Learning the Forward and Inverse Kinematics of a 6-DOF Concentric Tube Continuum Robot in $SE(3)$

Reinhard Grassmann, Vincent Modes, and Jessica Burgner-Kahrs, *Senior Member, IEEE*

***Abstract*— Recent physics-based models of concentric tube continuum robots are able to describe pose of the tip, given the preformed translation and rotation in joint space of the robot. However, such model-based approaches are associated with high computational load and highly non-linear modeling effort. A data-driven approach for computationally fast estimation of the kinematics without requiring the knowledge and the uncertainties in the physics-based model would be an asset. This paper introduces an approach to solve the forward kinematics as well as the inverse kinematics of concentric tube continuum robots with 6-DOF in three dimensional space $SE(3)$. Two artificial neural networks with ReLU (rectified linear unit) activation functions are designed in order to approximate the respective kinematics. Measured data from a robot prototype are used in order to train, validate, and test the proposed approach. We introduce a representation of the rotatory joints by trigonometric functions that improves the accuracy of the approximation. The results with experimental measurements show higher accuracy for the forward kinematics compared to the state of the art mechanics modeling. The tip error is less then 2.3 mm w.r.t. position (1 % of total robot length) and $1.1\,^\circ$ w.r.t. orientation. The single artificial neural network for the inverse kinematics approximation achieves a translation and rotation actuator error of 4.0 mm and $8.3\,^\circ$, respectively.**

I. INTRODUCTION

Concentric tube continuum robots, which are composed of multiple concentric, precurved super-elastic tubes, are inherently compliant and flexible, which enables them to be used in complex and sensitive environments. As stated in [4], these are ideal structures for use for robot assisted minimally-invasive surgery and, furthermore, they open up new applications. However, despite the simple mechanism of rotating and translating the component tubes, which can be seen in Fig. 1, modeling of the kinematics is characterized by highly non-linear behavior due to the elastic interactions between the concentric tubes. Achieving fast and accurate kinematics is an important aspect for online motion planning and control, which enables the first step to industrial applications.

a) State of the Art: The common modeling approach is based on the theory of special Cosserat rods for each tube, which undergoes bending and torsion [6], [19]. This leads to no analytical solutions for concentric tube continuum robots consisting of more than two tubes, or for precurvature that varies with total robot length. Thus, model based forward kinematics are typically solved numerically. Additional factors like tube tolerances and friction have been investigated, but not effectively integrated yet [9]. For example, in [17] frictional torques are included with the cost of solving a root finding problem at each integration step with respect to (w.r.t.) the total robot length. Therefore, additional factors are commonly not considered due to even higher computational load and modeling effort.

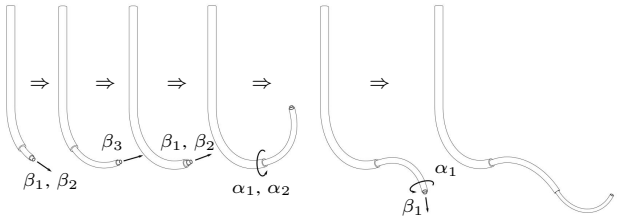

Fig. 1. Basic principle of a concentric tube continuum robot with three tubes. α_i and β_i are rotation and translation of a given tube i, respectively.

Regarding inverse kinematics, no analytical solutions have yet been found and proposed except for exterior simple systems like a continuum robot with two tubes. Therefore, common approaches from robotics are applied, e.g. numerical root finding methods [6] or differential-inverse-kinematics strategies [2]. Thus, computing the inverse kinematics is typically computationally slow. Furthermore, the convergence is not guaranteed.

In summary, a data-driven approximation of the kinematics without requiring an accurate physical model would be beneficial. Such approach would be more accurate in real scenarios than model-based methods. In [10] the inverse kinematics of a tendon-driven continuum robot is approximated with an artificial neural network and compared with a Jacobian-based approach. The latter provides poor results in terms of accuracy and computational time. In [22] they proposed three data-driven approaches to learn an accurate inverse kinematics model for a flexible surgical manipulator. The inverse kinematics and the forward kinematics of a pneumatically actuated continuum robot are learned separately in [18]. The previous approaches used measured data and considered the position of the end-effector. Note that concentric tube robots are structurally different such that existing data-based methods cannot be directly applied.

Regarding concentric tube robots, initial works [1] used simulated data to train artificial neural networks which approximated the forward and inverse kinematics. In order to estimate the tube rotation, each tube rotation is split into four quadrants, i.e. $[0, \pi/2)$, $[\pi/2, \pi)$, $[\pi, 3\pi/2)$, and $[3\pi/2, 2\pi)$, which leads to a complex output of the artificial neural network including a selection mechanism. Moreover, simplifications such as variable-curvature section, i.e. two tubes are combined to form one tube, and reduced pose, i.e.

All authors are with Laboratory for Continuum Robotics, Faculty of Mechanical Engineering, Leibniz Universität Hannover, 30167 Hanover, Germany grassmann@lkr.uni-hannover.de

 DOI: 10.1109/IROS.2018.8594451

consisting of three positions and two orientations, have been made. To the best of our knowledge, the initial results in [1] are the only published results on concentric tube continuum robots so far.

b) Contributions: In the present work, we introduce a complete approximation approach of the forward and inverse kinematics for concentric tube continuum robots based on neural networks. We simultaneously consider the tip position and orientation for the forward kinematics and, furthermore, approximate the inverse kinematics with a single artificial neural network. For the first time, the approximations with exhausive experimental data using a concentric tubes continuum robot prototype with three tubes (6-DOF) have been investigated.

In addition, we introduce a new trigonometric joint description, which leads to a simple, yet effective feature representation for learning purposes and positevely affects the approximation results. Our approach shows higher accuracy in comparison to model-based approaches. As a minor contribution, we propose a novel adjustment device based on parallax compensation for the zero point adjustment of the rotatory actuators.

II. METHODS

In this section, we motivate the need for a new trigonometric joint description as well as the need for quaternions. Moreover, we state the approximation errors for the forward and inverse kinematics. Finally, a brief description of the used artificial neural network is given.

A. Joint Space

For the following, it is important to note that the joint space of m revolute joints can be described as an m-torus [20]. In this paper, we consider concentric tube continuum robots composed of three tubes (Fig. 2). Without loss of generality, all statements can be generalized for concentric tube continuum robots with arbitrary number of tubes.

