Control of a Remote Swarm of Drones/Robots Through a Local (Possibly Model) Swarm

Serge Chaumette, Frédéric Guinand

Control of a Remote Swarm of Drones/Robots Through a Local (Possibly Model) Swarm

Serge Chaumette, Frédéric Guinand

Control of a Remote Swarm of Drones/Robots Through a Local (Possibly Model) Swarm: Qualitative and Quantitative Issues

Serge Chaumette
Univ. Bordeaux, LaBRI, UMR CNRS 5800, 33405 Talence Cedex, France
serge.chaumette@labri.fr

Frédéric Guinand
Normandy Univ, UNIHAVRE, LITIS, 76600 Le Havre, France
frederic.guinand@univ-lehavre.fr

ABSTRACT

Drones (aerial, terrestrial, marine, underwater, etc.) are more and more widely used in both civilian and military scenario. Still, they remain complex systems for which training, operation preparation and execution of effective operations require adapted tools and support. In this paper we propose such a tool, that we call *Thunderbird*, based on a shadow drone that is used to control and to get feedback form a drone on an effective field of operation. In this position paper we detail a number of issues that we have identified in the design of such a tool and we describe additional problems that arise when considering not only a single drone but a swarm of possibly heterogeneous drones. We also suggest some possible ways to cope with the identified issues. We eventually present a first prototype/proof of concept that we have developed.

1 INTRODUCTION

1.1 Context

Drones are now widely used on theaters of operations, either civilian or military. By definition the fields of operation are remote (note that the proper naming of a drone is a RPAS, *i.e.* Remote Piloted Aircraft System). This is one of the many reasons why a drone based mission requires a great deal of training and preparation. Training is a major issue because the operator must understand how to operate the aircraft and what to expect in terms of feedback so as to be able to react in real time. This is especially true in configurations like those that can be encountered by military forces or even special forces. Training is not the only issue. Running a real world operation is also a major concern and some sort of support must be provided.

1.2 Our proposal

We propose an approach where what happens on the remote field of operation (called *actual field*) is controlled from a local environment (called *shadow field*) and where the feedback gained from the sensors of the drone(s) operating on the remote field is sent back to the local system.

This can be achieved either by using a real scale reconstruction of the remote field or a by using a model that could be built for instance based on a previous 3D reconstruction process. A drone, called the *shadow drone*, is operated on the model, and its behavior is reproduced by the effective drone on the field, that we call the *actual drone* (see figure 1). Our system is called *Thunderbird* by reference to the British science-fiction television series created by Gerry and Sylvia Anderson in the sixties.

Thunderbird can be used for: training; preparation of operations; supervision of real time operations; a posteriori replay of operations. This represents a real progress because, till now, situation management [9] is mainly achieved using (paper or digital) maps that can be updated to reflect the current status of the operation on the field. Still, this does not give direct control over the systems that are deployed on the operation theater, which is one of the features that we aim at supporting.

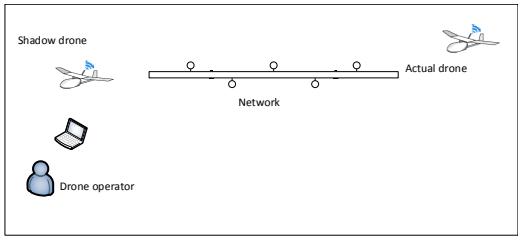

Figure 1: Global picture of the Thunderbird system

The authors have a strong background in the domain of drones and in the domain of swarming. They have lead or participated in many projects dealing with such architectures. The CARUS project [8] has been the initial experiment, and it probably was the first swarm of totally autonomous drones to really fly. Then came Asimut [4, 5] that was run within a consortium funded in the context of a project of the European Defense Agency. Based on this expertise and on our contacts with drone operators (military and civilian) we have elaborated the Thunderbird project to address the needs of the users.

2 MAJOR ISSUES

We have identified a number of key issues that have to be considered: scaling, latency, disruption and security. They are detailed bellow.

