A Physics Modeling of Butterfly's Flight Control by GA and ANN and Its Over-evolution Problem

R. Ooe, I. Suzuki, M. Yamamoto and M. Furukawa

Graduate School of Information and Science, Hokkaido University, Sapporo, Hokkaido, Japan
(Tel : +81-11-706-6445; Fax : +81-11-706-6443)
({ooe, ikuo, masahito, mack}@complex.eng.hokudai.ac.jp)

Abstract: We describe a simple physics model of a butterfly and an approach to its flight control by the genetic algorithm (GA) and the artificial neural network (ANN). A physics model consists of two kinetic equations which are led by a simplification of fluid force. A butterfly's flight is controlled by an ANN. The GA optimizes weights of the ANN for the suitable flight. This approach resulted in the flight which is obtained by maximizing the prepared fitness. However, the optimized ANN did not have a generality and a butterfly fell down in changing the initial height. A transition of fitness throughout processes of evolution showed that too much optimization tends to break the generality. We call this phenomenon "over-evolution". Changing conditions of experiments prevented the over-evolution.

Keywords: Physical simulation, Drag force, Butterfly, Real-coded genetic algorithm, Artificial neural network

I. INTRODUCTION

Flights by a flap of wings are difficult actions. The motion of life existing on the ground is restricted to the two-dimensional space, that is, the ground. On the other hand, life in the air, such as the bird and the butterfly, is able to move freely in the three-dimensional space without falling down. It is very interesting to understand these flying creatures deeply. However, there are few researches on them as the artificial life because of the requirements of many computational resources.

We first overlook previous works on the three-dimensional physical simulation. Terzopoulos et al [1]. modeled the artificial fish by springs and sensors. Although it seemed to be a real fish, the calculation of the drag force was dispensed with. Usami [2] simulated the motion of a fish using the moving particle method. However, it was confined to the two-dimensional simulation. Wu et al [3]. proposed a model of bird for the computer graphics. Their bird consisted of parts connected by springs and it was taken account of the lift and drag forces. It is one of a few studies of the flying creatures.

Sims [4] showed various shapes of artificial life and behaviors, such as walking, jumping and swimming. Both shapes and behaviors had been acquired by the evolutionary algorithm (EA). This work showed that the environment gave life various shapes and behaviors, so it is very remarkable. Reil et al [5]. applied EAs to the control problem of bipedal walking.

We had studied a modeling and simulating method for various types of artificial life using physics engines [6]. To simplify the flight mechanism and to reduce a computational amount, we modeled a butterfly's flight into two kinetic equations. In this paper, we describe the method of a model and control of butterfly. A model is described in Section Two and the control in Section Three. We also refer to the problem about an optimized controller. Section Four explains the details of this problem and a solution based on properties of actions. Finally, our work is summarized in Section Five as conclusion.

II. MODELING

1. Drag force

The external force that acts on a flying butterfly is drag force besides gravity. Simplified the drag force ΔD for each surface of an object is

$$\Delta D = \frac{1}{2} \rho \, \Delta A_p \, C_d \, v_p{}^2 \qquad (1)$$

where ρ is the density of air, ΔA_p is the area of a surface, C_d is the drag coefficient and v_p is the velocity relative to air.

The drag force works at the center of gravity and it is perpendicular to a surface. In applying (1) to an object, the drag force is calculated for only the surface facing to the direction of the velocity. If the surface is relatively large, we divide it into sub-surfaces and calculate for each sub-surface for accuracy.

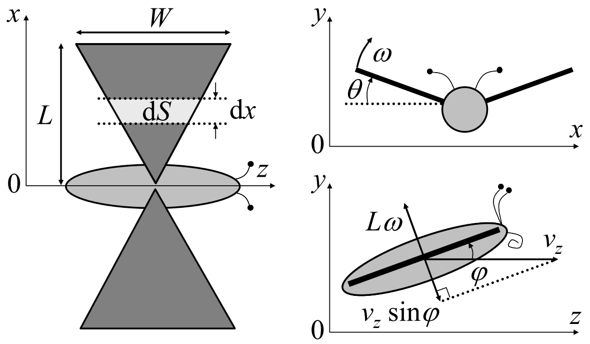

Fig.1. The shapes and parameters of the model

2. Modeling butterfly

Based on (1), two kinetic equations are formalized. For simplification, we assume that a wing of butterfly is an inverted triangle plane which is shown in Fig.1. Parameters used in the model are listed in Table 1.

