Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Effects of Packet Losses on Formation Control of Unmanned Aerial Vehicles

Luca Rosario Buonocore, * **Vincenzo Lippiello,** *
Sabato Manfredi, * **Fabio Ruggiero,** * **Bruno Siciliano** *

* *Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione, Università degli studi di Napoli Federico II, Via Claudio 21, 80125, Napoli, Italy. Email:*
{*lucarosario.buonocore,lippiell,smanfred,fabio.ruggiero, sicilian*}*@unina.it.*

Abstract: In this paper, the impact of the main packet loss models on the Unmanned Aerial Vehicle (UAV) formation control has been evaluated. A simulation environment has been built to introduce a centralized architecture, usually employed in mobile robotics to pursue global tasks, in the presence of loss models affecting the communication through robots hops. Simulation results show that the control performance in terms of tracking, collision avoidance and loss of connectivity are affected by the specific characteristics of packet loss. This suggests that the design of UAV formation control over wireless network has to be carried out strongly taking into account the effects of specific packet loss characteristics (due to wireless protocol, disturbance, traffic) on the performance quality.

1. INTRODUCTION

Development of distributed communication, computing, and control functionalities are providing the ability of monitoring and controlling complex or distributed processes by a large number of sensors, actuators, and computational units interconnected by wireless communication. These emerging network application paradigms such surveillance networks, formation flight, clusters of satellites, automated highway systems have led to the requirement of designing distributed consensus algorithms over network for estimation, detection, optimization and control (see [Ren and Atkins, 2007, Manfredi, 2013c] and references therein). One common feature of these researches is the sharing of information between agents in order to address a common objective. Algorithms solving the distributed robot coordination problem provide the means by which networks of agents can be coordinated. As VLSI (Very Large Scale Integration) technology advances and computing power grows, robots are becoming more and more intelligent, robust and power-efficient so that they possess the capabilities of communication and cooperative work [Bicchi et al., 2008]. This allows to implement wireless networked robotic system for reducing the need of human presence in dangerous applications (i.e. fire fighting, military or civilian search and rescue missions, security, surveillance). A large body of research, from various perspectives, has been produced both by ad hoc networking [Belhoul et al., 2007, Dixit et al., 2005] and by the robotics research communities to achieve self-organization and coordination of groups of robots through the use of optimization techniques and cooperative control, i.e., the one proposed by Costanzo et al. [2011], consensus algorithm by Manfredi [2013a].

Several architectures have been exploited in robotics to solve the problem about coordinate several agents (or nodes) to accomplish a certain goal. The outputs of single behaviors are combined in the centralized Null-Space Based (NSB) approach proposed by Antonelli et al. [2008, 2009] to compose a complex mission for swarms of small ground mobile robots. Such approach takes inspiration from the singularity-robust task-priority closed-loop inverse kinematic (CLIK) algorithm for industrial robots [Siciliano et al., 2008], which has been also employed by Lippiello et al. [2009], Caccavale et al. [2011, 2013] in many fields like robotic manipulation. On the other hand, Mariottini et al. [2009] present a leader-follower formation control based on uncalibrated omnidirectional cameras. A control strategy for distributed monitoring tasks trough the so-called geometric moments of the swarm is introduced by Morbidi et al. [2011]. Decentralized strategies for patrolling and monitoring have been considered by Acevedo et al. [2013a,b], including also issues given by limited communications. The design of a fast rendezvous leader-follower protocol is presented by Manfredi [2013b], in which the effect of collision phenomena on the wireless communication is taken into account. Finally, Parker [2008] gives an overview about multiple robot systems.