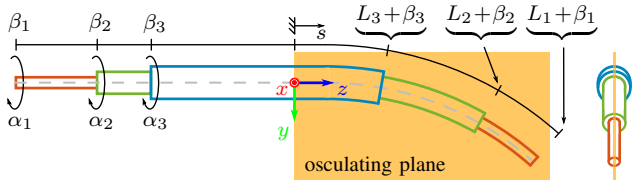

Fig. 2. Concentric tube continuum robot with three tubes. The tubes can be rotated (α_i) and translated (β_i) w.r.t. each other. The arc length s describes the robots shape with its respective tube length L_i. The osculating planes of the tubes correspond to the rotational zero position in the yz-plane.

The joint space $\mathcal{Q} = \alpha_1 \times \beta_1 \times \alpha_2 \times \beta_2 \times \alpha_3 \times \beta_3$ of a 6-DOF concentric tube continuum robot can be subdivided in two spaces, in particular $\mathcal{A}$ and $\mathcal{B}$, where $\mathcal{Q} = \mathcal{A} \times \mathcal{B}$ is valid. The space $\mathcal{A} = \alpha_1 \times \alpha_2 \times \alpha_3$ represents a 3-torus $\mathbb{T}^3 = \mathbb{S}^1 \times \mathbb{S}^1 \times \mathbb{S}^1$, where $\mathbb{S}^1$ is a 1-sphere. Topologically, $\mathbb{S}^1$ can be defined in terms of an embedding in two-dimensional Euclidean space, which leads to a circle. The translational part characterized by $\mathcal{B} = \beta_1 \times \beta_2 \times \beta_3$ can be described as a parallelepiped. Its conditions are

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

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

where L_i is the overall tube length of the i^{th} tube. The inequations (1) and (2) are visualized in Fig. 3 for an example with two tubes.

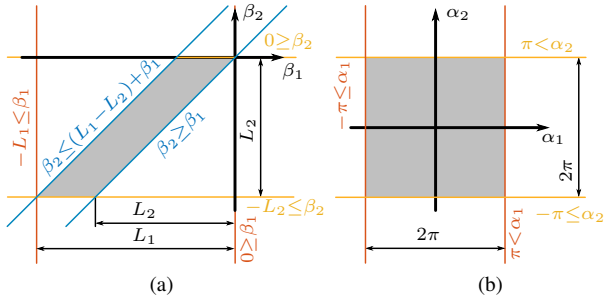

Fig. 3. Joint space $\mathcal{Q}$ of a 4-DOF concentric tube continuum robot with two tubes subdivided in $\mathcal{A}$ and $\mathcal{B}$. a) Translational joint space $\mathcal{B} = \beta_1 \times \beta_2 \subset \mathbb{R}^2$ with its restrictions. It forms a parallelogram. b) Rotational joint space $\mathcal{A} = \alpha_1 \times \alpha_2 = \mathbb{S}^1 \times \mathbb{S}^1 = \mathbb{T}^2$ forms a torus in $\mathbb{R}^3$ due to the fact that the opposite edges of the square are pasted together.

B. Trigonometric Joint Representation

Trigonometric functions were applied on the joint angle of a planar robot in order to constraint them [14]. We adapt this idea to concentric tube continuum robots and define the cylindrical Form γ_i as

$$\gamma_i = \{\gamma_{1,i},\ \gamma_{2,i},\ \gamma_{3,i}\} = \{\cos(\alpha_i), \sin(\alpha_i), \beta_i\}, \tag{3}$$

which describes the i^{th} tube as a triplet. The rotary joint α_i can be obtained by the $\operatorname{atan2}$ function, which gives the unambiguous correct result in the respective quadrants. Therefore, the joint α_i can be recovered by

$$\alpha_i = \operatorname{atan2}\{\gamma_{2,i},\ \gamma_{1,i}\}. \tag{4}$$

In Fig. 4, a geometrical relationship is established.

Note that $\alpha_i \in \mathbb{S}^1$ and that $\mathbb{S}^1$ can be described effectively in $\mathbb{R}^2$. Therefore, all entries of γ_i are elements of $\mathbb{R}$. This is an important fact, because an artificial neural network with real activation function φ expects real input values. Moreover, due to the transformation by means of trigonometric function and the use of (4) an artificial neural network does not have to approximate the $\operatorname{atan2}$ function, which is a discontinuous function. Therefore, it can be concluded that γ_i provides a suitable tube representation for an artificial neural network.

C. Quaternions

A unit quaternion is a hypercomplex number denoted by $\xi = \eta + \epsilon_1 \imath + \epsilon_2 \jmath + \epsilon_3 k$ with the property $\eta^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 = 1$. The quaternionic units $\imath, \jmath$, and k follow Hamilton's rule $\imath^2 = \jmath^2 = k^2 = -1$ and $\imath\jmath k = -1$. A quaternionic conjugate of ξ is defined as $\bar{\xi} = \eta - \epsilon_1 \imath - \epsilon_2 \jmath - \epsilon_3 k$. The associative

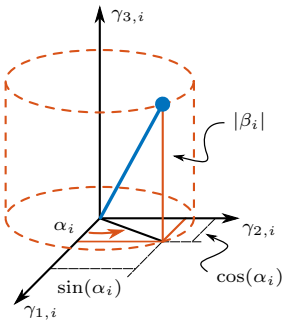

Fig. 4. Trigonometric joint representation with of α_i and β_i in the cylindrical coordinate system, where the radius is one, the height is equal to β_i, and the azimuth is α_i. The joint α_i can be recovered by the atan2 function.

and non-commutative product of two quaternions ξ and ξ' expands to

$$\begin{aligned}\xi\xi' &= (\,\eta + \epsilon_1\imath + \epsilon_2\jmath + \epsilon_3 k\,)\,(\,\eta' + \epsilon_1'\imath + \epsilon_2'\jmath + \epsilon_3' k\,) \\ &= (\,\eta\eta' - \epsilon_1\epsilon_1' - \epsilon_2\epsilon_2' - \epsilon_3\epsilon_3'\,) \\ &\quad + (\,\eta\epsilon_1' + \eta'\epsilon_1 + \epsilon_2\epsilon_3' - \epsilon_2'\epsilon_3\,)\,\imath \\ &\quad + (\,\eta\epsilon_2' + \eta'\epsilon_2 + \epsilon_3\epsilon_1' - \epsilon_3'\epsilon_1\,)\,\jmath \\ &\quad + (\,\eta\epsilon_3' + \eta'\epsilon_3 + \epsilon_1\epsilon_2' - \epsilon_1'\epsilon_2\,)\,k. \end{aligned} \tag{5}$$

Due to Hamilton's rule, the quaternionic units are similar to the cross product of two unit Cartesian vectors. Therefore, it is common to describe a quaternion ξ as follows

$$\xi = \cos(\vartheta) + (n_x\imath + n_y\jmath + n_z k)\sin(\vartheta), \tag{6}$$

where $\boldsymbol{n} = (n_x, n_y, n_z)^T$ is a fixed axis and ϑ is the angle between two given orientations.

