

# Control of a Soft Actuator using a Long Short-Term Memory Neural Network

Victor Yanev<sup>1</sup>, Maria Elena Giannaccini<sup>2</sup>, and Sumeet S. Aphale<sup>3</sup>

**Abstract**—Soft robots offer new opportunities because of their compliant physical structure and their wide range of applications. Currently the development of such robots is hampered by their low controllability. One of the main constituents of soft robots is the soft actuator. The aim of this project is to improve the control of a non-linear system, the soft actuator, and its interaction with the environment, by training a long short-term memory (LSTM) neural network to accurately predict the actuator's position in space, its curvature, and the force applied by its end-effector on an external object. The increased performance of the trained network resulted in an error as low as  $0.01 \pm 0.005$  N in estimating the force applied by the end effector on the external object. The results show significantly superior performance (on the order of 10 times) in the positional and curvature predictions of the LSTM network when using one marker per air-chamber.

**Keywords:** LSTM, soft robotics, control, soft pneumatic actuator, neural networks

## I. INTRODUCTION

The interest in soft robotics has increased significantly in the past decade owing to the fact that soft robots offer new possibilities to deal with problems that cannot be solved by robots built from rigid materials. Soft robots have compliant physical structures based on low-cost materials, such as hydrogels, electroactive polymers, and elastomers, which makes them cheaper to produce, more physically resilient, flexible, lightweight, adaptable, and safer for delicate environments due to their ability to change shape [1]. Soft robots have many applications, including manufacturing, manipulation, gripping, minimally invasive surgeries, rehabilitation, assistance of the elderly, safe human-robot interaction, handling of fragile materials, bio-inspired prosthetics, and artificial muscles [2]–[4].

Soft actuators are the main component of soft robots. They are deformable bodies, that allow interaction with the outer environment by achieving a desired actuation pattern. [5]. A popular category of actuators features soft pneumatic actuators (SPAs), where actuation is achieved using compressed air. These actuators are made of silicone rubber, a highly flexible and extensible elastomer with high-temperature resistance and good bio-compatibility [6].

Due to the nonlinearities in the behaviour of SPA, which are mainly made of viscoelastic materials and their complex geometry, modelling them analytically is extremely difficult,

VY<sup>1</sup>, MEG<sup>2</sup> and SSA<sup>3</sup> are with the Artificial Intelligence, Robotics and Mechatronic Systems Group (ARMS), School of Engineering, University of Aberdeen, Aberdeen, UK. Email: u01vy18@abdn.ac.uk<sup>1</sup>, elena.giannaccini@abdn.ac.uk<sup>2</sup>, s.aphale@abdn.ac.uk<sup>3</sup>

so approximation techniques are used to explain their behaviour. This, on the other hand, decreases the accuracy of the model and still does not solve the problem of correctly predicting the movement and behaviour of SPAs, which is pivotal to achieve effective control of these actuators.

Alternatively, Finite Element Method (FEM) models provide a more realistic description of the nonlinear response of the system, although they require higher computational effort, especially for flexible materials like silicone rubbers. Another advantage of FEM is that the deformation (and stress) in soft actuators can be easily visualized, leading to a better understanding of the influence of local strain on global actuator performance [7], [8].

Another method that has seen a considerable increase in interest and has developed tremendously in the last decade, is to model a Neural Network (NN) to capture the explicit relationship between input pressure, bending angle, and output force [9] and serve a similar functionality to an analytical model. Given the pressure as an input, the NN should return the actuator position as two-dimensional vectors and its curvature based on the data that has been used to train it. Therefore, this method of modelling requires a large data set about the movement of the actuator to be generated, so that the NN can give accurate predictions of the SPA's behaviour. The main drawback of this method is its black box nature - we cannot explicitly show and fully understand the behaviour of the system, but rather just use the NN model for getting predictions.

In this paper, the focus will lie on a data-driven model-free approach for predicting and controlling the position, curvature, and force output response of an SPA. A data set will be used that contains the results of an experiment conducted to test the SPA at different input pressure values and orientations, while recording the resulting feedback of the behaviour of the actuator. Empirical models will be derived based on the generated experimental data using two common data-driven modelling techniques: regression analysis and recurrent neural networks.

