Evolutionary algorithms as exploration and analysis helper tools, application to a flapping wings aircraft

Stéphane Doncieux

Evolutionary Algorithms as Exploration and Analysis Helper tools, Application to a Flapping Wings aircraft

Stéphane Doncieux

Abstract— **Evolutionary Algorithms are often used as an optimization tool. We use this ability 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. The obtained results are not the end of the process, but rather the begining. Their analysis provides information concerning the simulated prototype: the speed-energy relation is empirically evaluated. The analysis of the regularities of wings kinematics parameters also provides interesting insights that has led, in a further step, to the synthesis of an open-loop controller allowing to change speed during flight.**

I. INTRODUCTION

Evolutionary Algorithms (EA) nowadays belong to the classical toolbox of engineers as powerful optimization algorithms [1]. They differ from other techniques in the fact that they are derivative free optimization algorithms, i.e. they don't need to know the derivative of the function to be optimized, they don't even require to know an analytic expression of the function to be optimized. Actually, they just require the ability to evaluate the value it takes at different points.

Real world problems generally include 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. EA are indeed particularly adapted to this context, because of their population-based feature: they can approximate the whole Pareto front at once – or at least a significant part of it – whereas most other multi-objective algorithms only discover one trade-off at a time [2]. This particular feature has made them very popular and examples of applications are numerous [3], [4], [5].

Besides this use, EA, and in particular Multi-Objective EA (MOEA) can be used to better understand a particular problem through an analysis of the generated solutions. As MOEA don't generate a single solution but a set of solutions, an analysis of the relationships between those particular points may provide interesting insights on the problem that can lead, in a second step, to the design by an expert of the field of new and innovative solutions that go beyond the solutions handled by the MOEA. This allows, for instance, to evaluate the efficiency of simple and easy to analyse solutions before designing more complex ones while relying on the analysis to drive the design. Deb and Srinivasan [6] 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, we applied such an approach to a simulated flapping-wings aircraft. 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 [7]. 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 all range of speed, i.e. 4 to $14m.s{-}1$ [8]. Meanwhile, Park et al. have studied the swallow in a wind tunnel and observed an U-shaped relation between frequency and speed [9], 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 conduced in which prototypes with a given kinematics were put in a wind tunnel to study and characterize their behavior [10]. 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 ? The proposed methodology aims at providing some elements of answer to this question, in a setup that is as simple as possible. Contrary to biologists, we don't already have efficient aircrafts together with their controllers to help infer relations between the involved parameters. We are then facing a chicken-and-egg problem that the proposed approach tries to solve or at least to bootstrap.

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...) [11], [12], [13] . 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. [14] 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 work reported here, we are interested here in generating a continuum of speeds.

It should be noticed that the following results have been obtained in simulation and are thus highly dependent on

S. Doncieux is with ISIR Pierre and Marie Curie University, CNRS Pyramide Tour 55 Boite courrier 173 4, place Jussieu 75252 Paris cedex 05 France stephane.doncieux@isir.upmc.fr

simulation accuracy. Actually, such experiments may be done on a real prototype. We will discuss this point in section V.

The proposed approach is the following: we first use a MOEA to generate the set of trade-offs between two conflicting objectives (speed and energy). We analyze then the results. To illustrate possible use of this analysis, we then synthesize a simple open-loop controller allowing to adapt the speed while in flight by changing only the parameters of the sinusoidal kinematics of the wings.

II. METHOD

The approach we used here 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 finding what characterizes Pareto-optimal points. 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 result, generally, in a continuum between the features of these solutions. We will now turn to a more detailed description of the approach.

The first step consists in choosing two antagonistic objectives (or more). The objectives explicitely need to be antagonistic in order to garantee 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 chosen. 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. Besides expert of the field knowledge, search space choice must also take into account the analysis step of the method. As the goal is to understand how Pareto-optimal solutions are related, the chosen search space should allow to do it as easily as possible.

The next step is straightforward: launch the optimization to find a good approximation of the Pareto front. Deb and Srinivasan [6] suggest to do it in several different steps: perform a standard multi-objective search, perform a single objective optimization to find the extremum points of the Pareto front and then use NCM 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 fact that the Pareto front discovered by the optimization algorithms are indeed near the true Pareto front. 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, we have done several independant runs and also tried single optimization 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], we will focus in the following only on the points generated by this last MOEA.

