Implementation of "A Practical Model for Realistic Butterfly Flight Simulation"

Jiechang Guo UHID: 2084258

April 27, 2023

1 Problem

Butterflies are found everywhere and are known for their beautiful flying. However, how to generate the animation for the butterfly is still an under-explored problem. The paper "A Practical Model for Realistic Butterfly Flight Simulation" proposed a realistic model for simulating butterfly flights, for real-time graphics and animation applications[1]. In this project, we implemented the butterfly flight simulation based on the proposed model.

2 Method

The paper designed a model that includes how a butterfly moves its wings and body. Then they used a force-based model to control the butterfly's movements. The pipeline for the flight simulation is illustrated in figure 1 below. This model includes two types of forces, the aerodynamic forces from how air moves around the butterfly and vortex forces from the swirling air around the butterfly's wings. After calculating the force, the velocity of the butterfly is calculated based on the attraction target position and the calculated net force. In the end, given the velocity of periodic maneuvering functions, the rotations for each part of the butterfly will be calculated.

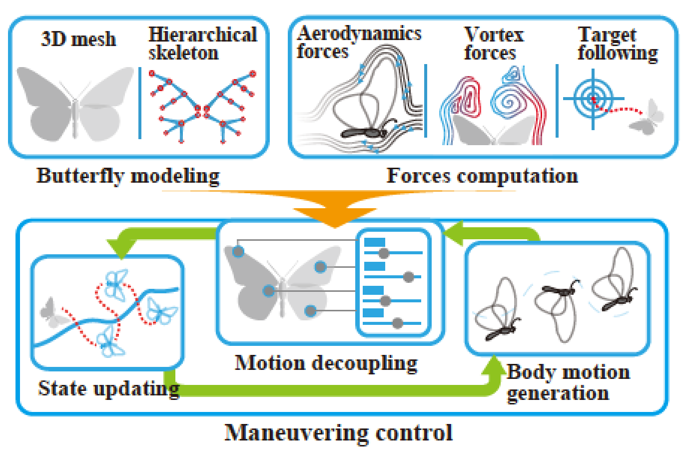

Figure 1: Pipeline[1].

2.1 Butterfly Structure

Here is an example of the structure of the butterfly defined in this paper. The constructed butterfly mesh model consists of five parts: head, thorax, abdomen, fore-wings, and hind-wings as shown in figure 2. The parameters for each part will be defined in the following section.

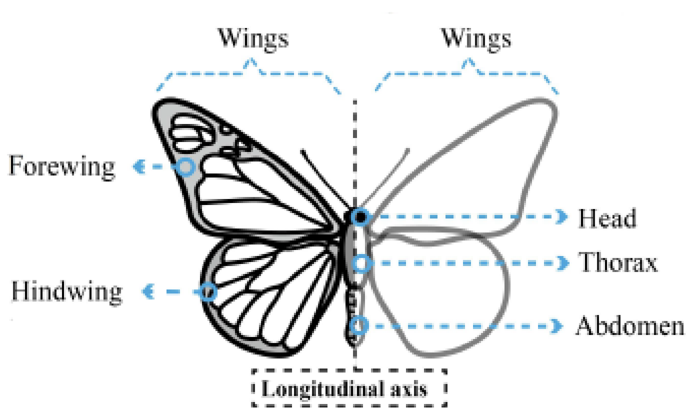

Figure 2: Butterfly structure[1].

2.2 Maneuvering Parameters

Butterfly wings are connected by joints and move in a certain way. The wings can only flap with a limited range of frequencies and phases. Researchers have also found that when a butterfly flies, it moves its body specifically the abdomen to balance the wing movements. To simulate these phenomena, they came up with specific parameters to model how the butterfly moves. Figure 3 shows how each part of the butterfly moves.

