A Dataset and Benchmark for Learning the Kinematics of Concentric Tube Continuum Robots

Reinhard M. Grassmann[1], *Student Member, IEEE,* Ryan Zeyuan Chen[1], *Student Member, IEEE,* Nan Liang[2], *Student Member, IEEE,* and Jessica Burgner-Kahrs[1], *Senior Member, IEEE*

***Abstract*— Establishing a physics-based model capturing the kinetostatic behavior of concentric tube continuum robots is challenging as elastic interactions between the flexible tubes constituting the robot result in a highly non-linear problem. The Goldstandard physics-based model using the Cosserat theory of elastic rods achieves reasonable approximations with $1.5 - 3$ % with respect to the robot's length, if well-calibrated. Learning-based models of concentric tube continuum robots have been shown to outperform the Goldstandard model with approximation errors below 1 %. Yet, the merits of learning-based models remain largely unexplored as no common dataset and benchmark exist.**

In this paper, we present a dataset captured from a three-tube concentric tube continuum robot for use in learning-based kinematics research. The dataset consists of 100 000 joint configurations and the corresponding four 6 dof sensors in $SE(3)$ measured with an electromagnetic tracking system (github.com/ContinuumRoboticsLab/CRL-Dataset-CTCR-Pose). With our dataset, we empower the continuum robotics and machine learning community to advance the field. We share our insights and lessons learned on joint space representation, shape representation in task space, and sampling strategies. Furthermore, we provide benchmark results for learning the forward kinematics using a simple, shallow feedforward neural network. The benchmark results for the tip error are 0.74 mm w.r.t. position (0.4 % of total robot length) and 6.49° w.r.t. orientation.

I. INTRODUCTION

Concentric tube continuum robots (CTCR) are composed of multiple concentric, pre-curved super-elastic tubes that are rotated and translated relative to one another [1], [2], see Fig. 1. Due to the elastic interactions between the tubes, highly non-linear behavior characterizes the mapping between task space and joint space of a CTCR. To reflect this interaction, model-based approaches [3] utilize static models based on the theory of Cosserat rods and are formulated as a boundary value problem which is solved numerically. In favour of computation time, some physical phenomena, e.g. friction or tube tolerances, are commonly neglected in these Goldstandard physics-based modeling approaches.

In recent years, alternative approaches have been investigated to mitigate the modelling effort regarding the complex elastic interactions between the tubes and hard-to-model physical phenomena [4]. Data-driven techniques based on neural networks have been applied to approximate the underlying continuous function. Neural networks have be applied to learn the forward kinematic (FK) mapping [5], [6], [7], [8], the inverse kinematic (IK) mapping [5], [6], [9], and motion generation [10], [11] while implicitly modelling the kinematics.

The feasibility of learning-based approaches have been shown in simulation [5], [9], [10], [11] and with data collected from a physical CTCR prototype [6], [7], [8], [12]. However, those datasets are not publicly available and thus limiting the advancement of learning-based approaches since demanding and time-consuming data acquisition depends on the availability of physical prototypes [13] and additional hardware such as sensing.

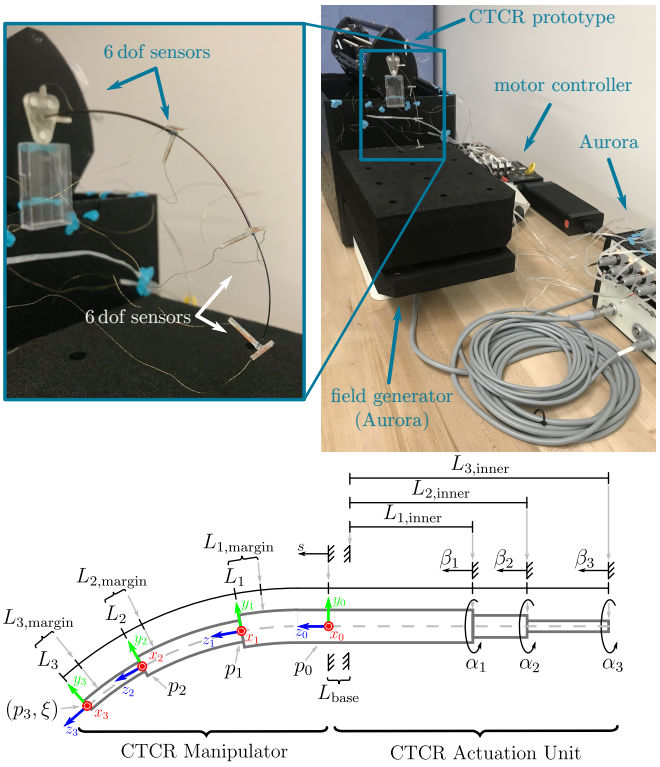

Fig. 1: Testbed with CTCR and used notation. The CTCR has three tubes with 6 dof sensors at each distal end measuring the pose by an electromagnetic tracking system called Aurora. Note that we follow a slightly different convention regarding the numbering and assignment of the tubes.

We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), [RGPIN-2019-04846].

[1]Reinhard M. Grassmann, Ryan Zeyuan Chen, and Jessica Burgner-Kahrs are with Continuum Robotics Laboratory, Department of Mathematical and Computational Sciences, University of Toronto, Mississauga, ON L5L 1C6, Canada reinhard.grassmann@utoronto.ca

[2]Nan Liang is with Autonomous Systems and Biomechatronics Laboratory, Mechanical and Industrial Engineering, University of Toronto, Toronto, ON M5S 3G8, Canada

In this paper, we present the first public dataset captured from a three-tube concentric tube continuum robot prototype. The dataset encompasses $100\,000$ measurements with full pose information for the tip of each tube as well as full joint space, i.e. relative tube translation and rotation. In addition to the dataset, we provide our insights on suitable joint space representation, task space representation, and one-shot sampling methods. We present results in approximating the forward kinematics using the dataset for different task space representations to serve as a benchmark for future research. Lastly, we share our lessons learned on learning-based approaches for CTCR and discuss open challenges in this domain as a call to the community.

II. METHODS

In this section, the different representation, performance, and error metrics are presented. Without loss of generality, in the following, we consider a typical CTCR with three tubes.

