



HAL
open science

Soaring behaviors in uavs: 'animat' design methodology and current results

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

► To cite this version:

Stéphane Doncieux, Jean-Baptiste Mouret, Jean- Arcady Meyer. Soaring behaviors in uavs: 'animat' design methodology and current results. 7th European Micro Air Vehicle Conference (MAV07), 2007, Toulouse, Unknown Region. pp.1-10. hal-02987435

HAL Id: hal-02987435

<https://hal.science/hal-02987435>

Submitted on 3 Nov 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Soaring behaviors in UAVs : 'animat' design methodology and current results

Stéphane Doncieux* and Jean-Baptiste Mouret† and Jean-Arcady Meyer‡
ISIR, Institut des Systèmes Intelligents et Robotique, 75016 Paris, France

Saving energy is a critical issue for mini and micro-UAVs. We used tools rooted in the 'animat' approach to generate energy saving behaviors for a glider robot. The connection weights of feed-forward neural networks were optimized by evolutionary algorithms to exhibit "soaring" behaviors, i.e. behaviors that capitalize on aerological conditions to extract energy from the environment, focusing on thermal and slope wind exploitation. Thermal soaring with spiral trajectories and slope soaring with eight-shaped trajectories were thus exhibited. The optimization criterion used for thermal soaring was the average altitude gain. For slope soaring, an additional criterion forced the glider to remain in a limited area. These criteria were high-level specifications of the desired behaviors and did not include any direct description of the strategy needed to get them.

I. Introduction

As the technology of battery manufacturing improves, substantial progress in the autonomy of small UAVs (wingspan of about two meters) may be expected in the near future. However, it seems improbable that flights largely exceeding one hour, like those performed by numerous animals, will be soon possible. The reason does not only stem from technological limitations tied to energy storage or effector efficiency, it is also related to the highly adapted energy-saving behaviors exhibited by these animals. Indeed, such behaviors aim at minimizing energy consumption and at extracting energy from the environment whenever possible. Birds, for instance, constantly exploit air masses movements to this end. Thanks to such dedicated soaring behaviors, albatrosses, in particular, are able to remain airborne for days without even flapping their wings.^{1,2} Our research efforts mainly focus on behaviors that allow exploiting three different sources of air movements, namely wind gradients, thermals and slope winds. The vertical gradient of an horizontal wind makes it possible to extract energy when following a particular cyclic trajectory:³ starting at a high altitude, wind in the back and diving, permits gaining speed; at low altitude, a sharp turn allows to face the wind and to exploit the speed thus gained to reach the starting altitude again. Repeating this cycle allows to travel in a given direction relative to the wind without losing any energy as evidenced by albatrosses. Thermals are simpler to exploit because they create a vertical air movement within which an aircraft just needs to remain to gain altitude, generally exhibiting a spiral trajectory centered on the thermal like those exhibited by birds of prey⁴ or glider pilots.⁵ Slope soaring is possible when the wind encounters a mountain, or at least a slope, and lifts the aircraft. To avoid being crashed into the nearby obstacle, instead of a spiral trajectory, an "eight-shaped" is better suited because it allows to fly with the slope safely maintained behind the aircraft. This article describes results obtained on the design of controllers for autonomous UAVs able to exploit such behaviors, as parts of the ROBUR project⁶ that aims at building a flapping-wing robot endowed with the decisional autonomy required for fulfilling given missions without the help of a human operator. The corresponding aircraft should have a wingspan of approximately one meter and should be able to remain airborne for long periods of time. To this end, we design controllers that produce behaviors similar to soaring behaviors in real birds. However, if qualitative information on how to exploit a particular environmental condition is long-known, this knowledge is hardly compatible with classical control approaches. Although human-designed controllers have been able to exhibit thermal soaring behavior in specific conditions,⁷ we propose instead a generic methodology for the design of such controllers that relies on high-level behavior description only, in fact on what the system should do rather than on how to do it. Such approach explores

*Prof. assistant, dept SIMA, Stephane.Doncieux@upmc.fr

†PhD student, dept SIMA, Jean-Baptiste.Mouret@lip6.fr

‡Emeritus Research Director, dept SIMA, Jean-Arcady.Meyer@lip6.fr

the available search space to answer the how to do it question. It is rooted in the animat approach,⁸ whose goal is to build autonomous artificial artifacts and to develop adaptive algorithms that allow them to learn by their own how to efficiently interact with their environment. In a similar perspective, Wharington already proposed thermal soaring controllers relying on an adaptive scheme. In an experiment described in its PhD,⁹ reinforcement learning was used to train a neural-based thermal locator. However, although efficient, its system required too much computational power to be run in real time.