## II. MATERIALS AND METHODS

### A. Background

Previous studies have shown that neural networks can successfully predict the movement and behaviour of soft actuators [10], [11]. We placed additional markers for each chamber of the actuator to improve the existing system's accuracy. In this way, the curvature and the exact position of each chamber at any given input pressure can be predicted by the neural network. This is relevant especially in cases

such as in our study where the actuator is interacting with external objects. This owes to the fact that the actuator's curvature gets distributed non-uniformly when it gets into contact with objects and keeping only the overall curvature is not sufficient to give a good estimation of the actuator's position in space.

### B. PneuNets bending actuator

The soft actuator which was used in our experiments is the PneuNets (pneumatic networks) soft actuator, originally developed by the Whitesides Research Group at Harvard [12]. The main body is comprised of a series of channels inside an elastomer, forming chambers arranged in a row, which inflate when pressurized, creating motion. As the air pressure is applied to these chambers, they push against the adjacent chambers to distribute the effect of pressurization.

The nature of the motion is dictated by the geometry of the embedded chambers and the material properties of their walls. When the actuator is pressurized, expansion occurs in the most compliant (least stiff) regions. Thus, if the actuator is composed of a single, homogenous elastomer, the thinnest areas will expand the most. The behavior of the actuator can be manipulated by selecting wall thicknesses that will result in a desired type of motion.

For the PneuNets actuator we are using [13], the strain-limiting layer is a piece of paper embedded in the base as a bottom layer which makes the actuator bend instead of expand in the axial direction. The thinnest wall sections are those between each chamber and the next. The actuator can be seen in Figure 1.

### C. Long Short-Term Memory Networks

Long short-term memory (LSTM) networks are a class of recurrent neural networks (RNN) widely used for time series predictions. RNNs have been shown to successfully predict the behaviour of soft actuators [14] [15]. In standard feed-forward neural networks (FFNN), all cases are considered to be independent. That is when fitting the model for a particular data point, there is no consideration for the previous data. On the other hand, RNNs have feedback connections, which gives them the ability to process not only single data points, but also entire sequences of data.

RNNs work by repeatedly updating a hidden state  $h$ , a vector that can have arbitrary dimensions. At any given time step  $t$ , the next hidden state  $h_t$  is calculated using the previous hidden state  $h_{t-1}$  and the next input  $x_t$ . Then, the next output  $y_t$  is calculated using  $h_t$ :

$$\begin{array}{ccc} y_0 & y_1 & y_n \\ \uparrow & \uparrow & \uparrow \\ h_0 \rightarrow h_1 \rightarrow \dots \rightarrow h_n \\ \uparrow & \uparrow & \uparrow \\ x_0 & x_1 & x_n \end{array} \quad (1)$$

To calculate the state, a typical regular RNN uses only three sets of weights and two sets of biases to perform its calculations:

- $W_{xh}$ , used for all  $x_t \rightarrow h_t$  links.
- $W_{hh}$ , used for all  $h_{t-1} \rightarrow h_t$  links.
- $W_{hy}$ , used for all  $h_t \rightarrow y_t$  links.
- $b_h$ , added when calculating  $h_t$ .
- $b_y$ , added when calculating  $y_t$ .

All the weights are applied using matrix multiplication, and the biases are added to the resulting products. Then, an activation function is applied to the value to determine the current state of the cell. The following equations define everything said so far:

$$\begin{aligned} h_t &= f_a(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \\ y_t &= W_{hy}h_t + b_y \end{aligned} \quad (2)$$

This dependency lets RNNs maintain information in memory over time. However, standard RNNs cannot be trained to solve problems that require learning long-term temporal dependencies. This is because while training the network a loss function is used to evaluate the predictions, the RNN uses gradient descent to minimize the loss but the gradient decays exponentially with time (called the vanishing gradient problem) [16].

