2025 25th International Conference on Process Control (PC)
June 3–6, 2025, Štrbské Pleso, Slovakia

Learning-Based Position Control for Continuum Robots

1st Peter Jan Sincak
Technical University of Kosice
Faculty of Mechanical Engineering
Kosice, Slovakia
peter.jan.sincak@tuke.sk

2nd Michal Kelemen
Technical University of Kosice
Faculty of Mechanical Engineering
Kosice, Slovakia
michal.kelemen@tuke.sk

3rd Martin Varga
Technical University of Kosice
Faculty of Mechanical Engineering
Kosice, Slovakia
martin.varga.2@tuke.sk

4th Ivan Virgala
Technical University of Kosice
Faculty of Mechanical Engineering
Kosice, Slovakia
ivan.virgala@tuke.sk

***Abstract*—The applications, where continuum robots are very useful, such as the inspection of hard to reach tight spaces, jet-engines inspections or applications in medicine, require precise control. This paper focuses on a supervised learning-based positional controller for cable-driven continuum robots. The proposed controller is based on training a feed-forward neural network using positional data. The positional data were gathered from a single kinematic structure by applying 30 000 combinations of cable shortenings and measuring the achieved tip positions of every single one of them. The proposed controller was tested on an experimental test bench, using three different kinematic structures. The results show that one of the structures achieved better results, however it was also used for data gathering. The difference in results comes from the different mechanical and material properties of these structures, even though at some points the measured trajectory was quite close to the targeted one. By evaluating these results, our future work will focus on characterising common features and introducing them to the learning process, so that more generalised models can be achieved.**

***Index Terms*—cable-driven robots, positional controller, continuum robots, supervised learning.**

I. INTRODUCTION

The applications of continuum robots like jet engine inspections, general inspections of tight spaces and cavities or applications in healthcare, show that not only the mechanical design is posing challenges but also the navigation and control of the designed kinematic structures is necessary for the utilisation of the potential of these robots. In order to navigate the robots into the desired position, the sensory feedback is important but challenging when it comes to the continuum robots, that have small diameters [1]. Therefore, open-loop control can be used as an option, for control when sensory feedback is unavailable, and hardware or an accurate enough model of the robot is available.

The control methodologies used on continuum robots are using two main approaches. The model-based approaches utilising classical analytical methods to map relations between the actuator space and task space of the robot or the model-free methods, using the learning-based approaches to learn these relations. In this paper, we adopt the latter approach to investigate the abilities of the supervised learning-based positional controller on multiple kinematic structures of continuum robots actuated by three cables.

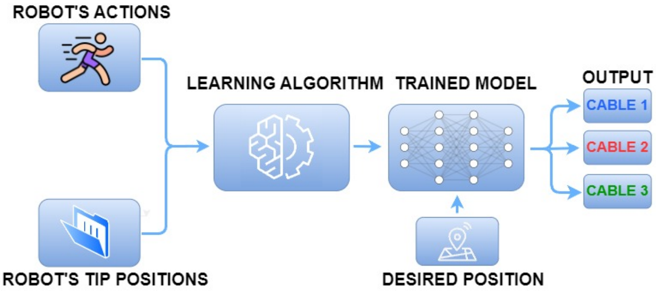


Fig. 1: Scheme representing supervised learning approach training neural network to position the robot into the desired position by predicting the cable shortenings, that control the robot.

The main contributions of this paper are in the supervised learning-based control algorithm, predicting cable shortenings to position the cable-driven continuum robot in the desired position, and the dataset that was created, composed of 30,000 XYZ coordinates of the tip positions of the robot and cable shortenings that caused the robot to reach those positions. The dataset can be further used for other types of control algorithms and research purposes. The scheme of the proposed approach can be seen in Fig. 1, where data were collected on real hardware. By applying 30,000 possible cable shortening combinations, achieved tip positions of the continuum robot were measured with the mocap system to obtain coordinates. This data were then used for training a regression type of model for predicting cable shortenings to achieve the desired position. The FFNN model was trained on a dataset gathered from a single kinematic architecture of a cable controlled continuum robot. By testing the model on a simple positioning task on multiple kinematic structures, the performance of the trained model is compared and the results show that the highest accuracy was achieved on the structure that was used for data gathering. The dataset, CAD files and training code can be found at https://github.com/ARM-Lab/Learning-Based-Position-Control-for-Continuum-Robots .

