A study on position control of continuum arm using MAML(Model-Agnostic Meta-Learning) for adapting changing conditions

Alok Ranjan Sahoo, *Member, IEEE* and Pavan Chakraborty, *Member, IEEE*

Abstract—Non linear deformation of spring based continuum manipulators cause difficulty in predicting the position of the tip. When we put different tools on the tip for various purposes, the difficulty further increases. Model less control of the manipulator has shown great success in tip positioning of these types of manipulators. One of the major drawbacks of model less control is the requirement of a large data set and time. Hence, this paper studies the effect of the implementation of MAML(Model-Agnostic Meta-Learning) for fast adaptation of different offset conditions. The effects have been studied in the simulation environment and on the real prototype. The continuum arm used for the experimentation is a tendon based non constant curvature spring based manipulator. An average error of 0.03m has been achieved on the prototype. MAML was successful in bringing down the relative tip positioning error of the manipulator from 7.02% to 1.55% in the simulation environment. It also showed success in bringing down the relative tip positioning error from 11.06% to 4.09% on real prototype. We also studied the effectiveness of the same in trajectory following.

Index Terms—**Keywords:** Artificial neural network, Continuum robot, Model-less control

I. INTRODUCTION

Conventional manipulators with rigid links and joints are highly successful in industrial floors and choreographed environments. High precision and relatively easier manipulation are their strengths. However, manipulation becomes difficult in cluttered environment. It lacks the required degrees of freedom to navigate through it. Conventional manipulators are also not safe to work along side human beings due to their rigid links [1].

If we look in to the nature, we can find that snakes [2]–[4], tentacles [5]–[7], elephant trunks [8]–[11] can pass through cluttered environments to reach their target. Their bodies are so flexible that they can bend in any direction at any point of their body. This flexibility helps them navigate through any constrained environment with minimal effect on the environment. If we can mimic them properly, we can have efficient manipulators for search and rescue operations.

Most of the continuum manipulators had uniform backbone [12], [13]. The reason behind it was that the mathematical modelling of uniform backbones are relatively easy. However, they failed to mimic the elephant trunks and tentacles properly.

Alok R. Sahoo, is with Robotics and Machine intelligence lab, Department of IT, IIIT Allahabad, India(e-mail: aloksh90@ieee.org)

Pavan Chakraborty is with Robotics and Machine intelligence lab, Department of IT, IIIT Allahabad, India(e-mail: pavan@iiita.ac.in).

Whenever one try to bend the tip section whole body up to the proximal sections bend in that direction. Hence, the robot fail to achieve the purpose of using soft robot. To address the problem, some tapered designs [10], [14], [15] were developed. The mathematical formulation of their kinematics are quite difficult. Most of them used constant curvature model [16], [17]. It reduced the complexity but the accuracy deteriorated when the robot has variable backbone and external force is significant [18]. Piece-wise constant curvature(PCC) was introduced to counter this problem [10], [19]. Mode shape function [20], [21], euler spiral methods [22], pythagorean hodograph methods [23] were also used. Considering the mechanics of the model, mechanics modelling [24], [25], FEM based approaches [26], Cosserat rod approaches [27], [28], euler-beam mechanics based approaches [29], lumped system modeling [30], [31], absolute nodal coordinate formulation approaches (ANCF) have been used [32]. However, most of them are computationally expensive and relatively slow.

Hence, model less controllers were introduced. It was shown that model less control can achieve higher accuracy in positioning [33] the manipulator. Further, researchers used goal babbling technique [34], RNN based control [35], [36], reinforcement learning [37], deep reinforcement learning [38] for it. However, these models suffer from following drawbacks. They require high amount of data and long period of time to

get trained. A small amount of changes in the characteristics of the manipulator can lead to failure in achieving the accuracy. The model needs to be trained from the scratch for the new conditions. Some model less algorithms [38] have shown good performance in noisy conditions or slightly changed conditions. However, the performance decreases as the magnitude of variation increases. They don't have a method to bring back the original accuracy. Satheeshbabu et al. [38] have shown the effect of increase in payload on the tip positioning. This affects the capability of the model to be used in real life. Some operations need the tool placed on the tip to be changed frequently. We need our model to adapt to the new condition quickly without any significant change in the accuracy. Meta learning has the potential to provide a solution to that.

