International Journal of Applied Electromagnetics and Mechanics 52 (2016) 883–889
DOI 10.3233/JAE-162206
IOS Press

Validity analysis of chaos generated with Neural-Network-Differential-Equation for robot to reduce fish's learning speed

Haruhiro Inukai, Mamoru Minami[*] and Akira Yanou
Graduate school of natural science and technology, Okayama University, Okayama-shi, Japan

Abstract. We have been investigating a method to compose some intelligent robots. Continuous catching and releasing experiment for fish induces the fish to find some escaping strategies staying at corners of a pool. To make fish-catching robot more intelligent than fish's adapting and escaping abilities, we have proposed a chaos-generator comprising Neural-Network-Differential-Equation (NNDE) and an evolving mechanism to have the system generate chaotic trajectories as many as possible. We believe that the fish could not be adaptive enough to escape from chasing net with chaos motions since unpredictable chaotic motions of net may go beyond the fish's adapting abilities to the net motions. In this report we confirmed that four type of chaos are generated by NNDE and one of them is valid for decreasing the fish's learning velocities, which we judged that could be thought that the chaos increased the robot's intelligence relatively against fish's escaping intelligence.

Keywords: Chaos, neural network, intelligent robot

1. Introduction

Animal world has been used conceptually by robotics as a source of inspiration for machine intelligence. For the purpose of studying animal behavior and intelligence, the model of interaction between animals and machines is proposed in researches like [1]. Crucial characteristic of machine intelligence is that the robot should be able to use input information from sensor to know how to behave in a changing environment and furthermore can learn from the environment for safety like avoiding obstacle. As known universally the robot intelligence has reached a relatively high level, still the word "intelligence" is an abstract term, so the measurement of the intelligence level of a robot has become necessary. A practical and systematic strategy for measuring machine intelligence quotient (MIQ) of human-machine cooperative systems is proposed in [2].

In our approach, we evaluate the intelligence degree between fish and the robot by Fish-Catching operation. We considered that the antagonistic relationship can be very meaningful as one way to discuss robotic intelligence. By evolutionary algorithms, visual servoing and Object Recognizing based on the input image from a CCD camera mounted on the manipulator has been studied in our laboratory (Fig. 1) [3], and we succeeded in catching a fish by a net attached at the hand of the manipulator based on

[*]Corresponding author: Mamoru Minami, Graduate School of Natural Science and Technology, Okayama University, 3-1-1 Tsushima-naka, Kita-ku, Okayama-shi 700-8530, Japan. E-mail: minami-m@cc.okayama-u.ac.jp.

the real-time visual tracking under the method of Gazing GA [4] to enhance the real-time pose-tracking ability.

Through above experiments, we have learned that it is not effective for fish catching to simply pursue the current fish position by visual servoing with velocity feedback control. Actually, the consistent tracking is sometimes impossible because the fish alter motion pattern suddenly maybe under some emotional reasons of fear. Those behaviors are thought to be caused by emotional factors and they can also be thought as a kind of innate fish intelligence, even though not in a high level. While observing the fish's adapting behavior to escape in the antagonistic relationship with the robot, that is continuous catching/releasing experiments, we found that we can define a "Fish Learning Speed" (FLS) representing decreasing velocity of fish number caught by the robot through continuous catching/releasing operation.

Through this measure we can compare the innate intelligence of the fish and the intelligence of the robot.

It has been well known that many chaotic signals exist in our body, for example, in nerves, in motions of eye-balls and in heart-beating periods [5]. Therefore we thought that imitating such animal's internal dynamics and putting chaos into robots have something meaningfulness to get the robot more intelligent. We embed chaos into the Robot Dynamics in order to supplement the deficiency of our Fish-Catching system [6]. Therefore what we have to pay attention to the fish's nature that the fish does continue to conceive always escaping strategy against new stressing situation. This means that robot's intelligence to override the fish's thinking ability needs infinite source of idea of catching motions.