A. Joint Space Representation

Due to the nested tubes of a CTCR, each translation β_i has to satisfy the inequalities given by

$$0 \geq \beta_1 \geq \beta_2 \geq \beta_3 \quad \text{and} \tag{1}$$

$$0 \leq L_1 + \beta_1 \leq L_2 + \beta_2 \leq L_3 + \beta_3, \tag{2}$$

where L_i is the total length of the i^{th} tube. The interdependencies derive from the fact that none of the distal ends of inner tubes can be pulled inside the respective outer tube. To circumvent both inequalities (1) and (2), the translations β_i can be transformed, which decorrelates the translational joint values [7]. The transformation matrix

$$\boldsymbol{M}_{\mathcal{B}} = \begin{bmatrix} -L_1 & 0 & 0 \\ -L_1 & L_1 - L_2 & 0 \\ -L_1 & L_1 - L_2 & L_2 - L_3 \end{bmatrix} \tag{3}$$

follows the used convention regarding the numbering of the tubes, see Fig. 1. By computing $\boldsymbol{M}_{\mathcal{B}}^{-1} \left[\beta_1, \beta_2, \beta_3\right]^\top$, transformed values are in the interval $[0, 1]$. By adding an additional scaling and shifting to (3), the interval $[-1, 1]$ can be enforced, which leads to the system of linear equations

$$\begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\boldsymbol{M}_{\mathcal{B}} & \frac{1}{2}\boldsymbol{M}_{\mathcal{B}} \cdot \boldsymbol{1}_{3\times 1} \\ \boldsymbol{0}_{1\times 3} & 1 \end{bmatrix} \begin{bmatrix} \beta_{1,\mathcal{U}} \\ \beta_{2,\mathcal{U}} \\ \beta_{3,\mathcal{U}} \\ 1 \end{bmatrix}, \tag{4}$$

where $\beta_{i,\mathcal{U}} \in [-1, 1]$ is the transformed β_i.

The axial rotations α_i are independent and can be transformed be means of trigonometric functions [6], i.e. $\left[\gamma_{i,1}, \gamma_{i,2}\right]^\top = \left[\cos\left(\alpha_i\right), \sin\left(\alpha_i\right)\right]^\top$. This representation is a unit vector in the Euclidean space, where the entries are orthogonal to each other and lie between -1 and 1.

Combining all transformations, the final representation is

$$\boldsymbol{q} = \left[\gamma_{1,1}, \gamma_{1,2}, \beta_{1,\mathcal{U}}, \gamma_{2,1}, \gamma_{2,2}, \beta_{2,\mathcal{U}}, \gamma_{3,1}, \gamma_{3,2}, \beta_{3,\mathcal{U}}\right]^\top, \tag{5}$$

where all values of (5) are automatically scaled between -1 and 1, unit-less,orthogonal.

B. Shape Representation in Task Space

Position and orientation are used to describe the CTCR shape in task space. We employ the position of the distal end of each tube, see Fig. 1. In addition, the orientation of the innermost tube is used, where quaternions are chosen as they were found to outperform other alternative representations in learning the forward kinematics [7].

The shape representation in task space is given by

$$\boldsymbol{x} = \left[x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3, \eta, \varepsilon_1, \varepsilon_2, \varepsilon_3\right]^\top, \tag{6}$$

where, $\left[x_i, y_i, z_i\right]^\top = \boldsymbol{p}_i$ are positions and $\left[\eta, \varepsilon_1, \varepsilon_2, \varepsilon_3\right]^\top = \xi$ describes the quaternion of the tip pose with the notion described in [7]. The representation (6) can be simplified to tip pose, tip position, or three points along the shape arranged at every distal end of each tube. This leads to

$$\boldsymbol{x}_{\text{pose}} = \left[x_3, y_3, z_3, \eta, \varepsilon_1, \varepsilon_2, \varepsilon_3\right]^\top, \tag{7}$$

$$\boldsymbol{x}_{\text{position}} = \left[x_3, y_3, z_3\right]^\top, \text{and} \tag{8}$$

$$\boldsymbol{x}_{\text{three-points}} = \left[x_1, y_1, z_1, x_2, y_2, z_2, x_3, y_3, z_3\right]^\top. \tag{9}$$

All four representations are considered in this paper.

To measure $\boldsymbol{x}_{\text{pose}}$ for a CTCR, a sensor needs to be placed at the distal end of the most inner tube. To avoid damaging the sensor, a safety margin is added which may reduce the task space. Repeating this process for all distal ends of each tube leads to the placement of sensors along the CTCR manipulator as shown in Fig. 1. This sensor placement allows the measuring of all four representations.

C. Approximation Error

For the sake of completeness, we describe translational and rotational errors for the FK and IK. The hat symbol is used to denote an approximated value.

1) Cartesian space: Two approximation errors for the FK can be defined, i.e. translational error and rotational error. For the i^{th} tube, translational error $e_{t,i}$ is given by

$$e_{t,i} = \sqrt{\left(x_i - \hat{x}_i\right)^2 + \left(y_i - \hat{y}_i\right)^2 + \left(z - \hat{z}_i\right)^2}, \tag{10}$$

whereas the rotational error e_ϑ is evaluated via

$$\begin{aligned} e_\vartheta = \min\{\, & 2\arccos\left(\eta\hat{\eta} + \varepsilon_1\hat{\varepsilon}_1 + \varepsilon_2\hat{\varepsilon}_2 + \varepsilon_3\hat{\varepsilon}_3\right), \\ & 2\arccos\left(\eta\hat{\eta} - \varepsilon_1\hat{\varepsilon}_1 - \varepsilon_2\hat{\varepsilon}_2 - \varepsilon_3\hat{\varepsilon}_3\right) \,\} \end{aligned} \tag{11}$$

respecting the antipodal property of quaternions. For simplicity, the tip position error is denoted by $e_t = e_{t,n}$ with n being the most inner tube.

2) Joint space: Both errors in the joint space are described in terms of Euclidean distance. The error

$$e_\beta = \sqrt{\sum_i^3 e_{\beta,i}^2} \quad \text{with} \quad e_{\beta,i} = |\beta_i - \hat{\beta}_i|. \tag{12}$$

described the translational error, whereas the rotational error

$$e_\alpha = \sqrt{\sum_i^3 e_{\alpha,i}^2} \quad \text{with} \quad e_{\alpha,i} = |\alpha_i - \hat{\alpha}_i| \tag{13}$$

is formulated similarly with the retrieved angles $\alpha_i = \operatorname{atan2}\left\{\gamma_{2,i}, \gamma_{2,i}\right\}$ and $\hat{\alpha}_i = \operatorname{atan2}\left\{\hat{\gamma}_{2,i}, \hat{\gamma}_{2,i}\right\}$.

