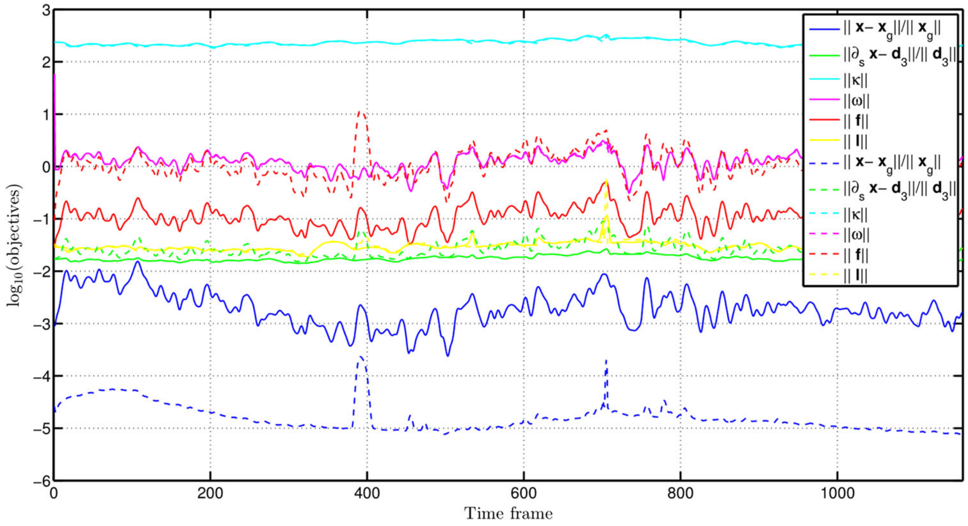
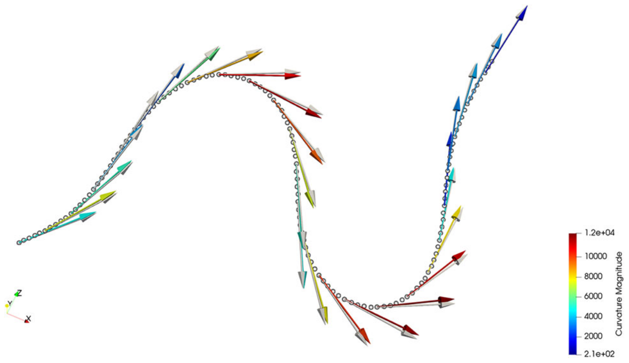


Fig. 20 Objective terms in (41) as a function of time; solid lines (—) are the initial parameter set as shown in Figure 18 and dashed lines (- - -) are a variation of λ_d in the initial parameter set with $\lambda_d = 10^{-2}$



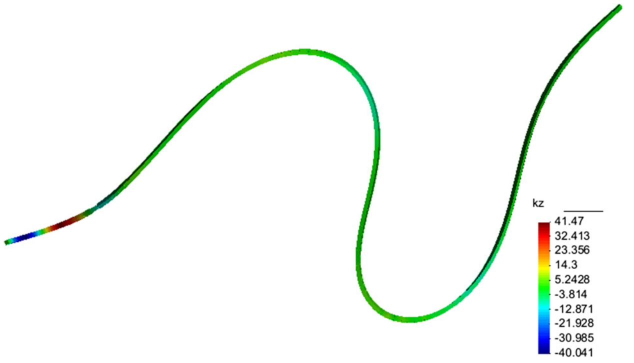


Fig. 21 Frame NO. 700 using $\lambda_d = 10^{-2}$, where the normal direction $\mathbf{d}_3$ of the cross section detaches from the tangential direction $\partial_s \mathbf{x}$ of the centreline: $\mathbf{d}_3$ in white colour and $\partial_s \mathbf{x}$ is colourful showing the magnitude of the generalised curvature. (Color figure online)

Fig. 22 Frame NO. 700 using $\boldsymbol{\Lambda}_\kappa = \mathrm{diag}\left(0, 0, 10^{-20}\right)$, where the worm undergoes a strong twist at its head: the magnitude of κ_3^l is much larger at the worm's head than other place along the body

vector format; in addition, we consider dilation of the cross section of the rod by differentiation of the reference and current arc lengths and derivation of the incompressibility condition in Section 2.4. We define a forward problem: to solve for the position vector and rotation vector given exter-