The main advantage of quaternions is that this representation of $SO(3)$ is singularity-free and global [21]. The disadvantage is the so-called antipodal property, which means the double coverage of the $SO(3)$, i.e. ξ and $-\xi$ represent the same orientation. Thus, multiplying a quaternion ξ by -1 does not change the orientation it represents. However, by forcing the value of η to be positive, only the half of the four-dimensional unit-sphere will be used and therefore, we can overcome the disadvantage of the antipodal property. This can be achieved by

$$\xi' = \operatorname{sign}(\eta)\xi, \tag{7}$$

where $\operatorname{sign}(\eta)$ gives the sign of η and $\operatorname{sign}(0)$ is $+1$ by definition.

The combination of a translational displacement $\boldsymbol{t}$ and a quaternion ξ will give the most efficient alternative among the point transformation formalism [8]. It is the so-called quaternion/vector pair $[\xi, \boldsymbol{t}]$. By using (7), we achieve a representation, which is singularity-free and globally defined as well as an bijective mapping. Those properties make the quaternion/vector pair $[\xi, \boldsymbol{t}]$ suitable for use with artificial neural networks.

D. Approximation Error

Now, we describe the approximation error for the forward and inverse kinematics in their domain, i.e. Cartesian space and joint space, respectively. For the present paper, the hat is used to denote an approximated value.

a) Cartesian space: The error e_x is given by

$$e_x = \sqrt{\left(t_x - \hat{t}_x\right)^2 + \left(t_y - \hat{t}_y\right)^2 + \left(t_z - \hat{t}_z\right)^2}, \tag{8}$$

where t_i and $\hat{t}_i$ are the positions along the corresponding Cartesian axis. By utilizing a unit quaternion, the rotational error e_ϑ is considered as

$$e_\vartheta = \arccos\left(\eta\hat{\eta} + \epsilon_1\hat{\epsilon}_1 + \epsilon_2\hat{\epsilon}_2 + \epsilon_3\hat{\epsilon}_3\right) \in [0, \pi], \tag{9}$$

where the approximated unit quaternion $\hat{\xi} = \hat{\eta} + \hat{\epsilon}_1\imath + \hat{\epsilon}_2\jmath + \hat{\epsilon}_3 k$ must satisfy the property $\hat{\eta}^2 + \hat{\epsilon}_1^2 + \hat{\epsilon}_2^2 + \hat{\epsilon}_3^2 = 1$. Thus, a quaternion $\hat{\xi}$ must be normalized, if it is approximated by the artificial neural network.

b) Joint space: Recapping the facts in Sec. II-A, joint space $\mathcal{Q}$ can be divided into $\mathcal{A}$ and $\mathcal{B}$. We will take advantage of this approach, in order to describe the approximation errors e_β and e_α. The translational error e_β is straightforward. It can be described as the Euclidean distance in $\mathcal{B}$ and, consequently, can be defined as

$$e_\beta = \sqrt{\left(\beta_1 - \hat{\beta}_1\right)^2 + \left(\beta_2 - \hat{\beta}_2\right)^2 + \left(\beta_3 - \hat{\beta}_3\right)^2}. \tag{10}$$

Note that $\mathcal{B}$ with its conditions (1) and (2) forms a convex set (see also Fig. 3a for visualization). Under the assumption that the rotational error e_ϑ is small, the tangent space of $\mathcal{A}$ can be determined. To recap briefly, $\mathcal{A}$ is a 3-torus $\mathbb{T}^3$. Since every manifold can be locally approximated by a tangent space, we can apply the Euclidean distance in $\mathcal{A}$. Thus, the rotational error e_α is given by

$$e_\alpha = \sqrt{e_{\alpha,1}^2 + e_{\alpha,2}^2 + e_{\alpha,3}^2}, \tag{11}$$

where the difference $e_{\alpha,i} = \alpha_i - \hat{\alpha}_i$ for the i^{th} tube can be computed by applying (4). For (11) we utilize the identity

$$\begin{aligned}&\operatorname{atan2}(s_1, c_1) \pm \operatorname{atan2}(s_2, c_2) = \\ &\qquad \operatorname{atan2}(s_1c_2 \pm s_2c_1,\ c_1c_2 \mp s_1s_2), \end{aligned} \tag{12}$$

where s_i and c_i indicate the relation to sinus and cosine function, respectively. Consequently, $e_{\alpha,i}$ is determined by

$$e_{\alpha,i} = \operatorname{atan2}(\gamma_{2,i}\hat{\gamma}_{1,i} - \hat{\gamma}_{2,i}\gamma_{1,i}, \gamma_{1,i}\hat{\gamma}_{1,i} + \gamma_{2,i}\hat{\gamma}_{2,i}) \tag{13}$$

for the i^{th} tube.

E. Artificial Neural Networks

We utilize feedforward networks in order to approximate the kinematics. An architecture of a feedforward network is summarized in Fig. 5. Such a feedforward network can approximate a smooth function in a compact set [13], [5], [7].

Furthermore, we consider all degrees of freedom w.r.t. three-dimensional space $SE(3)$ and w.r.t. a concentric tube continuum robot with three tubes. Thus, no simplifications are assumed. The Cartesian space $SE(3)$ is described as a quaternion/vector pair $[\xi, \boldsymbol{t}]$, whereas the joint space $\mathcal{Q}$ is represented by means of γ_i.

Preliminary examinations in simulation show that ReLU (rectified linear unit) are computationally efficient and could produce smaller approximation errors compared to the common used tanh activation function. Furthermore, it is also shown that the inputs and outputs do not need to be scaled. Therefore, the ReLU activation function $\varphi(x) = \max(0, x)$ are used in the hidden layer. The number of artificial neurons and of hidden layers are determined in preliminary examinations in simulation. The HE-initialization [12] is applied in order to initialize their weights. As usual, linear activation functions are used for the output layer. Their weights and all biases are initialized with a uniform distribution.

