2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Data-Driven Estimation of Forces Along the Backbone of Concentric Tube Continuum Robots

Heiko Donat[1*], Pouya Mohammadi[1], and Jochen Steil[1]

Abstract— **Concentric tube continuum robots (CTCRs) belong to the family of continuum robots with applications in minimally invasive surgeries. Because of this application domain, measuring the external forces along the body of the robot is paramount. CTCRs are made up of thin elastic rods and are intended to be applied inside the human body, where conventional sensor-based measurements are not feasible. Consequently, research is resorting to *estimate* the forces through geometric, numeric, or optimization methods. However, these methods often suffer from slow convergence. In this paper, we introduce a novel data-driven approach for estimating contact forces along the body of a CTCR that offers an estimation precision comparable to the current state-of-the-art optimization-based approaches, but exhibits nearly two orders of magnitude faster convergence. The proposed method is scalable and exhibits a significant performance in response to a wide range of external forces. The approach was evaluated in simulations and on a real 2-tube CTCR.**

I. Introduction

In robot-supported minimally invasive surgeries, estimating the contact forces along the instruments of medical robots is a fundamental concern, particularly when dealing with soft tissues. Miscalculating these forces and subsequently reacting improperly to them can cause severe damage to the internal organs of patients. This applies to concentric tube continuum robots (CTCRs), where the sensing of contact forces *along* the robot's body are generally important for navigation during an operation. However, the main challenge in measuring contact forces lies in the limitations established by the small form factor of the CTCR, that is, where and how to integrate a force/torque sensor.

A possible solution is Fiber Bragg Gratings (FBG) sensors, which enable real-time force and shape sensing and can be attached to the tubes of a CTCR [1]. However, FBG sensors have to be added at each tube, and integration can be time and cost intensive. Furthermore, CTCR tubes are supposed to be tailored to each patient, which aggravate the FBG integration challenge.

Consequently, some research focus on estimating (rather than measuring) forces from the deflection of continuum robots [2], [3], [4]. However, most methods focus on only estimating the tip contact forces. A method for estimating forces along elastic rods by explicitly modelling the force distribution using discrete shape data was introduced in [5]. In this study, two possible force distribution models are suggested: a Fourier series representation and a series of Dirac delta functions. However, point loads are expected to be more common in a contact scenario, and the conventional Fourier series only models point loads properly when a large number of high-frequency terms are used. Dirac delta representations, on the other hand, require many terms to represent wide and smooth force distributions. To solve this problem, the authors introduced in [6] a mixture of Gaussian as a force distribution model and extended their approach to tendon-driven continuum robots. Both approaches include the geometrically exact kinostatic model for CTCRs introduced in [7] to calculate the shape of the robot in their optimization approach. Although the model in [5] exhibits good precision and repeatability, it has other flaws inherent to its design. Inferring the forces from geometric deformation implies solving an optimization problem with a large number of parameters, where each step involves integrating a system of differential equations. In fact, [6] reports 2.1 seconds for their model to converge, which reduces its applicability in minimally invasive surgery.

Other than numerical estimation approaches, a different way of estimating contact forces is by applying machine learning methods. The machine learning approach — that to the best of our knowledge has remained unexplored for this type of tasks — offers promising properties and results. Our core idea is to leverage the model of [6] to learn the inverse of the solution given by [7]. That shifts the computational burden from online solving of the optimization problem to offline training of the neural network, while exploiting the network's fast online inference. Inverse-model learning has been applied to rigid robotics in kinematics e.g., [8]; statics e.g., [9]; dynamics e.g., [10]; and also for kinematics in soft (e.g., [11], [12]) and continuum robots (e.g., [13]). In this work, we propose an extension of our cascade of supervised learners applied in [3] to estimate multiple contact forces along the body of a CTCR. The proposed method offers comparable precision to the current state-of-the-art methods and performs almost two orders of magnitude faster.

While this approach is seemingly straightforward, in practice it remains a challenging problem. Specifically, the coverage of the training data is a crucial factor for a data-driven approach. Furthermore, as stated in [5] and [6] the estimation of external forces along the body of a continuum robot is an ill-posed problem, as different force distributions can produce the same deformation. In this work, we will tackle the stated problems as follows:

1) Due to the complex workspaces of CTCRs, we focus on the force estimation in a smaller workspace, which still covers 1/4 of the robot's original workspace. This

*Corresponding author
[1]All authors are with the The Institute of Robotics and Process Control - IRP, Technische Universität Braunschweig, Germany. hdo,pmo,jst@rob.cs.tu-bs.de

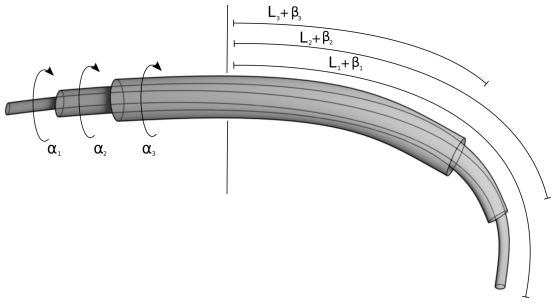

Fig. 1. A schematic of a 3-tube CTCR with sample configuration.

reduces sample cost and exclude partially redundancies. Because CTCRs will operate in confined body cavities and minimized workspaces are desired [14], we do not see our limitation as an obstacle to practical application.

