

Communication and Distributed Control in Multi-Agent Systems: Preliminary Model of Micro-unmanned Aerial Vehicle (MAV) Swarms

Fabio Ruini (fabio.ruini@plymouth.ac.uk)
Adaptive Behaviour and Cognition Research Group
School of Computing, Communications and Electronics
University of Plymouth, UK

Abstract

This report focuses on the use of Multi-Agent Systems for modelling of Micro-unmanned Aerial Vehicles (MAVs) in a distributed control task. The task regards a search scenario in the context of security and urban counter-terrorism. In the simulation developed, a swarm composed by four autonomous MAVs, driven by a neural network controller, has to approach a target placed somewhere within the given environment, and then carry out a detonation when close to it. In Part I of the report we provide an overview of the state-of-the-art literature on distributed control and communication in multi-agent systems. Then we introduce the domain of unmanned flight from a historical perspective and finally we review the most relevant work about autonomous aircrafts path-planning. The Part II we describe the simulation research. It includes a description of the MAV swarm software simulator developed and the results of the first set of simulations. We use genetic algorithms to evolve the neural controller of MAVs. In the experiments carried out to date, with the aim to identify the most appropriate set of sensors for the neural network inputs, the evolved population of MAV swarms is able to reach and destroy the target on average 93% of the time.

PART I

1. Distributed control and communication in Multi-Agent Systems (MAS)

Distributed control, particularly when it requires a certain level of coordination/cooperation (Reynolds [19], Nitschke [20], Baldassarre et al. [21]), is a notably interesting problem from both a technological and scientific perspective. Compared to centralised control where a central controller (e.g. human operator or airplane's "leader" agent) is responsible for the (pre)planning, task-assignment and supervision of the coordination task, in distributed control systems intelligent autonomous, or semi-autonomous, agents are capable of sensing, acting, cognition and communication and together contribute to the task solution. These network-centric systems only require partial interaction with other agents, and may necessitate

simpler architectures and individual resource requirements as knowledge is distributed in the population.

The significant advantages of this approach are that the system is more robust, adaptive and fault tolerant since there is no critical reliance on any specific individual, and that decentralization results in increased reliability, safety and speed of response (Baldassarre et al. [21], Eklund et al. [22]). In addition, distributed approaches have the benefit of not requiring the full pre-planning of the cooperative strategy. Adaptive solutions can emerge run-time through the interaction between autonomous individuals and from the task and environment requirements whose might not be fully accessible (known) at the beginning of the problem.

Studies on distributed control greatly benefit from the utilization of Multi-Agent Systems (MAS), since they provide a platform for simulation and testing of various hypotheses based on the principle of distributed (artificial) intelligence (Weiss [23]). Distributed control MAS approaches have been used in various domains, such as unmanned air, terrestrial/underwater vehicles, search and rescue scenarios, collective robotics, social cognition etc. For example, Sastry and colleagues have focused on coordination and distributed control in unmanned underwater vehicles (Eklund et al. [22]); Sykara and collaborators (Koes et al. [24]) have concentrated their attention on the study of hybrid rescue group systems based on humans, software agents, and autonomous robots. What they propose is coordination architecture capable of quickly finding optimal solutions to the combined problems of task allocation, scheduling, and path planning subject to system constraints. In the SWARM-BOT project (Baldassarre et al. [21]; Trianni and Dorigo [25]) groups of robots evolve a cooperative strategy for exploratory tasks. In such a study distributed coordination implies that the characteristics of the group's behaviour (e.g. individual sensorimotor strategies, the roles played by the different robots, the synchronization problems raised by their interactions) are not managed centrally by one or few "leaders" but are the result of self-organizing processes such as "positive feedback" (if each individual of a group follows a rule of the type "do what the majority does", the individuals' behaviours will tend to become homogeneous) or "consumption of building blocks" (e.g. if the number of individuals forming a group is limited, the process of convergence towards the same behaviour caused by a positive feedback mechanism will necessary slow down and then stop). Finally, various MAS models of social cognition have been proposed, such as those modelling animal collaborative tasks such as in ant colonies (which have inspired the SWARM-BOT application) and predator group behaviour (Barry and Dalrymple-Smith [26]).

One important issue that has not been directly addressed in distributed control MAS is that of explicit communication between agents, and between agents and humans in hybrid systems. Most MAS models that have considered communication typically refer to implicit forms of communication, such as visual cues in predator models (Barry and Dalrymple-Smith [26]) and stigmergy communication in colonies (Trianni and Dorigo [25]), or to the technical aspects of agent communication protocols (Weiss [23]). Instead, the use of explicit forms of communication (e.g. symbolic, language-like systems) can be crucial in tasks requiring higher level cognitive capabilities, such as planning and decision making, and for the integration of language and cognitive capabilities (Perlovsky [27] [28], Tikhanoff et al. [29]). By explicit forms of communication we mean the use of symbolic lexicons in which it is possible to

identify a clear symbol/meaning relationship grounded on the agents' collaborative task properties and processes. New studies on the role of explicit communication in MAS have many theoretical and technological implications. First, agents that are allowed to communicate explicitly during the execution of collaborative task might benefit from the exchange of information regarding properties of the task being processed. Such explicit communication systems do not have to be defined a priori by the human designer, but can autonomously emerge from social interaction between agents (Cangelosi et al. [30]; Marocco and Nolfi [31]). A second advantage of studying symbolic communication concerns the development of human-centered systems and hybrid human/agent/robot systems. This will support the reconciliation of human decision making schemes with machine performance and intelligent agents, keeping the human in the loop (Koes et al. [24]). Finally, the post-hoc analysis of the communication systems developed by the agents (with or without interactions with human users) can give meaningful insights on the optimal strategies upon which the distributed control strategy is based. This can be also used for the design and improvement of human-centered distributed control systems.