D. Random Sampling in the Joint Space

In this section, we provide general approaches which are applicable to restricted joint spaces. Each sample is generated from an uniform distribution $\mathcal{U}[-1,1]$. Without loss of generality, we will state the sampling method for three tubes. Note that, to efficiently sample and transform p configurations, the sampling methods can be stacked and vectorized, which is important for computationally intensive applications such as machine learning.

1) Rotational Joint Space: Due to the independence of each α_i, sampling in the rotational joint space is straightforward, i.e. the j^{th} value can be computed by $\alpha_i^{(j)} = \pi\,\mathcal{U}]-1;1]$, where the property of $\mathbb{S}^1$ is considered, i.e. $\alpha_i = \alpha_i + 2\pi k$ for $k \in \mathbb{N}$. In the case of a symmetrically restricted joint space, i.e. $-\alpha_{\text{min}} = \alpha_{\text{max}}$, and $\alpha_{\text{max}} < \pi$, i.e. no consideration of the wrap-around property of unit circle $\mathbb{S}^1$ is needed,

$$\begin{bmatrix}\alpha_1^{(j)}\\ \alpha_2^{(j)}\\ \alpha_3^{(j)}\end{bmatrix} = \alpha_{\text{max}} \begin{bmatrix}\mathcal{U}[-1;1]\\ \mathcal{U}[-1;1]\\ \mathcal{U}[-1;1]\end{bmatrix} \tag{14}$$

is the sampling method for the rotational joint space. Afterwards, each j^{th} value can be expressed as $\gamma_{i,1}^{(j)}$ and $\gamma_{i,2}^{(j)}$.

2) Translational Joint Space: The translational parameters β_i are interdependent, see (1) and (2). The inverse of transformation (3) disentangles the inequalities (1) and (2) as first shown in [7]. Subsequently, when applying (3) to a value between 0 and 1, a value is generated such that (1) and (2) hold. In the following, we consider L_i as the maximum tube length in the manipulator. Hence, for the unrestricted joint space, L_i is equal to the absolute value of the minimum displacement of β_i. Note that $\beta_i \leq 0$. The minimum displacement denoted by $\beta_{i,\text{min}}$ can be further restricted by considering a margin $L_{i,\text{margin}}$, see Fig. 1. In this case, we can define the minimum displacement as

$$\beta_{i,\text{min}} = -\left(L_i - \sum_{k=i}^{n} L_{k,\text{margin}}\right) = -L_i^*, \tag{15}$$

where the restriction is applied in a recursive manner. By substituting L_i in (3) with the new length $L_i^* = L_i - \sum_{k=i}^{n} L_{k,\text{margin}}$ and neglecting the homogeneous extension of right side of (4), we can state a sampling method for one sample, which is given by

$$\begin{bmatrix}\beta_1^{(j)}\\ \beta_2^{(j)}\\ \beta_3^{(j)}\end{bmatrix} = \frac{1}{2}\begin{bmatrix}\boldsymbol{M}_{\mathcal{B}}^* & \boldsymbol{M}_{\mathcal{B}}^* \cdot \mathbf{1}_{3\times 1}\end{bmatrix} \begin{bmatrix}\mathcal{U}[-1;1]\\ \mathcal{U}[-1;1]\\ \mathcal{U}[-1;1]\\ 1\end{bmatrix}, \tag{16}$$

where the star notation indicates the consideration of L_i^* in $\boldsymbol{M}_{\mathcal{B}}$ leading to $\boldsymbol{M}_{\mathcal{B}}^*$.

III. DATA ACQUISITION

Here, we provide implementation details on the data acquisition and a brief overview of the testbed with the used physical CTCR prototype. Moreover, we describe how we rearranged p random configurations such that the execution time on the physical CTCR prototype can be reduced.

A. Experimental Setup

To gather measurements for the dataset, we measure poses at the distal end of each tube and the base as indicated in Fig. 1. The constructed testbed is depicted in Fig. 1.

1) Tracking System: By using two $5\,\text{dof}$ electromagnetic tracking coil sensors being rigidly attached to each other, all $6\,\text{dof}$ of $SE(3)$ can be measured using an electromagnetic tracking system (AURORA v2, Northern Digital Inc., ON, Canada). We found that by using two $5\,\text{dof}$ coils instead of one $6\,\text{dof}$ coil, the roll about the tube's main axis can be measured straightforwardly. With this electromagnetic tracking, we can measure position in sub-millimetre (RMS of $0.7\,\text{mm}$) and orientation in sub-degree (RMS of $0.20°$) range without the line-of-sight restrictions of a camera system.

2) Concentric Tube Continuum Robot: The CTCR prototype is depicted in Fig. 1. It consists of three sliding carriages and six actuators (DCX 16 L, Maxon Motor AG, OW, Switzerland). Super-elastic pre-curved tubes (EUROFLEX GmbH, BW, Germany) made of Nickel-Titanium (Nitinol) are used. The manufacturer used a proprietary heat shape setting method to pre-bent the tubes. The geometrical parameters and mechanical properties of the used tube set are listed in Table I. Each tube is composed of a straight and a curved section with constant curvature. Note that the overall length L_i of each tube is not the physical length of the tube. By reducing the thickness of the carriage and the front plate of the real robot to an infinitesimal small thickness, the overall length L_i is determined. In our case, the straight length of the respective tube is shortened. This approach is similar to the configuration approach in [14] and gives the advantage that $\beta_i \in [-L_i, 0]$ subject to both inequalities, i.e. (1) and (2).

TABLE I: Tube parameters of the CTCR.

Parameter			Tubes[1]		
Term	Symbol	Unit	inner	middle	outer
Length, overall	L_i	mm	210	165	110
Length, curved	$L_{i,\text{c}}$	mm	41	100	100
Curvature[2]	$\kappa_{i,x}$	m^{-1}	28	12.4	4.37
Diameter, outer[2]	$d_{i,\text{outer}}$	mm	0.5	0.9	1.5
Diameter, inner[2]	$d_{i,\text{inner}}$	mm	0.4	0.7	1.2
Young's Modulus	E	GPa	50	50	50
Poisson's ratio	ν	1	0.3	0.3	0.3
Margin	$L_{i,\text{margin}}$	mm	10	10	60
Collet chuck to base[3]	$L_{i,\text{inner}}$	mm	143	51	5

[1] The outermost, middle, and innermost tubes are referenced with the index 1, 2, and 3, respectively. See Fig. 1 for reference.
[2] Values are taken from manufactures's data sheet.
[3] For completeness, $L_{\text{base}} = 16\,\text{mm}$, see Fig. 1.

