2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Modeling the dynamics of soft robots by discs and threads

Joshua A. Schultz[1], Haley Sanders[2], Phuc Duc Hong Bui[1], Brett Layer[2] and Marc Killpack[2]

***Abstract*— In this paper, we propose a new tractable ordinary differential equation formulation for dynamic simulation of fabric-reinforced inflatable soft robots. The method performs a lumped-parameter discretization of the continuum robot into discrete discs (inertia), spring elements, and threads (representing the inextensible fabric reinforcement). Using the repetition in the structure of the Lagrangian formulation of the dynamic equations of motion, a method is developed that outputs machine-readable analytical expressions for the equations of motion. The method does not require symbolic computation of derivatives. The recursive nature allows us to scale the model to an arbitrary number N discs, and can represent buckling, twisting, and pleating that is commonly seen in very soft robots. The expressions generated were validated against manually-derived equations of motion for the two-disc case using both Lagrangian and Newton-Euler means. A simulation environment which parses and evaluates the analytical expressions generated at run-time was used to numerically integrate and predict the response of a four-disc example robot. Trajectories observed varied smoothly and plausibly predicted the behavior envisioned in robots like these.**

I. INTRODUCTION

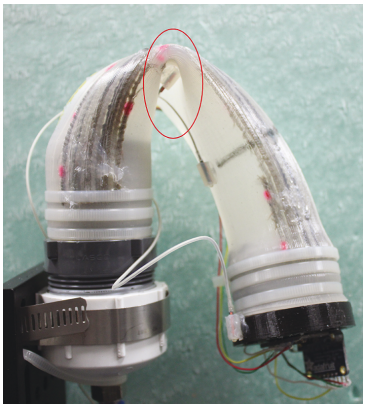
Soft Robots offer an exciting new paradigm with which to approach real-world tasks; instead of trying to strictly avoid contact with the environment save with the end-effector, they can intentionally bump, scrape, and push against the world with any part of their structure. Human operators have serendipitously discovered ways that this exciting modality can be used to perform tasks, such as crawling and grasping [1]. However, to fulfil their true promise, soft robots need to be able to conduct these tasks autonomously, without relying on human intelligence and intuition [2]. To make decisions in real time about how to take action, the robot needs to have some representation of its configuration.

Unlike traditional manipulators, a soft robot is a continuum with infinite degrees of freedom, and the choice of variables to represent its current configuration is not obvious. For real-time decision-making, and forward simulation and estimation using ordinary differential equations, some discretization of the continuum is necessary. There are a myriad of potential choices for how to discretize the continuum and assign variables, since they are fictitious and do not correspond to the motion of any specific physical part or feature. Algorithms could even jump from one discretization to another during operation.

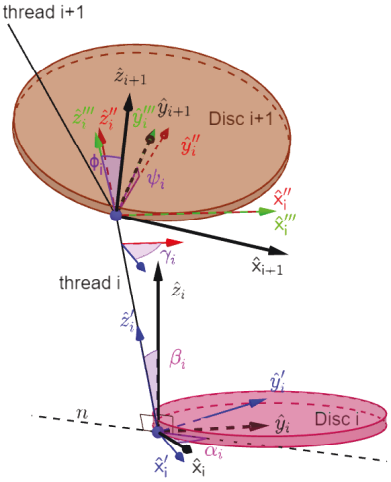
*This work is supported by the NSF grant No. 1935312 EFRI C3 SoRo: Between a Soft Robot and a Hard Place: Estimation and Control Algorithms that Exploit Soft Robots' Unique Abilities

[1]Department of Mechanical Engineering, the University of Tulsa, Tulsa, OK, 74104 joshua-schultz@utulsa.edu, phuc-bui@utulsa.edu

[2] Department of Mechanical Engineering, Brigham Young University, Provo, UT, hps22@byu.edu, marc_killpack@byu.edu



(a) wrinkling behavior in "Squishy" [3]



(b) coordinates and frame assignments

Fig. 1: The disc-thread parameterization can capture wrinkling behavior (a) using the coordinates shown in (b). Reproduced from [4] with permission.

Current literature on dynamic modeling for compliant continuum robots has its roots in what Marchese and Rus term "hard continuum manipulators" (HCMs) [5]. Methods for constructing system dynamic models based on Lagrangian methods [6] Hamilton's Principle [7] and the Principle of Virtual Power [8] provide a fundamental mathematical technique. The decision of how to model a continuum robot is usually a tradeoff between simplicity of the representation and fidelity. At one end of the spectrum, the piece-wise constant curvature assumption [9], [5], [10] simplifies the analysis and amount of computation needed, but loses accuracy under gravity [11], not to mention interaction with a comparatively stiff environment. At the other end, high-fidelity

2022 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9681-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRA46639.2022.9812286

models are predominantly based on Kirchoff/Cosserat Rod theory which can predict higher order behaviors, such as buckling [12]. Renda, et al. [13], use methods of geometric mechanics and nonlinear material deformation models for a cable-driven hydrostatic tentacle, extending this key method for HCMs to a truly soft robot. Boyer, et al. [14] applied related methods to the free-free case. Researchers have explored using Finite Element Methods [15] for computing kinematic behavior, but the computation of large numbers of nodes using iterative methods still poses computational difficulties in real-time beyond quasistatic motions.