LSTMs, on the other hand, use special units which include a memory cell that can maintain information for longer periods of time and make small modifications to it by using weights and biases just like RNNs. However, with LSTMs, the information flows through a set of gates which are used to control when information enters the memory cell, when it's outputted, and when it's forgotten [17]. This architecture lets them learn longer-term dependencies and gives them the ability to selectively remember or forget data. Thanks to this, the LSTM network can do time-lagged series predictions of complex patterns that have longer-term dependencies such as the movement of our soft actuator.

We used the TensorFlow deep learning library in Python to create our network. The reference pressure inputs (psi) and the current distance from the load cell (mm) were the inputs that were given to the network, and the outputs were the Cartesian coordinates of each chamber relative to the coordinates of the first chamber, the curvature at four different equidistant points (every three chambers) along the actuator's body, and the output force applied by the tip of the actuator to the load cell. The data set generated by the experiments was shuffled and split in the ratio 80:20 for training and testing. The network's layers consisted of: an LSTM layer with a size of 250, whose output is multiplied by a weight matrix in four hidden densely-connected layers of sizes 200, 150, 100, 50, followed by a vector layer that provides the final output from the network. All layers use the rectified linear unit activation function (ReLU). The mini batch size was 64 and the network parameters were optimized using the Adam algorithm [18].

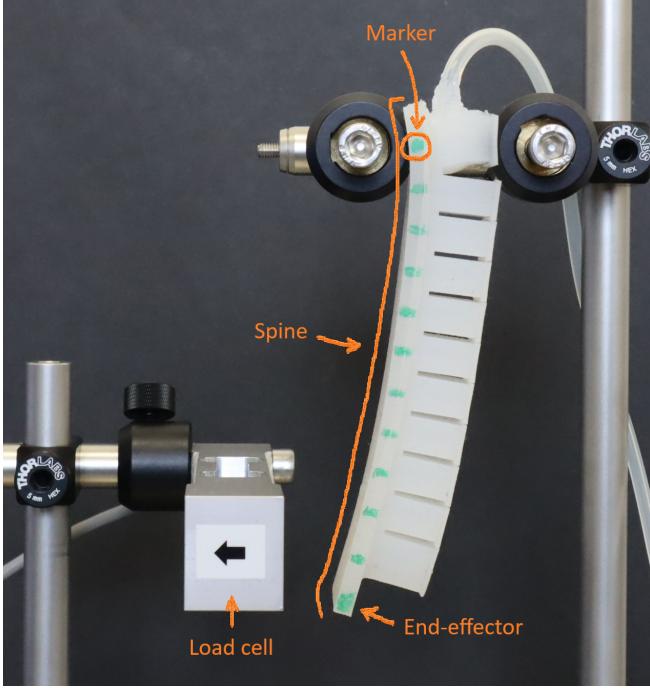


Fig. 1: Side view of the soft actuator with 12 markers placed along its body, one for each chamber and one for the end-effector, used to track their positions and calculate the curvature of different sections across the actuator’s length.

### III. EXPERIMENTAL SETUP

#### A. Physical Setup

In order to maximise the reproducibility of this study we decided to use the PneuNets pneumatic soft actuator detailed on soft robotics toolkit, the open-access website that focuses on explaining the design and fabrication of soft-robotic components. The silicone used to mould the actuator is DragonSkin 10 Slow (Smooth-On, USA). During the experiments the OBUG 3 kg load cell (Applied Measurements, UK) is fixed at 10 mm, 20 mm and 30 mm from the end-effector of the soft actuator. A set of images is taken for every soft actuator/load cell distance and for every different soft actuator internal pressure from 0 psi to 6 psi. Each image is matched to the relative load cell output. The images of the soft actuator are taken using a Canon EOS M50 mirrorless camera.

