



Article

Deep CNN-Based Static Modeling of Soft Robots Utilizing Absolute Nodal Coordinate Formulation

Haitham El-Hussieny ^{1,*} , Ibrahim A. Hameed ^{2,*} and Ayman A. Nada ¹

¹ Department of Mechatronics and Robotics Engineering, Egypt-Japan University of Science and Technology (E-JUST), Alexandria 21934, Egypt; ayman.nada@ejust.edu.eg

² Department of ICT and Natural Sciences, Norwegian University of Science and Technology, 7034 Trondheim, Norway

* Correspondence: haitham.elhussieny@ejust.edu.eg (H.E.-H.); ibib@ntnu.no (I.A.H.)

Abstract: Soft continuum robots, inspired by the adaptability and agility of natural soft-bodied organisms like octopuses and elephant trunks, present a frontier in robotics research. However, exploiting their full potential necessitates precise modeling and control for specific motion and manipulation tasks. This study introduces an innovative approach using Deep Convolutional Neural Networks (CNN) for the inverse quasi-static modeling of these robots within the Absolute Nodal Coordinate Formulation (ANCF) framework. The ANCF effectively represents the complex non-linear behavior of soft continuum robots, while the CNN-based models are optimized for computational efficiency and precision. This combination is crucial for addressing the complex inverse statics problems associated with ANCF-modeled robots. Extensive numerical experiments were conducted to assess the performance of these Deep CNN-based models, demonstrating their suitability for real-time simulation and control in statics modeling. Additionally, this study includes a detailed cross-validation experiment to identify the most effective model architecture, taking into account factors such as the number of layers, activation functions, and unit configurations. The results highlight the significant benefits of integrating Deep CNN with ANCF models, paving the way for advanced statics modeling in soft continuum robotics.

Keywords: soft robots; continuum robots; statics modeling; deep learning; CNN



Citation: El-Hussieny, H.; Hameed, I.A.; Nada, A.A. Deep CNN-Based Static Modeling of Soft Robots Utilizing Absolute Nodal Coordinate Formulation. *Biomimetics* **2023**, *8*, 611. <https://doi.org/10.3390/biomimetics8080611>

Academic Editor: Phil Husbands

Received: 12 November 2023

Revised: 26 November 2023

Accepted: 11 December 2023

Published: 14 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Recently, continuum robots have gained significant interest, as shown by the expanding research [1,2]. These robots have flexible backbones that allow for constant bending due to their elastic design, giving them notable dexterity, agility, and adaptability. Their capabilities make them ideal for tasks in tight spaces where flexibility, maneuverability, and safe human interactions are crucial, as emphasized in newer research [3–5]. Unlike traditional rigid robots, continuum robots can change their forms dynamically, which makes them adept at moving through both tight and unpredictable spaces. This versatility makes them fitting for a range of uses, from inspecting industrial pipelines [6] and aircraft engine upkeep [7] to medical procedures [3,8].

Continuum robots are a specific subset of soft robots that are composed of a continuous, flexible structure that can bend and twist in any direction. They have a wide range of applications, including medical interventions [9,10], deep-sea explorations [11], and agricultural tasks [12,13].

Continuum robots utilize various actuation techniques, such as mechanical, fluidic, and magnetic methods. Mechanical actuation methods involve technologies like concentric tubes and backbone-driven actuators. Meanwhile, fluidic actuation covers designs like pleated, corrugated, belloscope tip-driven, and endoscope systems [14]. To unlock the full capabilities of continuum robots, accurate modeling and control are essential to performing specific movement and manipulation tasks [15]. However, their naturally flexible and

malleable nature makes modeling their dynamics a complex task [16]. Unlike robots with rigid components, continuum robots present both distinct challenges and potentials in the creation of controllers [17,18]. One of the key challenges in controlling continuum robots is achieving precise tip control, which is necessary for performing delicate tasks in complex environments [19,20].

Various modeling techniques have been adopted, each varying in their base assumptions and scope of application. A thorough examination of these modeling strategies, their numerical methods, and an in-depth assessment of their practicality is presented in Armanini's research [21]. One significant approach is the lumped mass model [22], where a flexible link is visualized as segments of point masses linked by springs and dampers. In this model, every mass is affected by gravitational forces and the movements of its neighboring masses, with some potentially facing external forces or actuator-induced interactions. The piecewise constant curvature model is another approach that has gained traction [23–27], simplifying robot kinematics by segmenting them into parameters that reflect constant-curvature segments. However, this method's limitation lies in its strict adherence to constant curvature motions, occasionally neglecting crucial real-world influencers like gravity or friction.

Another model worth noting is the discrete elastic rod (DER) model introduced by Naughton et al. [28]. It is a computational approach used to simulate the behavior of thin, flexible rods in various applications, ranging from computer graphics to the study of biological fibers and soft robotics. Thus, a flexible rod is modeled using a series of lines connected at specific points and uses discrete differential geometry to capture bending and twisting. But, while advantageous in offering a balance between computational efficiency and physical accuracy, the DER model may not encapsulate the intricate designs seen in specific continuum robots. The simulation of the dynamics in soft material systems requires an intricate analysis of kinematics, mechanics, and tribology. These complexities might not be fully encompassed by the Discrete Elastic Rod (DER) model, which is generally employed for modeling slender structures such as rods, ribbons, and viscous threads [29].

Lately, finite-element-based models (FEM) have gained popularity in soft robotics [30–32]. These divide the robot's structure into one, two, or three-dimensional elements, resembling the classification in conventional robotics. Beams and rods like the Euler–Bernoulli beam, Timoshenko beam, and the Cosserat rod provide the material deformation understanding of the robot [33]. However, despite their suitability for minor deformation challenges [34], FEMs come with their set of drawbacks such as computational overheads, potential for numerical instabilities, inaccuracies emerging from mesh alterations, and failing in the modeling of large deformations.

While the previously mentioned methodologies offer valuable insights into modeling continuum robots, there is a noticeable gap in exhaustive data-driven statics modeling and control techniques. This gap hinders the full exploitation of these robots across varied domains. Addressing this issue requires embracing data-driven modeling approaches, especially capitalizing on machine learning as underscored in a recent review by Wang et al. [35]. Given the swift progress in artificial intelligence areas, including machine learning, robotics, neural networks, transformers, and generative networks, incorporating data-driven statics modeling and control methods for soft continuum robots is promising. Such advancements can lead to significant benefits, such as improved accuracy, enhanced safety, and better operational efficiency in robotic applications [36].

In this study, we utilize a deep convolutional neural network (Deep CNN) for the data-driven statics modeling and control of soft continuum robots. Once trained, the CNN model establishes a nonlinear function to handle input data, providing solutions across diverse inputs. The training and validation datasets are rooted in the Absolute Nodal Coordinate Formulation (ANCF) framework, as detailed in the research by Nada et al. [37], Shabana [38–40], and Taylor [41]. This formulation takes into account both geometric and material non-linearities within the multi-body dynamic system, thus presenting a closer-to-reality portrayal. Significantly, the beam element informed by ANCF moves beyond the

assumption of a rigid cross-section, factoring in distortional cross-sectional deformations. This results in a holistic model for robotic structures, as showcased by Wang et al. [42].

The contribution of this research is leveraging the strength of CNN-based models in computational efficiency and precision, thus addressing a critical limitation of using ANCF alone. This integration allows for finding solutions to the inverse statics problem, which was previously challenging to resolve analytically by the ANCF. By integrating Deep CNN with ANCF, our approach represents a significant advancement over existing methods, providing a more robust and efficient tool for statics modeling in soft continuum robotics, while allowing for the solving of the inverse statics for soft robots modeled via ANCF. In this setup, the model receives a target end-effector position and endeavors to calculate the necessary external moments at the robot’s tip to reach the given position. Relying solely on ANCF modeling makes addressing the inverse statics issue quite complex. However, by incorporating data-driven, model-free control methods, there is an opportunity to tap into the complete capabilities of continuum robots across various applications. To assess and oversee the effectiveness of the updated CNN model, we employ a k-fold cross-validation method to gauge the performance of various models.

This paper is organized as follows: Section 2 delves into the derivation of the Absolute Nodal Coordinate Formulation (ANCF), showing its significance in multibody system modeling concerning continuum robots. Section 3 elaborates on the CNN quasi inverse static modeling of the continuum robot within the ANCF framework, discussing data gathering and model architectures. Subsequently, Section 4 showcases the results from our data-driven statics modeling and control techniques, evaluating their efficacy and considering potential future research directions. Finally, Section 5 recapitulates the main findings of the paper, emphasizing the importance of data-driven statics modeling and control in the context of soft continuum robots.

2. Absolute Nodal Coordinate Formulation (ANCF)

In this section, we detail the steps of deriving the forward static modeling of the continuum robots based on the Absolute Nodal Coordinate Formulation (ANCF). The ANCF is a fundamental methodology in the realms of multi-body dynamics and finite element analysis, particularly adept at modeling structures undergoing significant deformations and complex geometries. This formulation is instrumental in characterizing the kinematics of intricate structures, such as flexible bodies, cables, and continuous systems, marking a departure from the conventional assumption of rigid cross-sections. This shift is crucial for accurately depicting the deformations experienced by continuum robots’ cross-sections during dynamic analyses. For instance, as depicted in Figure 1, a soft beam element with a circular cross-section connecting nodes 1 and 2 demonstrates complex deformations including spatial bending, twisting, and stretching. Leveraging this, the current research endeavors to develop an ANCF-based model for continuum robots, which is further refined through approximation using Deep Convolutional Neural Network (CNN) models. This approach ensures the incorporation of all relevant deformations, thus enhancing the fidelity and applicability of the model in dynamic scenarios.

2.1. Problem Definition

In the ANCF framework, absolute coordinates are employed to describe the deformation and motion of individual nodes within a structure, in contrast to local or relative coordinates. This use of absolute coordinates establishes a global reference frame, ensuring a more precise representation of node positions and orientations. This global reference system is referred to as the Structure Coordinate System {SCS: (XYZ)}, as introduced in [43].