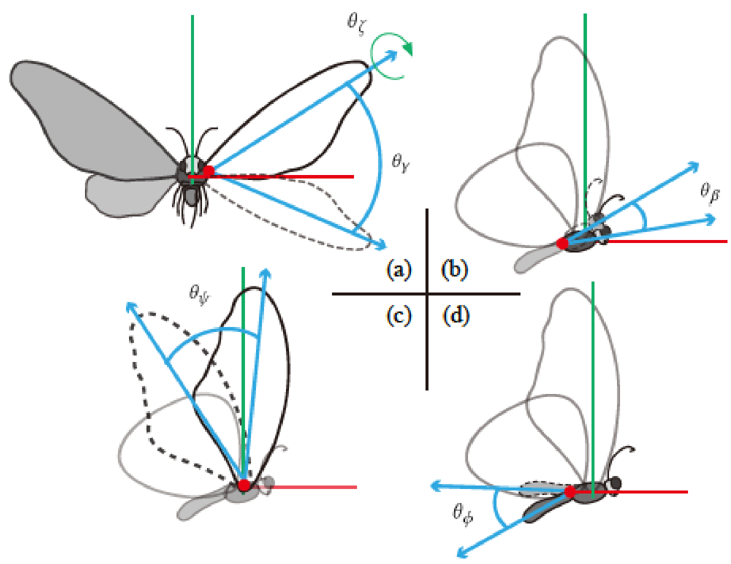

Figure 3: Butterfly Maneuvering Parameters[1].

Parameters for wings For simplicity, they take the bilateral wings' flapping with synchronous frequencies. The left and right wings will share the same parameters. The parameters of fore-wings are defined in 3DOF as the flapping angle θ_γ, feathering angle θ_ζ, and sweeping angle θ_ψ. The hind-wing only has 1DOF which is the flapping angle θ_γ shared with the fore-wing.
Parameter for thorax During the flight, the thorax coordinates the flapping of the wings through undulations. One controllable parameter is defined as the pitch angle θ_β of the thorax.
Parameter for abdomen When a butterfly intends to hover, climb up, or descend, its abdomen may visibly rotate along the body's longitudinal axis with the opposite phase to the flapping of its wings. The abdomen of the butterfly is assigned 1DOF defined as θ_ϕ.
The five parameters $\{\theta_\beta, \theta_\gamma, \theta_\zeta, \theta_\psi, \theta_\phi\}$ can effectively control the motion of the butterfly and simulate realistic flight animation.

2.3 Periodic Maneuvering Functions

The motion of the butterfly can be considered a periodic movement of the wings and body parts as shown in figure 4.

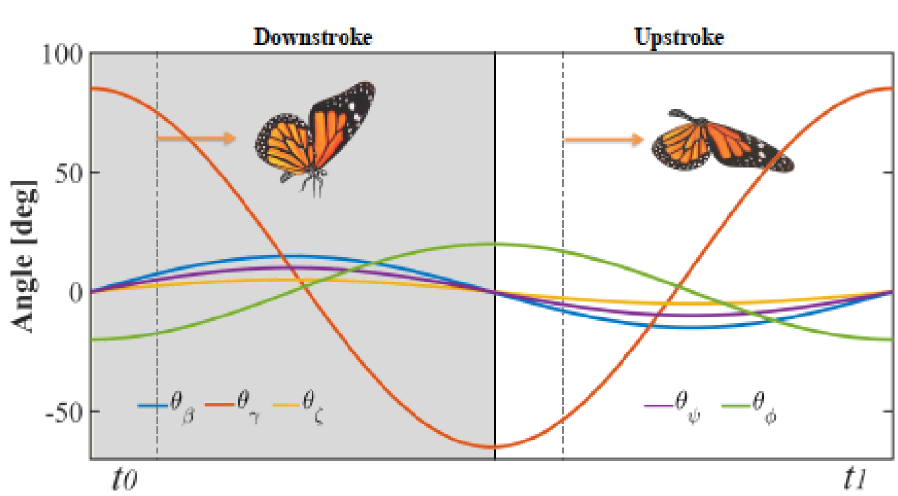

Figure 4: The phase shifts of the maneuvering angles of the butterfly during one wing flapping cycle.[1].