For all the above reasons, it might be of great interest: i) introduce a simulator to evaluate well known centralized architecture usually used in mobile robotics to pursue global tasks; ii) take into account the limitation of the

⋆ Authors are listed in alphabetical order.
This work was supported by the Network for Excellence of research programme MASTRI funded by POR Campania FSE 2007-2013.
The research leading to these results has been also supported by the ARCAS and SHERPA collaborative projects, which both have received funding from the European Community Seventh Framework Programme (FP7/2007-2013) under grant agreements ICT-287617 and ICT-600958, respectively. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

wireless communication supporting the robot cooperation. Specifically, the presence of different packet loss models affecting the communication through the robots hops are considered and their effects on wireless networked robotic performance are evaluated. It will pointed out that the effect of packet loss on the overall performance has to be taken into account in networked robotic architecture design. Few works in the literature carry out a similar analysis. For instance, Guerrero et al. [2012] analyse the impact of medium access protocol on average consensus problem over wireless networks for a group of quadrotors. That work provides a simulation environment modeling different network communication layers, analysing the impact of the Carrier Sense Multiple Access algorithm with Collision Avoidance (CSMA/CA) and Time Division Multiple Access algorithm(TDMA). As a result, the average consensus using CSMA/CA protocol presents a large convergence time due to both packets drop and time delay in the end-to-end transmission among quadrotors, while TDMA is more suitable for real-time communication.

Otherwise from Guerrero et al. [2012] who focus on the effect of channel access algorithm on the swarm performance, herein the effects on the swarm performance evaluation of the main packet loss models presented in the literature [Jiang and Schulzrinne, 2000] are considered. Additionally, further performance metrics (i.e. Maximum Agent Error, Collision Avoidance, and so on) are taken into account. In such a way, it is possible to highlight the problem of loss of communication between agents moving inside structured environments which is essential in aerial robotic applications. Using these models allows to analytically investigate and make more robust control algorithms. This analysis will lead to the choice of political communication in terms of traffic and network topology so as to improve the control performance.

The paper is organized as follows. Section 2 introduces the employed centralized architecture for the the network communication and task control. In Section 3, the considered loss models are revised. Section 4 provides a performance evaluation to understand the limitation of wireless communication in a centralized architecture. The paper concludes with final remarks and future work in Section 5.

2. MODELING OF THE CENTRALIZED ARCHITECTURE

2.1 Layout of the architecture

The use of UAVs in the applications depicted in Section 1 is growing day by day. These aerial platform are moving from “passive” tasks like surveillance, monitoring and inspection [Oh and Green, 2004, Marconi et al., 2012] into “active” tasks like grasping and manipulation [Lippiello and Ruggiero, 2012a,b].

The proposed architecture is made up of n agents, each of one representing the single UAV of the swarm. Without loss of generality, the ith vehicle is able to measure its position $\boldsymbol{p}_i \in \mathbb{R}^3$ with respect to a fixed inertia frame Σ_w, where $i = 1, \ldots, n$. Moreover, each vehicle is able to detect the position $\boldsymbol{o}_{i,k} \in \mathbb{R}^3$ of obstacles which are present within a distance ρ_i from $\boldsymbol{p}_i$, where $k = 1, \ldots, o_i$ with o_i, the total number of obstacles detected by the vehicle i. An obstacle can be both another vehicle of the swarm and a generic object in the environment that should be avoided.

Considering the dynamic model of a UAV as described by Nonami et al. [2010], it is possible to recognize that the position $\boldsymbol{p}_i$ and the so called *yaw* rotation around its vertical axis are flat outputs of the system [Fliess et al., 1995]. Therefore, without loss of generality, both the yaw rotation and the low-level control implemented by Nonami et al. [2010] might be neglected assuming that each UAV is controlled through its linear velocity.

Finally, a central node elaborates the measurements given by each vehicle with respect to a certain number of tasks and constraints to be fulfilled, providing online the desired trajectory that each UAV has to follow. More details are provided below.

2.2 NSB approach to control UAVs swarms

The goal of the NSB approach is to combine the outputs of elementary behaviors to command each robot, i.e., UAV, of the swarm. A single behavior can be defined either in a global or local fashion. If the number of degrees of freedom of the system is greater than the one required to fulfil the main behavior, other behaviors can be considered and arranged with different priorities, leading to a hierarchical framework. The approach here presented modifies what described by Antonelli et al. [2008, 2009] along the lines of the algorithm introduced by Caccavale et al. [2013].

