An ubiquitous based approach for Movement Coordination of Swarms of Unmanned Aerial Vehicles using mobile networks

Bruno Olivieri[1], Markus Endler[2]

[1,2]Departamento de Informática – Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio)
Rua Marquês de São Vicente 225, 22453-900– Rio de Janeiro – RJ – Brazil

{bolivieri, endler}@inf.puc-rio.br

Abstract. *Swarms of robots have several military and civilian applications such as search and rescue missions, terrain exploration, industrial control, public security, fire combat, and several others. One of the biggest challenges of cooperative swarming in such application domains is the real-time coordination of the robot's movements in a wide-area setting, where it is expensive, or even impossible, to set up a dedicated radio communication infra-structure for the robot control. Hence, one has to resort to wide-area wireless networks. This, however, implies in a higher communication latency, which may have a significant impact on the coordination synchronicity of the swarm. In this work we tackle the swarm formation problem in wide-area settings, and propose a bandwith-efficient multi-robot coordination protocol that uses cellular 2G/3G/4G networks. This protocol has the notion of a swarm leader and is implemented on the top of an in-house developed mobile middleware with group-cast communication capability.*

1. Introduction

A coordinated swarm of robots can be important for several applications, such as search and rescue missions, surveillance and monitoring of mass events. In particular, swarms of Unmanned Aerial Vehicles (UAV) equipped with cameras can become a strategic tool for military and civilian applications. Nevertheless, in order to get a good view coverage of a site, or a specific event/occurrence, it is necessary to coordinate the movements of the robots. The other motivation for real-time coordination is to avoid collision among the robots, which may be operating on the same altitude level.

According to Şahin[1], a robot swarm consist of a large number of homogeneous, autonomous and relatively incapable or inefficient robots with local sensing and communication capabilities. Robot swarms can be non-communicating, communicating or networking. In the first case, the robots navigate completely autonomous and do not communicate with each other, whereas communicating and networking swarms keep exchanging their position information so as to allow a higher level cooperation among them, such as coordinated movement. The main distinction between communicating and networking swarms is that while the former category assumes an ubiquitous and uniform wireless communication medium, in the latter category the swarm itself establishes and maintains an ad hoc communication network by searching for and adjusting to the best radio signals. Although networking robot swarms may be mandatory for certain applications in remote locations, communicating swarms is a natural fit for urban environments where one can take advantage of the ubiquity of mobile cellular networks.

In this context, we are specifically interested in supporting swarms of UAVs equipped with cameras and used for surveillance and public security during mass events, such as concerts, festivals, political demonstrations (e.g. the recent Brazilian "Vai-pra-Rua" movement) , New Year's Eve, etc. For such events, UAVs would not only enable a wide-angle view of the event, but also the ability to quickly spot the places of turmoil and help to identify the people involved.

Consider the following hypothetic scenario: on New Year's Eve at the Copacabana beach around one million people gather in a peaceful celebration. Nevertheless, there are always reports of local foci of turmoil, robbery and rampage. In order to prevent these small incidents, the metropolitan police of Rio set up a fleet of twenty autonomously flying UAVs equipped with cameras and in charge of monitoring the event. In addition to the fly-by-wire patrol flight mode (along pre-determined trajectories), each UAV is also capable of switching from the patrol flight mode to a swarm flight mode for turmoil recording. In the swarm flight mode, several vehicles enter a flight formation around the UAV which first spotted the place of rampage. By doing so, the set of UAVs not only increment the number of cameras able to record the scene, widening the view perspective, but also become less susceptible to the risks of vehicle knock-downs, possibly attempted by some people.

We conjecture that such patrol and swarm flying modes of UAV swarms can be achieved by using smartphone-based UAV control and conventional mobile networks. This approach brings two main advantages: (1) it can readily be deployed on a wide array of commercial and well tested UAV airships; (2) it extends the UAV remote control capabilities through the use of off-the-shelf smartphones, acting as a processing unit and a radio transmitter connected to the Internet. To reach the necessary wide-area communication support required for swarm coordination, our approach relies on conventional mobile networks. These provide almost anywhere data-links through 2G, 3G and even 4G technologies, specially in urban areas.