1) *What is meta learning??*: Humans can generalize things after a few similar experiences. Players can adapt to changes in their instruments after a few practices. Similarly, our manipulator should be intelligent enough to adapt to changes in the tip conditions. We know deep learning based controls are highly successful. Obviously, we would not like to train our model from scratch to get similar accuracy in changed conditions.

Hence, meta learning provides a paradigm where we don't need to train a model from scratch. The model could be trained with different sets of similar tasks. It uses this experience to improve the performance in any future task. This strategy is called "learning to learn". This is similar to how humans learn from their experiences. It also improves performance for a single task with multiple episodes.

The base learner or inner layer tries to do the basic task i.e classification or regression. The meta learner or the outer layer tries to update the base learner to improve the outer objective. The outer objective could be task generalization performance or learning time [39].

2) *Why MAML??*: The main purpose of MAML is that it can solve a new task with a very small number of training samples [40]. It is applicable to any model which is trained with gradient descent. It can be applied to classification, regression, reinforcement learning etc. It also requires a small number of gradient steps for good generalization of that task. Therefore, it has the potential to adapt to the changes in the manipulator within a very short span of time and a very small number of training data after the changes.

A. Problem definition

For using continuum manipulators in search and rescue operations or grasping, we need a kinematic model which can position the tip of the manipulator in desired position despite of the change in conditions due to grasping of a body or change in tool on the tip. Model based controllers fail measurably to tackle the changing conditions. MAML shows promising results in adapting to the changes with a very small updated training data set. Hence, in this paper we are trying to study the efficiency of MAML to adapt to the changing conditions. We have first tested it on the kinematic model of the manipulator from the previous work of the author [41]. Then, we have tested it on the prototype.

Fig. 1: Actuatorspace to task space

TABLE I: Nomenclature

Symbol	Meaning
κ	curvature of the arc
Φ	angle of plane containing curvature
l	arc length

II. FORMULATION OF LEARNING ARCHITECTURE

To model kinematics of a robot, one has to pass through two steps. First, there should be a mapping between actuator space (cable length/pneumatic force) to configuration space (κ, ϕ, l). This mapping will be unique for any individual robot. Hence, it is called robot specific. Then, one can use its geometry or DH parameters to find the mapping with the task space (tip position). This approach is generally followed to find the forward kinematics. Similarly, for inverse kinematics, one has to go backwards from task space to actuator space. In our approach, we are going to find the mapping directly from task space to actuator space (figure 1). We followed the model by Thuruthel et. al [42] for our work.

Let us consider a function ψ , which maps the actuator space to task space.

$$\psi(a) = p \quad (1)$$

Here, $a \in R^m$, is the actuator space and $p \in R^n$ is the task space. We know, $m < n$ for a redundant manipulator. Hence, direct inversion of it is not possible. To simplify it, we can linearize it for a_0 . We can use the jacobian matrix to see the transformation of actuator velocity to end effector velocity.

$$\dot{p} = J(a_0)\Delta a \quad (2)$$

$$J(a_i)a_{i+1} \approx p_{i+1} - \psi(a_i) + J(a_i)a_i \quad (3)$$

Here, a_i and a_{i+1} corresponds to the current and next position of the actuator respectively. Similarly, p_i and p_{i+1} corresponds to the current and next position of the end effector respectively. From this, we can have a position level controller. Now, we can train our MAML model to give a mapping $(p_{i+1}, a_i) \rightarrow a_{i+1}$. However, we know for search and rescue operations we will have to deal with constrained environment. Hence, a feedback of the current position of the tip is essential. Thuruthel et. al [42] have also shown that error percentage in closed loop controller is significantly less. Therefore, we will use a closed loop model for mapping $(p_{i+1}, a_i, p_i) \rightarrow a_{i+1}$. However, our model will explore the whole task space without using the concept of generalization. We will use data for whole

Fig. 2: Data collection in simulation

Fig. 3: Kinematic controller

range of actuation of each actuator. $q_{i+1} - q_i$ can be of any value from minimum actuatable length to maximum actuatable length.