II. RELATED WORK

The use of machine learning for the control of continuum robots can be helpful in this domain, as these robots have unique mechanical designs that can be challenging to accurately model and control. One of these approaches can be seen in [2], in this paper the authors show a reinforcement learning approach to learn how to actuate in 2D, a pneumatically controlled multi-section continuum robot. Their approach consists of Q-learning algorithm which is used for selecting the appropriate pressures for the pressure chambers that are moving the robot's tip to the desired location. In order to use this algorithm, both the state space and action space have to be discrete, so that the table of states-actions pairs can be created and used. The authors decided to use four predefined actions and split the task space into finite number of segments. This is used to train an agent to navigate the robot from any given state to a randomly selected target within the robot's reach. The agent then acts based on the trained policy and selects the learnt pressures for the pressure chambers. Similarly, the use of Deep Q learning was shown in [3], for positional control of pneumatically controlled continuum robot. The states of the robot are discretized into 0.01 m grid, centred around the selected target location. In [4], authors showed an approach for closed-loop dynamic control of soft robotic manipulator. This approach starts with a trained forward dynamic model that is trained on data collected from actual robot. Further, number of sample trajectories are generated forward dynamic model for training a closed loop policy. The use of such forward dynamic model eliminates the need for the analytical models. Similar approach of using the reinforcement learning was also shown in [5]. The authors proposed a control algorithm based on a deep reinforcement learning approach called Trust Region Policy Optimization (TRPO) [6]. The learning process was done on simulation of pneumatically actuated soft-robot, where the model was learnt by Long-short Term Memory network. In this paper, the TRPO is actually learning the controlling task by having a trust region that avoids the large updates of the policy network which in case of misstep can undo all the learning up to that point.

A different approach was shown in [7], here the authors proposed an approach using FFNN for learning inverse kinematics of soft cable driven continuum robot, that cannot be approximated by constant curve. In their approach, a FFNN network was used with the inputs were generated by direct kinematics model, using desired positions as inputs and predicting the cable tensions used for controlling the robot. Similarly in [8], authors proposed training a FFNN on collected data from real-world or finite element simulation, to predict optimal control inputs for the robot. Then minimise the gradient of the control objective including the network Jacobian, which has information about the correlations between the changes in the inputs and outputs. In [9], authors tested three different regression methods extreme learning machine, Gaussian mixture regression and K-nearest neighbours regression. All of these methods were trained on dataset to learn inverse kinematics of a tendon-driven surgical manipulator and tested by following a trajectory. The results showed the highest accuracy with the KNNR around 2 mm.

The use of reinforcement learning is gaining popularity in the soft robotics community [10]. However, it requires either interaction with real hardware, which can be impractical and slow or a model that can simulate the robot during the training phase, however it can lack imperfections and limitations of real robot. For these cases, it can be more beneficial to create a learning-based model, using supervised learning, that can be used for simple positioning tasks, as well as for training more sophisticated control algorithms.

III. OUR APPROACH

The approach that is proposed in this paper is a learning-based controller using supervised learning. In our approach, supervised learning was used to train a model that predicts how to actuate the robot in order to get it into the desired positions. In this case the supervised learning model is used to predict real values of cable displacements, therefore it is a regression type of model. The regression model relies on the data that were collected from a continuum robot prototype. For data collection, it was opted to generate 30 000 different random actions for the three motors controlling the cables of the robot in order to cover as much of the workspace of the robot as possible. Before every action, the robot started from the initial position, which was a straight position parallel to the ground and every cable was at the centre of its range of elongation or shortening. The robot received actions, positioned itself according to the actions and the tip position was measured. These XYZ coordinates were saved together with the actions for the robot. This process could be called labelling, pairing the inputs with the outputs of the system. Part of this process is captured in Fig.1. In this case, the supervisor's role is automated, and data labelling is done as the positions are measured. The dataset has to be divided into training data and testing data. The training data are used for training the model. The 80% of data points were used for training and the rest was used for testing. The testing data are data points, that the model has never seen during the training and are new to the model. The proposed model is made of feed-forward neural networks containing three hidden layers and an input layer with three neurons and an output layer with three neurons as well. The algorithm used for training the supervised learning model can be seen in Algorithm 1. The initial data from the dataset are shuffled to randomise the data. These are then split to training data and validation data, one is used only for training and one only for the validation of the model. It can be seen that to properly tune the hyper-parameters of the model,

```
Function main():
    Initialize model and data parameters;
    Load and shuffle input data from 'dataset';
    Split data into training and validation sets;
    // Define model building function
    Function build_model(hp):
        Build a model with hyperparameter tuning;
        Compile model with Adam optimiser and MSE loss;
        return Compiled model;
    // Configure and run hyperparameter tuner
    Create RandomSearch tuner to optimise model hyperparameters;
    Set tuner objective;
    Perform hyperparameter search using training and validation data;
    Get best hyperparameters found by the tuner;
    // Build and train final model
    Build model using best hyperparameters;
    Train final model ;
    // Evaluate and save model
    Evaluate model on validation data;
    Output model summary;
    Save trained model;
main();
```

