The 2012 IEEE/ASME International Conference on
Advanced Intelligent Mechatronics
July 11-14, 2012, Kaohsiung, Taiwan

Analyses of Chaos Generated by Neural-Network-Differential-Equation for Intelligent Fish-Catching

Takashi Tomono, Yuya Itou, Mamoru Minami and Akira Yanou

***Abstract*— Continuous catching and releasing experiment of several fishes make the fishes find some escaping strategies. To make fish-catching robot intelligent more 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 NNDE 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. In this report we examine interesting chaotic characters of plural chaos generated by NNDE through Lyapunov number, Poincare return map, initial value sensitivity and bifurcation map.**

I. INTRODUCTION

A new trend of machine intelligence [1] that differs from the classical AI has been applied intensively to the field of robotics and other research areas like intelligent control system. 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 [2]. 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, the robot can learn from the environment for safety like avoiding obstacle. As known universally that 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 [3].

In our approach to pursue intelligent robot, we will evaluate the intelligence degree between fish and the robot by Fish-Catching operation. We think that the system combined with chaos be smarter than the fish when the robot can beat the fish by catching it successfully even after the fish finds out some escaping strategy. As we did not find the research about the intelligence comparison between animal and robot, we have mainly dedicated ourselves to constructing a smart system that is more intelligent than the fish. We considered that the competitive relation can be very meaningful as one way to discuss robotic intelligence.

In recent years, visual tracking and servoing in which visual information is used to direct the end-effector of a manipulator toward a target object has been studied in some researches [4], [5]. By evolutionary algorithms [6], 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) [7], and we succeeded in catching a fish by a net attached at the hand of the manipulator based on the real-time visual tracking under the method of Gazing GA [8] to enhance the real-time searching ability.

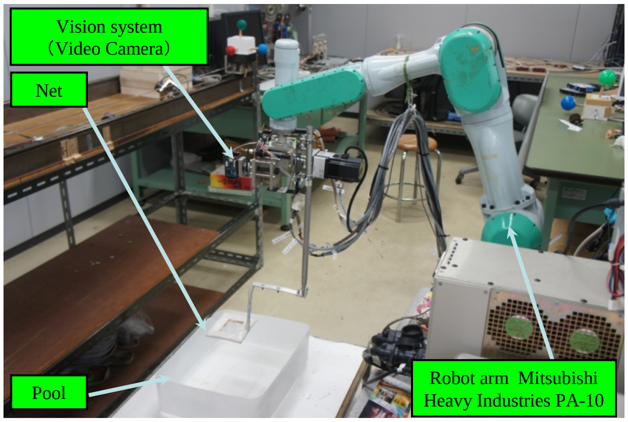


Fig. 1. Fish Catching system PA10

Through 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 can 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 deemed 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 competitive relations with the robot, that is continuous catching/releasing experiments, we found that we can define a "Fish's Intelligent Quotient"(FIQ) representing decreasing velocity of fish number caught by the net through continuous catching/releasing operation. Through this measure we can compare the innate intelligence of the fish and the artificial 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 [9], [10]. Therefore we thought that imitating such animal's internal dynamics and putting chaos into robots have something meaningfulness to address fish's intelligence. We embed chaos into the Robot Dynamics in order to supplement the deficiency of our Fish-Catching

The authors are with Graduate School of Natural Science and Technology, Okayama University, Okayama, Japan {tomono, itou, minami, yanou }@suri.sys.okayama-u.ac.jp

978-1-4673-2576-9/12/$31.00 ©2012 IEEE

system [11].

Therefore what we have to pay attention to the fish's nature is 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) that can produce 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[12], [13].

In this paper, we report analyses of chaos generated by NNDE, especially noting how the generated chaos changes depending on a single coefficient in neural network being varied slightly through bifurcation diagram.

II. FISH TRACKING AND CATCHING

The problem of recognition of a fish and detection of its position/orientation is converted to a searching problem of $\boldsymbol{r}(t) = [x(t), y(t)]^T$ in order to maximize $F(\boldsymbol{r}(t))$, where $F(\boldsymbol{r}(t))$ represents correlation function of images and fish-shaped matching model. $F(\boldsymbol{r}(t))$ is used as a fitness function of GA [8]. To recognize a target in a dynamic image input by video rate, 33 [fps], the recognition system must have real-time nature, that is, the searching model must converge to the fish in the successively input raw images. An evolutionary recognition process for dynamic images have been realized by such method whose model-based matching by evolving process in GA is applied at least only one time to one raw image input successively by video rate. We named it as "1-Step GA" [7]. When the converging speed of the model to the target in the dynamic images should be faster than the swimming speed of the fish in the dynamic images, then the position indicated by the highest genes represent the fish's position in the successively input images in real-time. We have confirmed that the above time-variant optimization problem to solve $\boldsymbol{r}(t)$ maximizing $F(\boldsymbol{r}(t))$ could be solved by "1-Step GA". $\boldsymbol{r}(t) = [x(t), y(t)]^T$ represents the fish's position in Camera Frame whose center is set at the center of catching net, then $\boldsymbol{r}(t)$ means position deviation from net to Fish, means $\boldsymbol{r}(t) = \Delta\boldsymbol{r}(t)$ The desired hand velocity at the i-th control period $\dot{\boldsymbol{r}}_d^i$ is calculated as