To generate such catching motion, we have proposed Neural-Network-Differential-Equation (NNDE) [7] that can generate plural chaos and inherently have a possibility to be able to generate infinite varieties of chaos, derived from the neural network's ability to approximate any nonlinear function as accurate as with desirable precision [8].

We actually applied a one of chaos generated by NNDE into the robot's net motion and confirmed it is valid for decreasing the fish's learning speed and then it can be thought that the chaos increased the robot's intelligence relatively against fish's escaping intelligence.

2. Problem of Fish-Catching

To compare intelligence between fish and the robot, we kept a procedure to catch a fish and release it immediately continuously for 30 minutes in once operation.

We released 5 fishes (length is about 40[mm]) in the pool in advance, and once the fish got caught, it would be released to the same pool at once. Based on the idea that the fish may get tired as time passing, we had expected that the capturing operation would become smoother.

But to our astonishment, the number of fishes caught decreased gradually.

The reason that catching number decreased may lie in the fish learning ability. For example, the fish learn how to run away around the net by circular swimming motion with about constant velocity, having made a steady state position error that the net cannot reach to the chasing fish. Or the fish stay in the opposite corner against the net in the pool. And also, the fish keep staying within the clearance between the edge of the pool and the net where the net is inhibited to enter. To solve these problems, and to compose more intelligent fish catching systems, we thought that the net's chaos behavior with many chaotic variety can be overcome those fish's escaping intelligence, since huge variety of chaos trajectories seems to be unpredictable for the fish to adapt them. This strategy to overcome fish's adaptive intelligence is based on a hypothesis that unpredictability of the motion of the chasing net produced by plural chaos can make the fish's learning logic confuse. Then we propose Neural-Network-Differential-Equation to generate chaos as many as possible.

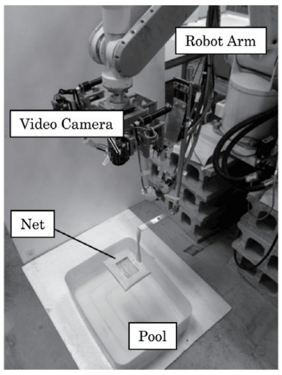

Fig. 1. Experiment system.

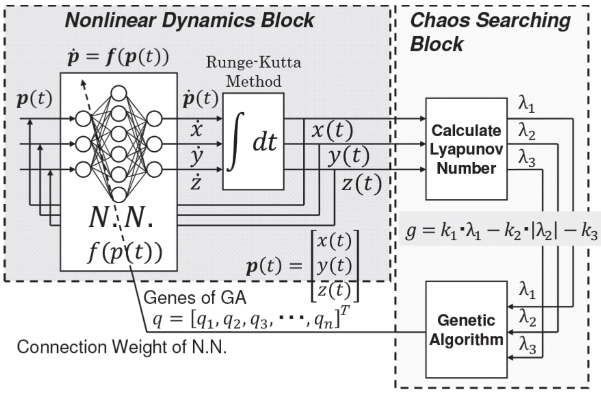

Fig. 2. Block diagram of chaos generation.

3. Fish Lerning Speed

To evaluate numerically how fast the fish can learn to escape the net, we adapted Linear Least-Square approximation to the fish-catching decreasing tendency, resulting in FLS $= -0.30$ as shown in Fig. 6 VS, which exhibit the number of fish caught by the robot in five minutes, on condition of the caught fish released into the same pool immediately. The decreasing coefficient -0.30 represents adapting or learning speed of the fish as a group when the fish's intelligence is compared to robot's catching ability. We named the coefficient as "*Fish Learning Speed*" (FLS), since the decreasing tendency that is the value of coefficient can represent the fish's learning speed to conceive a new escaping strategies – stay at corner or swim with constant speed on a circle trajectory.

