

Model-less Active Compliance for Continuum Robots using Recurrent Neural Networks

David Jakes, Zongyuan Ge, and Liao Wu, *Member, IEEE*

Abstract—Endowing continuum robots with compliance while it interacts with the internal environment of the human body is essential to prevent damage to the robot and the surrounding tissues. Compared with passive compliance, active compliance has the advantages in terms of increasing the force transmission ability and improving safety with monitored force output. Previous studies have demonstrated that active compliance can be achieved based on a complex model of the mechanics combined with a traditional machine learning technique such as a support vector machine. This paper proposes a recurrent neural network (RNN) based approach that avoids the complexity of modeling while capturing nonlinear factors such as hysteresis, friction and delay of the electronics that are not easy to model. The approach is tested on a 3-tendon single-segment continuum robot with force sensors on each cable. Experiments are conducted to demonstrate that the continuum robot with an RNN based feed-forward controller is capable of responding to external forces quickly and entering an unknown environment compliantly.

I. INTRODUCTION

Continuum robots have been identified as a class of robots that are ideal for minimally invasive surgery [1]. Applications of these robots span from neurosurgeries [2], trans-oral and trans-nasal procedures [3]–[5], intracardiac procedures [6], to orthopaedic surgeries [7]. Compared to their discrete rigid counterparts, the continuum robots can be made very slim due to the novel actuation methods used that leverage the deformation of the material that constructs the body of the robot. When used in surgical applications, this means the robot can be inserted into the human body through a small incision/orifice and a narrow passage, being minimally invasive to the patient.

Safety is one of the key issues to address in surgeries using continuum robots. Endowing a robot with compliance while it interacts with the environment is essential to prevent damage to the robot and the surrounding tissues. Passive compliance can be realized by the nature of many mechanisms for the construction of continuum robots like the cable-driven mechanism [8]. However, this approach limits the force transmission ability for tasks that are force demanding, such as tissue dissection and retraction, and increases the

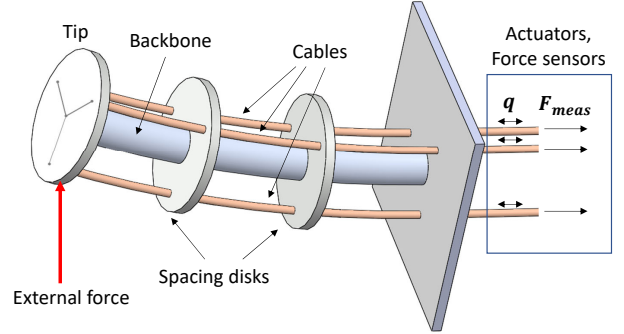


Fig. 1. A 3-tendon single-segment continuum robot with force sensors on each cable. Active compliance allows the robot to move compliantly by adjusting the actuating cables when experiencing external forces along the segment.

uncertainty of control due to the unmonitored flexibility while the robot is operating in the human body.

Recently, active compliant motion control for continuum robots has been studied [9], [10]. By measuring joint level actuation forces and relating them to the generalized forces in the configuration space that are mapped from the external wrenches, it was demonstrated that the active compliance can be achieved without expensive sensors deployed along the distal part of the robot. However, this was done based on a complex nonlinear model of the kinematics and mechanics of the structure. In addition to the complexity in analysis and implementation, there are also many factors that cannot be easily modeled, such as the hysteresis of cables, backlash in transmission, friction, characteristics of the electronic components, etc.

Machine learning techniques could be used to bypass complex modeling and tackle those nonlinear factors. In [10], a support vector machine (SVM) was used to capture the friction and correct the uncertainties in the complex modeling. However, the method still relied on elaborated analysis and modeling of the mechanics of the continuum robot. We argue that a model-less compliant motion control strategy could be achieved in the machine learning framework by directly relating the actuation with the force measurement. In addition, instead of using an SVM, the neural network (NN) based techniques, representing the-state-of-the-art, could produce better results in terms of capturing the nonlinear characteristics in the compliant motion control. In particular, the recurrent neural network (RNN) [11], [12] is a good tool to deal with the process-dependent factors such as hysteresis of cables and delay of electronic components.

