SwarMAV: A Swarm of Miniature Aerial Vehicles.

Renzo De Nardi*, Owen Holland*, John Woods, and Adrian Clark****

***Department of Computer Science **Department of Electronic Systems Engineering**
University of Essex
Wivenhoe Park
CO4 3SQ

(rdenar/owen/woodjt/alien@essex.ac.uk)

ABSTRACT

As the MAV (Micro or Miniature Aerial Vehicles) field matures, we expect to see that the platform's degree of autonomy, the information exchange, and the coordination with other manned and unmanned actors, will become at least as crucial as its aerodynamic design. The project described in this paper explores some aspects of a particularly exciting possible avenue of development: an autonomous swarm of MAVs able to exploit its inherent reliability (through redundancy), and its ability to exchange information among the members, in order to cope with a dynamically changing environment and achieve its mission. We describe the successful development of a rotorcraft-based prototype experimental platform weighing only 75g, and outline a strategy for the automatic design of a suitable controller.

BIOGRAPHIES

Renzo De Nardi is a PhD student in the Computer Science Department at Essex University. He received his Laurea in telecommunications engineering from the University of Padua in 2004. His current areas of interest include swarm intelligence, biologically inspired robotics and vehicular dynamics. *Owen Holland* is currently Professor of Computer Science at the University of Essex. Much of his research in the last 12 years has dealt with swarm intelligence and collective robotics; as well as UAVs, his current interests include humanoid robotics, machine consciousness, and particle swarm optimisation. *Dr. John Woods* is a lecturer in the Department of Electronic Systems Engineering, University of Essex. Although his field of expertise is image processing, he has a wide range of interests including telecommunications, autonomous vehicles and robotics. *Dr. Adrian Clark* is a reader in the Department of Electronic Systems Engineering, University of Essex, where he heads the Virtual Applications, Systems and Environments Laboratory. His principal research interests explore the interplay between computer vision and virtual reality, including hybrid vision/VR systems and human-computer interaction.

1. Introduction

The events of the last decade have produced an increasing focus on the application requirements for conventional UAVs. However, the burgeoning field of MAVs is still mainly focused on the demanding technical requirements for the individual aerial platform and its sensing capabilities. The dominant mission scenario is still limited to a single MAV controlled by a single minimally skilled operator.

What if, instead, we could take advantage of a group of MAVs cooperating to solve a task? The scenario suddenly looks very different [10]. Coordination and cooperation become the main problems to solve, but previously unthinkable possibilities are now open. Looking at a target from different viewpoints simultaneously, or quickly surveying a large area, suddenly become feasible tasks. The resulting system may be inherently more robust because of the potential redundancy of individuals, while the small size of the MAVs makes the whole group stealthy. The SwarMAV project currently under development at the University of Essex aims to provide a proof of concept system by building an indoor “flock” of MAVs.

The project combines two key ideas: using biologically inspired rules of group behaviour (flocking) to enable a group of UAVs to control its own motion; and wirelessly networking the swarm members together to form a single powerful computing resource – perhaps ultimately a kind of grid computer – to enable in situ processing of collectively gathered data.

The basic concepts required for such a system are described in the next section, and the enabling technologies are discussed in Section 3. Section 4 deals with the early implementation, and Section 5 summarises the results of the first experiments. The conclusions to date are presented in Section 6.

2 Basic concepts

2.1 Flocking

The term 'swarm intelligence' is generally used to refer to the emergence of coherent functions governed by low level interaction and/or communication between large numbers of individuals. This is clearly a very broad definition – perhaps too broad to be useful – and so when the interaction among group members leads to coordinated movement, the more specific term 'flocking' is preferred. The reference derives from flocks of birds, in which the individuals move with approximately the same velocity, and at a roughly similar inter-agent distance. The advantages of the flocking paradigm are clear: coordination is achieved without any high level centralized control, and is totally independent of group size. Flocking therefore embeds the very important properties of distributedness and scalability.

Reynolds [1] demonstrated that a few simple behavioural rules applied locally by each individual can actually lead to lifelike flocking of groups of simulated agents. Rules governing cohesion, separation and alignment, and modulated only by the relative positions of nearby flockmates, appear to be sufficient to guarantee flocking. At the same time, methods are available to enable the high-level external guidance of a flock while retaining the attractive low-level core functionality.

