Evolving Buildable Flapping Ornithopters

Floris van Breugel[1], Hod Lipson[2]

Computational Synthesis Lab
[1]Biological Engineering
[2]Mechanical & Aerospace Engineering
Cornell University, Ithaca NY 14853, USA
+1 (607) 255 1686

hod.lipson@cornell.edu

ABSTRACT

We present a process of evolving flapping flight control patterns for an ornithopter. We focus both on generating the flapping motion pattern as well as a realistic kinematic mechanism that can generate this motion in practice. The ultimate goal of the project is to translate the simulated models and behavior into a realized physical model capable of independent untethered flight.

Categories and Subject Descriptors

J.2 [Physical Sciences and Engineering]: *Aerospace, Engineering, Physics.*

General Terms

Algorithms, Design, Experimentation, Theory

Keywords

Evolution, Genetic Algorithm, Ornithopter, Flapping Flight, Machine Learning

1. INTRODUCTION

Humans have long been fascinated by flight, but have still not yet succeeded in creating an elegant device capable of hovering solely by flapping [2]. There are, however, many such designs in nature, ranging from hummingbirds to small insects. Hitherto man has always relied on the use of an airfoil and forward motion to create enough lift to fly, for examples see [5]. The challenge this project is addresses is the creation of an ornithopter that gains enough lift to fly only by means of the drag forces on a flat wing. This is a difficult problem and would be nearly impossible to solve manually. Thus it provides an excellent challenge for evolutionary design [1] [3] [11].

Much work has been done with evolving flight in simulation, but few attempts have been made to actually build the resultant machines or even consider its buildability or practicality in terms of weight to lift ratio [14] and [17]. Most work on evolutionary ornithopters are concerned with the flapping motion itself, without regard to the kinematic mechanism that might generate it in practice. Those attempts that have been made have shown promise [2], but we are thus far unaware of any successful ornithopters that fly without the aid of airfoils.

There are several advantages to developing an ornithopter that does not require forward motion to fly. The primary benefit of such flight is that it would allow the machine to hover. Currently the only such devices are helicopters, which are able to create limited amounts of lift due to the rotor design. A flapping ornithopter is able to provide more lift since it can use the entire wing surface to create a lifting force from drag. Flapping flight confers many other advantages such as increased agility, independence of body orientation, and high adaptability to different situations [16]. Such mechanisms are of great interest to aerospace engineers in general, military operations, search and rescue, and surveillance technologies [16].

Other projects are attempting to solve this problem using the benefit of scale [4] [8] [10] [12] [18]. It has been shown that the fluid dynamics that are important to flight vary with the size of the machine. On a small scale the air acts more as a viscous fluid through which insects must swim, and an airfoil gives no benefit. The research team at Berkeley is attempting to create a micro-mechanical device based on the mechanical properties of insect flight rather than using evolutionary design.

The goal of this project is to use evolution to create a large scale ornithopter (one the order of a 1-2m wingspan) with evolved control patterns. The larger scale will make it easier to work with and adjust. The first stage, on which this paper focuses, is to create a simulation of the ornithopter on the computer and using a genetic algorithm to evolve successful flight and control patterns. Once this has been accomplished, mechanical designs will be partially evolved to help aid in the physical design of an ornithopter. Then control patterns for this machine will be re-evolved in hardware in a similar fashion used for the simulated models to produce a real flapping ornithopter.

2. METHODS

2.1 Dynamics Simulation

Simple ornithopter machines and the associated physics were modeled using a rigid body dynamic simulator [15]. The simulator uses a highly stable integrator to model articulated rigid body structures, with emphasis on speed and stability rather than physical accuracy. This is important in selecting a physics engine which is used with a genetic algorithm as there are many iterations involved. In order to make the problem simpler, and the

Genetic and Evolutionary Computation Conference (GECCO) '05, June 25-29, 2005, Washington DC, USA.

program more robust, collisions were not used. Full gravity and reasonable weights, sizes, and torques were implemented to keep the model as realistic as possible. ODE does not include any aerodynamic equations. Aerodynamics were modeled with highly simplified drag equations to calculate drag forces on the wings at each integration step.