For the optimization of the weights ω and biases b, we used Adam [15] with mini-batch size of $N_{\text{bs}} = 128$, which are randomly extracted from the training set $\mathcal{S}_{\text{tra}}$ in each epoch N_{ep}. During the preliminary examinations and during the training with measured data, we discovered that neither under- nor over-fitting occurred by using the Adam optimizer. Therefore, there is no need for a validation set, which can be used in order to detect bad generalization, i.e. under- or over-fitting. Moreover, Adam converges much faster than a vanilla gradient descent, which is experimentally observed. Its individual adaptive learning rates and the use of the first and second moments of the gradients may cited as a cause for the described observations.

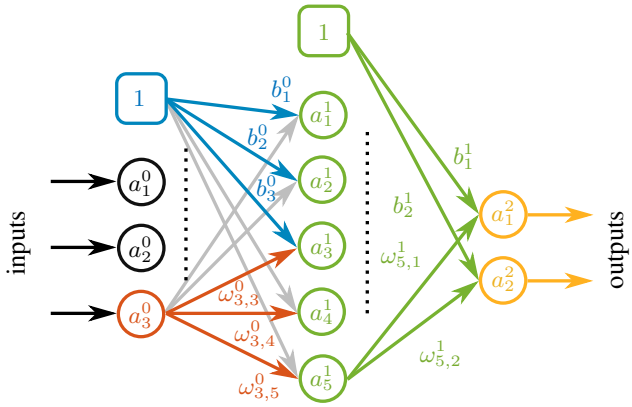

Fig. 5. Artificial neural networks as feedforward network. With $\omega_{n,m}^l$ we denote the weight for the connection between the m^{th} artificial neuron in the $(l-1)^{\text{th}}$ layer to the n^{th} neuron in the l^{th} layer, whereas b_n^l is used for the n^{th} bias in the l^{th} layer. Furthermore, the activation a_n^l denotes the output of the n^{th} artificial neuron in the l^{th} layer.

III. TESTBED AND DATA ACQUISITION

The purpose of the experimental setup is the acquisition of tip poses with all 6-DOF of $SE(3)$ in order to collect data for the training and test set. The constructed testbed is shown in Fig. 6. The geometrical parameters and mechanical properties of the used concentric tube continuum robot are listed in Table I.

A. Rotation Adjustment Device

As with conventional robots, in order to properly determine the kinematics, all joints must be adjusted by finding or defining the zero position for each joint. Consequently, all tubes and, therefore, α_i and β_i must be adjusted to ensure that all measured data is in the same reference. Note that the adjustment is not commonly defined for this type of robot. While adjusting β_i is straightforward, finding the zero position for α_i is more challenging. In order to adjust β_i, markers are applied such that the zero position of the respective carriage are indicated. An adjustment device is proposed and described in the following, which solves the adjustment for α_i.

TABLE I

TUBE PARAMETERS OF THE CONCENTRIC TUBE CONTINUUM ROBOT PROTOTYPE. THE INNERMOST, MIDDLE AND OUTER TUBES ARE REFERENCED FORM 1 TO 3, RESPECTIVELY.

Parameter			Set of tubes		
Term	Symbol	Unit	Tube 1	Tube 2	Tube 3
Length, overall	L	mm	370	305	170
Length, straight	L_{s}	mm	325	205	70
Curvature	κ_x	m^{-1}	15.8	9.27	4.37
Diameter, outer	D_{o}	mm	0.4	0.9	1.5
Diameter, inner	D_{i}	mm	0.3	0.7	1.2
Young's Modulus	E	GPa	50	50	50
Poisson's ratio	ν	1	0.3	0.3	0.3

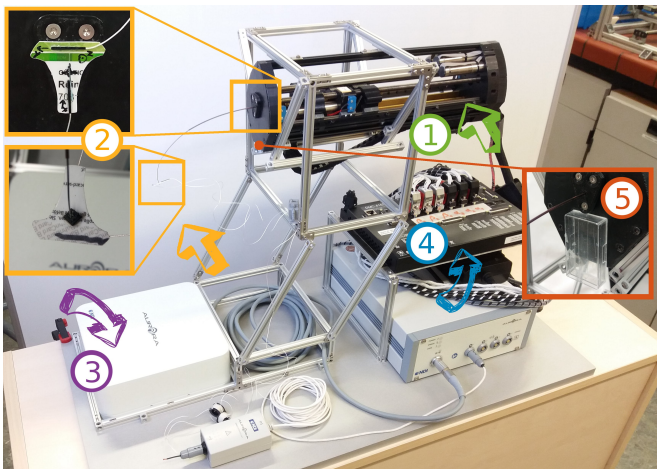

Fig. 6. Testbed with the concentric tube continuum robot (1), 6-DOF sensor consisting of two 5-DOF sensors, orthogonally mounted with respect to each other (2), electro magnetic tracking system (AURORA, Northern Digital Inc., ON, Canada) (3), motion controller (DCM4163, Galil Motion Control, CA, USA) (4), and rotation adjustment device prototype (5).

Our definition of the zero points is done by the elastic interaction between the tubes. For the sake of clarity, only the superposition of all tubes with stable equilibrium is considered and only tubes with curvature κ_x are treated.

Because there is exactly one osculating plane for each tube, the superposition of all three tubes results in an unique osculating plane, if all tubes are aligned (see Fig. 2). This requested osculating plane lies in the yz-plane of the concentric tube continuum robot's origin and defines the zero points of the rotational joints α_i (see Fig. 2). The underlying idea is that two unique parallel lines (see Fig. 7) always create one plane. Furthermore, all points of the robot backbone can be mapped to a straight line in the requested osculating plane, if all tubes are aligned. It follows that if all three lines lie in a common plane, then the requested osculating plane is

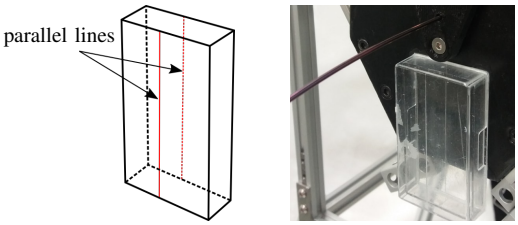

Fig. 7. Rotation adjustment device for the calibration of α_i DOFs: theoretical sketch (left), prototype (right).