A. Neural network architecture

The neural net consists of 4 hidden layers each of size 128. The input layer has inputs in the form of target tip position, current tip position and current actuator position. Hence, we have 10 nodes in the input layer. Similarly, the output layer has 4 nodes for target actuator positions. The neural network uses sigmoid transfer function and ReLU as activation function. Adam optimizer is used as the optimizer for gradient descent. α and β (learning rates) are kept at 0.01. The MSE(mean-square error) has been used as the loss function. The implementation has been done in pytorch. It was found that, small k and high number of epochs is more efficient than high number of k and small number of epochs for the case of continuum manipulator. Hence, we have taken $k=2$ in our case.

III. SIMULATION

A. Data collection in simulation

This simulation model is based on the mechanics modelling of the manipulator. The detailed modelling and characterization of the manipulator has been given in the previous work of the author [41]. The mechanics model takes the compression of each section in to consideration. The bending stiffness and axial stiffness of each section found from experimentations have been used for the simulation. As we already know the mapping of actuator space to task space (figure 2), we collected total 4000 sample points by random actuation of the actuators (motor babbling) using rand() function. We have actuated only the tip of the model with four actuators. To

TABLE II: Hyper parameters

α	β	k	epoch
0.01	0.01	2	2000

TABLE III: Results from simulation

Offset(inm)	Number of shots	Average error
0	0	0.0029
0.02	0	0.0149
0.02	10	0.0081
0.02	20	0.0049
0.02	30	0.0039
0.02	40	0.0027
0.04	0	0.0241
0.04	10	0.0186
0.04	20	0.0112
0.04	30	0.0081
0.04	40	0.0044
0.04	50	0.0027
0.06	0	0.0541
0.06	10	0.0442
0.06	20	0.0381
0.06	30	0.0301
0.06	40	0.0221
0.06	50	0.0198
0.06	60	0.0120
0.06	70	0.0122

ensure uniform data collection, data was collected for a single actuation of the manipulator restricting $a_{i+1} - a_i$ to 10 percent. This gave us 400 sample points. Next, 3600 sample points were collected from random actuation. Figure 3 shows the kinematic controller of it.

B. Experiments and results

The main purpose of this experimentation is to see the efficiency of MAML to adapt to the changes in the characteristics of the manipulator. Hence, we did 3 tests with offsets on the tip section. We have tested them with offset of 0.02 m, 0.04 m and 0.06 m respectively. For each change, we collected data for 10 shot difference until we get back the original accuracy.

From figure 4a, we see that even 0.02m offset can produce a maximum error of 0.03m. After 50 shots, it gave slightly better accuracy than the original accuracy (figure 4d). It follows same trend for 0.02m, 0.04m and 0.06m respectively. As we keep on increasing the offset, the requirement of number of shots keeps on increasing. From the table III, it can be noted that accuracy increases as we keep on increasing number of shots for lower offsets (0.02m, 0.04m). For higher offset (0.06m), increasing number of shots does not increase the accuracy after a certain accuracy is reached. From the histograms, it is evident that as number of shots increases maximum error decreases. From the simulation, it is proved that, MAML is highly successful in adopting small changes. It is partially effective for higher changes. We could achieve the accuracy with only 3 percent of the original data. Now, we need to see its efficiency in real world.