B. Initialization of the CTCR

All joints are adjusted by defining the zero position for each joint. Each β_i value is set to zero for a specific reference position of the respective carriage on the rail. For α_i, we use the definition and adjustment device for the rotational zero position described in [6]. Therefore, α_i are iteratively adjusted such that the shape of the tubes lies in the yz

TABLE II: Comparison of the complete and restricted joint spaces. It is worth noting that the absolute value of the minimum value of β_i is not the overall length of the respective tube.

	complete joint space[4]				restricted joint space			
	α_i in rad		β_i in mm		α_i in rad		β_i in mm	
i	min	max	min	max	min	max	min	max
1	$-\pi$	π	-110	0	$-\pi/3$	$\pi/3$	-50	0
2	$-\pi$	π	-165	0	$-\pi/3$	$\pi/3$	-155	0
3	$-\pi$	π	-210	0	$-\pi/3$	$\pi/3$	-200	0

[4] The rotational joints are mechanically not restricted and could turn beyond π rad in clockwise and counter-clockwise directions.

plane of the base. One advantage to other iterative dithering methods based on tip position [15], [16] for initialization is the consideration of the whole shape being a set of points along the backbone.

C. Restricted Joint Space

The joint space $\mathcal{Q}$ is restricted for four practical reasons: to limit the influence of the gravitational force of the electromagnetic sensors, to avoid damage to the attached sensors, to avoid interference between two adjacent sets of $5\,\mathrm{dof}$ sensors, and to increase the density of the data points.

Regarding the gravitational force, note that the diameter and thickness of the each tube are listed in Table I and the weight of all three sensors attached to the distal end is around $0.6\,\mathrm{g}$. Further note that the weight of the base sensor attached to the CTCR actuation unit does not effect the CTCR manipulator. To avoid damage to the attached sensor at the distal end of each tube and interference, offsets $L_{i,\text{margin}}$ such that each tube always extends by $10\,\mathrm{mm}$ are respected. The deployment of the most outer tube is further restricted such that the considered margins are $L_{1,\text{margin}} = 60\,\mathrm{mm}$, $L_{2,\text{margin}} = 10\,\mathrm{mm}$, and $L_{3,\text{margin}} = 10\,\mathrm{mm}$. The complete and restricted joint spaces are listed in Table II.

D. Short Path in Joint Space via Tick-Space $\mathcal{T}$

By arranging the computed p randomized configurations $\boldsymbol{q}^{(1)}, \boldsymbol{q}^{(2)}, \cdots, \boldsymbol{q}^{(p)} \in \mathcal{Q}$, the execution time for data acquisition can be reduced. An optimal point-to-point path planning in $\mathcal{Q}$ is linked to the problem of the *Travelling Salesman Problem*. Here, a suboptimal solution is used, which is yet effective and efficient to reduce the execution time on the physical hardware as opposed to simply executing the unordered, random joint space samples.

All p configurations $\boldsymbol{q}^{(p)}$ are transformed into space $\mathcal{T}$, which we denote ticks-space $\mathcal{T}$. For our CTCR prototype, all actuators and encoders are equal. Only transmission ratio $u_\alpha = 177.1$ and $u_\beta = 42\,497$ for α_i and β_i, respectively, are different. Therefore, transforming $\boldsymbol{q}^{(p)}$ into ticks-space $\mathcal{T}$ is feasible. For a given minimal angle $\alpha_{i,\min} = \min_p \alpha_i^{(p)}$ of p configurations, u_α, and u_β, a distance

$$\mathrm{D}_{\mathcal{T}}(\boldsymbol{q}) = \sum_{i=1}^{n} \left(w_i u_\alpha \left(\alpha_i - \alpha_{i,\min} \right) - w_{n+i} u_\beta \beta_i \right) \geq 0 \quad (17)$$

can be defined in $\mathcal{T}$ for n tubes, where w_i and w_{n+i} are semi-positive weights. Note that (17) it ignores the "wrap-around" property of an unrestricted α_i. However, for our application, this is not important due to the restricted joint space for the data acquisition with a real robot prototype, see Table II.

To show the effectiveness of (17), we measure two durations of real robot movements for 8000 random configurations including the measuring process. First, all random configurations are computed and, afterwards, executed by the robot prototype. The duration without applying (17) is $26.1\,\mathrm{h}$. Second, distances between generated random configurations and $[\alpha_{1,\min}, 0, \alpha_{2,\min}, 0, \alpha_{3,\min}, 0]$ are computed by means of (17) with $\omega_1 = \omega_2 = \omega_3 = 0.1$ and $\omega_4 = 2\omega_5 = 4\omega_6 = 0.4$. Following this, the configurations are ordered by the distances and executed by the robot. Applying the strategy reduces the duration of the movements by $49\,\%$ to $13.3\,\mathrm{h}$. The reduction is due to the shortening of the translational movement, which represents the temporal bottleneck. This can also be seen from transmission ratios, i.e. the rotation $u_\alpha = 177.1$ motor ticks per degree and the translation $u_\beta = 42\,497$ motor ticks per millimetre, as well as form the sampling range, see Table II.

E. Orientation and Position

The electromagnetic tracking system provides quaternions $\left[\eta_i^{(j)}, \varepsilon_{i,1}^{(j)}, \varepsilon_{i,2}^{(j)}, \varepsilon_{i,3}^{(j)}\right]^\top$ as an orientation representation and position vectors $\boldsymbol{t}_i^{(j)}$ as a position representation. To reduce the error due to sensor noise, we record $m = 5$ poses for each sample and sensor. Consequently, an appropriate pose must be determined from $m = 5$ measurements. This approach has been previously presented in [6]. Note that four different poses with index i are measured, see Fig. 1, and, therefore, this approach is applied to each pose separately.