Soft Robots are made of different materials than HCMs and function very differently. Both the piecewise constant curvature models and Kirchoff/Cosserat Rod models enforce continuity and differentiability at the boundaries of each discretized segment. Robots composed of fabric reinforced inflatable chambers violate these assumptions regularly in practice (see Fig. 1a for an example), suggesting a different approach is needed. In this paper, we discretize the physical robot at the outset, rather than discretizing an elastic continuum model of the robot. The model consists of a sequence of discs and inelastic threads. As the threads exhibit no resistance to rotation at the discs, the model does not inherently enforce any assumptions about continuity of slope or curvature at the discretization boundaries. For this reason, we believe that this model can capture kinking, pleating and folding, kinematic conditions that are likely to occur as inflatable elastomeric chambers perform complex interactions with the environment. This is similar to [16], [17], [18], except our model kinematics can capture kinking, pleating, and folding at any location along the inflatable chamber without knowledge of additional external forces or constraints in the model. In addition, gravity and external forces are naturally accounted for as part of the formulation and need not be sources of additional error.

II. The Disc-Thread Parametrization

As with other dynamic models of continuum soft robots, we will discretize the mass and stiffness of the continuum into discrete elements. Our discretization differs from prior models in that it does not consider the soft robot to be a continuum elastica with a partial differential equation (PDE) model broken into pieces, but rather discretizes the mass of the chamber walls and the deformable portion of the chamber as lumped-parameter elements. The inextensibility of the fabric reinforcement is treated separately as a holonomic constraint. The nonlinear equations of motion are then developed directly from the lumped parameter representation, without any PDEs involved. To reiterate, the benefit of this approach is that we expect to improve tractability and computational efficiency both for simulation and model-based estimation and control when performing complex interaction tasks.

The mass of each of the N segments of the chamber is represented by a disc (which need not be strictly circular) with corresponding mass and inertia tensor properties. The robot does not need to be initially straight, but should have a well-defined axial direction at each disc. As the fabric reinforcement is inextensible, we represent this by an inextensible thread connecting a pair of distinct points on each pair of discs. Thus the discs can rotate and translate in any manner, provided a constant distance between these two points is maintained (i.e. the thread is always taut). We use a single thread per disc pair as it allows the most kinematic freedom; additional threads could be adjoined as constraints or represented by an arbitrarily stiff spring. A pair of adjacent discs with the thread between them is illustrated in Fig. 1b. If the robot chamber has a continuous strip of fabric from end to end, as in the "Squishy" robot [3], the threads to the disc proximal and distal from a given disc $i+1$ in the chain will emanate from the same point on the disc as shown. The stiffness of the chamber walls is represented as discrete springs, with each spring connecting some point on disc i to disc $i+1$ $\forall i$. The advantage of this over previous models is that the motion of the discs relative to one another can represent behaviors seen in the physical robot such as wrinkling and buckling. For instance, thread $i+1$ could fold back nearly over thread i without incident in the disc-thread model, whereas with an elastica-based model such high curvatures would lead to extremely high stresses.

The development shown here retains the spirit of organization of the Denavit-Hartenberg parameterization, but is more parsimonious with regard to the number of coordinates, as certain fictitious revolute joints can be omitted. We will develop a set of generalized coordinates for the collections of discs and threads. As shown in Fig. 1b, a frame is assigned to each disc, with its origin placed where the disc is joined by the thread. If the first, or "root" disc is considered fixed, this, plus the holonomic constraint of the thread, reduces the number of degrees of freedom from $6N$ to $5(N-1)$.

