Feathered Flyer: Integrating Morphological Computation and Sensory Reflexes into a Physically Simulated Flapping-Wing Robot for Robust Flight Manoeuvre

YoonSik Shim and Phil Husbands

Centre for Computational Neuroscience and Robotics
University of Sussex, BN1 9QH, UK
Y.S.Shim@sussex.ac.uk &P.Husbands@sussex.ac.uk

Abstract. A new biologically inspired approach to a flapping wing controller which benefits from morphological computation and a Reflexive Pattern Generator (RPG) was tested using a simple physically simulated 3D flying robot. In order to tackle the difficulty of generating robust flapping flight and its manoeuvre, the robot employs simplified flexible "feathers" which are modelled as a series of subpanels attached to the wing skeleton using nonlinear angular springs. The neural controller receives sensory inputs from each feather to let them participate in pattern generation, the robot can also "feel" aerodynamic forces on its wings. From the synergy of flexible feathers and their sensory reflexes, the evolved robot exhibited flight manoeuvre using asymmetric wing movements as well as its tail, and rapidly adapted to external disturbances even in the absence of visual sensors. The reduced stiffness in flight control arising from the wing flexibility is discussed.

1 Introduction

Over 330 million years of evolution, nature developed diverse aerial creatures from insects to pterosaurs. Over the history of animal flight, birds are probably the most advanced flying species among higher vertebrates. Unlike insect flights, where the stroke angle and high frequency of wingbeats are nearly constant in time, birds must perform continuous adjustments to their articulated arms during a single stroke due to their comparatively low wingbeat frequencies [1]. Comparatively few studies have focused on the automatic generation of lower-frequency flapping flight found in species other than insects, such as birds or pterosaurs. These works successfully generated flight locomotion using various controllers from hand designed parametric functions to artificial neural networks or hybrids of these [2,3,4,5]. However, it turned out that the most challenging feature of an ornithopter is its stable and rapid aerial manoeuvering, and generating such behaviours in artificial systems still remains an open problem. A recent study [6], using staged multiobjective evolution, developed the turning flight of a simulated bird robot using nonlinear oscillators and neural networks, employing symmetric wingbeats in altitude control and a tail for the steering flight. However, the stabilisation and flight manoeuvre of a real bird is achieved mainly by lateral aerodynamic moments which are produced by the asymmetric movement of the primary wings. At most flight speeds the effect of a bird's tail is minimal, especially in flapping flight. The tail is rarely spread at higher speeds and is held spread and depressed without active change even at low speeds [7]. This paper presents a new approach to flapping flight control that addresses some of these issues.

In order to achieve a successful compromise between the contradictory properties of stability and manoeuverability of flapping flight control, the work described here employs two bio-inspired concepts: morphological computation [8] and mechanosensory reflexes which are embodied as Reflexive Pattern Generators [9]. These features are effectively merged into a simple and tractable robot model using a flexible wing composed of a series of sub-panels which have mechanosensors connected to the control system. To our knowledge, this is the first successful demonstration of the generation of robust lateral flight manoeuvre by asymmetric wingbeats using neural networks developed in an evolutionary framework. The rest of this paper describes the details of the robot model and its neural controller, followed by a demonstration of the result and ending with some discussion of issues arising.

2 The Robot

The robot comprises two single-armed wings which each have three degrees of freedom and a tail with two degrees of freedom. A wing is composed of four feathers which are attached to its skeleton using hinge joints with nonlinear angular springs. The Open Dynamics Engine, an open-source simulator by Russell Smith [10], was used to simulate the articulated rigid body dynamics.

A feathered model - which introduces a degree of 'morphological computation' [8] - has multiple advantages which are now briefly described . First, the flexible feathers acts as an "aerodynamic cushion" in that the feathers reduce the stiffness of motor control. For instance, in completely rigid wings, a small difference in stroke force between two wings will result in a drastic change of net aerodynamic force and moment on the body. Conversely, in a flexible wing, the change will be small due to the passive bending of each feather. In a certain range of motor output, there may exists a small region of identical net lift or net drag and this helps to reduce the control stiffness, thus leading to robustness against external perturbations as well.

Second, in real bird flight, feathers can hold laminar flow of air though a wide range of angle of incidence of a wing arm without stalling [11]. This is also the case in our simplified model, which has a wider range of effective stroke angles of the wings thus producing more lift than that of rigid wings.