#### B. Image processing

All chambers of the actuator as well as its end effector were marked with green dots acting as image processing detectable references used to track the kinematic positions and calculate the curvature of the actuator at different pressure values. First, every image was cropped to minimize the chance of external objects to be falsely recognized as a marker region. It was then converted from RGB to HSV color space. An HSV color threshold was then applied to extract the contours of the marker regions and the center point of each region was taken as the coordinates of its respective chamber. The extracted coordinates were then used as shown

in equation (3) to calculate the Menger curvature of the actuators at different chamber sections. Each section needs at least three markers in order to have its curvature calculated, thus, we divided the actuator into four chamber sections.

$$c(x, y, z) = \frac{1}{R} = \frac{4A}{|x-y||y-z||z-x|} \quad (3)$$

where  $R$  is the radius of the circle that passes through the three points  $x, y, z$  and  $|x-y|, |y-z|$ , and  $|z-x|$  are the side lengths of the triangle that is formed by connecting them.

### IV. RESULTS

To demonstrate that tracking the positions of all chambers by placing 12 markers across the actuator’s body will give a significantly more accurate description of the actuator’s shape at different pressure values compared to just tracking its end-effector, its base and its mid-point and calculating the overall curvature, we have trained two different neural networks. The first one was trained using the experimental data from all markers (excluding the first chamber of the actuator which is held by the stand making it practically static throughout the whole experiment). And the second one was trained using a subset of the experimental data with markers only for the first chamber, the middle chamber, and the end-effector, which were used to calculate the overall curvature of the actuator. This section will show the accuracy of the positional, curvature and output force predictions of the two LSTM networks (using twelve markers and three markers) and compare them to evaluate the effectiveness of our proposed methodology.

TABLE I: Magnitude of errors in positional estimations along the  $X$  and  $Y$  axes when using  $n$  markers with initial distance from the load cell of 10 mm, 20 mm, and 30 mm, where  $p$  is the input pressure, and  $L$  is the distance.

$p$ (psi)	$L$ (mm)	$X_{12}$	$Y_{12}$	$X_3$	$Y_3$
0	10	2.16	1.51	2.55	11.26
1	10	0.85	1.46	4.95	10.75
2	10	1.39	1.41	6.97	10.27
3	10	0.60	0.62	5.98	11.29
4	10	0.71	0.96	6.13	12.50
5	10	0.73	2.32	6.19	13.73
6	10	1.09	2.12	7.78	15.18
0	20	1.86	1.54	2.30	12.13
1	20	1.13	1.97	5.70	11.82
2	20	1.42	1.47	7.56	11.83
3	20	1.82	1.97	8.76	12.72
4	20	3.05	2.33	11.27	15.21
5	20	2.25	2.84	6.55	15.64
6	20	3.12	1.77	7.63	16.50
0	30	0.67	1.02	2.28	9.97
1	30	0.60	0.83	5.13	9.37
2	30	1.32	1.12	7.66	10.14
3	30	1.16	0.61	9.67	12.21
4	30	0.80	0.75	8.65	13.23
5	30	0.50	0.58	8.13	14.63
6	30	0.52	0.35	7.38	16.10