4. Chaos generate system

We proposed a new chaos generator using N.N.. In the chaos generator we proposed, nonlinear differential equation is expressed with N.N.. N.N. has been proven to have an ability to approximate any nonlinear functions with arbitrarily high accuracy [8]. Including the function expressed by N.N. in a differential equation, a nonlinear function part of the nonlinear differential equation can be changed variously. Considering N.N. which has input layer, hidden layer and output layer and has nonlinear mapping $\boldsymbol{p}(t) = [x(t),\, y(t),\, z(t)]^T$ to $\boldsymbol{f}(\boldsymbol{p}(t))$. The N.N. output is $\dot{\boldsymbol{p}}(t) = [\dot{x}(t),\, \dot{y}(t),\, \dot{z}(t)]^T$ and $\boldsymbol{p}(t)$ is obtained by integrating $\dot{\boldsymbol{p}}(t)$ with Runge-Kutta method. A closed loop system is composed by feedback $\boldsymbol{p}(t)$ to the N.N. input and this system represents $\dot{\boldsymbol{p}}(t) = \boldsymbol{f}(\boldsymbol{p}(t))$. A block diagram which represents $\dot{\boldsymbol{p}}(t) = \boldsymbol{f}(\boldsymbol{p}(t))$ as N.N. is shown in Fig. 2. Here, a method that search the N.N. weight coefficients which generate chaos is introduced. Figure 2 represents the block diagram that finds chaos by using Genetic Algorithm(GA). The GA evolves genes representing a vector $\boldsymbol{q_i} = [q_{1i}, q_{2i}, \ldots, q_{ni}]^T$ which means N.N. weight coefficients quantity to search N.N. weight coefficients maximizing g_i defined as $\dot{\boldsymbol{p}}(t) = \boldsymbol{f}(\boldsymbol{p}(t))$, $g_i = k_1 \cdot \lambda_1 i - k_2 \cdot |\lambda_2 i| - k_3 \cdot \lambda_3 i$. where k_1, k_2 and k_3 are positive coefficients. To evaluate $\boldsymbol{q_i}$, $\boldsymbol{q_i}$ is set to N.N. as weight coefficients and $\dot{\boldsymbol{p}}(t) = \boldsymbol{f}(\boldsymbol{p}(t))$ is solved by numerical integration and solved trajectory $p_i(t)$ appears. In addition, Lyapunov numbers $\boldsymbol{L} = [\lambda_1, \lambda_2, \lambda_3]^T$, $(\lambda_1 > \lambda_2 > \lambda_3)$ of the trajectory are calculated. Evolution of GA tries to maximize the fitness function g_i. The relationship between positive and negative Lyapunov spectrum is $(+, 0, -)$ and g_i is composed to get large value when $\lambda_1 i$, $\lambda_2 i$, $\lambda_3 i$ correspond with Lyapunov spectrum $(+, 0, -)$. g_i maximization by GA leads NNDE

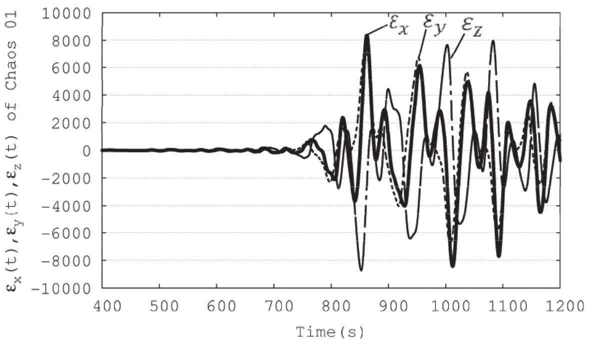

Fig. 3. Sensitivity of initial value (Chaos 01).

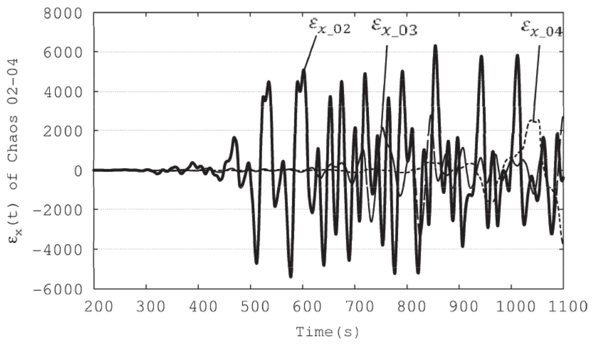

Fig. 4. Sensitivity of initial value (x-coordinate).

to have chaos trajectory. This procedure is repeated many times and chaos trajectory can be generated by searching a trajectory which satisfies Lyapunov number of chaos with GA. In this paper, N.N. consists of three layer consisted of 3 input, 6 intermediate, 3 output neurons, and the number of weights of N.N. is 36.