2.1 Scaling management

Scaling is the fact that the effective field of operation and the shadow area are possibly (most likely) of different sizes (see figure 2). We include in the scaling issue the fact that the shadow drone and the actual drone differ and that some sort of correspondence has to be established between each of their characteristics and features.

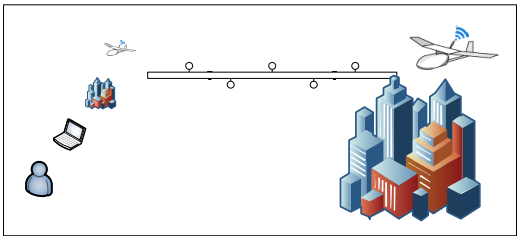

Figure 2: The scaling issue

2.1.1 Upward scaling. We call *upward scaling* the scaling that must be achieved when working with the shadow drone to map the operations to the actual drone.

Upward scaling must take the following into account:

(1) move orders have to be scaled from the shadow drone to the actual drone dimensions;
(2) move orders have to be scaled from the shadow field to the actual field dimensions.

2.1.2 Downward scaling. We call *downward scaling* the scaling that must be achieved when sending information from the actual drone back to the shadow drone.

Downward scaling must take the following into account:

(1) samples collected by sensors have to be scaled to the shadow drone sensors dimensions. It should be noted that this is not true for all kinds of sensor data, because, in a real operation, even if data collected by the actual drone must be scaled down, information should not be lost. To give one simple example, it would be critical to lose video information that would prevent the operator from detecting a potential enemy;
(2) geographic/location information have to be scaled to the shadow field characteristics;
(3) health management information have to be scaled to the shadow drone dimensions. This includes such features as the battery level for instance.

2.1.3 Thoughts on handling scaling. Addressing scaling is not as simple as it first might seem. Of course it is straightforward regarding sensor information that relate to the size of the actual area of operation. Other sensors are much more difficult to deal with. We have to address each sensor one by one and decide what to do: keep the same information for video, scale the measured values for a LIDAR, etc. A catalog of sensors should thus be defined and for each sensor it should be decided how to achieve the scaling process.

2.2 Latency management

Latency is the delay that is observed when transferring messages between the shadow drone/field and the actual drone/field (see figure 3).

2.2.1 Upward latency. We call *upward latency* the latency that is observed when communicating from the shadow drone to the actual drone.

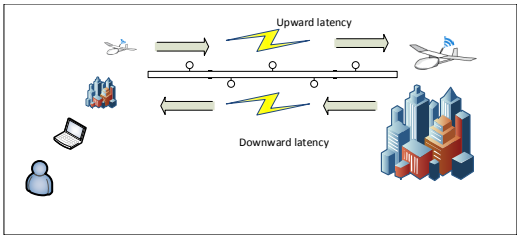

Figure 3: The latency issue

The consequences of upward latency are the following:

(1) the movements of the actual drone will take place after a certain delay once it has been done on the shadow drone. Among the possible consequences, the actual drone can crash into a wall or miss its landing spot;
(2) the operation of the payload is impacted. In a military context, the firing of a missile for instance can happen too late to reach its target and can thus cause collateral damages;
(3) the use of sensors (we consider them here separately even though they are part of the payload) can also be altered. The data that is required to be collected can possibly be collected too late, *i.e.* not when over the expected area.

The upward latency must thus be dealt with. Two directions can be considered:

(1) time stamping should be used;
(2) orders should be discarded (if they arrive 'too late') but this requires some sort of feedback synchronization with the control drone.

2.2.2 Downwards latency. We call *downward latency* the latency that can be observed when communicating from the actual drone to the shadow drone. This can lead to information reaching the shadow drone too late to make sense. For instance should the actual drone automatically make a move to avoid a missile, the corresponding move of the shadow drone could take place a bit later because of the downward latency and could lead to a crash on the shadow field.

It is most likely that similar approaches as those adopted for upward latency should be used to handle downward latency.

2.3 Disruption management

Disruption is the fact that the network becomes unavailable, broken (see figure 4). This can be temporary, but the delay involved is much bigger than what is considered when talking in terms of latency. The solution thus cannot only be to wait for the network to be restored, and for the expected messages, or even future messages to arrive. Specific processes and management policies have to be setup.