2.1.1 Aerodynamics
Since the wings are modeled as simple rectangular surfaces there is no lift due to an airfoil involved. All of the effective lift comes from the perpendicular drag produced when the wings are pushed down. The force produced is calculated using the simplified drag equation [6]:

$$Drag = \frac{1}{2}\rho C_d A_{\perp} V_{\perp}^{\ 2}$$

Where ρ is the fluid density and C_d is the drag co-efficient of a flat rectangular plate. Drag is calculated for the entire wing based on the average squared perpendicular velocity of the wing relative to the body. The resultant force is applied as a point force at the center of the wing, perpendicular to its plane. Drag on the body is assumed negligible compared to the wings, as the body will function only as a lightweight frame to carry the servos and computer chip. Methods for calculating drag were adapted for high speed simulations for a flat plate from more computational intensive methods [13] [14] [17].

2.1.2 Direct Actuation Control
In this program four rectangular wings were directly attached to a rectangular body with ball joints. The range of motion was limited to just under 90 degrees rotation in each direction in order to keep the flapping patterns realistic and to remove any discontinuities at exactly ninety degrees. The wings were controlled by setting the joint velocity (rotation) in each of the three directions at each time step according to the evolved control pattern. All four wings were forced to be mirror images of each other.

The control pattern used was a simple sinusoidal wave with separate evolved amplitudes and phases for each of the three directions. All three directions had the same frequency, which was also evolved.

2.1.3 Cable Actuation Control
In order to model feasible flight patterns more realistically, a mechanical design was developed based on servo control. The design included three servos attached to each wing through a symmetrical cable system such that when the servo turned in one direction one cable would pull up on the wing and the other would give slack at the bottom of the wing. Attachment points for the cables to the wings were chosen near the connection between the joint of the wing and body so as to maximize range of motion while still maintaining reasonable torque output. Placement of the attachment points was arbitrary and fixed, chosen based on symmetry. One of the actuators was attached along the center of the wing, close to the body with its cable running parallel to the wing. The other two points were placed symmetrically off line from the center and slightly further out than the first point. The cables for these joints were placed at slight angles with the wing.

These actuators were modeled in ODE as linear sliding joints between the attachment point on the wing and a specified point above the body. The points of attachment to the wing and above the body were modeled with ball joints. At each time step the axis of the sliding joints were recalculated.

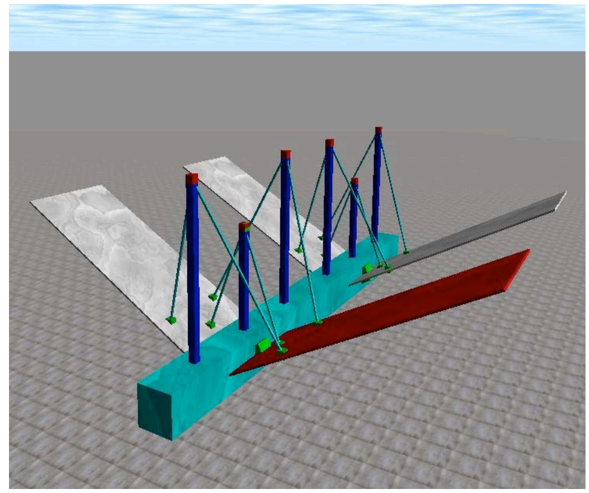

Figure 0. Cable Actuated Model. In practice cables extend symmetrically below wing and close a loop.

The control pattern used was again a pattern of velocity calculated at each time step applied to the joint. In this case the velocities represented linear velocity of the sliding joint. Two approaches were used for the control pattern: simple sinusoidal waves and 10 point Bezier curves. The sinusoidal curves used were of the same nature and form as in the directly actuated version.