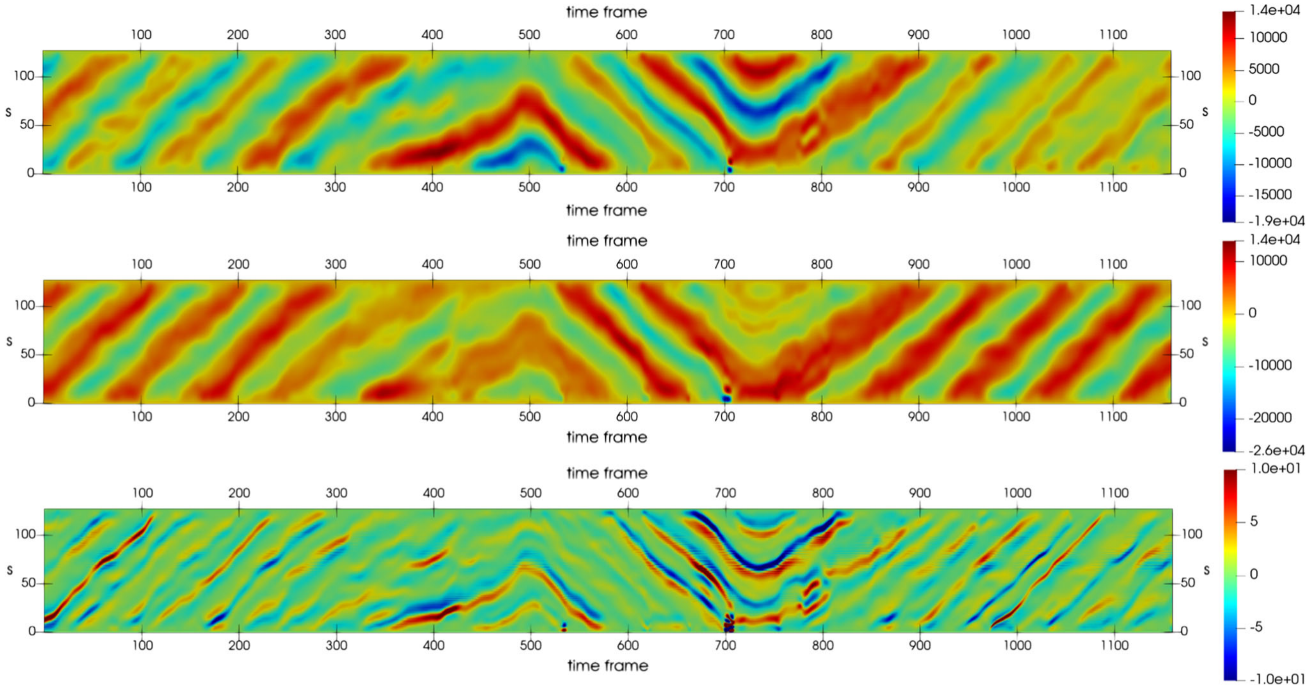


Fig. 23 Local curvature $\kappa_1^l, \kappa_2^l, \kappa_3^l$ (from top to bottom) along the *C. elegans* body centerline (vertical axis: 0 = head, 127 = tail) as a function of time (horizontal axis), as computed from data (clip 1) for all the time frames. $\lambda_f = 10^{-3}$, $\lambda_l = 10^{-6}$, $\lambda_d = 10^2$, $\boldsymbol{\Lambda}_\kappa = \text{diag}\left(0, 0, 10^{-20}\right)$ and $\boldsymbol{\Lambda}_\omega = \text{diag}\left(0, 0, 10^{-20}\right)$