The proposed methodology mainly relies on the use of evolutionary algorithms that automatically design general purpose controllers like neural networks or fuzzy rule-based systems. After completion of the evolutionary process, these controllers aren't changed anymore and require a computational power low enough to be run on-board and in real time using current technology. The working principle of evolutionary algorithms consists in exploring potential solutions through a trial-and-error process associated with selective pressures and dedicated search operators. Given a genotype, i.e. a formal description of a solution – that may be a mere vector of parameters or the description of a whole control architecture – and associated search operators, the algorithm explores the search space and converges towards solutions maximizing fitness criteria provided by the user. These criteria correspond to the description of what can be expected from a good solution and are expected to guide evolution towards solutions that are of interest for the user. To generate soaring behaviors, for instance, a convenient fitness criterion may be as simple as the total flight time because controllers maximizing such criterion will necessarily call upon dedicated behaviors extracting energy from the environment. It is also worth mentioning that evolutionary algorithms manipulate a population of individuals in parallel. This is of interest especially when dealing with several antagonistic criteria. In our case, for instance, we may wish to maximize the flight time while simultaneously maximizing the exploration of the flight area : exploring is then antagonistic to exploiting a particular potential energy source once discovered. In such circumstances, dedicated evolutionary algorithms generate at once an approximation of the Pareto front, i.e. the set of the best compromises. The end-user can then analyze the different solutions and choose the one that best suits his needs.

Recent work at ISIR concerned dynamic soaring control¹⁰ - i.e., the exploitation of wind gradients with evolutionary algorithms and controllers based on fuzzy-rules systems. Starting with an inefficient hand-designed controller, artificial evolution found controllers able to reach in simulation a performance similar to that of an albatross with an endless flight in a wind gradient whose reference speed was 10m.s-1. The tests of robustness that have been carried on showed that the behaviors were robust to initial conditions, but also that they required very accurate sensors and were sensitive to gradient features. We furthered this work to generate controllers for slope and thermal soaring behaviors.

After a presentation of the methodology, this paper describes our experiments and their results in simulation, as well as the prototype that will be used to test the resulting controllers in real conditions. Perspectives for future UAVs are also discussed.

II. Methodology

II.A. Model

II.A.1. Glider model

We used a semi-empirical, quasi steady-aerodynamics model in which our glider's wings and tail are made of independent panels. At each time step (0.02 sec in this work), the local incident airspeed is evaluated for each panel and three aerodynamic forces are evaluated: the leading edge lift, the parachute drag and the friction drag. All forces are summed to determine the movement of the whole glider. Further details on this model can be found in.¹¹

We have tuned the model to fit the features of the real platform that will be used to test the generated behaviors in future experiments (see section III.C). Each wing is made with an internal and an external panel and the tail has a "T" shape built with two other panels. The internal and the external panels of the wing have a dihedral angle of 12°.

II.A.2. Thermal model

The thermal model we used is described in.¹² It represents a thermal as a kind of paraboloid of revolution in which updraft velocity increases towards the center (figure 1-a). Thermal features are described by w^* , the convective velocity scale, and z_i , the convective mixing layer thickness. We modeled only one thermal here, whose parameters were: $w^* = 4.5$ and $z_i = 1401$.

At a given altitude, updraft velocities vary with a bell-shape distribution to reach a maximum value at the center of the thermal.

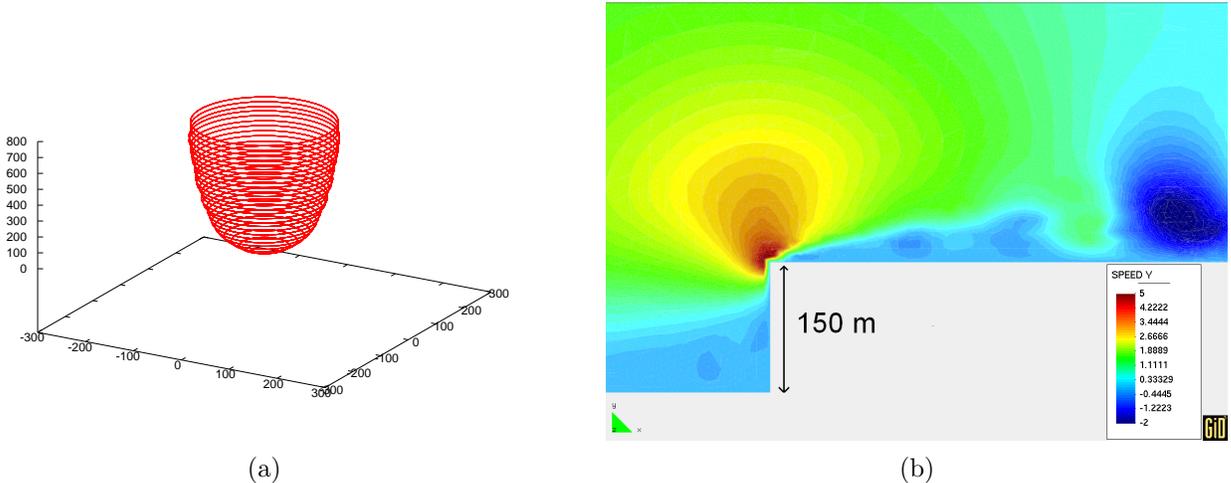


Figure 1. Aerological conditions used in the work. (a) Thermal shape. Circles indicate the limit of thermal influence for a given altitude. Inside a circle, the influence on the vertical speed increases from 0 on the periphery to its maximal value at the center of the circle. X,Y and Z axis are 3D-coordinate in meters. (b) Visualisation of the wind field (vertical component) near a 150m-high cliff. The wind blows from the left with a uniform speed of $11m.s^{-1}$.

