


Continuum-Body-Pose Estimation From Partial Sensor Information Using Recurrent Neural Networks

Kazutoshi Tanaka , *Member, IEEE*, Yuna Minami, Yuji Tokudome, Katsuma Inoue ,
Yasuo Kuniyoshi , *Member, IEEE*, and Kohei Nakajima 

Abstract—Soft continuum arms have significant potential for use in various applications due to their extremely high degrees of freedom. For example, these soft arms can be used for grasping and manipulating fragile materials in the deep sea or carrying a human to rescue in unstructured environments. However, in these situations, the environment is often dark and visual cues are not always usable. Therefore, these arms must estimate their pose from proprioceptive sensors to control their behavior and execute their tasks in dark places. Estimating the pose in a dynamic situation is still challenging because of the arms' high dimensionality and the complex structural changes in the body shape. Therefore, this study demonstrates a novel method for estimating the pose of proprioceptive bending sensors using recurrent neural networks (RNNs). In particular, an RNN framework known as deep reservoir computing was used for this purpose. Results from experiments using an octopus-inspired soft robotic arm clearly indicate that the proposed method significantly outperforms existing methods using long short-term memory models or linear models. We expect that our proposed method will enable behavioral control of these arms in dark places such as the deep sea, space, and inside the human body in future applications.

Index Terms—AI-based methods, control, learning for soft robots, modeling.

I. INTRODUCTION

A SOFT robotic arm changes the entire continuum-body-pose in a continuous, complex, and varied manner. A soft robotic arm has great potential for use in a variety of applications due to its extremely high degrees of freedom [1]. Possible applications include grasping and manipulating fragile materials in the deep sea or carrying humans [2] to rescue in unstructured environments [3], [4]. Therefore, the robotic arm must often

estimate its pose from proprioceptive sensors to execute their tasks in dark places. Estimating the entire continuum-body-pose of a soft robotic arm is currently a challenging task because of its high dimensionality and the fact that complex structural changes in the body shape often require many embedded sensors. Therefore, in this study, we aim to develop a novel method for estimating the entire continuum-body-pose of a soft robotic arm for various patterns of control inputs and movements.

A soft robotic arm is often driven at the base of the arm, generating passive body dynamics that is largely affected by the history of the motor commands [5], [6]. Therefore, recurrent neural networks (RNNs) that can deal with memories of past inputs are suitable for estimating the pose. In this study, we present a method to estimate the center line of the soft robotic arm as representative of the pose from the outputs of the bend sensors using echo state networks (ESNs) of reservoir computing (RC) as a type of RNN [7], [8], [9]. In particular, we make use of an architecture known as *deep reservoir*, which aligns multiple reservoirs in a feed-forward manner [10]. In general, RC can be trained stably and quickly, and accordingly, it is computationally inexpensive, making it suitable for physical implementations for edge computing [9], [11].

The contribution of this work is the proposal of a method for estimating the continuum-body-pose of a soft robotic arm from embedded bending sensors using an RC framework. This paper is organized as follows: Section II summarizes related research; Section III introduces the methods used for continuum-body-pose estimation, Section IV describes the experiments, and Section V discusses the comparative experiments. Finally, Section VII concludes the manuscript.

II. RELATED WORKS

The pose of a soft robotic arm can be estimated using embedded proprioceptive sensors if the relationship between the output of the sensors and the resultant pose have been revealed. This forms a type of indirect sensing task and learning from these relationships from the data has often been done using machine learning approaches [12], [13], [14].

Soter et al. proposed a method for predicting the visual information of an octopus-inspired soft robotic arm from bend sensors integrated into the soft body, using a combination of a stacked convolutional autoencoder (CAE) and an RNN [15]. This method predicts the visual images of the octopus-inspired soft robotic arm from the bend sensors. However, their method

Manuscript received 24 February 2022; accepted 10 August 2022. Date of publication 16 August 2022; date of current version 29 August 2022. This letter was recommended for publication by Associate Editor F. L. Hammond III and Editor Y.-L. Park upon evaluation of the reviewers' comments. The work of Kohei Nakajima was supported in part by JSPS KAKENHI Grant Number JP18H05472 and in part by the JST CREST under Grant JPMJCR2014. This work was supported in part by JSPS KAKENHI Grant Number JP19K14936 and in part by New Energy and Industrial Technology Development Organization (NEDO). (*Corresponding author: Kazutoshi Tanaka.*)

Kazutoshi Tanaka is with OMRON SINIC X Corporation, Tokyo 1130033, Japan (e-mail: kazutoshi.tanaka@sinix.com).

Yuna Minami, Yuji Tokudome, Katsuma Inoue, Yasuo Kuniyoshi, and Kohei Nakajima are with the Graduate School of Information Science and Technology, The University of Tokyo, Tokyo 113-8656, Japan (e-mail: ckmy315.ml0@gmail.com; cherishtravailer@yahoo.co.jp; k-inoue@isi.imi.i.u-tokyo.ac.jp; kuniyosh@isi.imi.i.u-tokyo.ac.jp; jc_mc_datsu@yahoo.co.jp).

Digital Object Identifier 10.1109/LRA.2022.3199034

trained and tested the CAE and RNN using only one pattern of arm motion specific to a single input series. Therefore, estimating the pose of a soft robotic arm, in general, may require additional techniques when the robot receives different patterns of input commands and executes various motions.

Thuruthel et al. presented a method for the real-time perception of a soft actuator using embedded bending sensors and RNNs [12]. Their method predicted the touching force on the actuator and the tip position from: the output of the soft sensors, the input pressure to the actuator, and the current state of the long-short term memory (LSTM) network. Although this method predicted the tip position of the soft actuator when it bent in one direction, complex changes in the shape of the soft robotic arm, such as bending multiple times, were not considered in this method.

Tariverdi et al. proposed a method for predicting the dynamics of soft robotic arms using RNNs [16]. A kind of soft robotic arm can be used for closed-loop applications using their method, which generates force and torque in the middle part of the arm and controls the movement of the arm. However, their method is only applicable for specific settings of a soft robotic arm, and it cannot be extended to the wholly passive dynamics of a soft robotic arm directly, which is the focus of this study.

