Chapter 1
Evolutionary Algorithms to Analyse and Design a Controller for a Flapping Wings Aircraft

Stéphane Doncieux and Mohamed Hamdaoui

Abstract Evolutionary Algorithms are now mature optimization tools, especially in a multi-objective context. This ability is used here to help explore, analyse and, on this basis, propose a controller for a complex robotics system: a flapping wings aircraft. A multi-objective optimization is performed to find the best parameters of sinusoidal wings kinematics. Multi-objective algorithms generate a set of trade-off solutions instead of a single solution. The feedback is then potentially more informative in a multi-objective context relative to the one of a single objective setup: the set of trade-off solutions can be analyzed to characterize the studied system. Such an approach is applied to study a simulated flapping wing aircraft. The speed-energy relation is empirically evaluated and the analysis of the relations between the parameters of the kinematics and speed has led, in a further step, to the synthesis of an open-loop controller allowing to change speed during flight.

1.1 INTRODUCTION

Evolutionary Algorithms (EA) nowadays belong to the classical toolbox of engineers as powerful optimization algorithms [5]. They differ from other techniques in the fact that they are black box, derivative free optimization algorithms. The function to be optimized, i.e. the cost function, also called the fitness function, may even not be known but just evaluated through a dedicated experimental device. This makes such tools valuable for non-linear and dynamical systems design and first applications have actually been performed directly on a real device [20, 21] as cited by [1].

S. Doncieux
ISIR Pierre and Marie Curie University, CNRS Pyramide Tour 55 Boite courrier 173 4, place Jussieu 75252 Paris cedex 05 France e-mail: stephane.doncieux@isir.upmc.fr, M. Hamdaoui
MAS Ecole centrale Paris, Grande Voie des Vignes Chatenay-Malabry, France e-mail: mhcsdl@gmail.com

Furthermore, as they are population-based, they can easily be adapted to a multi-objective context [6] and efficient multi-objective evolutionary algorithms (MOEA) now exists, as for instance NSGA-II [8] or ϵ-MOEA [7]. Actually, most real world problems involve multiple objectives to be optimized: cost and efficiency, for instance. When these objectives are antagonistic, there is no single optimal solution, but rather a set of optimal trade-off solutions. MOEA generate at once an approximation of this set of the best trade-off solutions whereas most other multi-objective algorithms only discover one trade-off solution at a time or require to know in advance some features of this set [16]. Examples of applications of MOEA are then numerous [3, 22, 4].

The cloud of optimal trade-off solutions generated by a multi-objective algorithm can be used to extract useful data on the considered problem. Actually, all these points share the specificity of being optimal relative to some user defined objectives and are arbitrarily close. They materialize then the possible values of the objectives and the relations between them and the evolved parameters. An expert of the field can exploit these data to design new and innovative solutions that go beyond the solutions handled by the MOEA. Deb and Srinivasan [9] suggested a systematic approach to these questions and called it *innovization*, for *innov*ation through optim*ization*. Their approach consists in first finding the set of Pareto-optimal solutions according to problem specific objectives. On the basis of these data, an analysis is made to identify regularities and relations between parameters. In this article, such an approach is applied to a simulated flapping-wings aircraft. The energy-speed relation is empirically evaluated and a simple open-loop speed controller is deduced from the analysis of the relations between the parameters and the speed.

Many studies have been performed by biologists that have observed natural devices of this kind. Thanks to statistical studies on flying animals or insects, biologists have identified relations between significant parameters like wing area, cruising speed, wing span, flapping wings frequency, wing load or mass [23]. Wing kinematics have also been studied for several species. Tobalske and Dial have observed, for instance, that pigeons and magpies use a relatively constant flapping frequency across their complete range of speed, i.e. 4 to $14m.s-1$ [25]. Meanwhile, Park et al. have studied the swallow in a wind tunnel and observed an U-shaped relation between frequency and speed [18], concluding that such relations may vary upon bird species. Physicists also try to unravel the physical mechanisms underlying the flapping flight. Some experiments have been conducted in which prototypes with a given kinematics were put in a wind tunnel to study and characterize their behavior [11]. In this case, the kinematics is given, but if we turn to a roboticist's point of view, the question becomes : for a prototype with known features, what are the interesting kinematics ? To our knowledge, no theory nor methodology do exist today to answer such practical questions. Likewise, no efficient flapping wing aircraft do exist together with its controller to help infer relations between the involved parameters. Birds or bats are sources of inspiration, but equivalent engineered flyers have fewer degrees-of-freedom and different aerodynamical features. Biological observations can't then be directly used. The proposed methodology aims at facilitating the design of such aircrafts by providing some simple yet efficient solutions that can be analyzed to better understand such systems before building more sophisticated solutions.

Stochastic optimization tools are very interesting in the context of flapping wings aircrafts due to the complexity of the relation between the parameters of a particular kinematics and the resulting speed (or aircraft crash...) [19, 24, 17] . The work presented here is different in that it does not aim at generating a single and particular optimal controller, but rather at generating a set of them, that are not the goal *per se* but rather a mean to study properties of the system. [15] had a similar goal, but a small number of speeds were initially chosen in the study and runs were launched for each value. In the present work, a continuum of speeds is explicitely sought. It should be noticed that the following results have been obtained in simulation and are thus highly dependent on simulation accuracy. Actually, such experiments may be done on a real prototype.

1.2 METHOD

The proposed approach consists in first generating a set of Pareto-optimal points. These points are particular in the sense that they share a common specificity: they are all optimal relative to some (antagonistic) user-defined objectives. Finding the common features of those points corresponds then to find what characterizes Pareto-optimal points: what are the critical parameter values ? How are they related to objectives? Furthermore, the MOEA we will use, i.e. NSGA-II, aims at finding a *dense* approximation of the Pareto front. This means that the generated solutions will be arbitrarily close one to the other. This proximity between points will make the analysis of the solutions easier and results, generally, in a continuum between the features of these solutions. A more detailled description of the approach follows.