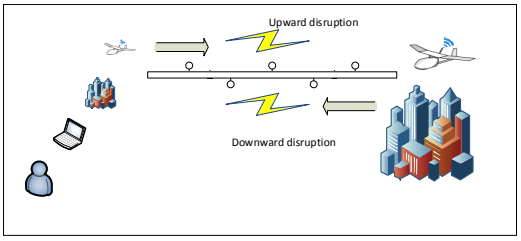

Figure 4: The disruption issue

Disruption cannot be ignored, especially in real world situations and even more when military contexts are considered. In the later case, maintaining the network operational is not only a question of health management, but it also depends on what the enemy is doing with it.

2.3.1 Upward disruption. We call *upward disruption* the disruption that can happen in the direction from the shadow drone to the actual drone.

Among the possible consequences of upward disruption, it might be the case that an an order sent from the shadow drone never reaches the actual drone.

This problem can be addressed by defining a set of orders that have no impact, when not executed, on the future orders to come. For instance, taking a picture, collecting some data, are operations that have no impact on the future orders, even though the way these data are processed may have an impact. Orders considered critical, if impossible to transfer, should lead to the abortion of the mission. As a consequence, a precise methodology must be proposed to help define a mission with as little critical operations as possible.

2.3.2 Downward disruption. We call *downward disruption* the disruption that can happen when communicating from the actual drone to the shadow drone.

Among the possible consequences of downward disruption are the following:

(1) collected data never come back from the actual drone to the shadow drone (and thus to the situation management system);
(2) for orders that require acknowledgement, the 'ack message' may never come back.

For the first problem, a hierarchy (in terms of importance) of the collected data items could be defined depending on the mission. Then, based on the classification of the lost data according to this hierarchy, the mission could either be continued without damage, possibly in degraded mode, or terminated immediately. The second problem can most likely be addressed by implementing an adapted version of the go-back-n protocol [10], that is, by numbering ack messages and validating non received ack messages as soon as a higher numbered ack message reaches the shadow drone.

2.4 Security management

When drones are considered, security is always a major concern. It should neither be easy for an attacker to take control of the drone, nor to get access to the data that have been collected.

Ensuring *upward security* is the fact of securing the communication (an thus the control of the drone) between the operator of the shadow drone and the actual drone.

Ensuring *downward security* is the fact of securing the communication (and thus the data) between the actual drone and the shadow drone/field.

We have been exploring the issues related to security within the context of several projects and we have proposed solutions to the above problems, considering both inter drone issues [1] and intra drone issues [2]. These solutions should be adapted to the context of the Thunderbird system.

3 EVALUATION THROUGH USE CASES

The goal of this project is to provide an environment that will make it possible to train drone pilots and to support operations and situation management in a real world configuration. Therefore we have defined a number of use cases that will be run with real drone operators:

(1) scenario 1: training. A drone pilot is using a ground control station and can see what the actual operation field looks like thanks to the actual drone. He/she uses the shadow model and when required he/she can get real feedback from the field.
(2) scenario 2: preparation of an operation. There is an episode of the Blind Spot American series created by Martin Gero where the terrorists have built a model of the offices of the FBI so as to prepare for an operation, replaying the attack till they felt ready to achieve it in the real life. The idea is the same except we are dealing with drones. Additionally, thanks to the actual drone, feed back from the actual field can be taken into account.
(3) scenario 3: on the field real world operation controlled from a shadow field. To run this scenario we will need to set it up with our industrial and military partners (among which Thales, the French DGA - Direction Générale de l'Armement -, fire fighters and special forces). This will provide us with real world feedback that we will be able to use to improve the system.

4 MULTI DRONES/SWARMING SCENARIO AND ISSUES

Swarms [7] are gaining interest in the industry because they offer a number of advantages, due to the fact that a swarm of drones is more that the parts that constitute it [6]. For instance, they make it possible to achieve continuous flight but also to support new capacities thanks to the combination of the sensors that they can embed.