TABLE II

COMPARE THE COMPLETE AND THE RESTRICTED JOINT SPACE. NOTE THAT THE OVERALL LENGTH OF EACH TUBE DOES NOT CORRESPOND TO THE MAXIMUM VALUE OF RESPECTIVE β.

	complete joint space				restricted joint space			
	α_i [°]		β_i [mm]		α_i [°]		β_i [mm]	
i	min	max	min	max	min	max	min	max
1	−180	180	−205	0	−60	60	−144	0
2	−180	180	−164	0	−60	60	−115	0
3	−180	180	−115	0	−60	60	−81	0

identical to the plane, which is spanned by the two unique parallel lines.

Based on the above observation and definition, an adjustment device is proposed. The prototype with the minimum requirements and properties is shown in Fig. 7. The prototype is made of a transparent plastic box. On opposite sides, lines were scratched, representing the mentioned parallel lines. Furthermore, the prototype is placed such that both lines intersect with the z-axis of the robot's base frame. Due to the transparent material of the prototype we can observe both lines simultaneously. Therefore, the mentioned yz-plane can be found by overlapping both parallel lines. Now, all α_i are changed until the backbone of the concentric tube continuum robot is in its yz-plane. In this case, the projection of the backbone onto the xy-plane is congruent with both scratched lines and both parallel lines, respectively. This approach also avoids the parallax error due to two unique parallel lines.

B. Random Sampling of the Joint Space

The samples of a given robot's joint space $\mathcal{Q}$ are generated from a uniform random distribution. Each sample $q = [\alpha_1, \beta_1, \alpha_2, \beta_2, \alpha_3, \beta_3] \in \mathcal{Q}$ consists of three rotational and three translational parameters. While the rotational parameters α_i of the robot configuration are independent, the translational parameters β_i interdependent, see (2) and (1). To limit the error of the gravitational force of the 6-DOF sensor attached to the robot tip and to increase the density of the data points, the joint space $\mathcal{Q}$ has been limited. Therefore, the translational part is restricted to $\beta_i \in [-0.7L_i, 0\,\mathrm{mm}]$ and the orientational part is restricted to $\alpha_i \in [-60\,°, 60\,°]$. Additionally, an offset L_m such that innermost tube always extends by $10\,\mathrm{mm}$ is added to condition (1) in order to prevent damage to the measuring sensor. Table II compares the complete and the restricted joint space.

C. Data Acquisition

In order to reduce the error due to the electro-magnetic tracking system, we measure $n = 5$ end-effector poses per sample. Consequently, an appropriate translation $\boldsymbol{t}$ and an appropriate orientation ξ must be determined.

a) Orientation: For orientation, a quaternion which yields the smallest quadratic error is searched. For this purpose, we utilize (9). Therefore, n equations of the form

$$1 = \eta\eta^{(i)} + \epsilon_1\epsilon_1^{(i)} + \epsilon_2\epsilon_2^{(i)} + \epsilon_3\epsilon_3^{(i)} \tag{14}$$

must be solved, where $\xi^{(i)} = \eta^{(i)} + \epsilon_1^{(i)}\imath + \epsilon_2^{(i)}\jmath + \epsilon_3^{(i)}k$ is the i^{th} quaternion from the i^{th} measurement and the quaternion $\xi = \eta + \epsilon_1\imath + \epsilon_2\jmath + \epsilon_3 k$ is the sought quaternion. By taking advantage of the Gaussian approach (least squares), the solution of the equation system

$$\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \eta^{(1)} & \epsilon_1^{(1)} & \epsilon_2^{(1)} & \epsilon_3^{(1)} \\ \eta^{(2)} & \epsilon_1^{(2)} & \epsilon_2^{(2)} & \epsilon_3^{(2)} \\ \vdots & \vdots & \vdots & \vdots \\ \eta^{(i)} & \epsilon_1^{(i)} & \epsilon_2^{(i)} & \epsilon_3^{(i)} \\ \vdots & \vdots & \vdots & \vdots \\ \eta^{(n-1)} & \epsilon_1^{(n-1)} & \epsilon_2^{(n-1)} & \epsilon_3^{(n-1)} \\ \eta^{(n)} & \epsilon_1^{(n)} & \epsilon_2^{(n)} & \epsilon_3^{(n)} \end{bmatrix} \begin{bmatrix} \eta \\ \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \end{bmatrix}, \tag{15}$$

is given by

$$[\eta \quad \epsilon_1 \quad \epsilon_2 \quad \epsilon_3]^T = \left(\boldsymbol{Q}_m^T\boldsymbol{Q}_m\right)\boldsymbol{Q}_m^T\mathbf{1}, \tag{16}$$

where $\boldsymbol{Q}_m \in \mathbb{R}^{n\times 4}$ is the so-called design matrix of $\xi^{(i)}$ and $\mathbf{1} \in \mathbb{R}^{n\times 1}$ is left side of (15). Note that (16) can be also derived from log-likelihood. Subsequently, the orientation is normalized with its Euclidean length. For the sake of clarity, (16) requires at least four measurements.

b) Position: A similar approach for the tip position can be chosen, i.e. $\mathbf{0} = \boldsymbol{t} - \boldsymbol{t}^{(i)}$. For n given positions, the mean position is determined by

$$\boldsymbol{t} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{t}^{(i)}, \tag{17}$$

where $\boldsymbol{t}^{(i)}$ corresponds to the i^{th} measurement.

IV. FORWARD KINEMATICS

In the following, the approximation of the forward kinematics is evaluated. For this purpose, we gathered $64{,}000$ pose samples with the prototype described in Section III. From this set $60{,}000$ samples are used to train a set of shallow neural networks while the remaining 4000 samples form our test data set $\mathcal{S}_{\text{test}}$. The joint description (3) is used as input layer. Table III lists the applied learning and architecture parameters. Because of the randomized HE-initialization [12] and the randomized selection of the mini-batch from the training set, networks with the same architecture give slightly different results. To compensate for this effect we trained 10 neural networks and averaged the results.

To investigate the influence of different joint representation on the kinematic approximation, we trained an another set of

TABLE III

FEEDFORWARD NETWORKS FOR THE APPROXIMATION OF FORWARD KINEMATICS AND INVERSE KINEMATICS. LISTED ARE THE TRAINING PARAMETERS, THE ARCHITECTURE OF THE FEEDFORWARD NETWORKS AS WELL AS THE APPROXIMATION ERRORS AVERAGED OVER 10 FEEDFORWARD NETWORKS. THE JOINTS AND THE POSE ARE REPRESENTED BY γ_i AND $[\xi, \boldsymbol{t}]$, RESPECTIVELY. NEITHER THE JOINTS γ_i NOR THE POSE $[\xi, \boldsymbol{t}]$ ARE SCALED. THE APPLIED ADAM OPTIMIZER [15] IS PARAMETRIZED WITH THE LEARNING RATE λ AND THE FURTHER PARAMTERS $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1 \times 10^{-8}$ AND FOLLOW THE NOTATION IN [15].