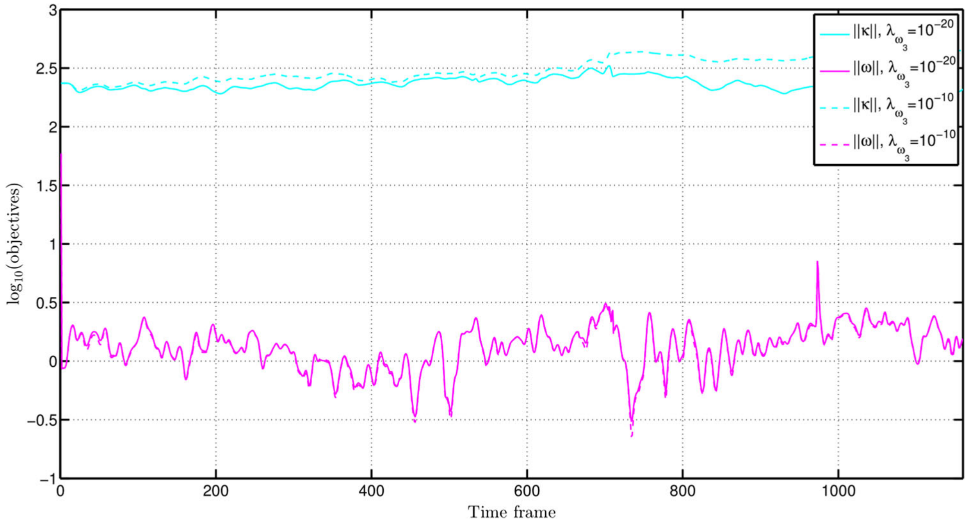


Fig. 24 Converged curvature and angular velocity as a function of time, using $\lambda_f = 10^{-3}$, $\lambda_l = 10^{-6}$, $\lambda_d = 10^2$, $\boldsymbol{\Lambda}_\kappa = \mathbf{0}$ and $\boldsymbol{\Lambda}_\omega = \text{diag}\left(0, 0, 10^{-10}\right)$

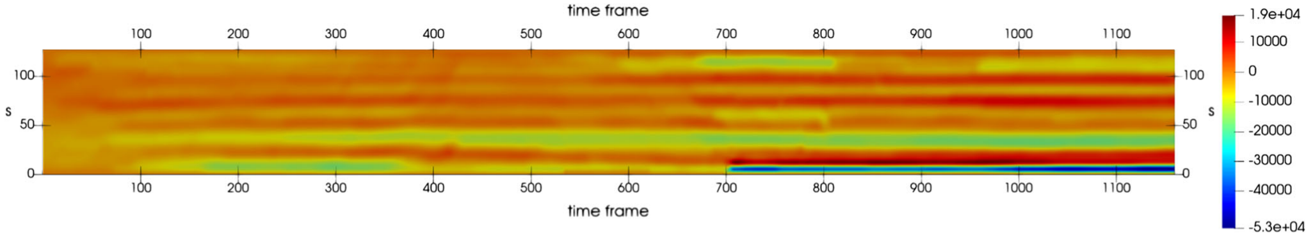


Fig. 25 Local curvature κ_3^l along the *C. elegans* body centerline (vertical axis: 0 = head, 127 = tail) as a function of time (horizontal axis). $\lambda_f = 10^{-3}$, $\lambda_l = 10^{-6}$, $\lambda_d = 10^2$, $\boldsymbol{\Lambda}_\kappa = \text{diag}\left(0, 0, 10^{-20}\right)$ and $\boldsymbol{\Lambda}_\omega = \text{diag}\left(0, 0, 10^{-10}\right)$

nal forces and torques. We have reproduced the numerical examples in [12] and found that this formulation is robust and convenient for analysis of complex dynamic behaviour of slender rods.

A well-posed inverse problem may be solving for external forces and torques, given *both* the position vector and rotation vector. However, accessing both position and rotational information may not be practical. For example, for a biological worm moving freely in a fluid environment, it is difficult to measure the worm's local orientation (rotation vector) while its centreline (position vector in the global frame) can be reconstructed from video footage [72]. A complementary problem that may be tackled with an analogous approach considers a robotic worm exploring an unknown space. Given sufficient local body sensors, the body posture (including bending and twisting) would be reliably detected by such a robot. However, in the absence of external location data, the robot may lack positional information. Motivated by these biological and engineering problems, we consider an ill-posed inverse problem which solves for rotation vector, external forces and torques given *only* the position vector. We present a robust and efficient optimal control method to solve this inverse problem: the objective is to minimise the discrepancy between the position vector and a given centreline of the Cosserat rod, and the control variables are the external forces and torques, with regularisations of the rotation vector. The regularisation terms provide constraints of the rotation vector so that the inverse problem is solvable. We have tested the proposed optimal control formulation using data from forward simulations and shown that the rotation vector can be accurately computed with appropriate and controllable regularisation parameters.