In general, a behavior can be represented by either a function aiming to reach a desired value, i.e., the centroid of the swarm has to reach a fixed point in the space, or a cost function that should be maximized(minimized), i.e., maximize the distance of the UAVs from the obstacles. The former behaviors are then called *tasks*, the latter instead *constraints*.

Let $\boldsymbol{p} = \begin{bmatrix}\boldsymbol{p}_1^{\mathrm{T}} \cdots \boldsymbol{p}_n^{\mathrm{T}}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{3n}$ be the configuration of the system. A generic task can be represented trough the jth task variable $\boldsymbol{\sigma}_j \in \mathbb{R}^{m_j}$ to be controlled, where $j = 1, \ldots, n_t$ with n_t the total number of the tasks and m_j the dimension of the jth task variable. $\boldsymbol{\sigma}_j$ is related to the configuration of the system as follows

$$\boldsymbol{\sigma}_j = \boldsymbol{f}_j(\boldsymbol{p}), \tag{1}$$

with $\boldsymbol{f}_j \in \mathbb{R}^{m_j}$ a proper vector function. Notice that j denotes also the priority of the task, i.e., 1 the highest priority, n_t the lowest. Taking the time derivative of (1) yields

$$\dot{\boldsymbol{\sigma}}_j = \frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{\sigma}_j}\dot{\boldsymbol{p}} = \boldsymbol{J}_j(\boldsymbol{p})\dot{\boldsymbol{p}}, \tag{2}$$

where $\boldsymbol{J}_j(\boldsymbol{p}) \in \mathbb{R}^{m_j \times 3n}$ is the configuration-dependent task Jacobian matrix.

On the other hand, each constraint can be described through a cost function $\mathcal{C}_h(\boldsymbol{p})$, where $h = 1, \ldots, n_c$, with n_c the total number of constraints. Each function $\mathcal{C}_h$ can increase its value when the configuration of the system is near to violate the constraint. In order to minimize such cost function, the swarm can be moved accordingly to the descend gradient $-\nabla_{\boldsymbol{p}}^{\mathrm{T}}\mathcal{C}_h(\boldsymbol{p})$ representing a fictitious force moving away the UAVs from the configurations violating the constraints. An overall cost function can be given by

$$\mathcal{C} = \sum_{h=1}^{n_c} \gamma_h \mathcal{C}_h(\boldsymbol{p}),$$

where γ_h is a positive weight related to the hth constraint.

Considering (2) with reference to the highest priority task, i.e., $j = 1$, following the approach by Caccavale et al. [2013], in order to get both an asymptotic fulfilment of the main task and a NSB approach for the other tasks and constraints, the following velocity control input can be chosen

$$\begin{aligned}\dot{\boldsymbol{p}} =& \boldsymbol{J}_1^{\dagger}(\boldsymbol{p})\left(\dot{\boldsymbol{\sigma}}_{1,d} + \boldsymbol{K}_1 \boldsymbol{e}_{\sigma,1}\right) + \sum_{j=2}^{n_t} \boldsymbol{P}(\boldsymbol{J}_j^A(\boldsymbol{p}))\boldsymbol{J}_j^{\dagger}\boldsymbol{K}_j\boldsymbol{e}_{\sigma,j} \\ & - k_{\nabla}\boldsymbol{P}(\boldsymbol{J}_{n_t+1}^A(\boldsymbol{p}))\nabla_{\boldsymbol{p}}^{\mathrm{T}}\mathcal{C},\end{aligned} \tag{3}$$