Another important advantage of using feathered wings is that a robot can sense the aerodynamic force on each wing by feather bending. This is motivated by the way birds sense airflow using their feathers. In the biological literature,

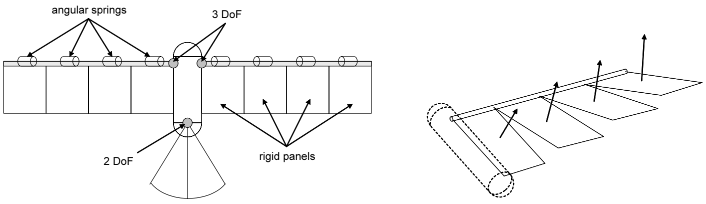


Fig. 1. The robot model

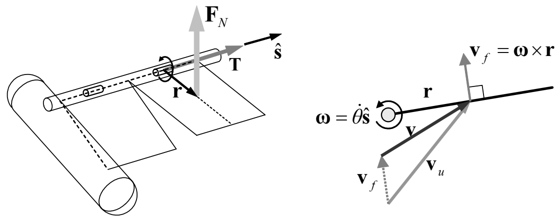


Fig. 2. Force and torque on a feather. $\mathbf{r}$ is feather's longitudinal vector from the wing skeleton to its center, $\mathbf{v}_u$ is the incoming air velocity projected to the surface perpendicular to $\hat{\mathbf{s}}$. The actual airstream $\mathbf{v}$ is obtained by subtracting feather velocity ($\mathbf{v}_f$) from $\mathbf{v}_u$.

a number of researchers (e.g. [12]) suggest that the airflow directed at feathers on various parts of birds influences their locomotor behaviour, in a similar way to the hairs and antennae of insects. Although a feather itself has no inherent sensing organ, its follicle is surrounded by a variety of mechanoreceptors with various response characteristics induced by its bending and vibration [13]. [14] measured and analysed the signals from feather mechanoreceptors and suggests that the feather sensor mechanism detects momentary stall and the location of flow separation on the wing as well as sensing the speed of airflow from the vibration frequency of secondary wing feathers.

This work utilises feather sensing in a way that is analogous to the use of touch or pressure sensors on legged robots to deal with uneven terrain and external perturbation. In the same way, locomotion in a fluid medium can also exploit pressure sensors for uncertain environmental influences such as turbulence. Supplementary to the gradual descending command from the optic and vestibular systems, in a bird agile sensory reflexes from oscillatory feathers can be effectively entrained to the pattern generation of wingbeats and play a crucial role in active stabilisation.

The total mass of the robot is 4.1 Kg, the wing span is 2.6 m, the aspect ratio is 5.8, and the wing loading is $0.35\,\mathrm{g/cm}^2$ (Fig. 1). Each axis of a wing joint represents its dihedral, sweep, and twist. The joint range of each DoF is $\pm 45\,°$ for dihedral (tilted upward by $22.5\,°$ to increase the static stability [15]) and $\pm 22.5\,°$ for sweep and twist. The tail joint controls tail bend and tail twist each of which has a joint range of $\pm 22.5\,°$, and the tail spread is fixed at $45\,°$. Each rectangular feather is subject to aerodynamic forces on its center of mass. Blade element theory with a quasi-steady assumption [16] was used as a simplified aerodynamics. Additional details can be found in [4,5].

A nonlinear angular spring for feather bending is simulated using a first order differential equation so that the bend angle smoothly decays toward the equilibrium position between aerodynamic torque and the spring torque. At each time step, the bend angle rate of the i th feather receiving aerodynamic torque T (=$|\mathbf{T}|$ in Fig. 2) is described by:

$$\dot{\theta}_i(t) = P(T_i(t-1) - k\theta_i(t)) \tag{1}$$

where P is a proportional factor and k is the spring constant. We set $P = 100$ and $k = 0.1$. The bending torque is calculated from the net aerodynamic force exerted on the center of mass of the feather. The new bend angle and its angular speed from previous bending torque are used again as the parts of a parameter set for calculating new aerodynamic forces and bending torque in a circular manner. This simplified heuristic model effectively simulates a completely damped angular spring for a feather of negligible mass without further oscillation.