Van Meerbeek et al. developed sensors made of internally illuminated elastomer foam for soft robot proprioception [17]. Optical fibers transmitted light into the foam and received diffuse waves from internal reflection. From the diffuse reflected light, machine learning models detected the deformation type from four different types: clockwise rotation, counterclockwise rotation, bent up, or bent down. These models also estimated the magnitude of the deformation type. Although these models detected the type and magnitude of the deformation, the continuous change of the center line of the soft robotic arm in a complex manner was not considered.

In contrast to these methods, our method estimates the center line of the soft robotic arm driven at the base from embedded bending sensors, which can change in a complex manner, by receiving multiple patterns of input commands.

III. METHOD

A. Problem Formulation: Continuum-Body-Pose Estimation

A soft robotic arm was immersed in the water tank for measurements as shown in Fig. 1. The soft robotic arm used in the present study was 447 mm long, made of silicone rubber (ECOFLEX 00-30 from Smooth-On, Inc.), equipped with ten flexible bending sensors (Flexpoint Sensor Systems, Inc.) embedded in the robot, and driven by a servo motor (Dynamixel RX-64, ROBOTIS) placed on the base. These bending sensors are light and flexible such that they do not affect the movement of the soft robot. The size of the tank is 1000x500x500 mm (WxDxH). Further details are described in [5], [18], [19]. A computer sent the target joint angle to the motor as a set of input commands and received a joint angle of the motor from the servo motor.

The input to the neural networks for the pose estimation comprised of twelve components including the target joint angle,

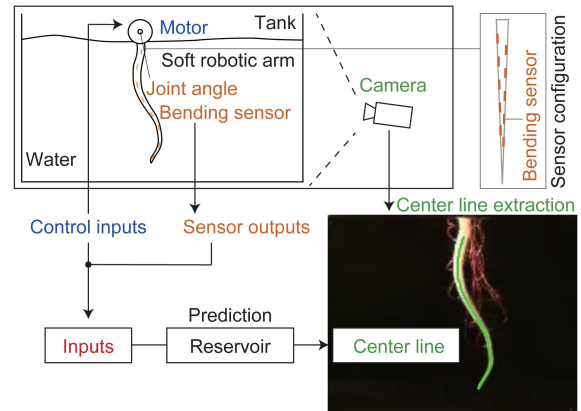


Fig. 1. Schematic representing the overall strategy of our experiments. With this schema, we aim to implement a continuum-body-pose estimation from embedded bending sensors using reservoir networks.

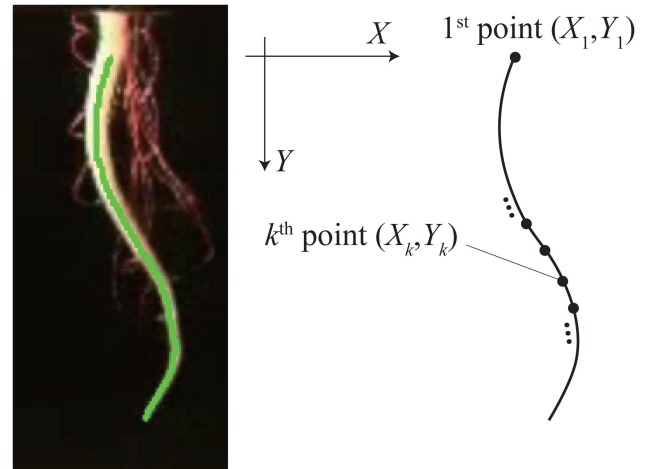


Fig. 2. Image and schematic illustration of the points of the center line. The number of the point was 1000. We set the Y position of the starting points, and its X position was calculated from the image area of the arm and the Y position.

actual joint angle, and outputs of the ten bending sensors. A vector $\mathbf{u} \in \mathbb{R}^{N_u}$ represented the input, where $N_u (= 12)$ is the number of variables of \mathbf{u} .

A camera on the side of the water tank captured the motion of the arm. As shown in Fig. 2, the L -point sequence $(\mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_L)$ was extracted from the captured video image, where $\mathbf{r}_k = [X_k, Y_k]^T$ is a vector that represents the position of the k -th point counted from the base to the tip. The center line of the soft robotic arm was represented as $\mathbf{y} = [X_1, Y_1, X_2, Y_2, \dots, X_L, Y_L]^T \in \mathbb{R}^{N_y}$ ($N_y = 2L, L = 1000$). The length of the arm in the video images is 110 px, which means 1 px represents 4 mm. Note that the points of the center line are often located between image points, and X_k and Y_k are not integers.

Two types of signal were used for input commands. The target joint angle of the motor was the control input, and its scale was normalized using the maximum angle as described in the final part of this subsection. The first was a random input of -1 and $+1$ (*Random Condition*) from a uniform probability distribution

and determined in an independent and identically distributed (i.i.d.) condition. The input was changed at every τ timesteps. The time length τ was set to 5, 11, and 20 timesteps [5], [18]. The second type of command was a superimposed sine wave (*Sine Condition*) [19]. The input at the timestep t is

$$I_t = 0.2 \sin\left(2\pi f_1 \frac{t}{\tau}\right) \sin\left(2\pi f_2 \frac{t}{\tau}\right) \sin\left(2\pi f_3 \frac{t}{\tau}\right), \quad (1)$$

where (f_1, f_2, f_3) is set to (2.11, 3.73, 4.33), and τ is set to 100, 200 and 300 timesteps, respectively.

Note that the input is independent from its past inputs in the Random Condition, because the binary sequence of the random input command is i.i.d., whereas the input had a temporal correlation in the Sine Condition.

The scale of the control input, angle, and positions were changed as $u = u_{\text{raw}}/c_u$, where u_{raw} are the raw data and c_u is a constant for scaling, respectively. The c_u of the control input was same to c_u of the angle and different from c_u of the positions. The c_u of the control input and the angle were set to the maximum angle. The c_u of the positions was set to 700 px, which is approximately the maximum value of X_k and Y_k . The samples do not have a unit after scaling.

