Environmental and Structural Effects on Physical Reservoir Computing with Tensegrity

Kenichi FUJITA^{a,*1}, Syogo YONEKURA^a, Satoshi NISHIKAWA^a, Ryuma NIIYAMA^a, Yasuo KUNIYOSHI^{a,*2}

^aUniversity of Tokyo, Eng Bldg2 7-3-1 Hongo Bunkyo-ku, Tokyo 113-8656, Japan

1*Corresponding Author: fujita@isi.imi.i.u-tokyo.ac.jp

2*Corresponding Author: kuniyoshi@isi.imi.i.u-tokyo.ac.jp

Abstract

It is said that some natural animals outsource functions such as control of their bodies to their bodies themselves. This kind of calculation is called morphological computation. Physical reservoir computing, which uses a body itself as a neural network-like system, is one approach of this.

This paper describes the effects of body structures and environments around the bodies for computational abilities. In this research, tensegrity-like structure, designed by compressive elements and tensile elements which are made of springs, is used as a body structure. We demonstrated that when spring constant was set to be smaller and easy to move their computational ability rose. We also showed that when input amplitude, which is a length of nodes that input was given, was set to be smaller, computational ability rose. In addition, the environments around the body were shown to have large effects on computational ability. The results are summarized into four features. First, a softer body has higher computational ability. Second, input amplitude can control the past memories of the system. Third, the best input position exists which maximizes the computational ability. Fourth, the environments have effects on the system and its computational ability.

Keywords: Reservoir Computing, Morphological Computation, Body Structure.

1. Introduction

Conventionally, robots are made of hard materials such as metal and plastic, while animals in nature have bodies made of

soft materials such as soft skin and muscle. Using the soft materials, the animals in nature have behaved in multiple and complicated situations and highly adapted to their environments even if the environments have been drastically changing. The ways to control their bodies are also in contrast between the robots and animals. The former implements the body control by making the degrees of freedom decreasing to simplify the control structures and schemas, while the latter implements it by remaining higher degrees of freedom to adapt many unforeseen situations^(1,2). The bodies of the animals in nature are a kind of complicated and distributed computer systems that can behave and adapt to their environments by themselves by loosening and tightening their degrees of freedom in accordance with their living environments. For example, when we observe polyclad, octopus and ameoba, they use their bodies as computational resources that enable to create multiple dynamics of their movement⁽¹⁾. Their behavior seems to be more complicated than their simple nerve system. This type of calculation, in which a body itself is used as a computational resource, is called Morphological Computation⁽³⁾, or shortly called MC. Our research, described in this paper, focuses on Physical Reservoir Computing⁽⁴⁾, called PRC, which is a hot topic area in the MC researches. In the RPC researches, each piece of body is used as a small reservoir defined in Reservoir Computing, and works like a kind of neural network. The pieces keep highly adaptable to their environments, but most of the previous PRC researches have not considered general rules and usages of them, but examined just a few concrete examples. Therefore, they have not taken it consideration to a variety of effects to the environments and a variety of parameters to figure out their

Fig. 1. Reservoir Computing

(differently colored node in Input layer represents bias node)

Fig. 2. Icosahedral Tensegrity: Red lines are compressive elements, green lines are tensile elements and blue lines are input positions.

structures.

This research performed to assess the effects of the parameters of body structures and environments around structures in terms of the computational ability of PRC. In this research, tensegrity-like structure, which consists of compressive elements and tensile elements designed by spring and is often used for modeling biological structures^(5,6), is used as a reservoir of the network. In this research mainly two experiments were done to evaluate the effects of body structures and their environments. One is the evaluation of computational ability of the reservoir when its physical characters such as figures and some parameters are changed. Another is the evaluation of the effects of the environments. This experiment was done to compare the experiments of learning non-linear function and non-linear oscillator under with or without interaction with environments.

The result is important to create a new frame for artificial intelligence, especially for the control for soft robot, the robot made of soft materials.