The seminal work of Reynolds stimulated a series of investigations in many different scientific fields, notably biology, computer science, physics, and most control system engineering, the latter being a particularly active area at present. Strategies for coordinated control based on behavioural rules have been proposed by several control system engineers; by using well-defined potential functions, the convergence and stability of a number of flocking control algorithms has now been demonstrated (Jadbabaie [8], Olfati-Saber [9]). Those results have opened the way for practical designs soundly based on control system engineering techniques.

In the robotics field several experimental results related to flocking have been obtained using wheeled robots (e.g. Kelly et al. [2], Mataric [3] and Regimi et al. [4]). In such work, a leader, virtual or explicit, is used to coordinate a degenerate flock-like movement; it is notable that none of these schemes has demonstrated a fluidity of motion similar to a group of real birds, though the reason for this is not clear. For obvious practical reasons, less work has been done using aerial platforms; typical examples are the minimalist approach undertaken at UWE using small blimps (Welsby et al. [5]), and a research program carried out by NASA [6]. The first of these suffered from the limitations of the chosen platform, and produced only a basic aggregation behaviour. The second example, from NASA Dryden Flight Research Center [6] demonstrated the coordination of two UAVs (two instrumented model aircraft) using Reynolds' flocking rules. GPS information was used to determine the relative position of the aircraft, and this seems to have been sufficient to guarantee coordination. Unfortunately further details about the project are not yet in the public domain. A rather different approach was taken by Crowther and Riviere [7]: instead of using real vehicles, they carried out detailed simulations using aerodynamic data from wind tunnel tests on a propeller driven UAV. They showed that flocking could be produced using only cohesion and alignment, but noted that the nature of the UAV flight control system was a significant practical constraint on implementation.

It is clear that, in principle, flocking represents a potentially useful solution for formation control, but a lot of ground still has to be covered, especially from the experimental point of view.

2.2 Cluster Computing

Collaboration among members of a group can be more extensive than simple coordinated motion. While physical and biological limitations may constrain the amount of information exchanged by members of a natural group, these limitations need not apply to an artificial group. There are several ways in which a system can be designed to link together and exploit the computational resources of all of the members. In particular, by exploiting high bandwidth wireless connections, it may be possible in principle to form a single powerful computer, more properly called a MCC (Mobile Cluster Computer). An early paper on this theme (Zheng et al. [13]), identified the main issues in this field, but surprisingly little work has been done since then. Important issues such as time-variant network topology, variable delays, and communication errors need to be addressed in order to achieve an understanding of the constraints on, and abilities of, such systems.

At the present time the primary challenge we face in this project is the engineering design of the aerial platform. Our initial aim is to produce a proof of concept flocking system possessing the resources that will allow the later implementation of a MCC. Since this will inevitably involve a computational platform with limited resources, we will not attempt to produce a grid-like system in this phase, but will use a conventional distributed computation approach instead to support and demonstrate fluid flocking. In the meantime, it is likely that the necessary conceptual and technological developments to enable a practical MCC will occur, ready to be adapted for the final phase of this project.

3 Enabling technologies

3.1 Aerial platforms

The UltraSwarm concept had its roots several years ago in a multi-robot system designed for the Microsystems Laboratory at the California Institute of Technology [10], but it is only recently that suitable technologies have become available for a full implementation. After early experimentation conducted at the University of Essex using large outdoor model aircraft [10], it became clear that for reasons of safety and controllability an indoor system would be preferable, at least in the early stages. Under the constraints imposed by using an indoor location, the possible choices for the aerial platform reduce to three: very light planes (i.e. slow flyers), lighter than air devices (i.e. blimps), and rotary wing aircraft.

Slow flyers are suitable for indoor flight, and have recently been used for research in vision based autonomous flight (Zufferey [12]). However, given their limited turning radius, and the need to maintain a minimum airspeed, a flock of such vehicles would need considerably more space than is offered by our facilities. (Our arena is cylindrical, with a diameter of 12m and a height of 6m.)