A. Drag force acting on wings

Wings rotate up and down on the body. A minute area $\mathrm{d}S(x)$ at a distance x from the body, and drag force $\mathrm{d}F(x)$ which acts on $\mathrm{d}S(x)$ is expressed by

$$\mathrm{d}S(x) = \frac{W\,x}{L}\mathrm{d}x \tag{2}$$

$$\mathrm{d}F(x) = \frac{1}{2}\rho\, C_d\, (x\,\omega)^2\, \mathrm{d}S \tag{3}$$

(3) is integrated from $x = 0$ to $x = L$, then the total drag force F acting on two wings is given by

$$F = \frac{\alpha}{4} v_w{}^2 \tag{4}$$

where α is a coefficient and v_w is a velocity of the end of a wing. They are expressed by

$$\alpha = \rho\, C_d L\, W, \qquad v_w = L\,\omega \tag{5}$$

B. Change of drag force by tilt angle

It is known that the flapping motion of a butterfly is expressed by the cosine function. In this case, the total amount of the drag force for one cycle of the flapping equals zero, then a butterfly goes down by gravity. Changing the tilt angle φ makes a butterfly fly up.

For simplification, we assume that the upward velocity v_y is relatively small. When a butterfly is tilted by φ and moving forward at speed of v_z, v_w is replaced by v_w'. It is expressed by

Table 1. The parameters of the model

Explanation	Variable
Length of wings	L
Width of wings	W
Angular velocity of wings	ω
Flapping angle of wings	θ
Representative area of a body	S_b
Mass of a butterfly	M
Tilt angle of a butterfly	φ
Horizontal distance	z
Forward velocity	v_z
Height	y
Upward velocity	v_y

$$v_w' = L\,\omega - v_z \sin\varphi \tag{6}$$

Accordingly, the drag force (4) is also re-expressed. Dividing it into the forces in the direction of y and z axis and considering the flapping angle θ lead to

$$\begin{aligned} M\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} &= \frac{\alpha}{4} v_w'^2 \cos\varphi\cos\theta + F_{by} \\ M\frac{\mathrm{d}^2 z}{\mathrm{d}t^2} &= \frac{\alpha}{4} v_w'^2 \sin\varphi\cos\theta + F_{bz} \end{aligned} \tag{7}$$

where F_{by} and F_{bz} are the drag forces acting on a body. They are expressed by

$$F_{by} = \frac{1}{2}\rho\, C_d\, S_b\, v_y{}^2\,, \quad F_{bz} = \frac{1}{2}\rho\, C_d\, S_b\, v_z{}^2 \tag{8}$$

III. CONTROL

The flight of a butterfly is controlled by an evolving artificial neural network (EANN) [7]. θ is given by the cosine function of 10[Hz], while φ is controlled by an EANN.

1. A controller by EANN

We use a three-layered feed-forward artificial neural network (ANN) as a controller. There are six neurons in the input layer, six in the middle layer and one in the output layer. The input signals are φ, ω, θ, v_y, y and v_z. The output is a difference value of φ.

A real-coded genetic algorithm (RCGA) is used to optimize weights of ANNs. The tournament selection of size = 2, the elite selection, the BLX-α crossover of α = 0.45 and multiplying each weight by a random value in range of [-2,2] for a mutation are used as operators of the RCGA. The group size is 40, the crossover rate is 1.0 and the mutation rate is 0.01 for each weights.