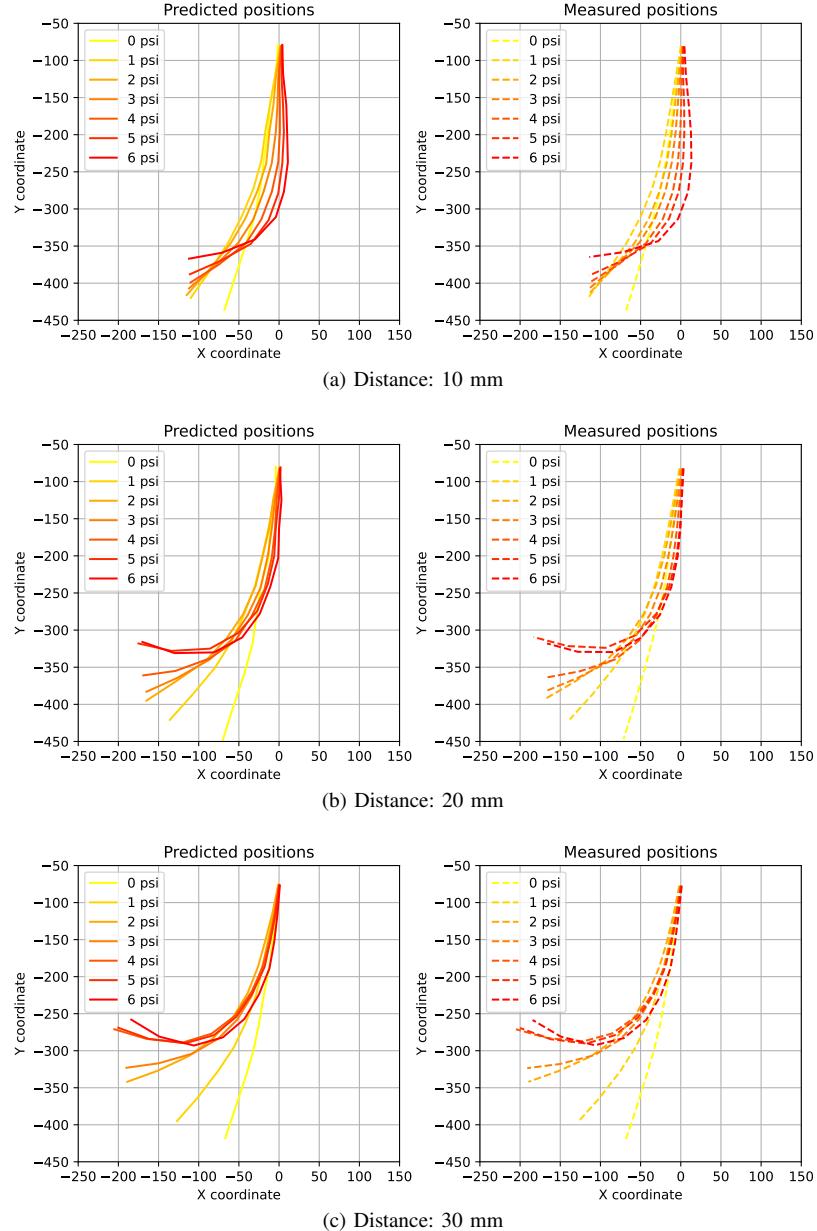


Fig. 2: Predicted positions using all markers and Measured positions at 10 mm (a), 20 mm (b) and 30 mm (c) distance from the load cell. The legend details the magnitude of the internal pressure in the actuator.

#### A. Positional predictions

The model for the positional predictions was trained by using a data set that maps each distance (10 mm, 20 mm, 30 mm) and input pressure (0 psi to 6 psi) to the relative positions of the actuator's chambers and end-effector (this is their XY-coordinates relative to the coordinates of the first chamber), extracted from the markers in the corresponding image. The reference pressure input (psi) and distance from the load cell (mm) are the inputs that the NN expects to get, and the output that it gives are the positions of the actuator's chambers and end-effector relative to the first chamber. A comparison of the accuracy of the positional predictions of the two LSTM networks is shown in Table I.

The predictions of the LSTM network with the proposed improvement can be seen in Figure 2. It is evident that the actuator behaves differently at different distances from the load cell. The actuator bends almost uniformly until it reaches its target and, depending on the distance between the load cell and the end-effector in the initial position, this reach happens at a different pressure value. After contact has been made, the actuator starts to bend in a non-uniform way, with the 7-12 sections bending more than the rest of the body. The 3-marker LSTM network cannot recognize this, and fails to give an accurate prediction of the displacement in those chambers. As seen in Table I, the 12-marker LSTM network gives 10 times more accurate predictions of the actuator's position.