For a two-noded beam element, as illustrated in Figure 1, the absolute nodal coordinates of a node k , where $k = (1, 2)$, situated in element j on body i , can be described by the following expression:

$$\bar{\mathbf{e}}^{ijk} = \begin{bmatrix} \bar{\mathbf{r}}^{ijkT} & \frac{\partial \bar{\mathbf{r}}^{ijkT}}{\partial x^i} & \frac{\partial \bar{\mathbf{r}}^{ijkT}}{\partial y^i} & \frac{\partial \bar{\mathbf{r}}^{ijkT}}{\partial z^i} \end{bmatrix}_{12 \times 1}^T \tag{1}$$

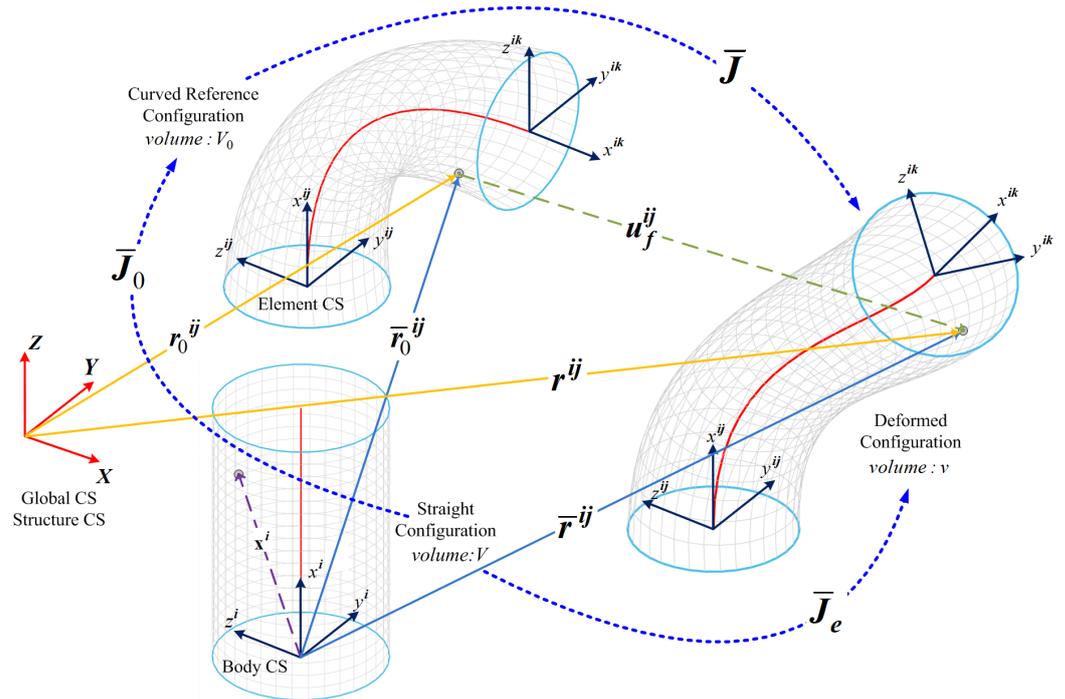


Figure 1. Diverse deformations exhibited by a 3D beam as represented by the ANCF.

The spatial position of node k within the given framework is denoted by $\bar{\mathbf{r}}^{ijk}$, while the gradients at node k with respect to the body coordinate system {BCS: $(x^i y^i z^i)$ } are given by $\partial \bar{\mathbf{r}}^{ijk} / \partial x^i$, $\partial \bar{\mathbf{r}}^{ijk} / \partial y^i$, and $\partial \bar{\mathbf{r}}^{ijk} / \partial z^i$. The BCS is aligned with the x-axis of the first element ($j = 1$) of the beam mesh, represented by (x^{i1}) . Hence, no rotation transformation is needed between the BCS and the element coordinate system {ECS: (x^{ij}, y^{ij}, z^{ij}) } at the first element of the beam mesh. The vector $\partial \bar{\mathbf{r}}^{ij} / \partial x^i$ defines the orientation of the beam’s centerline to the BCS, while the vectors $\partial \bar{\mathbf{r}}^{ij} / \partial y^i$ and $\partial \bar{\mathbf{r}}^{ij} / \partial z^i$ define the orientation of the height and width coordinates of the beam’s cross-section, respectively. These vectors may not be orthogonal unit vectors, as discussed in [40].

The transformation between the BCS: $(x^i y^i z^i)$ and the SCS: (XYZ) can be expressed by a constant transformation matrix \mathbf{A}_o , that is composed of the orthogonal unit vectors along the body reference frame, such that

$$\mathbf{A}_o = \frac{\partial \mathbf{X}}{\partial \mathbf{x}^i} \tag{2}$$

For the beam represented in Figure 1, assuming no translation is carried out, $\mathbf{A}_o \in \mathbb{R}^{3 \times 3}$ can be expressed as a pure rotation matrix

$$\mathbf{A}_o = \begin{bmatrix} X \cdot x^i & X \cdot y^i & X \cdot z^i \\ Y \cdot x^i & Y \cdot y^i & Y \cdot z^i \\ Z \cdot x^i & Z \cdot y^i & Z \cdot z^i \end{bmatrix} = \begin{bmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \tag{3}$$

Thus, the absolute nodal coordinates of the node k with respect to the SCS could be obtained with the transformation carried out, as $\mathbf{e}^{ijk} = \mathbf{T}_o \bar{\mathbf{e}}^{ijk}$, such that

$$\mathbf{e}^{ijk} = \begin{bmatrix} \mathbf{r}^{ijk} \\ \frac{\partial \mathbf{r}^{ijk}}{\partial X} \\ \frac{\partial \mathbf{r}^{ijk}}{\partial Y} \\ \frac{\partial \mathbf{r}^{ijk}}{\partial Z} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_o & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_o & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_o & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_o \end{bmatrix} \begin{bmatrix} \bar{\mathbf{r}}^{ijk} \\ \frac{\partial \bar{\mathbf{r}}^{ijk}}{\partial x^i} \\ \frac{\partial \bar{\mathbf{r}}^{ijk}}{\partial y^i} \\ \frac{\partial \bar{\mathbf{r}}^{ijk}}{\partial z^i} \end{bmatrix}$$

In this context, $\bar{\mathbf{e}}^{ijk}$ and \mathbf{e}^{ijk} denote the nodal coordinates defined within the (BCS) and the (SCS), respectively. This choice of representation ensures the maintenance of

inter-element continuity for global displacement gradients at these specific points. The nodal coordinates for an element consisting of two nodes can be succinctly expressed as a vector of

$$\bar{\mathbf{e}}^{ij} = [\bar{\mathbf{e}}^{ij1T} \quad \bar{\mathbf{e}}^{ij2T}]_{24 \times 1}^T.$$

Therefore, in the ANCF, the position of an arbitrary point on the body i , element j , $\bar{\mathbf{r}}^{ij}$ with respect to **BCS** ($x^i y^i z^i$), and \mathbf{r}^{ij} with respect to **SCS** (XYZ), can be defined as follows:

$$\bar{\mathbf{r}}^{ij}(\bar{\mathbf{u}}^{ij}, t) = \mathbf{S}^{ij}(\bar{\mathbf{u}}^{ij}) \bar{\mathbf{e}}^{ij}(t) \tag{4}$$

$$\mathbf{r}^{ij}(\bar{\mathbf{u}}^{ij}, t) = \mathbf{S}^{ij}(\bar{\mathbf{u}}^{ij}) \mathbf{T}_o^T \mathbf{e}^{ij}(t) \tag{5}$$

$$= \mathbf{A}_o \bar{\mathbf{r}}^{ij}(\bar{\mathbf{u}}^{ij}, t) \tag{6}$$

such that $\mathbf{e}^{ijk} = \mathbf{T}_o \bar{\mathbf{e}}^{ijk}$, and \mathbf{S}^{ij} is the element shape function matrix, and

$$\bar{\mathbf{u}}^{ij} = [x^{ij} \quad y^{ij} \quad z^{ij}]^T$$

is the local position of the arbitrary point with respect to the **ECS**, where, x^{ij} , y^{ij} , and z^{ij} are the coordinates along the element j . A straightforward procedure to construct the shape function is demonstrated by [42]. Equation (5) is crucial when the nodal coordinates are defined using the structural coordinates system, while Equation (6) is used in the simulation of the output results on the global (structural) coordinate system.

The initial configuration is defined such that the time is equal to zero; thus, the absolute position vector of an arbitrary point in the reference configuration can be described as

$$\bar{\mathbf{r}}^{ij}(\bar{\mathbf{u}}^{ij}, 0) = \bar{\mathbf{r}}_0^{ij} = \mathbf{S}^{ij}(\bar{\mathbf{u}}^{ij}) \bar{\mathbf{e}}_0^{ij} \tag{7}$$

where $\bar{\mathbf{e}}_0^{ij}$ is the vector of nodal coordinates in the reference configuration with respect to the body coordinate system (**BCS**).

The displacement field, $\bar{\mathbf{r}}^{ij}(\bar{\mathbf{u}}^{ij}, t)$, can be written as

$$\bar{\mathbf{r}}^{ij}(\bar{\mathbf{u}}^{ij}, t) = \bar{\mathbf{r}}_0^{ij} + \mathbf{u}_f^{ij} = \mathbf{S}^{ij}(\bar{\mathbf{u}}^{ij}) (\bar{\mathbf{e}}_0^{ij} + \bar{\mathbf{e}}_f^{ij}) \tag{8}$$

where $\bar{\mathbf{e}}_f^{ij}$ is the vector of nodal displacements with respect to the **BCS**. It can be shown that a line element $d\mathbf{x}^i$ in the straight configuration corresponds to a line element $d\bar{\mathbf{r}}_0^{ij}$ in the initial configuration and to a line element $d\bar{\mathbf{r}}^{ij}$ in the current configuration. One has the following relationships:

$$d\bar{\mathbf{r}}^{ij} = \bar{\mathbf{J}}^{ij} d\bar{\mathbf{r}}_0^{ij}, \quad d\bar{\mathbf{r}}^{ij} = \bar{\mathbf{J}}_e^{ij} d\mathbf{x}^i, \quad d\bar{\mathbf{r}}_0^{ij} = \bar{\mathbf{J}}_0^{ij} d\mathbf{x}^i$$