2) To tackle the ill-posedness of the problem, we first generate data for a single contact force and incrementally introduce new datasets to the learner containing one more contact force than the previous one. Thereby creating a prior towards a solution with the output of the preceding cascade layers. Furthermore, we will focus on point loads, as they are more common in contact force scenarios.

In this paper, we will show an advanced application of our deep direct cascade architecture (DDCA) [3] and leverage the transfer learning capability of this approach to incrementally extend the learner to estimate a range of contact forces. We explore the capabilities and limitations of our approach in simulation and validate its applicability on a real CTCR with two elastic tubes.

II. Method

In this section we will briefly introduce the robot's mechanism, configuration space, and its geometric model, describe the applied force model to eventually formally define the problem statement and describe the proposed cascade learning approach for its solution.

A. Robot model

CTCRs belong to the class of continuum robots. They consist of $m \geq 2$ concentric elastic Nitinol or Nylon tubes, where each tube features a pre-curvature κ_i. Each tube consists of a straight and a curved part whose lengths are expressed by $L_i = \{L_{s,i}, L_{c,i}\}$ respectively. The inner and outer diameters of tube i are denoted by $d_i = \{d_{in,i}, d_{out,i}\}$.

Each tube can be rotated and translated using an external actuation unit, as shown in Fig. 6. The configuration $\boldsymbol{q}$ of the robot, hence, is given by:

$$\boldsymbol{q} = (\alpha_i, \beta_i)^T, \quad i \in [1, m], \tag{1}$$

where α_i and β_i are the rotation and translation of the actuator of segment i. Fig. 1 depicts a sample configuration of a 3-tube CTCR. For each configuration $\boldsymbol{q}$, the physical shape of the CTCR will result in a curve, which can be expressed as moving frame $\boldsymbol{g(s)} \in \mathrm{SE}(3)$:

$$\boldsymbol{g}(s) = \begin{bmatrix} \boldsymbol{R}(s) & \boldsymbol{p}(s) \\ \boldsymbol{0}^T & 1 \end{bmatrix}, \tag{2}$$

with a positions vector $\boldsymbol{p(s)} \in \mathbb{R}^3$ and an orientation $\boldsymbol{R}(s) \in \mathrm{SO}(3)$ parameterized over the robot's curve length $s \in [0, S]$.

B. Forces acting on the robot

Consider $\mathcal{N} \geq 1$ external forces being exerted along the robot's tubes. These forces can occur e.g. due to a contact with the environment and can be modeled as point loads. Each external force along the elastic body of the robot causes a deformation. In this work, we assume that point loads occur only along the body of the robot such that the peaks of the forces are not directly at the base or the tip of the robot ($0 < s < S$). This restriction arises from the shooting method we use in combination with the model by [7] to calculate the inner forces and moments acting on the robot's tubes to produce the correct robot shape. This assumption does not impose any limitation on the method, as generated external forces still act very close to the tip, from which the tip force can be implied.

As discussed in Section I, it is possible to parameterize these forces using Fourier series and sliding Dirac-delta functions, as shown in [5]. In our work, we use the parameterize model of [6] based on a Gaussian Mixture Model. In the following, we will briefly describe the formulation of [6] using their own notation. Please refer to this work for further information and an in-depth study.

Consider an external normal force $\boldsymbol{f}_{ext}$ acting on the cross-section at the point s of the robot as

$$\boldsymbol{f}_{ext}(s) = \mathbf{R}(s) \begin{bmatrix} f^x_{ext}(s) & f^y_{ext}(s) & 0 \end{bmatrix}^T, \tag{3}$$

where $\mathbf{R}$ is a matrix for mapping to the moving frame $g(s)$ coordinates and $f^x_{ext}(s)$ and $f^y_{ext}(s)$ acting along the directors d_1 and d_2 of the cross-section. At each point along the rod, the external forces can be expressed by a mixture of Gaussian as

$$\begin{aligned} f^x_{ext}(s) &= \sum_{k=0}^{\mathcal{N}} a^x_k e^{-c_k(s-\mu_k)^2}, \\ f^y_{ext}(s) &= \sum_{k=0}^{\mathcal{N}} a^y_k e^{-c_k(s-\mu_k)^2}. \end{aligned} \tag{4}$$