Algorithm 1: Supervised learning algorithm

the keras tuner library was used. This library iterates over multiple sets of hyper-parameters to find the best ones. These are then used to train the final model. The hyperparameters that were tuned can be seen in Table I. Here, it is visible that only two parameters were necessary to tune to achieve results presented in the section IV-C. The tuner used in this case, was a hyperband tuner, which is based on training many models with different parameters combinations just for few epochs and keeping the best performing ones and training them further.The final model, was composed of 3 hidden layers with 320 neurons in each layer and using 1×10^{-2} learning rate for its training.

TABLE I: Hyperparameters

Parameter	**Hyperparameters tuning**		
	Range	***Step***	***Final***
Units	8 - 512	8	320
Learning rate	1×10^{-1} - 1×10^{-8}	0.1	1×10^{-2}

IV. EXPERIMENTAL RESULTS AND DISCUSSION

A. Hardware

For testing purposes, a test bench made for cable-driven continuum robots was used. This test bench consists of three servomotor units that pull the cables for actuation of a cable-driven continuum robot structure attached on the bench. The servomotor units are mounted on the other side of the face plate, where the kinematic structures are attached to. Each of the motors rotational movement is translated to linear movement, where each rotation corresponds to 8 mm of linear movement of a part called runner. Each of the cables are attached at the end of the robot and routed through the kinematic structures and set of pulleys to runner, where they are attached. Depending on the movement of the runner the cables can be shortened or extended and move the kinematic structure. The advantage of such testing bench is that different kinematic structures with similar dimensions can be easily swapped and tested without any additional changes. For our experiments, we were using three somewhat similar kinematic structures and tested the proposed control algorithm. We introduced these structures in [11], where we tested their behaviour. All of the kinematic structures have the same diameter, length and all of them are controlled by three cables. The actuation logic is based on shortening one or two cables at the same time, while the third cable is not actuated. Using this logic, it is ensured that all the cables are reasonably tensed during the movement, so that there are no loose cables, which could possibly get tangled up and influence the movement. This actuation logic, also handles the issue of multiple possible actuation combinations for a single end-effector position, since each action gives a unique end-effector position. The CR1 structure in Fig. 2 is made of flexible material, while CR2 and CR3 are made only of rigid PLA. In CR2 the top part of a sub-segment is made to fit into the bottom part of another sub-segment. By adding a central flexible backbone made of flexible tube, same as in CR1, the movement of sub-segments is limited but still possible in respect to each other. The CR3 is made of multiple rigid sub-segments, which are made to fit into each other. The bottom part of the sub-segment has a V groove, in which another sub-segment fits in. This groove also allows for the rotation of sub-segment in one axis perpendicular to the groove. The direction of the grooves is rotated by 90 degrees at every sub-segment, so that the movement into every direction is possible. As in the previous cases, the central backbone is made of flexible tube.