The generalized coordinates are defined with regard to three intermediate frames between frame i and $i+1$, each successive one denoted with an additional prime. In any given configuration, like the one shown in Fig. 1b, the inextensible thread in the model points in some direction relative to frame i. This admits a convenient frame, one with its coordinate axis $\hat{\mathbf{z}}_i'$ directed along the thread. The $\hat{\mathbf{z}}_i'$ axis is parameterized by two numerical values (angles) $\alpha_i, \beta_i \in [0, \pi]$. Imagine pulling a thread away from the $\hat{\mathbf{z}}_i$ axis in an arbitrary direction. The axis of rotation will lie in the plane of the disc; α_i is the angle between this axis of rotation and the $\hat{\mathbf{x}}_i$ axis, which lies tangent to the disc. The angle between $\hat{\mathbf{z}}_i'$ and $\hat{\mathbf{z}}_i$ is denoted β_i. This rotation is represented by the rotation matrix $R(\alpha_i, \beta_i)$. The subsequent frame will involve translation along the inextensible thread (of a fixed distance ℓ_i determined by the discretization, along with a rotation about the inextensible thread by an amount γ_i. γ_i is defined so that the $\hat{\mathbf{x}}_i'', \hat{\mathbf{y}}_i'', \hat{\mathbf{z}}_i''$ (red) frame is rotated from $\hat{\mathbf{x}}_i', \hat{\mathbf{y}}_i', \hat{\mathbf{z}}_i'$ until the $\hat{\mathbf{x}}_i''$ axis is perpendicular to both $\hat{\mathbf{z}}_i'$ and $\hat{\mathbf{y}}_{i+1}$, ($\hat{\mathbf{x}}_i'' = \hat{\mathbf{y}}_{i+1} \times \hat{\mathbf{z}}_i'$). This allows a rotation about the $\hat{\mathbf{x}}_i''$ axis into the $\hat{\mathbf{x}}_i''', \hat{\mathbf{y}}_i''', \hat{\mathbf{z}}_i'''$ frame (green), for which $\hat{\mathbf{x}}_i'' \equiv \hat{\mathbf{x}}_i'''$. The coordinate ψ_i is the angle measured between $\hat{\mathbf{y}}_i''$ and $\hat{\mathbf{y}}_i''' \equiv \hat{\mathbf{y}}_{i+1}$. By rotating about $\hat{\mathbf{y}}_i'''$ (coincident with the center line of disc $i+1$) by an angle ϕ_i, $\hat{\mathbf{x}}_i'''$ and $\hat{\mathbf{z}}_i'''$ are rotated into $\hat{\mathbf{x}}_{i+1}$ and $\hat{\mathbf{z}}_{i+1}$, respectively. The coordinate

directions after this sequence of rotations is now aligned with the disc $i+1$. Therefore,

$$ {}^{i}_{i+1}T = T(\alpha_i, \beta_i)T(\gamma_i)T(\psi_i)T(\phi_i), \quad (1) $$

consisting of 5 variable rotations and a fixed translation, where each homogeneous transformation matrix $T(\cdot)$ is defined as in Craig [19] and following the description above (only $T(\gamma_i)$ involves a translation).

This set of generalized coordinates $\alpha, \beta, \gamma, \psi, \phi$ is a minimal set of coordinates for the two disc system (two rigid bodies, one constraint). So in free space, collecting the values for each disc, we obtain a set of generalized coordinates:

$$ [\alpha_1, \beta_1, \gamma_1, \psi_1, \phi_1 \cdots \alpha_{N-1}, \beta_{N-1}, \gamma_{N-1}, \psi_{N-1}, \phi_{N-1}] = \mathbf{q}^\top \quad (2) $$

III. Equations of Motion

In this section we present a condensed overview of the process of developing machine-readable analytical expressions for the equations of motion. The expressions use the set of generalized coordinates of Section II. The formulation is developed by writing the Euler-Lagrange equations for the sequence of discs and threads with the constraints described in Section II. It is guided by two observations: first, that since the threads are inextensible, all motion is generated by rotating each thread about the disc proximal to it and the rotation of the discs about the thread proximal to it. Thus, the center of mass velocities ${}^{i}\mathbf{v}_{iG}$ are simply an algebraic function of the angular velocity of each disc ${}^{i}\boldsymbol{\omega}_i$. The second observation is that the sequence of discs forms an open kinematic chain, and each ${}^{i}\boldsymbol{\omega}_i$ is a function of the angular velocity of every disc proximal to it, but not the discs distal to it. (The subscript denotes the disc, and the leading superscript denotes the frame in which the components are represented).

A. Disc kinematics

We can derive the kinematic expressions as follows: Proceeding as if disc i is fixed, using the frame assignments as described in Section II we write the angular velocity of disc $i+1$ *relative* to the frame of disc i in terms of the generalized coordinates associated with the thread joining them:

$$ {}^{i}\boldsymbol{\omega}_{i+1,rel} = E(\alpha_i, \beta_i, \gamma_i, \psi_i) \begin{pmatrix} \dot{\alpha}_i \\ \dot{\beta}_i \\ \dot{\gamma}_i \\ \dot{\psi}_i \\ \dot{\phi}_i \end{pmatrix}, \quad (3) $$

The columns of $E(\alpha_i, \beta_i, \gamma_i, \psi_i)$ arise from products of the rotation matrices between the intermediate frames described in Section II with the columns of those same matrices corresponding to the axes of rotation of each q_j. When $i = 2$, (3) calculates the true angular velocity. For i beyond disc 2, We combine the ${}^{i}\boldsymbol{\omega}_{i+1,rel}$ with the values for discs 1 to $i-1$ to form the angular velocity relative to an inertial frame:

$$ {}^{i}\boldsymbol{\omega}_i = {}^{i}_{i-1}R \left(\sum_{k=2}^{i-1} {}^{i-1}_{k-1}R \; {}^{k-1}\boldsymbol{\omega}_{k,rel} + {}^{i-1}\boldsymbol{\omega}_{i,rel} \right), \quad (4) $$

where ${}^{i}_{i-1}R$ is the rotation matrix between the frame of disc $i-1$ and disc i. The ${}^{i}_{i-1}R$ are functions of $\mathbf{q}$.