In order to enable a swarm-like flight of UAVs there is need for some strategy to coordinate their flight movements in almost real-time. This work contributes with a coordination approach for the formation and control of UAV swarms based on mobile group-cast communication and establishment an efficient management of dynamic groups of mobile nodes (i.e. the UAVs). As main components of our approach we introduce a generic hardware architecture which employs off-the-shelf smartphones, a mobile communication middleware and a coordination protocol which uses the group communication facility of this middleware.

The remainder of the paper is structured as follows: In the next section, we present some related works. In Section 3 we then introduce the fundamental concepts related to the UAV hardware and operation. Section 4 presents our cloud-based communication middleware and its capabilities that are used by the coordination protocol. Then, in section 5 we present and discuss the proposed coordination protocol. In Section 6 we then introduce some discussions about this work. Finally in section 7 we present our plan to evaluate the coordination protocol and the next steps of our work.

2. Related Work

Semsch et al. presents an application[2] of monitoring urban areas with UAVs. In particular, the work studies how the height of buildings introduces occlusion for UAVs in a surveillance task. However, the path has to be pre-computed centrally and is supposed to be done by a single UAV. The surveillance area is divided into waypoints produced by the 3D Art Gallery algorithm[2]and then a custom TSP algorithm generates a path to be followed. The approach can accommodate more than just a single UAV, but these would fly autonomously along separate itineraries, and with no interaction between them.

Kingston et al. proposed an approach[3] to patrol a perimeter (e.g. a frontier) with a dynamic team of UAVs. They created a decentralized algorithm that works with an unknown and dynamic number of UAVs working together and where the system of UAVs adapts itself to maintain the perimeter vigilance. Their coordination algorithm is entirely distributed and runs on each UAV. Although the perimeter patrol problem is quite different from the swarm formation control problem, in both cases the UAVs need to constantly exchange their position information. But unlike our scenario, their UAVs only follow a pre-defined trajectory (the perimeter) and are not capable of following a moving target.

Weng et al. implemented[4] a distributed algorithm[4] based in the Human Immune System, which has some similarities with our surveillance scenario. Initially, the swarm of UAVs fly randomly within an area of surveillance. But as soon as some UAV, say V1, detects a moving intruder it requests help from the other UAVs within its radio range and starts following it. All the UAVs that receive the help request from V1 also change their routes to reach the intruder. As with our approach, all the helper UAVs keep receiving position updates of V1, so as to dynamically change their flight destinations in accordance to the intruder’s movements. Unlike our approach, however, inter-UAV communication is limited to the coverage of each UAV radio, (i.e. a networking swarm rather than a communicating swarm, according to Şahin[1]). And because of this limited communication range, there is no way to optimize the overall swarming task.

All those works show some limitation in the inter-UAV communication and coordination capabilities. Most of them also focus only at the patrol flight mode, without specific actions for swarm coordination for following moving targets. As part of our research, we also studied robotics middleware systems [5],[6] and looked for capabilities targeted at wide-area networks, but to the best of our knowledge, all of the systems are tailored to LAN infra-structure scenarios.

3. Hardware Architecture

At the UAV-side our approach introduces a smartphone-centered design. In this architecture, the smartphone is piggy-backed by the UAV and is responsible for local processing, wireless communication, sensing the airship’s position and converting remote flight-control commands to the control board of the airship. Virtually almost any UAV can be used, as our architecture has a modular integration component that acts as a bridge between the smartphone and the airship as it is. We require that the smartphone runs Java, has a GPS sensor, has a digital compass and has a 2G/3G/4G radio with Internet access capability. In regard to the airship we assume that it has enough energy to perform the tasks, i.e. do the takeoff and landing autonomously, and reach the swarm task area autonomously.