The gradients of the displacement vector, matrix $\bar{\mathbf{J}}^{ij}$, are defined as follows:

$$\bar{\mathbf{J}}^{ij} = \frac{\partial \bar{\mathbf{r}}^{ij}}{\partial \bar{\mathbf{r}}_0^{ij}} = \frac{\partial \bar{\mathbf{r}}^{ij}}{\partial \mathbf{x}^i} \frac{\partial \mathbf{x}^i}{\partial \bar{\mathbf{r}}_0^{ij}} = \bar{\mathbf{J}}_e^{ij} \bar{\mathbf{J}}_0^{ij-1} \tag{9}$$

If the element local coordinate system is parallel to the body coordinate system and the element is not curved, matrix $\bar{\mathbf{J}}_0^{ij}$ is an identity matrix. The relationship between the volume of the curved structure V_o in the initial configuration, to the volume of the straight configuration V is defined as

$$\begin{aligned} dV_o &= |\bar{\mathbf{J}}_0| dV \\ dv &= |\bar{\mathbf{J}}_e| dV \end{aligned}$$

Note that

$$\bar{\mathbf{J}}_0 = \frac{\partial \bar{\mathbf{r}}_0^{ij}}{\partial \mathbf{x}^i} = \frac{\partial \mathbf{S}^{ij}(\bar{\mathbf{u}}^{ij})}{\partial \mathbf{x}^i} \bar{\mathbf{e}}_0^{ij} = \frac{\partial \mathbf{S}^{ij}(\bar{\mathbf{u}}^{ij})}{\partial \mathbf{x}^i} \mathbf{T}_o^T \mathbf{e}_0^{ij}$$

where $|\bar{\mathbf{J}}_0|$ is the determinant of the matrix of position vector gradients $\bar{\mathbf{J}}_0$, which is constant. By defining \mathbf{p}^i as the unconstrained (free) vector of nodal coordinates over the flexible body i with the dimension of $DOFs \times 1$, $DOFs$ are the total number of degrees of freedom. Thus, Equation (4) can be rewritten as follows:

$$\bar{\mathbf{r}}^{ij} = \mathbf{S}^{ij} \bar{\mathbf{e}}^{ij} = \mathbf{S}^{ij} \mathbf{B}_1^{ij} \mathbf{B}_2^i \bar{\mathbf{p}}^i \tag{10}$$

where \mathbf{B}_1^{ij} is the connectivity matrix and \mathbf{B}_2^i is boundary conditions linear-transformation matrix.

2.2. Elastic Forces

The nonlinear Lagrangian strain tensor, $\epsilon_{\mathcal{L}}$, can be defined using the right Cauchy–Green deformation tensor as follows [39]:

$$\epsilon_{\mathcal{L}} = \frac{1}{2} (\mathbf{J}^T \mathbf{J} - \mathbf{I}) \tag{11}$$

The gradients of the displacement vector are defined in Equation (9), in which $\mathbf{J} = \partial \mathbf{r}^{ij} / \partial \mathbf{r}_0^{ij} = \mathbf{J}_e \mathbf{J}_0^{-1}$. It is possible to verify that, if the body experiences a rigid motion, matrix \mathbf{J} is orthonormal, and thus it is clear from Equation (10) that there is no strain. Note that

$$\mathbf{J}_e = \frac{\partial \mathbf{r}^{ij}}{\partial \mathbf{X}} = \frac{\partial \mathbf{A}_o \bar{\mathbf{r}}^{ij}}{\partial \mathbf{x}^i} \frac{\partial \mathbf{x}^i}{\partial \mathbf{X}} = \mathbf{A}_o \frac{\partial \bar{\mathbf{r}}^{ij}}{\partial \mathbf{x}^i} \mathbf{A}_o^T = \mathbf{A}_o \bar{\mathbf{J}}_e \mathbf{A}_o^T$$

Similarly, $\mathbf{J}_0 = \mathbf{A}_o \bar{\mathbf{J}}_0 \mathbf{A}_o^T$, thus

$$\mathbf{J} = \mathbf{A}_o \bar{\mathbf{J}}_e \bar{\mathbf{J}}_0^{-1} \mathbf{A}_o^T = \mathbf{A}_o \bar{\mathbf{J}}^{ij} \mathbf{A}_o^T \tag{12}$$

where $\bar{\mathbf{J}}^{ij} = \partial \bar{\mathbf{r}}^{ij} / \partial \bar{\mathbf{r}}_0^{ij}$, is defined in Equation (9). Therefore, the strain tensor with respect to the element coordinate system can be defined as

$$\epsilon_{\mathcal{L}} = \mathbf{A}_o \bar{\epsilon}_{\mathcal{L}} \mathbf{A}_o^T \tag{13}$$

$$\bar{\epsilon}_{\mathcal{L}} = \frac{1}{2} (\bar{\mathbf{J}}^T \bar{\mathbf{J}} - \mathbf{I}) \tag{14}$$

Because of the symmetry of the strain tensor, it is sufficient to identify only six strain components, three normal strains, and three shear strains, such that the strain vector can be written as

$$\boldsymbol{\epsilon} = [\epsilon_{xx} \quad \epsilon_{yy} \quad \epsilon_{zz} \quad 2\epsilon_{xy} \quad 2\epsilon_{xz} \quad 2\epsilon_{yz}]^T \tag{15}$$

The stress components of the element can be defined using the constitutive equation as follows

$$\boldsymbol{\sigma} = \mathbf{E} \boldsymbol{\epsilon} \tag{16}$$

where \mathbf{E} is the matrix of the elastic constants of the material [44]. The elastic forces of the element can be derived by using the following expression of the strain energy:

$$U = \frac{1}{2} \int_V \boldsymbol{\epsilon}^T \boldsymbol{\sigma} dV = \frac{1}{2} \int_V \boldsymbol{\epsilon}^T \mathbf{E} \boldsymbol{\epsilon} dV \tag{17}$$

The vector of the elastic forces, \mathbf{Q}_k^{ij} , that is produced due to deformation of element j , can be defined using the strain energy, U , as follows:

$$\mathbf{Q}_k^{ij} = \frac{\partial U^{ij}}{\partial \mathbf{e}^{ij}} \tag{18}$$

Several studies investigated the accuracy and usability of a continuum mechanics approach in the description of elastic forces of a three-dimensional beam element [45–47]. It should be mentioned that the strain tensor given by Equation (14) is defined as a function of the element coordinates ECS($x^{ij} y^{ij} z^{ij}$) and, therefore, the elastic forces are defined as a volume element that allows for describing the deformation of the beam cross-section.

2.3. Generalized External Forces

The principle of virtual work can be used to develop the vector of the generalized forces on the body i , element j , i.e., \mathbf{Q}_e^{ij} , that can be developed due to external Cartesian forces and moments. The virtual work carried out by applying an external force vector, acting on an arbitrary point on the element, can be written with respect to SCS (\mathbf{F}^{ij}) or BCS ($\bar{\mathbf{F}}^{ij}$) as follows

$$\delta W = \bar{\mathbf{F}}^{ijT} \delta \bar{\mathbf{r}}^{ij} = \mathbf{Q}_e^{ijT} \delta \mathbf{e}^{ij} \tag{19}$$

Substituting the value of $\bar{\mathbf{r}}$ from Equation (5) yields

$$\begin{aligned} \delta W &= \bar{\mathbf{F}}^{ijT} \delta \bar{\mathbf{r}}^{ij} \\ &= \bar{\mathbf{F}}^{ijT} \mathbf{S}^{ij} \delta \bar{\mathbf{e}}^{ij} \\ &= \bar{\mathbf{F}}^{ijT} \mathbf{S}^{ij} \mathbf{T}_0^{iT} \delta \mathbf{e}^{ij} \\ \mathbf{Q}_e^{ijT} &= \bar{\mathbf{F}}^{ijT} \mathbf{S}^{ij} \mathbf{T}_0^{iT} \end{aligned} \tag{20}$$

In the case of describing the generalized forces related to the unconstrained nodal coordinates of the body (i), i.e., \mathbf{p}^i , one can conclude that

$$\delta W = \bar{\mathbf{F}}^{ijT} \mathbf{S}^{ij} \mathbf{T}_0^{iT} \mathbf{B}_1^{ij} \mathbf{B}_2^i \delta \mathbf{p}^i \tag{21}$$

⇕

$$\mathbf{Q}_p^{iT} = \bar{\mathbf{F}}^{ijT} \mathbf{S}^{ij} \mathbf{T}_0^{iT} \mathbf{B}_1^{ij} \mathbf{B}_2^i \tag{22}$$

The absence of rotating angular coordinates in ANCF elements complicates the application of moments compared to the application of point forces. In the context of applied moments, it can be held that

$$\delta W = \bar{\mathbf{M}}^{ijT} \delta \bar{\boldsymbol{\theta}}^{ij} \tag{23}$$

which is the applied moment $\bar{\mathbf{M}}^{ij}$ at a given point on the body (i)-element (j) times the virtual change in orientation $\delta \bar{\boldsymbol{\theta}}^{ij}$, at that point. The virtual change $\delta \bar{\boldsymbol{\theta}}^{ij}$ can be expressed as

$$\delta \bar{\boldsymbol{\theta}}^{ij} = \tilde{\boldsymbol{\omega}}^{ij} \delta t \tag{24}$$

Therefore, the formulation of the virtual change in orientation $\delta \bar{\boldsymbol{\theta}}^{ij}$ can be obtained when the angular velocity for a material point, $\tilde{\boldsymbol{\omega}}^{ij}$, inside an ANCF element is defined. The vorticity tensor describes the angular velocity $\tilde{\boldsymbol{\omega}}^{ij}$ at the given point in the continuum element and can be expressed as follows [48]:

$$\tilde{\boldsymbol{\omega}}^{ij} = \begin{bmatrix} 0 & -\tilde{\omega}_z & \tilde{\omega}_y \\ \tilde{\omega}_z & 0 & -\tilde{\omega}_x \\ -\tilde{\omega}_y & \tilde{\omega}_x & 0 \end{bmatrix} = \frac{1}{2} (\mathbf{L} - \mathbf{L}^T) \tag{25}$$

where \mathbf{L} is the velocity gradient tensor that can be defined as

$$\mathbf{L} = \frac{\partial \dot{\bar{\mathbf{r}}}^{ij}}{\partial \bar{\mathbf{r}}^{ij}} \tag{26}$$