where $\dagger$ denotes the general pseudo-inverse of a matrix, $\boldsymbol{\sigma}_{j,d}(t)$ and $\dot{\boldsymbol{\sigma}}_{j,d}(t)$ represent the desired trajectory for the jth task, $\boldsymbol{e}_{\sigma,j} = \boldsymbol{\sigma}_{j,d} - \boldsymbol{\sigma}_j$ is the task error vector, $\boldsymbol{K}_j \in \mathbb{R}^{m_j \times m_j}$ is a suitable positive definite gain matrix, k_{∇} is a positive gain, $\boldsymbol{P}(\boldsymbol{J}_j^A(\boldsymbol{p}))$ is the matrix allowing the projection of a vector onto the null space of $\boldsymbol{J}_j^A(\boldsymbol{p})$ which is the augmented Jacobian given by

$$\boldsymbol{J}_j^A(\boldsymbol{p}) = \left[\boldsymbol{J}_1^{\mathrm{T}}(\boldsymbol{p})\ \boldsymbol{J}_2^{\mathrm{T}}(\boldsymbol{p})\ \cdots\ \boldsymbol{J}_{j-1}^{\mathrm{T}}(\boldsymbol{p})\right]^{\mathrm{T}},$$

and which benefits of some robustness properties in the stability proof as highlighted by Antonelli [2009]. In order to obtain the desired motion reference $\boldsymbol{p}_d(t) = \left[\boldsymbol{p}_{d,1}^{\mathrm{T}}\ \cdots\ \boldsymbol{p}_{d,n}^{\mathrm{T}}\right]^{\mathrm{T}}$ for the UAVs, the control law in (3) can be integrated.

The geometric interpretation of (3) is that each behavior is first evaluated alone while, before adding its contribution to the overall vehicle's velocity, a lower priority task is projected onto the null space of the immediately primary task so as to avoid conflicts between them. The NSB approach always accomplishes the main task, while the fulfilment of the others depends on the specific situation and should be thus discussed case by case.

If the system is close to violate one or more constraints, a high-level supervisor has to remove some tasks so as to fulfill the constraints. The adopted management of removal/insertion of tasks is described by Caccavale et al. [2013].

The two tasks defined in this paper are listed below in a decreasing priority order. Many others can be considered [Antonelli et al., 2009].

- **Swarm centroid.** The centroid of the swarm is given by the mean value of the position of each UAV
$$\boldsymbol{\sigma}_1 = \frac{1}{n}\sum_{i}^{n} \boldsymbol{p}_i, \tag{4}$$
and it is controlled to track a desired trajectory $\boldsymbol{\sigma}_{1,d}(t)$ and $\dot{\boldsymbol{\sigma}}_{1,d}(t)$.
- **Formation control.** Let Σ_b be a frame placed at the swarm's center of mass and moving with the platoon. It is possible to associate a desired position $\boldsymbol{p}_{i,d}^b(t)$to each UAV of the swarm, expressed with respect to Σ_b. The task variable is thus defined as follows
$$\boldsymbol{\sigma}_2 = \left[\boldsymbol{p}_1^{b^{\mathrm{T}}}\ \cdots\ \boldsymbol{p}_n^{b^{\mathrm{T}}}\right]^{\mathrm{T}}, \tag{5}$$
and it is controlled to track a desired trajectory $\boldsymbol{\sigma}_{2,d}(t)$.

The following constraint is considered in this paper.

- **Obstacle avoidance.** In order to avoid collisions between two UAVs or a vehicle and an object in the environment, it is imposed that the distance between a UAV and an obstacle should be greater than a safety value ρ_s. The next cost function can be hence considered
$$\mathcal{C}(\boldsymbol{p}) = \sum_{i=1}^{n}\sum_{k=1}^{o_i} c_{i,k}(\boldsymbol{p}_i), \tag{6}$$
$$c_{i,k}(\boldsymbol{p}_i) = \begin{cases} k_{i,k}\dfrac{\rho_s - d_{i,k}(\boldsymbol{p}_i)}{d_{i,k}(\boldsymbol{p}_i)^2} & \text{if } d_{i,k}(\boldsymbol{p}_i) \le \rho_s, \\ 0 & \text{if } d_{i,k}(\boldsymbol{p}_i) > \rho_s, \end{cases} \tag{7}$$
$$d_{i,k}(\boldsymbol{p}_i) = \|\boldsymbol{p}_i - \boldsymbol{o}_{i,k}\|, \tag{8}$$
where $k_{i,k}$ is a positive gain.

