BEYOND SWARM INTELLIGENCE: THE ULTRASWARM

Owen Holland[1], John Woods[2], Renzo De Nardi[1], and Adrian Clark[2]

[1]Department of Computer Science, [2]Department of Electronic Systems Engineering
University of Essex, Wivenhoe Park, CO4 3SQ, United Kingdom
owen/woodjt/rdenar/alien@essex.ac.uk

ABSTRACT

This paper explores the idea that it may be possible to combine two ideas – UAV flocking, and wireless cluster computing – in a single system, the UltraSwarm. The possible advantages of such a system are considered, and solutions to some of the technical problems are identified. Initial work on constructing such a system based around miniature electric helicopters is described.

1. INTRODUCTION

At one time or another we have all been impressed by the sheer agility of a flock of starlings flying in a city square at dusk – wheeling and manoeuvring so swiftly and precisely as to create the illusion of a single and very superior controlling intelligence. Artificial flock and swarm systems exploiting these abilities, which thanks to the seminal work of Craig Reynolds [1] are now well understood, have been the focus of active research for almost twenty years. But there is another way of looking at a flock of starlings: a typical flock will contain upwards of a thousand birds, and each bird will contain a gram or so of brain tissue, so in the aggregate the flock will contain about the same amount as a single human brain. If there were some way in which the starlings' brains could be linked together to form one human-sized nervous system, could the flock collectively achieve something approaching a human level of intelligence?

Of course, the knowledge that no such linkage is possible instantly takes the steam out of such a speculation (although some biologists have proposed analogies between ants and neurons, suggesting for example that the chemically-mediated interactions between individual ants in a colony and the chemically mediated interactions between individual neurons in a brain may support intelligent behavior in ways that are somehow similar). However, the constraints on the effective linkage of computational components clearly do not apply to artificial systems: the recent convergence between computation and communication means that distributed processing, in the form of cluster computing, is becoming the norm for high-performance computing. (In the latest top 500 supercomputer rankings, 58% are cluster machines.) In these clusters, large numbers of relatively low-powered computers are linked into a single architecture using high-bandwidth (1 or 2 Gigabits/sec) wired network connections. Might it be possible to construct a flock of individually simple artificial agents that flew like a flock of starlings, but was also able to process information like a cluster-based supercomputer *by using high-bandwidth wireless links between the agents' computational elements*?

We have named the general concept of combining swarm intelligence and wireless cluster computing *the UltraSwarm*. Although the genesis of the idea occurred in the context of flocking systems, the basic philosophy could also apply to swarm intelligence systems based on social insect behaviour. In both flock-based and social-insect-based UltraSwarms, the attraction lies in the potential for combining the two technologies of swarm intelligence and conventional computational abilities in a single hybrid system. Since it is frequently the case that what is difficult for one technology is easy for the other, there are grounds for expecting that such a hybrid system may outperform a system based on one technology alone, at least in certain applications.

This paper examines the UltraSwarm concept from four different angles: the problems that may arise in attempting to combine swarm intelligence with conventional computation; the technical aspects of wireless cluster computing using homogeneous mobile nodes; the applications that might match the abilities of an UltraSwarm; and the problems of building a practical UltraSwarm system as revealed by some preliminary experimental work.

2. WIRELESS CLUSTER COMPUTING WITH MOBILE NODES

The closest type of system to that required by an UltraSwarm is what has been called Mobile Cluster Computing (MCC), and many of the key problems were identified in an early paper on this theme [2]. However, apart from a brief but positive thread in the Beowulf discussion group [3], and an interesting simulation of a wireless Beowulf cluster using 802.11b as a class project at Michigan State University [4] there has been astonishingly little work on this area since then, with most research focusing on the very different problem of connecting mobile nodes with limited capacity to particular types of resources (e.g. the recent Special Issue of IEEE Internet Computing [5]). In 2004, the Korean Mobile Grid project, apparently the only coordinated project in this area, noted that '…up to now, mobile grid research is weak.' [6].