B. Disc kinetics

Since (4) gives us the angular velocity of each disc relative to an inertial reference frame, we can use it to write the Lagrangian $\mathcal{L}$ for the entire set of discs and threads:

$$ \mathcal{L} = \sum_{i=2}^{N} \left(\frac{m_i {}^{i}\mathbf{v}_{iG}^\top \, {}^{i}\mathbf{v}_{iG}}{2} + \frac{1}{2} \, {}^{i}\boldsymbol{\omega}_i^\top I_i^G \, {}^{i}\boldsymbol{\omega}_i \right) - \sum_{i=0}^{N-1} \sum_{j=1}^{M_i} \frac{\mathbf{f}_{ij}^{O\,\top} \mathbf{f}_{ij}^O}{2k_{ij}} - \sum_{i=2}^{N} m_i g ({}^{1}_{i}T \rho \hat{\mathbf{y}}) \cdot \hat{\mathbf{j}}, \quad (5) $$

with m_i and I_i^G ^{i} being the mass and inertia tensor about the center of mass of disc i, $\mathbf{f}_{ij}^O$ being the force in the jth spring joining disc i to disc $i-1$, and k_{ij} the spring constant, g the acceleration due to gravity, $\hat{\mathbf{j}}$ the direction against gravity, and ρ_i the radius of disc i. In this expression, ${}^{i}\boldsymbol{\omega}_i$, and as a result ${}^{i}\mathbf{v}_{iG}$, will be functions of the generalized coordinates $\mathbf{q}$ and their time derivatives $\dot{\mathbf{q}}$. In addition, the $\mathbf{f}_{ij}^O$ and ${}^{1}_{i}T$ are functions of $\mathbf{q}$. From this one can see that the equations of motion will follow directly from the partial derivatives $\frac{\partial^i \boldsymbol{\omega}_i}{\partial \dot{q}_j}$ (as well as $\frac{\partial^i \boldsymbol{\omega}_i}{\partial q_j}$, and partial derivatives within the spring and gravity terms). Using the first and second observations above, it can be shown that the partial derivatives $\frac{\partial^i \boldsymbol{\omega}_i}{\partial \dot{q}_j}$ follow the sequence shown in Table I. A similar process was used to derive tables for $\frac{\partial^i \boldsymbol{\omega}_i}{\partial \dot{q}_j}$, and the potential energy partial derviatives, but they are omitted here for brevity.

Due to the structure of Table I, it is possible to programmatically generate analytical expressions for each element of the table for any number of discs (up to available memory limits). Using a symbolic mathematics package, the expressions in the elements of the table can be generated iteratively by copying the expressions for the previous disc, using the string replace operator, and matrix multiplication, without any symbolic differentiation required. Since this is nothing more than operations on alphanumeric strings, the algorithm will successfully complete. When needed at runtime, the numerical quantities required are provided by a function evaluation on the expressions generated.

To generate the equations of motion, one typically computes the time derivative $\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathcal{L}}{\partial \dot{q}_j}$. Like with the construction of ${}^{i}\boldsymbol{\omega}_i$, it is possible to exploit the structure of the disc-thread parameterization so that this derivative need not be computed on these expressions explicitly. It can be shown that the j,k element of the mass matrix $\mathbf{M}$, M_{jk}, is given by

$$ M_{jk} = \sum_{i=1}^{N} \left(m_i \frac{\partial^i \mathbf{v}_{iG}^\top}{\partial \dot{q}_k} \frac{\partial^i \mathbf{v}_{iG}}{\partial \dot{q}_j} + \frac{\partial^i \boldsymbol{\omega}_i}{\partial \dot{q}_k}^\top \, {}^{i}I_G \frac{\partial^i \boldsymbol{\omega}_i}{\partial \dot{q}_j} \right). \quad (6) $$