3 The Controller

A bilaterally symmetric continuous-time recurrent neural network (CTRNN) [17] was used for a controller with a hyperbolic tangent function as an activation function. The state of a single neuron is computed by the following equation:

$$\tau_i \frac{dy_i}{dt} = -y_i + \sum_{j=1}^{N} \{w_{ji}\sigma(y_j + b_j)\} + I_i, \quad \sigma(x) = \tanh(x) \tag{2}$$

Where y_i is the state of i th neuron, τ_i is a time constant, w_{ji} is an incoming connection from the neuron j, b_j is the bias, and I_i is the external input. Connection weights and biases were set in the range [-4, 4] and time constants were set in the range [0.01, 1.0]. The sensor biases were from the range [-1, 1] and their time constants were fixed to 0.01. By using tanh as an activation function the symmetric bias range is distributed naturally around the center-crossing point [18] without further processing. Every parameter was encoded as a real value in the range [-1, 1]. Time constants were actually mapped into [-2, 0] linearly then rescaled by 10^x. The circuit was integrated using the forward Euler method with a step size of 0.01. The output signals from motorneurons are fed to the simulated servomotors as desired angular positions. The average signal of the two neurons at the left and right side ((R+L)/2) was used for tail bend, and the difference between the outputs ((R-L)/2) was used for tail twist. Fig. 3 shows the network architecture used in this work. Identical fully connected 8-node neural networks were used for each side of the robot. Every neuron receives inputs from all sensors. Where the two local axes of fuselage are given at each time step (side: $\mathbf{s}$, forward: $\mathbf{f}$), the pitch angle was calculated by the angle between global

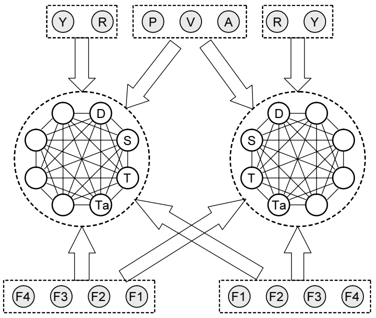


Fig. 3. The neural network architecture. All neurons and connections are bilaterally symmetric. Each lettered circle represents motor neurons (D: dihedral, S: sweep, T: twist, Ta: tail) and sensors (P: pitch, V: forward speed, A: altitude, R: roll, Y: yaw, F1-F4: feather bend rate). The neurons without letters are interneurons. Large arrows between subgroups enclosed by dashed lines indicate that every neuron in the target group receives connection from all sensors in the corresponding subgroups.

Y axis and **f**, the yaw angle was obtained from the azimuth angle between the target vector and **f**, and the roll angle was defined as an angle between **s** and $\mathbf{Y}\times\mathbf{f}$. Each attitude sensor value was scaled to [-1,1]. Roll and yaw sensors are connected separately at each side of the circuit in order to give inputs with reversed signs to the left and right. The speed and altitude sensor values were also processed to have the symmetric range. Feather bending rate is used as a sensor value because it is more sensitive to instantaneous aerodynamic forces. Note that there is no interconnection between the 8 node subnetworks to avoid unnecessary antiphase movement of the wings. However, both circuits are still able to communicate with each other indirectly through the physical environment by receiving sensory inputs from the feathers of the opposite wing. Setting the number of neurons to 8 ensures sufficient possibilities for searching two briefly distinct motor circuits which acts as the pattern generating motorneurons and the cascaded neural circuits [19].

4 Evolutionary Search

A geographically distributed genetic algorithm [20] was used, usually with a population of 100 (10×10 grids) individuals. Each boundary of the population grid was connected to its opposite side, forming a torus-shaped manifold in order to give an equal chance of mating to the individuals at the corners of the grid. Genoypes were strings of floats encoding the network connection weights, biases and time constants. The mutation rates of parents were changed adaptively after each mating according to the fitness score of their offspring with a threshold of 70 % of the average score of its parents. The initial mutation rate was 0.5 per genotype and this was slowly altered in steps of 0.001. Crossover was also used; supplementary details are to be found in [4].

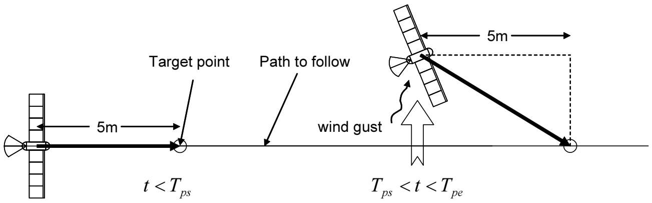


Fig. 4. Fitness evaluation. A wind gust is presented for a certain duration after successful take off. For each evaluation, the strength of gust and perturbation time (T_{ps}, T_{pe}) varies randomly in a small range (see the text).