The presented requirements are aligned with several Hobby Remote Controlled Models (R/C), such as aircrafts, helicopters, multi-rotors and Zeppelins. For practical purposes, we chose a specific multi-rotor controller, the MultiWii quadcopters since they are based on open hardware and open source. However our approach could be easily adapted for other flight controllers.

The choice of a quadcopter as the airship is due their simple mechanic and control characteristics. As they have flight control boards, they can stay autonomously hovering over some spot and be controlled with simple commands like: UP, DOWN, BACKWARDS, FORWARDS, LEFT, RIGHT and YAW. The flight control boards are off-the-shelf components that can be acquired in Hobby-Model stores with the expected requirements.

In order to integrate the smartphone application to send commands to the UAV we design an integration component as outlined in Figure 1. It is capable to receive directions from our control protocol described later in section IV and translates them in an ordinary Remote Control commands, in Pulse With Modulation (PWM) which is a protocol widely used in R/C. The integration component has a physical interface with the smartphone with an USB cable and with the R/C with 4 jump cables, on for each channel expected by the chosen quadcopter.

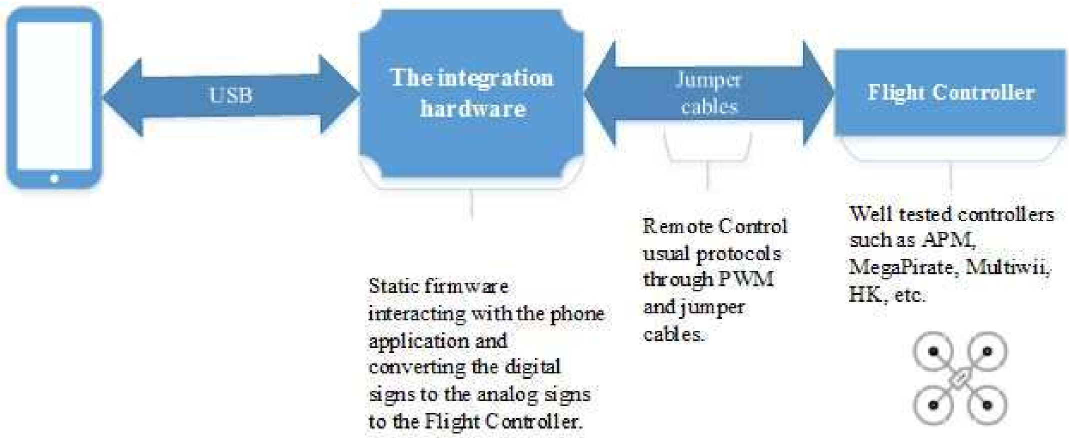

Figure 1: The hardware architecture of the UAV.

4. Communication Middleware

For the mobile communication among the UAVs, we used a communication middleware already developed in our lab. Among several other capabilities, this middleware features high-performance and reliable message delivery over the wireless medium (for any mobile node with IP connectivity), and dynamic group management and group-cast communication, which is extensively used in our coordination protocol.

The Scalable Data Delivery Layer (SSDL)[5] is a mobile communication middleware system aimed at supporting the development of applications with requirements for efficient and scalable communication among mobile nodes.

Essentially, SDDL connects stationary nodes deployed in a cluster or cloud (the "SDDL core" network) to any mobile node with an IP data connection. Within the SDDL core, it uses the OMG Data Distribution Service for Real-time Communication (DDS)[6] standard for high-performance communication. DDS is a standard from the OMG, which specifies a peer-to-peer architecture for real time Data-Centric Publish-Subscribe. DDS also has several Quality of Service (QoS) policies that can be set between producers and consumers of data (e.g. reliable communication, data persistency, priority lanes, etc.).