The proposed optimal control is applied to reconstruction of *C. elegans* locomotion based upon its centreline data from laboratory recordings. The solvability of this challenging inverse problem relies on meaningful regularisation terms. The proposed approach allows us to add different terms conveniently to model *C. elegans*' neuromusculature, and our inverse model is demonstrated to be robust to a range of regularisation parameters. There are five parameters (nine if considering components of $\boldsymbol{\Lambda}_\kappa$ and $\boldsymbol{\Lambda}_\omega$) as shown in Table 1, which also indicates the minimally-required non-zero parameters for the sake of convergence of our proposed method.

λ_f and λ_l correspond to the control force $\mathbf{f}$ and torque $\mathbf{l}$ respectively, which can stop $\|\mathbf{f}\|$ and $\|\mathbf{l}\|$ becoming infinite and have the effect of stabilising the proposed method. λ_f and λ_l can be robustly chosen from a range of values based on our numerical experiments, without a significant influence on the main outputs such as the centreline $\mathbf{x}$ and curvature $\boldsymbol{\kappa}$.

λ_d has a biological and anatomical grounding because it keeps the normal direction $\mathbf{d}_3$ of the worm's cross section close to the tangential direction $\partial_s \mathbf{x}$ of its body's centreline. It also has a numerical effect of stabilising the proposed method and differentiating the Cosserat rod and Kirchhoff rod models: larger λ_d tends to force $\mathbf{d}_3$ to be the same as $\partial_s \mathbf{x}$ (approaching the Kirchhoff rod consequently).

λ_{ω_3} in $\boldsymbol{\Lambda}_\omega = \operatorname{diag}\left(\lambda_{\omega_1}, \lambda_{\omega_2}, \lambda_{\omega_3}\right)$ has a clear numerical purpose, because our method needs it to be non-zero for convergence. The other two components of $\boldsymbol{\Lambda}_\omega$ and all the three components of $\boldsymbol{\Lambda}_\kappa = \operatorname{diag}\left(\lambda_{\kappa_1}, \lambda_{\kappa_2}, \lambda_{\kappa_3}\right)$ can be zero. However, setting $\boldsymbol{\Lambda}_\omega$ and $\boldsymbol{\Lambda}_\kappa$ to non-zero values allows us to model the muscular of *C. elegans* as pointed out in Section 3.4.

Several interesting topics have been stimulated by this study, which are briefly summarised as follows:

The proposed optimal control formulation is based on a combination of laboratory data and modelling of the worm's muscle structure. If we can collect the data of at least one cross section's movement of *C. elegans*, we can then apply a Dirichlet boundary condition of $\boldsymbol{\alpha}$ and use less regularisation terms as commented in Remark 6 (λ_{ω_3} can be zero then). Measuring the movement of cross sections of a hair-thin *C. elegans* in laboratory is technically difficult. However, setting a mark and following one cross section may be possible and would provide a possibility to validate the predictions of our inverse model based on the above assumptions.

Having computed the local frames (rotation vectors), we can then formulate these rotations into the objective function, and compute the external force $\mathbf{f}$ and torque $\mathbf{l}$ without regularisation λ_d-, $\lambda_\omega-$, and $\lambda_\kappa-$ terms. These force and torque terms will provide us *C. elegans*' muscle force quantitatively, which will help us to model and understand its neuromuscular system.

One more interesting topic is modelling time evolution. In this paper, a friction term ($\lambda_\omega-$ term) is introduced to link different time frames. An alternative approach is to introduce a viscoelastic constitutive model as adopted in [53], which is expected to more appropriate for modelling nematodes locomotion [5].

Another way to formulate the underlining problem is to apply a model for the force $\mathbf{f}$, such as slender body theory [49], then only use $\mathbf{l}$ as a control variable. Hopefully, this would lead to a well-posed problem without additional regularisation terms.

Acknowledgements The authors would like to thank Peter Bollada and Lukas Deutz for discussion of Cosserat rod theory which have improved the writing of the manuscript.