1) Orientation: Giving that $[\eta_i, \varepsilon_{i,1}, \varepsilon_{i,2}, \varepsilon_{i,3}]^\top$ is the true value and $\left[\eta_i^{(j)}, \varepsilon_{i,1}^{(j)}, \varepsilon_{i,2}^{(j)}, \varepsilon_{i,3}^{(i)}\right]^\top$ is measured, both quaternions must satisfied the unit length condition given by

$$1 = \eta_i \eta_i^{(j)} + \epsilon_{i,1} \epsilon_{i,1}^{(j)} + \epsilon_{i,2} \epsilon_{i,2}^{(j)} + \epsilon_{i,3} \epsilon_{i,3}^{(j)}. \quad (18)$$

Expressing (18) in matrix form with m equations leads to

$$\mathbf{1}_{m\times 1} = \boldsymbol{Q}_{\mathrm{m},i} \left[\eta_i \quad \epsilon_{i,1} \quad \epsilon_{i,2} \quad \epsilon_{i,3}\right]^\top. \quad (19)$$

By taking advantage of the Gaussian least squares approach and rearranging the above equation leads to

$$\left[\eta_i \quad \epsilon_{i,1} \quad \epsilon_{i,2} \quad \epsilon_{i,3}\right]^T = \left(\boldsymbol{Q}_{\mathrm{m},i}^\top \boldsymbol{Q}_{\mathrm{m},i}\right)^{-1} \boldsymbol{Q}_{\mathrm{m},i}^\top \mathbf{1}_{m\times 1} \quad (20)$$

yielding the sought-after quaternion with the smallest quadratic error, where $\boldsymbol{Q}_{\mathrm{m},i} \in \mathbb{R}^{5\times 4}$ is the design matrix and $\mathbf{1}_{m\times 1} = [1,1,1,1,1]^\top$ with $m = 5$. Subsequently, the quaternion is normalized with its Euclidean length.

2) Position: Choosing a similar approach to (20) being $\mathbf{0} = \boldsymbol{t}_i - \boldsymbol{t}_i^{(j)}$ leads to the mean position. Therefore,

$$\boldsymbol{t} = \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{t}^{(j)} \quad \text{with } m = 5 \quad (21)$$

TABLE III: Annotation of the dataset in the CSV file. The k^{th} sample consists of six absolute joint values, six relative joint values, and four poses as quaternion/vector pair, where pose with $i = 0$ is the base pose. For clarity, Fig. 1 illustrates the index assignment. Note that $\boldsymbol{q}_0$ has zero values for α_i and β_i per definition.

												i^{th} pose for $i = 0, 1, 2, 3$ (These columns repeat four times)						
Joint configuration $\boldsymbol{q}_k$ α_i in rad β_i in mm						Joint configuration difference $\Delta\boldsymbol{q}_k = \boldsymbol{q}_k - \boldsymbol{q}_{k-1}$ $\Delta\alpha_i$ in rad and $\Delta\beta$ in mm						Position $\boldsymbol{t}$ $[x, y, z]^\top$ in mm			Orientation $[\eta, \varepsilon_1, \varepsilon_2, \varepsilon_3]^\top$ unit less			
α_1	β_1	α_2	β_2	α_3	β_3	$\Delta\alpha_1$	$\Delta\beta_1$	$\Delta\alpha_2$	$\Delta\beta_2$	$\Delta\alpha_3$	$\Delta\beta_3$	x_i	y_i	z_i	η_i	$\epsilon_{i,1}$	$\epsilon_{i,2}$	$\epsilon_{i,3}$

is the sought-after position, where $\boldsymbol{t}_i^{(j)} = \left[x_i^{(j)}, y_i^{(j)}, z_i^{(j)}\right]^T$ corresponds to the j^{th} measurement of the i^{th} position p_i.

F. Acquired Dataset

Finally, we gathered $|\mathcal{S}| = 100\,000$ measured data with the CTCR prototype and methods introduced in this section. The dataset has been collected in eight sequences. Each sequence encompasses $12\,500$ dataset points. A dataset point consists all $6\,\text{dof}$ of Cartesian space $SE(3)$ described by the quaternion/vector-pairs for each sensor pose, all $6\,\text{dof}$ of joint space $\mathcal{Q}$, and the difference in joint space configuration to the previous configuration. The annotation information is provided in Table III and illustrated in Fig. 1. Note that only the tip orientation is indicated in Fig. 1. There are 33 columns and $100\,000$ rows in the file. From the first to the last column, the values for α_1, β_1, α_2, β_2, α_3, β_3, $\Delta\alpha_1$, $\Delta\beta_1$, $\Delta\alpha_2$, $\Delta\beta_2$, $\Delta\alpha_3$, $\Delta\beta_3$, x_0, y_0, z_0, η_0, $\epsilon_{0,1}$, $\epsilon_{0,2}$, $\epsilon_{0,3}$, x_1, y_1, z_1, η_1, $\epsilon_{1,1}$, $\epsilon_{1,2}$, $\epsilon_{1,3}$, x_2, y_2, z_2, η_2, $\epsilon_{2,1}$, $\epsilon_{2,2}$, $\epsilon_{2,3}$, x_3, y_3, z_3, η_3, $\epsilon_{3,1}$, $\epsilon_{3,2}$, and $\epsilon_{3,3}$ are stored. Note that the tip orientation $[\eta_3, \varepsilon_{3,1}, \varepsilon_{3,2}, \varepsilon_{3,3}]^\top$ is $[\eta, \varepsilon_1, \varepsilon_2, \varepsilon_3]^\top$ in (6) for $\boldsymbol{x}$ and (7) for $\boldsymbol{x}_{\text{pose}}$. Also note that the difference $\Delta\boldsymbol{q}$ between the current configuration $\boldsymbol{q}$ and the previous configuration is provided too, for instance $\Delta\alpha_k = \alpha_k - \alpha_{k-1}$ for the k^{th} sample and $\Delta\alpha_1 = \alpha_1$ for the 1^{st} sample. Position and orientation are expressed w.r.t. the frame of the electromagnetic tracking system located in the center of the field generator. The first pose with $i = 0$ of each row can be used to transform all poses into the base frame of the CTCR.

Note that while we did not observe any 'snapping' phenomena during the execution time, it is likely that the dataset captures a few 'snapping' occurrences due to two reasons. First, the acquisition spanned over several days – the CTCR prototype ran for nearly $23\,\text{h}$ for each sequence. Second, the path in joint space is optimized according to (17), which shortens the path without considering the required energy. Furthermore, other physical phenomena such as effects from high friction are most likely captured as well. We are confident that these physical phenomena can be extracted by investigating the differences in the joint space, i.e. $\Delta\alpha_i$ and $\Delta\beta_i$, and the corresponding pose differences.

IV. LEARNING THE FORWARD KINEMATICS

In the following, we present implementation details and the results of learning the kinematics. We utilize shallow neural networks and the presented CRL-Dataset-CTCR-Pose.