Blimps or similar airships have been used in several research programmes (Zufferey [12[; Welsby [2]), but they impose the fundamental and serious limitation of a very small payload. A helium blimp able to lift the sensors and computational power needed for our project would be quite large; a whole group of similar vehicles is unfortunately not compatible with our test arena. There are some additional drawbacks, in particular the slow dynamics, and the impossibility of having precise control over the motion of a blimp due to the dominating influence of air currents.

The development of small rotary wing aircraft suitable for indoor use has traditionally been limited by power problems. Fortunately, recent advances in battery technology – in particular the use of lithium polymer materials – have driven the emergence of new rotary wing designs, especially in the toy and model markets. These range from conventional single rotor helicopters, through a variety of coaxial helicopter types, and even some unusual four rotor configurations. Rotary wing models with a rotor size of about 35 cm and a gross weight of about 180g (e.g. the Hirobo XRB series [11]) easily fulfil our requirements in terms of size, payload, and endurance. The demands of the marketplace also mean that these models are cheap, and very robust.

Some additional advantages offered by the rotary wing solution are the ability to manoeuvre in three dimensions (very helpful for keeping position in a flock), and the ability to hover (very useful when there is a need to stay in place, e.g. for a surveillance task).

3.2 Miniature electronics.

The second substantial component in our system is the onboard electronics which will perform sensing and computation. Miniaturization, low weight, and low power consumption are fundamental requirements for this part of the system.

Several types of miniature computer have appeared lately in the marketplace, combining with various degrees of success the need for good computational capabilities with low power consumption. All the basic components of a computer are provided on a single board capable of running a fully fledged operating system. For our purposes, the best appears to be the Gumstix Basix400 platform, which offers a 400MHz Intel XScale® processor with 64MB of SDRAM and 4MB flash. Its most interesting features are probably its extremely small size (8cm x 2cm x 0.63cm), and the correspondingly low weight of only 8g. The board comes with a pre-installed version of the Linux operating system, which enables fast development of the on-board software.

Sensors are important components of every autonomous vehicle, but in the case of a vehicle with poor stability qualities – such as a typical helicopter – they become crucial. Stability augmentation systems and stability control systems based on inertial measurement have traditionally been used in the aeronautical community, but the size and weight of traditional inertial sensors has limited their use to full size aircraft. However, MEMS technology (Micro-Electro-Mechanical Systems) has recently revolutionized this field, allowing the production of small, light and relatively inexpensive inertial sensors. A complete 6 DoF IMU (3 axial accelerometers and 3 gyros) is now available in a self contained package with a typical volume of a few cubic centimetres, and weighing from 5 to 20 grams.

The accuracy and noise characteristics of the MEMS sensors are poorer than the traditional high-end counterparts, but careful post-processing of the data has been found to overcome these problems in practice. A MEMS based IMU will be used for the stabilisation of our chosen rotorcraft.

Inertial measurements are essential for stability control, but navigation and other high level tasks demand an additional global localization system. A typical solution adopted for outdoor UAVs is the use of GPS, but our need to work indoors precludes its use. Instead, a passive infra-red tracking system installed in our test arena will be used. It is accurate to a millimetre, and can output the coordinates of multiple objects with a latency of the order of 10 milliseconds. Each of the helicopters will be equipped with infra-red markers that will be simultaneously tracked by the system. The absolute position of the vehicles will be processed by the ground system and then relayed by wireless – in the form of relative positional data – to each of the flock members.

Mobile phone technologies have pushed the development not only of efficient batteries, but also of small and light image sensors. Good quality colour single chip cameras with plastic lenses are nowadays available with weights as low as 5 grams. Unfortunately, the current version of the Gumstix computer board does not allow for real time image processing, and so the video stream must be relayed wirelessly to a ground based computer that will perform the necessary computations. The camera information will not be used for the vehicle stabilization since the communication loop cannot guarantee the timeliness and reliability required. However it will be used for high level tasks (e.g. target reconnaissance) where the communication problem can be tolerated. A more powerful version of the Gumstix computer is expected in the first half of 2006, and the capability of onboard real-time image processing will then become a reality.