B. Center Line Extraction From Video Images

From the captured video image of the robotic arm, \mathbf{y} , the point sequence representing the center of the soft robotic arm, was extracted as follows (details can be found in [20]). Note that the extracted center line was used as the target data and for the evaluation.

First, the basal point $\mathbf{r}_1 = (X_1, Y_1)$ is estimated. Initially, the raw image is binarized, and the region of the object Ω is extracted from the binarized image. Next, the minimum Euclidean distance field $D(x)(x \in \Omega)$ is calculated from the contour of Ω . Based on $D(x)$, \mathbf{r}_1 is estimated. Note that we set Y_1 manually in our setup, and X_1 was calculated from Y_1 and $D(x)$.

Second, the tip point $\mathbf{r}_L = (X_L, Y_L)$ is estimated. First, the speed vector field is calculated as

$$F(x) = \exp(\alpha D(x)), \quad (2)$$

where α is a constant value. We set $\alpha = 0.5$. Next, a closed curve Γ propagates the normal curve to itself with speed $F(x)$ from \mathbf{r}_1 . The traveling time field $T(x)$ denotes the time when Γ passes over x . Assuming a special case in which the wave front moves in one direction, $F(x)$ and $T(x)$ are calculated as

$$|\nabla T(x)|F(x) = 1. \quad (3)$$

The local maximum point of $T(x)$ was selected as \mathbf{r}_L .

Finally, the L point sequence $\{\mathbf{r}_i\}(i = 1, \dots, L)$ distributed at regular intervals on the center line are extracted by connecting \mathbf{r}_1 and \mathbf{r}_L . The curve C that minimizes the accumulated value of cost function U is

$$C := \operatorname{argmin} \int_C U(C(s)) ds, \quad (4)$$

$$U := \exp(-\alpha D(x)) = (F(x))^{-1}. \quad (5)$$

A point sequence $\{\mathbf{q}_i\}(i = 1, \dots, L)$ on the minimum cost path is determined by backtracking along a gradient ∇T from \mathbf{r}_L until \mathbf{r}_1 is reached. The sequence $\{\mathbf{r}_i\}$ are calculated by smoothing $\{\mathbf{q}_i\}$.

C. Reservoir Computing System

In our experiments, we used a deep reservoir computing framework that aligns multiple RNNs in a feed-forward manner. First, we briefly explain the basic settings of a single-reservoir computing system in this section. We consider ESNs with N_u inputs, N_x reservoir neurons, and N_y output neurons. The middle layer receives the input \mathbf{u} and is referred to as the reservoir. A vector $\mathbf{x} \in \mathbb{R}^{N_x}$ represents the state of the nodes in the reservoir. The reservoir has weight matrices $W_{\text{in}} \in \mathbb{R}^{N_x \times N_u}$, $W \in \mathbb{R}^{N_x \times N_x}$, and $W_{\text{out}} \in \mathbb{R}^{N_y \times N_x}$. The state transition of \mathbf{x} at the discrete time t is

$$\mathbf{x}_{t+1} = \tanh(\rho W \mathbf{x}_t + k_{\text{in}} W_{\text{in}} \mathbf{u}_{t+1}), \quad (6)$$

where k_{in} is scalar and ρ is the spectral radius of W , respectively. The output of the reservoir is

$$\mathbf{y} = W_{\text{out}} \mathbf{x}. \quad (7)$$

The initial weights in W_{in} , W , and W_{out} were randomly determined using a uniform distribution in $(-1, +1)$. Only the matrix W_{out} was trained using back propagation with a loss of the mean square error (MSE). MSEs were calculated as $(1/N) \sum_{t=0}^{N-1} \|\epsilon\|^2$, where $\epsilon = y - \hat{y}$ is the vector of prediction errors, \hat{y} is the actual position, y is the predicted position, and N is the number of samples ($N = 500$). The parameters ρ and k_{in} were subsequently optimized using a grid search and back propagation, as described in [21], [22], where the grid search is applied to set the initial values of parameters to start back propagation. We tested $k_{\text{in}} = 10^k$ ($k = -4.0, -3.5, \dots, 1.0$) and $\rho = 0, 0.2, \dots, 1.2$ when using a grid search. The parameter $p \in \{k_{\text{in}}, \rho\}$ is updated as

$$p_{t+1} = p_t - k_p \frac{\partial E_{t+1}}{\partial p}, \quad E = \frac{1}{2} \|\epsilon\|^2 \quad (8)$$

where k_p is scalar. The gradient $\partial E / \partial p$ is calculated as

$$\frac{\partial \mathbf{x}_t}{\partial \rho} = \tanh'(\rho W \mathbf{x}_{t-1} + k_{\text{in}} W_{\text{in}} \mathbf{u}_t) .* \rho W \frac{\partial \mathbf{x}_{t-1}}{\partial \rho}, \quad (9)$$

$$\frac{\partial \mathbf{x}_t}{\partial k_{\text{in}}} = \tanh'(\rho W \mathbf{x}_{t-1} + k_{\text{in}} W_{\text{in}} \mathbf{u}_t) .* \left(\rho W \frac{\partial \mathbf{x}_{t-1}}{\partial k_{\text{in}}} + W_{\text{in}} \mathbf{u}_t \right), \quad (10)$$

$$\frac{\partial E_{t+1}}{\partial p} = -\epsilon_{t+1}^\top W_{\text{out}} \frac{\partial \mathbf{x}_{t+1}}{\partial p}; \mathbf{0}_{\mathbf{u}}, \quad (11)$$

where \tanh' is the derivative of \tanh , $\mathbf{0}_{\mathbf{u}}$ is a zero vector, and $.*$ denotes component-wise multiplication of two vectors, respectively. We set the learning rate, the number of epochs and the batch size for back propagation to 0.001, 100, and 1000, respectively. The model was trained using an Adam optimizer.

