Incremental Evolution of Target-Following Neuro-controllers for Flapping-Wing Animats

Jean-Baptiste Mouret, Stéphane Doncieux, and Jean-Arcady Meyer

Université Pierre et Marie Curie-Paris 6, UMR7606, AnimatLab/LIP6, Paris, F-75015 France;CNRS, UMR7606, Paris, F-75015 France
{jean.baptiste.mouret,stephane.doncieux,jean-arcady.meyer}@lip6.fr

Abstract. Using an incremental multi-objective evolutionary algorithm and the ModNet encoding, we generated working neuro-controllers for target-following behavior in a simulated flapping-wing animat. To this end, we evolved tail controllers that were combined with two closed-loop wing-beat controllers previously generated, and able to secure straight flight at constant altitude and speed. The corresponding results demonstrate that a wing-beat strategy that consists in continuously adapting the twist of the external wing panel leads to better manoeuvring capabilities than another strategy that adapts the beating amplitude. Such differences suggest that further improvements in flying control should better rely on some sort of automatic incremental evolution procedure than on any hand-designed decomposition of the problem.

1 Introduction

Birds continuously demonstrate capabilities which would be of great interest for most Unmanned Aerial Vehicles (UAVs). They are highly agile, able to take off without any runway, to settle on a branch, and to exploit thermals like a sailplane. Consequently, bio-inspired flapping-wing platforms could represent useful trade-offs able to benefit from both the manoeuvrability of helicopters and the energy efficiency of standard airplanes.

However, taking inspiration from birds to design a flapping-wing UAV requires a deep understanding of complex aerodynamic principles that nature learned to exploit in about 150 millions years of evolution. In particular, the numerous degrees of freedom of such UAV must be carefully synchronized to produce adequate thrust and lift forces. Additionally, the corresponding rhythmic movements must be continuously adapted to unpredictable changes in the surrounding air mass. To tackle such issues, that are currently not solved by traditional engineering approaches, the ROBUR project of the AnimatLab [7] aims at designing the morphology and control of a flapping-wing animat through artificial evolution.

In a previous work [15], we used a multi-objective evolutionary algorithm to generate wing-beat neuro-controllers able to secure a straight and horizontal flight at constant speed, even in cases where the flying animat was artificially slowed down. Two efficient control strategies emerged, but with no indication about which one should be used in more challenging conditions.

The goal of the research effort described in this article was to extend this work to the control of target-following behavior. To this end, we used an incremental approach that already proved to be efficient at designing wheeled [19, 2], legged [12] and flying [1] robots. We thus capitalized on the previously evolved wing-beat controllers, and let evolution combine their effect with that of newly generated tail controllers that would force the animat to orient itself towards a targeted direction. We also assumed that the corresponding results would help better assessing the relative advantages of the two wing-beat strategies just mentioned.

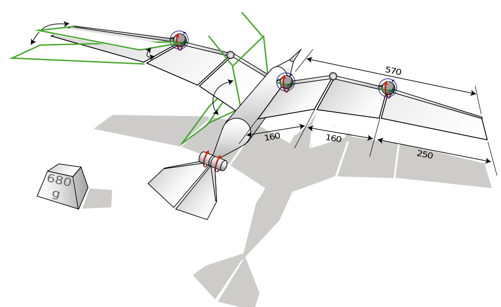


Fig. 1. The simulated bird is modelled using cones, cylinders and rigid panels. A wings internal panel can be moved along the dihedral and the twist axis, while its external panel can be moved along the twist and the sweep axis.

The corresponding experiments called upon a realistic aerodynamic simulator that computes lift and drag forces whatever the local airflow direction. The underlying aerodynamic model has been validated in a wind-tunnel for a fixed-wing UAV.

The simulated bird was made of cones and cylinders which made up its body, and of rigid panels that composed its wings and tail (figure 1). The total wingspan was 124 cm. Additional relevant details are to be found in [15].

This article starts with a summary of the previous results we obtained with wing-beat controllers. The next section describes the evolution of tail neuro-controllers for target-following. Sample trajectories and typical neural networks are then exposed, and the corresponding results are discussed.