The Gaussian parameters μ (the mean) and c (related to variance σ by $c = 1/2\sigma^2$) in eq. (4) are associated with the location of an external force and its spread respectively. $\mathcal{N}$ is the number of Gaussians and $\boldsymbol{a}_k = (a^x_k\ a^y_k)$ are the component amplitudes of the k-th Gaussian. For the sake of clarification, consider the simple case of $\mathcal{N} = 2$ Gaussian forces acting upon a rod in Figure 2.

Given the force profile $\boldsymbol{f}^{(\mathcal{N})}_{ext}$ as a Gaussian Mixture Model, we can apply $\boldsymbol{f}_{ext}(s)$ to the model of [7] given the equilibrium:

$$0 = \frac{d\boldsymbol{n}(s)}{ds} + \boldsymbol{f}_{ext}(s), \tag{5}$$

where $\boldsymbol{n}$ represents the inner forces action at arc-length s.

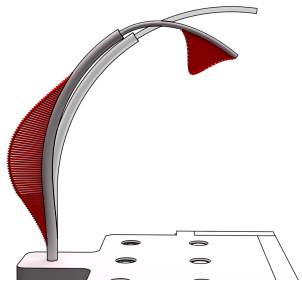

Fig. 2. An example of external forces acting on an elastic rod using a mixture of $\mathcal{N} = 2$ Gaussians (in red) with $a_1^x = 1.12\ N$ and $a_2^x = 0.35\ N$. The center of the forces are $s_1 = 0.05\ m$ and $s_2 = 0.17\ m$. Their spread $c_1 = 1000$ and $c_2 = 10000$ relates to the variance of each Gaussian.

C. Problem statement

In this work, we want to solve the following problem. Given the configuration $\boldsymbol{q}$ of a CTCR with known characteristics m, L, d, and a vector of positions $\boldsymbol{p}$ which describe the robot's shape, find the corresponding external forces acting on the elastic tubes of the robot leading to its deformation.

Our approach is the direct learning of the mapping from deflection to forces exerted along the robot. Generating the force model $f_{ext}(s)$ and applying its values together with the configuration $\boldsymbol{q}$ to the model of [7], we calculate discrete positions $p^{(i)} \in \mathbb{R}^3$, where i is evenly sampled from $[0, S]$ and $\boldsymbol{p} = [p^{(0)}, \ldots, p^{(K)}]^T$. Given this data, we apply a supervised learning approach to find the mapping:

$$(\boldsymbol{p}, \boldsymbol{q}) \mapsto \boldsymbol{f}_{ext}^{(\mathcal{N})}. \tag{6}$$

Due to the nature of supervised learning methods, the input and the output vector of a neural network have a fixed length depended on the encoding of the data. Thus, the sampling density determines the coverage of forces along the robot. We are using raw position data at discrete steps along the backbone, because all other geometric information requires preprocessing that we, for the sake of inference speed, want to avoid.

D. Cascade of supervised learners

In this study, we apply two types of neural network architectures to approximate the described mapping. Our baseline is a multilayer perceptron (MLP), which belongs to the class of fully connected feed-forward artificial neural networks and represents a general nonlinear regressor with a fixed architecture. We will compare it with a learning scheme that uses MLPs but trains them in a cascade, representing an incrementally built architecture. Our cascade approach, DDCA is based on the direct cascade architectures introduced in [15]. Cascading neural networks have been introduced under different notions of cascade correlation (CasCor) [16], direct cascade architectures (DCA) [17] and stacked network generalization [18]. The advantage of cascade networks, including DDCA, over common feed-forward networks is that the architecture does not have to be determined in advance and training is carried out in iterative steps on smaller networks, which reduces training time and memory usage and simplifies hyperparameter optimization [19], [20], [21]. Cascade networks incrementally build a cascade of shallow learners and train them to the same output in every layer. After training, the weights of the layers are frozen. This training approach mediates the vanishing gradient problem present in deep networks. As seen in [22], the pre-training of layers to gain better initialization and further achieve better minima is a specific appeal of deep neural networks. In cascade networks, each layer further minimizes the error, acting as a pretrained prior for the next cascade.