5. Chaos verification

In this section, Lyapunov number, Poincare return map, Sensitivity of initial value and Fractal dimension are introduced as indices of chaos.

5.1. *Lyapunov number*

As one of criteria to evaluate a chaos' character of time function $\boldsymbol{f}(\mathrm{t})$ at discrete time t_i in time domain, Lyapunov number expressed by the following equation is well known, $\lambda = \lim_{N\to\infty} \frac{1}{N} \sum_{i=0}^{N-1} \log |f'(t_i)|$, where positive value can represent that the irregular oscillation diverts from a standard trajectory, which expands like exponential function

5.2. *Poincare return map*

Poincare return map verifies whether trajectory has a structure of stretching and folding. Stretching means a trajectory goes away from a point of convergence and folding means a trajectory is brought back to a point of convergence. This structure is a basic chaos property.

5.3. *Sensitivity of initial value*

The small perturbation of the current trajectory may lead to significantly different future behavior. Sensitivity of initial value is popularly known as the "butterfly effect". This structure is a basic chaos property too.

5.4. *Fractal dimension*

In this subsection, Fractal dimension is introduced as an index of chaos. Fractal dimension represents self-similarity of chaos attractor and if Fractal dimension of a trajectory is not integer, it can be said that the trajectory has self-similarity.

Table 1
Character of chaos 02 $\sim$ 04

	Chaos 02	Chaos 03	Chaos 04
	0.01919	0.015934	0.01208
Lyapunov number	−0.00733	−0.002172	−0.00143
	−0.10379	−0.123026	−0.075448
Fractal dimension	1.78474	1.89099	1.8799
Poincare return map	Fig. 5(b)	Fig. 5(c)	Fig. 5(d)
Sensitivity of initial value		Fig. 4	

6. Chaos verification result

So far we have found four chaos patterns with different weight coefficients explored by GA mentioned in the previous section. We named them with serial numbers as chaos 01 $\sim$ 04. The followings are verifications of these chaos with each individual characters.

6.1. Chaos 01

In this subsection, chaos 01 is verified whether the trajectory has chaos properties or not from viewpoints of Lyapunov number, Sensitivity of initial value, Poincare return map and Fractal dimension.

Firstly, Lyapunov numbers of chaos01 are examined. Lyapunov numbers are $\lambda_1 = 0.014585$, $\lambda_2 = -0.003314$ and $\lambda_3 = -0.165381$. These corresponded to the Lyapunov spectrum of chaos, $(+, 0, -)$. Next, Poincare return map is examined.Poincare section is put at $x - z$ plane($x < 0$) and a difference between the origin and trajectory intersection with poincare section is defined as r. Chaos 01's poincare return map appear in Fig. 5(a). One dimensional map can be seen in the figure, from which we can understand that the map represents streaching (left half of the Fig. 5(a)) and folding (right half) that are essential characters to generate chaos.

Next, Sensitive of initial value is examined. Here, a difference between trajectories with minutely different initial value are shown as ε_x, ε_y, ε_z. Initial values of a trajectory ($x_1(t)$, $y_1(t)$, $z_1(t)$) are set as $x_1(0) = 1.00$, $y_1(0) = 1.00$, $z_1(0) = 1.00$ and initial value of another trajectory ($x_2(t)$, $y_2(t)$, $z_2(t)$) are set as $x_2(0) = 1.01$, $y_2(0) = 1.01$, $z_2(0) = 1.01$. ε_x, ε_y, ε_z are defined as $\varepsilon_x = x_1(t) - x_2(t)$, $\varepsilon_y = y_1(t) - y_2(t)$, $\varepsilon_z = z_1(t) - z_2(t)$. Figure 3 shows ε_x, ε_y, ε_z of chaos 01. Because ε_x, ε_y, ε_z are almost zero from 0 s to 400 s, they are not shown until 400 [s]. We can see from Fig. 3 that the difference between the two trajectories with minute difference of initial values diverts suddenly about 800 seconds having passed, this means the slight different initial values make large separation with each other, indicating sensitivity of initial value, which is one of the character of chaos.