D. Jakes is with the School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, Australia.

Z. Ge is with eResearch Centre, Nvidia AI technology centre, Airdoc Research, Monash University, Melbourne, Australia.

L. Wu is with the School of Mechanical and Manufacturing Engineering, University of New South Wales, Sydney, Australia. dr.liao.wu@ieee.org

A multimedia attachment demonstrating the experiments can be found at <https://youtu.be/5iW1QgeTBg>

This paper proposes a new method for compliant motion control of continuum robots by using an RNN. Using a 3-tendon single-segment continuum robot with force sensors on each cable (Fig. 1), an RNN is trained to predict the cable forces produced by backbone stiffness, friction and other factors from the actuator control signals during unloaded motion. Actuator positions are supplied to the RNN producing a feed-forward prediction to a simple motion controller which achieves compliance by directly controlling actuators to minimize the difference between predicted and measured tension. Comparison between RNN and other architectures is performed to show the superiority of RNN for this task. We validate the proposed approach on an experimental platform, demonstrating the effectiveness of a feed-forward RNN compliance controller which can bypass the complexity of nonlinear modeling and capture the effect of physical and electrical factors that are not modelable in continuum robots.

The rest of the paper is organized as follows. Sec. II describes the problem of active compliance and introduces the role of RNN in the framework. Sec. III gives a more detailed introduction to the RNN architecture used in this paper. Based on the above, a compliant motion control algorithm is presented in Sec. IV. Experimental validation of the method is elaborated in Sec. V to show the effectiveness. The paper is discussed and concluded in Sec. VI and VII, respectively.

II. PROBLEM DESCRIPTION

The design of continuum robots featuring an elastic backbone driven by three or more evenly-spaced tendons routed along the periphery of the tube is both prominent and well-validated [13]. A key control requirement is that tendon tension remains within safe ranges as excessive tension may cause tendons to snap, while slack can invalidate modeling assumptions and sensor measurements causing backlash and potential structural damage. The analysis in this section is performed on a single-segment robot with three tendons. Force sensors are located at the connection point between tendons and actuators to measure individual tendon tension.

Active compliance control requires the movement of actuators \mathbf{q} in a way that minimizes and therefore complies with measured external forces \mathbf{F}_{ext} . Assuming backbone incompressibility, external forces applied to the tip of the robot are balanced by increased tension in the tendons providing the robot configuration holding force. Ref [9], [10] have demonstrated that interaction forces at the tip and along the length of the backbone can be reconstructed from measurements of tendon forces.

In addition to external forces, sensor measurements \mathbf{F}_{meas} also include internal forces \mathbf{F}_{int} caused by the robot's natural elastic deformation, gravity, friction and dynamics while in motion, i.e.,

$$\mathbf{F}_{meas} = \mathbf{F}_{int} + \mathbf{F}_{ext}. \quad (1)$$

\mathbf{F}_{int} has a fixed but unknown relationship to the actuator state space,

$$\mathbf{F}_{int} = f(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}, \dots). \quad (2)$$

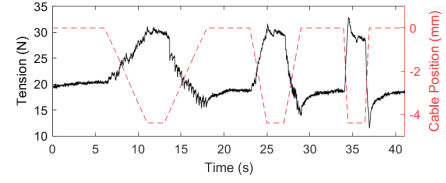


Fig. 2. Cable tension throughout linear motion. The first, second, and third waveform corresponds to cable motion speed at 1mm/s, 2.5mm/s, and 10mm/s, respectively.

In continuum robots, these governing equations are complex and highly nonlinear incurring significant modeling and computational effort.

The most prevalent model for static forces in continuum robots is the Cosserat rod theory [14] using an elastic energy formulation to derive forces and curvature from backbone stiffness and cable lengths. However, this model is purely static physics based and does not take effects accompanying dynamic motions into account.

Actuating the cables for change in configuration involves acceleration, backbone momentum and the conversion of elastic to kinetic energy, producing forces which are translated across the tendons. Typically, these forces are ignored under small velocities with the assumption that small mass and velocity produces negligible momentum and the elastic restoration rate is sufficiently fast.

Directional hysteresis, where transient forces depend on the direction of motion due to the microstructure of the elastic backbone, has been addressed using piece-wise curves selected based on the sign of the velocity [9].

These deviations from the static model can be observed in Fig. 2. At slow speeds, the cable tension is dominated by static elastic stiffness with a small directional hysteresis component adding an offset when increasing versus decreasing tension. As movement speed increases, the deviation from the static model increases and an overshoot-with-restitution curve is observed accompanying sudden changes in velocity. This is mainly due to effects such as friction, slip, and other uncertainties that are difficult to capture with traditional modeling.