Although we will not give any priority to the development of the MCC infrastructure, wireless connectivity will be provided between the flock members (and the ground station) to allow for data exchange and distributed computation. One choice for such communication is the 802.11b/g standard which allows for point to point connections at speeds up to 54Mbit/s. A small and light (5g) SDIO wireless LAN module can be interfaced with the Gumstix system, although it substantially increases the power requirements.

An alternative temporary solution is provided by the built-in Bluetooth module present in the Gumstix board. The bandwidth of 723.2 Kbit/s is considerably lower than 802.11, but the low power consumption and the lack of weight penalties make this possibility attractive. The information exchange between the flock members consists essentially of positional data, and the bandwidth allowed by the Bluetooth standard is considered sufficient for this purpose.

4 Implementation

4.1 The helicopter platform

The choice of an appropriate flying machine is the focal point in the initial development of the system. We have already mentioned the positive reasons for the choice of the rotary wing concept, but we have to remember that helicopters are notoriously unstable, and that autonomy is difficult to achieve. For this reason we have given priority to stability, and have evaluated two recent and novel helicopter designs (Fig 1 and Fig 2) characterized by strikingly improved stability.

Figure 1: Proxflyer model retrofitted with new motors, SBC and miniature camera.

Both machines use two counter-rotating rotors, an arrangement in which the torque generated by one of the rotors is compensated by the second one, stabilising the yaw movement of the helicopter fuselage. These designs also have a higher efficiency than a traditional single rotor design; as a result the same power can be generated with a smaller diameter rotor, reducing the size of the vehicle.

The first model evaluated was a Proxflyer (Figure 1). Commercialised as a toy, the Proxflyer has a weight of about 45g and is powered by three electric motors: two motors control the speed of the counter-rotating rotors while a third controls te small upward pointed tail rotor. The rotors have a fixed pitch, so the only available controls are up/down (increasing or decreasing the speed of both rotors), forward/backward (activating the tail rotor to tilt the helicopter in the horizontal plane), and yaw (increasing or decreasing the speed of one of the rotors). Unlike a conventional helicopter, this model does not have any ability to control lateral movement. The augmented stability is achieved by exploiting the gyroscopic forces acting on the rotors, which are enhanced by a peripheral ring. This clever design results in an extremely stable flying machine that can hover hands-off if properly trimmed. However, the low forward speed and lack of lateral control are intrinsic drawbacks of the design.

Figure 2: Hirobo XRB model

The second model evaluated (Figure 2) was the Hirobo XRB SR Lama. With its 195g of gross weight, it is a much more substantial machine than the Proxflyer. Its lower rotor is a fully articulated teetering design controlled by a 45 degree swashplate, while the second rotor is passively controlled through its linkage with the stabilising bar which is mounted at 45 degrees rather than at the more usual right angle. Two servos adjust the cyclic pitch of the first rotor as in a conventional model helicopter. Each of the rotors is powered by a separate motor, and the up/down motion and the yaw control operate on the same principles as the Proxflyer. The forward/backward and the lateral motion are controlled by tilting the lower rotor as in a conventional helicopter. The upper rotor is responsible for the powerful stabilising action, again allowing the vehicle to hover hands off with only a slight drift. It is possible to perturb the helicopter by making large random movements of the cyclic control, but stability is restored very quickly.

4.2 Control system

Model helicopters present several advantages; they are generally cheap, easily available, and extremely simple for obvious economic reasons. They are also robust, to the point where even the hardest of landings usually ends in no more than a smashed set of blades. On the other hand, cost competition ensures that limited attention is paid by the manufacturers to quality standard. Different flying qualities are therefore common even between models of the same type. In addition, the dynamic characteristics of these machines will be strongly dependent on any added payload – the models are not designed to carry any load, but in our experiments we often increase the all-up weight by 50% or more.

From the perspective of the design of a control system, such variability brings problems of both tuning and robustness. The problem gets even worse because in most cases these novel models rely on the clever intuition of a committed designer, rather than the outcome of a structured design process. In particular, no clear dynamic model of these novel helicopters can be expected to be available.