Equation (26) can be written in terms of the deformation gradient as follows:

$$\mathbf{L} = \frac{\partial \dot{\bar{\mathbf{r}}}^{ij}}{\partial \bar{\mathbf{r}}^{ij}} = \frac{\partial \dot{\bar{\mathbf{r}}}^{ij}}{\partial \bar{\mathbf{r}}_0^{ij}} \frac{\partial \bar{\mathbf{r}}_0^{ij}}{\partial \bar{\mathbf{r}}^{ij}} = \dot{\mathbf{J}}^{ij} (\mathbf{J}^{ij})^{-1} \tag{27}$$

Substituting the value of \mathbf{J}^{ij} , Equation (9), yields

$$\mathbf{L} = \dot{\mathbf{J}}_e^{ij} \mathbf{J}_0^{ij-1} \mathbf{J}_0^{ij} \mathbf{J}_e^{ij-1} = \dot{\mathbf{J}}_e^{ij} \mathbf{J}_e^{ij-1} \tag{28}$$

It can be seen that the velocity gradient \mathbf{L} can now be written in terms of the nodal coordinates and the time derivative of the nodal coordinates. Using Equation (25), one can conclude that

$$\bar{\omega}^{ij} = \begin{bmatrix} \bar{\omega}_x \\ \bar{\omega}_y \\ \bar{\omega}_z \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \mathbf{L}_{3,2} - \mathbf{L}_{2,3} \\ \mathbf{L}_{1,3} - \mathbf{L}_{3,1} \\ \mathbf{L}_{2,1} - \mathbf{L}_{1,2} \end{bmatrix} = \bar{\mathbf{G}}^{ij} \dot{\mathbf{e}} \tag{29}$$

Substituting the value of $\bar{\omega}^{ij}$ into Equation (24), and then into the virtual work of Equation (23), yields

$$\delta W = \bar{\mathbf{M}}^{ijT} \bar{\omega}^{ij} \delta t \tag{30}$$

$$= \bar{\mathbf{M}}^{ijT} \bar{\mathbf{G}}^{ij} \dot{\mathbf{e}}^{ij} \delta t \tag{31}$$

$$= \bar{\mathbf{M}}^{ijT} \bar{\mathbf{G}}^{ij} \delta \mathbf{e}^{ij} \tag{32}$$

$$= \bar{\mathbf{M}}^{ijT} \bar{\mathbf{G}}^{ij} \mathbf{T}_0^{iT} \delta \mathbf{e}^{ij} \tag{33}$$

Thus, the generalized forces due to external moments can be described as follows:

$$\mathbf{Q}_e^{ijT} = \bar{\mathbf{M}}^{ijT} \bar{\mathbf{G}}^{ij} \mathbf{T}_0^{iT} \tag{34}$$

$$\mathbf{Q}_p^{ijT} = \bar{\mathbf{M}}^{ijT} \bar{\mathbf{G}}^{ij} \mathbf{T}_0^{iT} \mathbf{B}_1^{ij} \mathbf{B}_2^i \tag{35}$$

The complete vector of the generalized forces, due to external forces and moments can be written as follows:

$$\mathbf{Q}_p^{ijT} = \left(\bar{\mathbf{F}}^{ijT} \mathbf{S}^{ij} + \bar{\mathbf{M}}^{ijT} \bar{\mathbf{G}}^{ij} \right) \mathbf{T}_0^{iT} \mathbf{B}_1^{ij} \mathbf{B}_2^i \tag{36}$$

2.4. Static Analysis

The equations of motion of the continuum robot structure, say, body i , can then be written in a matrix form in terms of the unconstrained generalized coordinates \mathbf{p} as

$$\mathbf{M}^i \ddot{\mathbf{p}}^i = \mathbf{Q}_k^{iT} + \mathbf{Q}_p^{iT} = \mathbf{Q}^{iT} \tag{37}$$

Note the complete set of generalized coordinates of the body, $\mathbf{e}^i = \mathbf{B}_1^{ij} \mathbf{B}_2^i \mathbf{p}^i$. The \mathbf{Q}_p^i is the vector of generalized external forces and moments that was discussed in the last section (Equation (36)) and \mathbf{Q}_k^i is the elastic forces, as expressed in Equation (18). The preceding equation accounts for all geometric nonlinearities since nonlinear strain-displacement relations are used (Equation (14)). Thus, the static analysis can be carried out by solving the following equation:

$$\mathbf{Q}^i(\mathbf{p}) = \mathbf{0} \tag{38}$$

The nonlinear equations can be solved using the Newton–Raphson iteration procedure in which the values of the nodal coordinates at step $(n + 1)$ can be calculated as

$$\mathbf{p}^{(n+1)} = \mathbf{p}^{(n)} + \Delta \mathbf{p} \tag{39}$$

$$\Delta \mathbf{p} = - \left[\frac{\partial \mathbf{Q}^{i(n)}}{\partial \mathbf{p}^{(n)}} \right]^{-1} \mathbf{Q}^{i(n)} \tag{40}$$

where $\partial \mathbf{Q}^{i(n)} / \partial \mathbf{p}^{(n)}$ is called a tangent stiffness matrix and vector $\mathbf{Q}^{i(n)}$ includes the elastic and external forces at iteration step n . In the solution process, the tangent stiffness matrix is calculated numerically using perturbations in the nodal coordinates and finite differences. The convergence criterion for the iteration is defined as follows:

$$|\Delta \mathbf{p}| < \varepsilon \tag{41}$$

where $|\cdot|$ is the Euclidean norm of the vector. Thus, given the Cartesian forces and/or moments, the generalized forces can be calculated, and the corresponding nodal coordinates can be obtained. Figure 2 shows the various configurations of continuum robots, whereby Cartesian forces are applied along the local x - and z - axes, and Cartesian moments are applied about the z - and y - axes to excite bending modes, where 12 elements of silicone rubber are forming the continuum robot. The last-node frame representing the tip point orientation (node 13) of both the un-deformed and deformed shapes are shown. The figures demonstrate the robot connectivity and continuity, while also validating the correctness of the elastic and external force formulation of the beam element. It is crucial to note that the derivation of the velocity transformation matrix $\bar{\mathbf{G}}^{ij}(\mathbf{e})$, which depends on the nodal coordinates, is significant in the context of applying moments to a cross-section undergoing rotation. The subsequent derivation of the elements of this matrix will be elaborated upon in the inverse solution.

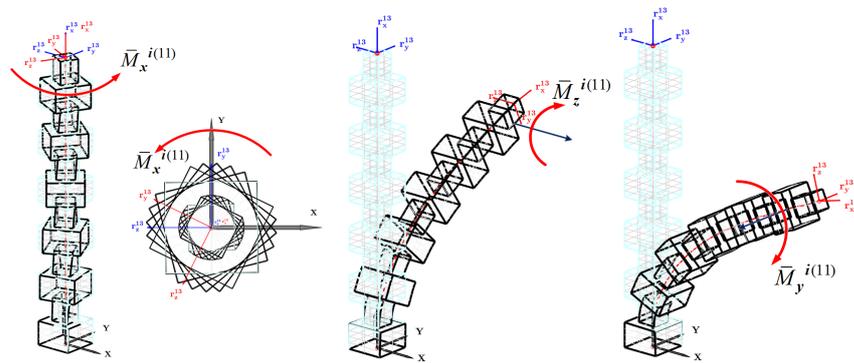


Figure 2. Effect of applying moments on continuum robot structure.

2.5. 3D-Beam Element with Circular Cross Section

A well-known structure of a 3D beam with a circular cross section, i.e., circular rods, is utilized in constructing continuum robots. In this case, the coordinates of the material point in the undeformed configuration with respect to the element frame are described in terms of the axial and radial displacements and an angle measured counter clockwise from the element frame. The coordinates of an arbitrary point can be described as $\bar{\mathbf{q}}^{ij}: [x^{ij} \ r^{ij} \ \theta^{ij}]^T$. The Cartesian coordinates are related to the cylindrical coordinates (see Figure 3), by

$$\left. \begin{aligned} x &= x \\ y &= r \cos(\theta) \\ z &= r \sin(\theta) \end{aligned} \right\} \tag{42}$$

Using Equation (42), the shape function can be described by the cylindrical coordinates, i.e., $\mathbf{S}^{ij}(\bar{\mathbf{u}}^{ij}) \rightarrow \mathbf{S}^{ij}(\bar{\mathbf{q}}^{ij})$. The transformation of the displacement field can be carried out as

$$\begin{bmatrix} \bar{u}_x \\ \bar{u}_y \\ \bar{u}_z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \bar{u}_x \\ \bar{u}_r \\ \bar{u}_\theta \end{bmatrix} \tag{43}$$

The Jacobian matrix can be described in cylindrical coordinates $\mathbf{q}:(x, r, \theta)$ as follows:

$$\bar{\mathbf{D}} = \frac{\partial \bar{\mathbf{u}}^{ij}}{\partial \bar{\mathbf{q}}^i} = \begin{bmatrix} \frac{\partial \bar{u}_x}{\partial x} & \frac{\partial \bar{u}_x}{\partial r} & \frac{\partial \bar{u}_x}{\partial \theta} \\ \frac{\partial \bar{u}_y}{\partial x} & \frac{\partial \bar{u}_y}{\partial r} & \frac{\partial \bar{u}_y}{\partial \theta} \\ \frac{\partial \bar{u}_z}{\partial x} & \frac{\partial \bar{u}_z}{\partial r} & \frac{\partial \bar{u}_z}{\partial \theta} \end{bmatrix} \tag{44}$$

Thus, the gradients of the displacement vector can be estimated as follows [49]:

$$\bar{\mathbf{J}}_e^{ij} = \begin{bmatrix} \bar{\mathbf{D}}_{1,1} & \bar{\mathbf{D}}_{2,1} & \frac{1}{r} \bar{\mathbf{D}}_{3,1} \\ \bar{\mathbf{D}}_{2,1} & \bar{\mathbf{D}}_{2,2} & \frac{1}{r} (\bar{\mathbf{D}}_{2,3} - \bar{u}_\theta) \\ \bar{\mathbf{D}}_{3,1} & \bar{\mathbf{D}}_{3,2} & \frac{1}{r} (\bar{\mathbf{D}}_{3,2} + \bar{u}_r) \end{bmatrix} \tag{45}$$

A similar procedure can be carried out to estimate $\bar{\mathbf{J}}_0^{ij}$, and then the deformation gradients $\bar{\mathbf{J}}^{ij}$ (Equation (9)) can be calculated. Based on the deformation gradient, the strain tensor of a beam element with a circular cross area can be calculated using Equation (14). The axial component ϵ_{xx} remains the same as for a rectangular cross section. Moreover, the variables of volume integration are the parameters (x, r, θ) and the volume integration boundaries are altered to $V^{ij} = \int_0^L \cdots \int_{-\pi}^{+\pi} \int_0^R \cdots r dr d\theta dx$. Using the mapped forms described above, the dynamic terms can be updated for circular cross section 3D beams.