In the past, machine learning has been used to augment kinematic models, achieving desired control stability/accuracy using feed-forward compensation for friction and other model uncertainties [10]. Given the complexity of kinematic modeling and partial application of machine learning techniques already in use, there is motivation for replacing the combination of techniques with a single machine learning algorithm producing feed-forward force estimates.

In this paper, we propose to use an RNN for the formulation of (2). The input matrix is formed by concatenating the actuator control vector \mathbf{q} over a period of recent discrete timesteps,

$$\mathbf{Q} = [\mathbf{q}_t \quad \mathbf{q}_{t-\Delta t} \quad \dots \quad \mathbf{q}_{t-n\Delta t}], \quad (3)$$

and the network learns the function,

$$\mathbf{F}_{int} = rnn(\mathbf{Q}). \quad (4)$$

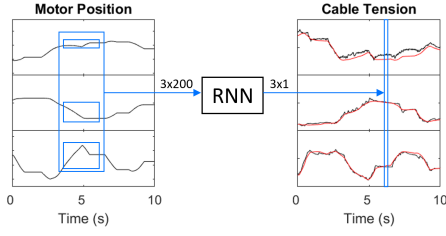


Fig. 3. An RNN based tension predictor is supplied with $n = 200$ discrete time points of control signal for each actuator, and predicts the tension in each cable at the next timestep $t + 1$. Recorded tension training data (black) is shown with prediction tension (red).

The concept of using RNN here is to use the continuous control input signal to predict the measured tension resulting from electrical and physical statics and dynamics. Fig. 3 shows how RNN is employed in this work for tension prediction. A detailed description of RNN is presented in the next section.

III. RECURRENT NEURAL NETWORK

Since our aim is to characterize the high-order correlation between sensor measurements and actuator state space, we implement RNN with especially long short term memory (LSTM) neurons as recurrent neurons that can effectively model the long-term temporal dependency in a continuous signal space. RNN is a neural network that uses internal hidden states to model the dynamic behaviour of sequences with arbitrary nonlinear functions. In this task, the hidden states may store relevant long-term temporal dependencies of velocity, acceleration, and other short-term memory properties such as hysteresis, delays and restitution. The full formulation of a typical LSTM is defined as follows,

$$\begin{aligned} i_t &= \sigma(W_{q,i} \mathbf{q}_t + W_{h,i} h_{t-1}) \\ f_t &= \sigma(W_{q,f} \mathbf{q}_t + W_{h,f} h_{t-1}) \\ o_t &= \sigma(W_{q,o} \mathbf{q}_t + W_{h,o} h_{t-1}) \\ g_t &= \phi(W_{q,c} \mathbf{q}_t + W_{h,c} h_{t-1}) \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \phi(c_t) \end{aligned} \quad (5)$$

In addition to RNN that has the hidden state h with N hidden units, LSTM has five extra gates compared to an RNN: an input gate i and input modulation gate g to indicate the input information ratio; a forget gate f and memory gate c to control the memory of the current state; an output gate o to control the level of output to the state. σ is the *sigmoid* nonlinearity which takes the input and output real-valued inputs to a $[0, 1]$ range while ϕ is denoted as *hyperbolic tangent* nonlinearity function similar to *sigmoid* but with range of $[-1, 1]$. These components make LSTM easier to optimize during training progress by back-propagating gradients further and thus enable the model to learn extremely long-term and complex dependency.

In this paper, we build the LSTM model by feeding input to the n LSTM cells ($n = 32, 64$) followed by a 32 neurons fully connected linear layer. One important issue of training

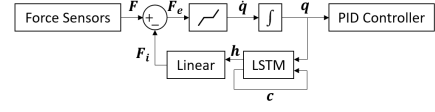


Fig. 4. Diagram of the RNN-based compliant motion controller in the actuator space.

an RNN model is to determine the input sequential length. We set the sequence length of 100 and 200 to validate the effectiveness of the model with respect to the time factor. By combining all input signals from three motors into a single hidden state, the network learns a holistic representation of the model, allowing common features to be combined with different weights to produce each of the three outputs. Training data is generated by recording tension measurements during unloaded motion using a rudimentary non-compliant control algorithm which traverses the available actuator space across a range of motion styles. More details about sample collection are described in Sec. V-C.

Training LSTM model can be achieved by using the mean squared error (MSE) loss on the three output of fully connected layer and employing a back-propagation algorithm with stochastic gradient descent (SGD). We employ a momentum rate of 0.9 and a learning rate of 0.005 for all experiments.