TABLE I
ADAM OPTIMIZER PARAMETERS

parameter	value
initial learning rate	0.001
maximum epochs	200
tolerance for optimization	$1 \cdot 10^{-4}$
β_1	0.9
β_2	0.999
ϵ	$1 \cdot 10^{-8}$

The incrementally generated deep cascade architecture (Fig. 3) further enables the adaption and transfer of prior learned knowledge, acquired by already trained layers. We will exploit these properties to improve the direct learning of the force fields acting on the rods of a CTCR. The first DDCA learner (named M_0) is trained with the input data X to reproduce the target data $\boldsymbol{f}_{ext}^{(1)}$. In this work, the input data X is represented by $(\boldsymbol{p}, \boldsymbol{q})$. For each force profile, the algorithm attempted to minimize the error metric to less than $0.01\ N$. When the loss converges, the learner's parameters are *frozen* and saved. By introducing a new learner (M_1) as a new cascade to the architecture, we can either improve the prediction result by performing training with the same $\boldsymbol{f}_{ext}^{(1)}$ as target, or, adapt to similar data $\boldsymbol{f}_{ext}^{(2)}$ (transfer learning) given the input $(X, \hat{\boldsymbol{f}}_{ext}^{(1)})$, where $\hat{}$ represents the estimation of the learner. New layers are added to the architecture until the global error converges, e.g., if the incremental decrease of the errors falls below the threshold $0.01\ N$. For each cascade n, the input data set is $X_n = (X, \hat{\boldsymbol{f}}_{ext}^{(1)}, \ldots, \hat{\boldsymbol{f}}_{ext}^{(n-1)})$ and its prediction is given by $\hat{\boldsymbol{f}}_{ext}^{(n)}$. In the end, we trained N layers on the same output to improve approximation. The main idea is to resolve the problem of $\mathcal{N}$ forces by creating layers responsible for a specific force profile, that is, for a certain number of peak forces along the rods. The capabilities of this algorithm in the application of tip force estimation were shown in [3].

III. EVALUATION

We evaluated the cascade of learners both in simulation and on a 2-tube CTCR.

A. Simulation Study

To evaluate our approach, we sample forces as described in Sec. II-B and calculate the deformation of the robot given the

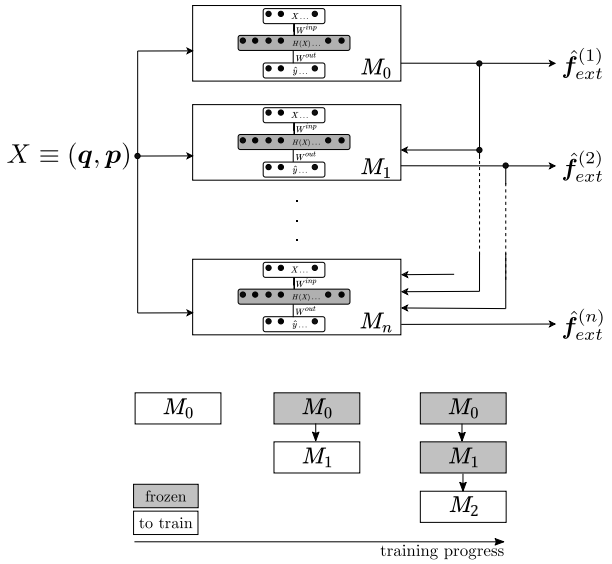

Fig. 3. General structure of the Cascade of Neural Networks. After convergence, a new Neural Network is added and trained on the next force profile and the enhanced input X_i.

model by [7]. As the error metric, we chose the Cartesian root mean squared error (RMSE): $E = \frac{1}{\mathcal{N}} \sum_k \frac{1}{I} \sum_i ||\hat{\boldsymbol{f}}_{ext}^{(k)}(i) - \boldsymbol{f}_{ext}^{(k)}(i)||_2$, where $\boldsymbol{f}_{ext}^{(k)}(i) \in \mathbb{R}^3$ and $\mathcal{N}$ is the amount of estimated force profiles with I samples. As a baseline learner, we use an MLP with the same depth and amount of hidden neurons as the cascaded networks and train it with the stochastic gradient-based optimizer ADAM [23].

First, we compare the results of the MLP baseline with those of our DDCA approach for forces only along the x-axis of the tube's cross-sections to demonstrate the advantages of our approach. Second, we evaluate generalization by testing the baseline MLP and our DDCA approach for three different configuration scenarios of the CTCR. The algorithms have been implemented in Python 3.10 using scikit-learn 1.1.2. All datasets for the evaluation consist of 1000 samples comprising as input positions $\boldsymbol{p}$ along the robot's tubes, the configuration of the robot $\boldsymbol{q}$ and target data $\boldsymbol{f}_{ext}^{(\mathcal{N})}$ ($|\boldsymbol{f}_{ext}^{(k)}| = 50$, $||\boldsymbol{a}_k|| \leq 1\ N, \forall k \in [1, \mathcal{N}]$). The data was split into training data (60 %) and test data (40 %).