2. Unmanned flight: An overview

First of all, it is important to define what an Unmanned Aerial Vehicle, also known as UAV, is. In order to do this we will adopt the definition provided by the Department of Defence Dictionary of Military and Associated Terms [W1], which states:

“A powered, aerial vehicle that does not carry a human operator, uses aerodynamic forces to provide vehicle lift, can fly autonomously or be piloted remotely, can be expendable or recoverable, and can carry a lethal or nonlethal payload. Ballistic or semiballistic vehicles, cruise missiles, and artillery projectiles are not considered unmanned aerial vehicles”

The main advantages of using UAV instead of traditional manned aircrafts consist in avoiding risks for pilots and, at the same time, increasing the chances of success for the missions they take part in. In fact, as Cambon and colleagues [W2] report, UAVs are typically employed in the so-called “dull, dirty and dangerous” missions. The “dull factor” is easily understandable: during long and repetitive missions (think, for example, about the B-2 crews that during the recent Kosovo conflict were constrained to fly 30-hour roundtrip missions from Missouri to Serbia) a machine could provide a better alertness state in comparison with a human, improving the overall probability of success for the mission. The “dirty” aspect is somehow related to tasks where the danger doesn't come from the enemy but from some other source instead. For example - despite wearing lead-lined flight suits and having their aircraft washed down upon landing - the U.S. pilots who flew data-gathering missions over Bikini Atoll in the Pacific immediately after nuclear tests in 1946 suffered radiation-related sickness. At last, the “dangerous” factor might be both physical and political. Physical if we consider that a manned aircraft exposes human pilots to any kind of risks, especially during reconnaissance missions (consider that during the WWII the American 3rd Reconnaissance Group lost over 25% of its pilots flying missions over North Africa in their initial months spent in theatre in 1942). Political if we think about the issues related to the capture of a soldier (and, during the Vietnam War, practically all the soldiers seized by Viet Cong were aircraft pilots). Some sources [1],

for example, link the growing interest of US Army in building unmanned aerial vehicles with the American U-2 spy plane shot down while flying over the Soviet Union's sky during May 1st, 1960 and the consequent capture of its pilot, Francis Gary Powers, by the Russians. Leading to the so-called "U-2 crisis" this event has clearly demonstrated to American governors how the capture of a pilot by the enemy, during the cold war, was no politically tolerable.

During the last few years, thanks to the quick improvements in components miniaturization, a new category of UAV has emerged. They are the so-called Micro-unmanned Aerial Vehicles (MAVs), properly known as Class I UAV according to the definition provided by the US Army [W3] that says:

"The Class I Unmanned Aerial Vehicle (UAV) provides the dismounted soldier with Reconnaissance, Surveillance, and Target Acquisition (RSTA). Estimated to weigh less than 41 pounds, the air vehicle operates in complex urban and wooded terrains with a vertical take-off and landing capability. It is interoperable with selected ground and air platforms and controlled by dismounted soldiers. The air vehicle also features a EO/IR/LD/LRF capability to perform the RSTA mission and utilizes a Heavy Fuel Engine (HFE) as its propulsion system. The Class I uses autonomous flight and navigation, but it will interact with the network and Soldier to dynamically update routes and target information. It provides dedicated reconnaissance support and early warning to the smallest echelons of the Brigade Combat Team (BCT) in environments not suited to larger assets. It will also perform limited communications relay in restricted terrain, a tremendous deficit in current operations. The system (which includes one air vehicle, a control device, and ground support equipment) is back-packable."

The MAV category is, in fact, just the natural outcome of the continuous UAV's technological improvements happened during the last decades. Year by year, MAVs are becoming autonomous vehicles even smaller and lighter than their ancestors, reaching true excellence points. Think for example to the EPFL's MC2, a 5-gram fixed-wing airplane, made out of carbon fibers and thin Mylar covering foils [2] [3], or to the MicroGlider developed by Wood and colleagues [4].

Even if we're now facing such a race toward the most extreme possible forms of miniaturization (that in turn could be useful for the original purpose that earlier UAV's builders were aiming: to collect the most accurate information about the enemy as possible), we've already reached a point in which MAVs can be successfully applied to innovative kinds of tasks.

2.1 Autonomous UAVs/MAVs path-planning

The fact that an aircraft is not able to carry any human pilots directly implies that it has to be driven in a different way. Typically, the UAVs¹ used nowadays in real applicative scenarios are dynamically remote controlled (think for example to the

¹ In this section we will use the term UAV in a generic way, meaning any possible kind of unmanned aerial vehicle, including MAVs.

famous Predator [5] [W4]) by a human crew staying in a remote position (the TCS, Tactical Control Station). Many UAVs, at the same time, also have their own guidance systems, through which they can fly autonomously. Usually, these guidance systems are slightly similar to automatic pilots used within the civilian aviation domain as they simply provide to keep the UAVs following a given pre-planned route.

During the last few years we've assisted to an increasing interest in the development of more intelligent autonomous UAVs' controller systems. The focusing toward autonomous guidance systems is not only an economical matter, even if using flying robots instead of the usual combination "manned airplane plus human pilot" would allow to save the enormous amount of money required for pilots training, skills upgrading, and so on. The point is that a computer software can frequently outperforms human in carrying out many different tasks, both in terms of reliability (think, for example, to the "dull" factor already outlined in the previous paragraph) and accuracy (a computer software is more accurate than a human pilot to perform an already planned manoeuvre and, most important, it is able to perform with a stable and short reaction time).

The problem of the limited performance that a human pilot could offer becomes even more serious if we think about how to drive an entire swarm of MAVs flying within an urban environment. To control the various formation members from a remote position means being able to manage an infinite flow of information that comes to the TCS every second, and respond to it in the correct way. This information flow is incomparably higher with respect to the one typically received by a pilot driving a Predator many kilometres higher than the sea level. It is practically impossible, for a human, to manage all this data. This is actually impossible if we want to employ tens of MAVs and move them as a real swarm. These are, in fact, the true reasons behind the increasing interest in autonomous robotics showed in these years.