II.A.3. Slope wind model

Modeling the influence of a slope on the wind is not a trivial issue. We chose to model it as a vector field covering the whole flight area. Each vector represents the speed and the direction of the wind at a given intersection point in a mesh. We used the ADFC Navier-Stokes solver of Madrid University that has been validated on Askervein hill in Scotland.¹³ We modeled a 150m-high cliff and wind features (speed and Reynolds number) using Gid, a software developed by the International Center for Numerical Methods in Engineering at Barcelona.¹⁴ The wind is supposed to be uniform before the cliff, with an horizontal speed of $11m.s^{-1}$ directed towards the cliff (figure 1-b). The cliff is supposed to be infinite along its edge and the wind features to be uniform along this axis.

II.B. Evolutionary algorithm

Evolutionary algorithms are inspired from Darwinian theory on the selection of natural species, with the noticeable difference that we explicit the criteria to be maximized and define our own genetic representation according to our needs.

Using an evolutionary algorithm requires to define the following three independent parts:

- selection algorithm: this algorithm describes how to apply the selective pressure that drives it towards efficient solutions (as evaluated by the fitness and evaluation procedure);
- genotype: the genotype corresponds to the data structure that the evolutionary algorithm manipulates. This can be anything, provided we can efficiently define the three required variation operators: random generation, mutation and crossover^a. The genotype describes a phenotype, i.e. the system we try to

^aThe random generation operators creates the initial population. The mutation applies a random modification to a genotype and crossover exchanges information between two (or more) different genotypes

design, such as a controller or a morphology in the case of robotics experiments, for instance;

- fitness and evaluation procedure: a value (or a vector of values in the case of multi-criteria evolutionary algorithms) must be associated to a given genotype. It will be used by the selection algorithm to rank individuals and determine which to keep and which to drop.

II.B.1. Fitness and evaluation procedure

Our goal is to generate controllers able to exploit miscellaneous aerological conditions. Instead of focusing on a single controller likely to generate the appropriate behavior in every possible condition, we chose to make the task easier and to design a controller adapted to each specific condition. A high-level controller selecting the right action to perform and where to go, depending on the mission and on internal motivations (“need to recharge batteries” for instance), will need to be built in the future for the glider to be truly autonomous. To this end, we will capitalize on the Psikharpax project¹⁵ which tackles this issue while designing an artificial rodent.

The controllers we consider here are dedicated to a particular aerological phenomenon, i.e. a thermal or a favorable slope wind, and are supposed to be automatically triggered in its neighborhood. They are not concerned with environmental exploration in general, or with the search of a suitable aerological phenomenon in particular.

THERMAL SOARING. The goal of these experiments is to evolve a controller that allows the glider to gain altitude thanks to a nearby thermal. The fitness function we used is then simply given by:

$$f_{alt,i}(ind) = \frac{\sum_{t=0}^T (z(t) - z_{start})}{T_{total}}$$

where $f_{alt,i}(ind)$ is the altitude criterion value of individual ind for evaluation i , T is the occurrence time of a crash (as detected by the condition $z < 0$ or in case of a numerical error in the simulation^b) and T_{total} is the length of the allotted evaluation period if no crash occurred ($T_{total} = 4000$ in these experiments). $z(t)$ is the current altitude and z_{start} is the initial altitude.

The fitness f of an individual is then:

$$f(ind) = f_{alt}(ind) = \frac{\sum_{i=0}^N f_{alt,i}(x)}{N}$$

where $f(ind)$ is the fitness value associated to individual ind and N is the number of evaluations.

To warrant a robust behavior not adapted to a single experiment, individuals were evaluated in a set of seven runs which differed by the corresponding initial positions relative to the thermal. These initial positions were respectively set at the center of the thermal, at four different locations inside the thermal and at two positions outside the thermal (with the glider facing it).

SLOPE SOARING The edge of our simulated slope being infinite, the optimal trajectory consists in a straight line with a carefully chosen orientation (“crab”-like trajectory). This behavior was easily generated by evolution when using the altitude criterion only. However, because an infinite edge is not realistic, and to avoid changing our slope model, we added a second criterion to the altitude criterion described above. This criterion measured the deviation from an horizontal virtual axis perpendicular to the edge of the slope, whose position has been arbitrarily chosen. This axis could characterize an area the glider was supposed to remain into, either for its own safety (place where the wind was the most favorable) or for mission purposes (observation for instance). The following criterion has thus been defined:

$$f_{dist,i}(ind) = \frac{\sum_{t=0}^T (y(t) - y_{target})}{T_{total}}$$

$$f_{dist}(ind) = \frac{\sum_{i=0}^N f_{dist,i}(x)}{N}$$

^bSuch problem occasionally occurred in our simulation at high speeds, for instance. As the corresponding behavior was unrealistic, we added a penalty $-f_i(x) = -100000$ to be sure that corresponding individuals would not reproduce.