The Bezier curves were created using an evolved set of 10 points and an evolved period that was the same for all three actuators. The amplitude (and sign) of the velocity was calculated according to the following equation [9]:

$$B(u) = \sum_{k=d}^{N} pk \frac{N!}{k!(N-k)!} u^k (1-u)^{N-k}$$

Where u is the normalized time, based on the set periodicity. B(u) gives the amplitude of the velocity of the linear actuators used to control the wing. By forcing the initial and final positions and slopes to match, the curve is forced to be smooth and periodic.

2.2 Use of Evolutionary Algorithm
The physics of controlled flapping flight is an extremely complicated problem with many parameters that are all linked together. The chances of a programmer manually designing a successful flapping pattern as well as the associated mechanical design to achieve that motion are very slim. Yet there are thousands of species of insects, birds, and bats that are all very well adapted to flapping flight. Modeled on the natural processes that have produced such remarkable mechanisms, simulated evolution has been shown to be a powerful tool for solving such difficult problems and producing effective behaviors [3] [11].

2.2.1 Implementation and Software
A single source code file is written in C++ and compiled with the Open Dynamics Engine for the physics and models. The code is platform independent and was compiled and run in a UNIX environment on a 1.5 GHz PowerBook G4.

2.2.2 *Algorithm*

The evolutionary algorithm used was a standard elitism genetic algorithm [7]. Each new individual of the initially random, and succeedingly evolved, population was created and simulated in the ODE environment. All individuals were started at a specified height with zero initial velocities and forces. The simulation was run for a set amount of time and the fitness was evaluated at the end. After each individual was tested, the population was ranked by fitness and the top 40% of the individuals were chosen randomly and evenly as parents to produce offspring to replace the bottom 40%. The middle 20% remained unchanged.

Matings were all between two different parents and produced two new offspring. Mutations were additively applied at an average rate of one per genome with a low end weighted exponential range between 0 and ±2 for the directly actuated models, and 0 and ±1 for the cable actuated models (for which the gene range was limited to ±1 and adjusted for appropriate range later). This ensured occasional large mutations and more commonly minor adjustments for fine-tuning. Two-point crossover was implemented at a rate of 80% in order to allow mixing of genes and increased variation. This method preserves possible functional groups of genes by moving large sections together.

The fitness of each individual was calculated based on the final height of the ornithopter at the end of the simulation (all simulations were set to last the same amount of time). At each time step the individual's bodies are checked for position, velocity, and angular velocity. If any of these values exceeds a realistic possible outcome (ie. if the model 'explodes'), it is given zero fitness and thereby essentially removed from the reproducing population. Typically this occurs if any part of the model becomes noticeably separated from the rest, or the model as a whole moves faster than 100m/s, or if the angular velocity exceeds 20 rad/sec. Stability is automatically selected for due to a long evaluation time (4-10 sec) since individuals with less stable control patterns will eventually tilt and fall.

3. RESULTS

3.1 Direct Actuation Control

A population of 100 initially random individuals was used. Each was tested in simulation for 10 seconds at a time step of 0.002 seconds. The total weight was 50 grams and the wings were fixed at 70 cm by 30 cm. The joints were given a maximum force rating of 1 N. The evolution seemed to converge after about 200 generations, with a runtime of 7 hrs. The fittest member of the final population produced a maximum total lifting force of 8 N and a maximum velocity of 11m/s upwards.

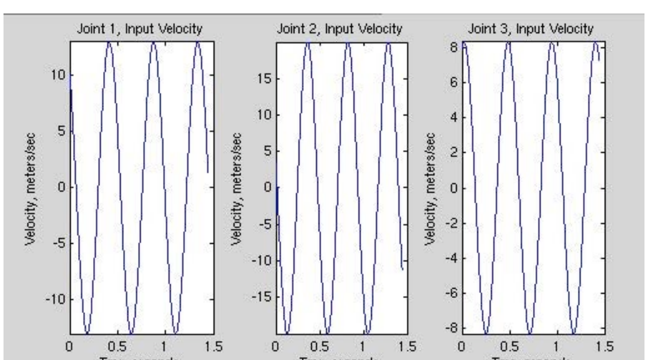

Figure 1. This graph shows the evolved pattern of input velocities for each of the three directions of control.