2.3 Network configuration

The proposed configuration is centralized, hence it is possible to recognize a master node and several slaves. Without loss of generality, it will be considered that the master is one of the n agents. The remaining $n - 1$ agents are referred to as salves. The master computes the references for each salve as in (3) and communicates to them trough wireless channels.

Taking into account what affirmed in Section 2.1, each agent can be controlled through its linear velocity, neglecting the effects on the tracking errors deriving from the vehicle dynamics and disturbances [Siciliano et al., 2008]. This implies that the UAV might be considered as an ideal positioning device. Nevertheless, the aerial vehicle is a very complex platform, therefore each UAV agent has been modelled through the following second order system [Manfredi, 2013b, Ren and Atkins, 2007]

$$\dot{\boldsymbol{s}}_i = \begin{bmatrix}\dot{\boldsymbol{x}}_{1,i} \\ \dot{\boldsymbol{x}}_{2,i}\end{bmatrix} = \begin{bmatrix}\dot{\boldsymbol{p}}_i \\ -\omega_{n,i}^2\boldsymbol{p}_i - 2\zeta_i\omega_{n,i}\dot{\boldsymbol{p}}_i + \omega_{n,i}^2\boldsymbol{p}_{d,i}\end{bmatrix}, \tag{9}$$

with $i = 1, \ldots, n$ and where $\omega_{n,i}$ and ζ_i are the natural frequencies and damping factor for each agent. In such a way, the UAV follows the reference position with a certain dynamics given by the choice of $\omega_{n,i}$ and ζ_i.

In order to evidence the interconnections between the master and the slaves, let assume without loss of generality that the first agent is the master, while the k-the agent, with $k = 2, \ldots, n$, is a slave. Considering with $a_{ij} = \{1, 0\}$, with $i = 1, \ldots, n$ and $j = 1, \ldots, n$, the communication channel between agents i and j, which could be active $a_{ij} = 1$ or not $a_{ij} = 0$, it is possible to recognize that

$$a_{ij} = \begin{cases} 1 & i = 1, j = 2, \ldots, n \\ 0 & \text{otherwise.} \end{cases}$$

3. LOSS MODELLING

Experimental packet traces reveal that packet losses are bursty in wireless networks. In a loss burst most of the transmitted packets will be lost, while in a loss-free burst there are few or no packet losses. It is straightforward to

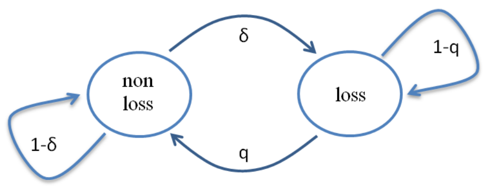

Fig. 1. Gilbert model.

use a two-state Markov chain to model the behavior of a wireless channel.

In the following, the main packet loss models will be introduced and considered in the performance evaluation. For more details, please refer to the related references [Hohlfeld et al., 2008, Tang and McKinley, 2003, Jiang and Schulzrinne, 2000].

3.1 Bernulli model

The simplest loss pattern is the Bernoulli model. Such model is constituted by a single loss parameter and each transmission is considered independent from the others. The loss probability is a parameter representing the measure of goodness about the transmission channel. Due to the independence of individual packages, this model can not take into account the channel memory and hence impulse noise. In the practice, the model can act in two different conditions, namely, *active link* with probability δ and *broken link* with probability $1 - \delta$.