The first step consists in choosing two antagonistic objectives (or more). The objectives explicitely need to be antagonistic in order to guarantee the existence of a set of trade-off solutions, rather than a single optimum. Although it may seem at first sight to restrict the field of application, in practice, it is easy to find such objectives: cost or energy related objectives are, for instance, generally antagonistic to performance related objectives (efficiency relative to the task to be solved, for instance).

Once these objectives are known, a search space has to be choosen. The search space defines the set of candidate solutions the EA will explore. All further results and analysis will be relative to this search space, its choice must then be done carefully.

The next step is straightforward: launch the optimization to find a good approximation of the Pareto front. Deb and Srinivasan [9] suggest to do it in several different steps: perform a standard multiobjective search, perform a single objective optimization to find the extremum points of the Pareto front and then use NCM [16] to find a set of points representing an uniformly distributed sampling of the Pareto front. All these different steps only aim at providing a better confidence in the quality of the

Pareto front approximation. In most real-world applications, there is no way to know it with precision, it is then important to do whatever possible to increase the confidence in the quality of the results, as the quality of the further analysis will critically depend on this part of the process. In the work reported here, several independant runs have been launched and also single objective optimizations have been tried for both extremum points and Pareto front points. As the points generated by single objective algorithms were clearly dominated by the points generated by NSGA-II[1], the focus in the following is only on the points generated by this last MOEA.

The cloud of Pareto-optimal points can then be used to extract useful information just like in [9]. The Pareto front in the objectives space shows the performance of the best solutions to be found relative to the given search space. Each parameter can also be plotted against a target objective or parameter and analytical functions approximating this relation can be found by regression, as it is done in [9]. But this can become cumbersome as the dimensionality of the Pareto front increases in the objectives or parameters spaces. For example, with n objectives and p parameters all the couples with the $n+p$ quantities have to be studied, which result in $\frac{(n+p-1)(n+p)}{2}$ plots to scrutinize. Looking at these plots individually can dramatically increase the time needed to extract the information from the Pareto front and may sometimes lead to inconclusive results. That is why specific visualization methods that can easily handle high dimensional data and can give at a glance a global view of the Pareto front in the objectives and/or parameters spaces have to be used to detect the groups of quantities that seem to be related and only for which, plots should be drawn.

In our case, the Pareto front is of dimension 8 in the parameters space and 2 in the objectives space. Therefore, visualizing the relationships between parameters and/or objectives on the Pareto front can become a lengthy and laborious process ($\sim$ 45 plots to review). Kohonen networks [12] are used here to facilitate this step. These networks are a nonlinear topology preserving projection method which allows to handle comfortably high dimensional data. The objective of Kohonen networks is to form a discrete, topological mapping of a Q-dimensional input space. The algorithm outputs a Kohonen map composed of output units often arranged in a 2-D rectangular or hexagonal grid, each ouput unit being characterized by its 2-D vector of coordinates and a Q-dimensional vector called codebook. To construct the mapping, the network is trained on a finite dataset of Q-dimensional vectors (input space) and produces a Kohonen map whose units are 2-D vectors (output space) in such a way that the topology of the data is preserved and that each vector of the input Q-dimensional dataset is associated to a unit on the map via its codebook. This unit is called *BMU* or best matching unit. Thus, Kohonen networks build a tractable low dimensional (the Kohonen map) representation of high dimensional datasets allowing a comfortable visualization of the data. A MATLAB® implementation of the method is freely provided by the SOM Toolbox[2]. This technique is used to visualize the Pareto front taking into account, for each Pareto-optimal solution, both

[1] we tried a simple rank-based EA.

[2] http://www.cis.hut.fi/somtoolbox/

parameters and objectives.

In the case of parameters describing a robot controller, this knowledge extraction step results in the ability to build a new and more complex controller relative to what EA has explored and generated : if the parameters are modified online using the found relationships we have a controller able to move along the Pareto front and thus to adapt to the context. Such an approach requires that such online modification is possible and efficient, what is not guaranteed, as the MOEA only explored constant parameters controllers. Anyway, the proposed new controller may be the subject of further refinements through another innovization step or through a simple optimization of its parameters concentrated on the transition phase between changes of parameters.

1.3 EXPERIMENTAL SETUP

To illustrate potential use of the method, a flapping wings controller able to adapt the speed of the aircraft to a given desired speed is designed while exploiting the data extracted from the Pareto-optimal set analysis. The principle of the controller consists in modifying online the parameters of the wings kinematics. The applied set of parameter values for the wing kinematics are the closest possible to those of Pareto-optimal points associated with a similar speed. This requires to extract the relationships between the speed and the different parameters of the kinematics and this is what the MOEA analysis is used for.

A cloud of Pareto-optimal points that correspond to different speeds is generated and analysed, using Kohonen maps, in both objectives and parameters spaces to estimate qualitatively the relationships between the different quantities at hands. This qualitative study will ease the identification of the parameters that noticeably vary with the velocity while allowing us to spot interesting regions of the Pareto front. Then, by doing some regression on the withheld kinematical parameters versus the velocity, a law of variation for each parameter relative to speed will be inferred and used to drive the online wing kinematic parameter settings.

Tests will then be performed in simulation to see if it actually allows to change the speed of the flapping wings aircraft. Starting from a simple family of functions, i.e. sinusoidal functions, allowing to fly at a constant speed, a controller able to adapt the speed of the aircraft will be designed[3].