2. Physical Reservoir Computing with Tensegrity

2.1 Physical Reservoir Computing

Reservoir computing, called RC, is a way of machine learning using the network consists of three layers such as input layer, recurrent layer, output layer as shown in Fig.1. There are

two important features in RC. First, the recurrent layer is randomly connected and these connections and weights between nodes do not change before and after learning. Such random connections give the network recurrent characteristic and make it possible to hold non-linearity and time delay. Second, different from other learning methods for neural networks, which usually change the all weights of the network for example by back propagation, only the weights between the recurrent and output layer (W_{out}) change to decrease the errors between target outputs and practical outputs with regression.

Conventionally, the recurrent layer of the RC is implemented on computer as imaginary nodes, although some researches replaces the imaginary nodes with physical structure. These researches are called Physical Reservoir Computing, called PRC⁽⁴⁾. Hauser *et al.* replaced nodes with material points, connections between nodes with springs, inputs into network with force to the material points and outputs of the network with the sum of the displacement of the springs. He showed this network could also work as reservoir for calculation and showed the network could learn non-linear limit cycle like Van der Pol oscillator if the feedback loop was added to the network⁽⁷⁾. In this research we used tensegrity, described below, as reservoir (recurrent layer of RC).

2.2 Tensegrity

Tensegrity is a stable structure that consists of tensile elements and compressive elements that are not adjacent but connected by the tensile elements, coined by Fuller from “Tensile” and “Integrity”⁽⁸⁾. Because of this stability, the structure is sometimes used for modeling biological structures like human body and cell^(5,6). In this research, we used Icosahedral Tensegrity for reservoir, which is one of the most general structures of tensegrities, as shown in Fig.2, and consists with six compressive elements shown as red line and 24 tensile elements shown as green line. In the Icosahedral Tensegrity, each end of the six compressive elements is located on each vertex of icosahedron and connected with tensile elements. K.Caluwaerts *et al.* used this structure as a reservoir and showed this structure can be used for PRC⁽⁹⁾.

2.3 Universal Computational ability

Universal Computational ability is defined by Maass *et al.* about the computational ability of the network⁽¹⁰⁾, which

guarantees the network has universal computational ability if the following two conditions are satisfied. The first requirement is that the network can learn arbitrary mapping which is independent from internal statement. The second requirement is that it can learn arbitrary mapping which is dependent on internal statement. These are examined by showing the network can learn and map non-linear functions and non-linear oscillation. This method is a major way to examine the ability of recurrent neural network or RC. Therefore, we also used these two tasks to examine the ability of the recurrent layer.

3. Setup for Reservoir Computing with Tensegrity

3.1 Non-linear function

We can examine the first requirement for the universal computational ability, whether the network can implement arbitrary mapping without any cyclic internal states for the input, by checking whether the network can possess the ability of expressing non-linear function. This mapping is not to be implemented, if the network does not have enough non-linearity and hold the effects of past input.

Input is given between nodes which are connected neither by compressive elements nor tensile elements, shown as blue lines in Fig.2, as a displacement of the length between nodes. In other words, when the length of the initial state between nodes, shown as blue lines, is L and input is $Input(t)$, now the length of the blue line is $L + input(t)$.

$$Input(t) = Gain \ln(t) \quad (1)$$

$$\ln(t) = 0.2 \sin(2\pi f_1 t) \sin(2\pi f_2 t) \sin(2\pi f_3 t) \quad (2)$$

In Eq.2 parameters are the same to the value of Hauser's. $f_1=2.11$, $f_2=3.73$ and $f_3=4.33$.