In wired clusters, there is now a good understanding of how overall system performance is determined by the factors of processor bandwidth, local memory, interconnect bandwidth, interconnect latency, and communications overhead. However, there is no such clarity in MCC, where instead of a fixed topology of dependable nodes with high bandwidth interconnects, there is a constantly varying collection of nodes which may be far from dependable, connected by error-prone high latency low bandwidth links with high communications overheads. There is as yet no convincing demonstration that these problems, all of which can be expected to occur in an UltraSwarm, are tractable. In particular, the management systems for cluster computers are always located on a single dedicated machine which is not itself part of the cluster. This is a particular source of vulnerability which, given the high reliability of modern computers, is unlikely to be addressed as a priority by the mainstream cluster computing community.

Although significant progress in this area will be necessary before a true UltraSwarm system can be constructed, we believe that such progress will inevitably be made in the next few years. In this paper we have chosen to concentrate on the engineering issues of producing a UAV flocking system that is in principle capable of providing the resource base for the development of an UltraSwarm. Initially, our practical deployment of collective computation via wireless links will take the form of a more conventional distributed computation approach; as it will be some years before we can expect to have an UltraSwarm platform capable of supporting wireless cluster computing, it is reasonable to assume that the required developments will be in place by the time we need them.

3. ULTRASWARM APPLICATIONS

An UltraSwarm offers the agility and potential reliability through redundancy of a classical swarm, along with the possibility of the local application of considerable computing power to the analysis of data acquired by the distributed sensor structure. The only alternative computational approach of streaming data to some remote computer installation for processing and subsequent return is potentially insecure, slow, and subject to interference. It is easy to devise mission scenarios from both military and planetary exploration contexts in which UltraSwarms have the potential to outperform other architectures.

We will give a single example. Consider a reconnaissance mission in which the target of interest can only be identified by combining and analysing views from several different perspectives of its suspected location. A single large UAV carrying sufficient onboard computation would have to make multiple passes over the location to gather the data; the size of the UAV, combined with the extended time over the target, would increase the likelihood of the mission being discovered. A smaller UAV would be more likely to escape detection, but would then have to send data back to some remote computer facility for analysis, and this necessarily high-powered transmission would open up the possibility not only of detecting the mission, but of intercepting and analysing the data. In contrast, a swarm of small and intrinsically stealthy UAVs in a suitable spatial arrangement could gather the multiple views in a single pass, and analyse them by collective computation using only local low power (but high bandwidth) wireless links.

4. EXPERIMENTAL APPROACHES

This work had its beginnings in the Microsystems Laboratory at the California Institute of Technology, where a number of projects from 1997 onwards used a group of up to fourteen autonomous mobile robots, the MooreBots, an updated version of the LinuxBots originally designed at the University of the West of England by Owen Holland and Alan Winfield [7]. Each robot ran Linux on a PC104 single board Intel 386 PC equipped with a PCMCIA wireless LAN card. Although the available computational power and communications bandwidth at the time was too low to make the exercise worth carrying out, a suggestion by Alex Holland prompted active discussion of the potential for configuring the robot collective as a wireless cluster computer so that data gathered by the collective could be analysed by the collective, rather than by an offboard machine as was typical of our experimental work at the time. (It is worth noting that some modern wireless LANs – for example, 802.11a and 802.11g – have effective throughputs greater than the 10Mbps of the Ethernet used in the earliest Beowulf cluster computer [8].)

4.1 The Flying Gridswarm

The first development of the concept was at Essex in 2003, when we proposed the construction of the Flying Gridswarm [9]. The idea was essentially to produce an airborne and updated version of the MooreBot computational and communications components, and to fly four or five such machines in a Reynolds-style flock while analysing visual data gathered by the flock using the PC104s as a wirelessly connected computer cluster. The weight and power requirements of a high performance PC104/802.11 combination dictated the use of a heavier than air machine, and we selected the Chris Foss WOT4 model aircraft as a suitable platform (Figure 1). Fitted with a 0.75 cu.in. 4 stroke engine, this is a high performance aerobatic model, with a high top speed (120 m.p.h.) and outstanding low speed manoeuvrability, along with the ability to carry a substantial payload. The prototype was successfully fitted with a commercially available autopilot (MicroPilot MP1100) as a first step towards autonomy. The

Figure 1. The Flying Gridswarm prototype, shown with a 30cm ruler.

real-time range and bearing information necessary for flocking were to be provided initially by individual beacons carried on each aircraft, coupled with complex directional receiving aerials, but in the longer term we expected to migrate to purely vision-based flocking. (Air-to-air video sequences of other aircraft shot from the WOT4 show this to be challenging but probably feasible.) As part of the project, a detailed dynamic simulation of a flock of WOT4 aircraft was produced [10]; this enabled an assessment of the relationship between the aircraft's dynamics, the control system, and the constraints (such as viewing angle) imposed by a practical vision system.