According to Richards and colleagues [6] current approaches for autonomous cooperative UAV control can be separated into few different groups²:

- *deliberative approach*: focused on developing a specific flight path for each UAV to follow. Such flight paths are rigid and no effort is made to alter them in the event that new information is received (such as the discovery of a hostile element in a warfare environment). In other words, the entire scenario is assumed as already known. This approach could be successfully employed for civilian flying planning, but it's too simply for our purposes;
- *adaptive replanning approach*: in order to achieve some degrees of flexibility, few deliberative systems incorporate an element of adaptive replanning. In the adaptive replanning, a centralized controller generates a specific flight path for each UAV to follow based on the information that is currently available. The UAV follows that flight path, sending sensor information back to the controller as it becomes available. As the controller receives new information,

² In reality, Richards, Whitley and Beveridge classify these approaches in four different groups. For simplicity purposes, we will limit the analysis to only three of these, excluding the so-called "behaviour based controller systems".

it may generate new flight paths that are broadcasted back to the UAVs. The new plans may, for example, take into account the location of a previously unknown enemy or the fact that a UAV was lost due to mechanical failure or for many other reasons. It appears immediately clear that the adaptive replanning has a number of drawbacks. Every time a new set of flight plans is generated, the centralized controller must transmit them to each UAV in the field. A non-trivial choice must be made as to when is the appropriate time to replan. The replanning process is not instantaneous, and by the time the new plan is sent to the UAVs it may already be obsolete;

- *reactive strategies*: rather than generate a specific flight path that must be updated during the missions, this approach tends to generate a so-called “reactive strategy” for every UAV. This kind of strategy is analogous to a single decision tree that controls the aircraft for the life of the mission. The decision tree determines changes in the UAV’s heading, based on immediate low-level information from sensors.

In the just mentioned work made by Richards et al., where a team of UAVs has to explore in a cooperative way a given area, the decision tree that control the various aircrafts is developed through genetic programming methodologies. Even if the main idea - according with the controller system can’t be something external to the UAVs but it has to be embodied instead - could be fully agreed, a more convenient approach might consist in the usage of evolutionary evolved neural networks (Parisi et al. [7], Nolfi and Parisi [8], Floreano and Mattiussi [9], Mitchell [10]), mainly for two reasons. First, it is easier to use neural networks instead of GP for this kind of task since the behavioural repository to give to the MAVs is far simpler. Second, if properly trained, neural networks allow a much greater generalization capability than a decision tree evolved through genetic programming.

Anyway, in both cases a computer simulation is required for reasons of cost and time (for an overview about the importance of simulations in modern science, see Casti [11], Parisi [12], Cecconi and Zappacosta [13]). The strategies developed have to be evaluated within the simulated environment, as the evolutionary process requires potentially thousands of strategy evaluations to converge on effective solutions. Evolutionary algorithms typically progress through thousands of failing solutions on their way to good solutions. In a simulator, these failures accrue no real cost, where repeated failures with real vehicles might.

Neural networks are frequently used in terrestrial and underwater robotics, but very rarely as controller systems for flying robots. The main exception encountered so far, reviewing the literature, consist in the work that Floreano and colleagues [14] [15] are carrying on at the EPFL. Their project is focused on employing fully autonomous MAV swarms, where each swarm’s member act as a signal repeater, in order to create a reliable communication infrastructure between human rescuers and base station working into areas hit by natural disasters. At the same time, Owen Holland and his research group [16] [17] are studying how to employ neural networks as controllers for autonomous helicopters.

Finally, even if this approach falls in the previously outlined category of adaptive replanning, other meaningful insights come from the work carried out within the Autonomous Flight System Laboratory, at the University of Washington. Stressing

the importance of using heterogeneous autonomous systems in place of traditional hierarchical structures, Rathbun and Capozzi [18] had developed an efficient path planning algorithm for situations where the UAVs need to modify their paths in order to avoid a number of other aircrafts flying in their vicinity.

PART II

3. Introduction to Simulation Experiments

Imagine the following situation. There is a terrorist - suppose a kamikaze - moving along the centre of a western city. We know that he is going to make an attack, detonating the explosive he is wearing, but we don't know exactly what the target of his action will be. The only thing we know is that we have no choice but to stop the terrorist at any cost. He just needs a movement of his finger to create a blast and provoke a lot of deaths. So, it is too risky trying to approach him directly. A sniper could be used instead. But we don't know where the attacker will move. So we need a big group of snipers, placed in the strategic points of the city. But it's not so easy to displace dozens of snipers within an urban city while passing unnoticed at the same time. Again, we cannot be sure of the outcome of the snipers' action. The attacker is walking, maybe frequently changing his marching direction, and there are a lot of people that could be hit while passing between the snipers and him. The sniper could also fire a non-lethal shot, leaving the terrorist able to accomplish, at least in part, his bloody mission.

Now, imagine a MAV, a Micro-unmanned Aerial Vehicle. Even better, imagine a MAV swarm. Silent as snipers, but more accurate. The swarm floats unnoticed above the city's sky, monitoring the movement of the attacker and using a common language to share important information between the various members. At a certain moment, the swarm decides that it's the right time to act. They start a nosedive and, in a couple of seconds, they reach their target. The swarm attacks vertically - from a higher position - in such a way that is invisible to the terrorist. Since the MAVs are silent as well, we can be sure that the target, which is walking, will never pay attention to what's happening above him. And even if he's looking upward, when the MAVs become visible to him it is too late to react. The target is eliminated through a low-potential self-detonation of the swarm's members, or the emission, by the MAVs, of liquid chemical inhibitors around his body makes him inoffensive.

Our research aims to demonstrate how this kind of scenario is possible and how it is possible to design a swarm of autonomous MAVs to be employed in such a urban terrorist scenario.

4. The MAV swarm simulator

During the five months spent at the University of Plymouth - within the Adaptive Behaviour and Cognition Research Group lead by Professor Angelo Cangelosi - we developed an ALife computer simulation that aims to reproduce a scenario similar to the one described above.

From a technical point of view, the source code of the simulation has been written in C++, using the Trolltech's Qt as graphical framework. The advantage of this kind of

approach mainly consists in the possibility to build (and run) the simulation on all the most widely used operating systems (i.e., Microsoft Windows³, MacOS X and Linux).