Baseline, Input Dimension and Inference Time: We optimized the hyperparameters of the MLP using a grid search. The hidden layer size resulted in 480 neurons which yields $3 \cdot l \cdot 480 + 480 \cdot 3 \cdot 50$ weights, where l is the amount of sampled points along the robot's backbone and the L2 regularization term $\alpha = 0.1$ (ADAM optimizer parameters are listed in Table I). We set the convergence criteria of the DDCA to $0.01\ N$ (cf. [3]). Based on the amount of DDCA layers, we built the MLP with the same layer depth. The MLP achieved an RMSE of $0.027\ N$ for forces along the tubes, whereas DDCA achieved an RMSE of $0.02\ N$. The difference in the overall RMSE is not significant; however, plotting an example force profile (cf. Fig. 4) reveals that DDCA produces a much smoother estimation with a closer match to the peaks in the force profile. The smoothing can be explained by the property

TABLE II

NUMBER OF POSITIONS ALONG ROBOT WITH RMSE

l [#positions]	**185**	**37**	**18**	**12**	**9**
Δpositions $[mm]$	1	5	10	15	20
RMSE $[N]$	0.039	0.020	0.015	0.018	0.0195

of DDCA by which each layer tackles a different scope of the error, which yields in this case a reduction of output noise. However, overshooting occurred along the edges of the peak force. Profiles like these commonly occur when two Gaussians with opposite signs are combined. Opposing estimations between layers are possible, which renders a possible caveat of this method, which however did not occur in most estimation results.

In this study, we use positions tracked along the tubes as the main input because they indirectly describe the deflection of the robot. However, the effect of the number of tracking positions on the estimation precision must be determined. Therefore, we evaluated the DDCA estimation for different amounts of tracked positions as listed in Table II. Apparently, markers with a distance of e.g. $1\ mm$ representing very precise measurement of the tube's shape do not improve the force estimation significantly. The best estimate was obtained with 12 tracking positions, which yields the before mentioned RMSE of $0.02\ N$.

Furthermore, we evaluated the inference time of the DDCA and achieved a mean calculation time of $0.08\ s$ on an Intel Core i7-6700HQ CPU at 2.60 GHz.

Evaluation for different robot configurations: For this study, we generated data for three scenarios:

1) $\alpha_i = 0, \beta_i = 0 \rightarrow$ the robot is in zero configuration.
2) $\beta_i = 0$ and $0 \leq \alpha_i < \frac{\pi}{2} \rightarrow$ the robot only rotates. We restricted α_i to exclude redundant solutions as normal feed-forward networks are not capable of handling redundancies.
3) $0 \leq \alpha_i < \frac{\pi}{2}$, $0 \leq \beta < 0.5 \rightarrow$ tubes are $90°$ rotatable half retractable.

We deliberately excluded redundancies, because this is an unexplored problem which is beyond the scope of this study.

For each scenario, we generated datasets with 1 to 3 force peaks, each with 1,000 data samples. The peak forces $||\boldsymbol{a}_k|| \leq 1\ N$. The parameters for the Gaussian distributions were uniformly sampled, with properties similar to those described in [6] and $c = 1000$ to simulate point loads.

For each scenario, we trained each learner with data with maximum of 1 to 3 force peaks sequentially. We tested one DDCA (DDCA-transfer), where for each dataset, a single cascade was added to evaluate the DDCA transfer capabilities. We also trained another DDCA (DDCA-transfer-opt) in which multiple cascades were generated for each dataset to push the training error below $0.01\ N$, which is the same approach for each dataset separately as in the baseline comparison. The cascade learner is trained as described in Sec. II-D. The MLP was generated with a