When considering a swarm of drones (aerial, terrestrial, etc., or even a heterogeneous swarm) in the context of the system that we have described above, the situation becomes even more complex, compared to what we have described till now in this paper. We have built a basic scenario that even though extremely simple raises many issues. We describe it bellow.

Consider the following situation. Assume the IC (Intelligence Center in military operations) commands (command C1) drone number 1 to destroy an enemy located at a given position P. The IC then commands (command C2) a ground robot to go to that position P to monitor the area from there. Because of the delays it might be the case that command C2 is executed before command C1. In this case this would lead to the destruction of the ground robot by the drone: a friendly fire incident.

This example clearly shows that delays are an issue with swarms even when our system is not in play. This phenomenon will clearly be even worse with the architecture that we have presented. Additionally, delays can drift differently between the different systems depending on the shadow/actual configuration, what makes it even more difficult to deal with.

5 INITIAL PROTOTYPE

We have developed a basic prototype using off the shelf systems (figure 7). The shadow drone is a Parrot Rolling Spider (figure 5) and the target drone is a Parrot AR Drone 2.0 (figure 6).

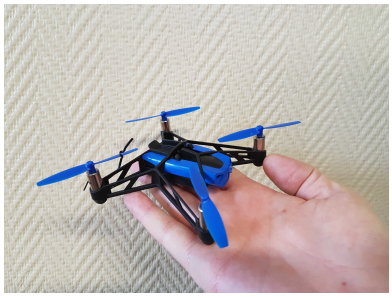

Figure 5: Parrot Rolling Spider

The duplication of information between the shadow and the local area is managed by a pair of Raspberry Pi located at each side. As of today, only some of the issues described above have been addressed in our prototype platform, but this platform constitutes an initial system on top of which we can experiment and develop the software required to address all of them. The mapping of the movements of the shadow drone to the actual drone is supported, what means that the user can pilot the actual drone by piloting the shadow drone. The camera video stream of the actual drone is sent back to the shadow location for operation control.

Figure 6: Parrot AR Drone 2.0

We first changed the operating system of the AR Drone 2.0 to boot it on our own Linux distribution called ARDroneXT [3]. Among the many services it offers it makes it possible to achieve swarming using the Parrot drones (this feature is not currently exploited by our prototype).

The Rolling Spider uses Bluetooth Low Energy (Bluetooth 4.0, aka BLE) and consequently the control could not be achieved the same way as with the ARDrone 2.0 that offers a Wi-Fi connection. An interface has thus been developed on a BLE Android mobile phone that is used as a gateway so that the commands to be sent are transmitted to the Rolling Spider using BLE (for simplicity reasons, this has not been depicted figure 7).

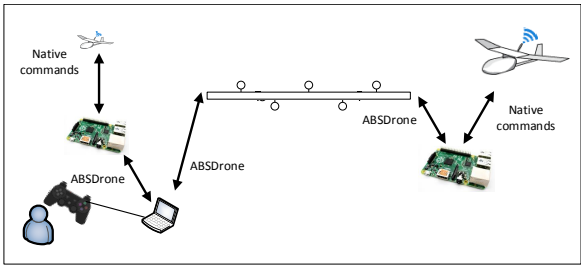

Figure 7: Prototype Thunderbird platform

To abstract from the physical drone a protocol called ABSDrone for *ABStract Drone* has been defined. It currently supports basic operations such as take off, move, land, etc. It is implemented using JSON as transport layer. We decided not to use a 'standard' protocol, like MAVLink for instance, because we also wanted to be able to use basic off the shelf drones (such as the USD 10 systems that can be found on the web), the controllers of which most of the time rely on proprietary protocols that we believe will be much easier to abstract using our protocol than using complex software layers like MAVLink.

Here is the JSON for an example command supported by ABSDrone. This command is used to move the drone in a given direction at a given speed:

```
{
"cmd": " MOVE ",
"payload": {
    "orientation" : {" LEFT "}
    "spd" : 0.5
}
```

The management of the video stream coming from the actual drone has also been implemented, so that it can be diverted to the shadow field.

6 CONCLUSION

In this paper we have described a new framework called Thunderbird that can be used to learn how to pilot a drone, to train when preparing for a field operation by repeating the process as many times as required, and to gain easier control in a real word operation to support situation management.