IV. EXPERIMENTAL SETUP

The manipulator is a 4 section double spring based tapered continuum manipulator. Total length of the manipulator is

Fig. 4: Histogram of the errors for 0.02m offset

Fig. 5: Histogram of the errors for 0.04m offset

0.77m. Length of each section is 0.19m. Each section is being actuated by 4 wires placed at 90 degrees offset. The back bone of the manipulator is a spring with continuous tapering. A PVC skin(second spring) is placed outside the internal spring back bone. This skin helps in routing the wire as well as providing higher torque. CAD diagram of the manipulator has been provided in the figure 8. Each stage have 4 actuators placed at the base plate at 90 degrees offset to each other. The motors are used are 12v dc geared motor with encoders. The encoders have 80160 counts per revolution. Such high precision encoders help us control small movement of the tip. Arduino mega has been used as the controller for it. For position feedback, we have used kinect v2. Complete experimental setup has been given in the figure 7. A slight modification has been done on the pvc skin of the manipulator. We have used a softer pvc skin on the tip section. This was done to make the tip section more flexible. The change in characteristics for the same has been incorporated in the simulation model.

A. Data collection from prototype

For data collection, first we discretized the actuator space with respected to the encoder positions. A minimum difference of 50000 counts between current encoder value and next encoder value was maintained. First, each actuator was given

a next actuator value of +200000 or -200000 counts. It was continued till the maximum range of the actuator. The maximum range was set to be 2000000 counts from zero position. The same procedure was followed for all 4 actuators. Then, actuator commands were given in couples following the aforesaid rules. This gave us 400 samples. Now random actuator targets were given using rand() function. A total of 4000 samples were collected from the prototype similar to simulation. Here, we have used different offsets to test the efficiency of MAML. It should be noted that placing different offsets also changes the dynamics of the manipulator. Hence, it would pose difficulty for the algorithm to adapt to the changes compared to the simulation environment.

B. Use of kinect V2 for position feedback

We used kinect v2 for track the tip of the position. A yellow color strip was placed on the tip of the manipulator. The work space and the background had no yellow color. Color thresholding was done to detect the yellow pixels. Centroid of the area was found. Corresponding XYZ co-ordinates were found using the kinect toolbox from pylibfreenect2.

V. EXPERIMENTS AND RESULTS

A. Random point target Experimentation

We have used offsets of 0.02m, 0.04m, 0.06m for testing. We gave 40 random points from the updated task space as

Fig. 6: Histogram of the errors for 0.06m offset

the target. Then, we found out the average tip positioning error and histogram for each case. It should be noted that the neural network is not getting trained during testing. These experiments are being conducted to find out minimum number of shots required to reach the original accuracy.

From the figure 9, it is evident that the average of the manipulator without any change of conditions is 0.0302m. The reason behind less accuracy may be due to compression of the sections. From the table IV, we see that the offset of 0.02m, 0.04m, 0.06m required 30, 40, 60 shots respectively. These are comparatively less number of shots required to get back to the default accuracy. This tells us that very high accuracy requires higher number of shots than lower accuracy. We see increase in number of shots also increase accuracy of the manipulator. For offset of 0.06m, we find the accuracy remains nearly same for 50 and 60 shots. From the histograms, it is evident

Fig. 7: Experimental setup

Fig. 8: CAD diagram of the manipulator

Fig. 9: Histogram of the errors with zero offset

that as number of shots increases maximum error decreases. Maximum number of times we find error of smaller magnitude as number of shots increases. We could achieve the original accuracy with only 3 percent of the original data in the real prototype as well. These results are similar to the results found in the simulation.