4.1 Fitness Evaluation

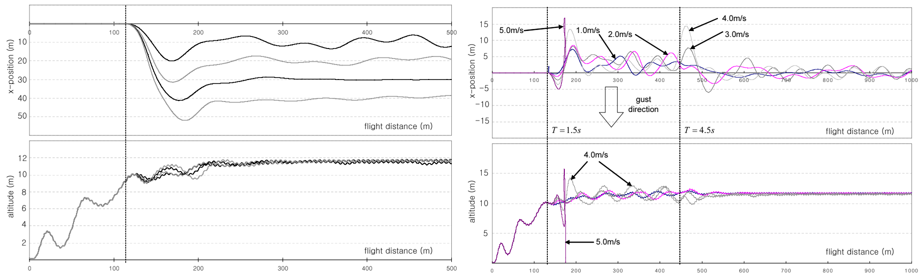
In contrast to the simulation of walking or swimming, the starting strategy for a flapping flight simulation influences the evolutionary search because of the instability of its environment. For instance, when the simulation starts with the flyer in the air, small differences in initial wing stroke between robots due to the slightly different locations of stable limit sets of neural circuit dynamics may cause a variety of consequent motions from gliding to somersaulting. In order to reduce the initial gap, robots were started by being pushed forward on the ground with no friction, and successful take offs survived to be evaluated. Aerodynamic force calculations were turned on after 1sec, and those individuals whose altitude is lower than 50 cm after 5 sec are discarded.

Over the entire flight duration, a robot should retain its flight path which is defined as a straight line of a certain altitude (10 m in this work). The actual target point is located 5m ahead from the robot and moves along the target path line. During flight, the robot receives perturbations from the wind gust for a few seconds as shown in Fig. 4. Before perturbation, the robot only has to reach the target altitude, but after perturbation, the robot should return to the path line as soon as possible and should maintain its flight path continuously. This simple strategy effectively ensures a selection pressure encapsulating several aspects for evolving sophisticated flight control requiring robustness and flight manoeuvre. Because of the highly nonlinear nature of the robot-environment interaction and the tight couplings between the spatial axes for aerodynamic force generation, the robot is forced to capture a variety of capabilities to survive through this simple perturbation strategy. The fitness function is as follows:

$$\text{fitness} = 100\{pf_1 + (1-p)f_2\} \tag{3}$$

$$f_1 = \frac{1}{1 + \frac{1}{T_{flight}} \sum_t^{T_{flight}} d_t}, \quad f_2 = \frac{T_{flight}}{T_{max}} \frac{A}{A_d} \frac{F}{F_d}, \quad p = 0.5 f_2 \tag{4}$$

where T_{flight} is the flight time, T_{max} is the maximum simulation time, d_t is the distance from target path at step t, A and F are average wingbeat amplitudes and frequencies, and A_d and F_d are the maximum desired amplitude and frequency which are set to 0.6 rad and 2 Hz. These force the existence of



(a) (b)

Fig. 5. (a) Path-following trajectory. The robot steers toward target path and slowly stabilises its trajectory. Steering toward four different path line (x=10,20,30,40 m) is depicted. Because of the bilateral network structure, steering behaviours toward both directions are symmetric. (b) Restabilisation after presence of wind gust. The wind is presented for 3 seconds during flight (between vertical dashed lines) and stabilisation against five wind speeds (1-5 m/s) are shown. The robot actively flies upstream and retains its balance during exposure to the wind gusts.