Lastly, Fractal dimension is examined. Fractal dimension of chaos 01 is 1.36058, which is non integer and chaos 01 has self-similarity. Therefore, the chaos property of chaos 01 has been verified from the viewpoint of Lyapunov number, Sensitivity of initial value, Poincare return map and Fractal dimension.

6.2. Chaos 02 $\sim$ 04

Verifications of chaos 02 $\sim$ 04 are summarized at Table 1. Lyapunov number column represent λ_1, λ_2, λ_3 from the top to bottom and Lyapunov spectrums of chaos 02 $\sim$ 04 are confirmed as $(+, 0, -)$. And also the table shows Fractal dimensions of chaos 02 $\sim$ 04 are non integer and thus chaos 02 $\sim$ 04 have self-similarity. Poincare return maps are shown in Figs 5(b), (c), (d) and one dimensional map can be seen in these figures. Time-profile of differences between trajectories with minutely different

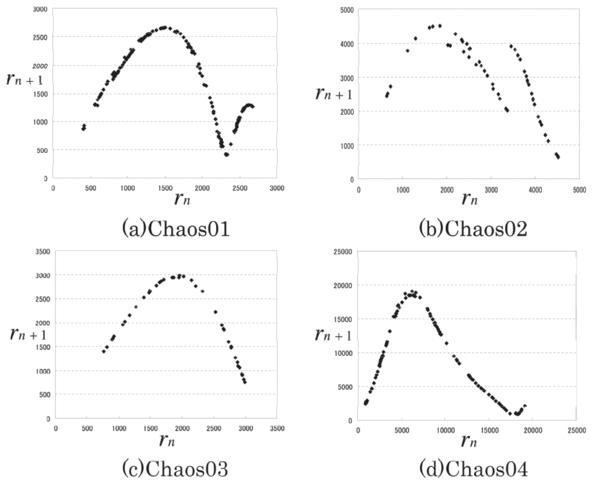

Fig. 5. Poincare returnmap.

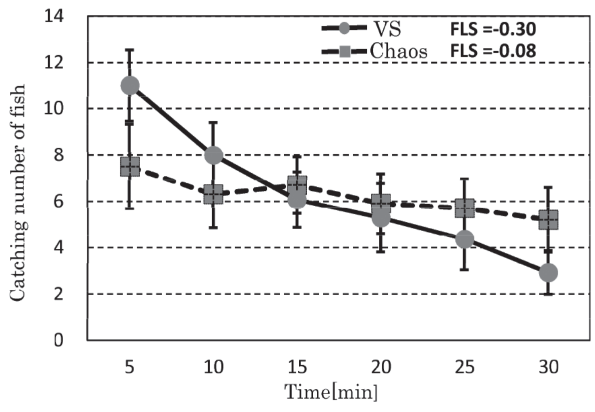

Fig. 6. Result of experiments.

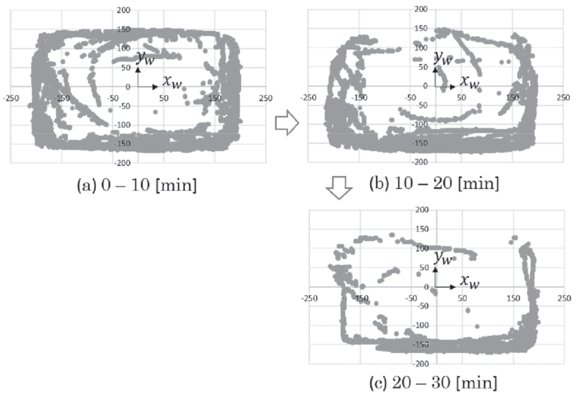

Fig. 7. Recognized position in visual serovoing.