IV. COMPLIANT MOTION CONTROL

Using the feed-forward tension estimator provided by RNN, an external force component can be extracted from sensor measurements and a compliant motion controller can be constructed to minimize this value, as shown in Fig. 4. Unlike typical controllers, which transform forces into the pose space, and determine the actuator space motion required to travel towards the desired conforming pose, we neglect to model the robot kinematics, and present an algorithm operating entirely in the actuator space. The cables have a very simple actuator-force relationship, namely pulling a cable increases tension, and releasing a cable decreases tension. Thus we take the unloaded tension prediction from the neural network and subtract it from the sensor measurements to obtain the external force,

$$\mathbf{F}_{ext} = \mathbf{F}_{meas} - \mathbf{F}_{int}. \quad (6)$$

A deadband threshold λ is required to prevent erroneous motion due to prediction and sensor inaccuracy, and forces exceeding the deadband produce a control velocity $\dot{\mathbf{q}}$ with a hand-tuned proportionality constant β .

$$\dot{\mathbf{q}} = \begin{cases} 0 & -\lambda \leq \mathbf{F}_{ext} \leq \lambda \\ -\beta(\mathbf{F}_{ext} - \lambda \text{sign}(\mathbf{F}_{ext})) & \text{otherwise} \end{cases} \quad (7)$$

The threshold λ is selected based on the performance of the neural network on the validation set at approximately $1.25 \times$ mean error. The responsiveness constant β is tuned

to a suitable balance of sensitivity and stability. The performance of the controller will be evaluated in the next section.

V. EXPERIMENTAL VALIDATION

A. Experimental setup

Experiments were performed on a single segment platform with three braided steel wire tendons and a flexible backbone of reinforced plastic hose. Four tendon guide disks of radius 10 mm were spaced evenly along the length of the segment at 33 mm intervals. Cables were driven by standard metal gear servos with 13 kg-cm of holding torque on a drive wheel of radius 20 mm providing 63 mm of actuation distance, and a maximum safe operating force of 65 N. Tension sensing was provided by strain gauges glued to small tabs of aluminium inserted between two segments of cable, routed through a custom amplification board to the analog-digital converters of a microcontroller unit. Measurements were taken at 20 kHz and provided to the control loop at 100 Hz after passing through a first-order low pass filter defined as

$$y[n] = \frac{255}{256}y[n-1] + \frac{1}{256}x[n] \quad (8)$$

where x is the unfiltered tension reading and y is the filtered value feeding the control loop.

Typically, the use of low-cost components, filtering and external PID controllers would incur significant additional modeling cost for control loop design. However, the holistic neural network approach enables learning the combined system transfer function provided sufficient training data and input history.

B. Tip force estimation

As the incompressible backbone presents a curvy shape in the natural state, the majority of cable tension is balanced by the backbone and elastic energy, but differing cable tensions create a torque about the center of the tip segment producing a force in the plane of the tip segment. The ratio between cable tension, tip torque, and force in the tip plane is encapsulated in a couple constant α . Projecting into the tip plane using the orientation of cable connections produces the form

$$\begin{bmatrix} F_x \\ F_y \end{bmatrix} = \frac{\alpha}{2} \begin{bmatrix} 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \end{bmatrix} \quad (9)$$

where the constant α represents the ratio between the force in a cable and magnitude of force in the tip acting in the corresponding direction.

With the experimental setup shown in Fig. 5, the robot was moved to several different positions, and known weights were added to the tip in the form of 9 g dollar coins. The measured external tension in each cable was transformed into the tip plane using (9) producing Fig. 6, from which we conclude that the constant α equals approximately 1/3 in this system.

Note this step is just for interpretation of the experimental results in the following sections. It is not mandatory for the compliant motion control since the control is performed directly in the actuator space.

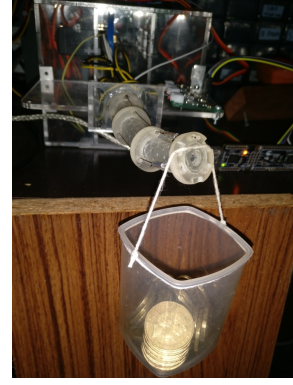


Fig. 5. Experimental setup for tip-cable force calibration. Known weights (coins) were applied to the tip at various orientations and positions.