Interestingly, during investigation of the beacon method, it became clear that, if data could be shared between all the individuals in the flock, range information alone was theoretically adequate in almost all circumstances for establishing the changing 3D spatial relationships between all individuals, and so flocking could be managed using range information alone. This strategy is examined further in section 5.1.3.

Although the Flying Gridswarm platform is very attractive from the point of view of size, payload, and power supply, it soon became clear that there were insuperable problems involved in getting access to a sufficiently large controlled airspace in the UK to cope with the all-too-likely event of a system failure during development. We have therefore switched our immediate focus to the examination of ways of exploring UltraSwarm systems using indoor flight.

4.2 Alternative platforms

Early experience with small helium blimps at the University of the West of England [11] had shown that it was possible to use them build a simple flocking system, although payload and manoeuvrability problems imposed severe constraints. A compromise technical solution involving buoyancy assisted flying machines (helium-filled powered aerofoils) is under investigation at Essex, but is not yet ready for multi-vehicle work. The use of small electric helicopters is a clear alternative possibility, but until recently the autonomous flight of such machines seemed impossible to achieve in practice because of their notorious instability. However, in the last year or so, two possible solutions have appeared: small (but expensive) gyro-stabilised indoor helicopters [12], and intrinsically stable co-axial helicopters [13]. The latter, currently marketed as remotely controlled toys, are very attractive indeed, and in fact have many of the key advantages of the elements of swarm systems: because they are simple, they are cheap and light - and because they are light, they are extremely robust, and can be crashed with impunity. Their stability is such that, when properly trimmed, they can be flown to a given position and left there to hover hands-off for a time with no adverse effects other than a very slight drift. They are also stable in forward flight, and so autonomy may be achievable with relatively few problems. Although the payload of these small machines is limited, recent developments in electronics make it feasible for them to carry enough computation and communication to serve as a testbed for the major elements of the UltraSwarm concept. Figure 1 shows a modified Proxflyer design, the Bladerunner, fitted with a miniature wireless color video camera.

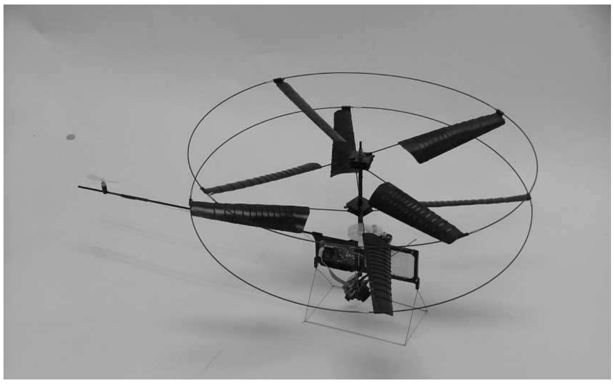

Figure 2. A Proxflyer Bladerunner fitted with a color video camera

5. A RESEARCH PROGRAM

In order to prepare the ground for producing a useful prototype, or proof of concept system, it is necessary to undertake a varied program of research. There are three key questions that need to be answered:

(1) Can the Proxflyer system be modified to carry sufficient sensing, computation, communication, and power to sustain the UltraSwarm concept?

(2) Will the modified helicopters be sufficiently stable to permit the development of an autonomous flight capability?

(3) Will it be possible to fly the helicopters sufficiently close to one another to permit meaningful flocking?
Each of these will be discussed in turn.