2 Wing-beat controllers

Contrary to airplanes which use their wings to sustain themselves, and a propeller to create thrust, birds use their wings to create both an upward and a thrust forces. A wing-beat is made of two distinct phases, the down-stroke and the up-stroke. During the down-stroke, the wings are fully extended and powered downward. The twist is tilted down during this phase, particularly towards the tip. As a consequence, the lift force, created by the pressure difference around the airfoil, is oriented forward and upward. During the up-stroke, the wing is partially folded, to reduce the drag. Additional information on these bird-flight kinematics can be found in [16, 10].

We previously evolved wing-beat neuro-controllers able to exploit such forces and to maintain a flying animat at a constant speed and altitude despite external disturbances. The design of these controllers drew inspiration from the work of biologists on Central Pattern Generators (CPGs) [5, 3, 13] and called upon both non-linear oscillators [3, 13] and standard McCulloch and Pitt's neurons. These controllers could use a speed sensor as input, and four actuators as outputs: the dihedral and the twist of a wing's internal panel, the twist and the sweep of its external panel. The symmetry between wing-beats was forced.

Evolutionary algorithms are the only means that allow to optimize both the topology and the parameters of this kind of neural-networks. The problem to be solved requiring multiple trade-offs – from the energy consumption minimization to the maximization of accelerations – the multi-objective evolutionary algorithm MOGA [8] was used, together with ModNet[6], a modular encoding scheme adapted to the task of evolving neural networks. The corresponding fitness function depended upon six objectives [15], which were evaluated in two stages.

Two different classes of optimal strategies emerged that both relied on the same kinematic principles. According to the first strategy, the twist of a wing's external panel is increased when an acceleration is required, hence increasing the thrust component of the lift force. The wing being folded during the up-stroke, the twist of this panel is not changed. The analysis of the corresponding controllers showed that they implement a simple proportional control of the external twist as a function of the difference between the current speed and the targeted speed.

According to the second wing-beat strategy, all degrees of freedom of a wing exhibit a sinusoidal movement. When an acceleration is required, the amplitude of the oscillations is increased and, consequently, the magnitude of the upward and forward components of the lift force are increased.

Videos of some animats exhibiting these strategies can be downloaded from our website: `http://animatlab.lip6.fr/RoburEvolvingEn`. To the best of our knowledge, this is the first time that closed-loops controllers are obtained for flapping-wing flight.

3 Target following

The detailed kinematics used by birds to change the direction of their flight largely remain an open question, the answer to which probably varies across bird species. It has been suggested that the tail might be used for such use, in a manner similar to the use of elevators in an aircraft [18]. However, some birds succeed to fly without their tail, and mostly rely on wing movements for that. For instance, it has been shown that pigeons use down-stroke velocity asymmetries and rapid alternating wing movements to turn [21, 20].

Likewise, although the control of standard airplanes and UAVs is a widely studied topic, described in many textbooks like [14] for instance, it remains to be proved that classical control methods can be directly used to control an artificial flapping-wing bird because the dynamics of such an engine are complex to model, and because wing-beats generate a lot of parasite movements, especially when orienting.

Additionally, it turns out that radio-controlled ornithopters built by hobbyists exploit their tail to execute simple manoeuvres, as many radio-controlled airplanes do.

For all these reasons, and because the control of a tail can easily be decoupled from the wing-beat control, in a first approach towards implementing useful flying capacities, we chose to evolve tail controllers that could be combined with the wing-beat controllers previously evolved (section 2). More precisely, the objective of these additional controllers was to orient the artificial bird towards a target point, for instance a GPS way-point or a visual landmark, while keeping its altitude constant. To this end, we used exactly the same methodology, calling upon MOGA and ModNet softwares, that the one evocated above and that led to efficient flapping-wing controllers.

To the best of our knowledge, only two papers previously dealt with the generation of kinematics for a flapping-wing artificial bird able to turn [22, 17]. But the goal of these research efforts was to create visually convincing movements and not to design closed-loop controllers. Consequently, the optimization of the kinematics parameters characterizing each trajectory were computationally too greedy to be tested as competitive approaches to the one that has been chosen here.