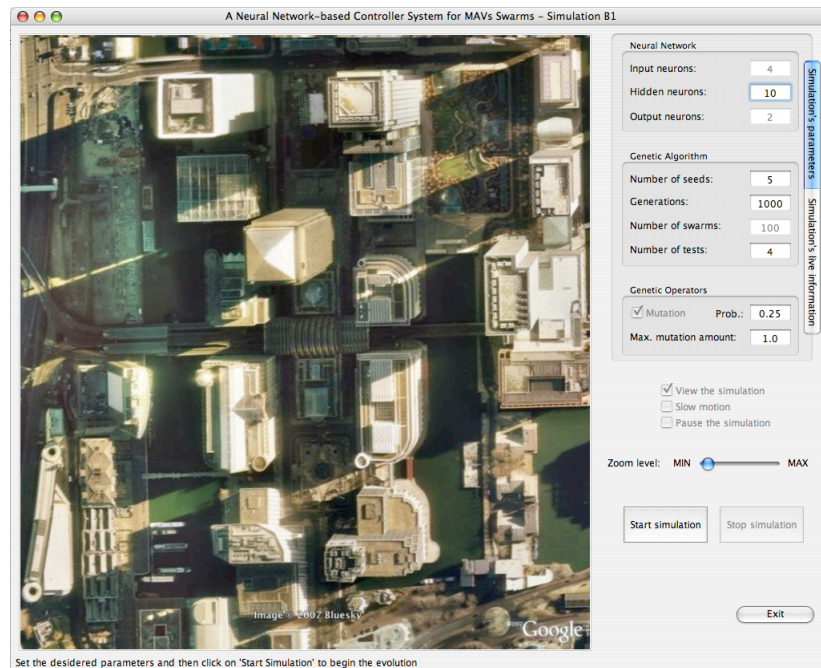


Figure 1 – A screenshot of the simulation's main interface.

Developing simulations where aircrafts are used in place of a more traditional terrestrial vehicle impose some particular constraints. The main one consists in the impossibility for our simulated MAVs to stop. In other words - since they are airplanes and not helicopters or other kinds of wheeled robots - the MAVs have always to move. For simplicity purpose, in first instance, they accomplish this task at a given and fixed speed. The MAVs do also have a given turning radius between 0° and 20° in the time unit, both clockwise and counter-clockwise. Moreover, flying requires energy. The MAVs are characterized with a certain autonomy level that decreases over the time (in a linear way, since the speed is fixed).

In order to define the MAVs' cruise speed and their autonomy (the turning radius had been instead chosen in a very conservative way), we have taken inspiration from an existent MAV model: the Wasp Block III.



Figure 2 – A picture of the Wasp Block III, the MAV model produced by Aerovironment and used as a model for our simulator.

³ Even if, at this moment, a known bug in the Qt framework denies the Windows' users from correctly seeing the images included in our simulation, unless users have the Qt library installed on their machine.

Built by the American factory Aerovironment (famous for being the same company which had already produced both the widely used Pointer [W5] and the more recent Hunter [W6]) this aircraft model is small enough to be used within an urban environment and his autonomy (see the table below) allows it more than 30 kilometres motion while waiting for the right moment when to attack the target.

Length	38 cm (15 in)
Wingspan	72 cm (28.5 in)
Weight	430 g (0.95 lb)
Speed	46-65 km/h (25-40 mph)
Range	5 km (3.1 miles)
Endurance	45 min.
Propulsion	Electric motor

Table 1 – Technical specifications of Aerovironment Wasp Block III [W7]

The environment where our simulation takes place is a rectangular area sized approximately 630x650 meters, representing a portion of London’s Canary Wharf quarter. A swarm is composed by 4 MAVs, with starting position close to the rectangle corners and facing the centre of the environment (with the addition of a certain amount of random noise to their starting orientation). A target is placed somewhere in the area, occupying a random position that is always visible to the swarm’s members.

The neural network which controls the MAV behaviour is a simple fully connected three-layer feed-forward network, detailed as follows:

- the input layer is formed by four neurons. One of them is dedicated to receive the sensorial input related to the distance that separates the MAV from the target; the other three are dedicated to the management of the angle between the two items instead;
- the ten neurons belonging to the hidden layer are characterized by a tan-sigmoid activation function, with minimum -1.0, maximum 1.0 and curve’s slope 1.0;
- the output layer is composed by two neurons. One of them, continuous, is dedicated to the MAV steering. Its output value can vary between -1.0 and +1.0, according respectively to a 20° left turn and to a 20° right turn. The other neuron is a Boolean one instead. When it turns to 1 the MAV detonates.

The training process takes place through a genetic algorithm. An initial population of 100 different swarms is created with connection’s weights and biases randomly assigned (consider that the MAVs belonging to the same swarm share the same connection’s weights and the same biases as well; they’re, in fact, clones). Each swarm is tested four times within four different environments, which vary only in terms of the position of the target. Each test starts with the MAVs displaced in their starting points, with the maximum amount of energy possible (5,000 energy units⁴).

⁴ Please consider that this value is strongly lower than the real one (that should be approximately 33,800). The decision to keep this value low is justified by the long time required to carry out a simulation when the MAVs have the same autonomy than their real counterparts. Sometimes, in fact,

Then, each swarm's member sequentially perceives its sensorial inputs, elaborates the behavioural response and actuates it (the action's implementation actually costs 3.01 energy units). The test ends when the target has been destroyed by a MAV detonated very close to it or when there are no more MAVs alive. Consider that a MAV – a part detonating - could also die if it moves out from the environment's boundaries, if it collides against a team-mate or if it finishes its autonomy.

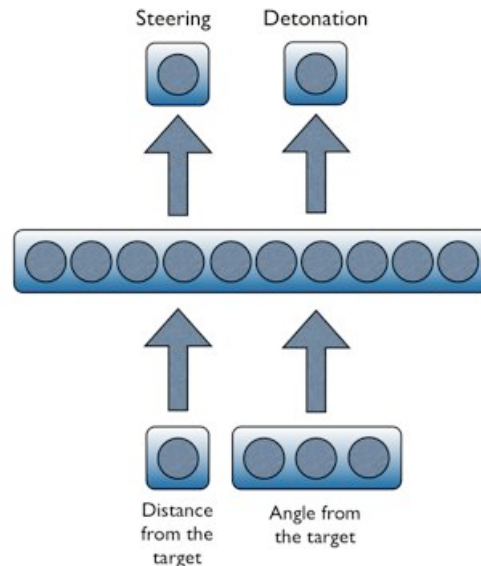


Figure 3 – The neural network's architecture of the MAVs.