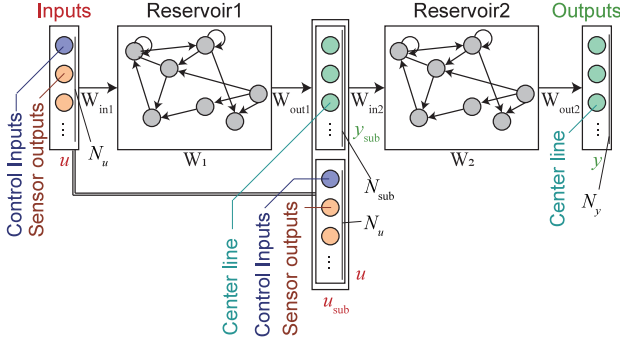


Fig. 3. Deep echo state network with a two-layered setting.

D. Deep Echo State Networks With a Two-Layered Setting

We speculated that using variables related to the pose of a soft robotic arm to estimate the pose may increase the accuracy of the pose estimation. Any points within the point sequence of the center line were expected to depend on each other under certain relations. For example, a point should be between its two neighbors. Thus, we established a working hypothesis that using the partial information of the center line positions may increase the accuracy of the estimation of the entire center line positions. Based on this consideration, we used partially predicted center line positions, control inputs, and sensor outputs to estimate the pose.

We introduce a deep ESN for our experiments, which is a type of deep reservoir architecture with two-layered settings, as shown in Fig. 3. As shown in the figure, reservoir1 estimates the partial information of the center line positions, and reservoir2 estimates the entire center line positions from the partial information. We selected L_{sub} points at equal intervals to be predicted by \mathbf{u} and to predict \mathbf{y} . The vector $\mathbf{y}_{\text{sub}} \in \mathbb{R}^{N_{\text{sub}}}$ ($N_{\text{sub}} = 2L_{\text{sub}}$) represents the positions of the points. Our proposed method trains a reservoir f_{pre} (referred to as reservoir1) that approximates $\mathbf{y}_{\text{sub}} = f_{\text{pre}}(\mathbf{u})$ and f_{post} (referred to as reservoir2) that approximates $\mathbf{y} = f_{\text{post}}(\mathbf{u}_{\text{sub}})$, where $\mathbf{u}_{\text{sub}} = [\mathbf{u}^T, \mathbf{y}_{\text{sub}}^T]^T$ is a vector. Reservoir1 and reservoir2 are trained by the same method described in Section III-C. The ρ and k_{in} of reservoir1 are optimized, and W_{out} of the reservoir is trained first. Next, ρ and k_{in} of reservoir2 are optimized, and W_{out} of the reservoir is trained. We set the number of nodes in reservoir1 and the number of nodes in reservoir2 to 500.

To determine the appropriate setting of \mathbf{y}_{sub} , we changed L_{sub} and compared the time-and-position averages of the MSEs, as shown in Fig. 4. The time-and-position average is $\frac{\sum_{t=1}^{N_t} \sum_{k=1}^L |X_k(t) - \hat{X}_k(t)|}{L/N_t}$, where $X_k(t)$ and $\hat{X}_k(t)$ are the estimated and actual X positions, respectively, of the k^{th} point at the timestep t and $N_t (= 500)$ is a number of the timesteps. The MSEs of the “Two layered” system were calculated using deep echo state networks with a two-layered setting, as shown in Fig. 3. The MSEs of the “One layered” system were calculated using one ESN with 1,000 nodes to investigate the effects of using deep ESNs. The MSEs of the “Lower bound” system were calculated by reservoir2 using $\hat{\mathbf{y}}_{\text{sub}}$ to investigate

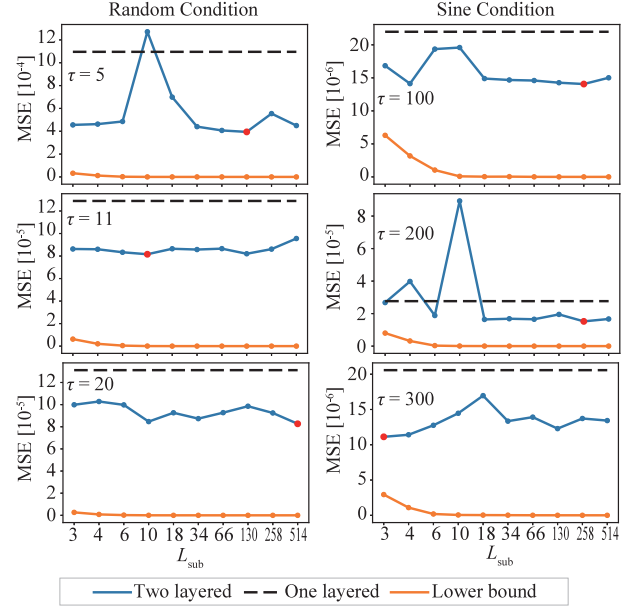


Fig. 4. The time-and-position averages of the MSEs when changing the number of intermediate points L_{sub} . The red dots show the minimum MSEs obtained using the two-layered ESNs.

the lower bound of the MSEs in cases where reservoir1 estimates \mathbf{y}_{sub} without error, where $\hat{\mathbf{y}}_{\text{sub}}$ is the actual \mathbf{y}_{sub} .

Fig. 4 shows that the MSEs of the “Two layered” system were smaller than those of the “One layered” system in the most cases. The figure also shows that the MSEs of the “Lower bound” system decreased as L_{sub} increased, but the changes of the MSEs of the “Two layered” system were small in most cases, which was unexpected. The reason why the change of MSEs of “Two layered” system were small can be speculated that the errors of reservoir1 increases as L_{sub} increases and the prediction errors of reservoir1 cause the errors of reservoir2 in the system. From these results, we used $L_{\text{sub}} = 3$ throughout the paper.

IV. EXPERIMENT

The measured samples in a condition were divided into 1,000, 15,000, 500, and 500 samples, which correspond to the samples used for washout, training, validation, and testing, respectively. The washout was the initialization of the ESN. The predictions were evaluated by MSEs using test samples.

The time series of the measurement and learning results are shown in Fig. 5. Fig. 5 shows the motor command and angle, the response of the bend sensors, the activation of the nodes in the reservoirs, the actual X positions of the sequence, the predicted X positions of the point sequence, and the difference between the actual X and the predicted X .

The graph of the command to the motor and the angle in the Random Condition with $\tau = 11$ shows that two target angles were given and that the angle of the motor converged to that target angle during the time $\tau = 11$.