The evolved control pattern seemed to focus on powerful and fast downbeats with the wings fully perpendicular to the direction of flapping while turning the wings in order to slice through the air with minimal drag on the way up, as would be expected based on the flight patterns of many hummingbirds.

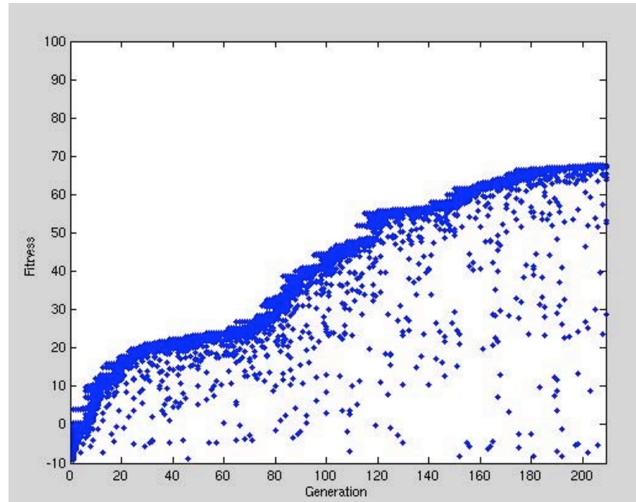

Figure 2. Fitness versus generation. This graph shows the progress of the population as it evolves. Each point represents an individual and its fitness, meters attained above starting point after 10 seconds.

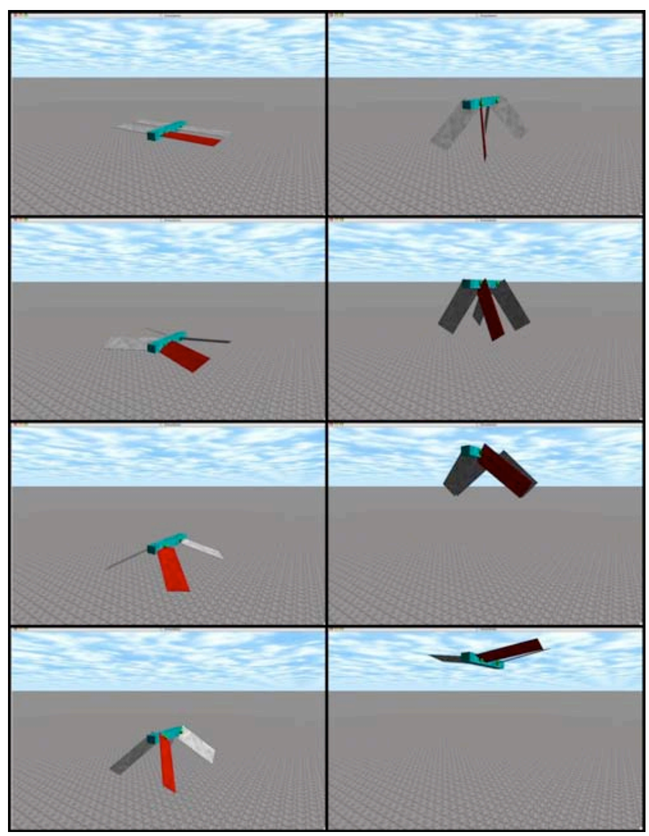

Figure 3. This image sequence shows one cycle of the evolved flapping pattern (top to down, left to right).