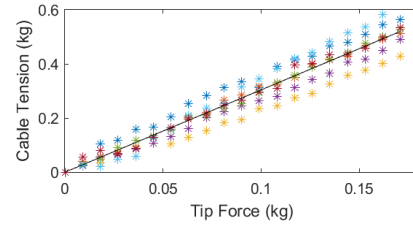


Fig. 6. Calibration of the relationship between tip force and cable tension. The horizontal axis is the measured tip force and the vertical axis is the force calculated by the sensed external tension in each cable. Six trials of measurement were performed at each interval. The results show that $\alpha = 1/3.04$ can be taken for the system.

C. Data collection for offline training

Collection of training data for offline training requires moving the robot in such a way as to explore the input space with sufficient variation to accurately capture the underlying relationships. We used an elementary model to allow smooth traversal of the task space, transforming the position of the three cables q_1, q_2, q_3 to a 2D position (x, y) and combined tension c using the vectors from the tip center to the cable attachment points in the tip plane,

$$\begin{bmatrix} x \\ y \\ c \end{bmatrix} = \begin{bmatrix} 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \\ 1 & -\frac{1}{2} & -\frac{1}{2} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}. \quad (10)$$

Moving a cable by one unit moves the position vector (x, y) by one unit in the corresponding direction. The constraint that all cables must remain within a given tension range, to prevent damage to cables, or tension falling to 0 and invalidating sensor relationships was enforced by learning a surface

$$c = f(x, y) \quad (11)$$

such that the tension in each cable was within the desired range. The surface f was explored by moving to new positions, and varying c to achieve optimal sensing tension in each cable, and recording the point (x, y, c) in a lookup table. Future movements sampled the surface using LOESS smoothing [15] and the exploration was continued until the suitable region of the x, y plane was filled to sufficient fidelity.

We note that the control model used for data collection does not need to be physically accurate, or complex, only topologically smooth, to allow reasoning about training motions.

In operation, we expect the robot to exhibit a range of motion styles, smooth curves, sudden changes in direction, and arbitrary starts and stops across a range of velocities. To collect representative training data, we created three parameters: velocity, jerkiness, and sleepiness. Velocity controls the rate at which we traverse the (x, y) surface, jerkiness controls the maximum angular rate of change in (x, y) , and sleepiness causes the sudden cessation of movement for small lengths of time. Target positions were randomly selected from (x, y) and changed whenever the robot entered a small radius of the current target. Every 20 seconds, a new set of velocity, jerkiness and sleepiness parameters were chosen. The motor control signals and corresponding recorded cable tensions were recorded at 100 Hz for six hours producing 2 million points of training data for the RNN. We then used eighty per cent of the generated data for training, with the remaining 20% kept for validation.

D. Comparison of NN architectures

In this section, we show the empirical performance of our deep sequential model LSTM with various hyper-parameter settings on the tension force prediction task. Furthermore, we also compared our RNN based LSTM model to a non-recurrent architecture convolutional neural network (CNN), which has outperformed LSTMs in neural machine translation [16]. CNN network is able to connect distant signals via shorter network paths than LSTMs. Although our task is very different from machine translation, it is worthwhile to test this theoretical argument empirically on another signal processing task and compare their differences.

CNNs are hierarchical based networks, and it captures local correlations in the signal. Unlike RNNs, CNN is easier and faster to train due to its architectural design. The main difference between RNNs based models and CNNs is the ability to capture the long-range dependencies. The receptive range of a CNN is computed with respect to the number of layers L and kernel size k . The longest context size is $L(k - 1)$. In this experiment we built a CNN with $L = 3$ and kernel size $k = 32, 64$. ReLU was attached to each convolution layer. Max-pool was employed on the feature map before connecting it to a fully-connected layer with 32 neurons.

Recurrent and non-recurrent architectures are hard to compare fairly because many factors may affect the performance. We trained both models with similar hyper-parameters (e.g. input sequence length, batch size, number of hidden states in the fully-connected layer) and techniques in PyTorch [17] except for those parameters specific with each architecture (e.g. number of epochs to converge).¹

Table I gives the performance of all the architectures, including experiments with different input sequential length

TABLE I
THE RESULTS OF DIFFERENT LSTM AND CNN MODELS ON
FORCE PREDICTION TASK.