The graphs of the nodes in reservoir1 and reservoir2 show that the different nodes were synchronized in the Random

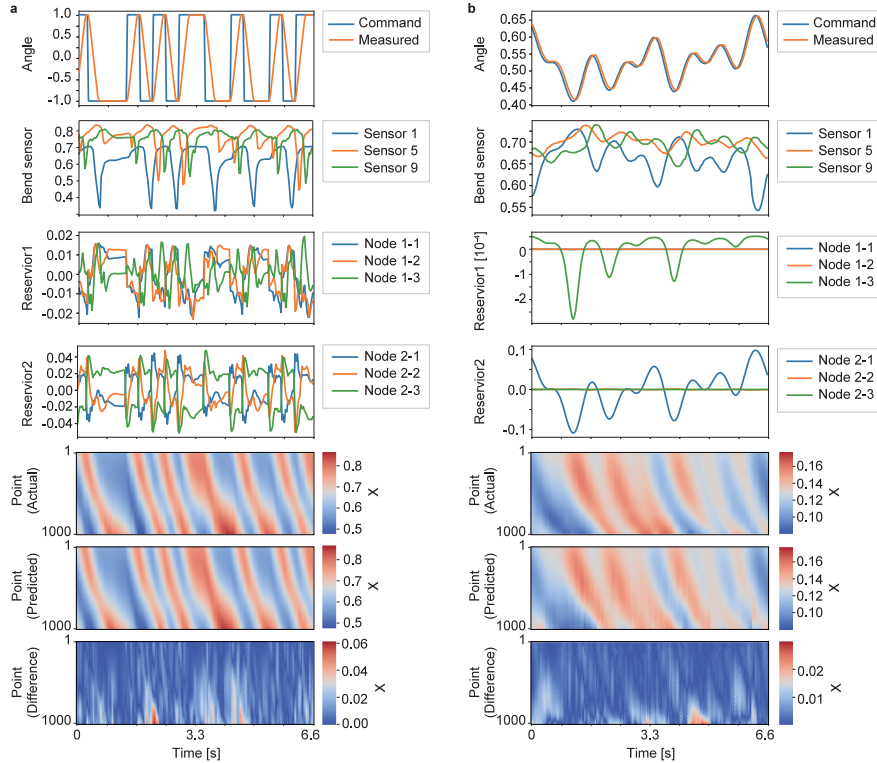


Fig. 5. Timeseries. (a) Random $\tau = 11$, (b) Sine $\tau = 300$. From top to bottom: the motor command and angle, the response of the bend sensors, the activation of nodes in the reservoir, the predicted X positions of the point sequence, the actual X positions of the sequence, and the difference between the predicted X and the actual X . The points are also shown in Fig. 2. In Random $\tau = 11$, the spectral radius of the reservoir1 $\rho = 1.43[10^{-5}]$ and that of the reservoir2 $\rho = 0.74$. In Sine $\tau = 300$, the spectral radius of the reservoir1 $\rho = 6.50[10^{-5}]$ and that of the reservoir2 $\rho = 2.77[10^{-7}]$. Note that the response of some nodes is small in Sine $\tau = 300$, which are the node 1-1, node 1-2, node 2-2, and node 2-3, but these nodes respond to the input and affect the output.

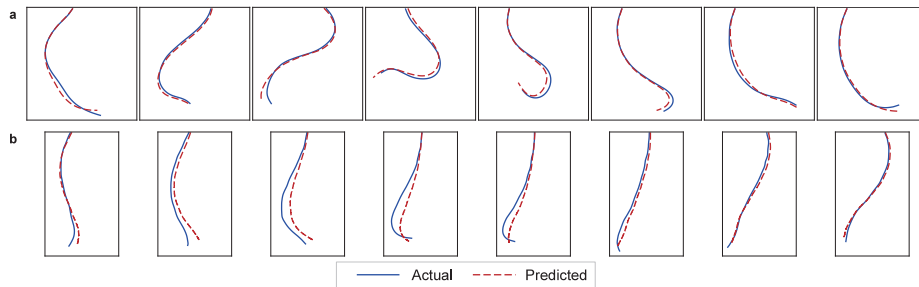


Fig. 6. Snapshots. (a) Random $\tau = 11$, (b) Sine $\tau = 300$. The snapshots show the actual center lines and predicted ones every 5 steps. See also Supplementary Videos.

Condition with $\tau = 11$. The spectral radius of the reservoir1 $\rho = 1.43[10^{-5}] < 1$ and that of the reservoir2 $\rho = 0.74 < 1$ in the condition and the response of these reservoirs has the echo state property. Thus, the generalized synchronization would appear [23], [24].

The graph showing the response of the bend sensor and the X position of the points on the center line shows the effect of the input propagated from the base to the tip.

Lastly, the prediction error figure shows that a large error occurred at the tip. This may be because the soft robotic arm was driven from the base, so the state of the location far from the base tends to be uncertain in relation to the input, whereas the location near the base could be reliably predicted.

Fig. 6 shows the snapshots of the predicted center line alongside the actual one. The snapshots demonstrate that the learned model was able to predict the center line positions from the base to the tip, even when the arm bent in multiple directions (see also the Supplementary Movies).

V. ABLATION STUDY

A. Type of Input Commands

In both the Random and Sine Conditions, the input changed as the value of τ changed, the behavior of the soft robotic arm changed, and the accuracy of the pose estimation changed. Thus, we compared the prediction accuracy for different inputs.