3.2 Cable Actuation Control

3.2.1 Sinusoidal Control Pattern

A population of 100 initially random individuals was created. The mass was set to 250 grams to simulate a lightweight balsa wood body frame, set of light servos and controller chip. An external power source was assumed to help increase the chances of a successful flapper. Wings were set to 30 cm by 70 cm, and each actuator was given a maximum range of 6 cm and a force rating of 6 N. Individuals were run for a total of 4 seconds at a simulation step of 0.002 seconds. An individual that exhibited positive flight did not evolve until after about 40 generations, and the evolution seemed to converge after 80 generations. The runtime was about 18 hours, which was much longer than for the directly controlled version because of the increased computation time for so many bodies and joints. The fittest member of the final population was barely able to maintain its altitude and was rather unstable

The evolved pattern resembled that of the figure eight pattern dragonflies tend to have. This is probably due to the limited range of motion that can be achieved compared to that of the directly controlled actuation. The pattern that was developed had a frequency of about 16 cycles per second and a maximum input velocity of 50 m/s. These values are beyond what is physically realizable with standard servomotors. In addition the fittest model was not nearly as stable as the direct actuation controlled models or the cable actuated Bezier pattern controlled models.

3.2.2 Bezier Control Pattern

Again a population of 100 initially random individuals was used. In order to make the simulations more realistic the mass was increased to 310 grams total, including 15-gram wings with a maximum linear force of 50 N. The mechanical layout and sizes were the same as in the sine control pattern. Initially the population contained no control patterns that were capable of generating positive lift, and one did not appear until after approximately eight generations. In the direct actuation control run the initial random population contained some machines that were already capable of flight. The fittest member after 73 generations flew upwards with a maximum total lifting force of 190 N and a maximum upwards velocity of 12 m/s.

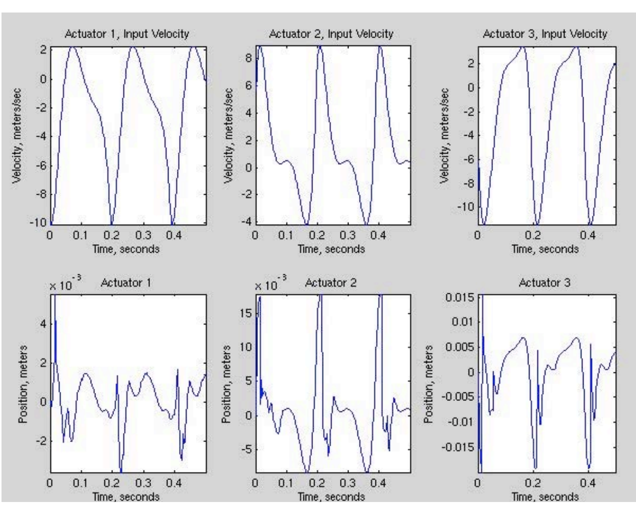

Figure 4. These graphs show the fully evolved pattern of input velocities and the resultant positions of the actuator ioints over time.

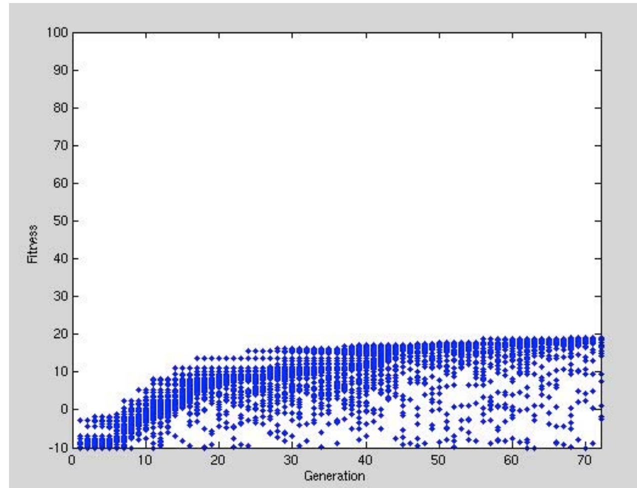

Figure 5. This graph shows the progress of the population as it evolves. Each point represents the fitness of one individual. Fitness is measured in meters attained after 4 seconds.