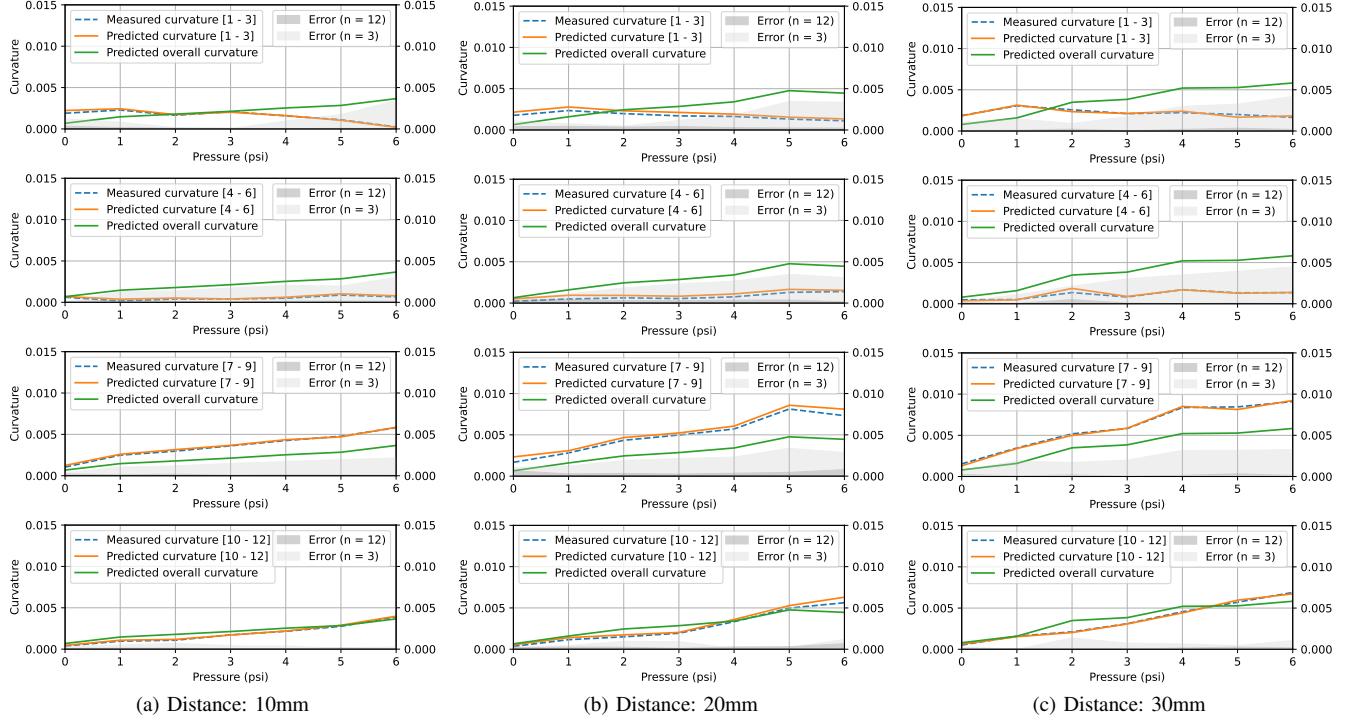


Fig. 3: Predicted curvature using all markers, Predicted overall curvature using three markers, Measured curvature and Prediction Error at 10 mm (a), 20 mm (b) and 30 mm (c), from the load cell. Here  $n$  stands for the amount of markers used.

### B. Curvature predictions

The model for curvature predictions was trained by using the same methodology as before, but instead of directly using the coordinates of the twelve markers, they were used to calculate the curvature at four different equidistant chamber regions along the actuator's body and feed these calculated curvatures as the training data to the model. The inputs remained the same as before. The output is the predicted curvature at each of the four chamber regions. The predictions can be seen in Figure 3.

As expected, the actuator's curvature increased proportionally with the input pressure. Intriguingly, the actuator experienced a different curvature for all regions along its body. The curvature at the chambers further away from the end effector in sections 1-6 was lower, whereas the curvature in the chambers which are closer to the end effector in sections 7-12 was higher, with it experiencing the highest curvature at chambers 7-9, which are situated around the region that lays at 3/4 along the actuator's length. The LSTM network which used only three markers could give a prediction only of the overall curvature of the actuator and, thus, it performed 10 times worse than the one which uses twelve markers. The three-marker LSTM network experienced the highest errors at chambers 1-3 and 4-6, where the difference between the predicted overall curvature and the measured curvature at that region was the highest (see Figure 3).