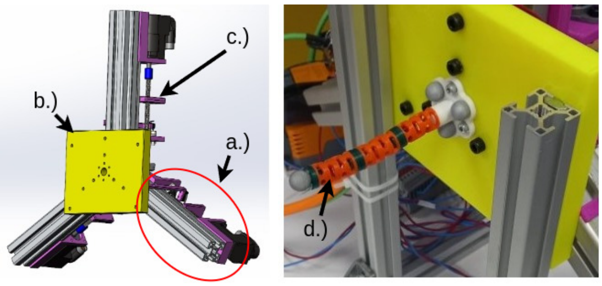


Fig. 2: Test bench consisting of three servomotor units and faceplate, with an instance of attached kinematic structure on the other side of the faceplate. a.) servomotor unit, b.) face plate, c.) runner, d.) kinematic structure CR1

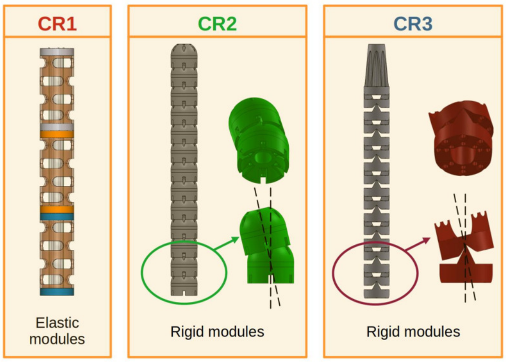


Fig. 3: Experimental cable-driven continuum robot kinematic structures [11].

B. *Experiment 1 - data collection*

The first experiment necessary for the proposed supervised learning algorithm is the collection of data from the real hardware. The use of such data has many advantages compared to artificial data generated from a model. Such a model is always an approximation of some parts of the real hardware to achieve lower computational costs. These approximations are therefore neglected and are not taken into account when learning-based algorithms are using these artificial data. The data from real hardware can therefore better capture the real state of the system and take into account all the imperfections, material properties and forces acting during the movement of the robot that would be difficult to take into account in the model. For this experiment, the first continuum robot structure from Fig. 3, CR1 was selected. The selected kinematic structure was mounted on the testing bench and a marker was placed at the tip of the continuum robot. The task of this experiment was to generate actions that would be able to cover as much of the robot's workspace as possible and capture the tip positions of the robot. The marker at the tip was tracked by the motion capture cameras and after every action the tip position was measured. The actions were ranging from 0 to 10 mm of cable shortenings with a step of 0.1 mm. By following the previously mentioned actuation logic, 30 000 possible action combinations were found and sent to the robot. Then each of the actions was paired with the measured achieved tip position of the robot, which can be seen in Fig.4.

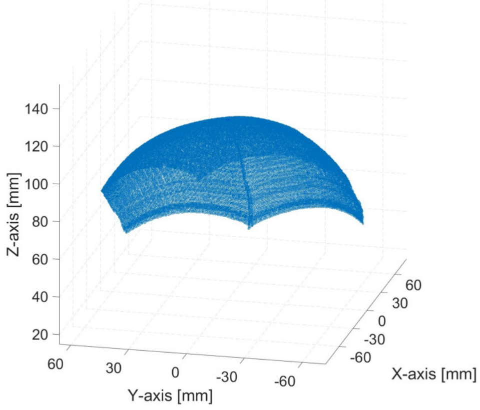
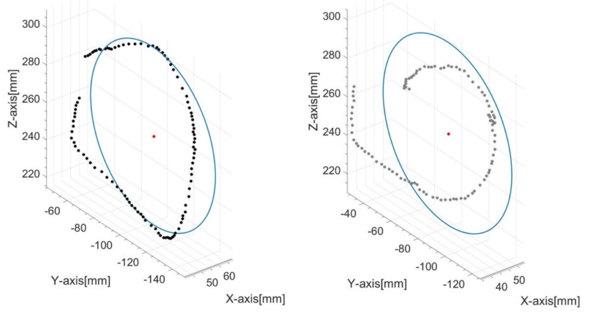


Fig. 4: Measured tip positions

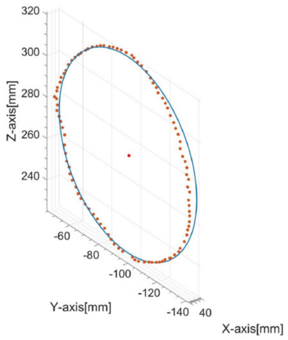
C. *Experiment 2 - trajectory following*