5.1 Sensing, computation, communication, and power

5.1.1 Sensing

The onboard sensing required for a proof of concept system must meet three requirements: it must supply sufficient information to enable autonomous flight; it must supply sufficient information to enable flocking; and (ideally) it must supply information to be used in the distributed computation task to be undertaken by the swarm as a whole. (The last consideration derives from the kind of task most likely to be undertaken by a deployed system, which will take advantage of the system's capacity for distributed sensing.) After reviewing a range of possibilities, we have reached the rather surprising conclusion that an onboard vision system offers the most immediate prospect of progress on all three fronts. We are fitting each helicopter with a downward-looking colour miniature video camera (a spycam); the deciding factor, which is intended purely as a temporary measure to speed up development, is that we are using wireless cameras which allow fast off-board vision processing, the results of which can then be passed back to the helicopters via the main communications network. The cameras, which weigh 6.7g but can easily be reduced to around 5.5g, can be adjusted to broadcast on different frequencies, and the available bandwidth allows up to four to operate in the same environment.

Figure 3 shows the camera's view of the arena from a height of almost 6 metres. (The checkerboard pattern is due to the tile segments of the electrically powered floor, used for long-endurance robot experiments). The field of view from the simple lens is adequate for our purposes, as is the 320 x 240 resolution. The CMOS camera chip is accessible, and the output, including the frame rate, can be configured in a useful variety of ways. Although the helicopters produce some vibration, we have found that a compliant mounting is all that is necessary to give a usable image.

The vision system will meet the sensing requirements in the following way: The floor of the arena will be marked with a small number of colored circular patches. The positions of these patches as seen by a given helicopter can be easily and rapidly determined, and can give unambiguous information about the helicopter's position in the arena, and also about its orientation and attitude. In most conventional helicopter systems, the timeliness with which this information can be obtained would not meet the requirements for flight control; however, the intrinsic stability and relatively slow movements of the Proxflyer machines mean that the combined processing and communication delays are unlikely to present a serious problem for achieving individual autonomous indoor flight.

The requirements for supporting Reynolds-type flocking are that each helicopter should have information about the range, bearing, and velocity of its neighbours. Since the ground-based computer system has information about the positions and orientations of all the helicopters, and since the speeds of the helicopters can be estimated from their previous positions, all of the information required can be processed and passed to each helicopter via the communications link.

Figure 3. The camera's view of the arena from a height of almost 6m.

Although, in the first instance, the sensing needed for autonomous flight and flocking will be processed by ground-based computers, in the medium term it is entirely feasible for all this information to be extracted by the onboard systems of the swarm members. However, it may also be possible to augment or supplant the visually-derived range information by taking advantage of the received signal strength indication (RSSI) built into the wireless communication circuitry of each machine – this is discussed further in section 5.1.3.

The final sensing requirement, that it should provide data for a credibly useful distributed computation task, is of course easily met. Most aerial systems involve reconnaissance of some kind, and the onboard cameras certainly supply the right kinds of data to support the fusion and extraction of information from multiple aerial viewpoints.

5.1.2 Computation

The requirements for onboard computation are rather more open-ended, but again recent technical developments provide a path for progress. The robots that inspired this study ran Linux, an extremely efficient, flexible, and configurable operating system that can deliver significant performance from hardware that is modest by today's standards. Within the last year or so, a number of miniature computer systems have been developed which combine small size, light weight, and the ability to run Linux. We have reviewed these, and the one that best meets our needs is the Gumstix system [14]. In its simplest form, the Basix200 platform, it offers a 200MHz Intel XScale® PXA255 processor with 64MB of SDRAM and 4MB flash memory, and comes with the Linux 2.6.11 kernel. In performance terms, this is some three generations more advanced than the Intel 386 processors in the original LinuxBots. Most remarkably, all this is achieved in a system 8cm x 2cm x 0.63cm, and weighing only 8g.

5.1.3 Communication

The communication requirements are rather more difficult to meet. Although the Gumstix system can be interfaced to an 802.11b wireless LAN card via an SDIO socket, the lightest such card weighs 5g, and the additional power requirements are severe. We expect that, within a year or so, further technical progress will have overcome these difficulties. In the meantime, however, by using a version of the Gumstix board with a built-in Bluetooth module (the Basix200-bt) we can have access to a wireless communication channel of up to 723.2kb/s with almost no weight and power penalties. Of course, Bluetooth is currently too slow to support a wireless implementation of a conventional computer cluster, but speeds are increasing all the time – Bluetooth 2.0 with EDR has a data transfer rate of 2.1 Mbps [15], and we expect such devices to be available in miniature low-power packages within the lifetime of the project. However, we believe that the currently available bandwidth is sufficient to support all the messaging required for autonomous flight and flocking, with some left over for what we hope will be more than a token amount of distributed computation.