The first goal is to generate the set of pareto-optimal points to be studied. Such points should represent a varying speed: speed will be the first objective. An antagonistic objective is required to generate a set of individuals and not a single point. Energy will be used to this end. Two optimizations will be performed: one will try to maximize speed, while the other one will try to minimize it. The largest range of

[3] This will still be an open-loop controller, as it won't take into account real speed, we will propose an extension to design a closed-loop controller in the discussion.

speeds is required to make the speed control interesting. In the case of an U-shaped relation between speed and energy, two kinds of experiments need to be launched:

- speed minimization will materialize the lowest speed points
- speed maximization will materialize the highest speed points

The points generated by the two experiments are then merged to generate the cloud of points to study. Wing kinematics are described by the following equations:

$$DI = a_{DI} \sin(2\pi t/p_{DI}) \tag{1.1}$$

$$TWi = r_{TWi} + a_{TWi} \sin(2\pi(t/p_{DI} + p_{TWi})) \tag{1.2}$$

$$TWe = r_{TWe} + a_{TWe} \sin(2\pi(t/p_{DI} + p_{TWe})) \tag{1.3}$$

where DI is the wing dihedral, TWi the internal twist and TWe the external twist. Wing kinematics is then described by eight parameters subject to the evolutionary optimization. The chosen ranges for each value is the following:

- amplitude of the dihedral: a_{DI} in $[0; 45^o]$
- period of the dihedral (and of all other degrees of freedom): p_{DI} in $[0.2; 1s]$
- reference of the internal twist: r_{TWi} in $[-22.5; 22.5^o]$
- amplitude of the internal twist: a_{TWi} in $[0; 45^o]$
- phase of the internal twist: p_{TWi} in $[0; 1]$
- reference of the external twist: r_{TWe} in $[-22.5; 22.5^o]$
- amplitude of the external twist: a_{TWe} in $[0; 45^o]$
- phase of the external twist: p_{TWe} in $[0; 1]$

NSGA-II is used to perform the search, with a population size of 500 and during 1000 generations. Evolved parameters are represented as vectors of real values with a polynomial mutation and a sbx crossover, as described in [6], p124.

Aerodynamical forces created by wing movements are computed with a semi-empiric, quasi-steady-aerodynamics model. Each wing is decomposed in three panels. For each panel, the local incident airspeed is evaluated and the leading edge lift, the parachute drag and the friction drag are computed and summed. See [15, 10] for a detailed description of the model. The parameters of the aircraft are given in Appendix. The panels are considered as not deformable solids, connected to their neighbors via joints, as shown on figure 1.1. The integration of the forces and the movements of the parts are computed using ODE[4].

The objectives to maximize are the following:

- experiment 1:
  - + average aircraft speed
  - - average mechanical power
- experiment 2:
  - - average aircraft speed

[4] http://ode.org

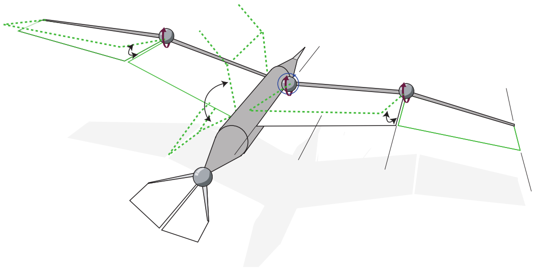


Fig. 1.1 Degrees-of-freedom of the simulated flapping wing aircraft.

- - average mechanical power

Following [15], the instantaneous mechanical power is computed as the scalar product between the instantaneous torque (τ) and the instantaneous rotational speed (ω) for a joint: $P = \sum_i |\tau_i.\omega_i|$. The mechanical power objective only takes into account the shoulder joints which are the main contributors to energy consumption. It should be noted that the power is always considered as positive. The mechanical power is then over estimated, as the torques required to accelerate or to slow down are considered as equivalent.

1.4 RESULTS

Three different runs have been launched for experiment 1 and experiment 2. Each run shows a similar Pareto front, except for points at highest energy. This result is not surprising as these individuals are at the limits of the simulation we used: small changes may have a huge consequence on the stability of the simulation, thus making such part of the search space difficult to explore.

For each experiment, the set of non dominated solutions out of the three runs is extracted and the two resulting sets are merged (figure 1.2). Dots follow an U-shape, with a minimal energy consumption of 24W at $10.7m.s^{-1}$. It can be noticed that the sets of non dominated solutions for experiment 1 and experiment 2 coincide for this point of minimal energy consumption with no particular discontinuity.

A Kohonen map of the Pareto front in both objectives and parameters spaces is built. Multiple representations of this map, colored according to the objectives and kinematic parameters, are provided on Figures 1.3-1.5. For example, the Figure 1.3 top represents the map colored by the values of the U-matrix with the color code specified on a colorbar next to the map. The U-matrix is a matrix of distances between the units of the map computed in the Q-dimensional input space: as each unit of the map has a Q-dimensional codebook, the distance between units in the Q-dimensional space is the distance (euclidian for example) between the Q-dimensional codebook vectors associated to these units. To quantify the relative

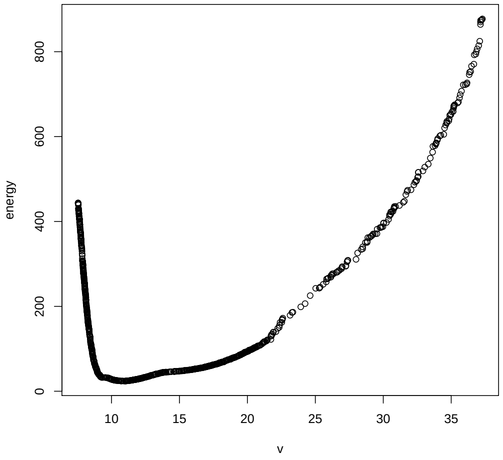


Fig. 1.2 Pareto-optimal points extracted from the results of the three speed optimization and the three speed minimization experiments. Each dot represents the optimal solution found by the EA for a given speed relative to the energy consumption. X-axis: speed in $m.s^{-1}$, Y-axis: instantaneous energy in W

importance of the different components of the Q-dimensional codebook vector in the U-matrix computation, some components may be set to zero while the others are taken into account in computation of the distance to assess their importance for the U-matrix. The U-matrix allows to spot groups of similar solutions (areas on the map where the U-matrix takes low values). The mapping error is quantified by computing the average distance between each vector in the data and its *BMU*. The error associated with the built mapping is of 8 $\%$.