The SDDL extends the DDS Data-Centric Publish-Subscribe to mobile nodes through a scalable gateway approach. Each Gateway is a DDS node that is also responsible for managing the wireless connections with a large number of mobile nodes and handling in- and out-bound messages from the mobile nodes. Thus, the middleware employs two communication protocols: DDS's (Distribution Service Service) Real Time Publish/Subscribe RTPS[6] for the communication within the SDDL core, and the Mobile Reliable UDP (MR-UDP)[5][7] for the inbound and outbound communication between the SDDL core and the mobile nodes. The SDDL core can accommodate several types of nodes: Gateways, Processing Nodes (i.e. nodes that process the mobile sensor data generated by the mobile nodes, such as their geographic positions), GroupDefiners, or monitoring nodes operated by humans (i.e. Controllers), for displaying the mobile node's current position (or UAV sensor data), managing groups, and sending message to the MNs.

For enabling communication and coordination among the UAVs of a swarm, SDDL is used as follows: the smart-phone at each UAV is the MN that uses MR-UDP for bi-directional communication with any other UAV smart-phone. This communication goes through one or more Gateways of the SDDL Core, in the cloud/cluster. The GroupDefiner, yet another SDDL Core node, is responsible for managing the group membership information of all UAVs. Its main task is to check all messages exchanged among UAVs and update the UAV's group membership accordingly. More specifically, the GroupDefiner dynamically groups UAVs by their current status: Patrol mode, Leader or Slave roles (see Section 5), and is capable of managing simultaneously multiple groups in swarm mode, by using the id of the Leader UAV to separate them. Whenever a GroupDefiner detects some group membership change of an UAV it announces this to all Gateways, so that the messages can be group-cast accordingly. By this, each position update message produced by an UAV will be automatically group-cast to all the UAVs that are in the same communication group, without need for each UAV to explicitly make a join request. For more information about SDDL's dynamic group management support, the reader is referred to Vasconcelos R.[7]. Fig. 2 shows the main components of the SDDL middleware involved in the UAV communication and group management. Where the groups are presented in small clouds.

5. Proposed Coordination Protocol

In this section we present the distributed coordination protocol running at each UAV. This protocol relies on the SDDL communication middleware and its group communication and management functions, explained in section 4. The coordination protocol is leader-based in the sense that one of the UAVs steers the motion of the entire swarm around it.

5.1 System Model

In order to fully understand the coordination protocol we shall first present the assumptions of the model capabilities underlying it, i.e. the system model. The main hypothesis are in respect to the characteristics of the UAVs; of the communication links, and the positioning technology. The main assumptions are:

- Each UAVs has following sensors: GPS, a digital compass and sonars;
- Each UAV is able to carry a smart-phone;
- UAVs do not fail or deplete their battery during operation;
- The (wired) interfaces between the smart-phone, the integration module and the UAV controller are reliable and latency of control commands is negligible;
- The Flight control of the UAVs reacts reliably to steering controls such as MoveToPosition (), IncreaseSpeed(), DecreaseSpeed(), GoToAltitude(), etc.
- The wireless signal of the mobile network covers the entire geographic region of interest;
- The maximum transmission delay over the wireless link of the mobile network is δ ms.
- The system is capable of avoiding collisions.

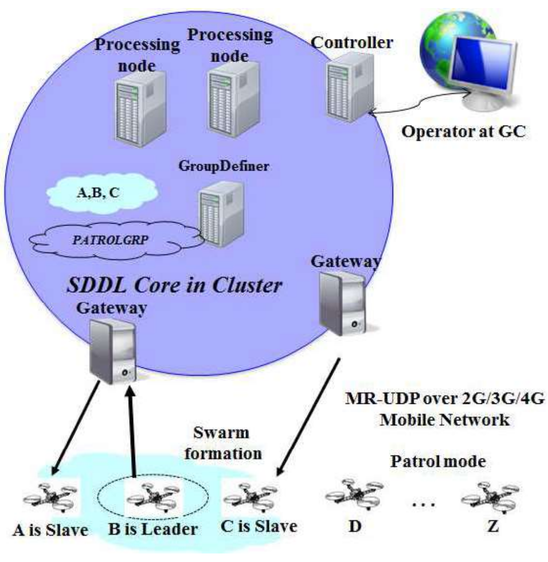

Figure 2: Components of the SDDL middleware involved in the UAV communication and group management.

