ACQUISITION OF JUMPING BEHAVIOR ON THE ARTIFICIAL CREATURE UNDER VIRTUAL PHYSICAL ENVIRONMENT

Yuta Umemura, Ikuo Suzuki, Masahito Yamamoto and Masashi Furukawa
Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan

Keywords: Artificial life, Physics modelling, Jumping, RCGA, Behavior simple, Behavior composed.

Abstract: Walking and jumping are very effective movement in a debris area. However, it is difficult to jump successively because it has a lot of difficulties (e.g. controlling the strong power at taking off and suppressing an impact at landing). This paper proposes how to acquire the successive jumping motion. We model an artificial creature like a locust under the physical virtual environment and control it by using Artificial Neural Network (ANN). In order to realize the successive jumping motion, this paper proposes a concept of “Behavior Simple (BS)” and “Behavior Composed (BC)”. The concept of BC is that a complex behavior is composed of plural simple behaviors. We consider that the successive jumping is divided into three BSs, taking off, getting up and returning leg back motion. After three BSs are trained by using the Real-Coded Genetic Algorithm (RCGA) independently, BC is trained by using RCGA as well. Experiments verify that the efficient successive jumping can be acquired.

1 INTRODUCTION

A study on jumping motion has been applied to various fields such as robotics (Fu et al., 2010), biological analysis (Gronenberg, 1996) and so on.

Jumping is a complex motion because it has a lot of difficulties (e.g. controlling strong power at taking off, keeping correct posture and suppressing an impact at landing) so that most of studies focus on only a part of jumping motion. Sutton and Burrows (2008) analyzed the mechanics of taking off of the locust. (McKinley et al., 1983) and (Nauwelaerts and Aerts, 2005) analyzed the power of landing forces. However, it is required to build up a more practical jumping model and a system considering some difficulties in jumping.

We create the locust model as jumping model under the virtual physical environment. It can jump, walk and act various kinds of movement. This paper proposes how to acquire the successive jumping motion that capable of repeating big jumping.

We consider that the successive jumping motion is divided into three simple behaviors, the kicking ground, getting up from the overturning situation and returning leg back motion. A concept of "Behavior Simple” (BS) and “Behavior Composed” (BC) (Furukawa et al., 2010) is applied to this idea. This concept is that a complex behavior is realized by switching plural simple behaviors. In this paper, the successive jumping motion is BC and the kicking ground, getting up and returning leg back motion are BSs. Optimizing the system of switching three BSs realizes the successive jumping motion as BC (Figure 1).

The rest of this paper is constructed as follows. Section two proposes the locust model and a simple experiment. Section three proposes three experiments to get three simple behaviors. Then, section four proposes one experiment to get the successive jumping motion. Finally, our work is summarized in section five as conclusion.

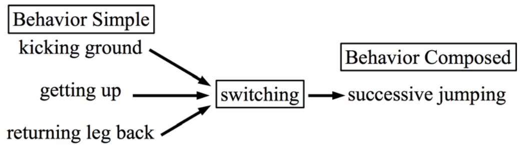

Figure 1: The successive jumping motion is realized by switching three BSs.

2 THE LOCUST MODEL

2.1 The Locust Model

We create the locust model under the virtual physical environment by using PhysX. PhysX is a physics motion engine developed by NVIDIA.

Umemura Y., Suzuki I., Yamamoto M. and Furukawa M..
ACQUISITION OF JUMPING BEHAVIOR ON THE ARTIFICIAL CREATURE UNDER VIRTUAL PHYSICAL ENVIRONMENT.
DOI: 10.5220/0003671903110314
In *Proceedings of the International Conference on Evolutionary Computation Theory and Applications* (ECTA-2011), pages 311-314
ISBN: 978-989-8425-83-6

The model consists of four parts, body, fore legs, middle legs and back legs as shown in Figure 2. The density of each part is 300 [kg/m^3] as the locust has.

The actuator is implemented to joints between a body and a femur and between a femur and a tibia at each side of back legs. The model is controlled by 4 actuators and the angle range of that is shown in Figure 3.

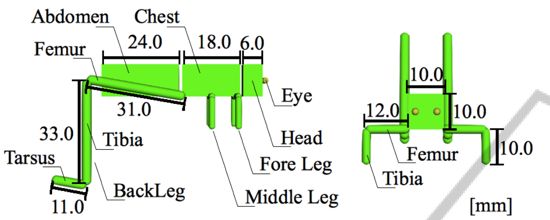

Figure 2: The locust model consists of a body, fore legs, middle legs, and back legs.

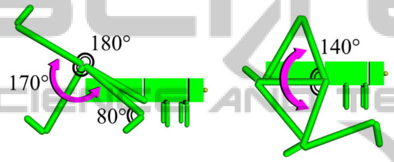

Figure 3: Each actuator rotates within specified angles.

2.2 Rule-based Successive Jumping Avoiding Overturning

One of the ideas to realize successive jumping is repeating jumping motion with avoiding overturning. This jumping motion is achieved by repeating the kicking ground motion and kicking legs back motion. Rule-based system is acquired by trial and error. In this system, the time during which the tarsus touches the ground is regarded as the touch time. We define the change time as a certain time needed for recovering the attitude. If the touch time reaches the change time and the tarsus touches the ground, the model acts by kicking the ground motion. The touch time is reset when the tarsus leaves the ground and until the touch time reaches the change time again, the model acts by the kicking legs back motion.