The cloud of Pareto-optimal points can then be used to extract useful information. The Pareto-front itself is interesting, as it shows the performance of the best solutions to be found relative to the given search space. Each parameter can also be plotted relative to one objective or the other, and some regression may be performed to find an analytical function approximating this relation. In the case of parameters describing a robot controller, this 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 though another innovization step or through a simple optimization of its parameters concentrated on the transition phase between changes of parameters.

III. EXPERIMENTAL SETUP

To illustrate the proposed method, we have applied it to the design of a flapping wings controller able to adapt the speed of the aircraft to a given desired speed. We will use a MOEA analysis to extract the relationships between the speed and the different parameters of the kinematics. The principle is the following: we generate a cloud of Pareto-optimal points that correspond to different speeds and we analyse how the parameters change relative to the speed. By doing some regression on these clouds of points, we will get a law of variation for each parameter relative to speed. We will then test it 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, we will then have a controller able to adapt the speed of the aircraft[2].

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. We should get the largest range of speeds, even in the case of an U-shaped relation between speed and energy: we will get speeds below the most energy efficient speed in the speed minimization experiments and likewise speeds above that value in the speed maximization

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

[2] 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.

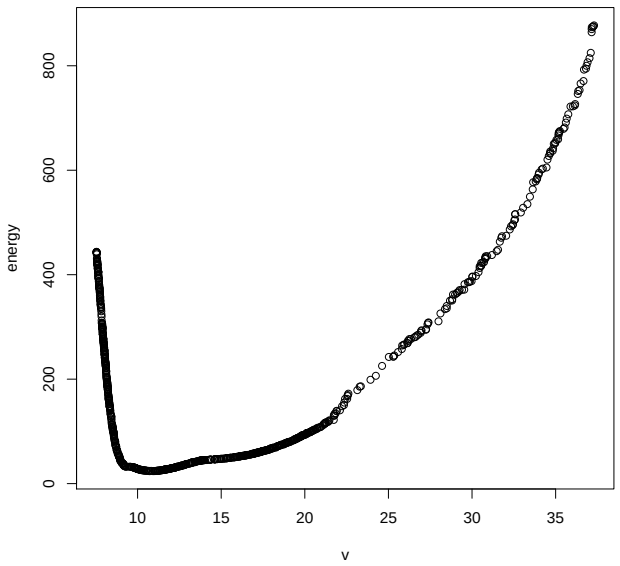


Fig. 1. Pareto-optimal points extracted from the results of the three speed optimization and the three speed minimization experiments. Each dot represents the optimal solution found by the EA for a given speed relative to the energy consumption. X-axis: speed in $m.s^{-1}$, Y-axis: instantaneous energy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

experiments. 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:

$$DI = a_{DI} \sin(2\pi t/p_{DI}) \quad (1)$$

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

$$TWe = r_{TWe} + a_{TWe} \sin(2\pi(t/p_{DI} + p_{TWe})) \quad (3)$$

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: $rTWe$ 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 $[0; 1]$

We have used NSGA-II 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 [2], p124.

We have used the simulator described in [14], [15]. The parameters of the aircraft are given in Appendix.

IV. RESULTS

Two different sets of runs have been performed: one attempting to maximize speed and the other one trying to minimize it. Both of them had a second objective: energy consumption, that had to be minimized. Three different runs have been launched for each set. Each run shows a similar pareto front, except for points at highest energy. This result is not surprizing 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.

Figure 1 shows the pareto fronts obtained by taking the non-dominated solutions of the three runs for each set: dots follow a U-shape, with a minimal energy consumption of 24W at $10.7m.s^{-1}$.

The parameters of the sinusoidal wing kinematics for each pareto optimal point can be plotted in order to find out how they are related to the speed (figure 2).

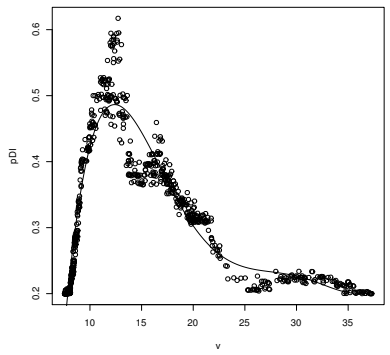
For each parameter, we have performed a polynomial regression, choosing the degree empirically: lower order have been tried first and the order has been increased until the match is visually satisfying. We have thus obtained the relations reported in Appendix, figure 5, were each parameter is a polynomial function of speed. Resulting approximation is also plotted on figure 2.