3.1 Sensors and actuators

In the absence of any a priori knowledge concerning the sensors that birds use to control their flying manoeuvres, we allowed evolution to incorporate, or not, four sensors in the neural controllers it would generate: an altitude sensor - assessing the difference between the current and targeted altitude - a direction sensor - reorienting the animats relative direction to the target - a roll sensor and a pitch sensor.

Furthermore, in the absence of precise specifications of real sensors to be embedded on a real platform, in this preliminary stage we chose to use ideal sensors that would prevent evolution from exploiting specific characteristics of specific devices.

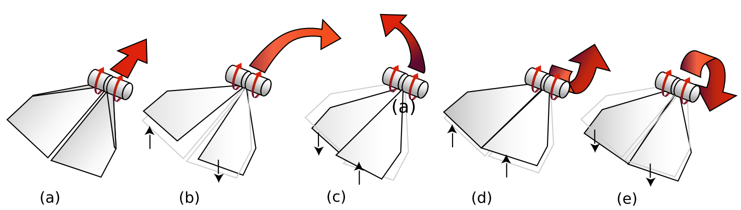


Fig. 2. A v-tail is used to control the artificial bird. (a) When the the panels are in the neutral position, the animat move along a straight line. (b) When the left panel is raised and the right one lowered, the animat turns right. (c) Symetrically, when the left panel is lowered and the right one rised, the animat turns left. (d) When both panels are rised, the animat goes up. (e) Symetricallly, when both panels are lowered, the animat goes down.

Besides the wing characteristics evocated above, an ideal servo-mechanism was used to move each of the two panels that constituted the animats V-shaped tail. To provide an intuitive control of the bird, these two actuators were mixed together in a way similar to that used in radio-controlled sailplanes. Thus, two virtual actuators are provided, one to control the pitch angle and one to act on the yaw and roll angles. Figure 2 displays some typical reactions of the simulated bird to various tail configurations.

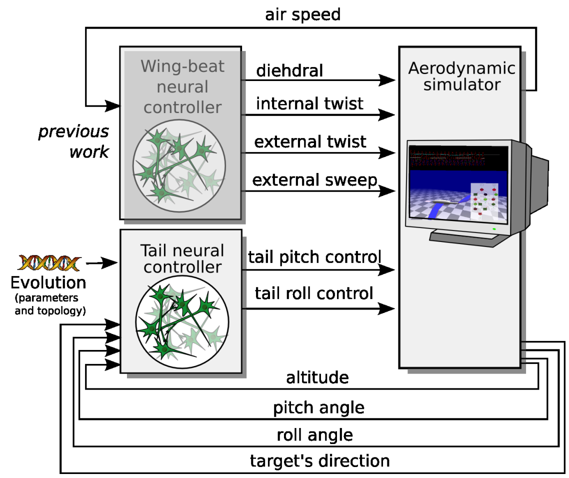


Fig. 3. Overview of the control loop. The wing-beat controller has been evolved in a previous work.

3.2 Fitness

A multi-objective evolutionary algorithm was used to seek controllers able to secure a constant-altitude flight while pointing towards a given target. Figure 3 shows an overview of the corresponding control loop.

Let us denote by N the total length of the evaluation. The bird started its flight at $15m.s^{-1}$ and the position of its center of gravity $g(n)$ was measured at each time-step n. The altitude objective was written as:

$$O_{alt} = -\frac{1}{N}\sum_{n=0}^{n=N} |\mathbf{g}_z(n) - \mathbf{Z}| \tag{1}$$

where $\mathbf{g}_z(n)$ denotes the altitude of the artificial bird at time-step n and Z the desired altitude.

Let first define the vector $\mathbf{v}(n)$ which links the current position $\mathbf{g}(n)$ to the target position $\mathbf{T}$:

$$\mathbf{v}(n) = \mathbf{T} - \mathbf{g}(n) \tag{2}$$

. We then defined the second objective, which rewarded how close the simulated bird was to the desired direction:

$$O_{tar} = -\frac{1}{N}\sum_{n=0}^{n=N} |atan2(\mathbf{v}_y(n), \mathbf{v}_x(n)) - \theta(n)| \tag{3}$$

The $atan2(y,x)$ function calculates the arc tangent of $\frac{y}{x}$ except that the signs of both arguments are used the determine the quadrant of the result.