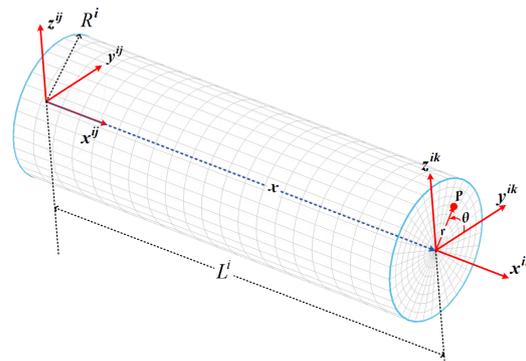


Figure 3. Beam element of circular cross section.

3. Deep-CNN Quasi-Statics Modeling

The primary objective of developing a deep CNN model is to construct a surrogate model that can effectively approximate the challenging inverse problem of ANCF statics modeling of continuum robots. The CNN model is intended to determine immediate solutions for the tip moments $\bar{\mathbf{M}}^i$ ¹³ that are necessary to achieve a specific tip pose, as defined by $\mathbf{p} \in \mathbb{R}^{12}$. In the current study, our focus was on the effects of constant moments applied to the robot tip. These moments, which could be implemented via applying tensions to cables in a cable-driven continuum robot, primarily induce pure bending. This type of bending is less likely to result in buckling deformation within the cross-section of the robot.

The primary distinction of the CNN utilized in our research lies in its inherent ability to effectively capture and analyze spatial correlations within the dataset. This aspect is particularly advantageous in our application, where the key features include the position and orientation of the robot’s tip. These features are inherently spatially correlated across the robot’s workspace, largely due to the physical constraints and specific dimensions of the robot. By doing so, the model can efficiently predict the required tip moments, thereby solving the inverse problem.

3.1. Data Collection and Processing

Data-driven modeling can be enhanced by increasing the size of the training dataset, resulting in more accurate predictions. However, this comes at the expense of substantial computational resources. Our research integrated the ANCF model, obtained in Equation (38), to generate the training datasets. We applied a set of moments $\mathbf{M} = [M_x, M_y, M_z]$ to the robot’s end-effector into the ANCF model and derived the robot’s corresponding poses $\mathbf{p} \in \mathbb{R}^{12}$ containing the position and gradients. The three moment values are swept one after the other across the entire range of the robot’s workspace. This dataset forms the foundation for our data-driven approach to model the robot’s behavior.

It can be quite time-consuming to explore all the possible combinations of moments for the robot’s tip. As a pragmatic approach, we generated a comprehensive dataset with 1520 data points. The collected data points were obtained within the expected workspace of the continuum robot, which has a fixed length of 40 cm. The sampling process was meticulously designed to ensure thorough coverage of the operational range of the robot, while also adhering to permissible moment limits, as depicted in Figure 4, showing the

positions and orientations of the robot’s end-effector. In order to ensure reliable model performance and generalization, the dataset was divided into three subsets: 70% for training, 15% for validation, and the remaining 15% for rigorous testing. This partitioning strategy was implemented to aid in achieving robust model training and evaluation.

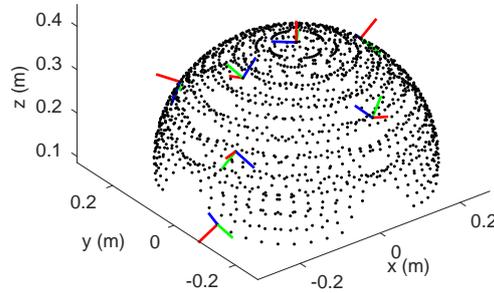


Figure 4. Sampled data points from the robot’s workspace were used as the training dataset for the CNN-based static modeling. The orientation of the robot’s end-effector is displayed as coordinate frames at selected data points, where the red, green, and blue axes represent the x , y , and z coordinates respectively.

In order to utilize CNNs for modeling the statics of the continuum robot, we must first convert each data point, represented as $p_i \in \mathbb{R}^{12}$, into an image. This is accomplished by stacking the robot tip position in the x , y , and z dimensions with the gradient vectors r as detailed in Figure 5. The resulting feature images, which represent the robot’s pose, are denoted as $\mathbf{I} \in \mathbb{R}^{1520 \times 3 \times 4}$. To enhance the model’s training stability, the training and testing data are normalized using the Z-score technique, which ensures that our data have a mean of $\mu = 0$ and a standard deviation of $\sigma = 1$. Each data point \mathbf{p} is normalized as follows:

$$\mathbf{p}' = (\mathbf{p} - \mu) / \sigma$$

It is important to note that the values of μ and σ are calculated solely based on the training data.

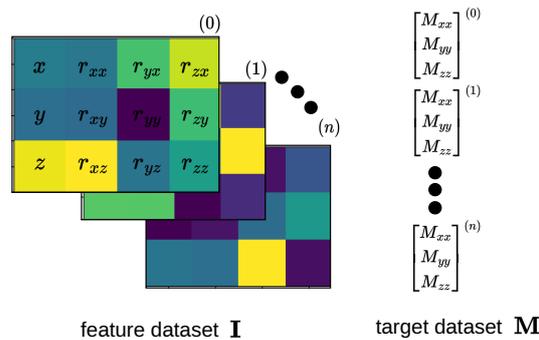


Figure 5. Stacking of robot’s position and gradients to form a 3×4 image representing the robot’s pose to be used in the CNN-based statics modeling of a continuum robot.

3.2. CNN Model Architectures

In this document, we present a CNN framework that has been specifically designed to approximate the inverse statics of the robot modeled using the ANCF. The input size of the network is fixed at 3×4 , corresponding to the representative image of the tip’s pose, while the objective of the network is to predict the required tip moments as its output. The CNN model begins with a convolutional layer, pooling layer, and flatten layer, followed by feeding the input data to the fully connected layers, as shown in Figure 6. The kernel size of the first convolution layer is chosen to be 16 while the hyperbolic tangent (tanh) activation function is selected.

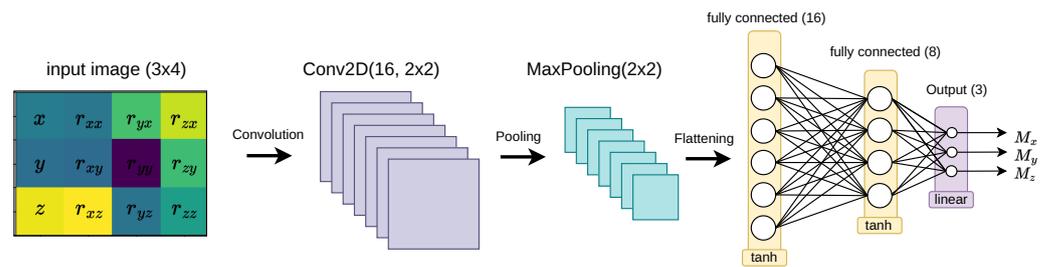


Figure 6. The CNN architectures employed in the inverse statics modeling of continuum robots.

3.3. Training of DNN Models

In the course of training our CNN models, we employed the Mean-Squared Error (MSE) loss function. This choice is of utmost significance in training the deep regressor model that approximates the inverse statics of the continuum robot. The MSE loss function effectively quantifies the divergence between the CNN model's estimated output, denoted as $\hat{y}(\zeta)$, and the corresponding reference output, y , which relates to either forward or inverse statics modeling. The Mean-Squared Error is expressed mathematically as follows:

$$\text{MSE} = \frac{1}{N} \sum_{i=0}^{N-1} [\hat{y}(\zeta)^i - y^i]^2 \quad (46)$$

The ζ symbol in the given equation denotes the network parameters, encompassing learnable weights and biases. To enhance the model's performance, the Adam optimizer [50] has been employed, with an initial learning rate of 10^{-2} as its hyperparameter. Furthermore, the customization of hyperparameters is warranted for individual numerical examples, such as the training batch size and the number of epochs, which will be explained in detail in the subsequent sections.

4. Results and Discussion

In this section, we conducted a series of numerical experiments to evaluate the effectiveness of the proposed CNN quasi-static models based on ANCF. Firstly, we assessed the performance of the CNN model by examining its training and validation losses throughout training, which allowed us to identify potential areas for improvement. Subsequently, we compared the trained model with the testing data collected, enabling us to establish its reliability and accuracy. Furthermore, we employed the proposed CNN-based inverse model to generate the moments required to track a specified trajectory in the task space, which yielded promising results. To determine the optimal model architecture, we conducted a five-fold cross-validation analysis on five models, enabling us to make informed decisions.

4.1. CNN Model Performance

The TensorFlow 2.x API was used to execute the training and prediction processes on a personal computer. It is worth noting that the choice of computer hardware significantly impacts the duration of the training process. This study used a central processing unit (CPU) with a clock speed of 2.8 GHz to train the models. Figure 7 depicts the Mean Squared Error (MSE) losses for the CNN inverse statics models across 500 epochs and their convergence towards minimum loss values in the training and validation datasets.

To evaluate the performance of the models, the testing dataset consisting of 512 samples with a batch size of 64 is used. A comparison between the CNN-based model's estimated and target data is represented in Figure 8, along with the corresponding absolute error. Due to space constraints, only the first 100 samples are displayed. Both models show a negligible error level, as evidenced by the close proximity between the estimated and target signals.