Four main similar areas or clusters can be seen on the map colored by the values of the U-matrix (see Figure 1.3, top). Three of them (Cluster 1-3) are located in the region of low velocities and the other one (Cluster 4) is in the region of high velocities as can be seen by comparing Figure 1.3, top and Figure 1.3, middle. The separation line between Cluster 1-3, in one hand, and Cluster 4, in the other hand, seems to correspond to the separation between the two merged Pareto fronts (one obtained for high velocities and the other for low velocities). To make sure of that we computed an U-matrix based on the values of the velocity only, represented on Figure 1.3, bottom, left, which shows that this separation line is strongly related to the velocity.

The internal twist amplitude a_{TWi} (see Figure 1.3, bottom, right) does not change much on the map as it takes high values for very low velocities and low values the rest of the time. This means that this kinematic parameter is not sensitive to the velocity except for very low velocities.

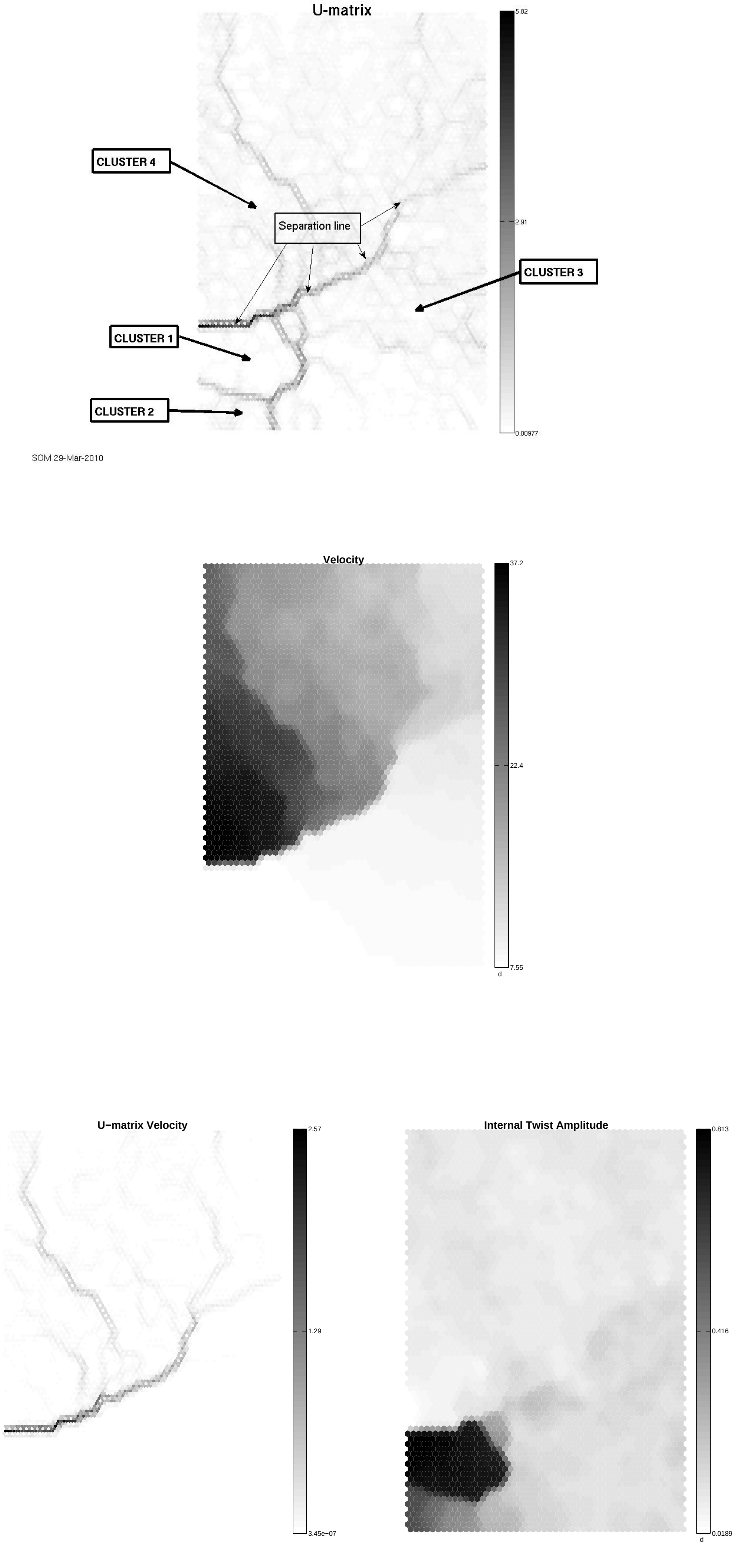


Fig. 1.3 Top, : the U-matrix and the four clusters. Middle: the Kohonen map colored by the velocity values. Bottom, left: the U-matrix based on the velocity only. Bottom, right: the Kohonen map colored by the Internal Twist Amplitude a_{TWi}.

On the opposite, the internal twist reference r_{TWi}, internal twist phase p_{TWi}, external twist amplitude a_{TWe}, external twist reference r_{TWe} (see Figure 1.4) and external twist phase p_{TWe} (see Figure 1.5, top, left) are related to the velocity on the map. Indeed, the separation line between Cluster 1-3, in one hand, and Cluster 4, on the other hand appears on the Kohonen maps colored by the values of r_{TWi}, p_{TWi}, a_{TWe}, r_{TWe} or p_{TWe}. As this separation line is closely related to the evolution of the velocity as it appears on the Kohonen map colored by velocity, this means that these kinematic parameters are closely related to the velocity.