3.2 Gilbert model

Gilbert model (1-st Markov chain model) is frequently employed in the study of packet-loss process in communication networks. In such a model each channel maintains a status flag which can be either labelled as *good* or *bad*. In the former state there is very few packets loss, while in the latter most packets are lost. Gilbert model has several parameters: temporal error correlation, probabilities of a good or bad channel, probabilities of error given that the channel is in the good or the bad state. In order to mitigate the computational complexity, a simplified version of the Gilbert-Elliot model is also employed. In this version, if a packet loss occurs while the channel is in the good state, the channel immediately switches to the bad state. Similarly, whenever a packet is received successfully while the system is in the bad state, the channel switches to the good state. Let assume the probabilities of staying in the good state and the bad state are the same proposed by Jiang and Schulzrinne [2000]. On the one hand, if the current packet has been successfully received, the probability that the next incoming packet is lost is denoted by δ. On the other hand, if the current packet has been lost, the probability that the current packet is successfully received is denoted by q. Therefore, the probabilities $1-\delta$ and $1-q$ are the conditional probabilities to remain in the good or bad state, respectively. The model is depicted in Figure 1. Typically, it happens that $\delta + q < 1$: in the case $\delta + q = 1$, the Gilbert model reduces to the Bernulli one. In this work, the simplified version of the Gilbert-Elliot model is considered.

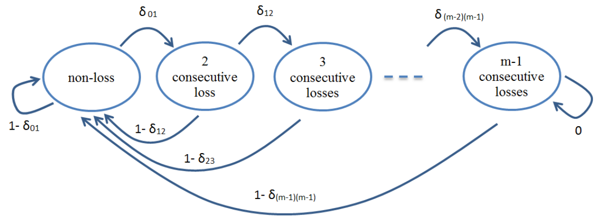

Fig. 2. A m-th order extended Gilbert model

3.3 Extended Gilbert model

This Extended Gilbert model presents one good and $m-1$ bad states to generate a hyper-geometrical distribution of the duration of the good and the bad phases with specific transmission as approximation of a wireless channel. Unlike the general Markov model assuming all past events that might affect the future state, in an extended Gilbert model only the past m consecutive packet loss will affect the future state. Figure 2 illustrates how the extended Gilbert model works. Such a model is an simplification of a general M-state model. A hyper-geometrical distribution is an useful approximation of loss burst with long term correlation, which are often observed on error prone wireless links.

4. PERFORMANCE EVALUATION

The behavior of a centralized algorithm in presence of noise in the communication of individual wireless channels a_{ij} is highlighted.

A swarm of $n = 6$ UAVs is considered in the simulations, resulting in a hexagonal configuration. The main task is to control the centroid of the swarm to follow a desired trajectory in an environment where some obstacles might get in the way. Hence, the second task is to keep the formation in a hexagonal configuration without violating the constraint to be far from such obstacles.

For the following simulations, a Gilbert model with six states has been employed. As shown by Xunq et al. [2005], the 3rd order Extended Gilbert Model model might be considered as a good approximation. The six states to correctly approximate the actual pattern of the network and the values of the model transition matrix are derived from Jiang and Schulzrinne [2000].

In case no issues in the communication channels are present, the algorithm will successfully achieve the desired tasks. However, in the practice, the algorithm suffers from the loss of packets in the communication network due to high amount of data exchanged between the UAVs. Therefore, the control algorithm has been tested in various scenarios, in which the complexity of each scenario has been constantly increased so as to stress the algorithm and to bring to light the behavior of the agents.

4.1 Technical details

The swarm starts with a regular hexagon formation in which the distance of each agent from the center of the formation is equal to 1m. The initial position of the centroid of the swarm starts at the origin $x = y =$

$z = 0$ and the final position is located at $x = 5$m. The coordinates y, z of the swarm's centroid are kept the same, without loss of generality, for all the simulation The initial and final velocities of the swarm centroid are set to zero. The simulation lasts 100 seconds: in this time, the swarm's centroid has to move from the starting point to the end one along a planned linear trajectory. The controller sample time has been selected as 0.1s.

The natural frequencies and damping factor in (9) for each agent has been set to 5rad/s and 0.65, respectively. The distance ρ_i in which each agent is able to detect obstacles has been set to 0.5m.