The fitness f of an individual combined the two criteria:

$$f(ind) = \{f_{alt}(ind), f_{dist}(ind)\}$$

II.B.2. Genotype & phenotype

Phenotypes were multi-layer perceptrons that controlled the glider (fig 2). Connection weights were evolved to optimize the fitness criteria just defined within the context of evaluation procedures specific to each experiment.

Each such network had four inputs:

- v_z : vertical speed,

$$v_z = \begin{cases} v_{z,real}/5 & \text{if } |v_{z,real}| \leq 5m.s^{-1} \\ 1 & \text{if } v_{z,real} > 5m.s^{-1} \\ -1 & \text{if } v_{z,real} < -5m.s^{-1} \end{cases}$$

- $roll$: roll angle, $roll = \frac{(roll_{real} + noise)}{\pi}$, where noise is a random value less than $0.2rad$
- $pitch$: pitch angle $pitch = \frac{(pitch_{real} + noise)}{\frac{\pi}{2}}$, where noise is a random value less than $0.2rad$
- for thermal soaring experiments, the fourth input is constant and always equal to 1. For slope soaring experiments, this input was the distance to the reference axis. This value was supposed to be obtained from some GPS device or from a visual sensor.

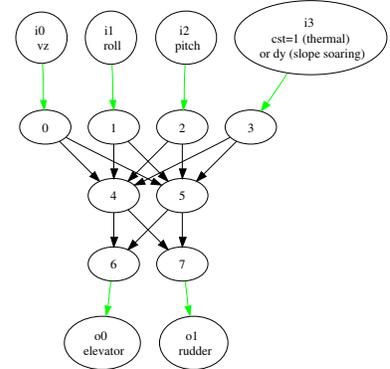


Figure 2. Neural network structure: a multi-layer perceptron with two hidden neurons.

The two outputs of each controller respectively controlled the rudder and elevator servomotors. To get neuron outputs varying between -1 and 1 , we applied the following adjustments:

$$\begin{aligned} elevator_{cmd} &= elevator_{neuronoutput} * 0.14 \text{ rad} \\ rudder_{cmd} &= rudder_{neuronoutput} * 0.70 \text{ rad} \end{aligned}$$

Finally, connection weights were forced to belong to the interval $[-2; 2]$ and were managed as an evolvable vector of real values, as done with Evolution Strategies.¹⁶

II.B.3. Selection algorithm

The selection algorithm we used is ϵ -MOEA,¹⁷ An efficient multi-objective evolutionary algorithm. Thermal soaring experiments could have been run with a more classical single-objective algorithm, but preferred to use ϵ -MOEA to minimize differences between the two experimental settings.

All experiments have been implemented on the SFERES¹⁸ software, which is a framework dedicated to the use of evolutionary algorithms on robotic simulations.

III. Results

III.A. Thermal soaring

As thermal soaring experiments called upon a single fitness criterion, the evolutionary runs generated a single solution only. After 200 generations, a fitness of 89 was reached, meaning that the average altitude of the glider during the evaluations is 89m above the starting altitude. The neural network with its evolved weights is represented on figure 3.

To check the robustness of the best evolved controller, we evaluated its performances in eleven initial conditions. The controller succeeded in exploiting the thermal in every case (fig 4). The trajectory is a sharp spiral that progressively shifts towards the center of the thermal.

To further evaluate the robustness of our controller, we tested it on a wide set of initial conditions: starting always at the same altitude (300m), the glider behavior was observed on a grid of x,y positions ranging between -140m and 140m with a step of 10m. At this altitude, the thermal radius is 111m. We counted as successes experiments in which the highest altitude reached was above 350m. On the 841 tests, 654 (77.8%) were successful (figure 4-d); considering individuals starting from inside the thermal only, the success rate increased up to 100%. The same test set on a thermal with a convective velocity scale $w^* = 2.5$ resulted in 73.1% successful tests (96.6% with starting positions inside the thermal).

Unsuccessful trajectories followed a spiral with a wider radius than those inside the thermal. This behavior enhances the probability of returning to the thermal, as the glider will cover a wider area, but it is still sub-optimal, as only an Archimedes spiral - i.e. a spiral with an increasing radius, as used in⁷ - would allow to return to the thermal. Generating such a trajectory would require using a kind of memory allowing to increase a value representing the turn radius. This is not possible with the control structure that was used here as it included no recurrent connections. Further experiments should rely on fixed recurrent structures or on algorithms allowing their automatic design as used in.^{19,20}

III.B. Slope soaring

Slope soaring experiments relied on two different criteria to be optimized. Therefore, as expected, evolutionary algorithms haven't generated a single solution, but a set of compromises between these two criteria (figure 5).