Declarations

Conflicts of interests The authors declare no conflict of interest.

Replication of results The simulation data and FreeFem++ code are shared at https://github.com/yongxingwang in order to reproduce the results presented in the paper.

A Time discretisation

We introduce a time discretisation scheme for weak form (36) or (40). Let $\mathbf{M}$ and $\mathbf{S}$ be the mass and stiff matrix respectively after spacial discretisation, and $\mathbf{F}$ be force vector, the spatial discretisation of (36) or (40) leads to the following algebra partial differential system:

$$\partial_t \mathbf{v} + \mathbf{S}\mathbf{u} = \mathbf{F}, \tag{66}$$

with

$$\mathbf{u} = \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\alpha} \end{pmatrix}, \quad \mathbf{v} = \partial_t \left(\mathbf{M}\mathbf{u}\right). \tag{67}$$

If the time domain is disretised as $t_0 = 0, t_1, \ldots$ with a time frame of $\Delta t = t_n - t_{n-1}$, then (66) may be discretised as follows.

$$\frac{\mathbf{v}_n - \mathbf{v}_{n-1}}{\Delta t} + \mathbf{S}_n \frac{\mathbf{u}_n + \mathbf{u}_{n-1}}{2} = \mathbf{F}_n, \tag{68}$$

with

$$\frac{\mathbf{v}_n + \mathbf{v}_{n-1}}{2} = \frac{\mathbf{M}_n \mathbf{u}_n - \mathbf{M}_{n-1}\mathbf{u}_{n-1}}{\Delta t}. \tag{69}$$

Rewrite (68) as

$$\frac{\mathbf{v}_n + \mathbf{v}_{n-1}}{\Delta t} - \frac{2\mathbf{v}_{n-1}}{\Delta t} + \mathbf{S}_n \frac{\mathbf{u}_n + \mathbf{u}_{n-1}}{2} = \mathbf{F}_n, \tag{70}$$

and substitute (69) into (70), we have the final linear algebra system:

$$\left[\mathbf{M}_n + \left(\frac{\Delta t}{2}\right)^2 \mathbf{S}_n\right] \mathbf{u}_n = \left[\mathbf{M}_{n-1} - \left(\frac{\Delta t}{2}\right)^2 \mathbf{S}_n\right] \mathbf{u}_{n-1} + \Delta t \mathbf{v}_{n-1} + \mathbf{F}_n, \tag{71}$$

with

$$\mathbf{v}_{n-1} = \frac{2\mathbf{M}_{n-1}\mathbf{u}_{n-1} - 2\mathbf{M}_{n-1}\mathbf{u}_{n-2}}{\Delta t} - \mathbf{v}_{n-2} \tag{72}$$

derived from (69).