The fitness formula through which the performance obtained by each swarm - after the conclusion of the four tests made - is measured is:

$$\text{Fitness} = -(\text{av. dist.}) + (\text{av. en.} / 50) + (\text{nr. of succ.} * 50) + (\text{nr. of MAVs alive} * 5)$$

where:

- *av. dist.* is the average distance between the target and the swarm's member exploded closest to it, calculated basing on the four tests;
- *av. en.* is the average amount of energy remained to the MAV detonated closest to the target, calculated basing on the four tests;
- *nr. of succ.* is the number of tests concluded by the given swarm with the elimination of the target;
- *nr. of MAVs alive* is the number of swarm's members remained alive after the 4 tests (maximum 3 MAVs x 4 tests = 12 MAVs);

This formulae tends to favour not only that swarms able to reach and destroy the target, but also the ones that are both quick in doing this task and capable to perform it losing the lowest possible number of MAVs.

The 20 swarms with the best performances according to this criterion are selected for the reproduction. Each of these swarm creates 5 copies of itself, which inherit its

especially during the first generations, might happen that some swarm's members move in loop, without reacting to the variation the sensorial perception, until the autonomy will finish.

connection's weights set, along with the biases related to the hidden and to the output layer (the input layer's neurons hasn't got any bias). A certain amount of random mutation (ranging between -1.0 and +1.0) is added to each inherited weight and bias with probability .25. The elitism operator is also applied in order to preserve the reproduction of the swarm that - within a given generation - performs best. In fact, the best swarm in a certain generation creates five copies of itself, but just four of these are subject to random mutations.

5. Experimental setup and results

As already stated in the previous paragraph, the MAV's neural network is able to perceive the target in terms of both angle and distance. From a technically viewpoint:

- the perceived angle ranges clockwise from 0 to 360°, starting from the MAV facing direction;
- the distance is measured in pixels and it's simply calculated as the Euclidian Distance (since our reference environment is basically a Cartesian plan) between the target and the centre of the MAV.

In order to find the most appropriate way to encode these sensorial inputs, a set of eights simulations aimed to this goal have been run. In these experimental setups, the evolution lasts for 500 generations. Angle and distance from the target's information received in input by the MAVs have been codified as follows:

Simulation	Angle	Distance
A1	Angle's range (0°->360°) segmented in 8 different sub-spaces, numbered according to a 3-bits Boolean encoding	Distance's range (0->1040) discretised in 11 intervals as shown in table 3
A2	Angle's range (0°->360°) segmented in 8 different sub-spaces, numbered according to a 3-bits Gray Code encoding	As Simulation A1
A3	Angle's range (0°->360°) encoded through two continuous neurons: the first receives in input the angle's sin, the latter is set with the value corresponding to the angle's cosine	As Simulation A1
A4	Angle's range (0°->360°) encoded through a single continuous neuron ranging from 0 to 1 (0° = 0.5, 180° = 0, 360° ≅ 1)	As Simulation A1
A5	As Simulation A1	Distance's range (0->1040) translated in the activation value of a continuous neuron ranging from 0 to 1 (0 = 1, 1 = 0)
A6	As Simulation A2	As Simulation A5
A7	As Simulation A3	As Simulation A5
A8	As Simulation A4	As Simulation A5

Table 2 – The different kinds of input encoding used during the first set of simulations

Distance	Normalized value
1040 <= distance < 900	0.0
900 <= distance < 800	0.1
800 <= distance < 700	0.2
700 <= distance < 600	0.3
600 <= distance < 500	0.4
500 <= distance < 400	0.5
400 <= distance < 300	0.6
300 <= distance < 200	0.7
200 <= distance < 100	0.8
100 <= distance < 2	0.9
2 <= distance <= 0	1.0

Table 3 – The rules followed to discretize the distance from the target in Simulations from A1 to A4

5.1 Results

Since the only genetic operator used is mutation, there is the risk that the solution space investigated by the evolutionary algorithms will not be the one containing the optimal solution. To mitigate this factor, each simulation has been repeated five times and the results have been then averaged. The plot and the table below show, in terms of average fitness, the performance comparison between the eight architectures analysed.

Table 4 contains a brief review of the results obtained by the eight simulations described in the previous paragraph.

Simulation	Average fitness*	Percentage of tests concluded successfully*
A1	110.4866	75.0900
A2	315.1785	93.4650
A3	-152.4581	11.6800
A4	55.3201	58.1450
A5	111.6630	75.3300
A6	240.1921	88.1400
A7	-142.2044	10.8950
A8	-287.0299	6.9700

Table 4 – Average fitness and percentage of tests concluded successfully. Comparison between the eight different kinds of input coding tested (*: average value of the last ten generations)

As we can easily see, both from this table and from the analysis of Figure 4, the encoding that has obtained the best results is the A2 (angle from the target divided in eight sub-spaces, numbered according to a Gray Code scale, distance discretised in eleven different values).

Looking more in details the A2 simulation's outcome, we can have a confirm of the fact that our simulated MAVs learn in a very efficient way how to reach and destroy the target.

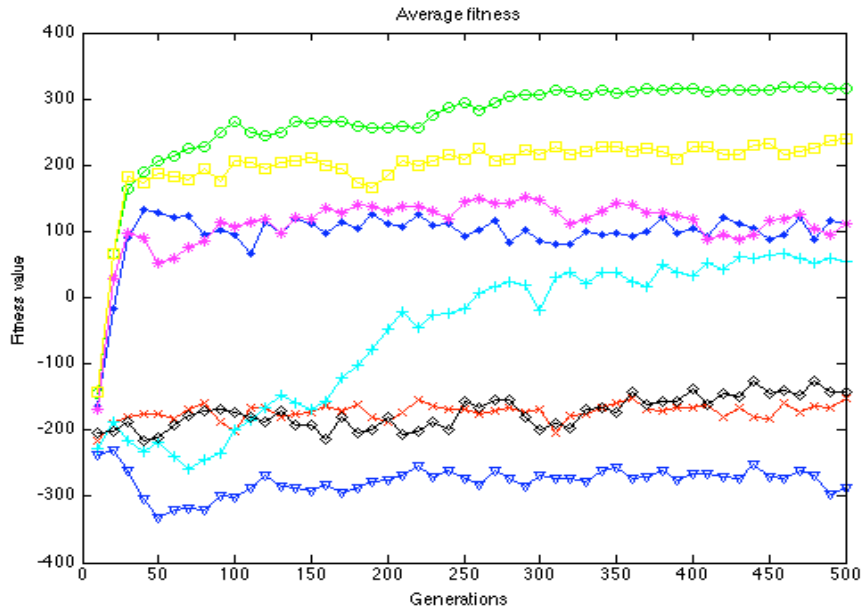


Figure 4 – Average fitness. Comparison between the eight different kinds of input encodings tested.