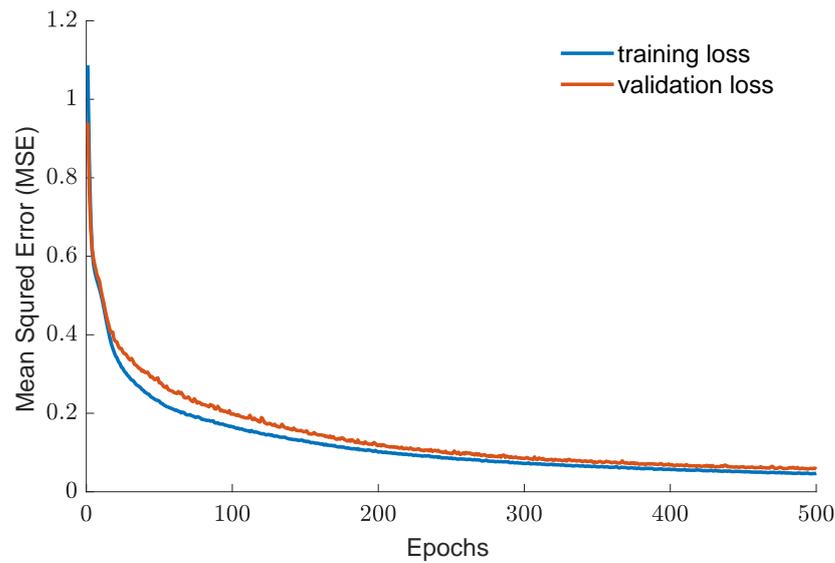


Figure 7. Graph showing the Mean Squared Error (MSE) of training and validation losses during the Convolutional Neural Network (CNN) training for the inverse static modeling of the continuum robot.

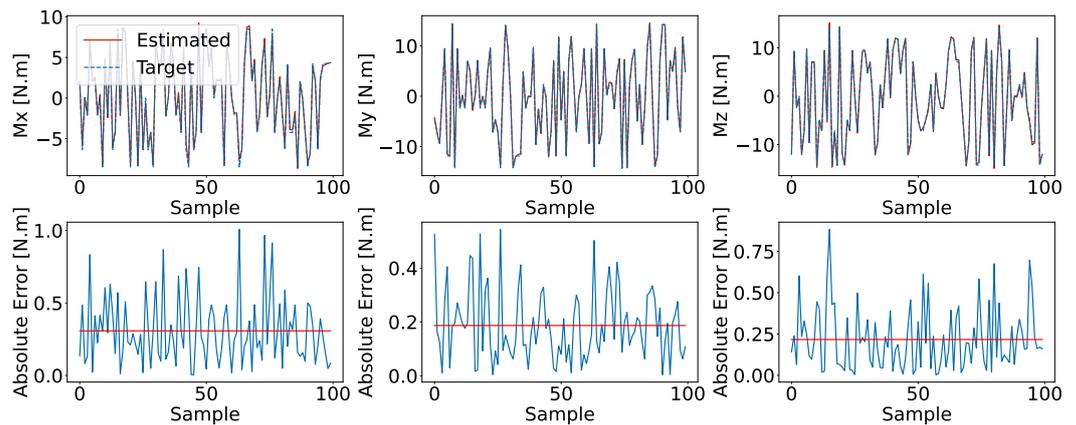


Figure 8. A comparison of the target versus estimated moments in the CNN-based inverse statics model, including the corresponding absolute error and the mean absolute errors highlighted with red horizontal lines.

4.2. Trajectory Generation

In order to evaluate the practicality of the proposed CNN-based quasi-inverse statics model, a reference trajectory has been provided, which specifies the desired poses over time for the robot’s tip to follow. The CNN model is responsible for computing the necessary moments to be applied at the robot’s tip to ensure it adheres to the reference trajectory. The reference trajectory is illustrated in Figure 9, highlighted in red and denoted as $\mathbf{p}_{ref} \in \mathbb{R}^{12}$ with the corresponding coordinates indicating the tip orientation. The reference trajectory is computed based on the Constant Curvature Model (CCM) [17], which is widely adopted in defining the kinematics of a single-segment continuum robot. The reference rotation trajectory \mathbf{R}_{ref} and the reference positions \mathbf{x}_{ref} are chosen as

$$\mathbf{R}_{ref} = \begin{bmatrix} C_\phi^2(C_\theta - 1) + 1 & S_\phi C_\phi(C_\theta - 1) & -C_\phi S_\theta \\ S_\phi C_\phi(C_\theta - 1) & C_\phi^2(C_\theta - 1) + 1 & -S_\phi S_\theta \\ C_\phi S_\theta & S_\phi S_\theta & C_\theta \end{bmatrix} \quad (47)$$

$$\mathbf{x}_{\text{ref}} = \begin{bmatrix} s C_{\phi} (C_{\theta} - 1) \\ s S_{\phi} (C_{\theta} - 1) \\ \frac{\theta}{s S_{\theta}} \\ \theta \end{bmatrix} \quad (48)$$

where $S_{\{\cdot\}}$ and $C_{\{\cdot\}}$ represent the sine and cosine functions of an angle, respectively. The s symbol represents the robot length of 40 cm, while θ and ϕ are the curvature and plan of curvature angle that are swept to generate the reference trajectory within the robot's workspace.

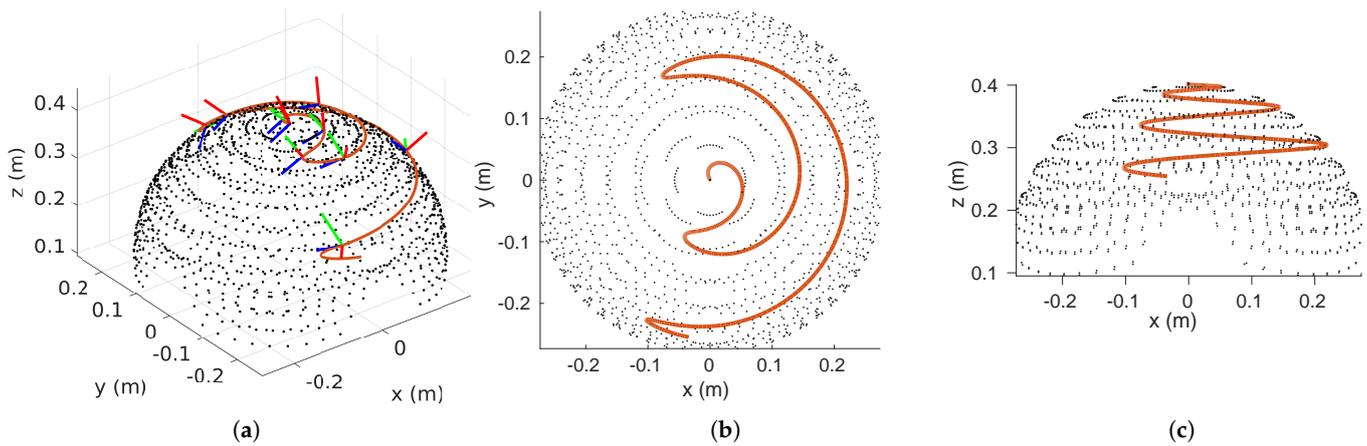


Figure 9. Reference trajectory generated to test the applicability of the proposed CNN inverse statics model of the continuum robot. (a) is showing the reference trajectory in orange over the dataset and the reference poses in RGB coordinates, (b,c) are the projection of the generated reference,

The CNN-based inverse statics model processes the input reference trajectory to provide the corresponding moments to be applied. These moments are seamlessly integrated into the analytical model derived from ANCF, as expressed in Equation (39). Figure 10 compares the trajectories generated using the CNN-based approach and the model-based approach, along with the associated tip-moment profiles. The observed similarities between both trajectories indicate the immense potential of the developed CNN-based inverse model for trajectory generation and control. While the error may seem significant, we believe it to be acceptable for two key reasons. Firstly, the larger discrepancies are predominantly observed along the z -axis, specifically below 0.2 m. As illustrated in Figure 9, this region of the workspace contains relatively few data points. This scarcity is a consequence of the range of moments applied during the dataset generation, which inherently limits the precision of solutions in this area. Secondly, the reference trajectory was constructed using the constant curvature model, primarily to ensure realistic orientations for the desired trajectory. However, it is important to note that the constant curvature model does not necessarily align precisely with the general ANCF model, which may contribute to the observed deviations.

4.3. K-Fold Cross Validation

K-fold cross-validation is a robust method for assessing the performance and generalizability of predictive models, particularly in fields such as machine learning and statistical analysis [51]. In this technique, the dataset is divided into k equal or nearly equal-sized subsets or folds. The process involves using one of these folds as the validation set to test the model, and the remaining $k - 1$ folds serve as the training set. This procedure is repeated k times, with each of the k subsets used exactly once as the validation data. The results from these k iterations are then averaged to produce a single estimation. This method is highly regarded for its ability to mitigate the bias associated with the random

sampling of the training and validation sets, as every data point is in a validation set exactly once and in a training set $k - 1$ times. It is particularly beneficial in situations where the available data are limited, as it maximizes both the training and testing data available for each iteration. By doing so, k-fold cross-validation provides a more comprehensive insight into the model’s performance, helping to identify issues such as overfitting or underfitting, and thus leading to more robust and reliable models.

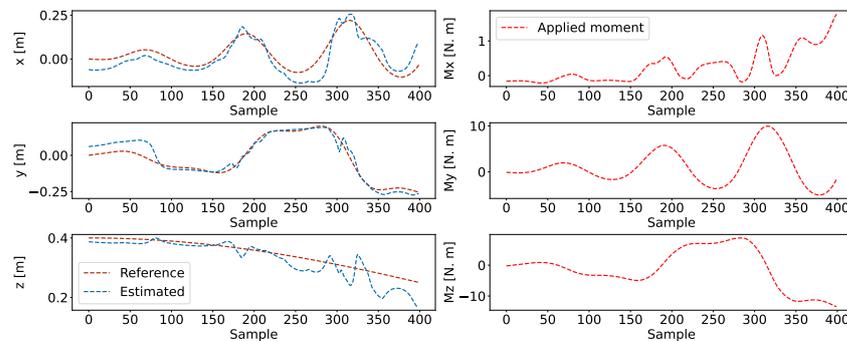


Figure 10. Comparative analysis of reference trajectories for CNN-based inverse statics testing versus trajectories derived from moments estimated by the CNN model.