In section III, we introduced proposed supervised learning algorithm that is trained on collected data from real hardware. In this experiment, we focused on following the trajectory within the robot's reachable workspace. Specifically, the robot followed a circular trajectory. The trajectory was sampled in 100 consecutive points that were fed as input to the trained neural network, which was predicting the necessary actions/cable shortenings to get with the tip of the robot to the target location. The location of a marker at the tip of the robot was measured and saved. The proposed algorithm was tested on all three structures to compare performance. In Fig. 5 the final followed trajectories of all three structures are shown. It can be seen that the trajectory was followed by all three of the kinematic structures, but some perform better than the others. In Fig. 5a the final trajectory of CR3 is shown. This structure is made of multiple rigid sub-segments. However, in addition, these sub-segments have an interlocking mechanism in the form of the grooves. These grooves are perpendicular to



(a) CR3

(b) CR2



(c) CR1

Fig. 5: Measured trajectories of kinematic structures a.) CR3, b.) CR2, c.) CR1

each other so that the structure can move to all the directions, but have higher rigidity. The measured trajectory is loosely flowing the pre-described trajectory, however significant deviations can be observed. The actual trajectory was not able to finish the full circle and at the end the trajectory is not connected. The maximal measured deviation from the targeted trajectory was 18.96 mm, minimal 7.72 mm and an average deviation was 13.74 mm. Similar results can be seen in Fig. 5b, where the algorithm was tested on CR2 structure. In this case, the deviation is even more significant and the measured trajectory is less closely resembling the targeted circle. Like the previous structure, this one is also made of rigid sub-segment, placed on top of each other, however the top of each sub-segment is formed in the semi-spherical shape so that the sub-segments can slide on top of each other. Based on the measured data the maximal measured deviation from targeted trajectory was 27.02, minimal 14.35 and average 20.15 mm. Much better results can be seen in Fig. 5c. In this figure the measured trajectory from structure CR 1 can be seen. In this case, the trajectory is much closer to the targeted trajectory. The maximal measured deviation from targeted trajectory 7.3 mm, minimal 2.91 mm and average 5.37 mm. All of these deviations can also be seen in Fig. 6, where we can see that the lowest deviations were achieved with the structure CR1. The first structure, performed much better and the average error was measured to be 5.37 mm compared to the CR2 and CR3, where it was 13.74 mm and 20.15 mm. Here, it was also observed that the majority of the deviation is in X-axis. Similar findings, were also stated in our previous paper [11], where we tested the repeatability of the structures.

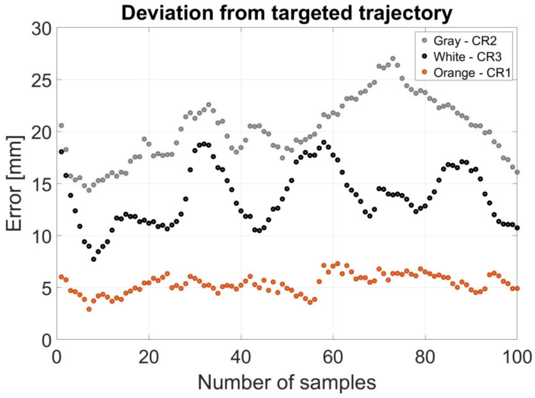


Fig. 6: Distance of actual trajectory from target trajectory

The predicted actions during the inference phase of experiment 2 in section IV-C were treated according to the actuation logic described earlier. In this case the NN predicted three values that were representing the cable shortenings. According to the logic presented earlier in this paper, one or two cables could be shortened at the same time. The NN, however predicted some actions for all three cables, but some actions were well below the minimum step of 0.1 mm that was set for the dataset. Hence, these values were very small and close to 0, so they were treated as 0 during the testing.