Perhaps the greatest drawback of the use of Bluetooth lies in the constraints on network connectivity imposed by the hardware. Where 802.11b allows direct communication between any members of the network, Bluetooth operates on a master-slave principle: once a network has been set up, the slave nodes (up to seven) can only communicate with one another via the master node. This imposes a time penalty on slave-to-slave communication, and a load penalty on the master. Our system will reduce the effects of these factors as far as possible by designating the ground-based computer system as the master.

On the positive side, Bluetooth is designed for local power-efficient communication between battery-powered devices, and the standard includes some infrastructure for limiting the transmission power to the minimum necessary. As part of this, most implementations provide RSSI (Received Signal Strength Indication) – the idea is that a node can automatically gather information about the strength with which the other nodes in the network are receiving its transmissions, and can therefore adjust its transmission power appropriately. This feature offers a means of estimating the range of other active and identifiable Bluetooth modules [15]. By combining information about the RSSI of all the nodes in a network, it turns out that it is also possible to use this feature as a basis for working out the relative positions of nodes in a given environment [16]. With the present level of technology, such positioning is rather slow and inaccurate, but future implementations of Bluetooth and 802.11 will inevitably improve on today's performance. This may provide an opportunity for using RSSI information to acquire range information which is sufficiently timely and accurate to support Reynolds-type flocking.

The standard Proxflyer helicopters are remotely operated via a typical model aircraft radio controller. This requires the helicopter to carry a corresponding radio receiver and decoder, along with three pulse width modulation power outputs for the upper, lower, and tail rotors, and some safety circuitry for limiting operation as a function of battery condition. Almost all of these functions can be replaced by the Gumstix Basix200-bt board, which provides a radio link, and output pins available for motor control; the helicopter's original printed wiring board (PWB) can therefore be discarded, saving weight and power; all that has to be added is a very small PWB for carrying the motor driver ICs and battery connections.

5.1.4 Power

The final question mark over the use of Proxflyer machines is the availability of sufficient power. The standard Bladerunner helicopter is fitted with a rechargeable lithium polymer battery giving sufficient power for the motors and onboard electronics to achieve flights lasting up to ten minutes. However, merely by mounting the camera (which requires extra components to step up the battery output voltage from around 3.6V to a minimum of 6V), the effects of the additional weight and current drain reduce the effective flight time to tens of seconds. In order to deal with this, it is necessary to use a larger battery, but this increases the weight to the extent that it is also necessary to replace the existing pair of rotor motors with more powerful versions, which add even more weight, requiring even more battery power. Fortunately, the effects of this vicious circle can be resolved satisfactorily by using two 310mAh lithium polymer batteries, giving several minutes' flight at the cost of some apparent loss of maneuverability. It is perhaps worth noting that none of this would have been possible without the development of lithium polymer battery technology, which has the ideal characteristics of very high energy density, high current capability, a stable voltage over the discharge cycle, and consistently low internal resistance.

5.2 Stability and autonomy

Although the unmodified helicopters are impressively stable in hover, and also in other flight modes, and the modified helicopters are still stable enough for a skilled human to control them remotely, there is no guarantee that we will be able to develop a successful autonomous flight system without achieving a formal understanding of the flight dynamics. The characteristics of conventional helicopters are well understood [17, 18], but the operating principles of the Proxflyer concept are novel, and as yet no mathematical model is available. The patent application [19] gives all the relevant constructional details, but the description of the method of operation is purely qualitative.

The two contra-rotating rotors are mounted co-axially, and each is powered via a separate motor. They do not have the collective and cyclic controls of a standard helicopter, but instead are mounted in an ingenious passively gimballed arrangement which automatically compensates for disturbances, provided that the helicopter is correctly set up. To ascend, both rotors are driven at the same high speed; to descend, both are driven at a lower speed. To change orientation, the speed of one rotor is increased, and the other reduced; the combined drag and inertial reaction causes the machine to turn in the direction of rotation of the slower rotor. (An electronic circuit controls the relative speeds to compensate for the effects on lift.) The horizontal tail rotor can rotate in either direction, and operates to tilt the vehicle so that it takes either a nose-down or tail-down attitude; in the first case the helicopter will then move forwards, and in the second it will move backwards. It cannot fly sideways like a conventional helicopter, but for our purposes this is actually an advantage, in that the available degrees of freedom are then similar to those of Reynolds' 'boids', the simulated mobile agents used to develop the original flocking concept [1].