TABLE I: Partial derivative of the angular velocities of each disc

$\dot{q}_j$	$\frac{\partial^2 \boldsymbol{\omega}_2}{\partial \dot{q}_j}$	$\frac{\partial^3 \boldsymbol{\omega}_3}{\partial \dot{q}_j}$	$\frac{\partial^4 \boldsymbol{\omega}_4}{\partial \dot{q}_j}$	$\cdots$
$\dot{\alpha}_1$	${}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\alpha}_1}$	${}^3_2R\ {}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\alpha}_1}$	${}^4_3R\ {}^3_2R\ {}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\alpha}_1}$	$\cdots$
$\dot{\beta}_1$	${}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\beta}_1}$	${}^3_2R\ {}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\beta}_1}$	${}^4_3R\ {}^3_2R\ {}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\beta}_1}$	$\cdots$
$\dot{\gamma}_1$	${}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\gamma}_1}$	${}^3_2R\ {}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\gamma}_1}$	${}^4_3R\ {}^3_2R\ {}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\gamma}_1}$	$\cdots$
$\dot{\psi}_1$	${}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\psi}_1}$	${}^3_2R\ {}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\psi}_1}$	${}^4_3R\ {}^3_2R\ {}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\psi}_1}$	$\cdots$
$\dot{\phi}_1$	${}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\phi}_1}$	${}^3_2R\ {}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\phi}_1}$	${}^4_3R\ {}^3_2R\ {}^2_1R \frac{\partial^1 \boldsymbol{\omega}_{2,rel}}{\partial \dot{\phi}_1}$	$\cdots$
$\dot{\alpha}_2$	0	${}^3_2R \frac{\partial^2 \boldsymbol{\omega}_{3,rel}}{\partial \dot{\alpha}_2}$	${}^4_3R\ {}^3_2R \frac{\partial^2 \boldsymbol{\omega}_{3,rel}}{\partial \dot{\alpha}_2}$	$\cdots$
$\dot{\beta}_2$		${}^3_2R \frac{\partial^2 \boldsymbol{\omega}_{3,rel}}{\partial \dot{\beta}_2}$	${}^4_3R\ {}^3_2R \frac{\partial^2 \boldsymbol{\omega}_{3,rel}}{\partial \dot{\beta}_2}$	$\cdots$
$\dot{\gamma}_2$		${}^3_2R \frac{\partial^2 \boldsymbol{\omega}_{3,rel}}{\partial \dot{\gamma}_2}$	${}^4_3R\ {}^3_2R \frac{\partial^2 \boldsymbol{\omega}_{3,rel}}{\partial \dot{\gamma}_2}$	$\cdots$
$\dot{\psi}_2$		${}^3_2R \frac{\partial^2 \boldsymbol{\omega}_{3,rel}}{\partial \dot{\psi}_2}$	${}^4_3R\ {}^3_2R \frac{\partial^2 \boldsymbol{\omega}_{3,rel}}{\partial \dot{\psi}_2}$	$\cdots$
$\dot{\phi}_2$		${}^3_2R \frac{\partial^2 \boldsymbol{\omega}_{3,rel}}{\partial \dot{\phi}_2}$	${}^4_3R\ {}^3_2R \frac{\partial^2 \boldsymbol{\omega}_{3,rel}}{\partial \dot{\phi}_2}$	$\cdots$
$\dot{\alpha}_2$	0	0	${}^4_3R\ {}^3_2R \frac{\partial^2 \boldsymbol{\omega}_{3,rel}}{\partial \dot{\alpha}_2}$	$\cdots$
$\dot{\beta}_3$			${}^4_3R \frac{\partial^3 \boldsymbol{\omega}_{4,rel}}{\partial \dot{\beta}_3}$	$\cdots$
$\dot{\gamma}_3$			${}^4_3R \frac{\partial^3 \boldsymbol{\omega}_{4,rel}}{\partial \dot{\gamma}_3}$	$\cdots$
$\dot{\psi}_3$			${}^4_3R \frac{\partial^3 \boldsymbol{\omega}_{4,rel}}{\partial \dot{\psi}_3}$	$\cdots$
$\dot{\phi}_3$			${}^4_3R \frac{\partial^3 \boldsymbol{\omega}_{4,rel}}{\partial \dot{\phi}_3}$	$\cdots$
$\vdots$	0	0	0	$\ddots$

Each term within (6) is just a function evaluation on the expressions for the angular velocity ${}^i\boldsymbol{\omega}_i$ and the elements of Table I. The structure of the equations of motion can be simplified further by introducing the concept of the *canonical momenta vector* $\mathbf{p} = \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}}$ [20]. We can then write the equations of motion in the following form:

$$\begin{pmatrix} \dot{\mathbf{q}} \\ \dot{\mathbf{p}} \end{pmatrix} = \begin{pmatrix} 0 & \mathbf{M}^{-1} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{q} \\ \mathbf{p} \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{\partial \mathcal{L}}{\partial \mathbf{q}} + \mathbf{Q} - \mathbf{A}_{\mathbf{q}}\boldsymbol{\lambda} \end{pmatrix}, \tag{7}$$

where $\frac{\partial \mathcal{L}}{\partial \mathbf{q}}$ is found using a process similar to the one used to find Table I, $\mathbf{Q}$ represents the generalized forces, which is derived from any physical forces which have an analytical model (e.g. asymmetrical weaves of fabric, other anisotropic behavior), and $\mathbf{A}_{\mathbf{q}}\boldsymbol{\lambda}$ is used to adjoin environmental contact constraints with Lagrange multipliers.

C. Effect of inflation

When inflated with pressurized air, on the timescale of robot motion we assume the pressure will be uniform within the chamber. This means that the effect of chamber pressure will integrate to zero over each disc, except for the final disc N in the chain. By defining an area element $\mathrm{d}\mathbf{A}$ as in Fig 2 and integrating over disc N, we determined analytical expressions for each of the generalized forces $Q_j \in \mathbf{Q}$ that appears in (7) due to the inflation pressure. The mathematical development has been abbreviated here. Inflation is only one force that can be represented through $\mathbf{Q}$. Given suitable models for fabric bunching, pleats, elastomer wall buckling and other configuration-dependent forces, these could be introduced by adding corresponding terms to $\mathbf{Q}$.