Figure 5 shows the fitness during the 500 generations. As expected, the average value grows with the time and at last tends to reach an asymptotic equilibrium state after 300 generations.

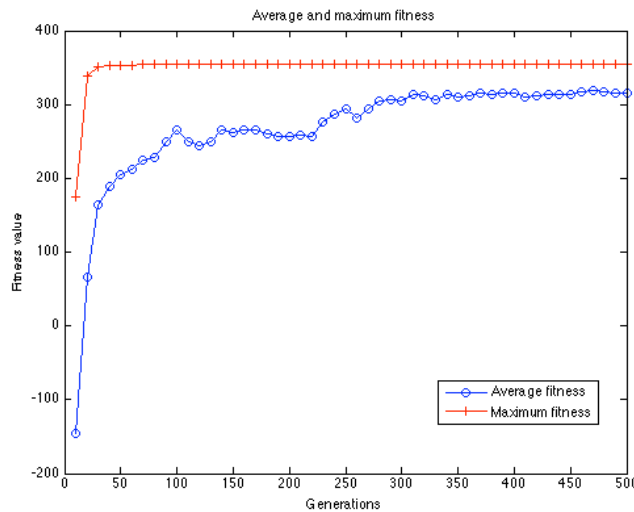


Figure 5 – Average and maximum fitness for Simulation A2

More interesting might be to have a look to the curve related to the maximum fitness (i.e., the fitness' value obtained by the best swarm within a given generation). In this case, we can surprisingly discover that after just few tens of generations (30-40) we've obtained swarms able to perfectly perform the given task. This conclusion is supported by the observation of the minimum distance's curve in Figure 6 (where the minimum distance between the target and the swarm's member detonated closest to it - averaged for the four tests made and related to the swarm with the better performance in this take - is plotted), which tends very quickly to zero.

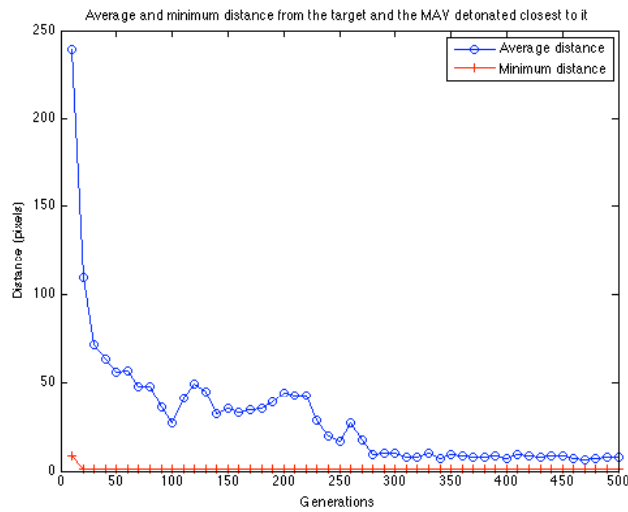


Figure 6 – Average and minimum detonation distance from the target for Simulation A2

The percentage of tests successfully concluded (Table 4 and Figure 7) is quite enthusiastic: 93.465%, on average, during the last ten generations. Considering also the random mutations introduced during the passage from a certain generation to the next one (which, practically, makes impossible to reach a 100% goal), the results is even more than a good one.

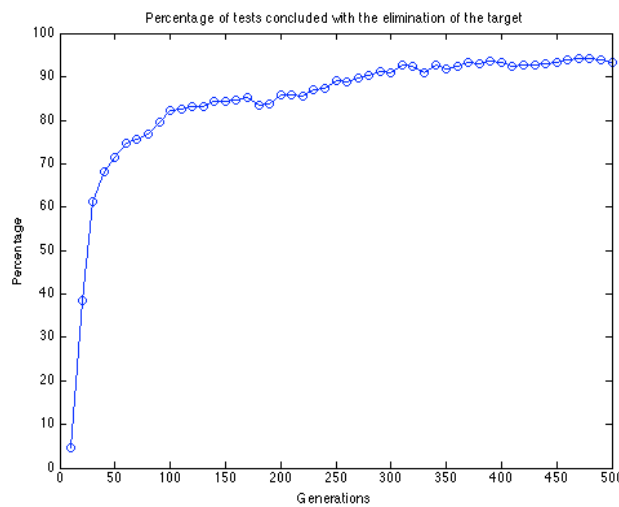


Figure 7 – Percentage of tests concluded with the elimination of the target for Simulation A2

Figure 8 shows us that, after a first phase during which the simulated MAVs learn how to accomplish the main task (to reach and destroy the target), the selective pressure is then exerted on a different factor: the speed. After 200 generations we're able to see the a clear tendency toward the evolution of swarms capable to perform ever quickly the given task.