This separation line appears too on the Kohonen map colored by the values of the energy (see Figure 1.5, top, right), which means that the energy is related to the velocity too. On the opposite we do not see this separation line for the dihedral period p_{DI} (see Figure 1.5, bottom, left) or the dihedral amplitude a_{DI} (see Figure 1.5, bottom, right), which means that these two kinematical parameters might not be related to the velocity, what can be confirmed by a plot of these two kinematic parameters against the velocity.

Finally, building a Kohonen map allowed us to know that the internal twist amplitude a_{TWi} does not strongly depend on the velocity (except for very low velocities) and that the other kinematic parameters except the dihedral period p_{DI} and the dihedral amplitude a_{DI} clearly depend on the velocity.

All the parameters of the sinusoidal wing kinematics can be plotted, for each Pareto optimal point, in order to check their relation to speed (figure 1.6).

The relation between each parameter and the speed has been approximated with polynomial regression methods. The corresponding relations are reported in Appendix. Resulting approximations are also plotted on figure 1.6.

We used then these approximated functions to pilot the parameters of the same simulated bird (figure 1.7). The control is oscillating, but as did the original controllers: the fitness did only measure the mean speed, not the ability of the controller to reduce its standard deviation. The control is not efficient at all for $8m.s^{-1}$, meaning that the approximation is not good for this speed. For other speeds, the simulated bird speed tends to oscillate but with a decreasing amplitude. For a speed of 10 or $11m.s^{-1}$ the bird altitude isn't perfectly maintained, resulting in an increasing speed error when the bird starts to dive. For each desired speed, the measured mean speeds are the following:

Desired speed	average speed	abs. error
8	10.15	27%
10	10.64	6.4%
11	11.23	2%
13	13.45	3.5%
15	15.39	2.6%
20	17.61	12%
25	33.94	36%

For speed ranging from 10 to $15m.s^{-1}$, the error is several percents, but it gets larger for small or high speeds. This is not surprising as the physical behavior is less stable in these cases.

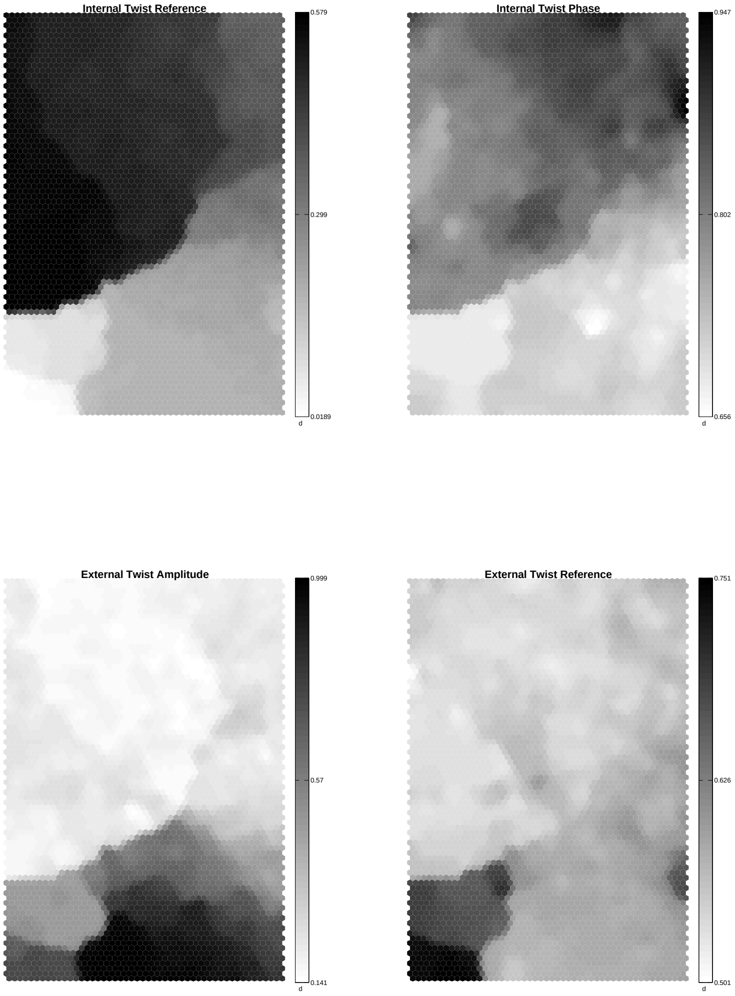


Fig. 1.4 Top, left: the Kohonen map colored by the Internal Twist Reference r_{TWi}. Top, right: the Kohonen map colored by the Internal Twist Phase p_{TWi}. Bottom, left: The Kohonen map colored by the External Twist Amplitude a_{TWe}. Bottom, right: the Kohonen map colored by the External Twist Reference r_{TWe}