In these circumstances the traditional method of aerodynamic control system design, based on the steps of modelling from first principles, followed by system identification, and the subsequent application of control theory, becomes very complicated. A method for automatically designing the controller to allow for the difference between various helicopters would clearly be advantageous, and we have examined this in some detail.

Various researchers have proposed approaches to this problem based on the techniques of neural networks and fuzzy logic (e.g. Buskey et al. [15]). All these approaches are based on the "teaching by showing" idea: a human operator performs several flight manoeuvres while all the state and control data of the helicopter are recorded. These data are then used as the training set for a neural network or a fuzzy associative map. The weakness of this technique lies in the nature of the recorded data; a human operator is often compensating for inputs that are not directly measured (e.g. wind direction), and the ability of the pilot will also have an influence on the quality of the model.

To cope with this, it is instead preferable to train a controller using a dynamic simulator. An interesting approach for automatically learning vehicular dynamics from flight data is proposed in Abbeel et al. [14]. In contrasting to previously proposed methods, the model is based on the prediction of accelerations, which simplifies the learning of the inertial effects. Special attention is devoted to producing good long term estimates of the vehicle motion. The experimental results demonstrate that a good model can be learned effectively even with a very basic specification of the vehicle dynamics.

By using an accurate model of the vehicle, the few manoeuvres recorded as flight data can be used for training as normal, and in addition any arbitrary manoeuvre can be performed and the outcome calculated. A controller based on an artificial neural network trained using reinforcement learning, in combination with a good dynamic model has proven to be particularly effective (Ng et al [16]).

With the need to develop a slightly different flight controller for each of the group members, we expect to encounter at least some of the problems mentioned above, and we therefore propose to design these controllers automatically.

The infra-red tracking system installed in our experimental site will permit us to gather the necessary data to produce an accurate nonlinear model. In particular, we will be able to estimate correctly the parameter sets of the models associated with specific payload configurations. The ability of the system to track each helicopter with an accuracy of 1mm at a frequency of 100Hz makes us confident of the feasibility of this process.

Once a model has been learned in this way, it will be used to evolve a neural network controller using a population based evolutionary algorithm. A population of initially random controllers is evaluated on their ability to perform a particular pre-specified task by using a simulator. The best performing individuals are selected, and used to produce (through mutation and crossover) a new generation of controllers. Repeating the process for a sufficient number of generations will eventually produce a controller well suited for the task.

Evolutionary methods are interesting for their generality. The designer is required only to supply a fitness function for evaluating each controller's performance. An on-line fitness function is not required since the fitness is evaluated only at the end of the task; this allows more freedom for the evolutionary process to come up with a novel solution. However, in this approach, the choice of the fitness function and the quality of the vehicle model are crucial, because any inaccuracy may provide the basis used by evolution to produce the solution. It should also be noted that, since a large number of controllers must be evaluated, the method is computationally expensive.

5 Experimental results

5.1: Helicopter test and development.

In a first step towards autonomy, the Proxflyer model was fitted with a Gumstix onboard computer and a miniature wireless video camera. After preliminary flight tests it was clear that the amount of lift offered by the original electric motors was far from being sufficient to produce sustained and controllable flight. A radical modification was needed; more powerful (but heavier) motors were therefore retrofitted to the original airframe. This in turn required the fitting of more powerful (and heavier) batteries. A successful compromise between power and weight was eventually reached by using two 300mAh Lithium polymer batteries in series to power the main motors.

After the removal of the original electronics, the Gumstix was mounted in the airframe together with a simple interface board for driving the three motors. The wireless camera sensor was modified to reduce its weight by carefully removing it from its plastic housing and manufacturing a new mount for the pinhole lens.

The typical power consumption of the single board computer was 130mA, with occasional peaks up to 160mA when the Bluetooth module was transmitting. The typical power consumption of the camera (about 40mAh at 6V) brought the total power required by the electronics up to 170-200mAh, a requirement easily met by a single additional 145mAh 3.7V Lithium polymer battery and a voltage doubler circuit. In this configuration the Proxflyer weighed 76.6g – almost double its original weight.