$$\dot{\boldsymbol{r}}_d^i = \boldsymbol{K}_P \Delta\boldsymbol{r}^i + \boldsymbol{K}_V(\Delta\boldsymbol{r}^i - \Delta\boldsymbol{r}^{i-1}) \qquad (1)$$

where $\Delta\boldsymbol{r}^i$ denotes the servoing position error detected by 1-Step GA [7]. $\boldsymbol{K}_P$ and $\boldsymbol{K}_V$ given are positive definite matrix to determine PD gain. Now we add chaos items to (1) above, and we also need to redefine the meaning of $\dot{\boldsymbol{r}}_d^i$.

The simple PD servo control method given by (1) is modulated to combine a visual servoing and chaos net motion into the controller as follows,

$$\Delta\boldsymbol{r}^i = k_1 \cdot \Delta\boldsymbol{r}^i_{fish} + k_2 \cdot \Delta\boldsymbol{r}^i_{chaos} \qquad (2)$$

Here $\Delta\boldsymbol{r}^i_{fish} = \begin{bmatrix} \Delta x^i_{fish} & \Delta y^i_{fish} \end{bmatrix}$, is the tracking error of fish from the center of camera frame, and $\Delta\boldsymbol{r}^i_{chaos} = \begin{bmatrix} \Delta x^i_{chaos} & \Delta y^i_{chaos} \end{bmatrix}$ denotes a chaotic oscillation in $x-y$ plane around the center camera frame. Therefore the hand motion pattern can be determined by the switch value k_1 and k_2. $k_1 = 1$ and $k_2 = 0$ indicate visual servoing, and $k_1 = 0$ and $k_2 = 1$ indicate the net will track chaotic trajectory made by NNDE being explained later in this report. The desired joint variable $\dot{\boldsymbol{q}}_d$ is determined by inverse kinematics from $\dot{\boldsymbol{r}}_d$ by using the Jacobian matrix $\boldsymbol{J}(\boldsymbol{q})$, and is expressed by

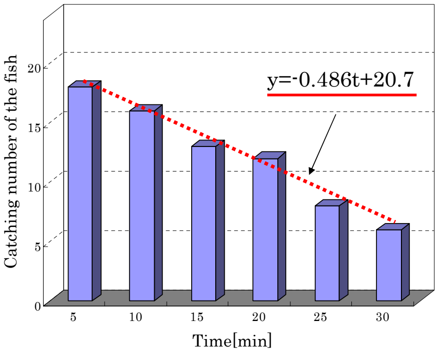
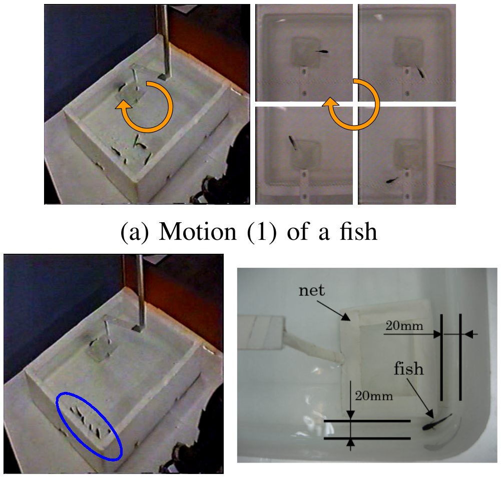


Fig. 2. Result of catching number



(a) Motion (1) of a fish

(b) Motion (2) of a fish

(c) Motion (3) of a fish

Fig. 3. Fish motion

$$\dot{\boldsymbol{q}}_d = \boldsymbol{J}^+(\boldsymbol{q})\dot{\boldsymbol{r}}_d \qquad (3)$$

where $\boldsymbol{J}^+(\boldsymbol{q})$ is the pseudo inverse matrix of $\boldsymbol{J}(\boldsymbol{q})$. The robot used in this experimental system is a 7-Link manipulator, Mitsubishi Heavy Industries PA-10 robot.