TABLE I
MSE OF DIFFERENT INPUT. *mean ± std*

	Random [10^{-4}]		Sine [10^{-5}]
$\tau = 5$	4.56 ± 15.78	$\tau = 100$	1.68 ± 3.98
$\tau = 11$	0.86 ± 2.30	$\tau = 200$	2.68 ± 10.61
$\tau = 20$	0.99 ± 2.63	$\tau = 300$	1.11 ± 3.03

TABLE II
MSE OF DIFFERENT METHOD. *mean ± std* [10^{-3}]

Method	ESN	LSTM	Linear
$\tau = 5$	0.46 ± 1.60	2.00 ± 6.19	2.27 ± 6.12
$\tau = 11$	0.01 ± 0.23	0.55 ± 1.96	1.30 ± 3.14
$\tau = 20$	0.01 ± 0.26	0.26 ± 0.96	1.51 ± 4.00

The means and standard deviations of the MSEs are shown in Table I. The prediction errors were larger in conditions with a smaller $\tau = 5$ in Random Conditions. This could be because if the input frequently changes with a smaller τ , the pose of the arm tends to be affected by the older control input. In such a case, a longer memory is needed to estimate the pose. Thus, it is more difficult to estimate the pose of a soft robotic arm with a smaller τ than with a larger τ in the Random Conditions. The size of τ does not relate to the size of the memory to estimate the pose in the Sine Conditions.

The errors in the Sine Conditions were smaller than those in the Random Conditions. This may be because the superimposed sine wave input in the Sine Conditions had temporal correlations and thus there was less uncertainty in the state.

B. Type of Regression

To evaluate the effects of nonlinear regression and the echo state networks, we compared the results of ESN (ESN) with those of the LSTM model (LSTM) and the linear model (Linear) in the Random conditions. For the linear model, the training data were approximated by the least-squares method. In the LSTM, the learning rate was set to 0.001, the number of intermediate layers was set to 1, the number of iterations was set to 20,000, and the number of nodes was set to 1,000. The model was trained using an Adam optimizer. These parameters of LSTM models were optimized in validation of the $\tau = 11$ conditions.

The means and standard error deviations of the MSEs are shown in Table II. For all τ conditions, the errors of LSTM were smaller than those of Linear, and those of ESN were smaller than those of LSTM. This suggests that the target function was approximated more accurately with nonlinear regression (ESN and LSTM) than with linear regression (Linear) and the errors of ESNs were smaller than those of LSTM for our conditions.

Training models of Linear, LSTM, and ESN in the $\tau = 11$ conditions took 0.01 s, 1716 s, and 661 s, respectively. The mean time of predicting 100 times using models of Linear, LSTM, and ESN in the conditions were 0.001 ms, 0.05 ms, and 0.05 ms, respectively. These results indicate that pose estimation using each model is fast enough to be used in real-time application. ESN learns the task faster than LSTM by updating k_{in} , ρ , and W_{out} while LSTM changes all the weights of networks.

TABLE III
MSE OF GENERALIZATION. *mean ± std* [10^{-3}]

ESN	LSTM	Linear
0.3 ± 0.8	1.3 ± 4.5	2.2 ± 5.3

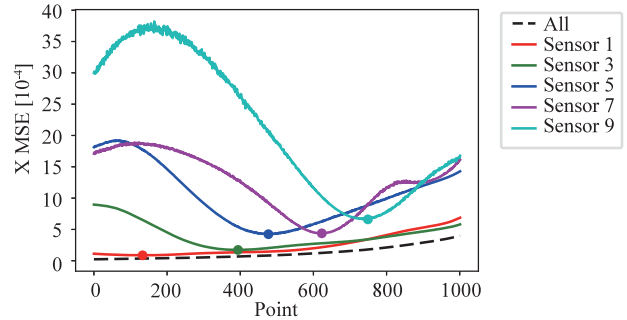


Fig. 7. The time averages of the MSEs when using only each bend sensor. Circles represent positions with the minimum error. The points are shown in Fig. 2.

C. Generalization

To clarify whether one set of the proposed models could estimate the pose of the soft robotic arm which receives control inputs with varying τ , the models were trained by including data obtained in different τ conditions. ESN, LSTM, and linear models were used as the models and compared. The models were trained using training samples from the Random Condition with $\tau = 5$, $\tau = 11$, and $\tau = 20$, and were tested using test samples under these conditions. LSTM models and linear models were trained as described in the previous sub-section.

The means and standard deviations of the MSEs are shown in Table III. The table shows that the errors of LSTM were smaller than those of Linear and those of ESN were smaller than those of LSTM. This result indicates that ESN can estimate the body pose of different input types with smaller errors than the Linear and LSTM models.

D. Pose Estimation Using One Bend Sensor

To investigate the contribution of the number and locations of the bending sensors in the soft arm to estimate the continuum-body-pose, we compared the prediction accuracy using only the outputs of a single bend sensor for \mathbf{u} in the Random Condition with $\tau = 11$.

Fig. 7 shows the time average of the MSEs of X_k when using all sensors, Sensor 1, Sensor 3, Sensor 5, Sensor 7, and Sensor 9, respectively. Sensor 1 is near the base and Sensor 9 is near the tip. This figure shows that the position with the minimum error moved from the base to the tip as the conditions changed from predicting based on the bend sensor at the base to predicting based on the bend sensor at the tip. This outcome suggests that each sensor provides the best estimation of the position nearest its embedded location in the arm. The figure also shows that the error in the tip position predicted by the sensor near the base was smaller than the error predicted by the sensor at the tip.

The reason why the MSEs using Sensor 1 near the base was smaller than those using Sensor 9 near the tip may be as follows. The motor on the base region moves the arm based on the control input. The movement of the arm is propagated from the base to the tip with a particular time delay. The control input determines the future pose of the arm. Sensor 1 near the base can receive the information of the current control input accurately from local bending of the arm. The RNNs can deal with memories of past inputs, and thus, they can estimate how the arm behaves in the tip region. However, Sensor 9 near the tip receives the past information of the control input. As such, it is difficult to estimate the going process in the base region of the arm currently. Furthermore, the information regarding the propagated control input along the arm would decay because of the perturbation from the waves in the water.

Interestingly, the MSEs of the 100th–500th positions using Sensor 1 were roughly the same as those of the MSEs using all sensors. This suggests that embedding a few bending sensors into the optimal locations of the soft robotic arm enables the estimation of the continuum-body-pose with approximately the same accuracy as embedding ten bending sensors.

E. Future Pose Estimation

The effects of the input commands appear at the base of the soft robotic arm at first, and move from the base to the tip, as shown in Fig. 5. Thus, the positions of the points far from the base of the soft robotic arm may be mainly determined by the past input commands and the past states, and the future pose may be estimated from the current and past input commands and states. To confirm whether the proposed model can estimate the future pose of the robot, we trained ESNs to predict the pose after i -steps ($i = 0, 1, 2, 4, \dots, 128$) in Random Conditions.