Since the original remote control electronics had been totally removed, new software was written to communicate with the rotorcraft via the Gumstix Bluetooth connection. From a Bluetooth-equipped base computer, UDP datagrams were sent to the onboard computer, where they were translated into motor commands. A GUI on the ground computer enabled the operator to visualize the current status of the helicopter commands. The Proxflyer, modified as described, was able to achieve a maximum flight time of 3 minutes. Even after all the modifications, the machine was still dynamically stable, but the manoeuvrability was greatly reduced despite the careful distribution of the extra weight within the airframe. In particular, the helicopter tended to drift laterally, in a direction over which we did not have control.

The experience with the Proxflyer model can be considered successful, but it was clear that we were pushing the vehicle too close to its limits: the payload was undermining the controllability, and the flight time was really too short. To overcome those problems we decided to use the new Hirobo XRB-SR model (Figure 2).

In a series of initial flight tests this vehicle exhibited an endurance of approximately 15 minutes in continuous flight, with excellent controllability. Even better, the action of the stabilizing bar successfully brought the helicopter back into a stable hover even after arbitrary cyclic pitch inputs generated by the pilot deliberately playing with the cyclic control. After the removal of the canopy and tail (which have only a cosmetic function), and the addition of a payload of 50g, the helicopter flew successfully, being slightly sluggish but still very manoeuvrable. These tests confirmed that this helicopter overcomes all the problems presented by the previous platform, opening the way for rapid development of the whole system.

5.2 Experimental results: control system

We are presently awaiting full installation of the infra-red tracking system, which prevents us as yet from obtaining the flight data from which to derive the helicopter model. However, since the evolutionary computation approach itself is independent of the specific characteristics of the helicopter, we have undertaken a preliminary investigation using Autopilot, a freely available helicopter simulator [17]. This simulator reproduces the dynamics of the XCell 60 model helicopter; the rotor dynamics is based on blade element theory, and the stabilizing bar is modelled as proposed in Mettler et al. [18]. The model is very comprehensive - it accounts for the effect of the canopy and tail, and also for the characteristics of the servos. This target helicopter is clearly more complicated than our chosen vehicle (its simulation is almost 'unflyable' manually), and so it is probably a good test bench for our design approach.

Neural network

The simple network of Figure 3 formed the basis of our initial research. In the network, the nodes compute the *tanh* of the sum of their inputs, and the arrows in the diagram indicate variable weight synaptic connections.

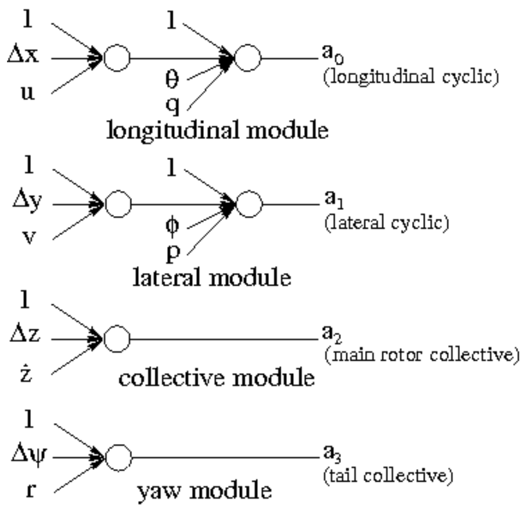

Figure 3: The neural controller.

The structure of the network closely resembles a classical PD controller; with the four separate parts controlling respectively the longitudinal motion (output a_0 = forward cyclic), the lateral motion (output a_1 = lateral cyclic), the collective (a_2) and the yaw (a_3). The network input is made up of a subset of the helicopter state $[u,v,z',\theta,\phi,\psi,p,q,r]$[1], the vector distance $[\Delta x, \Delta y, \Delta z]$ from the next waypoint expressed in the helicopter's frame of reference, and the deviation $\Delta\psi$ from the reference heading. The approach of the helicopter to within a foot of the current waypoint transfers the target point to the next waypoint in the sequence, which produces a change in the inputs $[\Delta x, \Delta y, \Delta z]$.