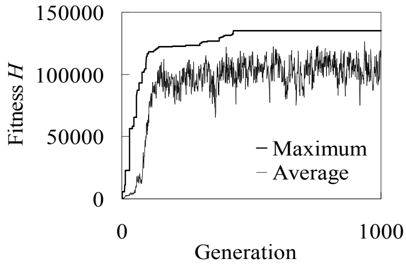

Fig.2. The fitness in evolution

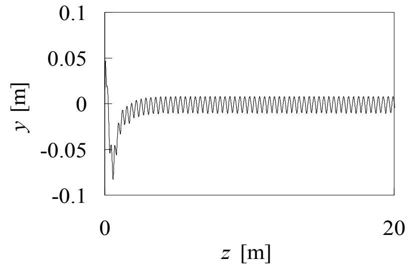

Fig.3. The track of a flight

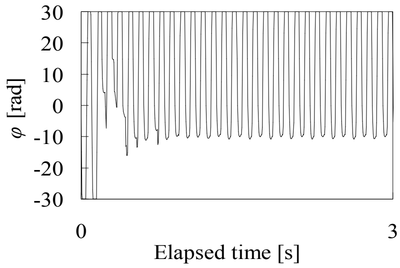

Fig.4. The optimized output φ

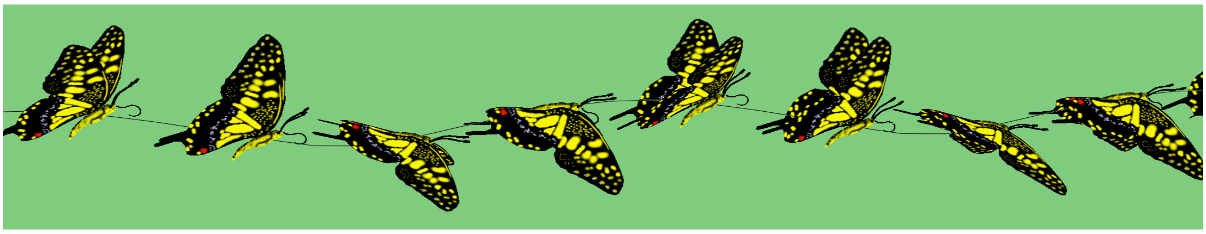

Fig.5. The snapshot of a optimized flight

2. A result of evolution

We have experimented on a simple evolution of our butterfly model. The fitness value H is given by

$$H = \sum_{t=0}^{t_{\max}} \left\{ \frac{v_z(t)}{\mathrm{abs}(y(t)) + 0.1} \right\} \tag{9}$$

where t is a time in the simulation and t_{max} is a terminated time of a simulation. H is very high when a butterfly goes forward rapidly and stays at height of 0[m]. In this experiment, we use t_{max} = 20.0[s], a time step in Runge-Kutta method is 1/240 and the initial height of a butterfly is 0[m].

Transitions of the fitness are shown in Fig.2. The maximum fitness hardly increase in late generations, therefore ANNs are evolved enough.

A track of the flight by the best ANN is shown in Fig.3 and a transition of φ in Fig.4. A height of a butterfly changes little in early time and finally keeps at 0[m]. φ always changes periodically like θ. However, a range of φ changes along with a change of a height. A snapshot of a flight is shown in Fig.5.

IV. OVER-EVOLUTION

It is known that an ANN has a generality. For example, an ANN learned by sample data usually recognizes other data in the pattern recognition. Therefore we have examined a generality of the evolved ANNs. Then we have discovered a problem.

1. Generality for changes of initial heights

The ANN learned by evolution in Section Three enables a butterfly to keep a height. Therefore we supposed that this ANN works well for different initial heights. When a butterfly starts at 10[m] or -10[m], however, it falls down quickly and can not keep at 0[m].

This might happen unfortunately, so we tried some times. Through some experiments, we found out that ANNs in a middle of evolution tend to be successful for different initial heights. Typical transitions of the fitness are shown in Fig.6. Fitness starting at different initial heights, that is to say, a generality, mostly increases in early generations. However, as the fitness for 0[m] that is an evolutionary guidance increases enough, a generality tends to be lost. A generality for -10[m] is easier to be lost than one for 10[m].

We call this reduction of generality "over-evolution". It resembles the over-learning, which appears in a training of ANNs with sample data sets. An excessive learning reduces a generality in both over-learnings and over-evolutions. However, an over-evolution does not always happen, but happens by the probability.

To show a characteristic of over-evolutions numerically, we have experimented on evolution three hundred times. Each evolution is continued to the two thousandth generation. The average fitness is shown in Fig.7. The average fitness for 0[m] gradually increases, while the average fitness for 10/-10[m] decreases after a certain generation in a middle of evolution. The probabilities that the fitness for 10/-10[m] is within a range from a half times to one and a half times as many