TABLE II: Comparison of experimental results

Measured Error	**Kinematic structures**		
	CR1	***CR2***	***CR3***
Maximum	7.3 mm	18.96 mm	27.02 mm
Average	5.37 mm	13.74 mm	20.15 mm
Minimum	2.91 mm	14.35 mm	7.72 mm

V. CONCLUSION

In this paper, we present a supervised learning control algorithm used to control a cable-driven continuum robot. Using motion capture cameras, we created a dataset containing 30 000 cable shortenings that control the robot's movement and their appropriate achieved tip positions of the robot. The proposed supervised learning algorithm is trained on the created dataset and its performance is tested on three somewhat similar kinematic structures of continuum robots. The best results were achieved with the structure that was used for data collection and the average error was 5.37 mm, however using structures CR2 and CR3 the resultant trajectory at some points was close to predefined trajectory. The worse results come from the different mechanical and material properties of the kinematic structures, this claim can be supported by our previous paper, where we tested the repeatability of these structures and CR1 showed the highest repeatability and the smoothest movement during motion. There, we also stated that worse performance of structures CR2 and CR3 compared to CR1, can be caused by higher number of rigid sub-segment stacked on top of each other.This is introducing additional friction and causing less predictable and smooth movement than the CR1 structure. Using this knowledge, in our future work we will focus on characterising some common features between the structures and introducing them to the learning-based controllers, e.g. as some kind of prior knowledge, could result in more general models. These could then be used for multiple kinematic structures and achieve better results.

ACKNOWLEDGMENT

This research was funded by Slovak Grant Agency VEGA 1/0436/22 Research on modeling methods and control algorithms of kinematically redundant mechanisms and KEGA 008TUKE-4/2024 Implementation of machine learning methods in the teaching of industrial automation and robotics.

REFERENCES

[1] P. J. Sincak, E. Prada, M. Lubica, R. Mykhailyshyn, M. Varga, T. Merva, and I. Virgala, "Sensing of Continuum Robots: A Review," *Sensors*, vol. 24, no. 4, p. 1311, Jan. 2024. [Online]. Available: https://www.mdpi.com/1424-8220/24/4/1311

[2] X. You, Y. Zhang, X. Chen, X. Liu, Z. Wang, H. Jiang, and X. Chen, "Model-free control for soft manipulators based on reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 2909–2915, iSSN: 2153-0866. [Online]. Available: https://ieeexplore.ieee.org/document/8206123

[3] S. Satheeshbabu, N. K. Uppalapati, G. Chowdhary, and G. Krishnan, “Open Loop Position Control of Soft Continuum Arm Using Deep Reinforcement Learning,” in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 5133–5139, iSSN: 2577-087X. [Online]. Available: https://ieeexplore.ieee.org/document/8793653

[4] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, “Model-Based Reinforcement Learning for Closed-Loop Dynamic Control of Soft Robotic Manipulators,” *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 124–134, Feb. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8531756

[5] A. Centurelli, L. Arleo, A. Rizzo, S. Tolu, C. Laschi, and E. Falotico, “Closed-Loop Dynamic Control of a Soft Manipulator Using Deep Reinforcement Learning,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4741–4748, Apr. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9697385

[6] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, “Trust Region Policy Optimization,” Tech. Rep., Apr. 2017, arXiv:1502.05477 [cs] type: article. [Online]. Available: http://arxiv.org/abs/1502.05477

[7] M. Giorelli, F. Renda, G. Ferri, and C. Laschi, “A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nov. 2013, pp. 5033–5039, iSSN: 2153-0866. [Online]. Available: https://ieeexplore.ieee.org/document/6697084

[8] J. M. Bern, Y. Schnider, P. Banzet, N. Kumar, and S. Coros, “Soft Robot Control With a Learned Differentiable Model,” in *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, May 2020, pp. 417–423. [Online]. Available: https://ieeexplore.ieee.org/document/9116011

[9] 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*, vol. 13, no. 3, p. e1774, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rcs.1774

[10] X. Wang, Y. Li, and K.-W. Kwok, “A Survey for Machine Learning-Based Control of Continuum Robots,” *Frontiers in Robotics and AI*, vol. 8, Oct. 2021, publisher: Frontiers. [Online]. Available: https://www.frontiersin.org/journals/robotics-and-ai/articles/10.3389/frobt.2021.730330/full

[11] M. Varga, P. J. Sinčák, T. Merva, P. Marcinko, L. Miková, E. Prada, M. Schnitzer, and I. Virgala, “Design and analyses of tendon-driven continuum robots for minimally invasive surgery,” *IFAC-PapersOnLine*, vol. 58, no. 9, pp. 25–30, Jan. 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2405896324004555