The average gained altitude ranged from 13m to 67m while the mean position along the edge of the slope ranged from 4m to 80m away from the reference axis. We focused our study on individuals maximizing the altitude criterion, which seemed, at this stage, more interesting to us.

Among the nine generated individuals, we selected individuals 0 and 2 of figure 5^c. The corresponding neural networks are shown in figure 6 and the trajectories they generate are represented in on figure 7. In both cases, the glider always faces the wind and oscillates around a position that is not the position of the axis we have defined. Depending on the speed of the glider relative to slope wind, this generates a trajectory with oscillations that sometimes looks like the classical eight-shaped trajectory.

It is highly probable that all individuals that turned back to the wind, were quickly eliminated by evolution, as the glider isn't fast enough to compensate for the back-wind shift while facing a front wind.

^cThe individual with the highest altitude criteria turned out to be less efficient than individuals 0 and 2 on evaluation periods longer than the one used during the evolutionary run.

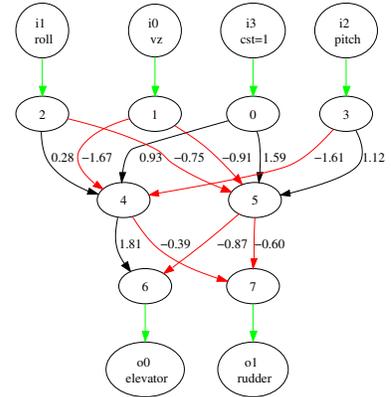


Figure 3. Neural network with the connection weights evolved for thermal soaring.

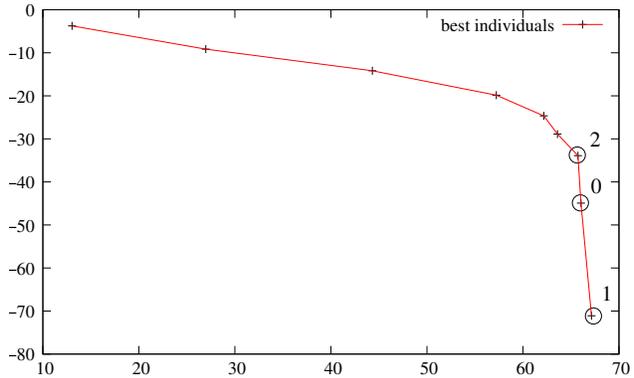


Figure 5. Best individuals generated by the evolutionary algorithm after 200 generations. The X-axis corresponds to the altitude criteria (in meters) and the Y-axis corresponds to the distance criteria (in meters). Circles correspond to individuals whose behavior is plotted on figure 7.