When each displacement of 24 tensile materials is $x_i(t)$ ($i = 0, \dots, 23$) and each velocity of displacement is $v_i(t)$ ($i = 0, \dots, 23$), output is given in Eq.3, where $w_{1,i}$ and $w_{2,i}$ are weights for displacement and velocity of displacement.

$$Out(t) = \sum_{i=0}^{23} w_{1,i} x_i(t) + \sum_{i=0}^{23} w_{2,i} v_i(t) \quad (3)$$

In RC, values of $w_{1,i}$ and $w_{2,i}$ are calculated by linear regression in Eq.8 and Eq.9, where learning is done during time t_1 to t_2 and $W, x(t), X, O$ are defined as Eq.4 to Eq.7.

$$W = [w_{1,1}, w_{1,2}, \dots, w_{1,23}, w_{2,1}, w_{2,2}, \dots, w_{2,23}] \quad (4)$$

$$x(t) = [x_1(t), x_2(t), \dots, x_{23}(t), v_1(t), v_2(t), \dots, v_{23}(t)]^T \quad (5)$$

$$X = [x(t_1), x(t_1 + 1), \dots, x(t_1 + t_2)] \quad (6)$$

$$O = [Out(t_1), Out(t_1 + 1), \dots, Out(t_1 + t_2)] \quad (7)$$

$$O = WX \quad (8)$$

$$W = OX^T(XX^T)^{-1} \quad (9)$$

In this research, the target output ($Out(t)$) is the function shown in Eq.10 ($y(t)$) which has chaotic output and was also used by Hauser and Nakajima to examine PRC^(1,4). To learn this mapping, reservoir needs non-linearity enough to hold the effects of past inputs, which decrease the size of effect as time passed. Therefore, this function is suitable for evaluating the computational ability of PRC. In Eq.10, n shows that n times of the past outputs have effect for making the next output. This function is called (n)th order system.

$$\begin{aligned} y(t+1) = & 0.3y(t) + 0.05y(t) \sum_{i=0}^{n-1} y(t-i) \\ & + 1.5\ln(t-n)\ln(t) + 0.1 \end{aligned} \quad (10)$$

3.2 Nonlinear oscillation

We can examine whether the network can learn arbitrary mapping with internal states by checking whether the network can learn mapping of non-linear oscillator, which is the second requirement for universal computational ability. Different from non-linear function in 3.1, learning non-linear oscillation has 2 inputs and 2 outputs and the outputs of the network are used for the inputs of the next time step (feedback loop). The outputs are calculated by Eq.13, where $w_{1,1,i}, w_{1,2,i}, w_{2,1,i}, w_{2,2,i}$ are weights.

$$\begin{cases} Input1(t) = \ln1(t) \\ Input2(t) = \ln2(t) \end{cases} \quad (11)$$

$$\begin{cases} \ln1(t) = Output1(t-1) \\ \ln2(t) = Output2(t-1) \end{cases} \quad (12)$$

$$\begin{cases} Output1(t) = \sum_{i=0}^{23} w_{1,1,i} x_i(t) + \sum_{i=0}^{23} w_{2,1,i} v_i(t) \\ Output2(t) = \sum_{i=0}^{23} w_{1,2,i} x_i(t) + \sum_{i=0}^{23} w_{2,2,i} v_i(t) \end{cases} \quad (13)$$

The way of learning is the same as that in 3.1. In the learning period of non-linear oscillator, values of the target function are given as training data and after the learning is finished, the inputs are took place by the values which are made from the outputs of the last step of the network. We used Van der Pol oscillator (Eq.14) in this research as a non-linear oscillator⁽⁷⁾, which is deeply related to acquirements of Central Pattern

Fig. 3. Spring constant and MSE

Generators.

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_1 + (1 - x_1^2)x_2 \end{cases} \quad (14)$$

We used the fourth-order Runge-Kutta method because this differential equation cannot be solved analytically and white noise was added on $x_i(t)$ and $v_i(t)$ to avoid over fitting.

3.3 Evaluation

To evaluate experimental results, we used Mean Square Error (MSE) calculated with Eq.15 where the time steps of the test is $TestLength$, output is $O(t)$ and target is $Target(t)$. In this calculation, the smaller MSE represents the smaller error and the better result.

$$MSE = \frac{\sum_{t=1}^{TestLength} (O(t) - Target(t))^2}{Testlength} \quad (15)$$

In the oscillation task, we examined whether it obeys some cyclic patterns without divergence.