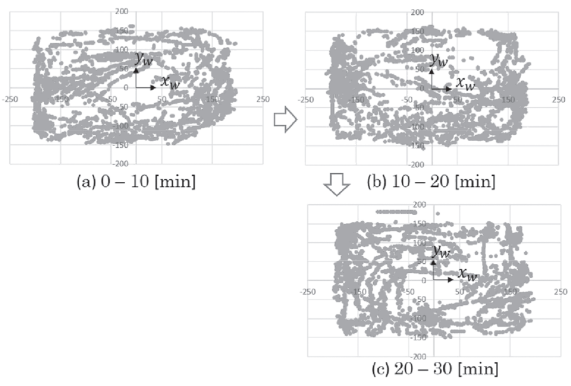

Fig. 8. Recognized position with chaos.

initial value are shown in Fig. 4. $\varepsilon_x\_i(i = 02, 03, 04)$ represent chaos 02 $\sim$ 04 x-coordinate differences between trajectories. Because ε_x_02, ε_x_03, ε_x_04 are almost zero from 0 s to 250 s, they are not shown until 250 s. Initial values are same as chaos 01's. As for y and z coordinates, they are similar to x, omitted to spare the space. We can see from Fig. 4 that the difference between the two trajectories with minute difference of initial values diverts suddenly. Each chaos properties are confirmed about chaos 02 $\sim$ 04 as well as chaos 01.

7. Applying chaos trajectories into robot's net motion

The chaos made by NNDE is actually adapted to the robot dynamics. Figure 6 shows the results of visual servoing and chaos experiments. In comparison with visual servoing (FLS = – 0.30), it is said that chaos trajectory (FLS = – 0.08) reduced the learning speed of fish.

Figures 7 and 8 show fish positions recognized by the robot in visual servoing and chaos experiments. Figures 7(a), (b) show fish escaped from the net in circular swimming and Fig. 7(c) shows fish gradually stayed at corner to escape from the net. In the case of chaos, Fig. 8 shows fish didn't tend to change the oneself movement like staying at corners as time goes by. It indicates that the robot tried to drive fish from the corners by chaos and the fish couldn't take escaping strategy that fish stay at corners. So, it is concluded that chaos has the catching possibilities for a escaping fish. From these results, we confirmed that four type of chaos that generated by NNDE are valid for decreasing the fish's learning velocities that could be thought that the chaos increased the robot's intelligence against fish's escaping intelligence.

References

[1] M. Böhlen, A robot in a cage-exploring interactions between animals and robots, *CIRA*, Monterey, (1999), pp. 214–219.

[2] H.-J. Park, K.B. Kook and L.K. Young, Measuring the machine intelligence quotient (MIQ) of human-machine cooperative systems, *IEEE Trans* **31** (2001), 89–96.

[3] M. Minami, J. Agubanhan and T. Asakura, Manipulator Visual Servoing and Tracking of Fish using Genetic Algorithm, *Int. J. of Industrial Robot* **29**(4) (1999), 278–289.

[4] H. Suzuki and M. Minami, Visual Servoing to catch fish Using Global/local GA Search, *IEEE/ASME Transactions on Mechatronics* **10**(3) (2005), 352–357.

[5] K. Aihara, Nyuraru sisutemu ni okeru kaosu (Chaos in Neural System), Tokyo: Kodansha, (1993), pp. 126–151, (in Japanese).

[6] J. Hirao and M. Minami, Intelligence Comparison between Fish and Robot using Chaos and Random, *Proceedings of the 2008 IEEE/ASME international Conference on Advanced Intelligent Mechatronics*, Xi'an, China, (2008), pp. 552–557.

[7] M. Minami, A. Yanou, Y. Ito and T. Tomono, Multiple Chaos Generator by Neural-Network-Differential-Equation for Intelligent Fish-Catching, *Journal of Communication and Computer* **10** (2013), pp. 823–831.

[8] K. Funahashi, On the Approximate Realization of Continuous Mappings by Neural Networks, *Neural Networks* **2** (1989), 183–191.