Figure 4 shows the initial state of the model. It is set on the origin of the coordinate system and its heading is set along the x-axis.

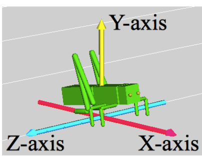

Figure 4: The model turns its head to forward direction of the x-axis of world axis at the initial state.

Figure 5 shows the trajectory of the successive jumping motion by using appropriate rules found empirically with the height of the model along the vertical axis and a travel distance of the model along the horizontal axis.

It is noticed that the rule-based successive jumping is not satisfactory because as jumping is repeated, the height and distance becomes small (Figure 5). Actually, the posture of the model becomes poor step by step because the model does not have the recovery motion. In addition, once it overturns, it cannot get up and jump any more. This paper aims at acquiring the successive jumping motion with big jumping and generalization.

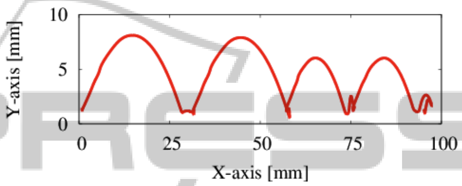

Figure 5: The successive jumping motion is acquired by repeating specified motion.

3 THE BEHAVIOR SIMPLE

To realize our purpose, it is not enough to control the model with simple control system.

Then, Artificial Neural Network (ANN) is employed to control all actuators. The model acquires appropriate motion by optimizing synaptic weights of ANN by Real-Coded Genetic Algorithm (RCGA). The evaluation function is defined in each experiment.

In this experiment, the ANN has nine units in the input layer, 10 units in the hidden layer and four units in the output layer. The following items represent nine inputs.

- angles at four actuators I_1~I_4 [0, 2 π]
- a touch sensor of back legs and the ground I_5 (off : 1 or on : -1)
- $1/(H+1)$ (H is the height of model) I_6 [0, 1]
- the inner product of the local axis (Figure 6) and the world axis (Figure 4) I_7~I_9 [-1, 1]

Four outputs, O_1~O_4 [-10° , 10°] is displacement angles every 1/120 [s] for each actuator.

In RCGA, we have 50 individuals as a population. RCGA is terminated when the generation number becomes 1000. As genetic operations, elite preserving strategy, crossover and mutation operations are used.

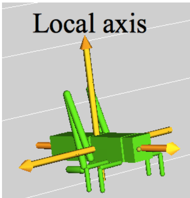

Figure 6: The local axis of the model is defined.

3.1 Acquiring the Kicking Ground Motion

The model acquires the kicking ground motion as the taking off motion. In this experiment, the posture of the model and the landing are not considered. The initial state and position are shown in Figure 4. We simulated 300 steps in each generation of RCGA for evaluating the individual. 1 step is 1/120 [s] on PhysX.

The fitness function is set as expressed in Eq.(1). Maximizing E means maximizing a square area of a rectangle consists of the initial point and the top point on a single jumping.

$$E = y_{\max} \cdot x_t \quad (1)$$

The trajectory of acquired motion and rule-based successive jumping motion is shown in Figure 7. The height of the acquired motion is about twice as high as the rule-based successive jumping motion.

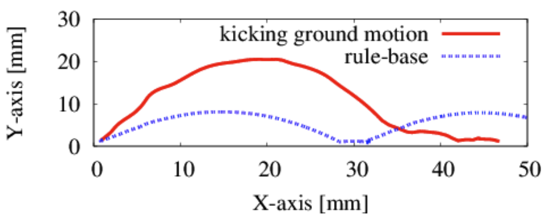

Figure 7: The height of the kicking ground motion is about twice as high as the rule-based successive jumping motion.

3.2 Acquiring the Getting Up Motion

The model acquires the getting up motion as a part of recovery motion. At first, the vector U (Figure 8) is defined which is one of the local axes (Figure 5) and represents an upward direction of the model. Five initial states are defined as the overturning states. Those are rotated from one side position to the opposite side position every 45 degrees (Figure 8). We simulated 1000 steps for each individual in this experiment.

The fitness function is set as Eq.(2). Maximizing U_y leads the model to get up from the initial state as soon as possible.

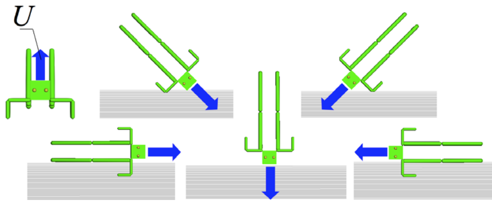

Figure 8: The vector U and 5 initial states are defined.

Figure 9 shows the one of the successful getting up motion and the model gets up from all of initial states.