5.2 Informal Description

We consider that the geographic region of interest is monitored by a set of m UAVs. All UAVs will flight either in Patrol mode or Swarm mode. In Patrol mode, the UAVs are patrolling autonomously within the virtual borders of the region of interest, previously determined by the operators, and are transmitting camera images to the operators' console (i.e. the Ground Control – GC). In this mode, the UAVs are members of the group named *PATROLGRP*, managed by the GroupDefiner. In the Swarm mode, any UAV may play the Leader or Slave role.

Whenever the Ground Control gets a relevant image from some UAV, this and other UAVs will enter the Swarm mode. For this, the operators of GC define the UAV that will assume the Leader role and a number, say n, of additional UAVs that will be requested to constitute the swarm (where $n < m$). At this moment all the UAVs in Patrol mode receive a group message (group *PATROLGRP*) with a request to send their current GPS coordinates to the UAV in the Leader role.

When the Leader receives this message, it waits for $2*\delta$ time units for the position information from the remaining UAVs, and then selects the n most appropriate UAVs that are to become part of its swarm. The criteria that determines which UAVs are the best choices can be the distance to the Leader, the airship's residual energy, or any combination of these or other airship data. The Leader UAV then sends its command informing his slaves their relative position to the selected UAVs, which enter the Slave role for this swarm readily. By intercepting these unicast messages, the GroupDefiner learns the set of UAVs that will become slaves of the Leader and puts them into the corresponding *LEADERIDGROUP*. This group is characterized by the ID of the Leader. From now on, any message sent by the Leader is automatically group-cast by SDDL to all its slave UAVs.

The swarm formation will take a form of a circle with radius r around the Leader, so as to widen the view range of the detected occurrence at ground. All n slave UAVs will be positioned on this circle with $\theta = 360/n$ degrees apart from each other. For this, each slave UAV will receive an unicast message from the Leader, containing θ and their relative position on this circle (clockwise) around the Leader's position (xL,yL), i.e. the first slave will take position $(r*\cos(\theta), r*\mathrm{sen}(\theta)$, the 2nd slave at $(r*\cos(2\theta), r*\mathrm{sen}(2\theta)$, and the i-th slave will position itself at $(r*\cos((i+1)\theta),r*\mathrm{sen}((i+1)\theta))$. All Slave UAVs acknowledge the receipt of this command and start moving to the target relative position in the circle formation around the Leader, which will be kept unchanged during the entire swarm formation. Angle θ is measured from direction North, which is known by each UAV through their compass.

From this moment on, the Leader will continuously disseminate its absolute position (xL,yL) through group-cast to *LEADERIDGROUP*. This position update message will be received by all slave UAVs of the Leader in $2*\delta$ ms, and will allow them to constantly update their absolute positions and keep the swarm formation.

Only when the operators of GC send a swarm dissolve message to group *LEADERIDGROUP*, all the UAVs will return to Patrol mode, and resume their task of ordinary patrolling, which they perform autonomously.

5.3 Pseudo-code

In the following, we illustrate the session V item B, with lower level of details, the pseudo-code proposed. All UAVs initialize with their variables myLeader as null and myStatus as Patrol. We use two message commands from SDDL: GCAST(group Id, message) and UNICAST(uav Id, message) GC send a newLeader message to the group *PATROLGRP*.