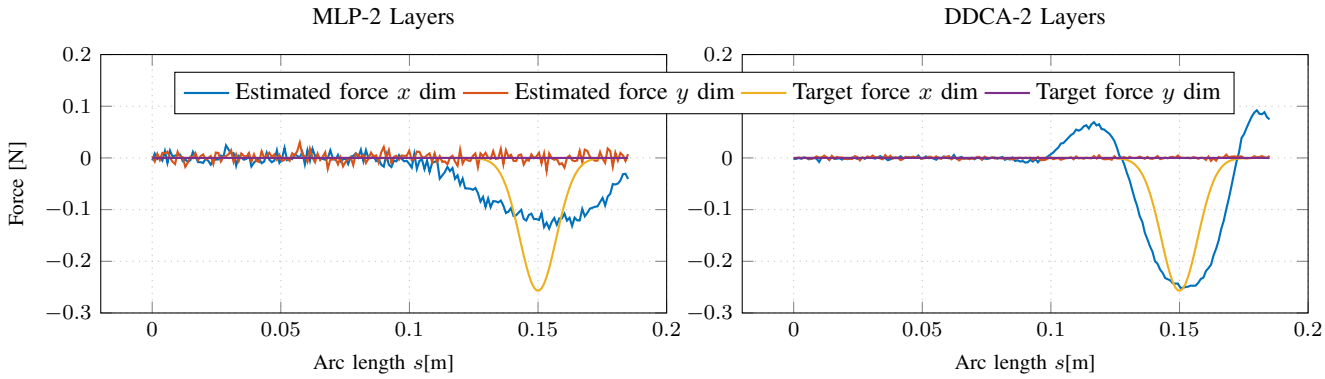

Fig. 4. Comparison of simulated force profiles generated by an MLP with two hidden layers and a DDCA with two hidden layers trained on simulated data: In a back-propagation-based learner such as the MLP in each optimization step, all weights are changed simultaneously. In DDCA, the first layer learns the general problem and the second layer removes persisting errors, which leads to a smoother and more precise estimation.

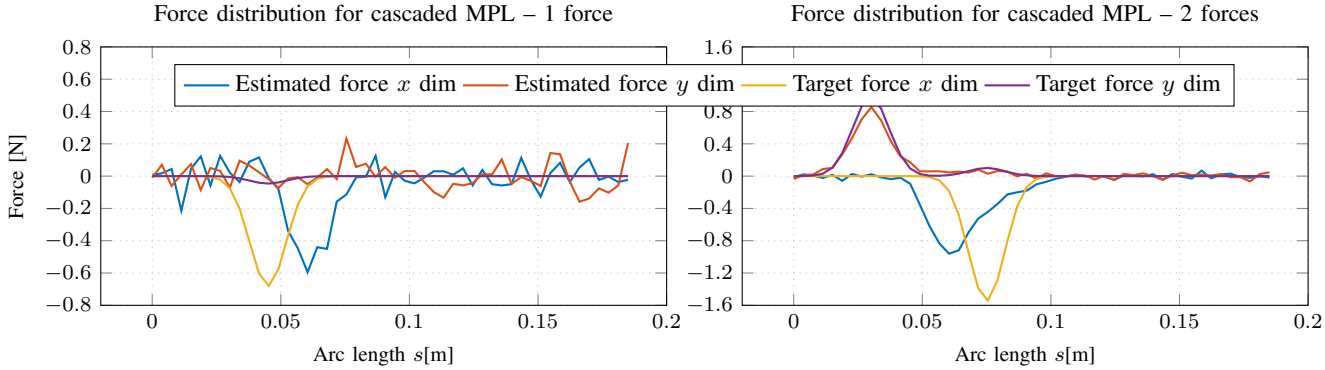

Fig. 5. Example results of the experiment on real hardware. Left: Cascade MLP with 1 force (only the Kuka LWR). Right: Cascade MLP with 2 forces (both a calibrated load and the Kuka LWR, see Fig. 6).

layer depth equal to the force peaks in the data. To fairly compare our baseline, we extended the dataset for each force profile with data with fewer force peaks. The results are listed in Table III. The results show an improved estimation precision for cascaded learners. The DDCAs showed a mean improvement of approximately $0.01\ N$ compared to the MLP. It can be observed that the additional movement of the robot and the increase in the number of force peaks reduced the estimation precision. However, the reduction in precision for the scenario of retractable tubes is linked to the fixed input size of the networks, because shorter tubes are tracked with fewer markers. We further see a small increase in the estimation error if we adapt the DDCA to three force peaks. To evaluate this trend for more than three force peaks, we trained a DDCA with data containing up to seven force peaks in the force distribution for scenario 1 and saw a rise of the RMSE to $0.097\ N$. Evaluating the location of the error showed that higher error occurred especially close to the tip of the robot. Because forces close to the base have a high impact to the positions close to the tip, this can introduce a high bias in lower cascade layers, posing the significance to investigate the application of geometric invariant features for very deep cascade architectures in future work. Although the difference between DDCAs and MLPs was not significant, DDCA enabled the retention of intermediate steps. This helps address the ill-posedness of the problem by providing multiple solutions for a single deformation for a range of contact forces. Furthermore, it helps improve the explainability of the output of the DDCA. In general, the results show that the estimation of forces along the robot with supervised learners produces comparable results to the state-of-the-art methods.

B. Experimental Evaluation

For the evaluation of the robot setup we used a 2-tube CTCR shown in Figure 6. The parameters of the robot are described in Table IV. The robot's tubes were additively manufactured from nylon 618 (Taulman3D, USA) stained with retroreflective paint at equidistant (1 cm) intervals, practically splitting the rod into multiple sections. The position of each section was calculated using a tracking system (Optitrack-Prime-13-System, NaturalPoint, USA). The calculated positions of all sections constitute a discretized representation of the rod's shape.