References

1. Abergel F, Temam R (1990) On some control problems in fluid mechanics. Theoret Comput Fluid Dyn 1(6):303–325
2. Antman SS (2005) Problems in nonlinear elasticity. Nonlinear Problems of Elasticity pp 513–584
3. Attavino A, Cerroni D, Da Vià R, Manservisi S, Menghini F (2017) Adjoint optimal control problems for the rans system. In: Journal of Physics: Conference Series, vol. 796, p 012008. IOP Publishing
4. Aulisa E, Manservisi S (2006) A multigrid approach to optimal control computations for Navier-Stokes flows. In: Robust Optimization-Directed Design, pp 3–23. Springer
5. Backholm M, Ryu WS, Dalnoki-Veress K (2013) Viscoelastic properties of the nematode Caenorhabditis elegans, a self-similar, shear-thinning worm. Proc Natl Acad Sci 110(12):4528–4533
6. Bauchau OA, Craig JI (2009) Euler-Bernoulli beam theory. In: Structural Analysis, pp 173–221. Springer
7. Bazilevs Y, Takizawa K, Tezduyar TE (2013) Computational fluid-structure interaction: methods and applications. John Wiley & Sons, New York
8. Berri S, Boyle JH, Tassieri M, Hope IA, Cohen N (2009) Forward locomotion of the nematode C. elegans is achieved through modulation of a single gait. HFSP J 3(3):186–193
9. Biewener A, Patek S (2018) Animal Locomotion. Oxford University Press, United Kingdom
10. Bilbao A, Patel AK, Rahman M, Vanapalli SA, Blawzdziewicz J (2018) Roll maneuvers are essential for active reorientation of Caenorhabditis elegans in 3d media. Proc Natl Acad Sci 115(16):E3616–E3625
11. Brenner S (1974) The genetics of Caenorhabditis elegans. Genetics 77(1):71–94
12. Cao DQ, Tucker RW (2008) Nonlinear dynamics of elastic rods using the Cosserat theory: Modelling and simulation. Int J Solids Struct 45(2):460–477
13. Chirco L, Manservisi S (2019) An adjoint based pressure boundary optimal control approach for fluid-structure interaction problems. Comput Fluids 182:118–127
14. Chirco L, Manservisi S (2020) On the optimal control of stationary fluid-structure interaction systems. Fluids 5(3):144
15. Choi H, Moin P, Kim J et al (1994) Active turbulence control for drag reduction in wall-bounded flows. J Fluid Mech 262:75–75
16. Cicconofri G, DeSimone A (2019) Modelling biological and bio-inspired swimming at microscopic scales: recent results and perspectives. Comput Fluids 179:799–805
17. Cohen N, Boyle JH (2010) Swimming at low reynolds number: a beginners guide to undulatory locomotion. Contemp Phys 51(2):103–123
18. Cosserat EMP (1970) Theory of deformable bodies. National Aeronautics and Space Administration
19. Davis R, Henshell R, Warburton G (1972) A Timoshenko beam element. J Sound Vib 22(4):475–487
20. Denham JE, Ranner T, Cohen N (2018) Signatures of proprioceptive control in Caenorhabditis elegans locomotion. Philos Trans R Soc B: Biol Sci 373(1758):20180208
21. Dill EH (1992) Kirchhoff's theory of rods. Archive for History of Exact Sciences pp 1–23
22. Dong M, Liao J, Du Z, Huang W (2020) Influences of lateral jet location and its number on the drag reduction of a blunted body in supersonic flows. Aeronaut J 124(1277):1055–1069
23. Elishakoff I (2020) Handbook on Timoshenko-Ehrenfest beam and Uflyand-Mindlin plate theories. World Scientific, Singapore