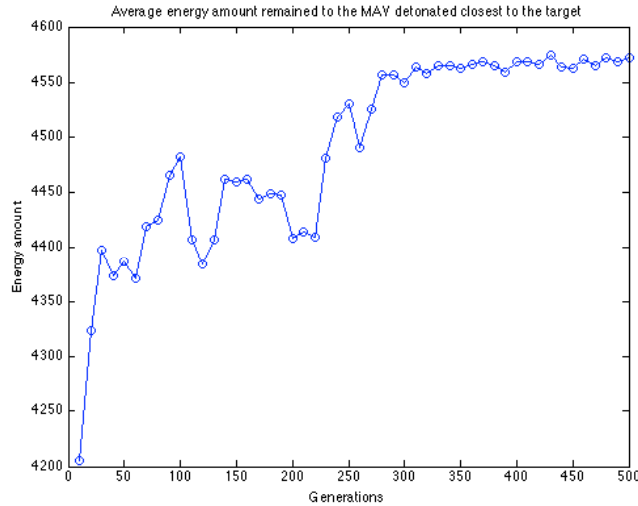


Figure 8 – Average energy amount remained to the MAV detonated closest to the target for Simulation A2

At the end, in Figure 9, we can find a graph where the final condition of the average swarm’s members, at the end of each test, is reported. The number of detonated MAVs, initially very high, tends to decrease over the time, reaching the final value of 1.5 for each swarm. This value sounds a bit too high yet, but it could be improved modifying in a proper way the fitness formulae (assigning more importance to the “number of MAVs alive at the end of each test” factor). Furthermore, since the MAVs are not able to perceive the environment boundaries, the percentage of them dying for this reason remains relevant and doesn’t get lower during the evolution. Considering all these factors, the final amount of MAVs alive at the end of each test (2.25) might be considered anyway a good result.

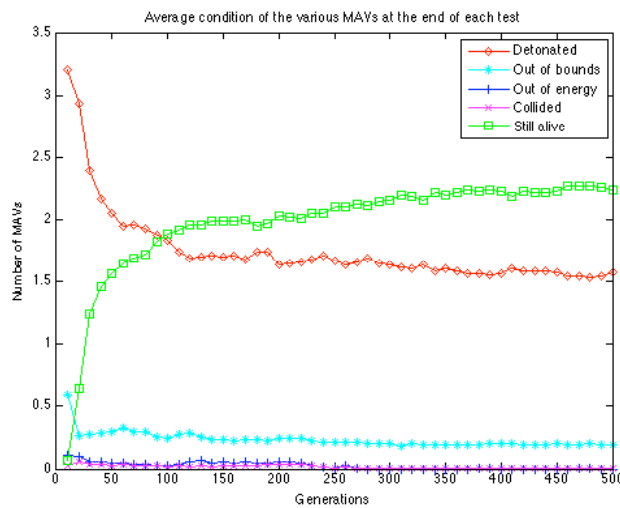


Figure 9 – Conditions of the MAVs belonging to the average swarm at the end of each test

6. Conclusions and further developments

In this work we have successfully demonstrated how a neural network controller for MAV swarms can be successfully evolved through ALife computer simulations based on evolutionary algorithms.

The next step will be to make our simulation more realistic, through the inclusion in the environment of some obstacles that the MAVs have to be able to avoid. Some preliminary work has already been done in this direction. As shown in Figure 10, we have mapped some of the buildings presents in the Canary Wharf and we have set them, in the context of our simulation, as “no fly zones” (a MAV that tries to enter into one of this areas will be immediately destroyed). We have also added to our simulated aircraft a ultra-sonic sensor that allows it to perceive the presence (and, in case, the distance as well) of any obstacle they are eventually facing.

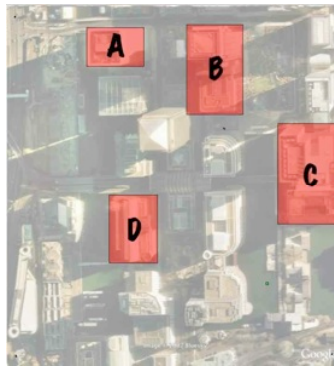


Figure 10 – A first mapping sketch of high buildings/obstacles.

With such a more realistic environment, we would like also to add to the MAVs’ behaviour a social dimension, trough the substitution of the target with a more robust one, which needs two contemporary hits in order to be destroyed. Then we will try to add to the simulated aircrafts the possibility to exchange communication signals between them, in order to achieve a better coordination level and a better performance in destroy the target consequently.

Another direction of study is to increase the number of MAV members belonging to a swarm and vary starting points. In this way we will be able to evolve a real swarm behaviour, like the one created by Reynolds in his classical work on flocks, herds and schools [19]. Moving toward a more ecological plausible scenario, we might be able as well to exploiting some ethological study, as the one carried out by Barry and Dalrymple-Smith [26] which suggest the employment of MAVs that are not clones, but that instead have particular individual characteristics (like, for example, a favourite direction to follow during the approach to their prey).

Gradually we aim to proceed toward an even more realistic scenario. We will use a three-dimensional environment, containing objects (not only MAVs and buildings, but also trees, people and car moving along, and so on), characterised by real physics properties.

Acknowledgments

This work has been made possible thanks to the euCognition support (euCognition Network Action NA097-3).

References

- [1] Newcome, L. (2004) *Unmanned Aviation. A Brief History of Unmanned Aerial Vehicles*. American Institute of Aeronautics and Astronautics;
- [2] Zufferey, J.C., Beyeler, A. and Floreano, D. (2007) *Insect-inspired Autonomous Microflyer*. International Symposium on Flying Insects and Robots, pp. 133-134;
- [3] Zufferey, J.C., Klaptocz, A., Beyeler, B., Nicoud, J.D. and Floreano, D. (2007) *A 10-gram Vision-based Flying Robot*. Journal of the Robotics Society of Japan. Special Issue on IROS'06. In press;
- [4] Wood, R.J., Avadhanula, S., Steltz, E., Seeman, M., Entwistle, J., Bachrach, A., Barrows, G., Sanders, S., and Fearing, R.S. (2007) *An Autonomous Palm-Sized Gliding Micro Air Vehicle*. IEEE Robotics & Automation Magazine, pp. 82-91;
- [5] Drew, J.C., Shaver, R., Lynch, K.F., Amouzegar, M.A. and Snyder, D. (2005) *Unmanned Aerial Vehicle. End-to-End Support Considerations*. RAND Corporation;
- [6] Richards, M.D., Whitley, D. and Beveridge, J.R. (2005) *Evolving Cooperative Strategies for UAV Teams*. Genetic and Evolutionary Computation Conference (GECCO 2005), ACM Press;
- [7] Parisi, D., Cecconi, F. and Nolfi, S. (1990) *Econets: Neural Networks that learn in an environment*. Network: Computation in Neural Systems, Vol. 1-2, pp. 149-168;
- [8] Nolfi, S. and Parisi, D. (1997) *Neural Networks in an Artificial Life Perspective*. W.Gerstner, A.Germond, M.Hasler, and J.D.Nicoud (Eds.) Artificial Neural Networks (ICANN97). Proceedings of the 7th International Conference on Artificial Neural Networks. Berlin: Springer Verlag, pp. 733-738;
- [9] Floreano, D. and Mattiussi, C. (2002) *Manuale sulle Reti Neurali*. Il Mulino, Italy.
- [10] Mitchell, M. (1998) *An introduction to Genetic Algorithms*. MIT Press;
- [11] Casti, J.L. (1998) *Would-be worlds. How Simulation is Changing the Frontiers of Science*. John Wiley & Sons Inc;
- [12] Parisi, D. (2001) *Simulazioni. La realtà rifatta nel computer*. Il Mulino, Italy;
- [13] Cecconi, F. and Zappacosta, S. (2007). *Simulazioni al computer: teoria ed applicazioni*. Aracne, Italy;