We applied external forces to the robot using a redundant serial manipulator (7 degrees of freedom, Kuka LWR-4+). By using an additional manipulator, we ensured that the forces were applied in a repeatable and precise manner, which eliminated any inaccuracies arising from human errors. The manipulator was equipped with a precise 6-axis force/torque

TABLE III

RMSE FOR SCENARIO 1, 2, AND 3 GIVEN 1, 2, AND 3 PEAK FORCES

RMSE for scenario 1			
	1	**2**	**3**
MLP	0.044	0.055	0.066
DDCA-transfer	0.031	0.066	0.070
DDCA-transfer-opt	0.030	0.041	0.050
RMSE for scenario 2			
	1	**2**	**3**
MLP	0.050	0.062	0.071
DDCA-transfer	0.040	0.050	0.058
DDCA-transfer-opt	0.040	0.049	0.057
RMSE for scenario 3			
	1	**2**	**3**
MLP	0.047	0.058	0.074
DDCA-transfer	0.057	0.063	0.072
DDCA-transfer-opt	0.043	0.056	0.071

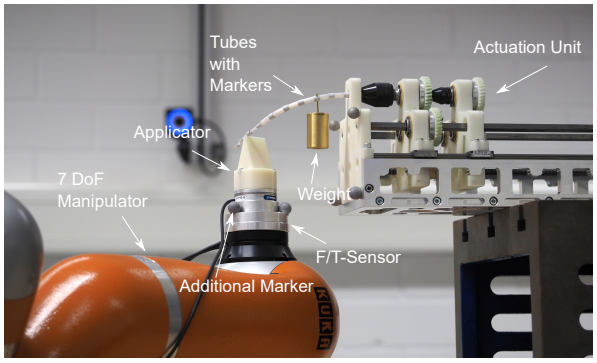

Fig. 6. Close-up experimental evaluation: A Kuka LWR has an F/T-sensor mounted and drives along the continuum robot's tubes with a contact. We attached an additional weight representing an additional second force.

sensor (F/T-Sensor)(ATI Nano 43, ATI, USA) between its last joint and its tip. The z-axis of the F/T-sensor is pointed upward. An applicator is mounted on the F/T-Sensor to enable a point force application to a predefined position (see Fig. 6). The Kuka LWR-4+ was moved along the tubes, pushing upward. To measure the position of the forces precisely, forces were applied to the attached markers. We generated 78 data samples for a single contact force. The robot was limited in its movement, as described in Sec. III-A scenario 2. Furthermore, we recorded 45 samples where a 100 g weight was placed 4.5 cm away from the tube base. Hereby creating a second force pushing downward. The robot was rotated by α in the same fashion as in the first data set.

Using 40 % of data to create an additional cascade layer in the DDCA architecture to adapt the network to the recorded data, we achieved a global RMSE of $0.12\ N$ for the first scenario and a global RMSE of $0.17\ N$ for the latter scenario. Fig. 5 shows two example solution of the network for one and two point forces.

The overall performance of the real robot was not as good as that of the simulated scenario. This is because of the following reasons. First, there is the deviation of the simulated model from the real model based on the material properties, which were estimated for this experiment. We should emphasize that this phenomenon, also known as the reality gap, is not an inherent shortcoming of the method, rather, it is a common phenomenon when switching from simulation to real robot. Second, the chosen input feature representation, which is in this case raw position data, is susceptible to small deviations and requires precise calibration. One approach to address this problem is the application of geometrically invariant features, such as curvature. However, this requires additional calculation steps for each estimation, which may lead to a reduced estimation frequency, and should be investigated in future work. The small deviations are visible by the shift of the force peaks along the x-axis in Fig. 5.

TABLE IV

PARAMETERS OF THE 2-TUBE CTCR USED IN THIS WORK.

i	L_s	L_c	κ	d_out	d_in
1	95	90	11.26	2.5	1.0
2	34	90	11.26	5	3

*All values in mm, except κ in mm^{-1}.

IV. CONCLUSION

In this study, we demonstrated the application of a deep direct cascade architecture to learn external forces along the body of a concentric tube continuum robot. We validated the approach through simulations and on a 2-tube CTCR with additively manufactured nylon tubes. Our approach produced smoother estimations than a comparable multi-layer perceptron with a similar depth. The algorithm is applicable to a real robot with a reasonable margin of error. We achieved an estimation time that was two orders of magnitude faster than that of the current state-of-the-art method with similar precision. The deep direct cascade approach offers a learning architecture with a fast convergence. Furthermore, in the case of multiple peak forces, multiple estimations are produced for 1 to $\mathcal{N}$ force peaks. With additional information about the environment, such as contact points, the correct layer can be chosen to determine the appropriate estimation.