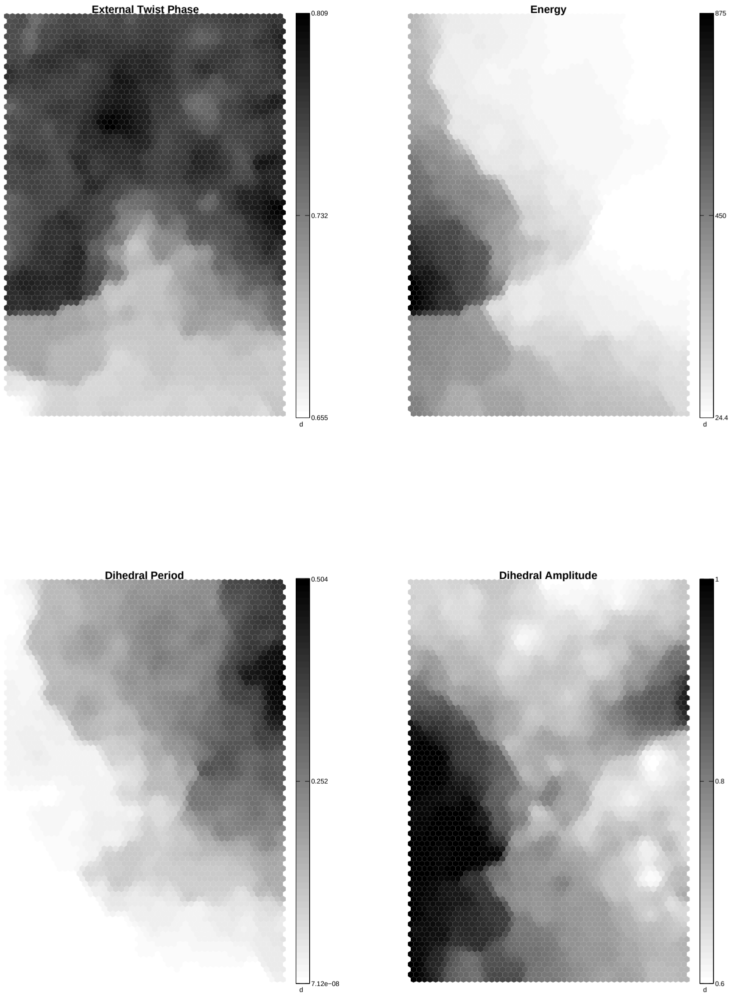
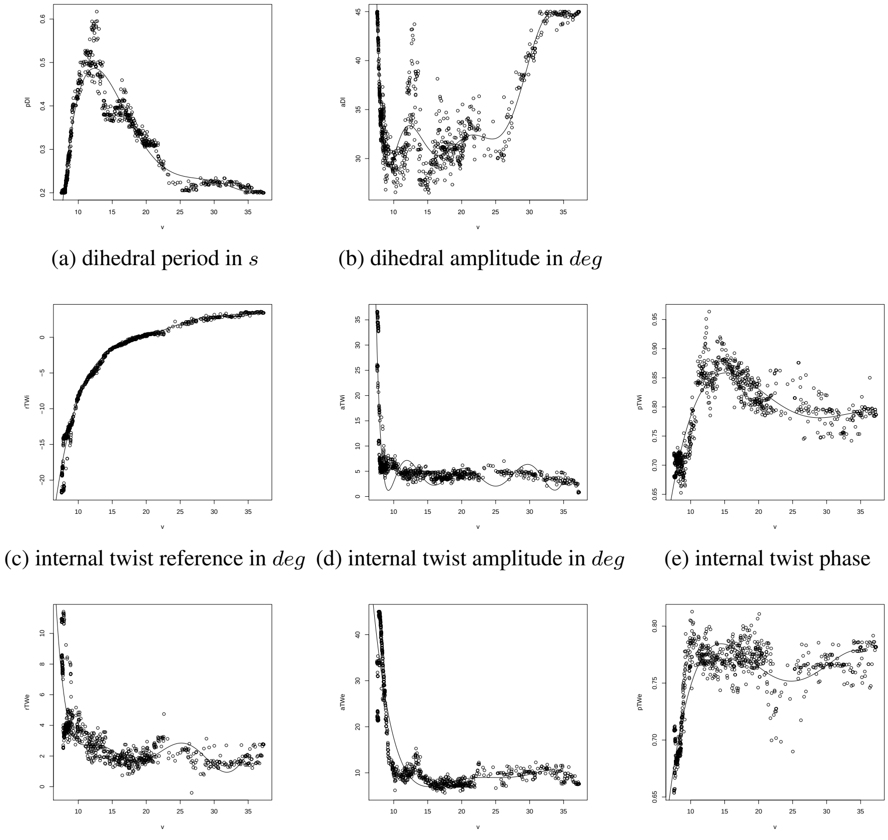


Fig. 1.5 Top, left: the Kohonen map colored by the External Twist Phase p_{TWe}. Top, right: the Kohonen map colored by the energy. Bottom, left: The Kohonen map colored by the Dihedral Period p_{DI}. Bottom, right: The Kohonen map colored by the Dihedral Amplitude a_{DI}.



(a) dihedral period in s

(b) dihedral amplitude in deg

(c) internal twist reference in deg

(d) internal twist amplitude in deg

(e) internal twist phase

(f) external twist reference in deg

(g) external twist amplitude in deg

(h) external twist phase

Fig. 1.6 Empirical dependency of each parameter relative to speed. Plot of the repartition of Pareto-optimal points for each parameter. Each dot represents a pareto optimal solution. Solid lines represent the approximated fit with a polynomial relation whose parameters are given in Appendix, figure 1.9.

We have also tried to change it online, i.e. during a flight. To avoid physically unrealistic behavior, we haven't abruptly changed the parameters, but we have waited for the wings' dihedral to pass near zero. Once a small angle is reached, we let the wings still and wait until the new kinematics also reaches a small angle and we switch the kinematics only then. Results are reported on figure 1.8. What we observe is that the bird actually changes its speed dynamically, simply as a result of wing kinematics parameter change (open-loop modification). Actually, we observe a kind of undesired memory in the system: the control at a given speed may show different performances (different static errors) depending on the historical context, i.e. depending on the initial conditions when the switch is performed. During evolution, every individual started from a single condition: horizontal flight at $11m.s^{-1}$;