III. PROBLEM OF FISH-CATCHING

In order to check the system reliability in tracking and catching process, we kept a procedure to catch a fish and release it immediately continuously for 30 minutes. 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. The result of this experiment is shown in Fig.2, in which vertical axis represents the number of fishes caught in successive 5 minutes and horizontal axis represents the catching time. We had expected that the capturing operation would become smoother as time passing on consideration that the fish may get tired. But

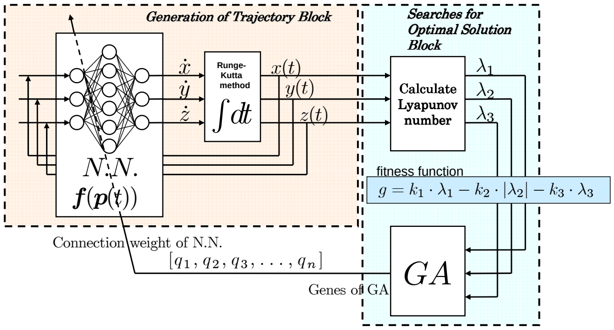

Generation of Trajectory Block
Searches for Optimal Solution Block
$\dot{x}$
$\dot{y}$
$\dot{z}$
Runge-Kutta method
$\int dt$
$x(t)$
$y(t)$
$z(t)$
Calculate Lyapunov number
λ_1
λ_2
λ_3
$N.N.$
$\boldsymbol{f}(\boldsymbol{p}(t))$
fitness function
$g = k_1 \cdot \lambda_1 - k_2 \cdot |\lambda_2| - k_3 \cdot \lambda_3$
Connection weight of N.N.
$[q_1, q_2, q_3, \ldots, q_n]$
Genes of GA
GA

Fig. 4. Block diagram of Chaos Generation

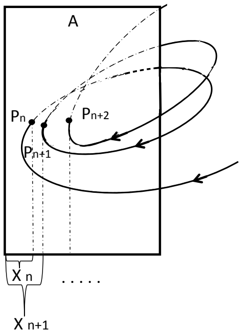


Fig. 5. Poincare section

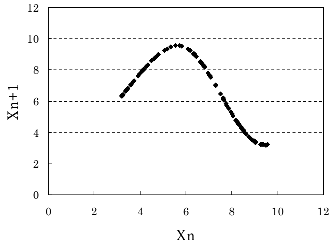


Fig. 6. poincare return map

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

The reason of decreased catching number may lie in the fish learning ability. For example, the fish can learn how to run away around the net as shown in Fig.3(a) 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 can stay in the opposite corner against the net in the pool shown in Fig.3(b). And also, the fish can keep staying within the clearance between the edge of the pool and the net shown in Fig.3(c) where the net is inhibited to enter.

To solve these problems, and to achieve more intelligent fish catching systems, we thought chaos behavior of the net with many chaotic varieties can be a possible method to 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 made the fish's learning logic confuse, getting the fish catching robot having made intelligence than the fish's. Then we propose Neural-Network-Differential-Equation to generate chaos as many as possible.

IV. FISH INTELLIGENCE QUOTIENT

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 $y = -0.486t + 20.7$ as shown in Fig.2. The decreasing coefficient -0.486 represents adapting or learning velocity of the fishes as a group when the fish's intelligence is compared with robotic catching. We named the coefficient as "*Fish's Intelligence Quotient*"(FIQ). The larger minus value means high intelligence quotient of the fish, zero does equal, and plus does less intelligent than robot's. To overcome the fish's intelligence, more intelligent robotic system needs to track and catch the fish effectively. in other words it comes to the problem on how to use the item $\Delta \boldsymbol{r}^i_{chaos}$ in (2) effectively to exceed the fish's intelligence.

V. NN-DIFFERENTIAL-EQUATION

Lorenz and Rossler models renowned as chaos generation comprise three differential equations, producing three-dimensional chaotic trajectory in phase space. Since a Neural-Network(N.N.) has been proven to have an ability to approximate any non-linear functions with arbitrarily high accuracy[12][13], we thought it is straightforward to make a differential equation including N.N. so that it can generate plural chaos by changing N.N.'s coefficients. We define next nonlinear differential equation including N.N. function $\boldsymbol{f}(\boldsymbol{p}(t))$ as

$$\dot{\boldsymbol{p}}(t) = \boldsymbol{f}(\boldsymbol{p}(t)). \tag{4}$$

$\boldsymbol{p}(t) = [x(t), y(t), z(t)]^T$ is state variable. The nonlinear function of $\boldsymbol{f}(\boldsymbol{p}(t))$ in (4) is constituted by N.N.'s connections, which is exhibited in left part of Fig.4 where the N.N. and integral function of outputs of N.N. and the feedback of the integrated value to the inputs of N.N. constitute nonlinear dynamical equation, (4). We call it as Neural-Network-Differential-Equation.

VI. CHAOS VERIFICATION METHOD

Since there have been no simple criterion to determine whether irregular oscillation is a chaos or not, we have to apply plural evaluations over the irregularities of trajectories produced by NNDE. The followings are criteria being used for judging the chaotic characters.