Model	Sequential length	Mean err (kg)
<i>LSTM-32</i>	100/200	0.304/0.282
<i>LSTM-64</i>	100/200	0.300/ 0.258
<i>CNN-32</i>	100/200	0.287/0.295
<i>CNN-64</i>	100/200	0.304/0.270

and number of neurons or filters in the hidden layer. The evaluation metric being used here is mean error. Recurrent models LSTM distinctly outperform CNN models, showing the importance of capturing long-term dependency in force prediction. Moreover, the majority of LSTM and CNNs achieve lower validation error when given more hidden states. We attribute this observation to the model capacity, which makes the model fit better to the training dataset. For both models, sequential length plays an important role in capturing long-term dependency. This becomes more obvious when comparing the performance gain between *LSTM-64* (-0.42) and *CNN-64* (-0.34). This affirms our assumption that recurrent models are better long-term features extractors than CNNs, and thus are more suitable for the task discussed in this paper.

E. Responsiveness evaluation

With the RNN-based tension estimator validated in the last section, a compliant motion controller was implemented in the actuator space. All elements of the control loop, sensor measurements and the neural network were discretised at 100 Hz. Control was implemented in C# on the windows platform with accompanying UI, communicating with a slave microcontroller which supplied sensor readings and motor control input over low-latency (3 ms) serial.

The responsiveness of the controller was first evaluated by manually applying a force to the tip, as shown in Fig. 7. The RNN was able to predict the internal tension in each cable and thus the external tension was separated to drive the actuators. Fig. 8 shows the control system performance compliantly in response to forces applied to the tip of the robot. When the measured external force on a cable exceeded the deadband threshold, the cable moved rapidly to reduce the input force at a rate proportional to the magnitude of the input.

In order to further evaluate the response time and sensitivity of the controller, a wooden block was used to push the tip backwards and forwards (Fig. 9), stopping at a marked line and waiting for the controller to reduce the interaction force. Fig. 10 shows the mean tip force in response to 20 impulses of varying intensity. Individual responses were correlated using the moment the force on one cable exceeded the dead-band threshold. The controller rapidly moved to reduce the impulse in the first 0.2 s, and then approached the dead-band asymptotically.

¹The RNN based network was trained for 2000 epochs in 16 hours on a GTX 1070 with batches of 20 k points, While CNN based network could converge within 100 epochs in less than 3 hours training time.

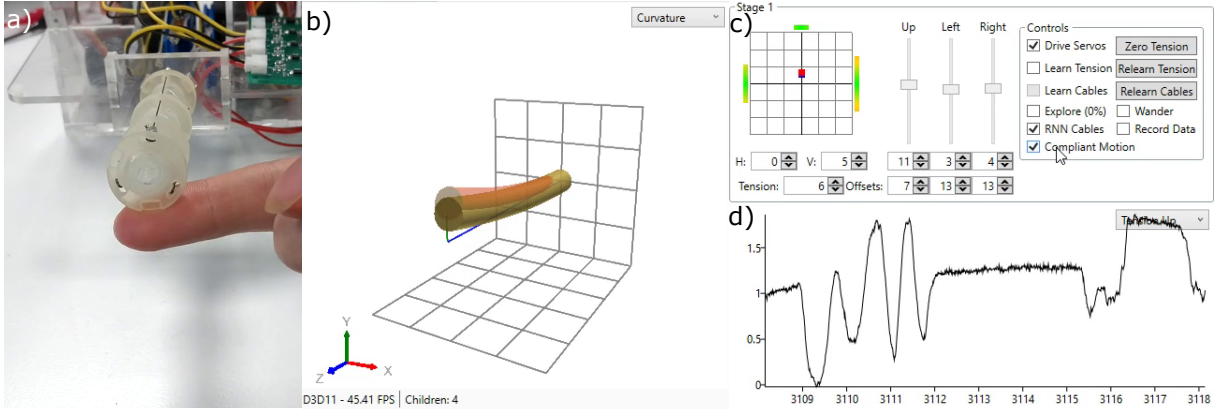


Fig. 7. Responsiveness evaluation. a) A force was applied to the tip of the robot. b) Estimation of the RNN. Yellow and brown parts represent the current state and estimated deformation based on cable tension measurement, respectively. c) Control panel. d) Cable tension measurement.

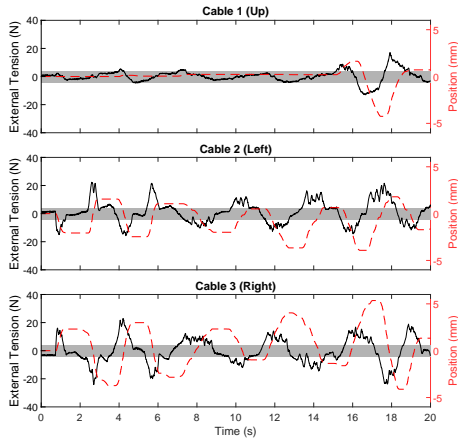


Fig. 8. Compliant response to measured external forces throughout various motions with deadband (gray).