The paper defined periodic maneuvering functions to calculate the five angles for the butterfly. The equation is given below.

$$\theta_*(\varphi_a^*(u), f^*(u), \varphi_p^*, \varphi_m^*, t) = \varphi_a^*(u)\cos\left(2\pi f^*(u)t + \varphi_p^*\right) + \varphi_m^*$$
$$where* \in \{\theta_\beta, \theta_\gamma, \theta_\zeta, \theta_\psi, \theta_\phi\}$$

The inputs of the function include the amplitude range $\varphi_a^*(u)$ which is a function of the butterfly velocity u, frequency range $f^*(u)$ which is also a function of the butterfly velocity u, the mean value of the angle φ_m^*, and the phase angle φ_p^* to enable the wing-abdomen, wing-thorax interaction, and also current time t.
In the equation above, the cosine function is used to enable the periodic motion control of the butterfly. The interaction between abdomen and wing, thorax and wing are enabled with different phase angles for each parameter. The table 5 below lists all the parameter values for the equation. Our implementation is based on these equations and parameters.

<table>
<tr><th rowspan="2">Angle</th><th colspan="4">Parameter Value</th></tr>
<tr><th>R_{f_a} (Hz)</th><th>R_{θ_a} (°)</th><th>φ_p (°)</th><th>φ_m (°)</th></tr>
<tr><td>θ_β</td><td>0~3</td><td>0~30</td><td>-90</td><td>0</td></tr>
<tr><td>θ_γ</td><td rowspan="4">0~11</td><td>0~150</td><td>0</td><td>10</td></tr>
<tr><td>θ_ζ</td><td>0~10</td><td rowspan="2">-90</td><td rowspan="2">0</td></tr>
<tr><td>θ_ψ</td><td>0~20</td></tr>
<tr><td>θ_ϕ</td><td>0~35</td><td>-180</td><td>-10</td></tr>
</table>

Figure 5: Value of Parameters.[1].

2.4 Velocity Computation for Target Following

In the paper, they calculate the velocity from the object's local acceleration which is caused by the aerodynamic force, vortex force, and gravity. The local acceleration can be calculated according to equation(1), where the sum of F_j is the net force(the combination of aerodynamic force and vortex force) for the four wings, g stands for the gravity, and m stands for the mass of the butterfly.

$$a_{loc} = \left(\sum_{j=1}^{4} F_j + F_{vor} + g\right)/m \qquad (1)$$

As well as the preferred acceleration defined in equation(2) that leads the butterfly to approach the target position. The preferred acceleration is determined by the target position q, the current position of the center of the butterfly body p, the mass of the butterfly m, and also the distance between the butterfly and the target point d in equation(3). If the distance is larger than the defined maximum sensory length L in the field of view of the butterfly, the butterfly would not be attracted. Also, a ramp function $R(d)$ in equation(5) is introduced to smoothly cool down the velocity of butterfly when the it approaches the target.

$$a_{pre} = R(d)\frac{1}{m}\frac{p-q}{|p-q|} \qquad (2)$$
$$d = min(1, |p-q|/L) \qquad (3)$$
$$R(d) = \begin{cases} \frac{15}{8}d - \frac{10}{8}d^3 + \frac{3}{8}d^5, & d \leq 1 \\ 0, & \text{otherwise} \end{cases} \qquad (4)$$

Finally, the velocity of the butterfly can be calculated as the following equation:

$$u_t = u_{t-1} + (a_{loc} + a_{pre})\Delta_t \qquad (5)$$

3 Implementation

The project is implemented on Apple M1 Pro with 16 GB memory using C# and Unity3D 2021.3.10f1. We focus on the implementation of maneuvering control and target following.
For the implementation of maneuvering control, according to the equation, we set the speed as a normalized scalar value from 0 to 1. To achieve faster-flapping frequency, we enable the control of frequency range for users.
For the velocity calculation to enable the butterfly the ability to approach an attraction target. We only implemented the preferred acceleration to enable the target following of the butterfly. And we assume that

the local acceleration is zero, which means that the flapping of the wing can cancel out the gravity. We use the free asset "Bezier Path Creator" from Unity asset store to generate the user-defined path[2]. At the runtime, we set the attraction target as the point with the closest distance to the center of the butterfly along the path.