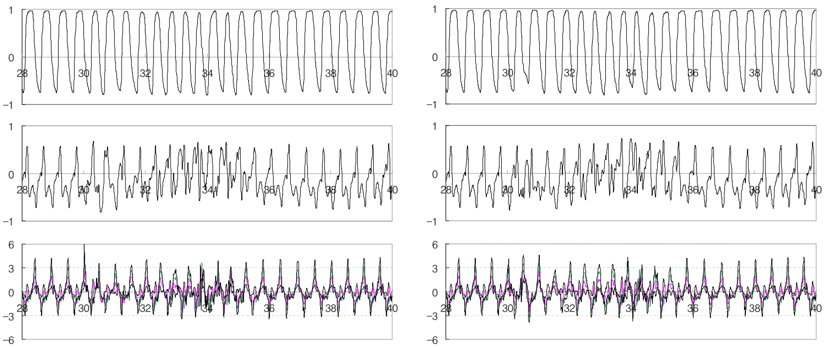
oscillatory wingbeats, and if the amplitude and frequency of a robot exceeds these limits, they are clipped to the maximum values. The fractional factor p varies according to f_2 so that the continuous flight with desired wingbeat amplitude and frequency is predominant over target following. The maximum value of $f_2(=1)$ occurs when the robot remains airborne during the whole evaluation period with sufficient wing oscillation, and the distance from target (f_1) can influence the fitness by up to half the maximum. If 0, the distance from the target does not influence the fitness at all and the evolutionary search concentrates on maintaining flight. This adaptive weighting scheme enables compromise between the two conflict objectives of stability (retaining balance) and manoeuverability (breaking balance to steer).

5 Result and Discussion

The total evaluation time was set long enough (60 s)to fully test the flight behaviours, and a wind gust was presented twice (horizontally and vertically) during the flight. The velocity of the wind and the time of its presentation varied in a small range of random values as shown in Table 1.

Table 1. Perturbation strategy. For each evaluation, the strength of gust and perturbation time (T_{ps}, T_{pe}) varies randomly in a small range (see the text).

gust	Start Time (s)	End Time (s)	V_x (m/s)	V_y
1st	$12\leq T_{ps} \leq 13$	$15\leq T_{pe} \leq 16$	$\pm 2 \leq V_x \leq \pm 4$	$-0.1\leq V_y \leq 0.1$
2nd	$30\leq T_{ps} \leq 31$	$33\leq T_{pe} \leq 34$	$-0.1\leq V_x \leq 0.1$	$\pm 1 \leq V_y \leq \pm 3$



Left Wing

Right Wing

Fig. 6. Motorneuron outputs (first row: dihedral, second: twist) and feather sensor inputs (last row) in blind flight (pitch, yaw and roll sensor were removed). Sweep did not show sensible differences between both wings and is omitted from the figure. At 30 s an angular impulse (3.0 kg·m/s) was exerted in yaw direction. Note that the wing twist is tightly coupled with the feather input signals.

5.1 Steering and Stabilisation

Evolution was typically run for 50000 to 70000 evaluations, and the fitness converged after around 30000 evaluations. The wingbeat frequencies ranged from 1Hz to 2Hz with flight speed of 9–12 m/s. The low flapping frequencies resulted from multiple causes including the time constant of the feather springs and the inaccuracies of the simulated aerodynamics. However, evolved robots showed robust flapping flight and were able to follow the path line without stalling even in the presence of wind gusts (Fig. 5). A few seconds after the start of a perturbation, the robot was pushed out by the wind then tried to retain its orientation toward some angle between the target and the upstream direction. This upstream flight was induced passively due to the forward-swept wing flapping. Because of the aerodynamic force on a wing due to the incoming air stream which is perpendicular to its leading edge, the outside wing from the wind source generates more lift and thrust, hence the robot turns toward the upstream direction. Successful robots used every degree of freedom to manoeuvre including its tail. The most sensitive response was seen in wing twist, and the smallest change was in wing sweep. Feather sensors participated in the controller circuits and properly coordinated pattern generation. As shown in Fig. 6, even in the absence of visual sensors, the robot showed surprising robustness in its locomotion without falling. If the feather sensor was removed, the controller still generated oscillatory outputs of small uncoordinated amplitude but was unable to fly. Therefore the evolved neural controller can be thought as a pseudo-MPG [21] because it does not have locomotor capacity without sensory feedbacks nor does it show pure reflex chain. Pure RPGs were also evolved in some robots but their performance was lower than pseudo-MPGs. It seems because the shape of feather signals are

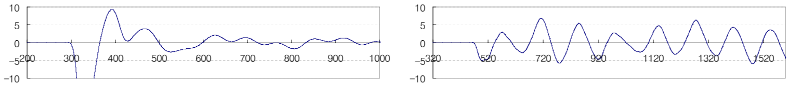


Fig. 7. Stabilisation trajectories of feathered (left) and non-feathered (right) robots after a perturbation. The robots received an angular impact of 3 Kg·m·s.