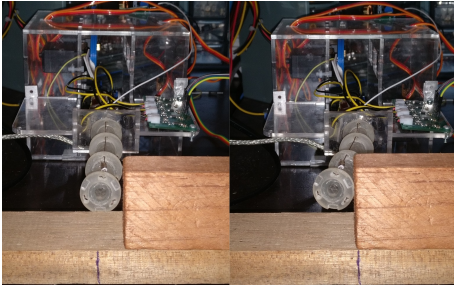


Fig. 9. Experimental setup for impulse response. The block is moved to the black line, applying a deflection to the robot tip.

F. Compliant insertion

To evaluate the controller performance when exploring unknown environments, the robot was repeatedly inserted into and retracted from a curved tube of radius 13 mm. Illustrated in Fig. 11, the robot successfully conformed to the tube shape, achieving a 90 degree bend with insertion, and returning to a straight shape during retraction. The controller had no knowledge of tube shape and conforms solely in response to contact between the robot and tube surface.

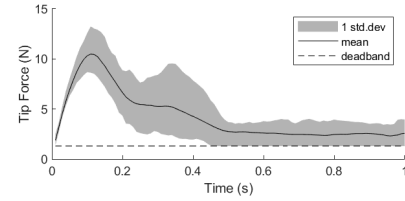


Fig. 10. Tip impulse reduction with compliance.

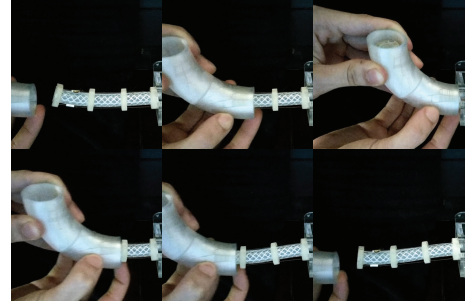


Fig. 11. Individual frames of compliant insertion and retraction demonstration.

Ten insertions were performed over a 90 s period, with reaction data shown in Fig. 12. During the insertion, contact force was maintained at 5-7 N with larger forces reduced more quickly providing a faster insertion. Fig. 13 shows a cumulative histogram indicating the duration for which forces exceeding a given threshold were experienced, averaged over insertion trials. Tip forces exceeding 6 N, 7 N and 8 N were experienced for an average duration of 0.54 s, 0.23 s and 0.07 s respectively. The average insertion/retraction was completed in 3.5 s at a speed of 29 mm/s and cable movement speed of 5 mm/s. These results show that the developed controller based on RNN was capable of endowing the continuum robot with active compliance while exploring unknown environments.

VI. DISCUSSION

The experiments show that a complex model of the mechanics of the continuum robot is not necessary for active

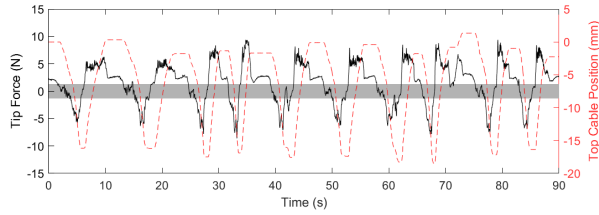


Fig. 12. Reaction to external forces during repeated insertion trials with deadband (grey).

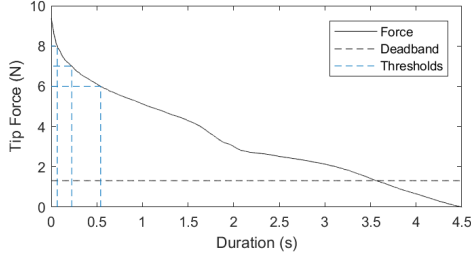


Fig. 13. Duration for which tip forces exceeding a given threshold are experienced in the average insertion.

compliant motion control. However, this does not mean having a model of the robot is useless. A model, even if being inaccurate, can direct the motion of the robot in the data collection procedure for offline training. Moreover, the complex model is avoided by implementing the controller in the actuator space directly for the compliant motion control task. For tasks where the tip of the robot needs to apply a certain force, an accurate model relating the external force output and the measurements of the actuating cables may be needed. This could also be achieved using machine learning techniques given a considerable amount of calibration data. However, the performance of such an approach is yet to be evaluated.