Actually, we tried to implement the aerodynamic force for the butterfly, however, we failed to figure out the appropriate parameters like the butterfly weight, area ratio, air density, and also the air velocity. Here we show that we successfully find the vertices and faces that belong to each wing for future calculation.

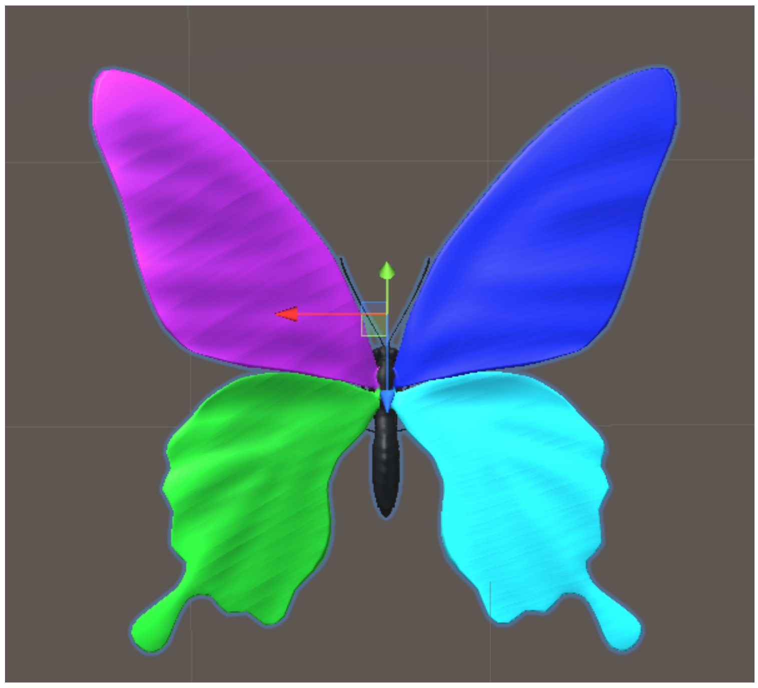

Figure 6: Wings segmentation.

4 Result

4.1 Wing and Abdomen Interaction

In figure 7, the butterfly is in its rest pose. In figure 8, the abdomen is rotated along the longitude axis of the butterfly. The results show the simulation of abdomen and wings interaction. The abdomen of the butterfly rotates with the opposite phase to the wings' flapping when it plans to hover, climb up, or move down. In the left image of 8, the butterfly's wings are flapped up, and the abdomen is lower than the thorax, on the other hand, in the right image of 8, the butterfly's wings are flapped down, the abdomen is higher than the thorax.

Figure 7: Butterfly in rest pose.