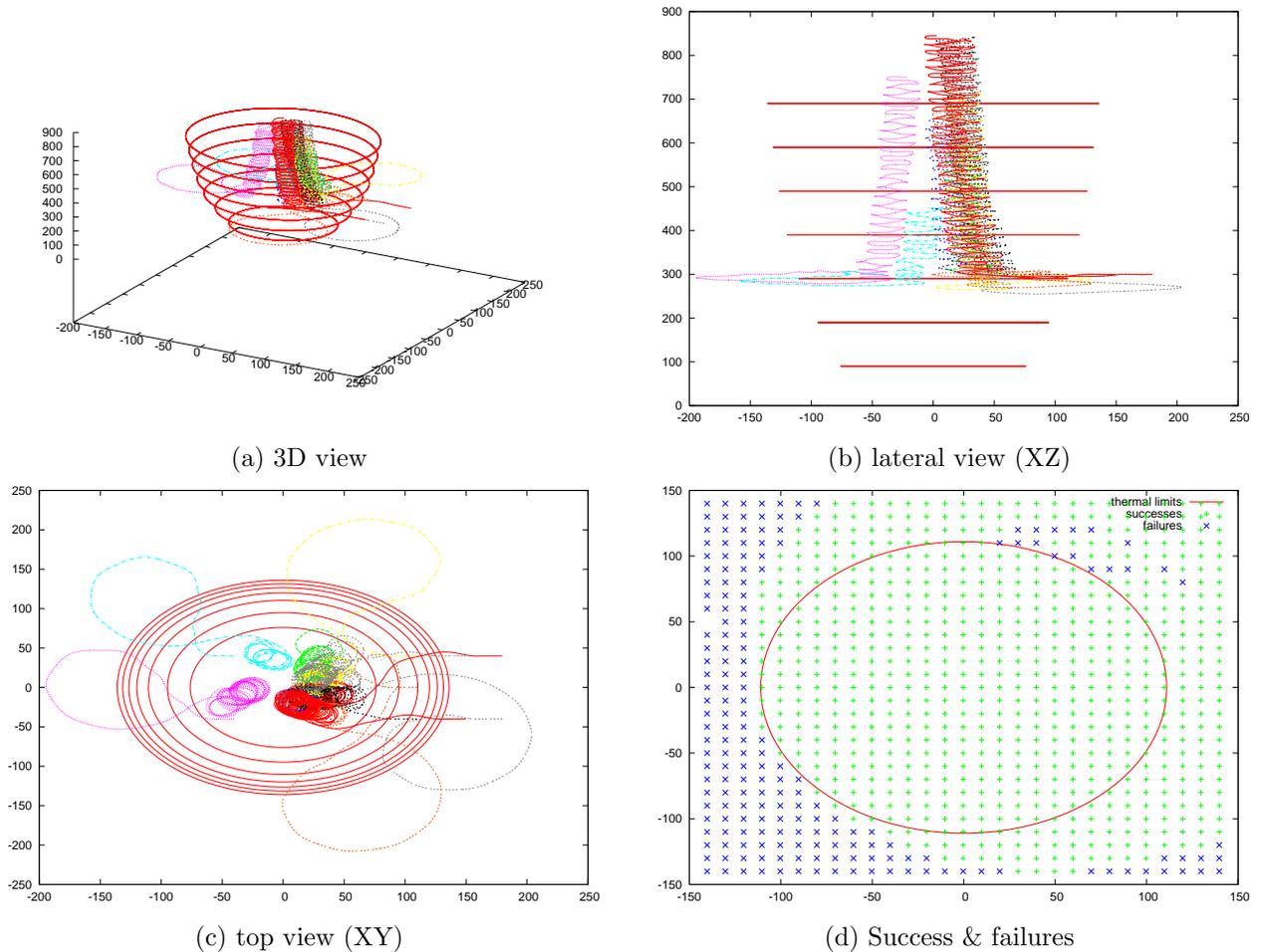


Figure 4. (a), (b), (c) Generated trajectories for eleven different initial conditions by a neural network evolved for thermal soaring. Thermal limits are shown in red. (d) Successes and failures for the corresponding initial positions. Each dot represents an initial position (the altitude is always 300m). In every experiment, the initial speed is oriented towards the left with an amplitude of $10m.s^{-1}$. Coordinates are in meters.

Starting from above the slope edge, individual 0 tends to move towards the front of the slope, while individual 2 tends to move towards the back of the slope. In either cases, these characteristics are probably very fragile and adapted to this particular slope and this particular wind speed. The system having no sensory input giving its position relative to the slope, it is not surprising that we observe a shift towards the front or the back of the slope. The focus here was to check if v_x provided enough information for the glider to remain in the most interesting region with respect to the slope. It seems that our evaluation length was not long enough to put a sufficient pressure on that point. Indeed, even with an evaluation that is four times longer than during evolution, the glider doesn't leave the updraft, nor does it clearly remain at its center (figure 7). Further experiment should consider a smaller updraft or a longer evaluation time.

III.C. Prototype

A motor-glider is currently built to test the evolved behavior in real conditions (figure 8). It is based on a "chimera"^d motor glider with the following features:

- wingspan: 1.92m
- fuselage length: 1.12m

^d<http://www.topmodel.fr>

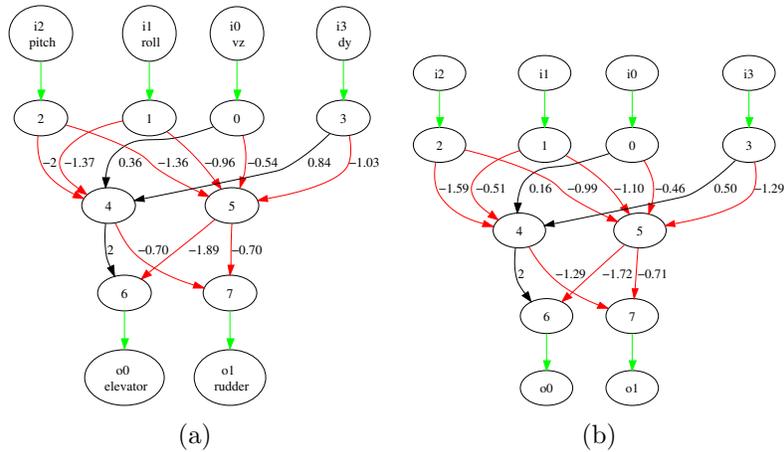


Figure 6. (a) Neural controller of individual 0. (b) Neural controller of individual 2.

- weight: 1.3kg (off-the-shelf version)
- wing area: 43.8dm²
- wing profile: S3021

We equipped it with a Micronav IMU and a Stargate processing board from Crossbow. Preliminary flights have been done, but low-level control remains to be completed before soaring experiments can be tested with this prototype.

IV. Discussion



Figure 8. Motor-glider under development to test thermal and slope soaring in real conditions.