VII. CONCLUSION

Compliance is one of the key capabilities of continuum robots to enhance safety while used in minimally invasive surgery. Active compliant motion control has been developed based on complex derivation of the mechanics model of continuum robots. This paper proposed a model-less approach leveraging an RNN to capture the highly nonlinear factors of the system such as hysteresis of the cables, friction, and delays of electronics, that are usually difficult to model accurately.

Using a 3-tendon single-segment continuum robot with force sensors on each cable, an RNN was trained to predict the internal cable forces produced by backbone stiffness, friction and other factors from the actuator control signals during unloaded motion. Then, a feed-forward motion controller was developed directly in the actuator space to minimize the difference between the predicted cable tension provided by the RNN and the measured tension. The superiority of the RNN over the other architectures was validated by comparison. A set of experiments were conducted to demonstrate

that the continuum robot with an RNN based feed-forward controller was capable of responding to external forces quickly and entering an unknown environment compliantly.

While the positive results on the simple platform show promising feasibility of the machine learning based approach for constructing compliance controller, more experiments need to be carried out on continuum robots that are better designed for real applications. In addition, an extension of the method to multiple-segment continuum robots is worth further study.

REFERENCES

- [1] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.
- [2] E. J. Butler, R. Hammond-Oakley, S. Chawarski, A. H. Gosline, P. Codd, T. Anor, J. R. Madsen, P. E. Dupont, and J. Lock, "Robotic Neuro-Endoscope with concentric tube augmentation," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 2941–2946.
- [3] N. Simaan, K. Xu, W. Wei, A. Kapoor, P. Kazanzides, R. Taylor, and P. Flint, "Design and integration of a telerobotic system for minimally invasive surgery of the throat," *The International Journal of Robotics Research*, vol. 28, no. 9, pp. 1134–1153, 2009.
- [4] L. Wu, S. Song, K. Wu, C. M. Lim, and H. Ren, "Development of a compact continuum tubular robotic system for nasopharyngeal biopsy," *Medical and Biological Engineering and Computing*, in press, 2016.
- [5] H. Yu, L. Wu, K. Wu, and H. Ren, "Development of a multi-channel concentric tube robotic system with active vision for transnasal nasopharyngeal carcinoma procedures," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 1172–1178, 2016.
- [6] A. H. Gosline, N. V. Vasilyev, E. J. Butler, C. Folk, A. Cohen, R. Chen, N. Lang, P. J. del Nido, and P. E. Dupont, "Percutaneous intracardiac beating-heart surgery using metal mems tissue approximation tools," *The International Journal of Robotics Research*, vol. 31, no. 9, pp. 1081–1093, 2012.
- [7] L. Paul, T. Chant, R. Crawford, J. Roberts, and L. Wu, "Prototype development of a hand-held steerable tool for hip arthroscopy," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2017, pp. 989–995.
- [8] I. A. Gravagne and I. D. Walker, "Manipulability, force, and compliance analysis for planar continuum manipulators," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 3, pp. 263–273, 2002.
- [9] R. E. Goldman, A. Bajo, and N. Simaan, "Compliant motion control for continuum robots with intrinsic actuation sensing," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1126–1132.
- [10] —, "Compliant motion control for multisegment continuum robots with actuation force sensing," *IEEE Trans. Robotics*, vol. 30, no. 4, pp. 890–902, 2014.
- [11] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.
- [12] H. Mayer, F. Gomez, D. Wierstra, I. Nagy, A. Knoll, and J. Schmidhuber, "A system for robotic heart surgery that learns to tie knots using recurrent neural networks," *Advanced Robotics*, vol. 22, no. 13-14, pp. 1521–1537, 2008.
- [13] I. D. Walker, "Continuous backbone "continuum" robot manipulators," *Isrm robotics*, vol. 2013, 2013.
- [14] M. Mahvash and P. E. Dupont, "Stiffness control of surgical continuum manipulators," *IEEE Transactions on Robotics*, vol. 27, no. 2, pp. 334–345, 2011.
- [15] W. S. Cleveland and S. J. Devlin, "Locally weighted regression: an approach to regression analysis by local fitting," *Journal of the American statistical association*, vol. 83, no. 403, pp. 596–610, 1988.
- [16] G. Tang, M. Müller, A. Rios, and R. Sennrich, "Why self-attention? a targeted evaluation of neural machine translation architectures," *arXiv preprint arXiv:1808.08946*, 2018.
- [17] N. Ketkar, "Introduction to pytorch," in *Deep learning with python*. Springer, 2017, pp. 195–208.