4. Experimental Result

4.1 Structural effects on calculation

4.1.1 Effects of body parameters

Learning experiments were done to examine its computational ability with tensegrity for non-linear function in which we change the values of the parameters in structure such as spring constant and amplitude (Size of Gain in Eq.1). In addition to that, we examined their dependency on tasks by changing n in Eq.10.

Parameter values are shown in Table 1. In the spring constant changing experiment (experiment 1), damping coefficient was set to 20 [kg/s] and amplitude was set to 1.36 [m]. In the gain changing experiment (experiment 2), spring constant was set to 600 [kg/s²], damping coefficient was set to 20 [kg/s]. In this section, actuation is given between node 3 and 10, node 1 and 6, and node 0 and 5 (Node number follows

Fig. 4. Input's amplitude and MSE

Fig.2).

The result of experiment 1 is shown in Fig.3. In all tasks the larger spring constant resulted in the larger MSE. The changing rate was bigger when n in Eq.10 was bigger. In addition to that when spring constant exceeded 700 [kg/s²], the rate of changing MSE became the highest during 400 [kg/s²] and 700 [kg/s²]. It can be said that structure that is easy to move has high computational ability because larger spring constant means that it is hard for structure to move freely.

The result of experiment 2 is showed in Fig.4. MSE dropped down to about 1×10^{-5} , when amplitude became 0.2 – 0.7. After that, MSE gradually increased. Especially in the 20th order system, the rate of increasing was bigger than others. The 20th order system shows there was a bottom of MSE between 0.1 – 0.2 (amplitude). As a result, when the amplitude is bigger than 0.7, only the 20th order system's MSE became larger than others. This means that when the amplitude is large, the effects of the past inputs are cancelled by the new large inputs and it is hard for network to hold the memory of the past inputs.

4.1.2 Effects of Position and Length of materials

This experiment examined whether tensegrity's figures have effects on computational ability. With the 1000 different tensegrity figures (Fig.5b), which had 6 compressive elements and 24 tensile elements, are connected like Icosahedral Tensegrity, the positions of nodes were set on the surface of

Table 1. Value of parameters

exp	Spring constant (kg/s ²)	Dumping coefficient (kg/s)	Amplitude (m)
Exp1	400-1000	20	1.36
Exp2	600	20	0.1-1.35

Fig. 5. Various Tensegrities

Fig. 6. MSE's map when one node moves: These figures shows the change of MSE when the position and length of the compressive element is changed. x-y is the position of the changed node. Difference between (a) and (b) is the nodes input is given between them. MSE depends on the where the inputs are given.

2.0m cube randomly. To avoid physical conflict, the positions of the input nodes were the same as Icosahedral Tensegrity. In this experiment the amplitude of the inputs was 1.2m, spring constant was $475 \text{ [kg/s}^2\text{]}$ and damping coefficient was 1 [kg/s] . The histogram of MSE of the 10th order system is shown in Fig.5b, where MSE of Icosahedral Tensegrity was 2.371×10^{-5} . Though multiple regression was done to analyze the relation between positions, no relation was found between them.

To simplify the problem, another experiment was done. In this experiment the only one position of node (node 3 in Icosahedral Tensegrity shown in Fig.2) was moved on the plane where node 3 and node 1 were first on and perpendicular to the compressive material which respectively connect to node 0 and node 1. Figure 6a shows the results when inputs were given between node 6 and 11, and node 3 and 8. Figure 6b shows the result when inputs are given between node 6 and node 11, and between node 2 and node 8. These Figures plot MSE as a value of z axis. X axis and y axis are the position of the node. As these figures shows, the distribution of MSE

Fig. 7. Nonlinear function

was totally different when the input positions were differently given. These show the computational ability of tensegrity depends on the relation of positions of inputs and compressive materials.