24. Euler L (1980) The rational mechanics of flexible or elastic bodies 1638-1788: introduction to Vol. X and XI. Springer Science & Business Media, Germany
25. Fang-Yen C, Wyart M, Xie J, Kawai R, Kodger T, Chen S, Wen Q, Samuel AD (2010) Biomechanical analysis of gait adaptation in the nematode Caenorhabditis elegans. Proc Natl Acad Sci 107(47):20323–20328
26. Fattorini H, Sritharan S (1992) Existence of optimal controls for viscous flow problems. Proc R Soc Lond A 439(1905):81–102
27. Feng Z, Cronin CJ, Wittig JH, Sternberg PW, Schafer WR (2004) An imaging system for standardized quantitative analysis of C. elegans behavior. BMC Bioinf 5(1):1–6
28. Fursikov A, Gunzburger MD, Hou L (2005) Optimal boundary control for the evolutionary Navier-Stokes system: the three-dimensional case. SIAM J Control Optim 43(6):2191–2232
29. Fütterer T, Klar A, Wegener R (2009) An energy conserving numerical scheme for the dynamics of hyperelastic rods. International Journal of Differential Equations 2012
30. Gavrikov A, Kostin G (2021) Optimal control of longitudinal motion of an elastic rod using boundary forces. J Comput Syst Sci Int 60(5):740–755
31. Gazzola M, Dudte L, McCormick A, Mahadevan L (2018) Forward and inverse problems in the mechanics of soft filaments. R Soc Open Sci 5(6):171628
32. Geng W, Cosman P, Berry CC, Feng Z, Schafer WR (2004) Automatic tracking, feature extraction and classification of C. elegans phenotypes. IEEE Trans Biomed Eng 51(10):1811–1820
33. Glowinski R, Pironneau O (1975) On the numerical computation of the minimum-drag profile in laminar flow. J Fluid Mech 72(2):385–389
34. Goodno BJ, Gere JM (2020) Mechanics of materials. Cengage Learning
35. Gray J, Lissmann HW (1964) The locomotion of nematodes. J Exp Biol 41(1):135–154
36. Gunzburger M, Hou L, Manservisim S, Yan Y (1998) Computations of optimal controls for incompressible flows. Int J Comput Fluid Dynam 11(1–2):181–191
37. Gunzburger M, Hou L, Svobodny TP (1991) Analysis and finite element approximation of optimal control problems for the stationary Navier-Stokes equations with Dirichlet controls. ESAIM: Math Model Numer Anal 25(6):711–748
38. Gunzburger MD (2002) Perspectives in flow control and optimization. SIAM
39. Gunzburger MD (2012) Flow control, vol 68. Springer Science & Business Media, Germany
40. Gunzburger MD, Manservisi S (2000) Analysis and approximation of the velocity tracking problem for Navier-Stokes flows with distributed control. SIAM J Numer Anal 37(5):1481–1512
41. Gunzburger MD, Manservisi S (2000) The velocity tracking problem for Navier-Stokes flows with boundary control. SIAM J Control Optim 39(2):594–634
42. Hecht F (2012) New development in freefem++. J Numer Math 20(3–4):251–266
43. Hou L, Ravindran S, Yan Y (1997) Numerical solution of optimal distributed control problems for incompressible flows. Int J Comput Fluid Dynam 8(2):99–114
44. Hou L, Yan Y (1997) Dynamics and approximations of a velocity tracking problem for the Navier-Stokes flows with piecewise distributed controls. SIAM J Control Optim 35(6):1847–1885
45. Ibrahimbegovic A, Knopf-Lenoir C, Kučerová A, Villon P (2004) Optimal design and optimal control of structures undergoing finite rotations and elastic deformations. Int J Numer Meth Eng 61(14):2428–2460
46. Jeon S, Choi J, Jeon WP, Choi H, Park J (2004) Active control of flow over a sphere for drag reduction at a subcritical reynolds number. J Fluid Mech 517:113
47. Johari S, Nock V, Alkaisi MM, Wang W (2013) On-chip analysis of C. elegans muscular forces and locomotion patterns in microstructured environments. Lab Chip 13(9):1699–1707
48. Karbowski J, Schindelman G, Cronin CJ, Seah A, Sternberg PW (2008) Systems level circuit model of C. elegans undulatory locomotion: mathematical modeling and molecular genetics. J Comput Neurosci 24(3):253–276
49. Keller JB, Rubinow SI (1976) Slender-body theory for slow viscous flow. J Fluid Mech 75(4):705–714
50. Kim J (2011) Physics and control of wall turbulence for drag reduction. Philos Trans R Soc A: Math, Phys Eng Sci 369(1940):1396–1411
51. Kostin G (2020) Verified solution to optimal control problems of elastic rod motion based on the ritz method. Acta Cybernet 24(3):393–408
52. Kwon N, Hwang AB, You YJ, Lee V, Ho SJ, Je J (2015) Dissection of C. elegans behavioral genetics in 3D environments. Sci Rep 5(1):1–9
53. Lang H, Leyendecker S, Linn J (2013) Numerical experiments for viscoelastic Cosserat rods with Kelvin-Voigt damping. Citeseer
54. Lebois F, Sauvage P, Py C, Cardoso O, Ladoux B, Hersen P, Di Meglio JM (2012) Locomotion control of Caenorhabditis elegans through confinement. Biophys J 102(12):2791–2798
55. Lions JL (1988) Exact controllability, stabilization and perturbations for distributed systems. SIAM Rev 30(1):1–68
56. Love AEH (2013) A treatise on the mathematical theory of elasticity. Cambridge University Press, Cambridge
57. Manservisi S, Menghini F (2016) Numerical simulations of optimal control problems for the reynolds averaged Navier-Stokes system closed with a two-equation turbulence model. Comput Fluids 125:130–143
58. Manservisi S, Menghini F (2016) Optimal control problems for the Navier-Stokes system coupled with the k-ω turbulence model. Comput Math Appl 71(11):2389–2406
59. McNally J, Fernandez E, Robertson G, Kumar R, Taira K, Alvi F, Yamaguchi Y, Murayama K (2015) Drag reduction on a flat-back ground vehicle with active flow control. J Wind Eng Ind Aerodyn 145:292–303
60. Meric R, Altan Ş (1989) Optimal control of memory dependent nonlocal elastic solids. Int J Eng Sci 27(4):455–461
61. Mohammadi B, Pironneau O (2010) Applied shape optimization for fluids. Oxford University Press, Oxford
62. Moubachir M, Zolesio JP (2006) Moving shape analysis and control: applications to fluid–structure interactions. CRC Press, Florida
63. Mujika A, Leškovskỳ P, Álvarez R, Otaduy MA, Epelde G (2017) Modeling behavioral experiment interaction and environmental stimuli for a synthetic C. elegans. Front Neuroinform 11:71
64. Nagarajaiah S, Narasimhan S (2007) Optimal control of structures. In: Optimization of Structural and Mechanical Systems, pp 221–244. World Scientific
65. Pironneau O (1974) On optimum design in fluid mechanics. J Fluid Mech 64(1):97–110
66. Pošta M, Roubíček T (2007) Optimal control of Navier-Stokes equations by Oseen approximation. Comput Math Appl 53(3–4):569–581
67. Rall LB (2014) Nonlinear functional analysis and applications: proceedings of an advanced seminar conducted by the Mathematics Research Center, the University of Wisconsin, Madison, October 12-14, 1970. Elsevier
68. Ranner T (2020) A stable finite element method for low inertia undulatory locomotion in three dimensions. Appl Numer Math 156:422–445
69. Ronzhina M, Manita L (2021) Singularity of optimal control for a Timoshenko beam. In: Journal of Physics: Conference Series, vol. 1740, p 012068. IOP Publishing