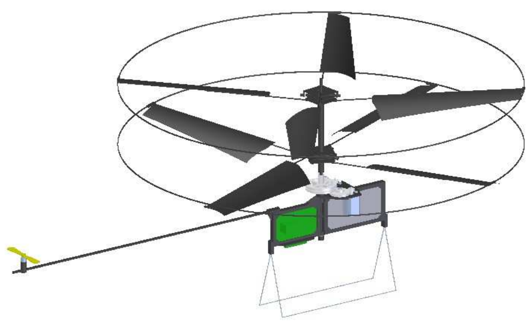

Figure 4. The Solid Edge model of a Bladerunner

In order to enable us to devise an autonomous controller, we have committed to the development of a dynamic simulation of a modified Proxflyer machine. Our first step was to produce a quantitatively accurate static 3D model of the original machine. A Bladerunner was stripped down, and all parts were weighed and measured. Using the 'Solid Edge' software package [20], we then created the model (see Figure 4). This enabled us to obtain calculated values for the position of the centre of gravity (the exact position of the centre of gravity is known to be crucial for the machine's stability and maneuverability), and for all the relevant moments of inertia. The next step will be to use the model as a guide to the optimum placement of the components in the helicopter modified to carry extra batteries, the camera, and the Gumstix board. Finally, we will integrate this model with a quantitative model of the dynamics verified against the helicopter's actual performance. (We are grateful to Petter Muren of Proxflyer, and to Bob Glade of Lockheed Martin Aerospace for technical assistance with the Proxflyer dynamic simulation.) The dynamic simulation will then be used to develop the controller.

5.3 Flying close and flocking

Although the use of these small helicopters offers a good prospect of achieving true flocking, albeit only in an indoor environment, there is one potential drawback: like all helicopters, they are vulnerable to disturbed air, and the major source of disturbed air will be the other helicopters. One of the remarkable things about flocking, especially in birds, is the way in which the flock agents combine rapid maneuvering with an avoidance of collisions. If the flock agents always maintain a very large minimum separation, the phenomenon loses much of its interest.

For conventional helicopters, the accepted rule of thumb is that they should stay at least two rotor diameters clear of one another when hovering. Experience suggests that a separation at least as great as this is also appropriate for the Proxflyer machines. However, flocking involves motion in the same direction, and this means that one helicopter may fly into the disturbed air left by another. If this happens, it may have a number of possible effects: it may destabilise the helicopter and lead to a collision or crash; or it may affect the dynamics of the helicopter so that a different control regime is required. Although the second possibility is relatively benign, the inhomogeneity and asymmetry it introduces into the operation of the flocking mechanism may destabilise the flocking itself. As far as we are aware, there are no studies of flocking which explicitly take into account effects like these, although there is some work dealing with flock agents with significant dynamics [21], and dynamic coupling of flock agents via vortices is known to occur in migrating pink-footed geese [22] and other species.

Even if the problem of disturbed air proves to be serious, it need not affect our study, because it should still be possible to flock successfully by reducing the speed of the flock so that the disturbance created by one helicopter has dissipated by the time the next helicopter arrives. This may interfere with the spectacle (which should still be more impressive and informative than would be the case with a blimp-based system) but will not affect the integrity of the basic process.

Figure 5. This shows a modified Proxflyer machine, with camera and Gumstix board, hovering in the arena. It is being remotely controlled via the Bluetooth link from the ground based computer – the control interface is visible in the top left of the screen, and the relatively large and heavy aerial can be seen pointing downwards from the body of the helicopter. The bottom left of the screen shows the view from the camera. The right half of the screen shows a web page served up over the Bluetooth link by the web server running on the Gumstix board.

6. IMPLEMENTATION

We have now implemented all the basic components described in the previous section. Figure 5 shows a Proxflyer machine in which all the original electronics have been replaced by the Gumstix board and a simple motor control interface board. The onboard camera has been modified to reduce the weight to 5.5g. The original single battery has been replaced with two 300mAh batteries in series to power the new more powerful motors, and a further 145mAh battery has been added to power the electronics. The total weight is now 76.6g, but the extra power of the motors provides sufficient lifting power, and the machine is stable.