In this subsection, the performance of the proposed CNN model was rigorously evaluated utilizing the K-Fold Cross-Validation method. This methodological approach is pivotal for extensively testing an inverse statics model across a range of architectural configurations, thereby ensuring their robustness and reliability. The dataset was partitioned into $k = 5$ distinct subsets, facilitating an iterative process of training and evaluating the model. Such a comprehensive assessment strategy enables a deeper understanding of the model’s performance under diverse conditions. More critically, this evaluation framework not only aids in selecting the most efficacious architectural configuration but also offers insightful perspectives on the generalization capabilities of the model in realistic scenarios.

In this research, a series of five Deep Neural Network (DNN) models were developed, utilizing both feed-forward and Convolutional Neural Network (CNN) architectures. The models encompass a range of trainable parameters, varying from 259 to 2515, and are structured across two to three layers, as elaborated in Table 1. Convolutional layers within these models are indicated by the prefix ‘C’, followed by a subscript denoting the number of kernels and the size. Fully connected layers are represented with the prefix ‘F’, accompanied by a subscript indicating the number of neurons. For convolutional layers, a stride of 1 was consistently employed, without the inclusion of padding. The activation functions were predominantly set to the hyperbolic tangent (tanh), except for the output layer where a linear activation function was implemented. In scenarios where the model lacked a convolutional layer at the input stage, a flattened layer was integrated to transform the input pose image into a column vector.

Table 1. Selected architectures for the five-fold cross validation experiment.

Model	Layers	Size	Average MSE
1	F_{16}, F_3	259	4.19
2	$C_{8,2}, F_3$	187	8.17
3	$C_{8,2}, C_{4,2}, F_3$	199	7.19
4	$C_{16,2}, C_{8,2}, F_3$	651	4.54
5	$C_{32,2}, C_{16,2}, F_8, F_3$	2515	1.07

The training of these models was executed using the Adam optimizer across 100 epochs, with Mean Squared Error (MSE) serving as the loss function. This process was further augmented by a five-fold cross-validation approach to ensure a comprehensive evaluation and validation. The convergence patterns of the learning algorithm, encompassing all

networks and folds, are depicted in Figure 11a. Additionally, Figure 11b provides a detailed analysis of the mean values and standard deviations of the MSE losses for each model architecture, specifically focusing on the CNN inverse statics models.

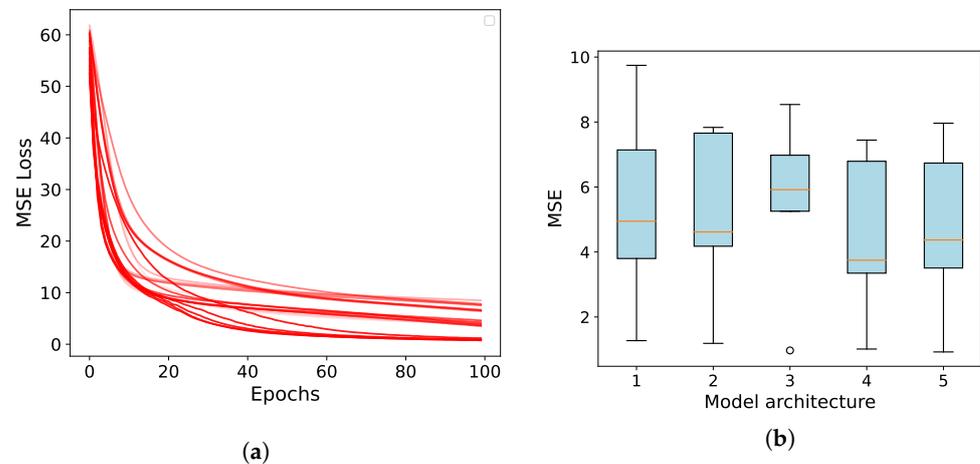


Figure 11. (a) Training losses expressed as mean squared error (MSE) for five deep neural network (DNN) models during five-fold cross-validation. (b) The mean and standard deviation of the MSE achieved by each model across the five folds.

Notably, networks with a greater number of parameters are represented in dark red, indicating that larger networks exhibit improved convergence towards smaller MSE values. To simplify our selection process, we opted for a neural network that performed reasonably well in both the training and testing phases. This exploration of compact and fast-learning architectures holds the promise of bolstering the overall system's robustness.

5. Conclusions

In conclusion, this research introduces a data-driven modeling approach, employing a deep convolutional neural network (CNN), tailored for real-time solutions in the inverse statics of soft continuum robots within the Absolute Nodal Coordinate Formulation (ANCF) framework. The CNN model is adeptly designed to address the spatial continuities and inherent nonlinearity of ANCF, facilitating the resolution of complex inverse quasi-statics in soft continuum robotics. The model demonstrates an acceptable accuracy in providing ANCF solutions with minimal tolerance levels, effectively tackling challenging inverse quasi-static problems in continuum robots.

Integrating the Deep CNN with the ANCF, as newly proposed in this research, marks a considerable improvement over the current approaches. It offers a more powerful and efficient solution for statics modeling in soft continuum robotics. This advancement facilitates the resolution of inverse statics problems for soft robots modeled using ANCF. In our system, the model is designed to receive a specified target position for the end-effector and then compute the requisite external moments at the robot's tip to achieve this position.

The study also underscores the utility of this approach in generating adequate trajectories for continuum robots by accurately determining the necessary moments to be applied at the robot's extremities. Through comprehensive numerical studies, the efficacy of the proposed method has been established, highlighting its applicability in practical scenarios. However, further research is required to extend this method to the dynamic modeling of continuum robots using the ANCF framework. Future work should also explore the integration of CNN-based control in continuum robotics, along with the experimental validation of the deep learning models, in order to further substantiate and enhance the applicability of these methods.

In the evolving landscape of continuum robotics, the integration of deep learning models in data-driven model predictive control (MPC), especially when applied to robots

modeled using the Absolute Nodal Coordinate Formulation (ANCF), presents a fertile ground for future research. The fusion of deep learning and MPC opens up innovative possibilities, ranging from enhanced learning algorithms capable of comprehending complex dynamics to the implementation of real-time control systems that can seamlessly integrate feedback and adjust to changing conditions. One of the paramount challenges in this domain is improving computational efficiency, which can be achieved through optimized neural network structures and the utilization of advanced hardware accelerators. This advancement is essential for ensuring the robustness and reliability of these systems across diverse operational environments, including industrial and medical applications.

Author Contributions: Conceptualization, A.A.N. and H.E.-H.; methodology, A.A.N. and H.E.-H.; software, H.E.-H. and A.A.N.; validation, I.A.H. and A.A.N.; formal analysis, investigation, I.A.H.; writing—original draft preparation, H.E.-H.; writing—review and editing, H.E.-H. and A.A.N.; visualization, I.A.H.; supervision, I.A.H.; project administration, H.E.-H.; funding acquisition, I.A.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANCF	Absolute Nodal Coordinate Formulation
CNN	Convolutional Neural Networks
MSE	Mean Square Error
CCM	Constant Curvature Model

References

1. Russo, M.; Sadati, S.M.H.; Dong, X.; Mohammad, A.; Walker, I.D.; Bergeles, C.; Xu, K.; Axinte, D.A. Continuum robots: An overview. *Adv. Intell. Syst.* **2023**, *5*, 2200367. [[CrossRef](#)]
2. Burgner-Kahrs, J.; Rucker, D.C.; Choset, H. Continuum robots for medical applications: A survey. *IEEE Trans. Robot.* **2015**, *31*, 1261–1280. [[CrossRef](#)]
3. Runciman, M.; Darzi, A.; Mylonas, G.P. Soft robotics in minimally invasive surgery. *Soft Robot.* **2019**, *6*, 423–443. [[CrossRef](#)] [[PubMed](#)]
4. Seleem, I.A.; El-Hussieny, H.; Ishii, H. Imitation-Based Motion Planning and Control of a Multi-Section Continuum Robot Interacting with the Environment. *IEEE Robot. Autom. Lett.* **2023**, *8*, 1351–1358. [[CrossRef](#)]
5. Stroppa, F.; Selvaggio, M.; Agharese, N.; Luo, M.; Blumenschein, L.H.; Hawkes, E.W.; Okamura, A.M. Shared-Control Teleoperation Paradigms on a Soft-Growing Robot Manipulator. *J. Intell. Robot. Syst.* **2023**, *109*, 30. [[CrossRef](#)]
6. Wooten, M.; Frazelle, C.; Walker, I.D.; Kapadia, A.; Lee, J.H. Exploration and inspection with vine-inspired continuum robots. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 5526–5533.
7. Li, G.; Yu, J.; Dong, D.; Pan, J.; Wu, H.; Cao, S.; Pei, X.; Huang, X.; Yi, J. Systematic Design of a 3-DOF Dual-Segment Continuum Robot for In Situ Maintenance in Nuclear Power Plants. *Machines* **2022**, *10*, 596. [[CrossRef](#)]
8. Shi, C.; Luo, X.; Qi, P.; Li, T.; Song, S.; Najdovski, Z.; Fukuda, T.; Ren, H. Shape sensing techniques for continuum robots in minimally invasive surgery: A survey. *IEEE Trans. Biomed. Eng.* **2016**, *64*, 1665–1678. [[CrossRef](#)]
9. Chikhaoui, M.T.; Burgner-Kahrs, J. Control of continuum robots for medical applications: State of the art. In Proceedings of the ACTUATOR 2018; 16th International Conference on New Actuators, Bremen, Germany, 25–27 June 2018; pp. 1–11.
10. Dupont, P.E.; Simaan, N.; Choset, H.; Rucker, C. Continuum robots for medical interventions. *Proc. IEEE* **2022**, *110*, 847–870. [[CrossRef](#)]
11. Li, G.; Wong, T.W.; Shih, B.; Guo, C.; Wang, L.; Liu, J.; Wang, T.; Liu, X.; Yan, J.; Wu, B.; et al. Bioinspired soft robots for deep-sea exploration. *Nat. Commun.* **2023**, *14*, 7097. [[CrossRef](#)]