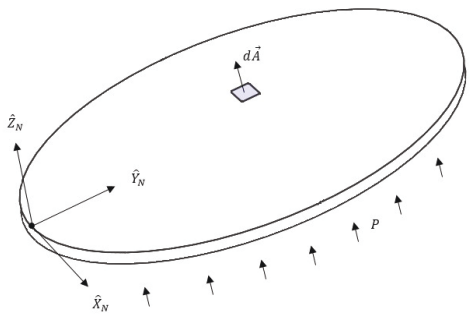


Fig. 2: Disc N under chamber pressure

IV. Model Implementation and Validation

Despite the number of potential software libraries that can perform symbolic computations related to Lagrange's equations, the complexity of these equations for our proposed model quickly exceeds the capabilities of current libraries as the number of discs increases. Therefore, we developed a method to construct the equations of motion that does not rely on symbolic differentiation, but rather replacement operators on strings according to the recursive structure exhibited in Table I. Beginning with (3), our method constructs the expression for $\boldsymbol{\omega}$ and its partial derivatives for disc 2, and proceeding iteratively, builds all expressions in Table I. This is saved to an ASCII text file which can be parsed and automatically converted to Python code. The resulting equations can then be evaluated for specific values of $\mathbf{q}$. These equations include methods to calculate the Mass matrix, Gravity vector, spring forces (which are only dependent on the relative position of two discs), and required velocity vectors for each disc. We then use the resulting equations to perform numerical integration using a fourth-order Runge-Kutta method. In the results presented in this paper, the robot was assumed to be lightweight and the velocities relatively small. Because of this the contribution of centripetal and Coriolis accelerations were minimal. To simplify the implementation, the $\frac{\partial \mathcal{L}}{\partial \mathbf{q}}$ term in (7) included only the effects of elasticity and gravity.

As the analytical expressions involved reach several pages with even only a few discs, validation of the equations of motion and the correctness of the implementation is certainly a concern. Aslanov and Yudintsev in their analysis of a space tug [21] approached a somewhat similar problem, albeit consisting of a single rigid body connected to a thread, under a somewhat different set of assumptions. They too used a Lagrangian formulation to develop the equations of motion, and validated the approach by comparing the system trajectories to those generated by the variation of the angular momentum vector about the point occupied by space tug. Inspired by this approach, we examined the two-disc (one fixed, one moving) system. For the system with a single moving disc, it is still tractable, though cumbersome, to generate equations of motion for the five generalized coordinates using both Euler-Lagrange and Newton-Euler methods. We compared the numerical values of the mass matrix, gravity vector, and spring forces from our Python code implementation with those generated by the direct Newton-Euler and Euler-Lagrange approaches for the two disc system. We used a wide selection of arbitrary values for $\alpha_i, \beta_i, \gamma_i, \psi_i$, and ϕ_i and found agreement to within reasonable numerical accuracy among all three models.

V. Simulation Results

We simulated a fabric-reinforced soft robot discretized into four discs, with the first (root) disc stationary. This is the same discretization as Bui and Schultz [4], [22] used in their estimator demonstration which was designed based on the disc-thread parametrization. While it is possible to simulate higher numbers of discs, four discs is sufficiently complex to demonstrate the method without running into limitations from long run-times, numerical integration difficulties, and memory usage challenges, none of which have yet been optimized in the current implementation. The four-disc discretization means that there are 15 state variables ($\mathbf{q} \in \mathbb{R}^{15}$). Constants were normalized so the simulation would be unitless. To represent the elasticity of the elastomer chamber, we included three springs between each set of discs. The springs were placed at 90, 180, and 270 degrees counterclockwise from the thread around the discs, each with a spring constant of 1. For this example, the resting lengths of the springs were selected so that in the unstrained configuration, the chamber would be straight. For simplicity, each disc was given a radius of 1, and the thread lengths were all 1 as well. Discs 1–3 had a mass of 2, and disc 4 had a mass of 4 to represent additional payload. The inertia tensor for a thin hoop was used for each disc. Damping was introduced by subtracting a term $b_j \dot{q}_j$ from each of the lower 15 rows of (7) with $b_j = 100 \ \forall j$ a damping constant.

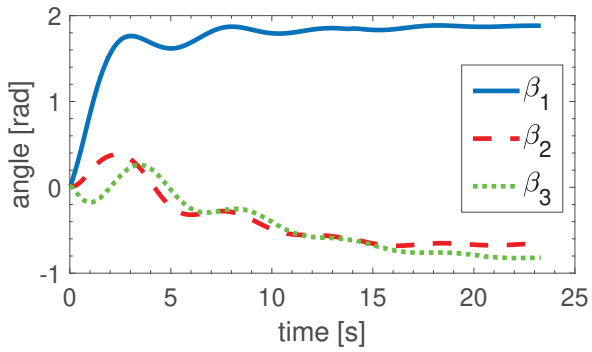
A variety of initial conditions were chosen to simulate. The simulation took about 2 seconds to run on a standard desktop computer. In some scenarios the chamber pressure was set to zero and the discs were allowed to fall under gravity. The disc trajectories returned by the simulation output appeared plausible. In certain simulations, we varied the inflation pressure of the chamber until gravity and the effects of inflation would reach some equilibrium pose. Starting the simulation at an initial condition away from this equilibrium eventually converged toward the pressure-gravity equilibrium point.