A. Lyapunov exponent

As one of criteria to evaluate a chaos' character of time function $f(t)$ at discrete time t_i in time domain, Lyapunov exponent expressed by the following equation is well known,

$$\lambda = \lim_{N \to \infty} \frac{1}{N} \sum_{i=0}^{N-1} \log \left| \frac{df}{dt}(t_i) \right|, \tag{5}$$

where positive value can represent that the irregular oscillation diverts from a standard trajectory, which expands like a function of e^{at} $(a > 0)$.

B. Poincare section

Poincare section is to verify further whether trajectories can be identified as chaos. Next, the Poincare section is explained. First of all, we examine an simple closed curve in three dimensions as shown in Fig5. The plane "A" that intersects with this closed trajectory pointed by "P" is defined as the Poincare section. The intersecting points are named as P_n, P_{n+1},P_{n+2},$\cdots$, and corresponding x-axis position

$$\underbrace{\overbrace{10011\ldots1}^{q_1}\overbrace{01001\ldots1}^{q_2}\cdots\overbrace{11010\ldots0}^{q_n}}_{\textbf{Individual Combination of Connection weight of N.N.}}$$

Fig. 7. Gene of GA

on A are x_n ,x_{n+1},x_{n+2},$\cdots$, which are all pointed to the Poincare Return Map as x_n,x_{n+1},$\cdots$ as shown in Fig.6. With the poincare return map of Fig.6 representing a shape of "Λ", the closed curve has a structure of stretching and folding, corresponding left half and right one. This structure is a basic character of chaos. Looking at the left half of Fig.6, we can see the inclination coefficient $dx_{n+1}/dx_n > 1$ and right half has $dx_{n+1}/dx_n < -1$, representing that left half has expansion and the other does contraction.

C. 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".

D. bifurcation diagram

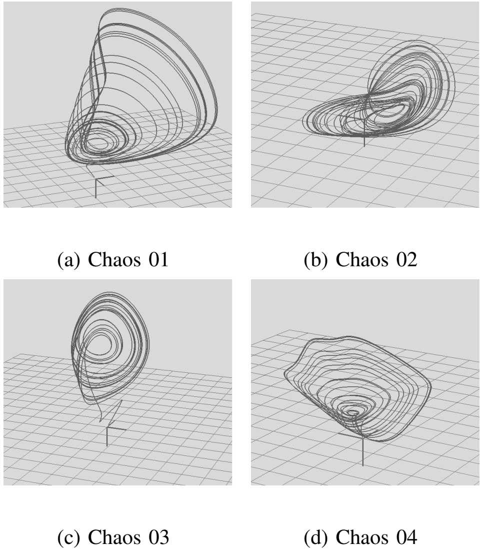
After a solved trajectory $\boldsymbol{p}(t) = [x(t), y(t), z(t)]$ has been obtained by numerical integration with found coefficients $\boldsymbol{q}$ by gene in GA procedure, the crossing point of $\boldsymbol{p}_k(t)$ with $x - z$ plain when the trajectory $\boldsymbol{p}(t)$ pass through the plain k times are all recorded. In case of the period of $\boldsymbol{p}(t)$ being 1 cycle, one fixed point appears on the $x - y$ plain. If it is 2 cycle, two fixed points are spotted. Further, with the $\boldsymbol{p}(t)$ having k cycle, the spots on $x - y$ plain number k. A diagram depicting an expanding on contracting transitions of crossing points depending on changing a parameter relating to the eq.(4), in this case one coefficient value of a neural network. When $\boldsymbol{p}(t)$ is chaos, it has infinite period, then the crossing spots appears on the $x - z$ plain infinitely.

VII. CHAOS GENERATE SYSTEM

Fig.4 represents the block diagram to find chaos by using GA and Lyapunov number. This GA is not 1-Step GA, described in Chapter II but used as a normal GA's procedure that evolves genes representing neural network coefficient's value. The trajectory $\boldsymbol{p}(t)$ in time domain obtained from Neural-Network-Differential-Equation is used for the calculation of Lyapunov number. Here, $\boldsymbol{L} = [\lambda_1, \lambda_2, \lambda_3]^T, (\lambda_1 > \lambda_2 > \lambda > 3)$ is a Lyapunov number. Using this $\boldsymbol{L}$ for the evolution of GA, fitness function is defined as follows,

$$g = k_1 \cdot \lambda_1 - k_2 \cdot |\lambda_2| - k_3 \cdot \lambda_3. \qquad (6)$$