Evolutionary process

As proposed in section 4.2, an evolutionary algorithm, technically a (20+40) Evolutionary Strategy, is used to determine the connection weights of the neural network controller. The network structure is fixed, and its connection weights are encoded as an array of real numbers. The initial population is made up of 60 networks with small random weight values. Each network is then evaluated on the task, and the whole population is ranked according to fitness. The 20 best performing individuals (the elite) are retained and the remainder are discarded. The new generation is formed by the elite plus 40 new individuals generated from the elite. Each new individual is simply a mutated copy of a randomly selected member of the elite; mutation consists of adding a random value (drawn from a Gaussian distribution with mean 0 and standard deviation 0.01) to each connection weight in the network. Recombination is not used. The procedure is then repeated until some termination criterion is reached.

The automatic design of the controller is divided into two distinct phases: a first phase in which only the weights of the yaw module are evolved, and a second phase during which the weights of the whole network are allowed to evolve.

During the first phase, the yaw module is evolved using a fitness function that simply penalises deviation from the target heading. Each trial lasts for only twenty timesteps (an extremely short time), yet evolution reliably produces a fairly good solution, able to maintain a heading, within tens of generations.

In the second phase, the initial population is generated from the controller produced in the first. All the weights in the network are now free to evolve. An individual's fitness is its score on the waypoint task, implemented as a chain of randomly placed waypoints, with an average distance between adjacent waypoints of 17.5 ft. The fitness achieved is proportional to the number of waypoints reached in a fixed time (P_{chain}); a scaling factor (w_n) is also applied

[1] The state vector follows the conventional notation used in the aircraft control community; u, v, and z' are respectively the velocities in the helicopter's frame of reference, and the altitude derivative in the inertial frame, while θ,ϕ,ψ, and p, q, r are respectively the attitudes and rotational velocities.

to penalize individuals for deviation from the shortest path:

$$f = \frac{\sum_{i=0}^{N} \left(w_n P_{chain} \left| z - z_{next} \right| - \left| \psi - \psi_{ref} \right| \right)}{N}.$$

N is the number of waypoints, and the factors $\left| z - z_{next} \right|$ and $\left| \psi - \psi_{ref} \right|$ are penalties for not maintaining the optimal altitude and heading. The duration of the task is fixed at 2000 timesteps (40s of simulated time). Good solutions are generally produced within the first 500 generations of the evolutionary process.

The trajectory of the best evolved controller, performing a typical waypoint task is shown in Figure 4. Dots mark the position of the helicopter every 10 timesteps. Although the randomly generated waypoints produce some very irregular sequences, the controller performs well, closely following the shortest path. In particular, stability is maintained even after a waypoint switch where the inputs [Δx, Δy, Δz] change abruptly.

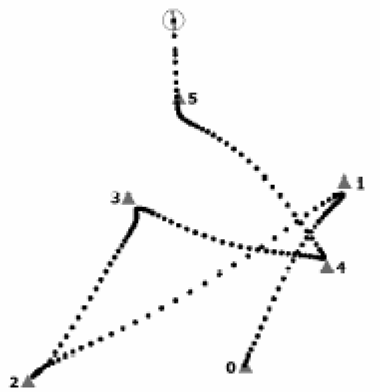

Figure 4: A trace of the helicopter's trajectory after completing 900 (out of 3000) timesteps of a typical waypoint task.

Comments and considerations

The simple network of Figure 3 has two very desirable properties: a small number of connections, and a small number of well separated parts. The first quality greatly reduces the dimensionality of the space in which the evolutionary algorithm searches for solutions, improving the speed of the evolutionary process; the second overcomes the problem of neuronal interference that tends to arise between the parts of a single network. Unfortunately, such a simple network neglects any sort of coupling between the helicopter axes, a situation which cannot be tolerated if effects such as side-slip are to be avoided. Although this first network was useful for this initial proof-of-concept investigation, a higher degree of interconnectedness will be required in the final network.

The use of any kind of domain knowledge to constrain the network structure is a double edged sword: it can simplify the evolutionary process, but at the same time may rule out the discovery of an unconventional but good solution. Ultimately we would like to supply as little domain knowledge as possible to remove constraints on the evolutionary process, but it will not be possible to make an informed judgment on this until the final dynamic model is available.