A. Neural Networks and Implementation Details

To learn the kinematics, fully-connected feedforward networks (FFN) are implemented in PyTorch. For the first and hidden layers, ReLU activation functions are used. Linear activation functions are used for the output layer. The number of activation functions in the hidden layer is 200, whereas the total number of weights and biases, denoted by $\mathcal{C}$, as well as the input and output definitions depend on the chosen representations. While (5) has a constant number of entries, i.e. nine, the shape representation depends on the used simplification applied to (6). The number of entries for $\boldsymbol{x}$, $\boldsymbol{x}_{\text{pose}}$, $\boldsymbol{x}_{\text{position}}$, and $\boldsymbol{x}_{\text{three-points}}$ is thirteen, seven, three, and nine, respectively.

The weights of the layer to which the ReLU activation functions are applied are initialized by He-initialization [17] also known as Kaiming-initialization. The remaining weights and all biases are initialized with $\mathcal{U}[0; 1]$.

To optimize the trainable parameters, the Adam optimizer [18] with mini-batch size of $N_{\text{bs}} = 128$, learning rate of 0.0005, and mean-squared error loss is used for 1000 epochs. Note that training on a GeForce RTX 2080 Ti has no time advantage compared to CPU, when using our dataset. Therefore, all FFNs are solely trained on a computer with Xeon $3.60\,\text{GHz} \times 8$ running on a 64-bit Linux operating system. For the FK, the input layer considers a joint description (5) and the output layer learns one of the selected shape representations, i.e. (6), (7), (8) or (9). The units depend on the chosen representations and are millimeter, rad, or one (unit-less).

The dataset $|\mathcal{S}| = 100\,000$ is randomly divided in training set $|\mathcal{S}_{\text{tra}}| = 80\,000$, validation set $|\mathcal{S}_{\text{val}}| = 10\,000$, and test set $|\mathcal{S}_{\text{test}}| = 10\,000$. Note that no exploration is performed and, therefore, $|\mathcal{S}_{\text{val}}|$ and $|\mathcal{S}_{\text{test}}|$ may not be representative. Further, neither the position values nor the orientation values are normalized.

B. Results and Discussion

The influence of different shape representations on the errors e_t and e_θ during the training process is shown in Fig. 2. The errors in tip position, i.e. e_t, are in sub-millimeter range between $0.69\,\text{mm}$ and $0.75\,\text{mm}$ corresponding to a relative error of $0.3\,\%$ and $0.4\,\%$, respectively.

Comparing the results for $\boldsymbol{x}$ and $\boldsymbol{x}_{\text{pose}}$, the errors for the tip pose are similar. However, the convergence of e_θ for $\boldsymbol{x}$ is a bit faster although noisier after 100 epochs, see Fig. 2. The evaluated e_t and e_θ are $0.7\,\text{mm}$ and $6.68°$, and $0.74\,\text{mm}$

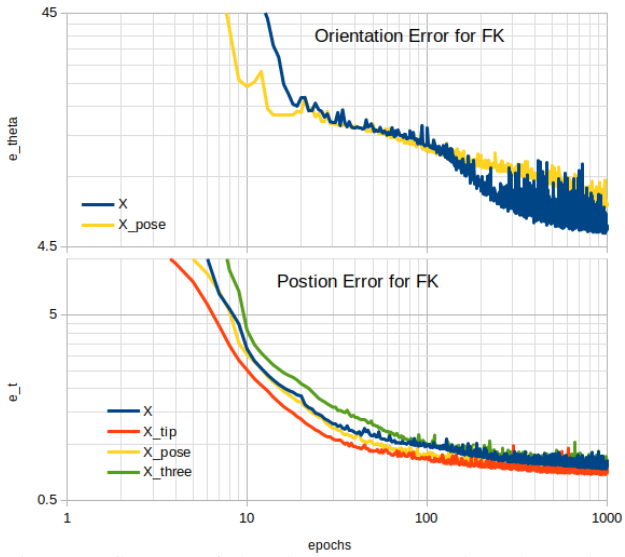

Fig. 2: Influence of the shape representation shown by the development of approximation error on the validation set over the epochs.

and $6.49°$, respectively. Figure 2 shows a steady decline of e_θ indicating higher accuracy can be achieved with more epochs and better hyper-parameters.

Albeit marginal, the best FK accuracy is achieved using $\boldsymbol{x}$, which considers both CTCR positional information along the backbone and tip orientation. It is expected as more geometric information from the measurements is utilized.

A comparison with other approaches is listed in Table IV. Consistent with [6], [7] regarding the tip error, our method outperforms the physics-based method based on Cosserat rod theory being the Goldstandard in our research community. Notably, there is a significant improvement in the position error over other learning-based methods. However, no improvement in the orientation error is achieved compared to [6], [7]. This might be due to the larger test set and different used datasets. Note that the dataset in [6] does not use the sampling method (16) and, therefore, the distributions of the samples are different.

V. LESSONS LEARNED AND OPEN CHALLENGES

In this section, we share our insights on learning the CTCR kinematics and discuss open challenges as guidelines to the community.

A. Representation

The key to a successful approximation of the CTCR kinematics is the choice of appropriate representations. In our previous work [6], the trigonometric joint representation has been introduced. Unlike e_t and e_θ using non-transformed α_i, the approximation errors of FFN utilizing transformed α_i do not saturate achieving a four to five times smaller e_t and e_θ after a short training time. We expanded on this approach in [7], where β_i are transformed by utilizing (3). This simple and effective transformation is capable of decorrelating β_i and leads to a further reduction of errors, while accelerating the convergence. For establishing the benchmark presented in this paper, we extend (3) to (4) such that all entries of (5) are linearly independent, orthogonal, and lie in the interval $[-1, 1]$. Therefore, we hypothesize that the proposed joint representation (5) is a well-suited feature description for machine learning applications.

Regarding the task space representation, we found that advantageous representations are mandatory even for low-dimensional machine learning applications [7]. Using quaternions leads to higher accuracy and faster convergence compared to all different sets of Euler angles. Therefore, we recommend to use the singularity-free quaternion/vector-pairs for $SE(3)$ representations, which we follow for data gathering and representation for the CRL-Dataset-CTCR-Pose as well as for the input of the FFN for the benchmark.

While a physics-based forward model outputs the full shape of a CTCR in task space, usually as a spatial curve $\boldsymbol{g}(s) \in SE(3)$ discretized by arclength, learning-based approaches have been focussed on tip position or tip pose mostly. Information about the continuum robot shape is yet relevant information for most applications. Our proposed representation (6) denoted by $\boldsymbol{x}$ is an expansion of the tip pose used in [7] as we now include positions along the shape, namely the tip pose of each tube constituting the CTCR.