Algorithm 1: *UAV Update* Routine

```
OnNewMsg(Msg){
  If (Msg.type) == newLeader then
    If (Msg.id)! == myId then //It'll be slave
      newMsg ←
  new slaveCandidate(myId, Gps.currentPosition())
      UNICAST(Msg.id, newMsg)
    else { //It will be leader
      myStatus ← Leader
      wait 2δ for slaves candidates positions
      mySlaves ← ChooseSlaves(Msg.SwarmSize)
      for each mySlaves do {
        newMsg ← new beSlave(myId, slaveAngle)
        newMsg.slaveAngle ← ((360 ÷ Msg.n) * ++
  i) + 30
        UNICAST(mySlaves.id, newMsg)
      }
    }
  }
} //Leader receiving a slave
  else if (Msg.type) == slaveCandidate then
    SlavesCandidates.add(Msg.sender, Msg.pos)
  // A chosen slave
  else if (Msg.type) == beSlave then {
    this.myLeader ← Msg.sender
    this.myStatus ← Slave
    this.myAngle ← Msg.slaveAngle
} // Slave receiving the leader position
  else if (Msg.type) == leaderPosition then {
    tmpPos.X ← r.cos (myAngle)
    tmpPos.Y ← r.sen (myAngle)
    destination ← tmpPos ⊕ Msg.position // this is
  the (xL,yL)
    // Going to the expected location
    this.moveToPosition(destination)
}
Main(args[]){
  While true{
    if myStatus == Patrol then
      hoverarround()
    else if myStatus == Leader then {
      wait 2δ // or use a timer
      Msg.type ← leaderPosition
      Msg.position ← myGps.currentPosition()
      GCAST(LEADERIDGROUP, Msg)
    }
  }
} // ⊕ stands for the Coordination addition.
```

6. Discussion

Our approach is based in three main pillars: the smart-phone centric hardware architecture; the group-cast communication and dynamic group management capability of SDDL; and the distributed coordination protocol, each of which entail some benefits and limitations.

Since a central requirement of our work was to support UAV coordination in a wide-area setting and using a 2G/3G/4G mobile network, we decided to use a smartphone-centric architecture for the UAVs, developing a simple integration module to interface the UAV flight control board. The choice for a smart-phone as the communication hub has the advantage of a straight-forward implementation of the coordination logic (the coordination protocol), and of a modular design, but also entails some constraints. For example, the smartphone ought to be light-weight. Moreover, the wired interfaces between the smartphone, the integration module and the fight control board have to be constructed so as to be resistant to vibration, dust and moisture/rain. Moreover, the smartphone operation time is limited by its battery capacity, unless its power supply is connected to the airship's power supply.

At the communication level, our approach benefits from our middleware's reliable mobile communication and built in group-cast and group management support, which have been largely tested and used so far. Reliable message delivery to-and-from the mobile nodes (e.g. the UAVs) is guaranteed by the MR-UDP protocol, which makes several retransmission attempts for un-acknowledged packets. Moreover, SDDL's simple end efficient dynamic group management and group-cast capabilities showed to be well suited for our communication needs in the coordination protocol, and made the communication among groups UAVs quite straight-forward. Although we have already tested the performance of the group-cast communication (for up to 250 MNs) in a WLAN we still have measure how it performs in a mobile cellular network environment.

Our proposed coordination protocol is a distributed algorithm with a low message complexity, both in the phase of swarm formation, and during swarm mode flight: as soon as a new Leader is determined, it will broadcast a message to all UAVs of which some will reply, yielding a maximum of 2*(m-1) messages. Then, the Leader sends out an unicast messages for the n selected slaves (n messages) which also determines the members of the LEADERIDGROUP. To keep the UAVs flying in swarm formation, the Leader needs only to send one group-cast message (to LEADERIDGROUP) each time it changes its position, as each of its slave UAVs will be able to re-calculate their own absolute position by themselves.

Concerning UAV collision avoidance, in principle we assume that each UAV is equipped with sensors such as sonars or laser range that can be used to re-calculate the UAVs paths according to the other UAV's positions. Alternatively, collision avoidance in the swarm could also be handled using the SDDL core, but with a slightly different inter-UAV protocol: each UVA would group-cast its position and direction vector to a processing node within the SDDL core, at high frequency. This processing node would then check for possible collisions among the members of a LEADERIDGROUP, and promptly alert the corresponding UAVs whenever necessary.