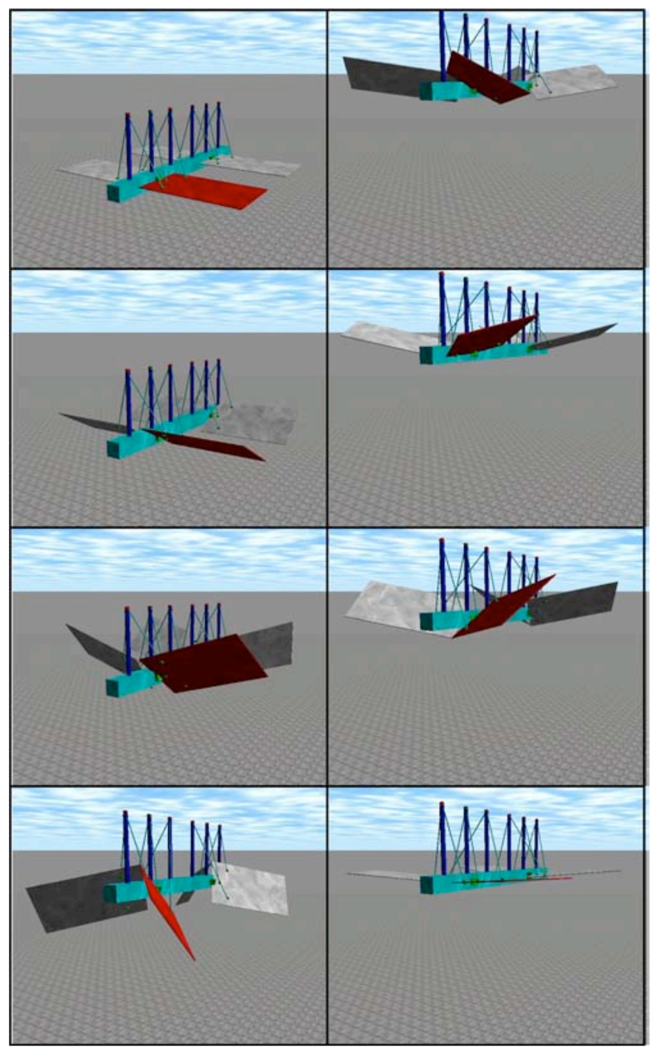

Figure 6. This image sequence shows the fully evolved Bezier control pattern over one period.

Although the masses in the two experiments were different, it can be seen in simulation and from the evolution graphs that the evolution with a realistic mechanical design and limitations is a much more difficult problem. This may be why the simple sine control pattern was not able to find an effective and stable pattern for the cable actuated design.

The evolved model controlled by the Bezier curves is a stable and effective flier, which uses patterns and a design that approach what can realistically be built with available servomotors. The maximum linear velocity of all three actuators reached 8 m/s and the pattern had a frequency of approximately 6 cycles per second. The frequency is a bit high, and the accelerations are relatively large at the peaks. Future work will aim to minimize these aspects as well as lowering the maximum force requirement.

4. DISCUSSION & FUTURE WORK

The ultimate goal of this project is to create a physical ornithopter with specifications close to those of the evolved simulated models. In order to create more realistic flapping patterns that can be achieved with off-the-shelf servo motors more work will need to be done to minimize the frequency and velocities. Rather than using a constant fitness function a second criteria can be added that rewards systems with patterns that are more energy efficient, which would mean less spikes in acceleration and lower maximum velocities. In order to fully take advantage of the Bezier curves the number of points can be made variable so as to remove any extraneous behaviors that would otherwise make the pattern less efficient.

So far the only parts that have been evolved are control patterns. Since changing physical aspects of the model will be difficult in a hardware version it would be wise to let the evolution take care of all aspects of the model. Preliminary tests have been run with the evolution having complete control of all parameters including wing size and actuator placement. What was immediately apparent is that the evolution selected for the largest wing surface area allowed to provide the largest amount of drag, yet resulting in no net lift. Since there are so many inter-related parameters another approach may be necessary such as evolution in stages and/or a variable fitness function.