- [14] Floreano, D., Hauert, S., Leven, S., and Zufferey, J.-C., (2007) *Evolutionary Swarms of Flying Robots*. International Symposium on Flying Insects and Robots, pp. 35-36;
- [15] Floreano, D., Mitri, S., Magnenat, S. and Keller, L. (2007) *Evolutionary Conditions for the Emergence of Communication in Robots*. Current Biology, 17, pp. 514-519;
- [16] Holland, O., Woods, J., De Nardi, R. and Clark, A. (2005) *Beyond Swarm Intelligence: the Ultraswarm*. IEEE Swarm Intelligence Symposium (SIS2005);
- [17] De Nardi, R., Holland, O., Woods, J. and Clark, A. (2006) *SwarMAV: A Swarm of Miniature Aerial Vehicles*. 21st Bristol UAV Systems Conference;
- [18] Rathbun, D. and Capozzi, B. (2002) *An Evolution Based Path Planning Algorithm for Autonomous Motion of a UAV through Uncertain Environments*. Proceedings of the AIAA Digital Avionics Systems Conference;
- [19] Reynolds, C. W. (1987) *Flocks, Herds, and Schools: A Distributed Behavioral Model*. Computer Graphics, 21(4) (SIGGRAPH '87 Conference Proceedings), pp. 25-34;
- [20] Nitschke, G. (2005) *Emergence of Cooperation: State of the Art*. Artificial Life, Vol. 11-3, pp. 367-396;
- [21] Baldassarre, G., Parisi, D. and Nolfi, S. (2006) *Distributed Coordination of Simulated Robots Based on Self-Organization*. Artificial Life, Vol. 12-3, pp. 289-311;
- [22] Eklund, J.M., Sprinkle, J.S. and Sastry, S. (2005) *Template based planning and distributed control for networks of unmanned underwater vehicles*. 44th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC'05);
- [23] Weiss, R. (Ed.) (2000) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press;
- [24] Koes, M., Nourbakhsh, I. and Sycara, K. (2006) *Constraint Optimization Coordination Architecture for Search and Rescue robotics*. Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) 2006, pp. 3977-3982;
- [25] Trianni, V. and Dorigo, M. (2006) *Self-organization and Communication in Groups of Simulated and Physical Robots*. Biological Cybernetics, 95(3), pp. 213-231;
- [26] Barrym A.M. and Dalrymple-Smith, H. (2005) *Visual Communication and Social Structure. The Group Predations of Lions*. Proceedings of International Workshop on the Modelling Natural Action Selection (MNAS05);
- [27] Perlovsky, L.I. (2004) *Integrating language and cognition*. IEEE Connections, 2(2), pp. 8-13;

[28] Perlovsky, L.I. (2006) *Neural Networks, Fuzzy Models and Dynamic Logic*. Chapter in R. Kohler and A. Mehler, eds., *Aspect of Automatic Text Analysis* (Festschrift in Honor of Burghard Rieger), Springer, Germany, pp. 363-383;

[29] Tikhonoff V., Fontanari, J.F., Cangelosi, A. and Perlovsky, L.I. (in press) *Language and Cognition Integration through Modeling Field Theory: Category Formation for Symbol Grounding*. Proceedings of ICANN06 International Conference on Artificial Neural Networks;

[30] Cangelosi, A. and Parisi, D. (2001) *Simulating the Evolution of Language*. Springer-Verlag;

[31] Marocco, D. and Nolfi, S. (in press) *Emergence of Communication in Embodied Agents Evolved for the Ability to Solve a Collective Navigation Problem*. Connection Sciences.

Web references

[W1] Joint Publication 1-02 (2007) Department of Defense Dictionary of Military and Associated Terms.

http://www.dtic.mil/doctrine/jel/new_pubs/jp1_02.pdf

[W2] Cambone, S.A., Krieg, K.J., Pace P. and Wells II, L. (2005) Unmanned Aircraft Systems Roadmap 2005-2030.

<http://uav.navair.navy.mil/roadmap05/USRoadmapAug%2005.pdf>

[W3] US Army, Class I Unmanned Aerial Vehicles (UAV),

<http://www.army.mil/fcs/uav1.html>

[W4] Defense Update, RQ-1A/MQ-1 Predator,

<http://www.defense-update.com/products/p/predator.htm>

[W5] Defense Update, Pointer Miniature Aerial Vehicle,

<http://www.defense-update.com/products/p/pointer-uav.htm>

[W6] Defense Update, Raven Miniature UAV,

<http://www.defense-update.com/products/r/raven.htm>

[W7] Directory of U.S. Military Rockets and Missiles,

<http://www.designation-systems.net/dusrm/app4/wasp.html>.

Software

The source code of the simulations, and the pre-compiled binary packages (working on Apple Mac OS X and Microsoft Windows) as well, can be found at web address <http://www.tech.plym.ac.uk/soc/research/ABC/plymay/>, within the section “Description of the Model, Results and Downloads”.