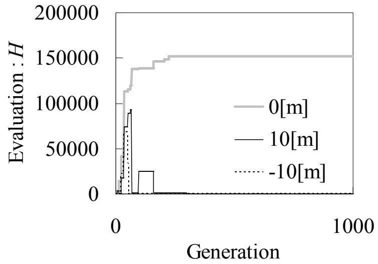

Fig.6. The typical evolution

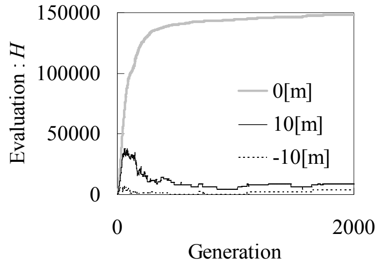

Fig.7. The average fitness of 0[m]

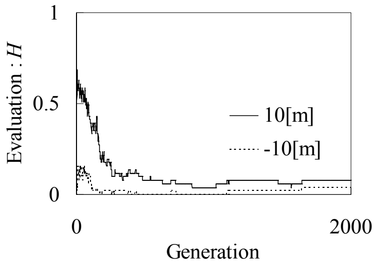

Fig.8. The probability

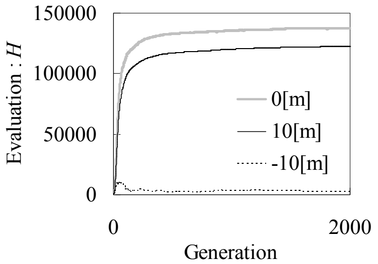

Fig.9. The average fitness of 10[m]

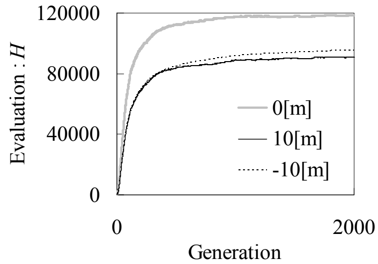

Fig.10. The average fitness of -10[m]

as the fitness for 0[m] are shown in Fig.8. As evolution progresses, the probability decreases.

2. Variation of generality by aimed tasks

Based on the results above, we set a hypothesis; the generality depends on the behavior led by given conditions. In our evolution, the best behavior is a flying forward as possible as fast at height of 0[m]. Compared with 0[m], a descending from 10[m] to 0[m] is additionally needed in starting at 10[m]. In the same as 10[m], flying up is additionally needed in starting at -10[m]. The best behavior in starting at 10[m] or -10 m] includes one in starting at 0[m], therefore we supposed that a generality of the ANN learned by the evolution in starting at 10[m] or -10[m] is higher than one at 0[m].

The average fitness in three hundred evolution started at 10[m] is shown in Fig.9 and one of -10[m] is shown in Fig.10. The ANN evolved by starting at 10[m] performs well in starting at 0[m] and 10[m] only, while the ANN evolved by starting at -10[m] is successful in all three cases. Flying up is more difficult action than flying down, therefore this result probably happened. In this simulation, the flight starting at a low height is a "critical task", which leads to higher generality.

VI. CONCLUSION

We modeled a butterfly based on the physical law and controlled it by EANNs. By many evolutionary simulations, it is shown that an over-evolution problem can happen in evolution of EANNs. The generality of EANNs are rarely discussed. However, if a universal method to achieve high generality is discovered, EANNs become more useful. Our research can be a key for leading to this great discovery. In the future, there will be need to examine whether an over-evolution can appear in other model or not.

REFERENCES

[1] Terzopoulos D, Tu X, Grzeszczuk R (1994), Artificial fishes: Autonomous locomotion, perception, behavior, and learning in a simulated physical world. Artificial Life IV, pp.327–351
[2] Usami Y (2007), Re-examination of swimming motion of virtually evolved creature based on fluid dynamics. Advances in Artificial Life, pp.183-192
[3] Wu J, Popovic Z (2003), Realistic modeling of bird flight animations. Proceedings of ACM SIGGRAPH 2003, pp.888-895
[4] Sims K (1994), Evolving virtual creatures. Proceedings of the 21st annual conference on Computer graphics and interactive techniques, pp.15-22
[5] Reil T, Husbands P (2002), Evolution of central pattern generators for bipedal walking in a real-time physics environment. IEEE Transactions on evolutionary computation 6(2):159-168
[6] Yamamoto M, Iwadate K, Ooe R, et al (2009), Autonomous animated robots. International Journal of CAD/CAM 9(1):85-91
[7] Yao X (1999), Evolving artificial neural networks. Proceedings of the IEEE 87(9):1423-1447