In the development work carried out so far, we have neglected the noise and error usually present in the estimated state variable used for control. The next stage of this work will augment the helicopter model by taking into account the likely state estimation error. This will enforce a degree of robustness in the evolved controller, and should simplify the task of porting it to the real platform.

6 Conclusions

Although the project is still at an early stage, initial work has confirmed the availability of small rotorcraft suitable for extended and agile flight in our indoor arena. Existing technologies will be adequate for equipping the vehicles with onboard computational, communications, and sensory resources, for providing accurate localisation for navigation, and also for identifying the flight characteristics of the machines. Preliminary investigation of a technique for automating the design of the flight controllers has proved promising. Although there is still much work to be done, a credible technical basis for the SwarMAV system has now been established, and development can now proceed with some assurance of success.

References

1. C.W. REYNOLDS. Flocks, herds, and schools: A distributed behavioral model. *In Proceedings of the Conference on Computer Graphics (SIGGRAPH*), volume 21:4, pages 2534, 1987.
2. I.D. KELLY & D.A. KEATING. Flocking by the fusion of sonar and active infrared sensors on physical autonomous mobile robots. In *Proceedings of The Third Int. Conf. on Mechatronics and Machine Vision in* Practice, volume 1, pages 14, 1996.
3. M.J. MATARIC. Interaction and intelligent behavior. *PhD thesis*. *Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.*
4. REGMI, R. SANDOVAL, R. BYRNE, H. TANNER AND C.T. ABDALLAH, "Experimental Implementation of Flocking Algorithms in Wheeled Mobile Robots," *2005 American Control Conference*, *pp 4917 – 4922.*
5. J. WELSBY and C. MELHUISH. Autonomous minimalist following in three dimensions: A study with small-scale dirigibles. In *Proceedings of Towards Intelligent Mobile Robots*. Manchester, 2001.
6. NASA. Dryden Flight Research Center. New flight software allows UAVs to team up for virtual fire experiment.

http://www.nasa.gov/centers/dryden/news/NewsReleases/2005/05-12.html.

7. CROWTHER, W.J. and RIVIERE, X. Flocking of Autonomous Unmanned Air Vehicles. *17th Bristol UAV Conference, 2002.*
8. JADBABAIE , J. LIN, and S.A. MORSE. Coordination of groups of mobile autonomous agent using nearest neighbor rules. *IEEE Trans. Autom. Control, 48(6), 2003.*
9. R. OLFATI-SABER. Flocking for Multi-Agent Dynamic Systems: Algorithms and Theory. *IEEE Trans. on Automatic Control, vol. 51, Mar. 2006.*
10. O. HOLLAND, J. WOODS, R. DE NARDI, AND A. CLARK. Beyond Swarm Intelligence: The Ultraswarm. *Proceedings of the IEEE Swarm Intelligence Symposium (SIS2005)*
11. Systems: Algorithms and Theory, HIROBO LIMITED. XRB Lama helicopter *http://model.hirobo.co.jp/products/0301905/index.html.*
12. J.C. ZUFFEREY. Bio-inspired Vision-based Flying Robots. *PhD thesis, EPFL, 2005.*
13. H. ZHENG, R. BUYYA and S. BHATTACHARYA *(1999) Mobile Cluster Computing and Timeliness Issues, Informatica, 23: 1 1999*
14. P. ABBEEL, V. GANAPATHI and A. Y. NG. Modeling Vehicular Dynamics, with Application to Modeling Helicopters. *Neural Information Processing Systems, Vancouver, Canada, December 2005*
15. G. BUSKEY, J. ROBERTS, P. CORKE, M. DUNBABIN and G. WYETH. The CSIRO autonomous helicopter project. *In Proceeding of the International Symposium on Experimental Robotics, 2002.*
16. A.Y. NG, H.J. KIM, M.I. JORDAN, S. SASTRY, and S BALLIANDA. Autonomous helicopter flight via reinforcement learning. *Advances in Neural Information Processing Systems, 2004.*
17. AUTOPILOT. Do it yourself UAV. *http://autopilot.sourceforge.net.*
18. B. METTLER and M.B. KANADE and T. TISCHLER. System identification modeling of a small-scale unmanned rotorcraft for flight control design. *Journal of the American Helicopter Society, 47:5360, 2002.*