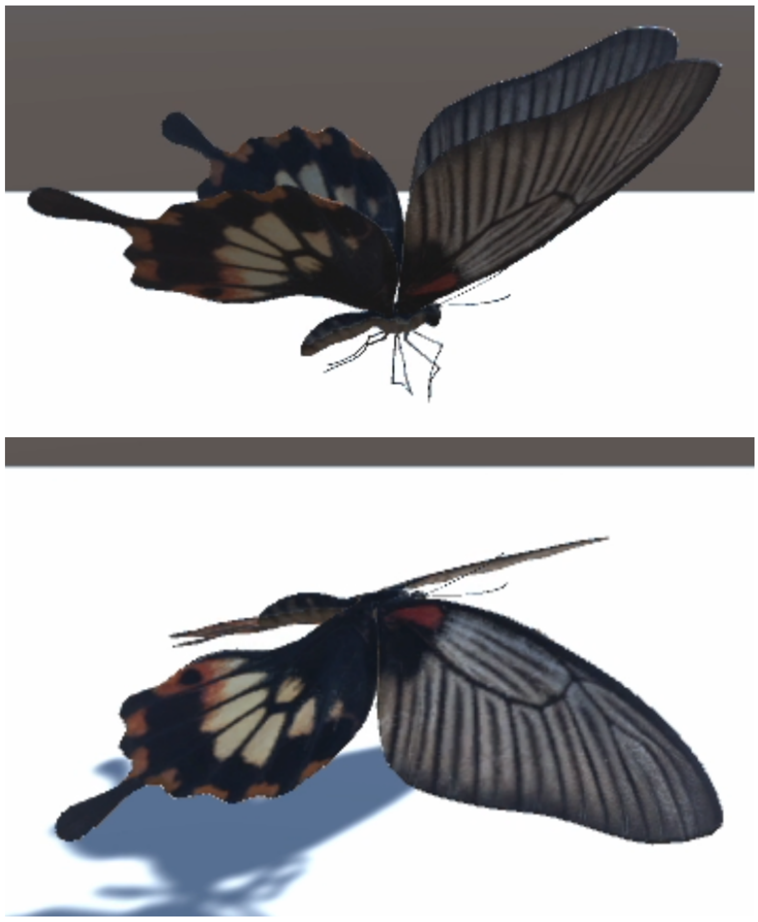

Figure 8: Wing and Abdomen Interaction.

4.2 Speed and Frequency Control for Butterfly motion

In figure 9, there are two sliders for users to control the parameters of the butterfly motion. The speed can be set from 0 to 1, and the frequency is from 0 to 30. The higher the values are, the faster the butterfly will flap its wings. Please check the supplement video for detail.

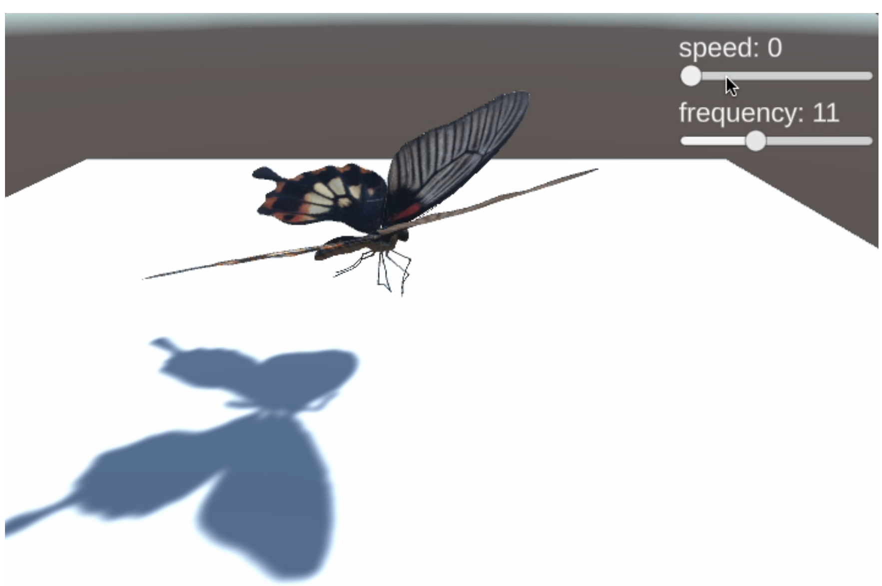

Figure 9: Speed and Frequency Control GUI.

4.3 Butterfly Fly along User-defined Path

In figure 10, the butterfly is flying along a user-defined path in green.