These two objectives had to be maximized, with an optimal value of 0. They might be evaluated for k different targets. In this case, the results of successive evaluations were summed together.

$$O_{alt} = \frac{1}{k}\sum_{i=0}^{i=k} O_{alt,i} \tag{4}$$

$$O_{tar} = \frac{1}{k}\sum_{i=0}^{i=k} O_{tar,i} \tag{5}$$

3.3 Experimental setup

We used the objectives just defined in conjunction with the MOGA algorithm and the ModNet encoding. Population's size was 350 with 60 % of elites. Six different targets were used, towards which the animat was expected to fly. The total evaluation length was 18000 time-steps, simulating 54 seconds of flight (9 seconds for each target).

Thanks to the use of a so-called model-module pool[6, 15], the neural controllers that were evolved could incorporate and modify across successive generations any number of three different modules:

- a "derivative" module, which computed an approximation of the derivative of its input signal;
- an "integral" module, which computed an approximation of the integral of its input signal;
- a generic module, made of standard McCulloch and Pitt's neurons, with an evolvable structure.

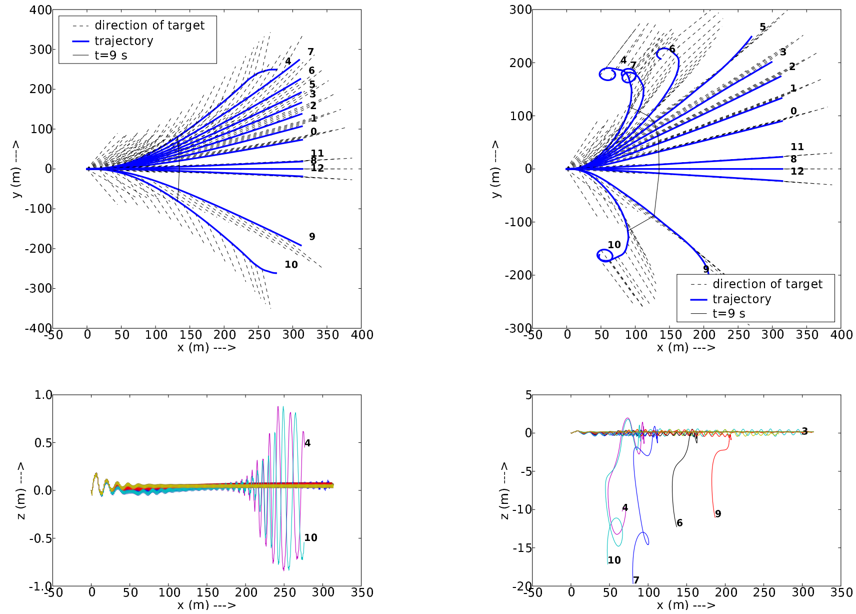
Using two previously evolved controllers, each exhibiting a different wing-beat strategy (section 2), three evolutionary runs of 500 generations were performed to evolve tail controllers likely to complement them. About 12 hours on 20 Pentium at 2*Ghz* were required for each run.

4 Results

4.1 First wing-beat controller

Starting with a wing-beat controller that adapted the twist of the external panel to maintain a targeted flying speed, the individuals that constituted the Pareto front of the last generation segregated in two populations: those that obtained good results on the altitude objective, and those that were efficient with respect to the target objective. The behavior of two randomly-selected individuals representative of each of these populations is shown on Figure 4.