We have presented what we consider the major issues that need to be addressed so that the system can deal with the remote operation of a drone using a local shadow drone. We have given a number of directions that can be explored so as to cope with these issues.

We eventually described a first prototype that has been implemented at our labs. Based on this proof of concept, many directions are to be explored: definition of a model for the global system, management of the scaling issue, the delays, the disruptions, the security. The prototype also needs to be developed further to reach a state where the evaluation use cases described in this paper can be tested on the field with real operators.

ACKNOWLEDGMENTS

The authors would like to thank R. Druon, for developing the initial prototype of the system at Bordeaux University while being Bachelor Student.

They also would like to thank C. Edouard at Normandy University in Le Havre for working on the project while being Bachelor Student.

REFERENCES

[1] R. N. Akram, P.-F. Bonnefoi, Serge Chaumette, K. Markantonakis, and D. Sauveron. Secure Autonomous UAVs Fleets by Using New Specific Embedded Secure Elements. In *15th IEEE International Conference on Trust, Security, and Privacy in Computing and Communications (IEEE TrustCom-16)*, Tianjin, China, August 2016.

[2] Raja Naeem Akram, Konstantinos Markantonakis, Keith Mayes, Pierre-François Bonnefoi, Damien Sauveron, and Serge Chaumette. An efficient, secure and trusted channel protocol for avionics wireless networks. *CoRR*, abs/1608.04116, 2016.

[3] Vincent Autefage and Serge Chaumette. ArDroneXT - Ar.Drone 2 eXTension for swarming and service hosting. Research report, LaBRI - Laboratoire Bordelais de Recherche en Informatique, December 2013. This is an Operating System Setup Guide.

[4] Pascal Bouvry, Serge Chaumette, Grégoire Danoy, Gilles Guerrini, Gilles Jurquet, Achim Kuwertz, Wilmuth Müller, Martin Rosalie, and Jennifer Sander. Using Heterogeneous Multilevel Swarms of UAVs and High-Level Data Fusion to Support Situation Management in Surveillance Scenarios. In *International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI 2016)*, Baden-Baden, Germany, September 2016.

[5] Pascal Bouvry, Serge Chaumette, Grégoire Danoy, Gilles Guerrini, Gilles Jurquet, Achim Kuwertz, Wilmuth Müller, Martin Rosalie, Jennifer Sander, and Florian Segor. ASIMUT project: Aid to SItuation Management based on MUltimodal, MUltiUAVs, MUltilevel acquisition Techniques. In *3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications (DroNet)*, DroNet '17 Proceedings of the 3rd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications, pages 17 – 20, Niagara Falls, United States, June 2017.

[6] Serge Chaumette. A swarm of drones is more than the sum of the drones that make it up. In *Conference on Complex Systems (CCS2016)*, Amsterdam, Netherlands, September 2016.

[7] Serge Chaumette, Jin Hyun Kim, Kamesh Namuduri, and James P.G. Sterbenz. *UAV Networks and Communications*. Cambridge University Press, 2016.

[8] Serge Chaumette, Rémi Laplace, Christophe Mazel, Raphaël Mirault, A. Dunand, Y. Lecoutre, and J.-N. Perbet. Carus, an operational retasking application for a swarm of autonomous uavs: First return on experience. In *MILCOM 2011 - 2011 IEEE Military Communications Conference, Baltimore, MD, USA, November 7-10, 2011*, pages 2003–2010. IEEE, 2011.

[9] Gabriel Jakobson, John Buford, and Lundy Lewis. Situation Management: Basic Concepts and Approaches. In William Cartwright, Georg Gartner, Liqiu Meng, Michael .. Peterson, Vasily .. Popovich, Manfred Schrenk, and Kyrill .. Korolenko, editors, *Information Fusion and Geographic Information Systems*, Lecture Notes in Geoinformation and Cartography, chapter 2, pages 18–33. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[10] Andrew Tanenbaum. *Computer Networks*. Prentice Hall Professional Technical Reference, 4th edition, 2002.