Kinematics	Training with Adam optimizer				Architecture				Approximation error			
	N_{bs}	N_{es}	$\lvert\mathcal{S}_{\text{tra}}\rvert$	λ	φ	N_{ip}	N_{h}	N_{op}	e_x [mm]	e_ϑ [°]	e_α [°]	e_β [mm]
Forward	128	200	60,000	0.0002	ReLU	9	100	7	2.23 ± 0.25	1.04 ± 0.08	-	-
Inverse	128	200	80,000	0.0007	ReLU	7	(100, 200)	9	-	-	8.21 ± 0.28	4.0 ± 0.6

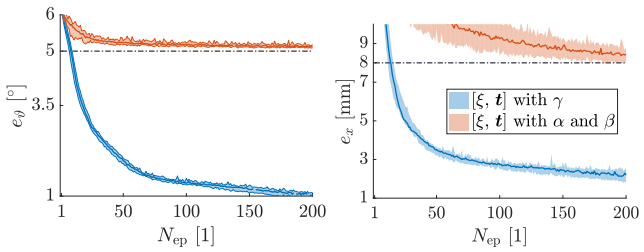

Fig. 8. Influence of the representation of the joint space shown by the development of approximation errors on the test set $\mathcal{S}_{\text{test}}$ over the epochs N_{ep} of the training for every ten feedforward network. The solid red and blue lines show the median values (solid line), which are limited by the minimum and maximum approximation errors (shaded region) at the respective epoch. It can be seen that the feedforward network with γ as representation of the joints reduces the approximation errors more strongly. The ten feedforward networks with untransformed α_i, however, saturate.

networks which directly accept α_i and β_i in the input layer. This reduced the number of parameters in the first layer of the network, while the remaining parameters are the same as for the previous set of networks, i.e. $N_{\text{op}} = 7$ and $N_{\text{h}} = 100$.

Finally, we compared the results of the learned forward kinematics with the samples from our test set $\mathcal{S}_{\text{test}}$ and a kinematic model based on Kirchhoff rod theory [19].

A. Results

The approximation errors of the set of networks with γ as input $e_x = 2.23 \pm 0.25\,\text{mm}$ and $e_\vartheta = 1.04 \pm 0.08\,°$ have been determined from the entire test set $\mathcal{S}_{\text{test}}$ and have been evaluated at $N_{\text{es}} = 200$. This corresponds to a relative position error of $1\,\%$ w.r.t. the total length of the robot.

In Fig. 8 the course of the approximation error e_x and e_ϑ given by (8) and (9), respectively, are shown. It compares the results between kinematic approximation of both set of networks, i.e. the network with joint description γ_i and the network with commonly used joint description α_i and β_i.

Figure 9 shows 100 positions taken from the test set $\mathcal{S}_{\text{test}}$. They are compared with positions calculated by a Kirchhoff based unloaded kinematic model and by a network which utilize (3). The model-based approach [19] achieves an approximation error in position $e_x = 27.4 \pm 5.1\,\text{mm}$ and in orientation $e_\vartheta = 88.3 \pm 37.3\,°$. The approximated pose of the feedforward network results in approximation errors of $e_x = 2.7 \pm 1.2\,\text{mm}$ and $e_\vartheta = 1.3 \pm 0.9\,°$.

B. Discussion

A better approximation can be achieved by transforming joints by means of trigonometric functions (3), which can be observed in Fig. 8. Regarding the set of networks which directly accept α_i and β_i as input, it can be argued that the worse approximation is due to the lower network capacity, which describes the total number of weights and biases in artificial neural network. However, this is still true for different capacities and different number of activation functions in the hidden layer. Therefore, the increase in the parameters resulting from the increase in inputs cannot be cited as a reason.

Referring to Fig. 9, the simulated positions using the Kirchhoff model [19] cannot adequately reflect the measurement data due to the non-modeled external forces generated by the 6-DOF sensor. Force-inclusive models maybe used, however, the complex form of the 6-DOF Sensor introduces difficulties to model forces and moments to the tip, which depend on the pose of the tip. Note that by incorporating external loading, tip error of $2.91\,\text{mm}$ and $1.5\,\%$ of the length is achieved in [19] with a model-based approach. Therefore, by including external forces, we would not expect a better results than in [19] reported relative tip error. For comparison our relative tip error is $1\,\%$ of the total robot length. Furthermore, non-modeled tube tolerances support the approximation error, see Table I. By contrast, these difficult to model effects are covered with small approximation errors by the proposed feedforward network. Hence, this allows the determination of the forward kinematics by observing the end-effector pose for different joints. Beside of the values of the approximation errors, based on the position of four randomized reference positions, the "distortion" of the workspace can be intuitively recognized through modeling or approximation.

Comparing the resulting approximation errors from Fig. 9 w.r.t. Fig. 8, we observe larger approximation errors in the subset of the test set $\mathcal{S}_{\text{test}}$, which has 100 measured data. This indicates that approximation errors depend on the coverage of the workspace. As the inequalities in (1) and (2) need to be satisfied by $\mathcal{B}$, the intended uniform random distribution becomes in fact a non-uniform random distribution in the translational joint space $\mathcal{B}$. As stated and visualized in [3], the non-uniform distribution of β_2 and β_3 reflects the impact of (1) and (2). To force the uniform distribution as well as the division of $\mathcal{B}$ into equidistant intervals is subject of future research.

Another sources of approximation error are a dependence on motion history [6] as well as the influence of hysteresis and clearance [11]. This can be investigated in future work.