The filmstrip in Fig. 3 and animation in the accompanying video show one example. In this example, the chamber pressure was set to 12000 Pa, and the initial condition was $\beta_1 = 0.1, \beta_2 = \beta_3 = 0.01$. All other generalized coordinates were set to be initially zero. The robot initially falls under gravity due to the long moment arm of the more distal discs. Once the robot reaches a downward pose, with smaller moment arms due to gravity, the inflation is able to lift the robot until it reaches the curled pose at the right, where inflation forces and gravity are approaching equilibrium. This is qualitatively consistent with the behavior of the "Squishy" robot at The University of Tulsa and the "Kaa" robot at Brigham Young (see [23], [24]) under similar conditions.

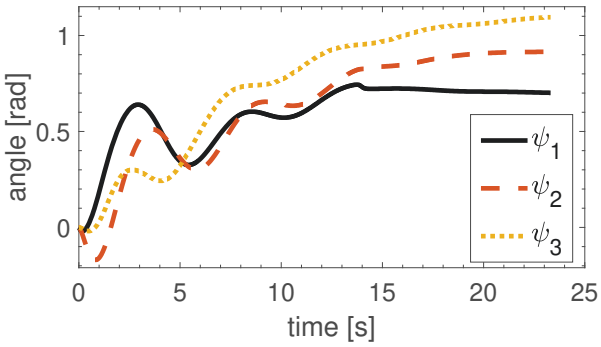
Because the robot was configured to be straight at the reference configuration (essentially unstrained), the robot shown in Fig. 3 did not have a tendency for the α_i, γ_i, and ϕ_i to change due to the motion. This would be different for robots with a curved fabric strip, such as "Squishy". As expected, the generalized coordinates initially set to zero did not change during the simulation. The time histories of the β and ψ values for each disc are shown in Fig 4. Notice that ψ_1 shows some visible oscillation and all discs initially show a second order response. In addition, the proximal discs change more rapidly than the more distal ones, as would be expected due to the larger gravitational moment arm.

Some difficulties were observed that are inherent to the disc-thread mathematical formulation. Certain configurations

Fig. 3: Frames showing the motion of the simulation output



(a) β vs. time each disc



(b) ψ vs. time each disc

Fig. 4: Simulation output showing the time evolution of generalized coordinates β and ψ for each moving disc

caused mass matrix singularities. One of these occurs when a β_i is zero. The physical explanation is that the axis of rotation corresponding to changes in α_i lines up with that corresponding to changes in γ_i. At this configuration there are an infinite combinations of the two variables that represent the same pose. There are many other examples of singularities that can occur and this is an area for further study. Several of the simulations run had a tendency to become numerically unstable after several seconds, with the trajectories deviating from the smooth ones observed earlier in the simulation. Examination of the condition number showed that the mass matrix was poorly conditioned at the time this behavior occurred. A combination of heuristics near poses determined to be singular, further analysis, and refinements should improve performance of the implementation.

VI. Conclusion

In this paper, we developed a formulation for the dynamic equations of an inflatable, fabric-reinforced soft robot and a simulation environment based on the dynamic equations. This formulation differs from prior ones in that we discretize the elastomeric materials into discrete discs and springs, with the fabric reinforcement as an inextensible thread, rather than discretizing the partial differential equations for the continuum. This lumped parameter model of the system leads directly to ordinary differential equations of motion. Because the discretization does not impose any restrictions on curvature from disc to disc, this model is capable of capturing behaviors of soft robots such as kinking, pleating, wrinkling, and folding without difficulty. In discretizations of a PDE continuum model, these high curvatures would lead to high stresses and be difficult to represent.

We presented a method for building up analytical expressions for the equations of motion of a disc-thread discretization. It produces a set of machine-readable ASCII expressions which can be evaluated for the mass matrix, stiffness, gravitation, and acceleration terms at run-time. These expressions can be computed even for high numbers of discs. Numerical simulations showed disc and thread motions that qualitatively seemed to follow soft robot behavior.

While the simulation environment successfully produces smooth, plausible output trajectories, there are plenty of opportunities for future work. The numerical properties of the simulation could be improved and motions near singularities should be further investigated. The effects of the the number of discs on simulation accuracy should be assessed. A parameter identification effort on existing soft robots would allow this method to be used in applications to predict real robots' behavior. The simulation could be extended to consider changes in disc dimensions under chamber pressure or dynamic effect. Studies using this simulation environment with large payloads or high-speed motions could possibly predict unique soft robot behaviors, such as oscillations which involve kinking from side to side at moderate pressures with heavy payloads. Finally, the examples simulated were relatively slow moving, to where centripetal and Coriolis terms could be neglected. While these terms are included in the mathematical formulation, we plan to implement and validate them in future versions of the simulation environment for more faithful simulation of high-speed motions.

REFERENCES

[1] F. Ilievski, A. D. Mazzeo, R. F. Shepherd, X. Chen, and G. M. Whitesides, "Soft robotics for chemists." *Angewandte Chemie (International ed. in English)*, vol. 50, no. 8, pp. 1890–5, feb 2011.