4.2 Effects of Environment

4.2.1 Non-linear function

Two experiments of learning non-linear functions were done to assess the effects of the environments. In each situation, tensegrity was put in the environment with and without interaction to the ground and this tensegrity was used as reservoir of PRC, and 10th order system was used as a target function, when spring constant was $475 \text{ [kg/s}^2\text{]}$, damping coefficient was 20.0 [kg/s] , and amplitude was 0.4 [m] . Inputs were given between node 3 and 10, node 1 and 6, and node 0 and 5. Results are shown in Fig.7. The red line in Fig.7 shows result without environmental effect and the green line shows result with environmental effect. In the former case, MSE was 3.45503×10^{-5} . This means the error between target and output was about 0.006. Because the value of 10th order system is about 0.15 to 0.25, 0.006 is about 10% and this is small enough. In the latter case MSE was 2.05×10^{-4} . The reason why this is bigger than the former one is that when tensegrity hits the ground, the output from the reservoir was highly affected and error became bigger. This surge of error can be seen in Fig.7 between 43 [s] and 47 [s].

4.2.2 Non-linear oscillator

We examined whether tensegrity can learn non-linear oscillator with and without interaction to environments. In the experiment without interaction, spring constant was $466 \text{ [kg/s}^2\text{]}$, damping coefficient was 20 [kg/s] . Input x_1 was given between node 6 and 11, and node 3 and 8. x_2 was given between node 1 and 7, and node 6 and 10. The result is shown

Fig. 8. Nonlinear oscillator

Fig. 9. Frequency of nonlinear oscillator: These figures shows the effects of the environments. (a) shows the change of cycle. When environmental effects are added the peak of frequency is changed. (b) shows the how environment effects the cycle. The horizontal axis of (b) shows reflection coefficient of the ground.

in Fig.8. Experiment without interaction to environment is drawn as red line and with interaction is as green line. As it showed in Fig.8, the network could express Van der pol oscillator more than 70 [s], though sometimes it deviated from the cyclic oscillation even the same parameters were used. On the other hand, at the time there was interaction with environment, oscillator converged into different one from target. Amplitude of x_1 of this oscillator was -1 to 2, while in target, the amplitude is -2 to 2. That of x_2 was -0.5 to 2, while in target, the amplitude is -3 to 3. Fig.9a shows the result of Fourier transform of these oscillator. In this figure, a red line shows oscillator with interaction, a green line shows without interaction and blue line shows target cycle. When interaction was added, the peak of the graph moves to right compared to target. This means the cycle becomes shorter than the target. Fig. 9b shows the average frequency of converged oscillator when reflection coefficient was changed. This shows the reflection coefficient gives influences on the system. Changing of the way of interaction between Tensegrity and the ground changes the dynamics of it and this finally leads to different

Fig. 10. Human body-like structure

Fig. 11. Oscillator with body-like model

oscillation.

Learning nonlinear oscillator is not limited ability to icosahedral tensegrity. We made another type of model such as Fig.10, which is human body-like tensegrity based on anatomical knowledge⁽¹¹⁾. The result is shown in Fig.11. Compare to the result of icosahedral tensegrity, the figure of oscillator become similar to original one especially the value of x_1 , because the number of points, which has connection with the ground, is smaller than icosahedral one and the environmental effects became smaller.

5. Discussion

The important points of this research is that body parameters and environment has large effects on the computational abilities. Especially, the result that softer body structure has higher computational abilities is critical for adapting PRC to the soft robots, because soft robots tend to get their adaptability for the environments with their soft bodies. To think their adaptability, the environmental effects found in this research is also important, since almost all the past researches about PRC did not emphasis environmental effects. For example, Nakajima *et al.* used octopus arm robot moving under the water for the reservoir⁽¹⁾, but because there were no comparison of changed environmental conditions, his research was not enough to discuss the environmental effects. In these

past researches, learning and testing were done in a steady state but in this research the states were actively changed the contact with the ground and this contact changed dynamics and led to the change of the converged cycle. This changing suggests that if the body structures like tensegrity is used for brain, some other body structures such as Fig.10 or other learning method which counteract environmental effects are needed, without doing so, tensegrity cannot learn specific functions. On the other hand, from the view of morphological computation, environmentally modulated outputs can be used for adaptive behavior like a switch which changes behavior depending on environments.