The sum of polynomial basis functions has been proposed as an alternative representation for the shape of a CTCR [8]. The coefficients of these polynomial basis function were then learned. Exploring the merit of different curve representations in task space provides a promising avenue for future research.

B. Sampling Strategy

To sample joint configurations, we use (14) and (16). While sampling the revolute joint space of α_i is common practice, sampling the translational joint space of β_i via (16) is novel. The distribution of β_i looks different to the distribution shown in [20] and, therefore, also to the distribution in the dataset used in [6]. Figure 3 shows the distributions of each joint as found in the presented dataset. We hypothesise that our improved sampling strategy caused the improvements in e_t over prior results, see Table IV. Investigating the impact of the distribution on the performance of learning-based models is another promising avenue for future work. A related open challenge is to find a suitable distribution to maximise exploration in the task space.

C. Architecture and Loss Function

To learn the FK of a CTCR using the proposed representations, we implemented a ”vanilla” benchmark as described in Section IV. This shallow FNN outperforms the physics-based Goldstandard as indicated in Table IV and we argue that it is sufficient for FK learning.

We refrain from reporting on our attempt to learn the IK using the same ”vanilla” approach as it does not yield satisfactory results. In fact, only a few authors [5], [6], [9] attempted to address learning-based IK thus far. In [6], we use a thirteen times higher $\mathcal{C}$ to learn the IK, while, in [9],

TABLE IV: Baseline and benchmark for the forward kinematics of three-tube concentric tube continuum robots.

		accuracy[5]			evaluation[6]			
	Taskspace Representation	e_t in mm	e_t in %	e_θ in °	sim	real	$\mathcal{C}$	Reference
Physics-based								
	discrete spatial curve	2.91	1.5 to 3	–	–	✓	–	Rucker et al. [3]
	discrete spatial curve	4.2 ± 2.0	(2.1)	8.6 ± 4.1[7]	–	✓	–	Dupont et al. [19]
Benchmark								
	$\boldsymbol{x}_{\text{position}}$	0.69 ± 1.15	0.3	–	–	✓	2603	our
	$\boldsymbol{x}_{\text{three-points}}$	0.75 ± 1.15	0.4	–	–	✓	3809	our
	$\boldsymbol{x}_{\text{pose}}$	0.70 ± 1.15	0.3	6.68 ± 3.98	–	✓	3407	our
	$\boldsymbol{x}$	0.74 ± 1.15	0.4	6.49 ± 3.27	–	✓	4613	our
Prior work								
	tip pose	0.2	–	(0.11)	✓	–	–	Bergeles et al. [5]
	tip pose via LWPR[8]	(0.98[9])	–	1.11	–	✓	–	Fagogenis et al. [12]
	tip pose	2.23 ± 0.25	1	1.04 ± 0.08	–	✓	1707	Grassmann et al. [6]
	tip pose	1.6	0.79	1.4	–	✓	≈ 1300	Grassmann & Burgner-Kahrs [7]
	tip pose	7.6	–	9.0	✓	–	–	Grassmann & Burgner-Kahrs [7]
	polynomial basis functions[10]	3.3	(1.34)	–	–	✓	–	Kuntz et al. [8]
	tip position	0.5[9]	(0.33)	–	✓	–	–	Iyengar et al. [10]
	tip position	1.37[9]	(0.32)	–	✓	–	–	Iyengar & Stoyanov [11]

[5] Based on stated values in the reference, accuracy in parentheses have been applied afterward. For simplicity, the relative tip position error is determined by e_t divided by the maximum length of the CTCR manipulator.
[6] *sim* and *real* stand for evaluation in simulation and with a physical prototype, respectively.
[7] roll angle error
[8] tip pose computed via locally weighted projection regression (LWPR)
[9] path tracking error
[10] FFN outputs coefficient of the polynomial basis functions

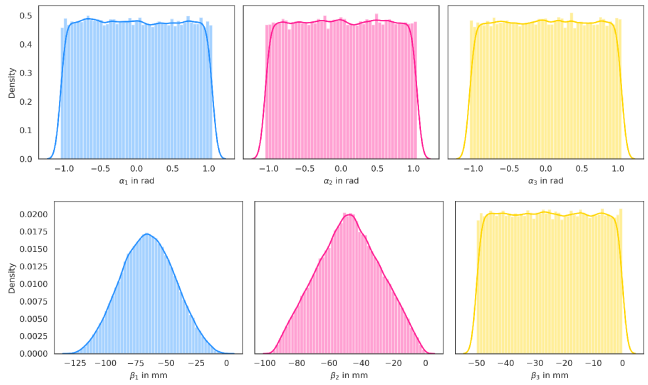

Fig. 3: Distribution of α_i and β_i in the dataset.

an FFN with $\mathcal{C} > 600\,000$ is used, cf. Table IV. To date, it is unclear which architecture and loss function lead to good approximations of the IK. Hence, learning an accurate and feasible IK remains an open challenge.

Investigating neural networks that preserve $SE(3)$ properties and loss functions customized to joint representation (5) are interesting future research directions. Ultimately, one may develop a CTCR specific FFN architecture derived on first principles based on the physics-based model [3], [19], which mimics the approaches in [21], [22], [23].

D. Synthesizing Learning- and Physics-based Approaches

The direct learning of CTCR kinematics by using measured data from a robot prototype can achieve higher accuracy than state-of-the-art physics-models [19], [3] as shown in [6], see Table IV. A learning-based model can capture non-modeled effects and physical phenomena. Therefore, neither calibration of a physics-based model nor an accurate physics-based model have to be applied to achieve computationally inexpensive and accurate FK. However, transferring and extracting the captured information remains unsolved.

Physics-based modelling can benefit from utilizing the insights gained in learning-based approaches. For instance, in conjunction with a physics-based model, the dataset can be used to identify and learn physical phenomena, such as friction and hysteresis. Those physical phenomena are commonly not considered in physics-based models. Learning these missing parts of a physics-based model is an open research direction. Combining the efforts of the continuum robotics community and machine learning practitioners could ultimately lead to the discovery of unknown problem structures and derivation of novel knowledge related to learning-based as well as physics-based approaches for continuum and soft robots.

E. Dataset and Benchmark

All results depend on the utilized dataset, which is different across the literature. Therefore, a direct comparison of our results with other approaches as listed in Table IV is not reasonable as the hardware and gathered datasets are different. This is amplified by a large and growing number of different existing prototypes as shown in [13], which imposes a barrier for comparing results.

To allow for reproducible research results in continuum robotics, we provide a publicly available dataset for the kinematics of CTCR. Our results on the CTCR FK using this dataset in Table IV can serve as a benchmark to replicate and compare future approaches.