This fitness function incorporated the chaotic property of the Lyapunov spectrum, which is one of factors to be essential for generating chaos trajectory. Here, because we discuss three-dimensional chaotic attractor in phase space, there are 3 Lyapunov numbers. The relationship between positive and negative Lyapunov spectrum is $(+, 0, -)$, which means resulted time trajectory of (4) may be thought to be chaos. Parentheses indicate the sign of the Lyapunov spectrum. In other words, λ_1 is positive, λ_2 is also positive or negative small values, λ_3 is negative case, the fitness function of (6) appears to have relatively large positive value when $\lambda_1 > 0$, $\lambda_2 \approx 0$, $\lambda_3 < 0$. In addition k_1, k_2 and k_3 are positive coefficients. The gene of GA is defined as shown in Fig.7, with connection weights of N.N. being $\boldsymbol{q} = [q_1, q_2, \ldots, q_n]^T$. In this report we adopted a network of $3 \times 6 \times 3$ as shown in Fig.4, then the number of connections and their coefficients is 36, i.e., n=36. The bit length of $\boldsymbol{q}_i$ is 16 bits. Because the gene is expressed in binary, converted to decimal and normalized into a range from 0 to 1. Then, generating a trajectory $\boldsymbol{p}(t)$ based on a given gene having been determined by GA at one previous generation and calculating Lyapunov number, and evolving new generation of gene are repeated. This GA's evolution can find $\boldsymbol{q}$ to have a highest value of g defined by (6), that means possible chaos trajectory.


(a) Chaos 01

(b) Chaos 02

(c) Chaos 03

(d) Chaos 04

Fig. 8. Generated Chaos

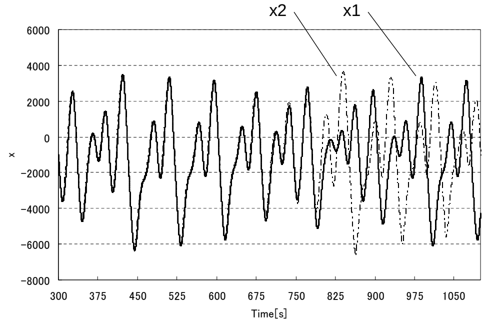


Fig. 9. Generated trajectory 01 of x (300[s] to 1100[s])

VIII. VERIFICATION RESULTS

So far we have found four chaos patterns with different neural coefficients explored by GA mentioned in the previous section. We named them with a serial number as chaos

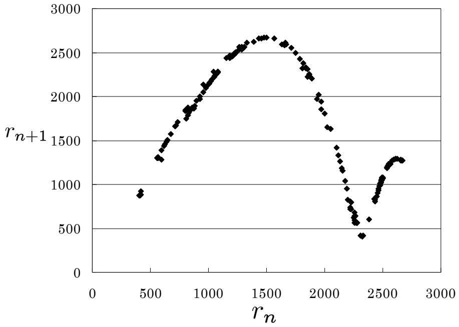


Fig. 10. Poincare return map of Chaos01

TABLE I
LYAPUNOV NUMBER

	chaos01	chaos02	chaos03	chaos04
λ_1	0.014585	0.01919	0.015934	0.01208
λ_2	-0.003314	0.00733	-0.002172	-0.00143
λ_3	-0.165381	-0.10379	-0.123026	-0.075448

01∼chaos 04. The followings are the introduction of those chaos with each individual character.

A. Chaos 01

We searched Lyapunov number, an sensitivity of initial value, and the Poincare return map about chaos 01.

1) Lyapunov number: Lyapunov numbers are $\lambda_1 = 0.014585$, $\lambda_2 = -0.003314$ and $\lambda_3 = -0.165381$. These are corresponding to the Lyapunov spectrum of chaos,$(+, 0, -)$.

2) Sensitivity to initial value: Two time-profile of trajectories with minutely different initial value are shown in Fig.9. The trajectories of $(x_1(t), y_1(t), z_1(t))$ are the results that originated from the initial values of $x_1(0) = 1.00, y_1(0) = 1.00, z_1(0) = 1.00$ and $(x_2(t), y_2(t), z_2(t))$ are from $x_2(0) = 1.01, y_2(0) = 1.01, z_2(0) = 1.01$. Trajectories of x_1 and x_2 are shown in Fig.9 .

We can see from Fig.9 that the two trajectories with minute difference of initial values divert often 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. As for y and z coordinates, they are similar, omitted to spare the space.

3) Poincare return map: Chaos 01's poincare return map is shown in Fig.10. One dimensional map can be seen in Fig.10, from which we can understand that the map represents expanding (left half of the Fig.10) and contracting (right half) that are essential characters to generate chaos. Therefore, the property of chaos 01 has been confirmed from the viewpoint of Lyapunov number, an sensitivity of initial value, and the Poincare return map.

B. Chaos 02 ∼ 04

We have found other three chaos, 02, 03 and 04, where Lyapunov numbers are listed in Table 1, including chaos 01 also. We confirmed all chaos trajectories have the Lyapunov spectrum of chaos, $(+, 0, -)$.