We used then these approximated functions to pilot the parameters of the same simulated bird (figure 3). 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:

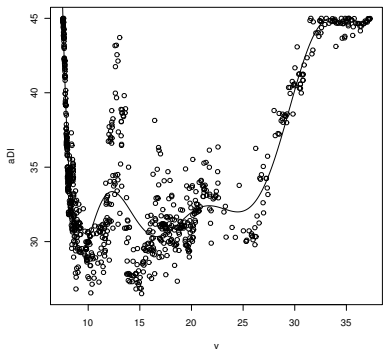
Desired speed	average speed	abs. error
8	10.15	27%
10	10.64	6.4%
11	11.23	2%
13	13.45	3.5%
15	15.39	2.6%
20	17.61	12%
25	33.94	36%

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.

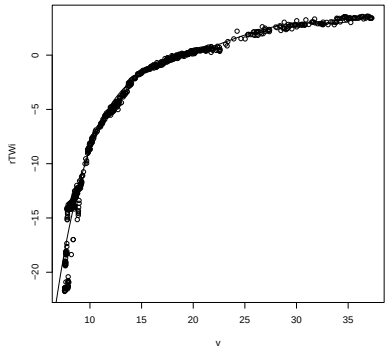
We have also tried to change it online, i.e. during a flight. To avoid physically unrealistic behavior, we haven't abruptly changed the parameters, but we have waited for the wings' dihedral to pass near zero. Once a small angle is reached, we let the wings still and wait until the new kinematics also reaches a small angle and we switch the kinematics only



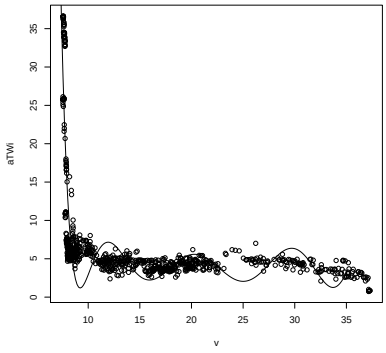
(a) dihedral period in s



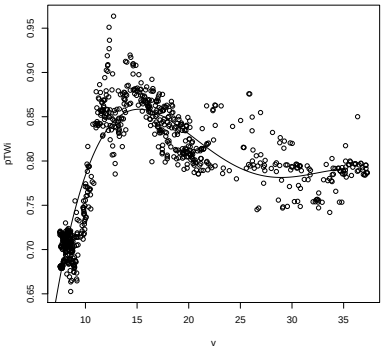
(b) dihedral amplitude in deg



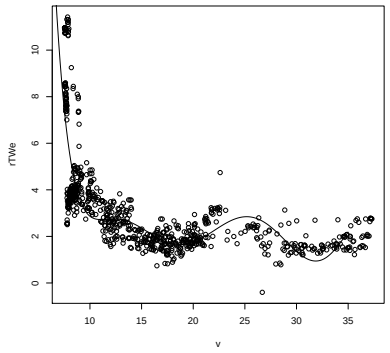
(c) internal twist reference in deg



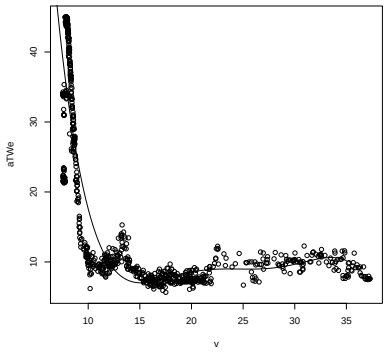
(d) internal twist amplitude in deg



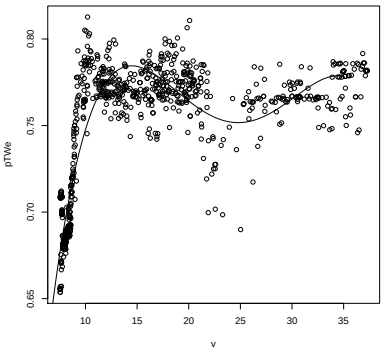
(e) internal twist phase



(f) external twist reference in deg



(g) external twist amplitude in deg



(h) external twist phase

Fig. 2. 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 5.

then. Results are reported on figure 4. What we observe is that the bird actually changes its speed dynamically, simply as a result of wing kinematics parameter change (open-loop modification). The control is certainly not perfect, but we show here that the adaptation is possible with a simple open-loop scheme. 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 $11m.s^{-1}$; here, the 'starting' speed may be very different. Anyway the bird remains stable and at a relatively constant altitude (remember that there is no active control at all: none on speed but none on altitude also).