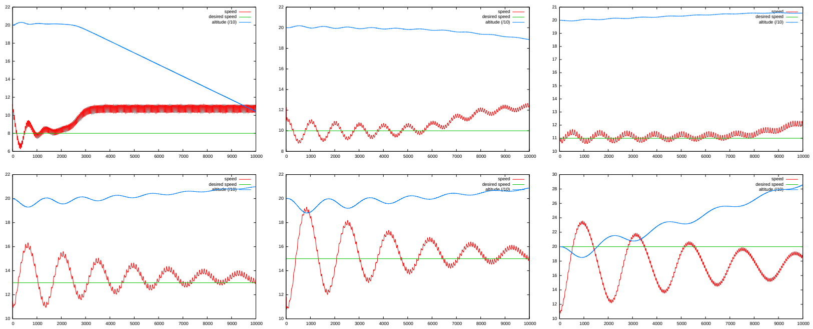


Fig. 1.7 Speed of the simulated bird controlled by a sinusoidal kinematics whose parameters are obtained from the regression at a given desired speed. The controller is an open-loop controller, real speed is not taken into account to finely tune it.

here, the 'starting' speed may be very different. Anyway the bird remains stable and at a relatively constant altitude (there is no active control at all: none on speed but none on altitude also).

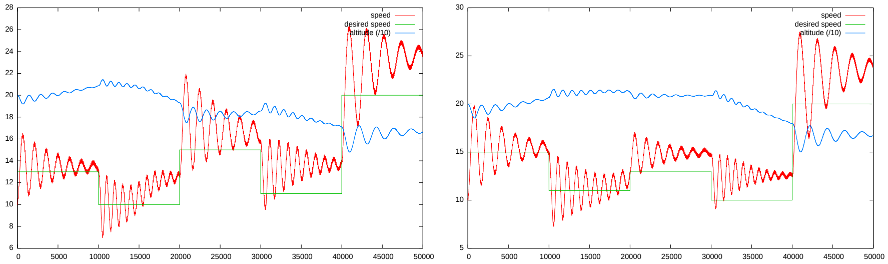


Fig. 1.8 Speed of the simulated bird controlled by a sinusoidal kinematics whose parameters are obtained from the regression at a given desired speed. The controller is an open-loop controller, real speed is not taken into account to finely tune it. The parameter of the wing kinematics are changed during flight.

1.5 DISCUSSION AND FUTURE WORK

All the search was performed on the basis of a simplified simulation. The conclusions are then only relative to the model that was used and are to be confirmed by experiments on a real device. Anyway, such an approach can be used directly on a real prototype. Its main drawback is then the high number of experiments required to find the Pareto front. Some solutions do exist to this problem. A simplified simulation may be used first to reject the most inefficient solutions while testing on the prototype only the most pertinent ones. The discrepancy between an experiment in simulation and reality may also be used to drive a model learning loop aimed at reducing the gap between the two [2, 13, 14].

The simple open loop controller that we have synthesized is clearly not optimal. An optimization trying to generate solutions more robust to the initial conditions might be required to avoid the observed memory effect. Another innovization step might also be launched to study the transition between different speeds, the compromise here being for instance between the speed of transition and its stability. Likewise, starting from the generated open-loop controller, we could close the loop and optimize the parameter of a controller in which the speed error is added to the desired speed.

All this study is based on the features of Pareto optimal points found by the MOEA. The generated Pareto fronts correspond to performance to be expected with a simulated bird flying with sinusoidal wing kinematics. Changing the family of wing kinematics may change the shape of Pareto fronts. At least, using such an approach, we might empirically compare different kinds of kinematics and look at the advantage of using more generic periodic functions, for instance.

MOEA were used to generate the set of Pareto optimal solutions. Other methods could have been used. In many simple cases, an exhaustive search might do the job. Here, it would be difficult to use it as we have eight continuous parameters. If we discretize and consider only ten different values, then we have a 10^8 search space. A single MOEA run did around 200,000 different evaluations. As we did 6 different runs (3 for speed maximization and 3 for speed minimization), the total number of evaluations we made is then $1.2 * 10^6$, thus two orders of magnitude below an exhaustive search. Furthermore the search space explored by the MOEA was continuous and not a discretized one.

The main point here is that EA might be used for other purposes than mere optimization. Here, each generated point is not an interesting solution in itself, it was interesting as a mean to capture some regularities of the problem that have then been exploited manually. This suggests thus another use of EA as an exploration tool during the very first steps of a design process, whereas it is usually used at the very end, when there just remains several parameters to tune. This particular use requires the analysis of an expert and is not aimed at automatically designing a solution. It is rather a tool aimed at helping the engineer or the scientist to gain better insights about the problem to be solved.

1.6 CONCLUSIONS

In this work, we have exploited the ability of MOEA to generate a set of Pareto optimal points not just to discover such points and choose one among them, but rather to get some insights on the relationships between those points. We have used it to empirically evaluate the trade-off between speed and energy with a constant morphology.

The Pareto front has then ben visualized by Kohonen maps to spot the kinematic parameters that noticeably vary with the velocity. Each selected parameter have then been expressed as a function of speed thanks to a regression performed on Pareto-optimal solutions. An open-loop controller able to change speed along flight has been synthesized and tested in simulation.

1.7 ACKNOWLEDGMENTS

This work has been supported by the DGA/D4S REI research grant #06.34.022.