However, this raises the question of which would be the wireless communication latency Rl required to implement such collision avoidance. More precisely, we have to relate it to Dm, the safe distance between UAV; the UAV cruise speed Cs; and the collision processing time Pt. Assuming that all UAVs groupcast their position+vector data at exactly the same instants, and in the worst case UAVs are heading towards each other, their relative speed is 2*Cs, then the following simple equation determines the maximum communication latency:

$$2\,.\,Rl = \frac{Dm}{2\,.\,Cs} - Pt$$

As an exercise lets consider Dm is 15m and Cs to be 10km/h (faster than an operator could see a suspect situation from the air). Then, the total period of 2Rl + Pt is around 2.8 seconds, which is plenty enough time to detect a possible collision with the communication latency used in the model.

7. Conclusion and Future Work

This work describes an approach to coordinate the movements of UAVs flying in swarm formation using on-board smart-phones and conventional mobile cellular networks as the communication infrastructure. The coordination is achieved by a message-efficient distributed protocol that uses the group-cast and group management facilities of our mobile communication middleware SDDL. We believe that our approach contributes towards the goal of managing swarms of affordable and off-the-shelf UAVs using mobile networks and using them for wide-area surveillance applications in metropolitan settings. However, our work is still in research and design phases, and we know that much remains to be done until we get a prototype that can be deployed in a real-world scenario.

At current stage, we are implementing and testing the coordination protocol that executes on the smart-phones. As next step, we will test the coordination protocol on simulated mobile nodes using SDDLs testing environment, for several scenarios and under different synthetic communication latencies. In these tests, we will analyze how accurate will be the movement synchronization among a set of simulated mobile nodes. In parallel, we will construct the Integration module and build two or three prototypes of a MultiWii quadcopter with an on-board smartphone. After this, we will make test of remote controlling such a quadcopter through communication over the mobile network. Finally, we will do swarm flight tests with these quadcopters in a real-world setting. In parallel, we will be working on fixing problems and optimizing the software running on the smart-phones. This entire research and development cycle is to be finished at the beginning of 2015.

ACKNOWLEDGMENT

This work was partially sponsored by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Ministério da Educação (MEC), Brazil.

REFERENCES

[1] H. Çelikkanat and E. Şahin, "Steering self-organized robot flocks through externally guided individuals," *Neural Comput. Appl.*, vol. 19, no. 6, pp. 849–865, Mar. 2010.

[2] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, "Autonomous UAV Surveillance in Complex Urban Environments," in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, 2009, vol. 2, pp. 82–85.

[3] D. Kingston, R. W. Beard, and R. S. Holt, "Decentralized Perimeter Surveillance Using a Team of UAVs," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1394–1404, Dec. 2008.

[4] L. Weng, Q. Liu, M. Xia, and Y. D. Song, "Immune network-based swarm intelligence and its application to unmanned aerial vehicle (UAV) swarm coordination," *Neurocomputing*, no. 0, p. -, 2013.

[5] L. David, R. Vasconcelos, L. Alves, R. André, and M. Endler, "A DDS-based middleware for scalable tracking, communication and collaboration of mobile nodes," *J. Internet Serv. Appl.*, vol. 4, no. 1, p. 16, 2013.

[6] O. Portal, "OMG Data Distribution Portal | Data Distribution Service Portal," 2014. [Online]. Available: http://portals.omg.org/dds/. [Accessed: 06-Mar-2014].

[7] R. Vasconcelos, L. David, and M. Endler, "Towards Efficient Group Management and Communication for Large-Scale Mobile Applications," *5th Int. Work. Pervasive Collab. Soc. Netw. (PerCol), co-located with Percom 2014*, vol. 1, p. to appear, Abr. 2014.