V. DISCUSSION

The main objective of this work is to get information from a set of Pareto optimal solutions. A first question is : can we get the Pareto-optimal points with other methods ? Of course yes, but MOEA are certainly a good alternative to do that. In many simple cases, an exhaustive search might do

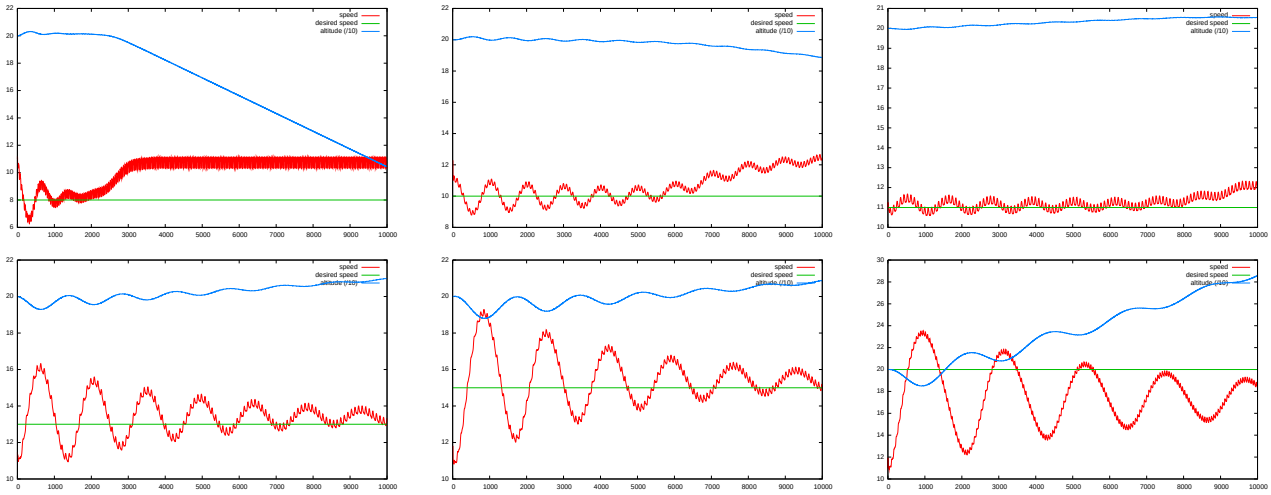


Fig. 3. Speed of the simulated bird controlled by a sinusoidal kinematics whose parameters are obtained from the regression at a given desired speed. The controller is an open-loop controller, real speed is not taken into account to finely tune it.

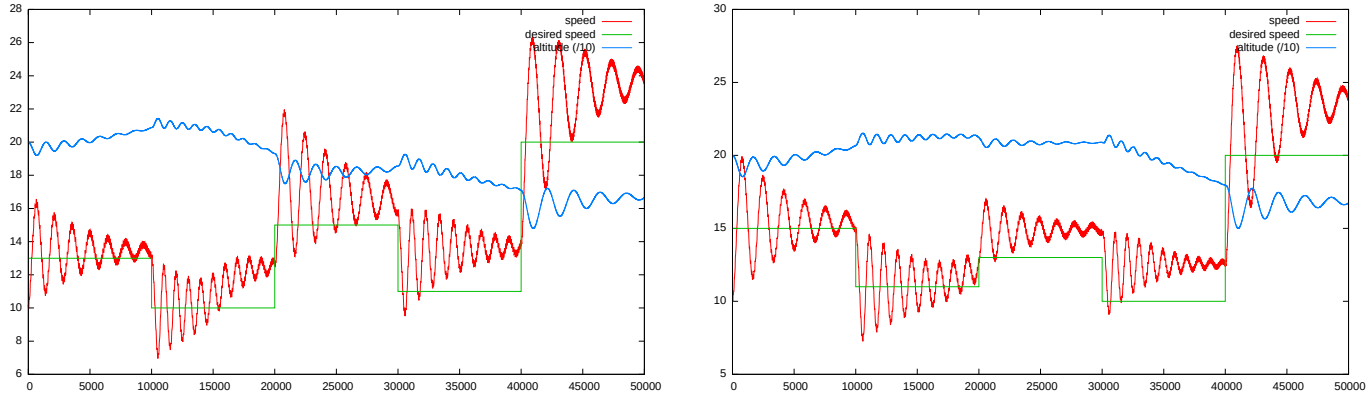


Fig. 4. Speed of the simulated bird controlled by a sinusoidal kinematics whose parameters are obtained from the regression at a given desired speed. The controller is an open-loop controller, real speed is not taken into account to finely tune it. The parameter of the wing kinematics are changed during flight.

the job. Here, it would be difficult to use 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.