B. Trajectory following

In this experiment, we tried to find the capability of the manipulator to a trajectory following task. Here, we took two random points. Distance between the two points was 0.2872m. The difference in x direction, y direction and z direction were 0.20m, 0.20m and 0.05m respectively. We had given 19 control points in between them. Each control point was given as input one after another. We waited for 3s after each actuation of the

Fig. 10: Histogram of the errors for 0.02m offset

Fig. 11: Histogram of the errors for 0.04m offset

TABLE IV: Results from prototype

Offset(inm)	Numberofshots	Averageerror
0	0	0.0302
0.02	0	0.0440
0.02	10	0.0385
0.02	20	0.0350
0.02	30	0.0297
0.04	0	0.0601
0.04	10	0.0514
0.04	20	0.0455
0.04	30	0.0365
0.04	40	0.0311
0.06	0	0.0852
0.06	10	0.0755
0.06	20	0.0512
0.06	30	0.0412
0.06	40	0.0373
0.06	50	0.0322
0.06	60	0.0315

manipulator to record the tip position. Hence, we could avoid the effect of vibration. The same test was conducted twice. In the first run, it was done with an offset of 0.04m. Then, it was updated with 40 shots. We performed the same test again with the updated neural network.

From figure 15a, we see that average error goes up to 0.065m and maximum error touches up to 0.13m. After 40 shots, the result improved significantly. The average error and maximum error came down to 0.035m and 0.06m respectively. The average error is higher than the result we got in random point experimentation.

VI. CONCLUSION

From the above experiments, it can be concluded that MAML can be successfully implemented to adapt to the changing conditions. MAML was effective to bring down the relative tip positioning error of the manipulator from 7.02% to 1.55% in simulation environment. It also showed success to bring down the relative tip positioning error from 11.06% to 4.09% on real prototype. We had to use only 3 percent of the original training data to get back the original accuracy. When the magnitude of disturbance is small, then MAML can bring back the accuracy efficiently. Further increase in number of shots also show better accuracy than the original accuracy. When the magnitude of the disturbance is relatively large (above 10 percent of the manipulator length), then the algorithm fails to bring back the original accuracy. However, up to a certain number of shots the accuracy will increase but after a certain number of shots, increasing number of shots won't have any positive effect in increasing the accuracy.

VII. REFERENCE

REFERENCES

- [1] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.
- [2] D. Shen, Q. Zhang, C. Wang, X. Wang, and M. Tian, "Design and analysis of a snake-inspired crawling robot driven by alterable angle scales," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3744–3751, 2021.

Fig. 12: Histogram of the errors for 0.06m offset

Fig. 13: Trajectory following task with 0.04m offset and zero shot

Fig. 14: Trajectory following task with 0.04m offset and 40 shots

(a) Error graph for trajectory following task with zero shot (b) Error graph for trajectory following task with 40 shots

Fig. 15: Error graphs

- [3] I. Virgala, M. Kelemen, E. Prada, M. Sukop, T. Kot, Z. Bobovský, M. Varga, and P. Ferenčík, "A snake robot for locomotion in a pipe using trapezium-like travelling wave," *Mechanism and Machine Theory*, vol. 158, p. 104221, 2021.
- [4] K. Y. Pettersen, "Snake robots," *Annual Reviews in Control*, vol. 44, pp. 19–44, 2017.
- [5] J. Jørgensen, K. B. Bojesen, and E. Jochum, "Is a soft robot more "natural"? exploring the perception of soft robotics in human–robot interaction," *International Journal of Social Robotics*, pp. 1–19, 2021.
- [6] J. Frás, M. Macias, Y. Noh, and K. Althoefer, "Fluidical bending actuator designed for soft octopus robot tentacle," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 253–257, IEEE, 2018.
- [7] R. V. Martinez, J. L. Branch, C. R. Fish, L. Jin, R. F. Shepherd, R. M. Nunes, Z. Suo, and G. M. Whitesides, "Robotic tentacles with three-dimensional mobility based on flexible elastomers," *Advanced materials*, vol. 25, no. 2, pp. 205–212, 2013.
- [8] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *Journal of robotic systems*, vol. 20, no. 2, pp. 45–63, 2003.
- [9] K. Wu and G. Zheng, "Fem-based gain-scheduling control of a soft trunk robot," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3081–3088, 2021.
- [10] T. Mahl, A. Hildebrandt, and O. Sawodny, "A variable curvature continuum kinematics for kinematic control of the bionic handling assistant," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 935–949, 2014.
- [11] J. Yang, E. P. Pitarch, J. Potratz, S. Beck, and K. Abdel-Malek, "Synthesis and analysis of a flexible elephant trunk robot," *Advanced Robotics*, vol. 20, no. 6, pp. 631–659, 2006.
- [12] J. Peng, W. Xu, T. Liu, H. Yuan, and B. Liang, "End-effector pose and arm-shape synchronous planning methods of a hyper-redundant manipulator for spacecraft repairing," *Mechanism and Machine Theory*, vol. 155, p. 104062, 2021.
- [13] G. Gao, H. Wang, J. Fan, Q. Xia, L. Li, and H. Ren, "Study on stretch-retractable single-section continuum manipulator," *Advanced Robotics*, vol. 33, no. 1, pp. 1–12, 2019.
- [14] M. M. Tonapi, I. S. Godage, A. Vijaykumar, and I. D. Walker, "A novel continuum robotic cable aimed at applications in space," *Advanced Robotics*, vol. 29, no. 13, pp. 861–875, 2015.
- [15] X. Dong, D. Axinte, D. Palmer, S. Cobos, M. Raffles, A. Rabani, and J. Kell, "Development of a slender continuum robotic system for on-wing inspection/repair of gas turbine engines," *Robotics and Computer-Integrated Manufacturing*, vol. 44, pp. 218–229, 2017.
- [16] R. J. Webster III and B. A. Jones, "Design and kinematic modeling