While both individuals succeeded to execute light turns, especially during the evaluation period (9 s), they didn't deal successfully with larger changes of directions beyond



(a) (b)

Fig. 4. (a) Top: Examples of target-following trajectories for an individual using the first type of wing-beat controller. Bottom: This individual obtained good results on the altitude objective. (b) Trajectories of another individual (Top) with good results on the target objective (Bottom).The targets that were used here were different from those that served to evaluate the individuals. For flights not exceeding the corresponding evaluation period (9 sec), the corresponding behaviors were satisfactory. Beyond this period, some trajectories led to stalling, particularly when the targeted orientation angle was high.

the evaluation period. They used a different approach to handle the latter case. The first individual, which had the best fitness for the altitude objective, stopped orienting when the required change in direction was too large. As a consequence, it did not loose altitude, but at the price of not aligning correctly with the target. The second individual, which had a better fitness on the target objective, often started orienting in the right direction, but ended stalling along a spiral trajectory. However, because such events occurred after the evaluation period, this individual was not much penalized with respect to the altitude objective.

We performed a Multiperturbation Shapley value Analysis (MSA) [11] to understand the inner workings of the first individual's tail controller (figure 5). The most useful neurons were those numbered 0, 1, 2, 3, 4 and 5, while the other neurons didn't seem to contribute a lot to the animat's orienting behavior. In particular, neurons 1, 3 and 4 had a large contribution to the first objective, while neurons 0 and 2 mostly contributed to the second objective only. This result indicates that the two objectives were decoupled by the evolutionary process, a conclusion that is confirmed by the close ob-

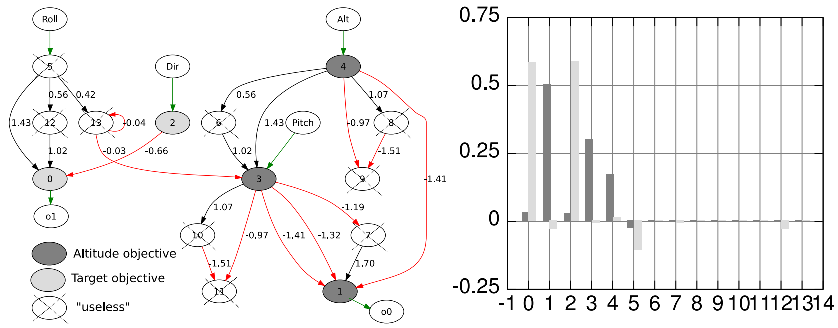


Fig. 5. Left: the neural network that produced the trajectories shown in figure 4 (a). Right: the results of the corresponding MSA. The y-axis gives the numbers identifying the neurons. The x-axis gives the Shapley values. High values indicate important contributions to each of the two considered objectives (altitude and target). Grey levels characterize which neuron is important to the realization of which objective.

servation of the organization of this controller (Figure 5), which appears as almost split in two separate networks. The first one, on the left side of the figure, controls the flight direction using the target's direction sensor. The second one, on the right side, controls the elevator. These two independent controllers are linked by a connexion with a very low weight, which explains the null Shapley value of neuron 11.

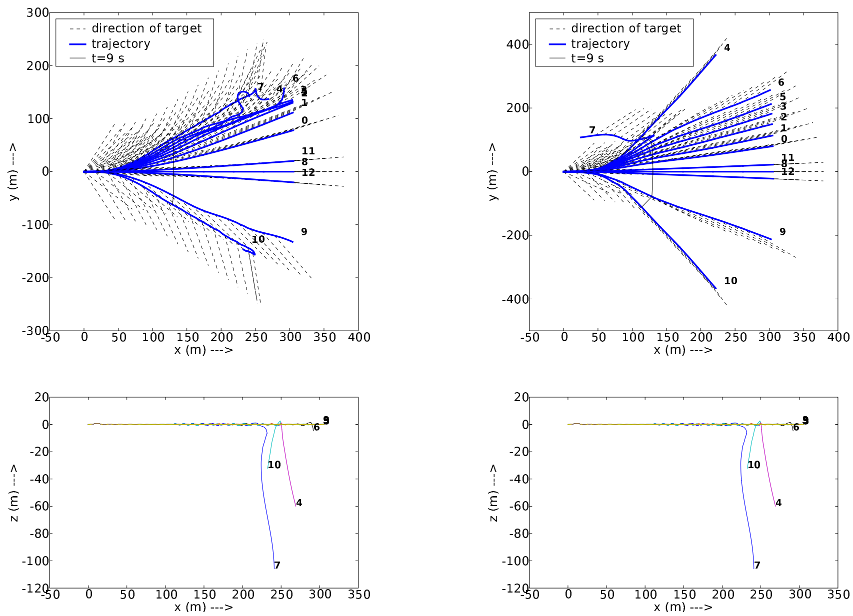
By assigning a null or negative contribution to neurons 5, 12 and 11, the MSA also indicates that the roll sensor is not useful to this controller. Such could be also the case with the pitch sensor, as it will be shown later for an other controller, but the MSA is not conclusive on this point because one cannot decide if the utility of neuron 3 must be attributed to information brought by the altitude sensor, by the pitch sensor, or by both sensors. Be that as it may, we believe that the role of the roll sensor would be much greater if the animat had to fly in an unstable air mass, for instance to secure a constant roll angle despite external perturbations. This hypothesis could be tested by adding perturbations during the evaluation procedure, as we did in [15].