The helicopter can now be controlled across the Bluetooth link, via the remote control GUI seen in Figure 5. The three channel controller has three components: (a) a Linux device driver installed on the Gumstix – a `char` driver providing functions to set and read the PWM duty cycle independently for each channel; (b) a UDP server daemon running on the Gumstix, accepting datagrams from a remote client; (c) a UDP client (with GUI) installed on the ground based computer, which sends data to the Gumstix server. This client can also contact the server and display the exact contents of the PWM registers inside the Gumstix's XScale processor.

We have illustrated the flexibility and power of the Linux/Bluetooth system in Figure 5 by showing a web page served up across the Bluetooth link by a web server running on the Gumstix board. (This is probably the smallest flying web server ever built!) The camera picture is transmitted directly to a TV tuner card by the camera's built-in wireless link, and is therefore available for ground based vision processing, the results of which can be sent back to the helicopter, or indeed to all the helicopters, via Bluetooth. We are now in a position to investigate autonomous flight by a single machine, using the technique outlined in Section 5.

7. CONCLUDING REMARKS

Although the experimental work described here is still in the early stages, it is clear that an UltraSwarm platform is potentially capable of implementation with the present generation of technology.

8. ACKNOWLEDGMENTS

The authors are grateful to Petter Muren for advice on modifying Proxflyer machines to carry greater payloads, and to Bob Glade for discussions on the dynamic modelling of Proxflyer systems. This research was supported by a grant from the University of Essex Research Promotion Fund.

9. REFERENCES

[1] Reynolds, C. W. (1987) Flocks, Herds, and Schools: A Distributed Behavioral Model, Computer Graphics, 21(4) (SIGGRAPH '87 Conference Proceedings) pages 25-34.

.[2] Haihong Zheng, Rajkumar Buyya and Sourav Bhattacharya (1999) Mobile Cluster Computing and Timeliness Issues, Informatica, 23: 1 1999

[3] http://www.beowulf.org/archive/2002-November/008867.html

[4] http://john.oleszkiewicz.net/work/proj/wireless/

[5] IEEE, Internet Computing, July/August 2004

[6] http://gridcenter.or.kr/MobileGrid/index.php

[7] A.F.T.Winfield and O.E.Holland (2000) The application of wireless local area network technology to the control of mobile robots, Microprocessors and Microsystems, 23: 10, pp 597-607

[8] http://www.beowulf.org/overview/history.html

[9] http://gridswarms.essex.ac.uk/index.html

[10] De Nardi, R. (2004) Flocking of UAVs: Software model and limited vision simulations. Tesi di Laurea, Dipartimento Di Ingegneria dell'Informazione, Universita degli Studi di Padova.

[11] Welsby J and Melhuish C. (2001) Autonomous Minimalist Following in Three Dimensions: A Study with Small Scale Dirigibles. Proceedings of Towards Intelligent Mobile Robots (TIMR). Technical Report Series, Manchester University, Department of Computer Science.

[12] http://www.draganflyer.com

[13] http://www.proxflyer.com

[14] http://www.gumstix.com, http://www.gumstix.org

[15] http://www.bluetooth.com/news/releases.asp?A=2&PID=1437&ARC=1

[16] S.Feldmann, K Kyamakya,A Zapater, and Z Lue. An Indoor Bluetooth-Based Positioning System: Concept, Implementation and Experimental Evaluation. International Conference on Wireless Networks 2003: 109-113

[17] W.Z. Stepniewsky and C.N. Keys (1984) Rotary-wing aerodynamics. Dover Publications

[18] Wayne Johnson (1994) Helicopter theory. Dover Publications.

[19] US Patent Application Publication US 2004/0245376A1

[20] www.solidedge.com

[21] Brogan D.C. and Hodgins J.K. (1997) Group behaviours for systems with significant dynamics. Autonomous Robots 4.1 pp. 137-153

[22] C.J. Cutts and J.R. Speakman, (1994) Energy savings in formation flight of pink-footed geese. Journal of Experimental Biology, vol. 189, pp. 251-261