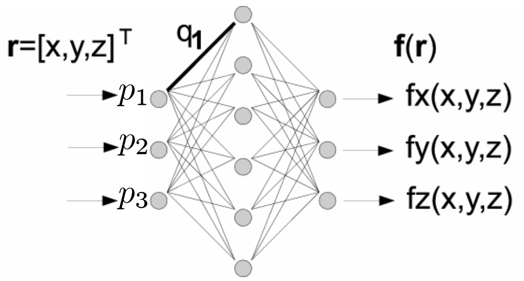


Fig. 11. Neural Network for nonlinear function generation

IX. SENSITIVITY OF NEURON'S WEIGHT

We have noticed weight coefficient of N.N. that generated chaos 03 are almost similar to chaos 04's. That is, only one weight coefficient is different, that is "q_1" in Fig.11. We think "q_1" is related to the generation of chaos [14]. So we increased the weight q_1 gradually from "-1.0" to "$+1.0$" with q_1 increasing by 0.0001 and compare their trajectories. The range of q_1 is $-1.0 \leq q_1 \leq 1.0$ and q_1 is increased from -1.0 by 0.1. In the case of $-1.0 \leq q_1 \leq -0.4$, $0.2 \leq q_1 \leq 0.3$ and $0.9 \leq q_1 \leq 1.0$, the trajectories diverged without cyclic motions.

A. Bifurcation

1) Confirmation of pitchfork bifurcation: Figure12 shows pitchfork bifurcation diagram with q_1 increasing from -0.27 to 0.17 increasing by 0.0001. The value of r in vertical axis means the x - axis coordinate of the spotted point with the solved trajectory $\boldsymbol{p}(t)$ passed through $x - z$ plain. As increasing q_1, the trajectory $\boldsymbol{p}(t)$ is bifurcated into two, four, and falls into chaos behavior as you can see the expanded depiction of Fig.12, around $q_1 = -0.2$, which is corresponding to the designated by the point "A" in Fig.13. This is said to be "period doubling", and typical phenomenon in chaos. The case of $q_1 = -0.12$, the spotted points are scattered densely, meaning the trajectory diverged, resulting in no periodical spotted point in $x - z$ plain.

2) Confirmation of a window with 3 period: Figure14 shows the bifurcation diagram with q_1 changing from 0.4 to 0.85, which is expanded from a part "B" of Fig.13. A window with 3 period appeared when $0.47 \leq q_1 \leq 0.5$, which is the typical phenomenon of chaos.

B. Generated trajectories in $x - y - z$ space

Three typical trajectories of limit cycle, chaos, periodic trajectory with three-period solution are examined below, which correspond to $q_1 = -0.2807$ with mark α, -0.1084, mark β, 0.1744, mark γ, in Fig.12, and $q_1 = 0.4898$ with mark δ in Fig.14.

1) In case of $q_1 = -0.2807$: About Fig.15, the left figure shows the whole 3D trajectory. The right graph shows points that the trajectory passed through $x - z$ plane when q_1 is set at -0.2807. The horizontal axis means the number of the trajectory that was passed the plane, and the value of r in