4.2 Second wing-beat controller

Figure 6 shows the behavior of two individuals populating the Pareto front of the last generation, starting with a wing-beat controller that adapted the amplitude of the oscillations to maintain a targeted flying speed. These individuals were randomly-selected among those that respectively obtained good results on the altitude objective, and good results on the target objective.

The first individual exhibits a behavior similar to the one displayed on figure 4 because it stopped orienting when the required changes of direction were large. However, it stalled after having flied for about 200m. Moreover, the error between the animat's direction and the targeted one was larger on figure 6 (a) than on figure 4 (a).

The second individual got the best fitness evaluation according to the angle objective. Surprisingly, although the animat succeeded to perform the largest turns (labeled



(a) (b)

Fig. 6. (a) Top: Examples of target-following trajectories for an individual using the second type of wing-beat controller. Bottom: This individual obtained good results on the altitude objective. (b) Trajectories of another individual (Top) with good results on the target objective (Bottom). Both individuals performed correctly during the evaluation period (9*s*).

9 and 10), it couldnt avoid climbing up. This was due to the fact that the wing-beat controller reached a saturated state, according to which the wings were steadily flapped with a maximum strength, thus producing a maximum thrust. By flying faster, the bird generated more lift and, accordingly, went up. This saturated state was not reached for every target, as demonstrated by the stalling trajectory 7.

The analysis of the corresponding neural controller (figure 7) indicates that it is also split in two sub-networks, one controlling the direction and the other controlling the altitude. For this controller, both the roll and the pitch sensors seem useless. Again, these two sensors would probably be required for flights in an unstable air mass.

4.3 Objective space

Figure 8 displays the two objective scores attained by each animat generated during the three evolutionary runs, for the two wing-beat strategies.

It thus turns out that individuals exploiting the first strategy are distributed in most of the objective space, with a higher concentration close to the Pareto front. The shape of this front expresses the necessity of a trade-off, since no individual gets an optimal score on both objectives. The evolutionary process generated a lot of such trade-off solutions situated near the optimum, i.e., at coordinates 0,0.

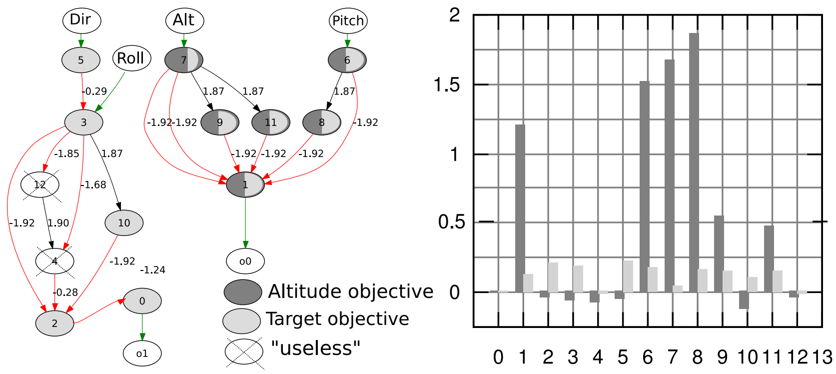
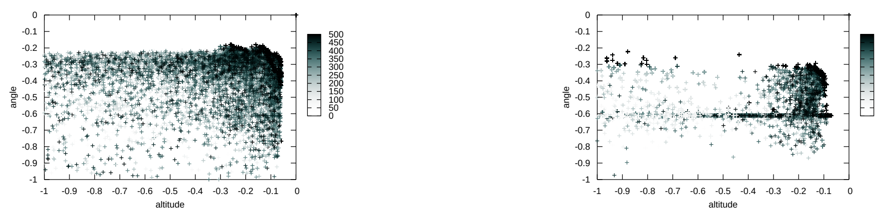