While the method is very promising, there are still open research questions for future work. To improve the estimations and make the approach stable towards calibration errors and noise, a geometrical invariant feature representation for the curvature of the tubes has to be investigated. Contradicting cascade estimations are a possible caveat of the method as seen in Fig. 4, which has to be investigated in future work. Furthermore, a thorough investigation of the approach for the full workspace in combination with redundancy resolution has to be conducted.

ACKNOWLEDGEMENT

We would like to extend our gratitude to Mr. Yazan Soliman who helped us with commendable dedication in conducting the experimental evaluation of the paper.

REFERENCES

[1] C. Shi, X. Luo, P. Qi, T. Li, S. Song, Z. Najdovski, T. Fukuda, and H. Ren, "Shape sensing techniques for continuum robots in minimally invasive surgery: A survey," *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 8, pp. 1665–1678, Aug. 2017.

[2] D. C. Rucker and R. J. Webster, "Deflection-based force sensing for continuum robots: A probabilistic approach," *Proc. IEEE International Conference on Intelligent Robots and Systems*, pp. 3764–3769, Sep. 2011.

[3] 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, Nov. 2020, conference Name: IEEE Transactions on Medical Robotics and Bionics.

[4] F. Feng, W. Hong, and L. Xie, "A learning-based tip contact force estimation method for tendon-driven continuum manipulator," *Scientific Reports*, vol. 11, no. 1, Sept. 2021.

[5] V. A. Aloi and D. C. Rucker, "Estimating Loads Along Elastic Rods," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 2867–2873, iSSN: 2577-087X.

[6] V. Aloi, K. T. Dang, E. J. Barth, and C. Rucker, "Estimating forces along continuum robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8877–8884, 2022.

[7] 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, Oct. 2010, conference Name: IEEE Transactions on Robotics.

[8] A. D'Souza, S. Vijayakumar, and S. Schaal, "Learning inverse kinematics," in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the Next Millennium (Cat. No. 01CH37180)*, vol. 1. IEEE, 2001, pp. 298–303.

[9] R. Rayyes, D. Kubus, and J. Steil, "Learning inverse statics models efficiently with symmetry-based exploration," *Frontiers in Neurorobotics*, vol. 12, Oct. 2018.

[10] K. Hitzler, F. Meier, S. Schaal, and T. Asfour, "Learning and adaptation of inverse dynamics models: A comparison," in *2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids)*. IEEE, Oct. 2019.

[11] T. G. Thuruthel, E. Falotico, M. Cianchetti, and C. Laschi, "Learning global inverse kinematics solutions for a continuum robot," in *ROMANSY 21 - Robot Design, Dynamics and Control*. Springer International Publishing, 2016, pp. 47–54.

[12] G. Fang, Y. Tian, Z.-X. Yang, J. Geraedts, and C. Wang, "Jacobian-based learning for inverse kinematics of soft robots," 12 2020.

[13] 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 *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 5125–5132.

[14] C. Bergeles, A. H. Gosline, N. V. Vasilyev, P. J. Codd, P. J. del Nido, and P. E. Dupont, "Concentric tube robot design and optimization based on task and anatomical constraints," *IEEE Transactions on Robotics*, vol. 31, no. 1, pp. 67–84, 2015.

[15] E. Littmann and H. Ritter, "Learning and Generalization in Cascade Network Architectures," *Neural Computation*, vol. 8, no. 7, pp. 1521–1539, Oct. 1996.

[16] S. E. Fahlman and C. Lebiere, "The cascade-correlation learning architecture," *Proc. 2nd International Conference on Neural Information Processing Systems*, p. 524–532, Oct. 1989.

[17] E. Littmann and H. Ritter, "Cascade Network Architectures," in *Joint Conference On Neural Networks*, 1992, pp. 398–404.

[18] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[19] B. L. S. da Silva, F. K. Inaba, E. O. T. Salles, and P. M. Ciarelli, "Fast deep stacked networks based on extreme learning machine applied to regression problems," *Neural Networks*, vol. 131, pp. 14–28, 2020.

[20] L. Deng, X. He, and J. Gao, "Deep stacking networks for information retrieval," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 3153–3157.

[21] B. Hutchinson, L. Deng, and D. Yu, "Tensor deep stacking networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1944–1957, 2013.

[22] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 07 2006.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014. [Online]. Available: https://arxiv.org/abs/1412.6980