### C. Force predictions

The LSTM network using three-markers and the one using twelve-markers performed the same in terms of their force predictions. This was expected since their model was trained by using the force output data which comes from the load cell and does not depend on any visual measurements done by image processing. All inputs to the network remained unchanged. The output was the force applied by the end-effector to the load cell. The accuracy of the force predictions and the total error is shown in Figure 4. The learned model performed with an average error of  $0.01 \pm 0.005$  N in estimating the output force of the end-effector.

As expected, the load cell experiences the highest applied force at 10 mm distance, increasing proportionally with the pneumatic pressure inside the actuator as expected. For longer distances, the applied force output decreases after the end effector at the end of "spine" starts to lose contact with the load cell. From 5 psi to 6 psi at 20 mm distance and from 4 psi to 5 psi at 30 mm the actuator experiences a saturation in force. These pressure values corresponds in points where the actuator starts making contact with the load cell.

As seen in Figure 4 the maximum force is 0.7 N for 10 mm, 0.45 N for 20 mm and 0.4 N for 30 mm. This is because the actuator - once is further than 10mm - in the higher pressure experiments does not make contact with its "spine" but with the distal pressurised chamber. The force it can exert with the "spine" is higher than the force it can exert with the distal pressure chamber, as the chamber will bend much more easily than the "spine".

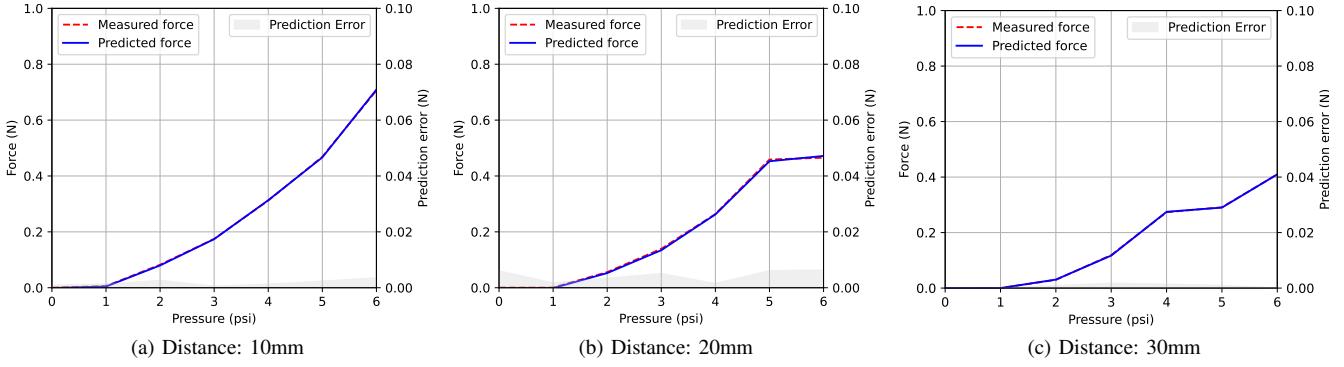


Fig. 4: Predicted force output, Measured force output and Prediction error at 10 mm (a), 20 mm (b) and 30 mm (c), from the load cell.

## V. CONCLUSIONS AND FUTURE WORK

The work conducted in this study demonstrates the effectiveness of a LSTM network at characterising the physical interaction between a PneuNet actuator and a stationary object. It is shown that the combination of a LSTM network and one marker per pressure chamber makes it possible to predict both position and the curvature of the bending actuator during the interaction. The LSTM network can also accurately predict the force exerted on the object. These results characterise and predict the interaction between a non-linear pneumatic, soft actuator and a rigid object. For example, the characterisation we created allows us to know how far a rigid object is from the actuator once we know its internal pressure and the load cell response. The ability of the neural networks at informing this interaction can be used in multiple applications including grasping and wearable devices. The non-linearity of soft actuators decreases their controllability, especially when considering the interaction with the environment. Our results show that neural networks are indeed useful at increasing the predictability of such interactions. In future iterations of this work we plan to include a bend sensor to compare its readings with the LSTM predictions. In addition, we plan to use a Cosserat rod model as an input to the network to reduce training time.