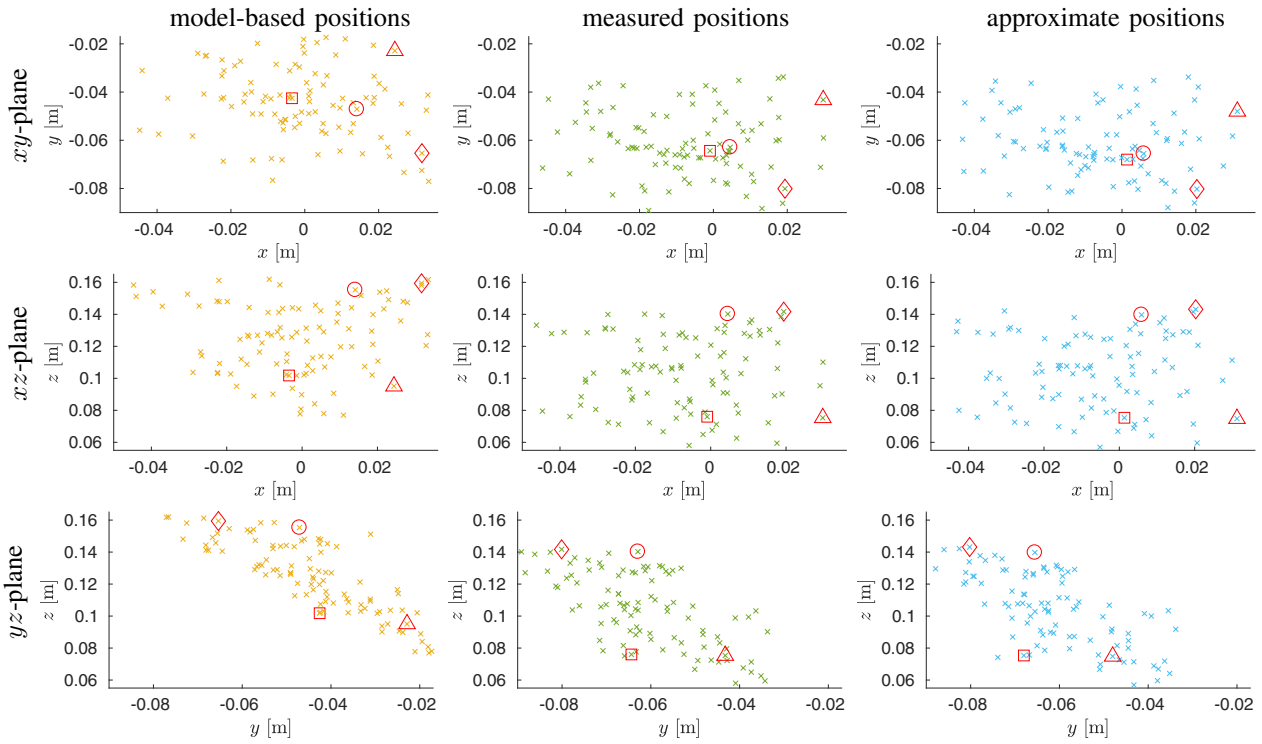

Fig. 9. Projected three-dimensional points in the three planes of the base coordinate system of the robot. From the left to the right column: computed positions via the unloaded Kirchhoff model, measured positions from a robot prototype, approximation of the forward kinematics with data from the test set $\mathcal{S}_{\text{test}}$. The four points give a visual aid in order to see the "distortion" between the different approaches. For the sake of clarity, it may be better to show the deformed shape of each method. However, our neural network provides only the pose of the tip.

V. INVERSE KINEMATICS

In the following, the approximation of the inverse kinematics is reported. Preliminary experiments showed that a network with a single hidden layer cannot approximate the kinematics with a satisfying level of accuracy. Therefore, the network is extended to two hidden layers. By incorporating additional hidden layers with ReLU activation functions, networks are more accurate [16]. The used learning parameters and architecture as well as the achieved approximation errors are summarized in Table III. To train the network 84,000 pose samples have been recorded with the robot prototype from which 80,000 samples are used as training data $\mathcal{S}_{\text{tra}}$ while the remaining 4000 data points form the test set $\mathcal{S}_{\text{test}}$.

A. Results

In Fig. 10 the course of the approximation error e_α and e_β given by (11) and (10), respectively, are shown. We achieve an approximation error of $e_\beta = 4.0 \pm 0.6\,\text{mm}$ and $e_\alpha = 8.21 \pm 0.28\,°$.

In Fig. 11 the individual errors are presented. At $N_{\text{ep}} = 200$ the respective individual approximation error in tube rotation is $e_{\alpha,1} = 3.39 \pm 0.70\,°$, $e_{\alpha,2} = 6.17 \pm 0.27\,°$, and $e_{\alpha,3} = 2.29 \pm 0.21\,°$, respectively. In terms of tube translation, we achieve $e_{\beta,1} = 1.00 \pm 0.17\,\text{mm}$ for the innermost tube, $e_{\beta,2} = 2.00 \pm 0.19\,\text{mm}$ for the middle tube, and $e_{\beta,3} = 2.70 \pm 0.78\,\text{mm}$ for the outer tube.

B. Discussion

The results show good generalization on the test set $\mathcal{S}_{\text{test}}$. This is achieved in a short training duration ($N_{\text{es}} = 200$) and can be further improved. Preliminary tests indicated that a longer training duration (e.g. $N_{\text{es}} = 1000$) have the potential to reduce the position error by more than 10 %. Unfortunately, this comes to the cost of a considerably increased training duration which prevented us from performing experiments in significant quantities.

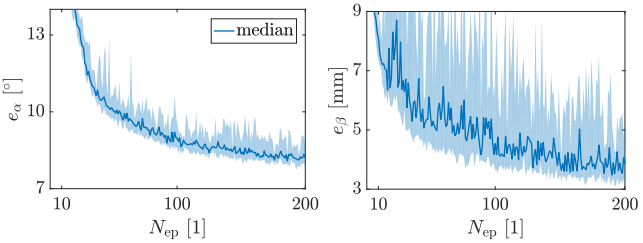

Fig. 10. ReLU networks for the approximation of the inverse kinematics evaluated at test set $\mathcal{S}_{\text{test}}$. Course of the approximation errors e_α and e_β are the median over ten runs with different random initializations, error bars show minimum and maximum, respectively.

Figure 11 shows that the influence of the individual $e_{\alpha,i}$ to e_α are not equally distributed. Especially, $e_{\alpha,1}$ and $e_{\alpha,2}$ show the tendency to decrease with a longer training duration while $e_{\alpha,3}$ is nearly saturated after $N_{\text{es}} = 200$.