- of constant curvature continuum robots: A review,” *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.
- [17] M. Rolf and J. J. Steil, “Constant curvature continuum kinematics as fast approximate model for the bionic handling assistant,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3440–3446, IEEE, 2012.
- [18] Y. Shapiro, A. Wolf, and K. Gabor, “Bi-bellows: Pneumatic bending actuator,” *Sensors and Actuators A: Physical*, vol. 167, no. 2, pp. 484–494, 2011.
- [19] H. Wang, B. Yang, Y. Liu, W. Chen, X. Liang, and R. Pfeifer, “Visual servoing of soft robot manipulator in constrained environments as fast adaptive controller,” *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 41–50, 2016.
- [20] I. S. Godage, E. Guglielmino, D. T. Branson, G. A. Medrano-Cerda, and D. G. Caldwell, “Novel modal approach for kinematics of multi-section continuum arms,” in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1093–1098, IEEE, 2011.
- [21] I. S. Godage, G. A. Medrano-Cerda, D. T. Branson, E. Guglielmino, and D. G. Caldwell, “Modal kinematics for multisection continuum arms,” *Bioinspiration & biomimetics*, vol. 10, no. 3, p. 035002, 2015.
- [22] P. S. Gonthina, A. D. Kapadia, I. S. Godage, and I. D. Walker, “Modeling variable curvature parallel continuum robots using euler curves,” in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 1679–1685, IEEE, 2019.
- [23] I. Singh, Y. Amara, A. Melingui, P. Mani Pathak, and R. Merzouki, “Modeling of continuum manipulators using pythagorean hodograph curves,” *Soft robotics*, vol. 5, no. 4, pp. 425–442, 2018.
- [24] D. B. Camarillo, C. F. Milne, C. R. Carlson, M. R. Zinn, and J. K. Salisbury, “Mechanics modeling of tendon-driven continuum manipulators,” *IEEE transactions on robotics*, vol. 24, no. 6, pp. 1262–1273, 2008.
- [25] D. B. Camarillo, C. R. Carlson, and J. K. Salisbury, “Configuration tracking for continuum manipulators with coupled tendon drive,” *IEEE transactions on robotics*, vol. 25, no. 4, pp. 798–808, 2009.
- [26] T. M. Bieze, F. Largilliere, A. Kruszewski, Z. Zhang, R. Merzouki, and C. Duriez, “Finite element method-based kinematics and closed-loop control of soft, continuum manipulators,” *Soft robotics*, vol. 5, no. 3, pp. 348–364, 2018.
- [27] J. Till, C. E. Bryson, S. Chung, A. Orekhov, and D. C. Rucker, “Efficient computation of multiple coupled cosserat rod models for real-time simulation and control of parallel continuum manipulators,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5067–5074, IEEE, 2015.
- [28] H. B. Gilbert and I. S. Godage, “Validation of an extensible rod model for soft continuum manipulators,” in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pp. 711–716, IEEE, 2019.
- [29] G. Gao, H. Wang, Q. Xia, M. Song, and H. Ren, “Study on the load capacity of a single-section continuum manipulator,” *Mechanism and Machine Theory*, vol. 104, pp. 313–326, 2016.
- [30] I. S. Godage, R. Wirz, I. D. Walker, and R. J. Webster III, “Accurate and efficient dynamics for variable-length continuum arms: A center of gravity approach,” *Soft Robotics*, vol. 2, no. 3, pp. 96–106, 2015.