Fig. 8(a) shows the time average of the MSEs of X_k when predicting the pose at i steps in the future. This figure shows that the position with the minimum error moved from the base to the tip as the far future positions were predicted in the $i = 0, 1, 2, 4, 8, 16$ conditions. This is because the future input is randomly determined, and the future tip position is determined by the propagation of the current input from the base.

Fig. 8(b) shows that MSEs were saturated when predicting the pose after 64 steps. This indicates that the soft arm retains information about the previous inputs (approximately 64 input commands) in its state, which makes it possible for the external networks to predict its state only from the input stream within a certain range of time.

VI. DISCUSSION

The experimental results indicate that ESNs can estimate the continuum-body-pose of a soft robotic arm using embedded proprioceptive sensors. Pose estimation from partial observation enables the soft robotic arm to execute its tasks in dark places such as the deep sea [2], [25], space [26], and inside a human body [27]. The results of our ablation study can contribute to the development of a soft robotic arm with pose estimation for manipulation and locomotion in a dark place.

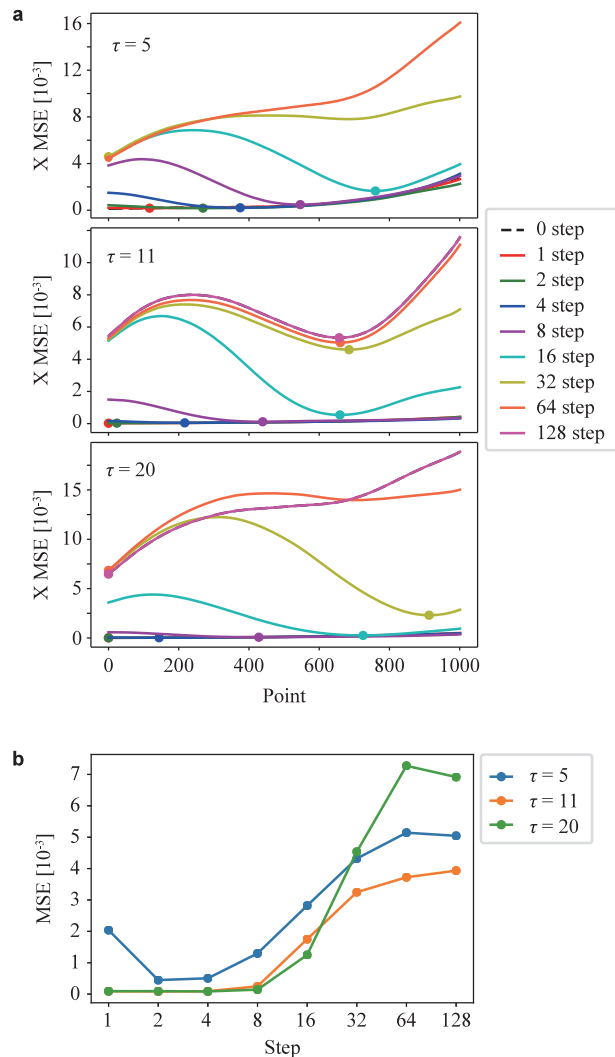


Fig. 8. (a) The time averages of the MSE when predicting future pose. Circles represent positions with the minimum error. (b) The time-and-position averages of the MSE for each future prediction step. The points are shown in Fig. 2.

In medical applications, a soft robot (a continuum robot) can be used for ophthalmic surgery [28], [29], single-port access surgery [30], [31], arthroscopy [32], and colonoscopy [33]. Although many medical soft robotic arms have been developed, the development of a method for shape sensing (*i.e.*, pose estimation) is still one of the main challenges for such robots [27]. Thus, our method for estimating the continuum-body-pose can contribute to a soft robot for medical applications.

In this study, the arm is moved with one motor in a plane. Using multiple motors enables the arm to move in a 3D space. In addition, attaching bending sensors to different directions on the circumference facilitates our pose estimation scheme of the arm to be workable in a 3D space.

Our approach can be extended in several ways. For example, although our final goal includes manipulating objects with a soft continuum arm, in this study we have only tested our scheme for passively driven cases, with no interaction or contact with external objects during movement. By adding training data and

including cases in which a soft robotic arm contacts external objects, our approach would be applicable in a straightforward manner for such cases.

Another direction would be to investigate the optimal network architecture for the RNN. For example, in the current study, our architecture outperformed the conventional LSTM. We speculate that this is due to the limited number of training datasets obtained in our experiments. When the training data is limited, or if there is a limit to the time spent for training (which is frequently the case in soft robotic applications) then RC is beneficial because there are fewer parameters to be tuned compared to the conventional LSTM. However, there may be other situations in which LSTM works better. Furthermore, there still exist many free parameters to be investigated for the setting of deep ESNs. Although we used a two-layered deep ESN for our experiments, the number layers for this architecture or the training scheme for it have many variations, and these variations should be tested according to each application scenario so that the appropriate setting can be selected.

VII. CONCLUSION

This letter presents a method for estimating the continuum-body-pose of a soft robotic arm using embedded proprioceptive sensors and validates this method experimentally. Our future work will be to investigate the optimal network architecture for the RNN and to develop a soft robotic arm using a method for manipulation and locomotion in dark places.