As stated before we restricted the rotational angles α_i between $-60\,°$ and $60\,°$ to limit the influence of the applied tip load, which is caused by the mass of the attached sensor. However, we assured that the investigated robot workspace covers two quadrants. While the inverse kinematics trained in [1] estimates a specific set of α angles for each tube and for each quadrant of the Cartesian robot workspace, we could show that a feedforward network is capable to solve the kinematics directly without this kind of redundancy. Our

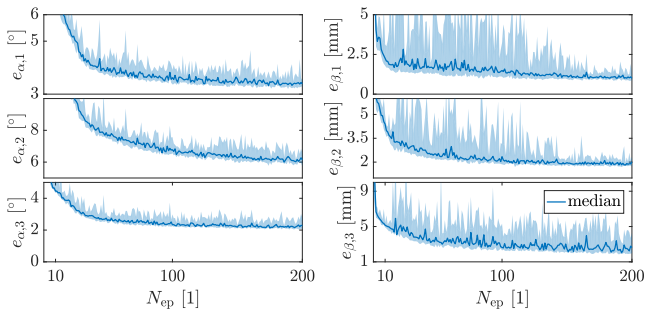

Fig. 11. The individual components $e_{\beta,i}$ and $e_{\alpha,i}$ of the respective approximation errors. $e_{\beta,i}$ is defined as absolute value $\text{abs}(\beta_i - \hat{\beta}_i)$, whereas $e_{\alpha,i}$ is defined in (13). The ribbons show the median, the maximum and the minimum approximation errors for ten runs with different random initializations. Please note the different scaling in y-axes.

preliminary examination in simulation shows that this is still true for all four quadrants and for the entire joint space $\mathcal{Q}$.

Same error source for the forward kinematics applies to the inverse kinematics due to the fact that the same measured data is used in the training set $\mathcal{S}_{\text{tra}}$. Furthermore, we did not consider the Pythagorean trigonometric identity for the rotary joint α_i, which is given by

$$1 = \cos^2(\alpha_i) + \sin^2(\alpha_i) = \gamma_{1,i}^2 + \gamma_{2,i}^2. \tag{18}$$

Thus, (13) and, therefore, (11) produce an error because it cannot be guaranteed that $\hat{\gamma}_{1,i}$ and $\hat{\gamma}_{2,i}$ are between -1 and 1, and are scaled equally. Note that if $\hat{\gamma}_{1,i}$ and $\hat{\gamma}_{2,i}$ differ from $\gamma_{1,i}$ and $\gamma_{2,i}$, respectively, by a scaling, then the errors mentioned above would cancel out each other. The incorporation of (18) can improve the approximation error and is subject of future research.

For the sake of clarity, we do not compare our direct solution for the inverse kinematics with other model-based method due to the fact that the expected results are highly depending on the model-based method itself and on the chosen method, e.g. root finding. This can be carried out systematically in future work.

VI. CONCLUSIONS

In this paper, we considered the problem of accurate kinematics calculation of concentric tube continuum robots. Neural networks and a novel joint representation have been applied in order to approximate the forward and inverse kinematics with measured experimental data from a robot prototype with three tubes. No simplification has been made regarding the degrees of freedom of the robot in three dimensional task space. We achieved higher accuracy for the forward kinematics compared to model-based approaches and provide a direct solution for the inverse kinematics due to the proposed transformation by trigonometric functions. We are confident that with longer training and better parameters, the approximation error can be further reduced, as first trials showed. Moreover, we proposed an adjustment device in order to find the zero point of the rotatory joints. This rapidly allows the determination of common plane by observing the robot's backbone shape. To conclude, no accurate physics-based model and no calibration of model parameters have to be applied in order to achieve fast and accurate kinematics.

ACKNOWLEDGMENT

The authors would like to thank Thien-Dang Nguyen for his valuable contribution by building the robot prototype.

REFERENCES

[1] C. Bergeles, F.-Y. Lin, and G.-Z. Yang. Concentric tube robot kinematics using neural networks. In *Hamlyn Symposium on Medical Robotics*, pages 13–14, 2015.

[2] J. Burgner, D. C. Rucker, H. B. Gilbert, P. J. Swaney, P. T. Russell, K. D. Weaver, and R. J. Webster. A telerobotic system for transnasal surgery. *IEEE/ASME Transactions on Mechatronics*, 19(3):996–1006, 2014.

[3] 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*, pages 1269–1275, 2014.

[4] J. Burgner-Kahrs, D. C. Rucker, and H. Choset. Continuum robots for medical applications: A survey. *IEEE Transactions on Robotics*, 31(6):1261–1280, 2015.

[5] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–341, 1989.

[6] P. E. Dupont, J. Lock, B. Itkowitz, and E. Butler. Design and control of concentric-tube robots. *IEEE Transactions on Robotics*, 26(2):209–225, 2010.

[7] K.-I. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural networks*, 2(3):183–192, 1989.

[8] J. Funda and R. P. Paul. A computational analysis of screw transformations in robotics. *IEEE Transactions on Robotics and Automation*, 6(3):348–356, 1990.

[9] H. B. Gilbert, D. C. Rucker, and R. J. Webster III. *Robotics Research: The 16th International Symposium ISRR*, chapter Concentric Tube Robots: The State of the Art and Future Directions, pages 253–269. Springer, 2016.

[10] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi. Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature. *IEEE Transactions on Robotics*, 31(4):823–834, 2015.

[11] C. Greiner-Petter and T. Sattel. On the influence of pseudoelastic material behaviour in planar shape-memory tubular continuum structures. *Smart Materials and Structures*, 26(12):125024, 2017.

[12] 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*, pages 1026–1034, 2015.

[13] K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[14] M. I. Jordan and D. E. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive science*, 16(3):307–354, 1992.

[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, pages 1–15, 2015.

[16] S. Liang and R. Srikant. Why deep neural networks for function approximation? *arXiv preprint arXiv:1610.04161*, 2017.

[17] J. Lock and P. E. Dupont. Friction modeling in concentric tube robots. In *IEEE International Conference on Robotics and Automation*, pages 1139–1146, 2011.

[18] A. Melingui, R. Merzouki, J. B. Mbede, C. Escande, and N. Benoudjit. Neural networks based approach for inverse kinematic modeling of a compact bionic handling assistant trunk. In *IEEE International Symposium on Industrial Electronics*, pages 1239–1244. IEEE, 2014.

[19] 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*, 26(5):769–780, 2010.

[20] J. M. Selig. *Geometric fundamentals of robotics*. Springer Science & Business Media, 2nd edition, 2005.

[21] J. Stuelpnagel. On the parametrization of the three-dimensional rotation group. *Society for Industrial and Applied Mathematics Review*, 6(4):422–430, 1964.

[22] W. Xu, J. Chen, H. Y. Lau, and H. Ren. Data-driven methods towards learning the highly nonlinear inverse kinematics of tendon-driven surgical manipulators. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 13(3):1–11, 2016.