These results obtained so far are very encouraging. Nevertheless, the thermal model we used represented only one particular kind of thermal and thermals are known to be very different one another.²¹ Likewise, the slope soaring behavior should be tested on different slope winds to confirm its efficiency. Updraft “centering”, or at least the ability to remain in the updraft for slope soaring should be further explored as the controllers described here do not seem to solve the problem. In the perspective of an integrated system, our slope soaring controller can nevertheless be used for temporary altitude gain. The practicability of our “dy” sensor should also be assessed and the sensibility of soaring behaviors to coarse measurements should also be checked. In any case, experiments on a real glider are necessary to validate the results.

V. Conclusion

We presented a methodology rooted in the animat approach for automatic UAV behavior design. A high-level description of the desired behavior is provided to an evolutionary algorithm that opti-

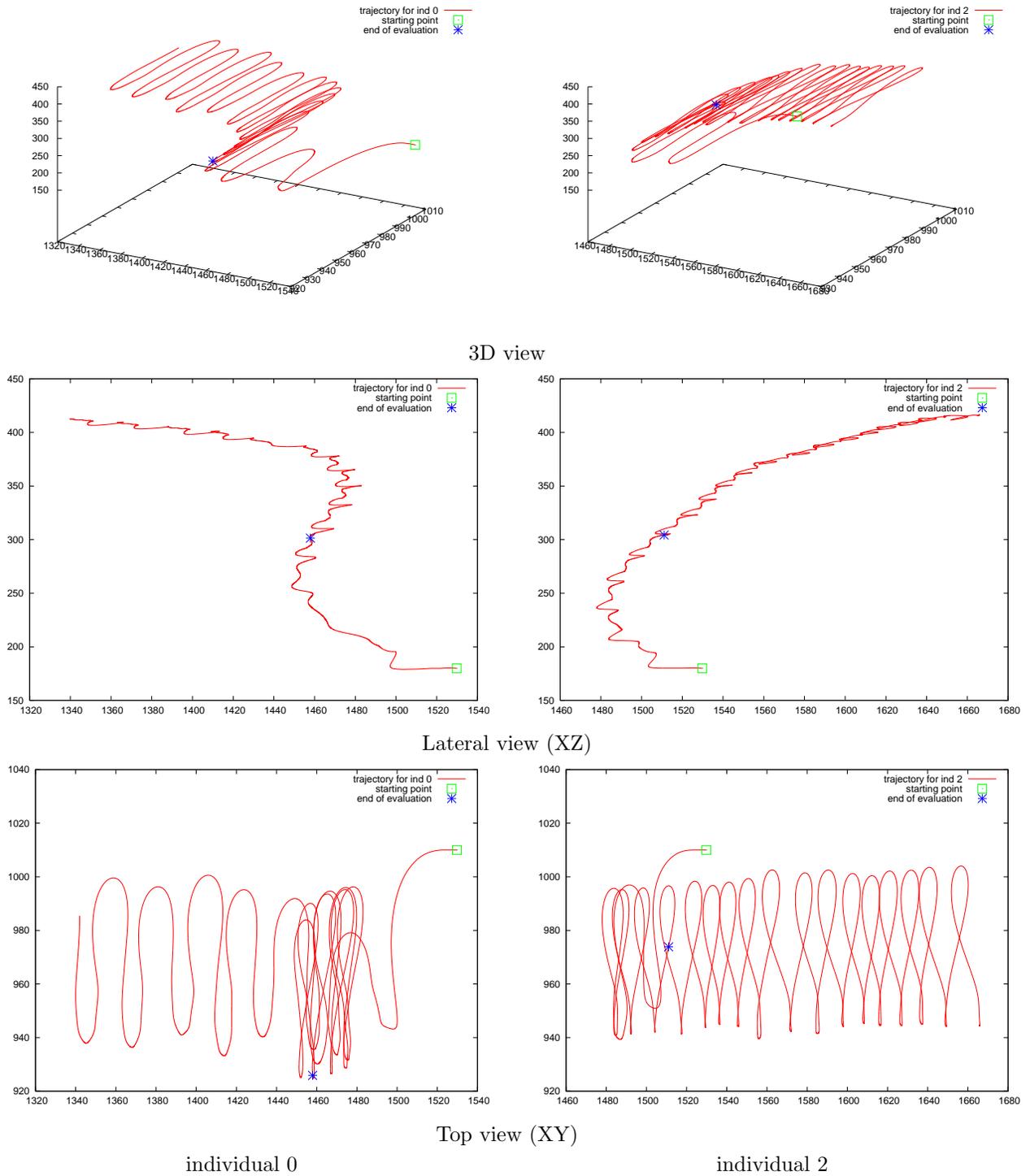


Figure 7. Trajectories of two individuals evolved for slope soaring. The corresponding evaluation periods are four times longer than those that were used to evaluate individuals during an evolutionary run (8min vs 2min).

mizes the connection weights of a feedforward neural network. This methodology has been applied to generate thermal and slope soaring behaviors for a simulated glider. Efficient gains in altitude have been demonstrated.