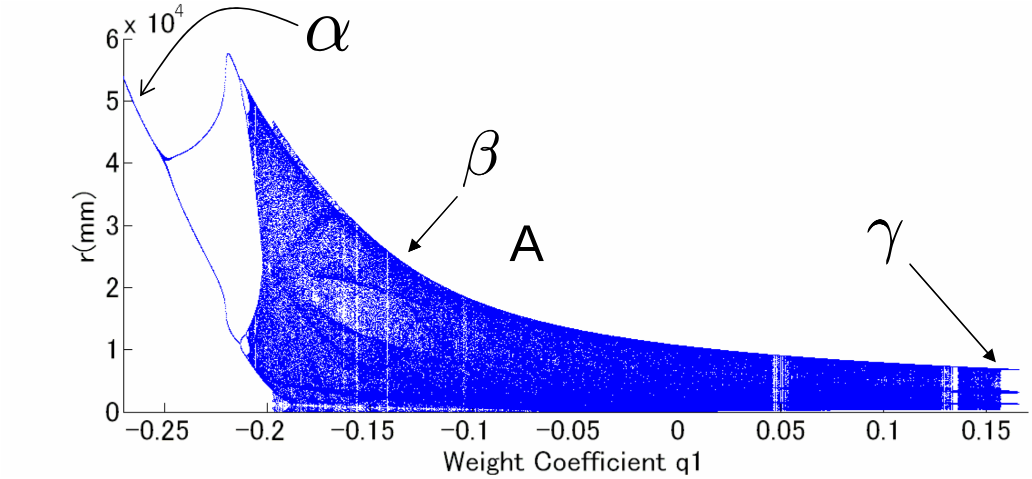


Fig. 12. bifurcation diagram 1

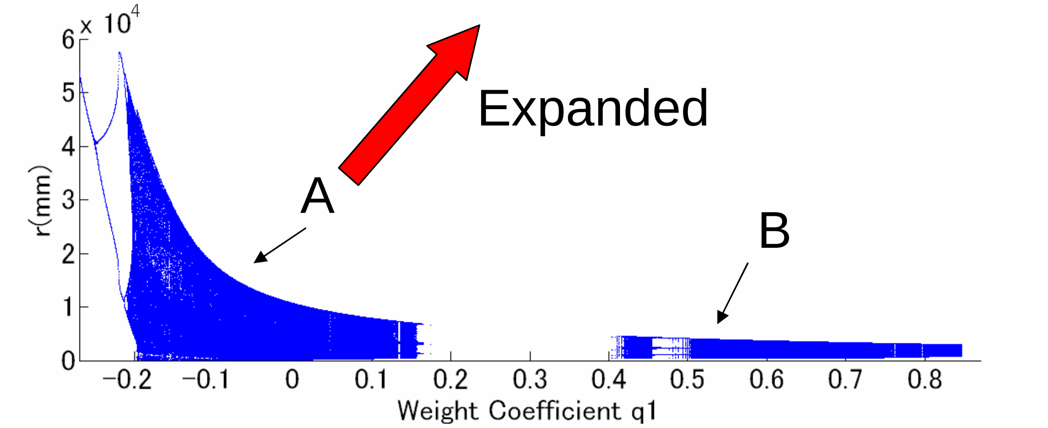


Fig. 13. bifurcation diagram 2

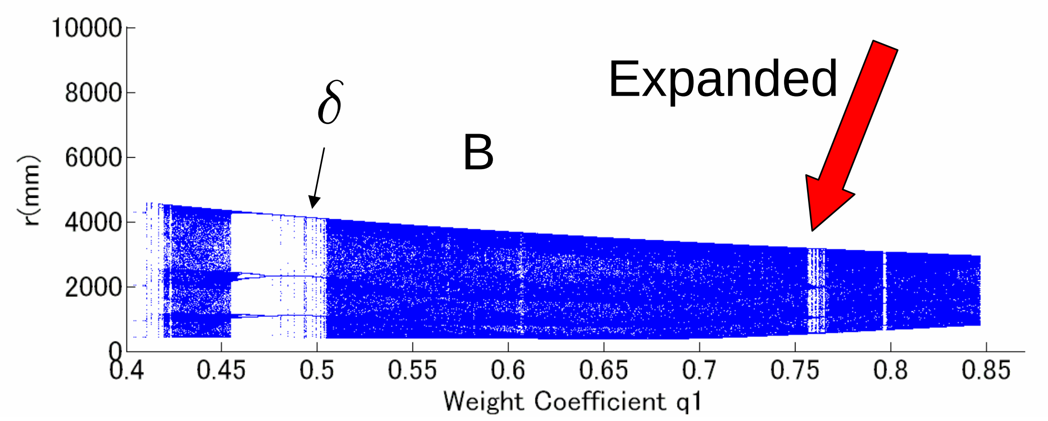


Fig. 14. bifurcation diagram 3

vertical axis means the x - axis coordinate of the spotted point in $x-z$ plain. From the graph, this trajectory has one period because it always shows about $r = 65000$, meaning that generated trajectory represents a limit cycle.

2) In case of $q_1 = -0.1084$: About Fig.16, the left figure shows the whole 3D trajectory when $q_1 = -0.1084$.

The right graph shows points that the trajectory was passed the plane. From the graph, this trajectory has random pass-through points, representing chaotic character.

3) In case of $q_1 = 0.1744$*:* About Fig.17, the left figure shows the whole 3D trajectory. The right graph shows points that the trajectory was passed the $x-z$ plane. From the graph, this trajectory has three periods because it shows three points about $r = 3000,\ \ r = 6500,\ \ r = 1000$ by rotation.

4) In case of $q_1 = 0.4898$*:* About Fig.18, the left figure shows the whole 3D trajectory. The right graph shows points that the trajectory was passed half plane. From the graph, this trajectory is three periods because it shows about $r = 4000$, $r = 1000,\ \ r = 2500$ by rotation. This window like section has been called as window of chaos.

X. Conclusion

This paper proposed chaos generating system composed of Neural Network and GA's evolving ability to change the Neural-Network-Differential-Equation to be able to generate chaos. This chaos generating system has exploited the neural network's nature of approximation of any nonlinear function with any desired accuracy.

Furthermore periodic trajectories and trajectory that has huge variation of periods are confirmed by changing q_1, leaving a typical characters of chaos such as bifurcation diagram against changing a sole coefficient of one neuron. By this diagram, transition between chaos and not chaos has been examined intensively. That is, we thought the proposed NNDE is a system that generates chaos. A chaos motion can make fishes confuse, so we think it will be effective for fish catching. We will utilize these chaos motion for overcoming fish's escaping ability from chasing net from now, and confirm the effectiveness.