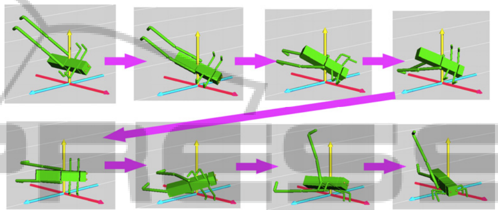

Figure 9: The model gets up from plural overturning states.

$$E = \sum_{S=1}^{5} \sum_{t}^{1000} U_y(t) \quad (2)$$

The model gets up from not only 5 initial states but also other initial states. We set 180 additional initial states which are rotated from one side position to the opposite side position in every 1 degree. The model gets up from 65 additional initial states by acquired motion. It verifies that acquired getting up motion is effective to recover from overturning.

3.3 Acquiring the Returning Leg Back Motion

The locust acquires the returning leg back motion as a part of recovery motion. In experiment acquiring the getting up motion, it is not evaluated about the angle of each actuator. When the model gets up, it may not be the appropriate angle for the next jumping.

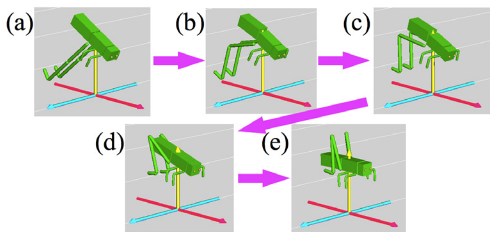

Figure 10: The model gradually bending each joint to the jumping initial position.

The model must learn the angle ready for the next jumping.

We simulate 1000 steps and the initial position is set as shown in Figure 10 (a). The fitness function is set as shown in Eq.(3).

$$E = -\sum_{t=1}^{200}\sum_{i=1}^{4}\left|\varphi_i - \theta_i\right| \tag{3}$$

Where, ϕ is the initial angle of each actuator and θ is the angle of each actuator at t step. The model acquires the motion for bending each actuator to the goal position (Figure 10 (e)).

4 THE BEHAVIOR COMPOSED

After the locust acquires three BSs, the switching BSs system is optimized and it leads to acquire successive jumping motion. A higher ranked ANN is used as switching system. It consists of nine units in the input layer, 10 units in the hidden layer and three units in the output layer. Its nine inputs are the same as those of BSs' ANN. Its three outputs [0, 1] correspond to three BSs' ANN. One of the BSs' ANN is selected which corresponding output has the largest value among those of three.

We set 3000 steps for each individual. A higher ranked ANN select one of BSs' ANN in every 120 steps. We define the initial position as shown in Figure 11 to make it easy to acquire the successive jumping motion. The fitness function is set as Eq.(4). Maximizing the accumulated heights of the model leads to jump successively.

$$E = \sum_{t=1}^{3000} y_t \tag{4}$$

Figure 12 shows the trajectory of the acquired motion and rule-based successive jumping motion. The lower part indicates the selected BS at that time. The acquired motion is more than twice as high as rule-based successive jumping motion at all jumps. It verifies that the switching BSs system works well and appropriate BSs' ANN is selected at that time.

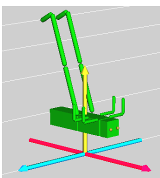

Figure 11: The initial state of acquiring the successive jumping motion is one of the initial states of the getting up motion.

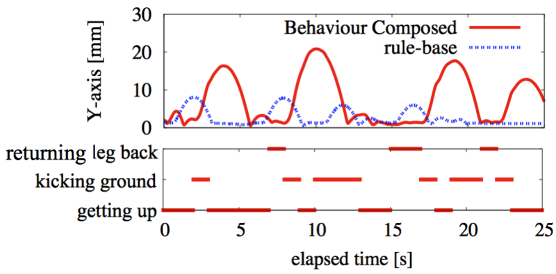

Figure 12: Successive jumping acquired by BC and selected BS.

5 CONCLUSIONS

We create the model under the virtual environment as jumping model and use ANN as the controller.

Three BSs, the kicking ground, the getting up and the returning leg back motion are acquired respectively by RCGA. After that, the model acquire the successive jumping motion as BC by compositing three BSs.

More complex behavior which includes jumping, walking and so on are acquired in a future work.

REFERENCES

X. Fu, F. Li, W. Liu, C. Stefanini and P. Dario. (2010). Design of a bionic Saltatorial Leg for Jumping Mini Robot. *ICIRA* (pp. 477-487).

Masashi Furukawa, Makoto Morinaga, Ryosuke Ooe, Michiko Watanabe, Ikuo Suzuki, and Masahito Yamamoto. (2010). Behavior Composed for Artificial Flying Creature. *SCIS & ISIS*.

W. Gronenberg. (1996). Fast actions in small animals: springs and click mechanisms. *J Comp Physiol A*, 178(6), 727-734

P. A. McKinley, J. L. Smith, and R. J. Gregor. (1983). Responses of elbow extensors to landing forces during jump downs in cats. *Experimental Brain Research*, 49(2), 218-228.

S. Nauwelaerts and P. Aerts. (2006). Take-off and landing forces in jumping frogs. *Journal of experimental biology*, 209(1), 66.

G. P. Sutton and M. Burrows. (2008). The mechanics of elevation control in locust jumping. *J Comp Physiol A*, 194(6), 557-563.