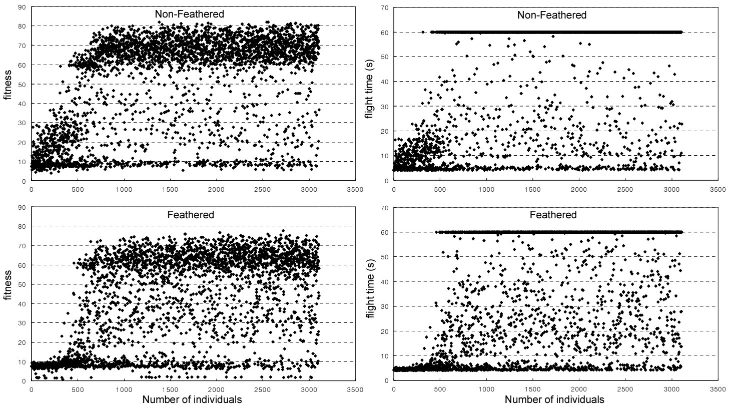


Fig. 8. Every individual whose flight time is longer than 4 s was plotted over 3500 evaluations. Upper graphs are the fitness and the airborne time of non-feathered robots, and the lower graphs are those of feathered robots. We can see the feathered model has more individuals in the middle range, which means that the feathered wing has higher evolutionary potential and more diverse choice in motor control than non-feathered.

relatively sharp with small high-frequency peaks, it is likely they cause the jerky wing motions found in certain situations by intensive sensor dependancy.

5.2 Feather vs No Feather: Reduced Stiffness in Motor Control

It is well known that a flexible wing is beneficial in producing a better lift-drag ratio both in fixed and flapping wing flight. However, there are several difficulties in directly comparing general flight performances and stabilities of an actively-controlled flight between flexible and rigid wings. It is because these properties cannot be considered separately from the other crucial characteristics of flight such as the wingbeat kinematics or flight speed [22]. Nonetheless, Fig. 7 shows clear difference in stabilising behaviours between the two species.

In addition, a qualitative observation was made to examine the difference in the evolutionary process in these two cases. The robots were evolved only for a straight, levelled flight at a certain altitude, and the diversity of the population was observed in each case. Fig. 8. shows the plots of all individuals which succeeded in take-off (flew higher than 50 cm after 4 s). The feathered model has a significantly thicker group of middle ranged fitnesses which implies that exactness in motor control for successful flight is less necessary (lower control stiffness) than for the rigid wing. In addition, a simple quantitative measurement of the diversity (randomness) of chromosome vectors of those individuals were performed using a Nearest Neighbour Distance method [23]:

$$D_{NN} = \sum_{i=1}^{N} (\frac{Min(d_{ij})}{N}) \quad i \neq j, \quad j = 1, ..., N \tag{5}$$

where D_{NN} is the mean nearest neighbour distance and $Min(d_{ij})$ is the nearest distance between vector i and others. The mean distance of the feathered group was 0.422 and the non-feathered group was 0.407, which means the feathered model has a wider variety of genotypes for successful take-off. Future work will further explore the significance of these differences.

6 Limitations and Future Work

Although the proposed model shows robust aerial locomotion to a certain extent, it is rather conceptual and requires substantial improvement in order to bridge the reality gap. Firstly, the simplified aerodynamics does not reflect the possible flow separation at the cleft between feathers. These inaccuracies could be reduced by using a single-paneled flexible wing or a series of partially overlapped elastic feathers as on a real bird wing. With the proper redefinition of mechanosensory information which might be changed by the modification of the mechanical model, the evolutionary framework will still be able to find the appropriate controllers for efficient flight. Secondly, additional adaptation processes could be employed for transferring the model to reality. Careful analysis of the effect of reduced control stiffness will enable us to draw the contours of valid regime over which the adaptability of controllers takes effect in the real environment. Also, although the model had less degrees of freedom than those of real birds, we believe this work can be extended effectively to more realistic models. Some movie clips are available at http://www.informatics.sussex.ac.uk/users/yss20/feathered.html.