In addition to these technical aspects, there are some useful aspects for biology. For example, Ingber's idea⁽⁶⁾ is thought to be difficult to demonstrate, though there is a possibility that the cell is tensegrity structure. However, if some mechanism of cells could be connected to the reservoir computing Ingber's idea can be tested. To connect natural animal's mechanism and reservoir computing can lead to deeper understanding of thinking about body structure and environmental effects.

To make more useful system with PRC, unsupervised learning or other methods such as generic algorithm should be introduced. In this research and the past researches, the only supervised learning was used as a method, however if this system are used in the real world, the system have to adapt to unknown changing world and supervised learning is not enough.

6. Conclusions

This paper confirmed body parameters such as spring constant and amplitude have large effects on computational ability on function mapping of PRC with tensegrity. These parameters are related to the dynamics of tensegrity. The experiment demonstrated the more flexible dynamics of the tensegrity make the higher computational ability and larger amplitude cancel past input effects. In addition to that, positions of compressive elements and inputs have also large effects on computational ability. To take it consideration to environmental effects, the interaction with the ground changed the system and led to different oscillator. Especially, the way of interaction with the ground had large effect on the result. These results can lead to a new method for controlling robot,

especially for soft robot and lead to understand biological system.

Acknowledgement

This work was supported in parts by NEDO Next Generation AI & Robotics Core Technology Development P15009-15101156-0 and JSPS KAKENHI Grant Number JP15H05320

References

- (1) K. Nakajima, H. Hauser, R. Kang, E. Gunglielmino, D. G. Caldwell, R. Pfeifer, "A soft body as a reservoir: case studies in a dynamic model of octopus-inspired soft robotic arm," *FRONT COMPUT NEUROSC*, vol.7-91, 2013.
- (2) K. Nakajima, H. Hauser, R. Kang, E. Gunglielmino, D. G. Caldwell, and R. Pfeifer. Computing with a muscular-hydrostat system. *ICRA*, pp. 1504–1511, 2013.
- (3) V. C.Muller and M. Hoffmann. What is morphological computation? on how the body contributes to cognition and control. *Artificial Life*, Vol. 23, No. 1, p. forthcoming, 2016.
- (4) H. Hauser, A. J. Ijspeert, R. M. Fuchslin, R. Pfeifer, W. Maass, "Towards a theoretical foundation for morphological computation with compliant bodies," *BIOL. CYBERN*, vol.105, No. 5, pp.355-370, 2011
- (5) R. L. Swanson, "Biotensegrity: A Unifying Theory of Biological Architecture With Applications to Osteopathic Practice, Education, and Research—A Review and Analysis," *JAOA*, vol.113, No. 1, pp.34-52, 2013.
- (6) D. E. Ingber, N. Wang, D. Stamenovic, "Tensegrity, cellular biophysics, and the mechanics of living systems," *REP PROG PHYS*, vol.77, No. 4, 2014.
- (7) H. Hauser, A. J. Ijspeert, R. M. Fuchslin, R. Pfeifer, W. Maass, "The role of feedback in morphological computation with compliant bodies," *BIOL. CYBERN*, vol.106, No. 10, pp.595-613, 2012.
- (8) B. Fuller, "Synergetics Explorations in the geometry of thinking," Macmilan Publishing Co., 1975.
- (9) K. Caluwaerts, M. D' Haene, D. Verstraeten, and B. Schrauwen. Locomotion without a brain: physical reservoir computing in tensegrity structures. *Artificial Life*, Vol. 19, No. 1, pp. 35–66, 2012
- (10) W. Maass, "Liquid state machines: Motivation, theory, and

applications,” Imperial College Press, pp.275-296, 2010.

(11) Intensiondesignes.

<http://www.intensiondesigns.com/>. Accessed:
2017-06-08