The evolved controllers do not call upon a direct evaluation of specific parameters describing the aerological conditions, but they respond to the perceived roll, pitch and vertical speed, for instance, so that their action results in a behavior that is appropriate to gain altitude. They do not "know" where they are and what they do, they just react to what they perceive. This simplicity affords a low computational cost that is compatible with current onboard computing devices.

VI. Acknowledgements

This study benefited from a grant from DGA. The corresponding work has been initiated during the internships of Mathieu Schmitt for slope soaring and Guillaume Tatur for thermal soaring.

References

- ¹Pennycuik, C. J., "The flight of petrels and albatrosses (Procellariiformes), observed in South Georgia and its vicinity," *Phil. Trans. R. Soc. Lond. B*, Vol. 300, 1982, pp. 75–106.
- ²Spaar, R. and Bruderer, B., "Migration by flapping or soaring: flight strategies of Marsh, Montagus and Pallid Harriers in southern Israel," *The Condor*, Vol. 99, No. 2, March-April 1997, pp. 458–469.
- ³Davies, M., "An exploratory analysis of dynamic soaring trajectories in shear layer," Tech. rep., Stanford University, 2004.
- ⁴Hedenström, A., Rosén, M., Akesson, S., and Spina, F., "Flight performance during hunting excursions in Eleonoras falcon *falco eleonora*," *The Journal of Experimental Biology*, Vol. 202, 1999, pp. 2029–2039.
- ⁵Reichmann, H., *Cross Country Soaring*, Soaring Society of America, Inc., 1993, ISBN 1-883813-01-8.
- ⁶Doncieux, S., Mouret, J.-B., Angeli, A., Barate, R., Meyer, J.-A., and de Margerie, E., "Building an Artificial Bird: Goals and Accomplishments of the ROBUR Project," *Proceedings of the European Micro Aerial Vehicles (EMAV 2006) conference*, 2006.
- ⁷Allen, M. J., "Autonomous Soaring for Improved Endurance of a Small Uninhabited Air Vehicle," *Meeting Paper AIAA-2005-1025, Research Engineering, NASA Dryden Flight Research Center*, 2005.
- ⁸Meyer, J.-A., "Artificial Life and the Animat Approach to Artificial Intelligence," *Artificial Intelligence*, edited by M. Boden, Academic Press, 1996, pp. 325–354.
- ⁹Wharington, J. and of Technology, R. M. I., *Autonomous Control of Soaring Aircraft by Reinforcement Learning*, Royal Melbourne Institute of Technology, 1998.
- ¹⁰Barate, R., Doncieux, S., and Meyer, J.-A., "Design of a bio-inspired controller for dynamic soaring in a simulated UAV," *Bioinspiration & Biomimetics*, Vol. 1, 2006, pp. 76–88.
- ¹¹Druot, T., "Technical report on the implementation and validation of a flight mechanics simulator for flapping articulated wings," Tech. rep., Available on <http://animatlab.lip6.fr>, 2004.
- ¹²Allen, M., "Updraft Model for Development of Autonomous Soaring Uninhabited Air Vehicles," *44 th AIAA Aerospace Sciences Meeting and Exhibit*, 2006, pp. 1–19.
- ¹³Gallego, A., "Creation de un tunel de viento virtual," 2002.
- ¹⁴Cimne, "GiD - The Personal Pre and Postprocessor. International Center for Numerical Methods in Engineering, Barcelona, Spain. [Online]. Available: <http://gid.cimne.upc.es/>," 2003.
- ¹⁵Meyer, J.-A., Guillot, A., Girard, B., Khamassi, M., Pirim, P., and Berthoz, A., "The Psikharpx Project: Towards Building an Artificial Rat," *Robotics and Autonomous Systems*, Vol. 50, No. 4, 2005, pp. 211–223.
- ¹⁶Bäck, Hoffmeister, F., and Schwefel, H., "A survey of evolution strategies," *Proceedings of the 4th International Conference on Genetic Algorithms*, Morgan Kaufmann, 1991.
- ¹⁷Deb, K., Mohan, M., and Mishra, S., "Evaluating the-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions," *Evolutionary Computation*, Vol. 13, No. 4, 2005, pp. 501–525.
- ¹⁸Landau, S., Doncieux, S., Drogoul, A., and Meyer, J.-A., "SFERES: un framework pour la conception de systemes multi-agents adaptatifs," *Technique et Science Informatiques*, Vol. 21, No. 4, 2002, pp. 427–446.
- ¹⁹Doncieux, S. and Meyer, J.-A., "Evolution of neurocontrollers for complex systems: alternatives to the incremental approach," *Proceedings of The International Conference on Artificial Intelligence and Applications (AIA 2004)*, 2004.
- ²⁰Doncieux, S. and Meyer, J.-A., "Evolving Modular Neural Networks to Solve Challenging Control Problems," *Proceedings of the Fourth International ICSC Symposium on engineering of intelligent systems (EIS 2004)*, 2004.
- ²¹Reichmann, H., *Cross-country Soaring*, Thomson Publications, 1978.