12. Azami, O.; Ishibashi, K.; Komagata, M.; Yamamoto, K. Development of hydraulically-driven soft hand for handling heavy vegetables and its experimental evaluation. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023, pp. 2577–2583.
13. Car, M.; Ferreira, B.A.; Vuletic, J.; Orsag, M. Structured Ecological Cultivation with Autonomous Robots in Agriculture: Toward a Fully Autonomous Robotic Indoor Farming System. *IEEE Robot. Autom. Mag.* **2023**, *2*, 1–13. [[CrossRef](#)]
14. Seleem, I.A.; El-Hussieny, H.; Ishii, H. Recent Developments of Actuation Mechanisms for Continuum Robots: A Review. *Int. J. Control. Autom. Syst.* **2023**, *21*, 1592–1609. [[CrossRef](#)] [[PubMed](#)]
15. Della Santina, C.; Katzschmann, R.K.; Bicchi, A.; Rus, D. Soft Robotic Modeling and Control: Bringing Together Articulated Soft Robots and Soft-Bodied Robots. *Int. J. Robot. Res.* **2021**, *40*, 3–6. [[CrossRef](#)]
16. George Thuruthel, T.; Renda, F.; Iida, F. First-order dynamic modeling and control of soft robots. *Front. Robot. AI* **2020**, *7*, 95. [[CrossRef](#)]
17. El-Hussieny, H.; Hameed, I.A.; Zaky, A.B. Plant-Inspired Soft Growing Robots: A Control Approach Using Nonlinear Model Predictive Techniques. *Appl. Sci.* **2023**, *13*, 2601. [[CrossRef](#)]
18. El-Hussieny, H.; Hameed, I.A.; Ryu, J.H. Nonlinear model predictive growth control of a class of plant-inspired soft growing robots. *IEEE Access* **2020**, *8*, 214495–214503. [[CrossRef](#)]
19. AlAttar, A.; Hmida, I.B.; Renda, F.; Kormushev, P. Kinematic-Model-Free Tip Position Control of Reconfigurable and Growing Soft Continuum Robots. In Proceedings of the 2023 IEEE International Conference on Soft Robotics (RoboSoft), Singapore, 3–7 April 2023; pp. 1–7.
20. Lin, D.; Dong, X.; Yang, C. Position Tracking for Continuum Robots with Joint Limit Constraints. In Proceedings of the 2019 IEEE International Conference on Unmanned Systems and Artificial Intelligence (ICUSAI), Xi'an, China, 22–24 November 2019; pp. 39–44.
21. Armanini, C.; Boyer, F.; Mathew, A.T.; Duriez, C.; Renda, F. Soft Robots Modeling: A Structured Overview. *IEEE Trans. Robot.* **2023**, *39*, 1728–1748. [[CrossRef](#)]
22. Habibi, H.; Yang, C.; Kang, R.; Walker, I.D.; Godage, I.S.; Dong, X.; Branson, D.T. Modelling an Actuated Large Deformation Soft Continuum Robot Surface Undergoing External Forces Using a Lumped-Mass Approach. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 5958–5963. [[CrossRef](#)]
23. Webster, R.J.; Jones, B.A. Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review. *Int. J. Robot. Res.* **2010**, *29*, 1661–1683. [[CrossRef](#)]
24. Falkenhahn, V.; Mahl, T.; Hildebrandt, A.; Neumann, R.; Sawodny, O. Dynamic modeling of constant curvature continuum robots using the Euler-Lagrange formalism. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 2428–2433. [[CrossRef](#)]
25. Seleem, I.A.; Assal, S.F.; Ishii, H.; El-Hussieny, H. Demonstration-Guided Pose Planning and Tracking for Multi-Section Continuum Robots Considering Robot Dynamics. *IEEE Access* **2019**, *7*, 166690–166703. [[CrossRef](#)]
26. Seleem, I.A.; El-Hussieny, H.; Assal, S.F.; Ishii, H. Development and Stability Analysis of an Imitation Learning-Based Pose Planning Approach for Multi-Section Continuum Robot. *IEEE Access* **2020**, *8*, 99366–99379. [[CrossRef](#)]
27. Qi, F.; Chen, B.; Gao, S.; She, S. Dynamic model and control for a cable-driven continuum manipulator used for minimally invasive surgery. *Int. J. Med. Robot. Comput. Assist. Surg.* **2021**, *17*, e2234. [[CrossRef](#)]
28. Naughton, N.; Sun, J.; Tekinalp, A.; Parthasarathy, T.; Chowdhary, G.; Gazzola, M. Elastica: A Compliant Mechanics Environment for Soft Robotic Control. *IEEE Robot. Autom. Lett.* **2021**, *6*, 3389–3396. [[CrossRef](#)]
29. Huang, W.; Huang, X.; Majidi, C.; Jawed, M.K. Dynamic simulation of articulated soft robots. *Nat. Commun.* **2020**, *11*, 2233. [[CrossRef](#)] [[PubMed](#)]
30. Bieze, T.M.; Largilliere, F.; Kruszewski, A.; Zhang, Z.; Merzouki, R.; Duriez, C. Finite Element Method-Based Kinematics and Closed-Loop Control of Soft, Continuum Manipulators. *Soft Robot.* **2018**, *5*, 348–364. [[CrossRef](#)] [[PubMed](#)]
31. Grazioso, S.; Gironimo, G.D.; Siciliano, B. A Geometrically Exact Model for Soft Continuum Robots: The Finite Element Deformation Space Formulation. *Soft Robot.* **2019**, *6*, 790–811. [[CrossRef](#)] [[PubMed](#)]
32. du Pasquier, C.; Jeong, S.; Okamura, A.M. Finite Element Modeling of Pneumatic Bending Actuators for Inflated-Beam Robots. *IEEE Robot. Autom. Lett.* **2023**, *8*, 7416–7423. [[CrossRef](#)]
33. Sadati, S.M.; Naghibi, S.E.; Shiva, A.; Michael, B.; Renson, L.; Howard, M.; Rucker, C.D.; Althoefer, K.; Nanayakkara, T.; Zschaler, S.; et al. TMTDyn: A Matlab package for modeling and control of hybrid rigid–continuum robots based on discretized lumped systems and reduced-order models. *Int. J. Robot. Res.* **2021**, *40*, 296–347. [[CrossRef](#)]
34. Gravagne, I.A.; Rahn, C.D.; Walker, I.D. Large deflection dynamics and control for planar continuum robots. *IEEE/ASME Trans. Mechatron.* **2003**, *8*, 299–307. [[CrossRef](#)]
35. Wang, X.; Li, Y.; Kwok, K.W. A survey for machine learning-based control of continuum robots. *Front. Robot. AI* **2021**, *8*, 730330. [[CrossRef](#)]
36. Pierson, H.A.; Gashler, M.S. Deep learning in robotics: A review of recent research. *Adv. Robot.* **2017**, *31*, 821–835. [[CrossRef](#)]
37. Nada, A.A.; Hussein, B.A.; Megahed, S.M.; Shabana, A.A. Floating Frame of Reference and Absolute Nodal Coordinate Formulations in the Large Deformation Analysis of Robotic Manipulators: A Comparative Experimental and Numerical Study. In *Volume 4: 7th International Conference on Multibody Systems, Nonlinear Dynamics, and Control, Parts A, B and C*; ASME: New York, NY, USA, 2009; Volume 4, pp. 889–900. [[CrossRef](#)]

38. Shabana, A.A. *Dynamics of Multibody Systems*; University Press: Cambridge, UK, 2013; Volume 9781107042, pp. 1–384. [[CrossRef](#)]
39. Shabana, A.A. Definition of ANCF Finite Elements. *J. Comput. Nonlinear Dyn.* **2015**, *10*, 054506. . [[CrossRef](#)]
40. Shabana, A.A. An overview of the ANCF approach, justifications for its use, implementation issues, and future research directions. *Multibody Syst. Dyn.* **2023**, *58*, 433–477. [[CrossRef](#)]
41. Taylor, M.; Serban, R.; Negrut, D. An efficiency comparison of different ANCF implementations. *Int. J. Non-Linear Mech.* **2023**, *149*, 104308. [[CrossRef](#)]
42. Wang, T.; Mikkola, A.; Matikainen, M.K. An Overview of Higher-Order Beam Elements Based on the Absolute Nodal Coordinate Formulation. *J. Comput. Nonlinear Dyn.* **2022**, *17*, 091001. [[CrossRef](#)]
43. Bayoumy, A.; Nada, A.; Megahed, S. Modeling slope discontinuity of large size wind-turbine blade using absolute nodal coordinate formulation. In Proceedings of the ASME Design Engineering Technical Conference, Chicago, IL, USA, 12–15 August 2012; Volume 6. [[CrossRef](#)]
44. Nada, A.A.; Hussein, B.A.; Megahed, S.M.; Shabana, A.A. Use of the floating frame of reference formulation in large deformation analysis: Experimental and numerical validation. *Proc. Inst. Mech. Eng. Part K J. -Multi-Body Dyn.* **2010**, *224*, 45–58. [[CrossRef](#)]
45. García-Vallejo, D.; Mayo, J.; Escalona, J.L.; Domínguez, J. Efficient evaluation of the elastic forces and the jacobian in the absolute nodal coordinate formulation. *Nonlinear Dyn.* **2004**, *35*, 313–329. [[CrossRef](#)]
46. Sugiyama, H.; Suda, Y. A curved beam element in the analysis of flexible multi-body systems using the absolute nodal coordinates. *Proc. Inst. Mech. Eng. Part K J. Multi-Body Dyn.* **2007**, *221*, 219–231. [[CrossRef](#)]
47. Sugiyama, H.; Escalona, J.L.; Shabana, A.A. Spatial Joint Constraints in Flexible Multibody Systems Using the Absolute Nodal Coordinate Formulation. *Proc. ASME Des. Eng. Tech. Conf.* **2008**, *5A*, 467–476. [[CrossRef](#)]
48. Bechtel, S.E.; Lowe, R.L. *Fundamentals of Continuum Mechanics: With Applications to Mechanical, Thermomechanical, and Smart Materials*; Academic Press: Cambridge, MA, USA, 2015; p. 330.
49. Nada, A.; Al-Shahrani, A. Use of mixed coordinates in modeling wind turbines including tubular tower. *Mech. Sci.* **2019**, *10*, 35–46. [[CrossRef](#)]
50. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
51. Anguita, D.; Ghio, A.; Ridella, S.; Sterpi, D. K-Fold Cross Validation for Error Rate Estimate in Support Vector Machines. In Proceedings of The 2009 International Conference on Data Mining, Las Vegas, NV, USA, 13–16 July 2009; pp. 291–297.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.