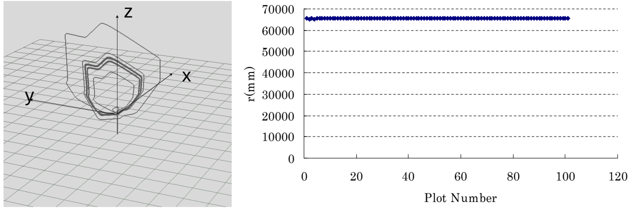


Fig. 15. q_1 = -0.2807

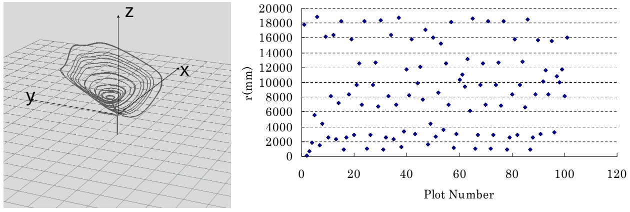


Fig. 16. q_1 = -0.1084

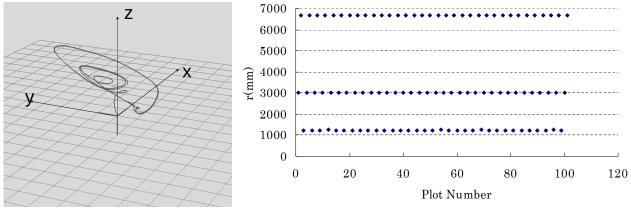


Fig. 17. q_1 = 0.1744

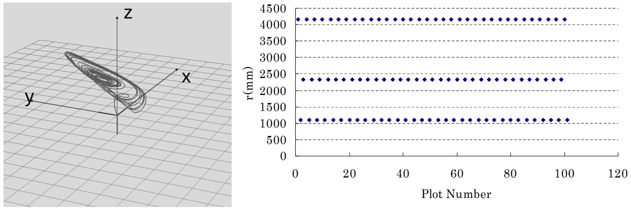


Fig. 18. q_1 = 0.4898

References

[1] T. Fukuda, and K. Shimojima: "Intelligent Control for Robotics", Computational Intelligence, 1995, pp.202-215.

[2] M. Bohlen: "A robot in a cage-exploring interactions between animals and robots", CIRA. 1999, pp.214-219.

[3] Hee-Jun Park, Byung Kook Kim, Kye Young Lim: "measuring the machine intelligence quotient (MIQ) of human-machine cooperative systems", IEEE Trans. vol.31, 2001, pp.89-96.

[4] R. Kelly: "Robust Asymptotically Stable Visual Servoing of Planar Robots", IEEE Trans. Robot. Automat., vol.12, no.5, 1996, pp.759-766.

[5] P.Y. Oh, and P.K. Allen: "Visual servoing by partitioning degrees of freedom", IEEE Trans. Robot. Automat., vol.17, pp.1-17, Feb.2001

[6] M. Minami, H. Suzuki, J. Agbanhan, T. Asakura: "Visual Servoing to Fish and Catching Using Global/Local GA Search", Int. Conf. on Advanced Intelligent Mechatronics, Proc., 2001, pp.183-188.

[7] M. Minami, J. Agubanhan, and T. Asakura: "Manipulator Visual Servoing and Tracking of Fish using Genetic Algorithm", Int. J. of Industrial Robot, Vol.29, No.4, 1999, pp.278-289.

[8] Visual Servoing to catch fish Using Global/local GA Search Hidekazu Suzuki, Mamoru Minami IEEE/ASME Transactions on Mechatronics, Vol.10, Issue 3, 352-357 (2005.6)

[9] K. Aihara: "Chaos in Neural System", pp.126-151, 1993 (in Japanese).

[10] R. FitzHugh: "Impulses and physiological states in theoretical models of nerve membrane", Biophy.J.1, pp.445-466 (1961).

[11] Jun Hirao and Mamoru Minami, "Intelligence Comparison between Fish and Robot using Chaos and Random", Proceedings of the 2008 IEEE/ASME international Conference on Advanced Intelligent Mechatronics July 2 - 5, 2008, Xi ' an, China, pp552-557.

[12] C. T. Lin and C. S. Lee, "Neural Fuzzy Systems", *Englewood Cliffs*, NJ:Prentice Hall PTR, 1996.

[13] Limin Peng and Peng-Yung Woo, "Neural-Fuzzy Control System for Robotic Manipulators", *IEEE Control Systems Magazine*, 2002, pp.53-63.

[14] Y Itou, T Tomono and M Minami, "Multiple Chaos Generator by Neural-Network-Differential-Equation for Intelligent Fish-Catching", International Conference of Industrial Electronics, Control and Instrumentation, 2011, pp.2319-2324.