Fig. 7. Left: The neural network that produced the trajectories shown in figure 6 (a). Right: Its MSA results.



(a) (b)

Fig. 8. . Exploration of the objective space by all individuals exploiting the first (a) or the second (b) wing-beat strategy. Each dot corresponds to an individual. Grey levels denote the last generation for which this individual was present in the population.

The exploration of the objective space by individuals exploiting the second strategy was quite different. First, the best scores thus attained are substantially lower than in the previous case, especially on the target objective. Second, the explored solutions cover a much limited range of possible scores.

The horizontal line with a target objective value of -0.6 corresponds to individuals that did not turn at all. These individuals quickly disappeared when we used the first wing-beat controller, but they were still present after 500 generations when the second controller was used.

All these results suggest that optimizing a target-following behavior is more difficult with the second controller than with the first.

5 Discussion

The above results demonstrate that, once neural controllers for straight-line flight have been evolved, it is possible to capitalize on the corresponding networks to evolve additional tail controllers that exhibit minimal target-following capacities. However, as mentioned above, better results would probably be obtained by relaxing the symmetry constraint that was imposed here on wing movements, and by exploiting the manoeuvrability capacities thus offered. Such approach might entail the joint evolution of wing-beat and tail control, a task that seems highly challenging according to current technology. To raise its chances of success, the recourse to some sort of automatic incremental methodology seems mandatory. Indeed, it has been shown here that one cannot rely on fundamental principles or empirical knowledge to decide which, among two available wing-beat controllers, would be better suited to pave the way to additional flying manoeuvrability. Nevertheless, their aptitudes for this endeavour ultimately turned out to be quite different.

Two main reasons motive the use of incremental evolutionary approaches to evolutionary robotics:

– The bootstrap problem. In many real-life situations, an intermediate action is required – e.g. pushing a button to switch-on a light – before receiving any reward – e.g. going to the light to get food. By decomposing the problem into sub-tasks, one may guide the evolutionary process towards satisfying solutions that would otherwise be hard or impossible to discover.
– The search space problem. By imposing intermediary stages, one reduces the size of the search space, hence speeding-up the evolutionary process.

Some promising work has been carried out to automatize such a procedure using cooperative co-evolution [9, 4]. We plan to assess the applicability of similar approaches to flying behavior in the near future.

6 Conclusion

Although evolving flying robots seems a greater challenge than evolving crawling, walking, or swimming artefacts, we have shown that a suitable evolutionary algorithm, combined with an efficient coding and a two-stage approach, made it possible to generate close-loop controllers for a target-following flying behavior. However our results suggest that future improvements should better rely on some sort of automatic incremental evolution procedure than on any hand-designed decomposition of the considered problem.

Acknowledgements This work was supported by a UPMC BQR grant.

References

1. G. J. Barlow, C. K. Oh, and E. Grant. Incremental evolution of autonomous controllers for unmanned aerial vehicles using multi-objective genetic programming. In *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems (CIS)*, pages 688–693, Singapore, 1-3 December 2004. IEEE.
2. J. Chavas, C. Corne, P. Horvai, J. Kodjabachian, and J.-A. Meyer. Incremental evolution of neural controllers for robust obstacle-avoidance in Khepera. In P. Husbands and J.-A. Meyer, editors, *Proceedings of The First European Workshop on Evolutionary Robotics - EvoRobot98*, pages 227–247. Springer-Verlag, 1998.