Prior work suggests that optimal [24] or equidistant [9] sensor placements may be advantageous to gather relevant shape information. Yet, such sensor placement is challenging to realize on a physical CTCR. For the dataset, we attached electromagnetic tracking coils to each tube, which allowed for straightforward integration. In contrast, we are not aware of a sensing mean which allows for constant, equidistant measurements along the length, which is changing as the tubes are translated w.r.t. one another. For example, fiducial markers along the shape as used in [25] are not suitable because they are fixed to the respective tube and can disappear inside the outer tubes. Consequently, there exist simulation results that cannot be transferred to a physical prototype as of today for the lack of data [9]. Note that one can argue that pixels are an equidistant discretization of an image and, therefore, image-based approaches [8], [9] provide equidistant measurements along the length. However, the coverage of a CTCR in the image space depends on the CTCR's configuration and the camera setup, e.g. camera position for each view. Although promising, there are open questions regarding the use of learning-based approaches utilizing multi-view images from CTCR. Furthermore, it should be noted that fully embedded sensors such as fiber Bragg grating sensors can only achieve equidistant discretization if the overall length of the CTCR manipulator is constant, which limits the length variation. Generating datasets for different CTCR in terms of tube parameters as well as datasets with additional sensor information on the shape in task space remains a challenge.

VI. CONCLUSIONS

By providing the first public dataset for a three-tube CTCR, we intend to democratize research on learning-based and physics-based modelling of the kinematics. The continuum robotics community as well as machine learning practitioners can utilize the dataset for replication, benchmarking, and learning physical phenomena, such as fiction and hysteresis. While future work evolves around the exploitation of the CRL-Dataset-CTCR-Pose dataset, the long-term vision is to leverage gained findings and insights to amplify the improvement in physics-based modelling.

ACKNOWLEDGMENT

The authors thank Thien-Dang Nguyen for designing and building the robot prototype.

REFERENCES

[1] P. Sears and P. Dupont, "A steerable needle technology using curved concentric tubes," in *IEEE/RSJ international conference on intelligent robots and systems*, 2006, pp. 2850–2856.

[2] R. J. Webster, A. M. Okamura, and N. J. Cowan, "Toward active cannulas: Miniature snake-like surgical robots," in *IEEE/RSJ international conference on intelligent robots and systems*, 2006, pp. 2857–2863.

[3] D. C. Rucker, B. A. Jones, and R. J. Webster III, "A geometrically exact model for externally loaded concentric-tube continuum robots," *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 769–780, 2010.

[4] Z. Mitros, S. H. Sadati, R. Henry, L. Da Cruz, and C. Bergeles, "From theoretical work to clinical translation: Progress in concentric tube robots," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, 2021.

[5] C. Bergeles, F. Lin, and G. Yang, "Concentric tube robot kinematics using neural networks," in *Hamlyn Symposium on Medical Robotics*, 2015, pp. 1–2.

[6] R. Grassmann, V. Modes, and J. Burgner-Kahrs, "Learning the forward and inverse kinematics of a 6-dof concentric tube continuum robot in SE(3)," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2018, pp. 5125–5132.

[7] R. M. Grassmann and J. Burgner-Kahrs, "On the merits of joint space and orientation representations in learning the forward kinematics in SE(3)," in *Robotics: Science and Systems*, 2019.

[8] A. Kuntz, A. Sethi, R. J. Webster, and R. Alterovitz, "Learning the complete shape of concentric tube robots," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 2, pp. 140–147, 2020.

[9] N. Liang, R. M. Grassmann, S. Lilge, and J. Burgner-Kahrs, "Learning-based inverse kinematics from shape as input for concentric tube continuum robots," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 1387–1393.

[10] K. Iyengar, G. Dwyer, and D. Stoyanov, "Investigating exploration for deep reinforcement learning of concentric tube robot control," *International Journal of Computer Assisted Radiology and Surgery*, vol. 15, pp. 1157–1165, 2020.

[11] K. Iyengar and D. Stoyanov, "Deep reinforcement learning for concentric tube robot control with a goal-based curriculum," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 1459–1465.

[12] G. Fagogenis, C. Bergeles, and P. E. Dupont, "Adaptive nonparametric kinematic modeling of concentric tube robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 4324–4329.

[13] R. M. Grassmann, S. Lilge, P. Le, and J. Burgner-Kahrs, "Ctcr prototype development: An obstacle in the research community?" in *Robotics: Science and Systems – Workshop on Robotics Retrospectives*, 2020.

[14] T. Lozano-Perez, "Spatial planning: A configuration space approach," *IEEE Transactions on Computers*, vol. 32, no. 2, pp. 108–120, 1983.

[15] J. Ha, G. Fagogenis, and P. Dupont, "Effect of path history on concentric tube robot model calibration," in *Hamlyn Symposium on Medical Robotics*, 2017, pp. 77–78.

[16] V. Modes and J. Burgner-Kahrs, "Calibration of concentric tube continuum robots: Automatic alignment of precurved elastic tubes," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 103–110, 2019.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[19] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler, "Design and control of concentric-tube robots," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 209–225, 2010.

[20] J. Burgner-Kahrs, H. B. Gilbert, J. Granna, P. J. Swaney, and R. J. Webster, "Workspace characterization for concentric tube continuum robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1269–1275.

[21] L. Nguyen, R. Patel, and K. Khorasani, "Neural network architectures for the forward kinematics problem in robotics," in *international joint conference on neural networks*, 1990, pp. 393–399.

[22] Z. Geng and L. Haynes, "Neural network solution for the forward kinematics problem of a stewart platform," in *IEEE International Conference on Robotics and Automation*, 1991, pp. 2650–2651.

[23] F. D. Ledezma and S. Haddadin, "First-order-principles-based constructive network topologies: An application to robot inverse dynamics," in *IEEE-RAS 17th International Conference on Humanoid Robotics*, 2017, pp. 438–445.

[24] A. W. Mahoney, T. L. Bruns, P. J. Swaney, and R. J. Webster, "On the inseparable nature of sensor selection, sensor placement, and state estimation for continuum robots or "where to put your sensors and how to use them"," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 4472–4478.

[25] H. Donat, S. Lilge, J. Burgner-Kahrs, and J. J. Steil, "Estimating tip contact forces for concentric tube continuum robots based on backbone deflection," *IEEE Transactions on Medical Robotics and Bionics*, vol. 2, no. 4, pp. 619–630, 2020.