REFERENCES

- [1] C. Laschi, M. Cianchetti, B. Mazzolai, L. Margheri, M. Follador, and P. Dario, "Soft robot arm inspired by the octopus," *Adv. Robot.*, vol. 26, no. 7, pp. 709–727, 2012.
- [2] K. C. Galloway et al., "Soft robotic grippers for biological sampling on deep reefs," *Soft Robot.*, vol. 3, no. 1, pp. 23–33, 2016.
- [3] H. Tsukagoshi, A. Kitagawa, and M. Segawa, "Active hose: An artificial elephant's nose with maneuverability for rescue operation," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 3, 2001, pp. 2454–2459.
- [4] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker, "Soft robotics: Biological inspiration, state of the art, and future research," *Appl. Bionics Biomech.*, vol. 5, no. 3, pp. 99–117, 2008.
- [5] K. Nakajima, T. Li, H. Hauser, and R. Pfeifer, "Exploiting short-term memory in soft body dynamics as a computational resource," *J. Roy. Soc. Interface*, vol. 11, 2014, Art. no. 20140437.
- [6] K. Nakajima, N. Schmidt, and R. Pfeifer, "Measuring information transfer in a soft robotic arm," *Bioinspiration Biomimetics*, vol. 10, no. 3, 2015, Art. no. 035007.
- [7] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, 2002.
- [8] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks—with an erratum note," *Bonn, Germany: German Nat. Res. Center Inf. Technol. GMD Tech. Rep.*, vol. 148, no. 34, 2001, Art. no. 13.
- [9] K. Nakajima and I. Fischer, *Reservoir Computing: Theory, Physical Implementations, and Applications*. Berlin, Germany: Springer, 2021.
- [10] C. Gallicchio, A. Micheli, and L. Pedrelli, "Deep reservoir computing: A critical experimental analysis," *Neurocomputing*, vol. 268, pp. 87–99, 2017.
- [11] K. Nakajima, "Physical reservoir computing—an introductory perspective," *Japanese J. Appl. Phys.*, vol. 59, no. 6, 2020, Art. no. 060501.
- [12] T. G. Thuruthel, B. Shih, C. Laschi, and M. T. Tolley, "Soft robot perception using embedded soft sensors and recurrent neural networks," *Sci. Robot.*, vol. 4, no. 26, 2019, Art. no. eaav1488.
- [13] K. Chin, T. Hellebrekers, and C. Majidi, "Machine learning for soft robotic sensing and control," *Adv. Intell. Syst.*, vol. 2, no. 6, 2020, Art. no. 1900171.
- [14] J. Y. Loo, Z. Y. Ding, V. M. Baskaran, S. G. Nurzaman, and C. P. Tan, "Robust multimodal indirect sensing for soft robots via neural network-aided filter-based estimation," *Soft Robot.*, vol. 9, no. 3, pp. 591–612, 2022.
- [15] G. Soter, A. Conn, H. Hauser, and J. Rossiter, "Bodily aware soft robots: Integration of proprioceptive and exteroceptive sensors," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 2448–2453.
- [16] A. Tariverdi et al., "A recurrent neural-network-based real-time dynamic model for soft continuum manipulators," *Front. Robot. AI*, vol. 8, 2021, Art. no. 631303.
- [17] I. Van Meerbeek, C. De Sa, and R. Shepherd, "Soft optoelectronic sensory foams with proprioception," *Sci. Robot.*, vol. 3, no. 24, 2018, Art. no. eaau2489.
- [18] K. Nakajima, H. Hauser, T. Li, and R. Pfeifer, "Exploiting the dynamics of soft materials for machine learning," *Soft Robot.*, vol. 5, pp. 339–347, 2018.
- [19] K. Nakajima, H. Hauser, T. Li, and R. Pfeifer, "Information processing via physical soft body," *Sci. Rep.*, vol. 5, 2015, Art. no. 10487.
- [20] K. Inoue, Y. Kuniyoshi, K. Kagaya, and K. Nakajima, "Skeletonizing the dynamics of soft continuum body from video," *Soft Robot.*, vol. 9, no. 2, pp. 201–211, 2022.
- [21] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Netw.*, vol. 20, no. 3, pp. 335–352, 2007.
- [22] L. A. Thiede and U. Parlitz, "Gradient based hyperparameter optimization in echo state networks," *Neural Netw.*, vol. 115, pp. 23–29, 2019.
- [23] Z. Lu, B. R. Hunt, and E. Ott, "Attractor reconstruction by machine learning," *Chaos: An Interdiscipl. J. Nonlinear Sci.*, vol. 28, no. 6, 2018, Art. no. 061104.
- [24] M. Inubushi, K. Yoshimura, Y. Ikeda, and Y. Nagasawa, "On the characteristics and structures of dynamical systems suitable for reservoir computing," in *Reservoir Computing*, Berlin, Germany: Springer, 2021, pp. 97–116.
- [25] G. Li et al., "Self-powered soft robot in the mariana trench," *Nature*, vol. 591, no. 7848, pp. 66–71, 2021.
- [26] D. Nahar, P. M. Yanik, and I. D. Walker, "Robot tendrils: Long, thin continuum robots for inspection in space operations," in *Proc. IEEE Aerosp. Conf.*, 2017, pp. 1–8.
- [27] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1261–1280, Dec. 2015.
- [28] H. Yu, J.-H. Shen, K. M. Joos, and N. Simaan, "Design, calibration and preliminary testing of a robotic telemanipulator for OCT guided retinal surgery," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2013, pp. 225–231.
- [29] X. He, V. Van Geirt, P. Gehlbach, R. Taylor, and I. Iordachita, "IRIS: Integrated robotic intraocular snake," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2015, pp. 1764–1769.
- [30] J. Ding, R. E. Goldman, K. Xu, P. K. Allen, D. L. Fowler, and N. Simaan, "Design and coordination kinematics of an insertable robotic effectors platform for single-port access surgery," *IEEE/ASME Trans. Mechatron.*, vol. 18, no. 5, pp. 1612–1624, Oct. 2013.
- [31] K. Xu, J. Zhao, and M. Fu, "Development of the SJTU unfoldable robotic system (SURS) for single port laparoscopy," *IEEE/ASME Trans. Mechatron.*, vol. 20, no. 5, pp. 2133–2145, Oct. 2015.
- [32] M. D. M. Kutzer, S. M. Segreti, C. Y. Brown, M. Armand, R. H. Taylor, and S. C. Mears, "Design of a new cable-driven manipulator with a large open lumen: Preliminary applications in the minimally-invasive removal of osteolysis," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 2913–2920.
- [33] G. Chen, M. T. Pham, and T. Redarce, "Sensor-based guidance control of a continuum robot for a semi-autonomous colonoscopy," *Robot. Auton. Syst.*, vol. 57, no. 6–7, pp. 712–722, 2009.