APPENDIX

Parameters of the MOEA:

- MOEA: NSGA-II
- population size: 500
- number of generation: 1000
- number of independant run performed for each experimental context: 3
- mutation rate 0.1
- mutation type: polynomial, η_m: 15 and η_c: 10
- crossover: sbx

param	v^0	v^1	v^2	v^3	v^4	v^5	v^6	v^7	v^8	v^9
St_r	1.3e2	57.0	11	-1.1	6.5e-02	-2.3e-3	4.5e-5	3.7e-7	-	-
p_{DI}	-3.89e0	1.07e0	-9.68e-2	4.09e-3	-8.27e-5	6.45e-7	-	-	-	-
r_{TWi}	-1.16e2	2.44e1	-2.07	8.78e-2	-1.82e-3	1.47e-5	-	-	-	-
p_{TWi}	-2.89e-1	2.10e-1	-1.35e-2	3.58e-4	-3.39e-6	-	-	-	-	-
a_{TWi}	1.30e4	-6.60e3	1.44e3	-1.76e2	1.34e1	-6.57e-1	2.09e-2	-4.14e-4	4.66e-6	-2.28e-8
r_{TWe}	3.55e2	-1.38e2	2.24e1	-1.94e0	9.62e-2	-2.76e-3	4.22e-5	-2.68e-7	-	-
p_{TWe}	2.11e-2	1.49e-1	-1.04e-2	2.97e-4	-3.03e-6	-	-	-	-	-
a_{TWe}	3.00e2	-6.69e1	5.94e0	-2.55e-1	5.35e-3	-4.38e-5	-	-	-	-

Fig. 1.9 Parameters of the polynomials approximating the relations $parameter = f(v)$ for each parameter of wing kinematics submitted to optimization.

Parameters of the aircraft:

- wing span: $1.93m$
- aspect ratio: 8.5
- wing area: $0.407m^2$
- total mass: $1.3kg$

- fuselage mass: $0.915kg$
- wing mass: $0.4kg$
- elevator/rudder: $0.038kg$

References

1. Back, T., Hoffmeister, F., Schwefel, H.: A survey of evolution strategies. In: Proceedings of the Fourth International Conference on Genetic Algorithms (1991)
2. Bongard, J., Lipson, H.: Nonlinear System Identification Using Coevolution of Models and Tests. Evolutionary Computation, IEEE Transactions on **9**(4), 361–384 (2005)
3. Coello, C., Lamont, G., Van Veldhuizen, D.: Evolutionary algorithms for solving multi-objective problems. Springer-Verlag New York Inc (2007)
4. Coello-Coello, C., Lamont, G. (eds.): Applications of Multi-Objective Evolutionary Algorithms. World Scientific (2004)
5. Dasgupta, D., Z., M. (eds.): Evolutionary Algorithms in Engineering Applications. Springer-Verlag Berlin and Heidelberg GmbH and Co. K (1997)
6. Deb, K.: Multi-objective optimization using evolutionay algorithms. Wiley (2001)
7. Deb, K., Mohan, M., Mishra, S.: Evaluating the epsilon-Domination Based Multi-Objective Evolutionary Algorithm for a Quick Computation of Pareto-Optimal Solutions. Evolutionary Computatition **13**(4), 501–525 (2005)
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002)
9. Deb, K., Srinivasan, A.: INNOVIZATION: Discovery of Innovative Design Principles Through Multiobjective Evolutionary Optimization. Multiobjective Problem Solving from Nature: From Concepts to Applications p. 243 (2007)
10. Druot, T.: Technical report on the implementation and validation of a flight mechanics simulator for flapping articulated wings. Tech. rep. (2004)
11. Hubel, T., Tropea, C.: Experimental investigation of a apping wing model. Experiments in Fluids **46**, 945–961 (2009)
12. Kohonen, T.: Self-Organizing Maps, *Springer series in information sciences, ISSN 0720-678X ; 30*, vol. 1, 3rd edn. Springer, Berlin, Germany (2001). ISBN : 3-540-67921-9
13. Koos, S., Mouret, J., Doncieux, S.: Automatic system identication based on coevolution of models and tests. In: Proceedings of IEEE Conference on Evolutionary Computation (2009)
14. Koos, S., Mouret, J.B., Doncieux, S.: Crossing the Reality Gap in Evolutionary Robotics by Promoting Transferable Controllers. In: GECCO'10: Proceedings of the 12th annual conference on Genetic and evolutionary computation ACM, publisher (2010)
15. de Margerie, E., Mouret, J.B., Doncieux, S., Meyer, J.A.: Artificial evolution of the morphology and kinematics in a flapping-wing mini UAV. Bioinspir. Biomim. **2**, 65–82 (2007)
16. Messac, A., Mattson, C.: Normal constraint method with guarantee of even representation of complete pareto frontier. AIAA journal **42**(10), 2101–2111 (2004)
17. Mouret, J.B., Doncieux, S., Meyer, J.A.: Incremental evolution of target-following neuro-controllers for flapp ing-wing animats. In: S. Nolfi, G. Baldassare, R. Calabretta, J. Hallam, D. Marocco, J.A. Meyer, O. Miglino, D. Parisi (eds.) From Animals to Animats: Proceedings of the 9th International Con ference on the Simulation of Adaptive Behavior (SAB), pp. 606–618. Rome, Italy (2006)
18. Park, K., Rosen, M., Hedenstrom, A.: Flight kinematics of the barn swallow (hirundo rustica) over a wide range of speeds in a wind tunnel. J. Exp. Biol. **204**(15), 2741–2750 (2001)
19. Rakotomamonjy, T.: Modelisation et controle du vol d'un microdrone a ailes battantes. Ph.D. thesis, Universite Paul Cezanne (2006)
20. Rechenberg, I., Eigen, M.: Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution (1973)

21. Schwefel, H.: Evolutionsstrategie und numerische Optimierung. TU Berlin, Germany (1975)
22. Tan, K., Khor, E., Lee, T.: Multiobjective evolutionary algorithms and applications. Springer (2005)
23. Tennekes, H.: The simple science of ight (from insects to jumbo jets). MIT Press (1996)
24. Thomson, S., Mattson, C., Colton, M., Harston, S.P., Carlson, D., Culter, M.: Experiment-based optimization of flapping wings kinematics. In: Proceedings of the 47th Aerospace sciences meeting (2009)
25. Tobalske, B., Dial, K.: Flight kinematics of black-billed magpies and pigeons over a wide range of speeds. J. Exp. Biol. **199**(2), 263–280 (1996)