70. Sadek I, Sloss J, Adali S, Bruch J Jr (1997) Optimal boundary control of the longitudinal vibrations of a rod using a maximum principle. J Vib Control 3(2):235–254
71. Sadek I, Sloss J, Bruch J, Adali S (1986) Optimal control of a Timoshenko beam by distributed forces. J Optim Theory Appl 50(3):451–461
72. Salfelder F, Yuval O, Ilett TP, Hogg DC, Ranner T, Cohen N (2021) Markerless 3D spatio-temporal reconstruction of microscopic swimmers from video. In: visual observation and analysis of vertebrate and insect behavior 2020. Leeds
73. Seidman TI, Antman SS (2001) Optimal control of a nonlinearly viscoelastic rod. In: Control of Nonlinear Distributed Parameter Systems, pp 294–305. CRC Press
74. Sonneville V, Cardona A, Brüls O (2014) Geometrically exact beam finite element formulated on the special euclidean group se (3). Comput Methods Appl Mech Eng 268:451–474
75. Spillmann J, Teschner M (2007) CoRdE: Cosserat rod elements for the dynamic simulation of one-dimensional elastic objects. In: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp 63–72
76. Strange K (2006) An overview of C. elegans biology. C. elegans pp 1–11
77. Tian P (1997) Generalized optimal control of elastic and inelastic structures subjected to earthquake excitation. University of Missouri-Rolla, United States
78. Tröltzsch F (2010) Optimal control of partial differential equations: theory, methods, and applications, vol. 112. American Mathematical Soc
79. Van Khang N, Phuc VD, Van Huong NT et al (2018) Optimal control of transverse vibration of Euler-Bernoulli beam with multiple dynamic vibration absorbers using Taguchi's method. Vietnam J Mech 40(3):265–283
80. Wang Y (2022) A monolithic one-velocity-field optimal control formulation for fluid-structure interaction problems with large solid deformation. J Fluids Struct 111:103577
81. Wang Y, Jimack PK, Walkley MA, Yang D, Thompson HM (2021) An optimal control method for time-dependent fluid-structure interaction problems. Struct Multidiscip Optim 64(4):1939–1962
82. Wick T, Wollner W (2020) Optimization with nonstationary, nonlinear monolithic fluid-structure interaction. Int J Numer Methods Eng
83. Zelikin MI, Manita LA (2006) Optimal control for a Timoshenko beam. Comptes Rendus Mécanique 334(5):292–297
84. Zhong WX (2006) Duality system in applied mechanics and optimal control, vol 5. Springer Science & Business Media, Germany