## ACKNOWLEDGMENTS

This work was supported by the Carnegie Trust Vacation Scholarship funding, awarded to Victor Yanev. All of the code written for this paper is available and can be enquired by emailing one of the authors.

## REFERENCES

- [1] S. I. Rich, R. J. Wood, and C. Majidi, “Untethered soft robotics,” *Nature Electronics*, vol. 1, no. 2, pp. 102–112, 2018.
- [2] M. Cianchetti, C. Laschi, A. Menciassi, and P. Dario, “Biomedical applications of soft robotics,” *Nature Reviews Materials*, vol. 3, no. 6, pp. 143–153, 2018.
- [3] Y. K. Lee and I. Shimoyama, “A skeletal framework artificial hand actuated by pneumatic artificial muscles,” in *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*, vol. 2. IEEE, 1999, pp. 926–931.
- [4] Y. Hao, Z. Gong, Z. Xie, S. Guan, X. Yang, Z. Ren, T. Wang, and L. Wen, “Universal soft pneumatic robotic gripper with variable effective length,” in *2016 35th Chinese control conference (CCC)*. IEEE, 2016, pp. 6109–6114.
- [5] N. El-Atab, R. B. Mishra, F. Al-Modaf, L. Joharji, A. A. Alsharif, H. Alamoudi, M. Diaz, N. Qaiser, and M. M. Hussain, “Soft actuators for soft robotic applications: a review,” *Advanced Intelligent Systems*, vol. 2, no. 10, p. 2000128, 2020.
- [6] N. Farrow and N. Correll, “A soft pneumatic actuator that can sense grasp and touch,” in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 2317–2323.
- [7] P. Polygerinos, Z. Wang, J. T. Overvelde, K. C. Galloway, R. J. Wood, K. Bertoldi, and C. J. Walsh, “Modeling of soft fiber-reinforced bending actuators,” *IEEE Transactions on Robotics*, vol. 31, no. 3, pp. 778–789, 2015.
- [8] M. S. Xavier, A. J. Fleming, and Y. K. Yong, “Finite element modeling of soft fluidic actuators: Overview and recent developments,” *Advanced Intelligent Systems*, vol. 3, no. 2, p. 2000187, 2021.
- [9] M. E. Giannaccini, C. Xiang, and A. Atyabi, “Novel design of a soft lightweight pneumatic continuum robot arm with decoupled variable stiffness and positioning,” *Soft robotics*, vol. 5, no. 1, 2018.
- [10] P. Preechayosbon and E. Rombokas, “Sensuator: A hybrid sensor-actuator approach to soft robotic proprioception using recurrent neural networks,” in *Actuators*, vol. 10, no. 2. Multidisciplinary Digital Publishing Institute, 2021, p. 30.
- [11] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, “Control strategies for soft robotic manipulators: A survey,” *Soft robotics*, vol. 5, no. 2, pp. 149–163, 2018.
- [12] R. V. Martinez, C. R. Fish, X. Chen, and G. M. Whitesides, “Elastomeric origami: Programmable paper-elastomer composites as pneumatic actuators,” *Adv. Func. Mater.*, vol. 22, pp. 1376–1384, 12/8/11 2012, 1145.
- [13] P. Polygerinos, S. Lyne, Z. Wang, L. F. Nicolini, B. Mosadegh, G. M. Whitesides, and C. J. Walsh, “Towards a soft pneumatic glove for hand rehabilitation,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013, p. 1512–1517.
- [14] T. G. Thuruthel, B. Shih, C. Laschi, and M. T. Tolley, “Soft robot perception using embedded soft sensors and recurrent neural networks,” *Science Robotics*, vol. 4, no. 26, 2019.
- [15] D. Kim, S.-H. Kim, T. Kim, B. B. Kang, M. Lee, W. Park, S. Ku, D. Kim, J. Kwon, H. Lee *et al.*, “Review of machine learning methods in soft robotics,” *Plos one*, vol. 16, no. 2, p. e0246102, 2021.
- [16] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *arXiv preprint arXiv:1412.3555*, 2014.
- [17] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [18] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.