Once a realistic design and set of control patterns is evolved a physical ornithopter will be built to best resemble the simulated model. Since the aerodynamics used in this simulation are drastically simplified (necessary to reduce total required runtime to a reasonable amount) the control pattern will not be much like one that will work in real air. Thus the purpose of this simulation is to show that something can achieve flapping flight with roughly the evolved configurations. The physical model will then be attached to a load cell sensor to measure x y and z forces and the evolution process will be performed in hardware. The final stage of the project will be to evolve control mechanisms to allow fully independent untethered flight (using evolved control systems similar to those shown in [19]). This will ensure that a real flapping mechanism can be evolved.

5. REFERENCES

[1] Atmar, W. “Notes on the simulation of evolution,” *IEEE Trans. Neural Networks*, vol. 5, no. 1, 1994, 130–148.

[2] Augustsson, P., Wolff, K., and Nordin, P. Creation of a learning, flying robot by means of evolution. In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 02), 2002.

[3] Banzhaf, W., Nordin, J. P., Keller, R. E. and Francone, F. D. Genetic Programming An Introduction: On the Automatic Evolution of Computer Programs and Its Applications. San Francisco: Morgan Kaufmann Publishers, Inc. Heidelberg: dpunkt verlag, 1998.

[4] Cox, E. Garcia, and Goldfarb, M. Actuator development for a flapping microrobotic mav. In *SPIE Microrobotics Symp*, pages 102-108, Boston, MA, 1998.

[5] DeLaurier, J. D. An aerodynamic model for flapping wing flight. Aeronautical J, 1993, 125-30.

[6] Fox, R. W. and McDonald, A. T. *Introduction to fluid mechanics*. fifth edition, Wiley, 1976.

[7] Mitchell, Melanie. *An Introduction to Genetic Algorithms*. MIT Press, 1998.

[8] Motazed, D. Vos, and Drela, M., Aerodynamics and flight control design for hovering MAVs. In *Proc of Amer Control Conf*, Philadelphia, PA, 1998.

[9] Mottenson, M. E. *Geometric Modeling*. John Wiley & Sons, 2nd edition, 1997.

[10] Mueller, T. J. Fixed and flapping wing aerodynamics for micro air vehicle applications. *Progress in Astronautics,* volume 195, 2001.

[11] Nolfi, S. and Floreano, D. Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines. Massachusetts: The MIT Press, 2000.

[12] Pornsin-Sirirak, T.N., Lee, S.W., Nassef, H., Grasmeyer, J., Tai, Y.C., Ho, C.M., and Keennon, M. MEMS wing technology for a battery-powered ornithopter. In *Proc. of IEEE 13th Annual Intl Conf on MEMS*, pages 799-804, Piscataway, NJ, 2001.

[13] Ramakrishnananda, B. and Wong, K. C. Animating bird flight using aerodynamics. *The Visual Computer.* 15(10), 1999, 494-508.

[14] Shim, Y., Kim, S. and Kim, C. Evolving Flying Creatures with Path Following Behavior. In *Proceedings of 9th International Conference on the Simulation and Synthesis of Living Systems* (ALIFE IX), Boston, Massachusetts, 2004.

[15] Smith R. Open Dynamics Engine (ODE), http://ode.org/, 2000.

[16] Smith, M. J. C. Reinstating inquiry into mechanized flapping-wing flight: realizing the integrity of the ornithopter. In Proc. 35th Aerospace Sciences Meeting and Exhibit , pp. 1–11. Reston, VA: AI AA, 1997.

[17] Wu, X. C. and Popovi´c, Z. Realistic Modeling of Bird Flight Animations. in *Proceeding of SIGGRAPH ’87*, 2003, 888-895.

[18] Yan, J., Wood, R., Avandhanula, S., Sitti, M. and Fearing, R. Towards flapping wing control for a micromechanical flying insect. In *IEEE International Conference on Robotics and Automation*, 1997.

[19] Zufferey, J.C. Bio-inspired Vision-based Flying Robots. Thèse n° 3194, Faculté Sciences et Techniques de l'Ingénieur, EPF, 2005.