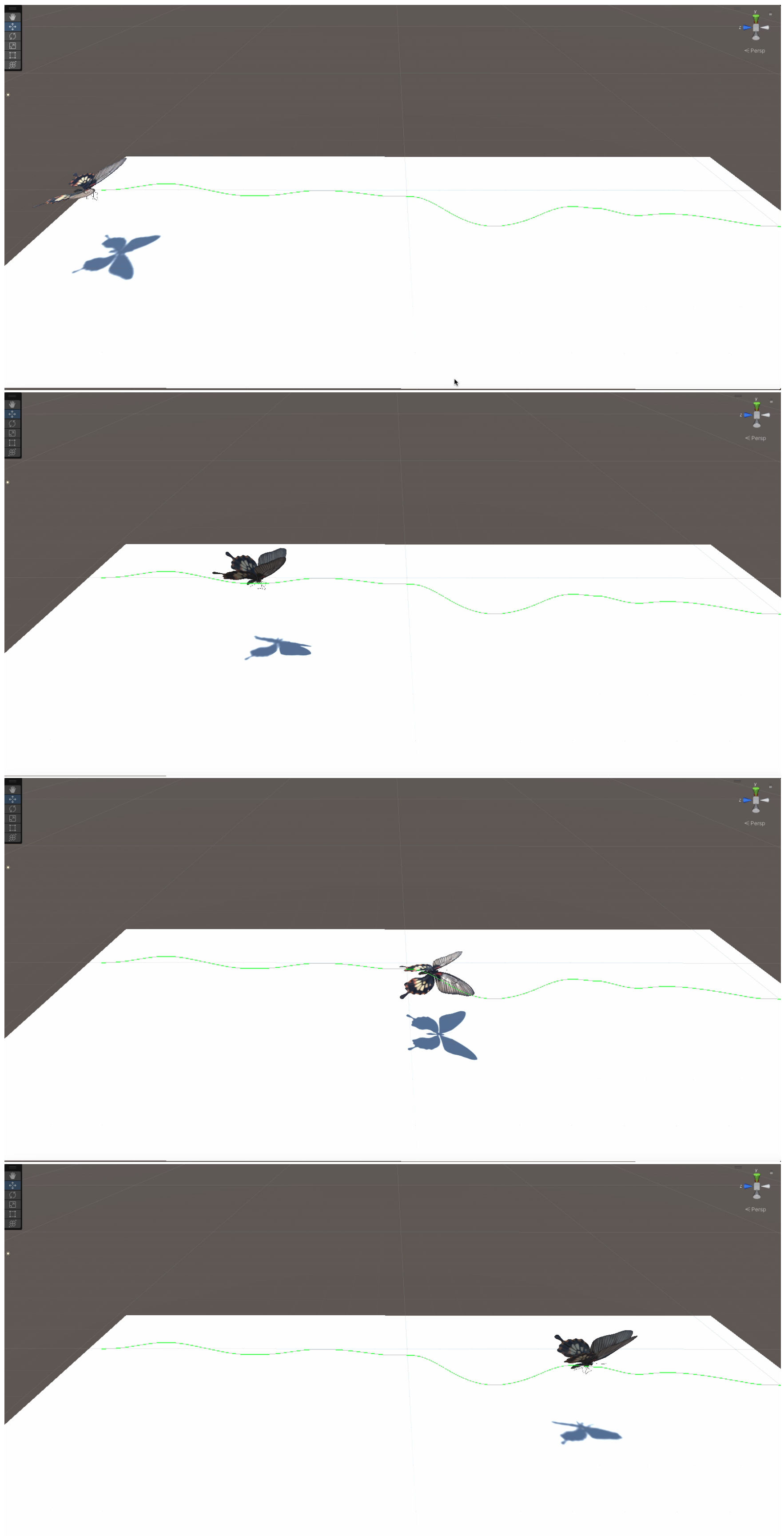

Figure 10: Fly along User-defined Path.

5 Future Work

In the future, we plan to implement the aerodynamic forces and vortex forces. And also introduce more subtle animation on the wings joints other than only 3DOF for the root of each wing.

References

[1] Qiang Chen et al. "A Practical Model for Realistic Butterfly Flight Simulation". In: *ACM Transactions on Graphics* 41.3 (). DOI: 10.1145/3510459. URL: https://par.nsf.gov/biblio/10359111.

[2] Sebastian Lague. *Bézier Path Creator*. https://assetstore.unity.com/packages/tools/utilities/b-zier-path-creator-136082.