References

1. Clarke, F., Ekeland, I.: Progress of the REMANTA project on MAV with flapping wings and of the International Universities mini UAV Competition. European Micro Air Vehicle Conference and Flight Competition (EMAV). (2006) Braunschweig, Germany.
2. Wu, X.C., Popović, Z.: Realistic modeling of bird flight animations. The 30th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH) San Diego, CA. (2003) 888–895
3. Augustsson, P., Wolff, K., Nordin, P.: Creation of a learning, flying robot by means of evolution. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO) Morgan Kaufmann, New York (2002) 1279–1285

4. Shim, Y.S., Kim, C.H.: Evolving Physically Simulated Flying Creatures for Efficient Cruising. Artificial Life. **12** (2006) 561–591
5. Shim, Y.S., Kim, S.J., Kim, C.H.: Evolving Flying Creatures with Path-following Behavior. The 9th International Conference on the Simulation and Synthesis of Living Systems (ALIFE IX) Boston, MA. (2004) 125–132
6. Mouret, J.B., Doncieux, S., Meyer, J.A.: Incremental Evolution of Target-following Neuro-controllers for Flapping-wing Animats. From Animals to Animats: Proceedings of the 9th International Conference on the Simulation of Adaptive Behavior (SAB) Rome, Italy (2006) 606–618
7. Warrick, D.R., Bundle, M.W., Dial, K.P.: Bird Maneuvering Flight: Blurred Bodies, Clear Heads. Integ. And Comp. Biol. **41** (2002) 141–148
8. Pfeifer, R., Iida, F.: Morphological Computation: Connecting Body, Brain and Environment. Japanese Scientific Monthly, (2005) **58** 48–54
9. Gallagher, J.C.: Evolution and Analysis of Non-Autonomous Neural Networks for Walking: Reflexive Pattern Generators. Congress on Evolutionary Computation, Seoul, Korea (2001)
10. Smith, R.: Intelligent Motion Control with an Artificial Cerebellum. Unpublished doctoral dissertation, University of Auckland, New Zealand. (1998) (ODE engine available at http://ode.org)
11. Burtt Jr., E.H., Ichida, J.M.: Selection for Feather Structure. Acta Zoologica Sinica, (2006) **52** 131–135
12. Gewecke, M., Woike, M.: Breast Feathers as an Air-current Sense Organ for the Control of flight Behaviour in a Songbird (Carduelis Spinus). Z. Tierpsychol., (1978) **47** 293–298
13. Necker, R.: Somatosensory System. In Physiology and Behavior of the Pigeon, Academic Press, London (1983) 169–192
14. Brown, R.E., Fedde, M.R.: Airflow Sensors in the Avian Wing. J. Exp. Biol., (1993) **179** 13–30
15. Thomas, A.L.R., Taylor, G.K.: Animal Flight Dynamics I. Stability in Gliding Flight. J. theor. Biol., (2001) **212** 399–424
16. Weis-Fogh, T., Jensen, M.: Biology and physics of locust flight I: Basic principles in insect flight. A critical review. Philosophical Transactions of the Royal Society of London B, (1956) **239** 415–458
17. Beer, R.D.: On the Dynamics of Small Continuous-Time Recurrent Neural Networks. Adaptive Behavior, (1995) **3** 471–511
18. Mathayomchan, B., Beer, R.D.: Center-crossing Recurrent Neural Networks for the Evolution of Rhythmic Behavior. Neural Computation, (2002) **14** 2043–2051
19. Haschke, R., Steil, J.J.: Input Space Bifurcation Manifolds of Recurrent Neural Networks. Neurocomputing, (2005) **64C** 25–38
20. Husbands, P.: Distributed Coevolutionary Genetic Algorithms for Multi-criteria and Multi-constraint Optimisation. Evolutionary Computing, AISB Workshop Selected Papers, Vol. 865. (LNCS) Springer-Verlag (1994) 150–165
21. Gallagher, J.C.: Evolution and Analysis of Mixed Mode Neural Networks for Walking: Mixed Pattern Generators. Evolutionary Computation, Proceedings of the 2001 Congress, Vol. 1. (2001) 397–402
22. Taylor, G.K., Thomas, A.L.R.: Animal Flight Dynamics II. Longitudinal Stability in Flapping Flight. J. theor. Biol., (2002) **214** 351–370
23. Clark, P.J., Evans, F.C.: Distance to Nearest Neighbor as a Measure of Spatial Relationship in Populations. Ecology, (1954) **34** 445–453