[2] J. Schultz, Y. Mengüç, M. Tolley, and B. Vanderborght, "What Is the Path Ahead for Soft Robotics?" *Soft Robotics*, vol. 3, no. 4, pp. 159–160, 2016.

[3] J. G. Williamson, C. Schell, M. Keller, and J. Schultz, "Extending the reach of single-chamber inflatable soft robots using magnetorheological fluids," in *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)*, 2021, pp. 119–125.

[4] P. D. H. Bui and J. A. Schultz, "States and Contact Forces Estimation for a Fabric-Reinforced Inflatable Soft Robot," in *2021 IEEE International Conference on Robotics and Automation (ICRA2021)*, Xi'An, 2021, pp. 11 820–11 826.

[5] A. D. Marchese and D. Rus, "Design , kinematics , and control of a soft spatial fluidic elastomer manipulator," *International Journal of Robotics Research*, vol. 35, no. 7, pp. 840–869, 2016.

[6] J. Spillman and M. Teschner, "C O R D E : Cosserat Rod Elements for the Dynamic Simulation of One-Dimensional Elastic Objects," in *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, vol. 1, 2007, pp. 63–72.

[7] I. A. Gravagne, C. D. Rahn, and I. D. Walker, "Planar Continuum Robots," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM*, vol. 8, no. 2, pp. 299–307, 2003.

[8] W. S. Rone and P. Ben-Tzvi, "Multi-Segment Continuum Robot Shape Estimation Using Passive Cable Displacement," in *2013 IEEE International Symposium on Robotic and Sensors Environments*, no. 1334227, Washington, DC, 2013, pp. 21–23.

[9] M. A. Csencsits and B. A. Jones, "Closed-Form Inverse Kinematics for Continuum Manipulators," *Advanced Robotics*, vol. 23, no. October 2009, pp. 2077–2091, 2015.

[10] N. Farrow and N. Correll, "A Soft Pneumatic Actuator that Can Sense Grasp and Touch," in *proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 2317–2323.

[11] D. Trivedi, C. D. Rahn, M. Kier, and I. D. Walker, "Soft robotics : Biological inspiration , state of the art , and future research," *Applied Bionics and Biomechanics*, vol. 5, no. 3, pp. 99–117, 2008.

[12] J. Till and D. C. Rucker, "Elastic Stability of Cosserat Rods and Parallel Continuum Robots," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 718–733, 2017.

[13] F. Renda, J. Dias, and L. Seneviratne, "Discrete Cosserat Approach for Multisection Soft Manipulator Dynamics," *International Journal of Robotics Research*, vol. 34, no. 6, pp. 1518–1533, 2018.

[14] F. Boyer, V. Lebastard, F. Candelier, and F. Renda, "Dynamics of Continuum and Soft Robots: A Strain Parameterization Based Approach," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 847–863, 2021.

[15] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013, pp. 3982–3987.

[16] N. S. Usevitch, Z. M. Hammond, M. Schwager, A. M. Okamura, E. W. Hawkes, and S. Follmer, "An untethered isoperimetric soft robot," *Science Robotics*, vol. 5, no. 40, p. eaaz0492, Mar. 2020, publisher: American Association for the Advancement of Science. [Online]. Available: https://www.science.org/doi/full/10.1126/scirobotics.aaz0492

[17] M. Jiang, Q. Yu, and N. Gravish, "Vacuum induced tube pinching enables reconfigurable flexure joints with controllable bend axis and stiffness," in *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)*, Apr. 2021, pp. 315–320.

[18] Y. Jiang, M. Sharifzadeh, and D. M. Aukes, "Reconfigurable Soft Flexure Hinges via Pinched Tubes," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2020, pp. 8843–8850, iSSN: 2153-0866.

[19] J. J. Craig, *Introduction to Robotics, Mechanics and Control*, 3rd ed., D. A. George, Ed. Upper Saddle River: Pearson Prentice-Hall, 2005.

[20] J. García de Jalon and E. Bayo, *Kinematic and Dynamic Simulation of Multi-Body Systems: The Real-Time Challenge*. New York: Springer-Verlag, 1994.

[21] V. S. Aslanov and V. V. Yudintsev, "Dynamics of large debris connected to space tug by a tether," *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 6, pp. 1654–1660, 2013.

[22] P. D. Bui and J. A. Schultz, "A Semilinear Parameter-Varying Observer Method for Fabric-Reinforced Soft Robots," *Frontiers in Robotics and AI*, vol. 8, no. November, pp. 1–12, 2021.

[23] P. Hyatt, D. Kraus, V. Sherrod, L. Rupert, N. Day, and M. D. Killpack, "Configuration estimation for accurate position control of large-scale soft robots," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 88–99, 2018.

[24] J. S. Terry, J. Whitaker, R. W. Beard, and M. D. Killpack, "Adaptive control of large-scale soft robot manipulators with unknown payloads," in *Dynamic Systems and Control Conference*, vol. 59162. American Society of Mechanical Engineers, 2019, p. V003T20A003.