3. A. H. Cohen, P. J. Holmes, and R. H. Rand. The nature of the coupling between segmental oscillators of the lamprey spinal generator for locomotion : A mathematicaal model. *Journal of Mathematical Biology*, 13:345–369, 1982.
4. Edwin D. de Jong and Jordan B. Pollack. Ideal evaluation from coevolution. *Evolutionary Computation*, 12(2), 2004.
5. F. Delcomyn. Neural basis for rhythmic behaviour in animals. *Science*, 210:492–498, 1980.
6. S. Doncieux and J.-A. Meyer. Evolving neural networks for the control of a lenticular blimp. In G. R. Raidl, S. Cagnoni, J. J. Romero Cardalda, D. W. Corne, J. Gottlieb, A. Guillot, E. Hart, C. G. Johnson, E. Marchiori, J.-A. Meyer, and M. Middendorf, editors, *Applications of Evolutionary Computing, EvoWorkshops2003: EvoBIO, EvoCOP, EvoIASP, EvoMUSART, EvoROB, EvoSTIM*. Springer Verlag, 2003.
7. S. Doncieux, J.-B. Mouret, L. Muratet, and J.-A. Meyer. The ROBUR project: towards an autonomous flapping-wing animat. In *Proceedings of the Journées MicroDrones*, Toulouse, 2004.
8. C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization : formulation, discussion and generalization. In *Proceedings of the Fourth International Conference on Evolutionary Programming*, pages 416–423, 1993.
9. G. L. Haith, S. P. Colombano, J. D. Lohn, and D. Stassinopoulos. Coevolution for problem simplification. *Proceedings of the Genetic and Evolutionary Computation Conference*, 1999.
10. T.L. Hedrick, B.W. Tobalske, and A.A. Biewener. Estimates of circulation and gait change based on a three-dimensional kinematic analysis of flight in cockatiels (nymphicus hollandicus) and ringed turtle doves (streptopelia risoria). *Journal of Experimental Biology*, 205:1389–1409, 2002.
11. A. Keinan, B. Sandbank, C. C. Hilgetag, I. Meilijson, and E. Ruppin. Fair attribution of functional contribution in artificial and biological networks. *Neural Computuation*, 16(9):1887–1915, 2004.
12. J. Kodjabachian and J.-A. Meyer. Evolution and development of neural networks controlling locomotion, gradient-following, and obstacle-avoidance in artificial insects. *IEEE Transactions on Neural Networks*, 9:796–812, 1997.
13. K. Matsuoka. Mechanisms of frequency and pattern control in the neural rhythm generators. *Biological Cybernetics*, 56:345–353, 1987.
14. D. McLean. *Automatic Flight Control Systems*. Prentice Hall, New York, 1990.
15. J.-B. Mouret, S. Doncieux, T. Druot, and J.-A. Meyer. Evolution of closed-loop neurocontrollers for flapping-wing animats. 2006. submitted for publication.
16. U. M. Norberg. *Vertebrate Flight*. Springer-Verlag, 1990.
17. Y. S. Shim, S. J. Kim, , and C. H. Kim. Evolving flying creatures with path following behaviors. In *The 9th International Conference on the Simulation and Synthesis of Living Systems (ALIFE IX)*, pages 125–132, 2004.
18. A. L. R. Thomas. On the tails of birds. *BioScience*, 47(4):215–225, April 1997.
19. J. Urzelai, D. Floreano, M. Dorigo, and M. Colombetti. Incremental robot shaping. *Connection Science Journal*, 10(384):341–360, 1998.
20. D. R. Warrick, M. W. Bundle, and K. P. Dial. Bird maneuvering flight: Blurred bodies, clear heads1. *INTEG. AND COMP. BIOL.*, 2002.
21. D. R. Warrick and K. P. Dial. Kinematic, aerodynamic and anatomical mechanisms in the slow, maneuvering flight of pigeons. *The Journal of Experimental Biology*, 201:655ˇ2013672, 1998.
22. J. Wu and Z. Popoviè. Realistic modeling of bird flight animations. *ACM Trans. Graph.*, 22(3):888–895, 2003.