All the search was performed on the basis of a simplified simulation. The conclusions are then only relative 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 [16], [17].

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.

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 is 'just' 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.

VI. 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 considering the morphology. Each parameter of the sinusoidal wing kinematics has 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.

VII. 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

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] D. Dasgupta and M. Z., Eds., *Evolutionary Algorithms in Engineering Applications*. Springer-Verlag Berlin and Heidelberg GmbH and Co. K, 1997.

[2] K. Deb, *Multi-objective optimization using evolutionay algorithms*. Wiley, 2001.

[3] C. Coello, G. Lamont, and D. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. Springer-Verlag New York Inc, 2007.

[4] K. Tan, E. Khor, and T. Lee, *Multiobjective evolutionary algorithms and applications*. Springer, 2005.

[5] C. Coello-Coello and G. Lamont, Eds., *Applications of Multi-Objective Evolutionary Algorithms*. World Scientific, 2004.

[6] K. Deb and A. Srinivasan, "INNOVIZATION: Discovery of Innovative Design Principles Through Multiobjective Evolutionary Optimization," *Multiobjective Problem Solving from Nature: From Concepts to Applications*, p. 243, 2007.

[7] H. Tennekes, *The simple science of ight (from insects to jumbo jets)*. MIT Press, 1996.

[8] B. Tobalske and K. Dial, "Flight kinematics of black-billed magpies and pigeons over a wide range of speeds," *J. Exp. Biol.*, vol. 199, no. 2, pp. 263–280, 1996.

[9] K. Park, M. Rosen, and A. Hedenstrom, "Flight kinematics of the barn swallow (hirundo rustica) over a wide range of speeds in a wind tunnel," *J. Exp. Biol.*, vol. 204, no. 15, pp. 2741–2750, 2001.

[10] T. Hubel and C. Tropea, "Experimental investigation of a apping wing model," *Experiments in Fluids*, vol. 46, pp. 945–961, 2009.

[11] T. Rakotomamonjy, "Modelisation et controle du vol d'un microdrone a ailes battantes," Ph.D. dissertation, Universite Paul Cezanne, 2006.

[12] S. Thomson, C. Mattson, M. Colton, S. P. Harston, D. Carlson, and M. Culter, "Experiment-based optimizatin of apping wings kinematics," in *Proceedings of the 47th Aerospace sciences meeting*, ???

[13] J.-B. Mouret, S. Doncieux, and J.-A. Meyer, "Incremental evolution of target-following neuro-controllers for flapp ing-wing animats," in *From Animals to Animats: Proceedings of the 9th International Con ference on the Simulation of Adaptive Behavior (SAB)*, S. Nolfi, G. Baldassare, R. Calabretta, J. Hallam, D. Marocco, J.-A. Meyer, O. Miglino, and D. Parisi, Eds., Rome, Italy, 2006, pp. 606–618.

[14] E. de Margerie, J.-B. Mouret, S. Doncieux, and J.-A. Meyer, "Artificial evolution of the morphology and kinematics in a flapping-wing mini UAV," *Bioinspir. Biomim.*, vol. 2, pp. 65–82, 2007. [Online]. Available: http://stacks.iop.org/1748-3190/2/65

[15] T. Druot, "Technical report on the implementation and validation of a flight mechanics simulator for flapping articulated wings," Available at http://animatlab.lip6.fr, Tech. Rep., 2004.

[16] J. Bongard and H. Lipson, "Nonlinear System Identification Using Coevolution of Models and Tests," *Evolutionary Computation, IEEE Transactions on*, vol. 9, no. 4, pp. 361–384, 2005.

[17] S. Koos, J. Mouret, and S. Doncieux, "Automatic system identication based on coevolution of models and tests," in *Proceedings of IEEE Conference on Evolutionary Computation*, 2009.

param	v^0	v^1	v^2	v^3	v^4	v^5	v^6	v^7	v^8	v^9
a_{DI}	5.83e3	-2.85e3	5.98e2	-7.03e1	5.12e0	-2.40e-1	7.25e-3	-1.36e-4	1.46e-6	-6.73e-9
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. 5. Parameters of the polynomials approximating the relations $parameter = f(v)$ for each parameter of wing kinematics submitted to optimization.