- [31] S. Sadati, A. Shiva, S. E. Naghibi, D. C. Rucker, L. Renson, C. Bergeles, K. Althoefer, T. Nanayakkara, H. Hauser, and I. D. Walker, “Reduced order vs. discretized lumped system models with absolute and relative states for continuum manipulators,” in *Robotics: Science and Systems*, p. 10, 2019.
- [32] X. Huang, J. Zou, and G. Gu, “Kinematic modeling and control of variable curvature soft continuum robots,” *IEEE/ASME Transactions on Mechatronics*, 2021.
- [33] M. Giorrelli, 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*, vol. 31, no. 4, pp. 823–834, 2015.
- [34] M. Rolf, J. J. Steil, and M. Gienger, “Online goal babbling for rapid bootstrapping of inverse models in high dimensions,” in *2011 IEEE International Conference on Development and Learning (ICDL)*, vol. 2, pp. 1–8, IEEE, 2011.
- [35] T. G. Thuruthel, E. Falotico, M. Manti, and C. Laschi, “Stable open loop control of soft robotic manipulators,” *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1292–1298, 2018.
- [36] T. G. Thuruthel, E. Falotico, F. Renda, and C. Laschi, “Learning dynamic models for open loop predictive control of soft robotic manipulators,” *Bioinspiration & biomimetics*, vol. 12, no. 6, p. 066003, 2017.
- [37] 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)*, pp. 2909–2915, IEEE, 2017.
- [38] 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)*, pp. 5133–5139, IEEE, 2019.
- [39] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, “Meta-learning in neural networks: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.
- [40] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proceedings of the 34th International Conference on Machine Learning* (D. Precup and Y. W. Teh, eds.), vol. 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135, PMLR, 06–11 Aug 2017.
- [41] A. Sahoo and P. Chakraborty, “Development & analysis of a bio-inspired wire-driven variable stiffness double spring based tapered multi-section flexible robot,” *Industrial Robot: the international journal of robotics research and application*, (Accepted)2021. (in press).
- [42] T. George Thuruthel, E. Falotico, M. Manti, A. Pratesi, M. Cianchetti, and C. Laschi, “Learning closed loop kinematic controllers for continuum manipulators in unstructured environments,” *Soft robotics*, vol. 4, no. 3, pp. 285–296, 2017.

ALOK RANJAN SAHOO received the B.Tech degree in Mechanical engineering. He received the M.Tech degree in Robotics from IIIT Allahabad. He is currently pursuing PhD in Robotics from IIIT Allahabad. His research interests are Robotics, machine learning and computer vision.

PAVAN CHAKRABORTY received the Ph.D. degree from the Indian Institute of Astrophysics, India, in 2001. He is currently an Associate Professor with the Indian Institute of Information Technology Allahabad. His research interests include human gait analysis, human prosthetic, bio-metrics, image processing, graphics and visual computing, graphical projections, robotics and instrumentation, real time simulation, high performance computing (HPC), artificial life simulation and intelligence, and human-computer

interaction.