The tasks and the constraints employed in these simulations have been already introduced in Section 2.2. The following gains have been hence chosen for the two tasks $\boldsymbol{K}_1 = 30\boldsymbol{I}_3$ and $\boldsymbol{K}_2 = 5\boldsymbol{I}_{18}$, respectively, where $\boldsymbol{I}_\nu$ is a $(\nu \times \nu)$ identity matrix. Gain k_∇ in (3) has been set to 1. The threshold ρ_s in (7) for the obstacle avoidance constraint has been selected equal to 1m, while $k_{i,k} = 20$, with $i = 1, \ldots, n$ and $k = 1, \ldots, o_i$.

In the following evaluation three main performance metrics will be considered, namely:

- *Maximum Agent Error* (MAE). It is the maximum among the maximum errors of each agent with respect to their reference values. This index is then equals to $\max\{\max \|\boldsymbol{p}_{d,1} - \boldsymbol{p}_1\|, \ldots, \max \|\boldsymbol{p}_{d,n} - \boldsymbol{p}_n\|\}$.
- *Maximum Agent-Center Distance* (MAC). It is the maximum among the maximum distances reached by each agent from the center of the swarm.
- *UAV collision*. This parameter is set to *yes* whether some collisions happen between the agents themselves or with an obstacle; otherwise it is set to *no*. Collisions are verified a posteriori by looking at the path of each agent. The effects of a collision on the agent's path after the impact have not been considered in the simulations.

Obviously, the employed loss models are probabilistic. Hence, the data that will be reported are just estimations of the average performance. However, this arrangement allows the identification of a threshold distance transmission, within which the UAVs are considered close enough to participate in the communication. Beyond this threshold, agents could be considered virtually lost. Hence, in order to emulate the loss of connectivity between an agent and the master, a node might be considered lost if it overcomes a certain distance from the swarm. Such a value has been set at $\sigma_t = 10$m in these simulations. If the master is lost, the mission is considered failed. Moreover, as an assumption, the master is never lost in the performed simulations.

Without loss of generality, in the simulations the UAVs move in a plane, i.e. $z = 0$. Moreover, the obstacles are considered as points in the space. Three scenario have been then considered in the following simulations. In the former, just one obstacle has been randomly put in the environment. In the second scenario, instead, a set of obstacles randomly get in the way during the movement of the swarm. In the latter, a corridor has been emulated through two walls made up of two straight lines full of obstacles separated themselves by very small distances: each agent of the swarm has to remain inside the corridor avoiding to be too close with respect to the walls, the others agents and other two obstacles.

4.2 Scenario 1. Single obstacle.

In case of ideal communication, the swarm perfectly behaves in presence of a single obstacle. The agents deviate from the desired path of a small amount necessary to avoid the obstacle. This behavior is summarized in Table 1. Already in this simple scenario, the insertion of loss models results in lower performance. This phenomenon is amplified especially in models with memory, in fact a prolonged loss of communication involves an accumulation of error that must be recovered from the controller.

Model	MAE [m]	MAC [m]	UAV Collision
Ideal	0.41	1.32	No
Bernulli	0.42	1.28	No
Gilbert	0.67	1.35	No
Gilbert n	0.83	1.73	No

Table 1. Simulation results about Scenario 1.

4.3 Scenario 2. Cluster of obstacles.

In this case study, the structure of the obstacles has been complicated with respect to the previous case. The presence of multiple obstacles at the same time worsens performance. Table 2 summarizes the obtained results and shows that, in the presence of data loss, some agents might be considered lost since they are too far from the master. Some collisions may also happen.

Figure 3 shows the time histories of the distance of each agent from the center of the swarm. In such a figure no package losses are considered in the network, hence no issues are revealed. Figure 4 shows a 3D plot of the trajectory for each agent. On the other hand, when simple Gilbert model for data loss is considered, some agents might exit the swarm, as depicted in Figure 5.

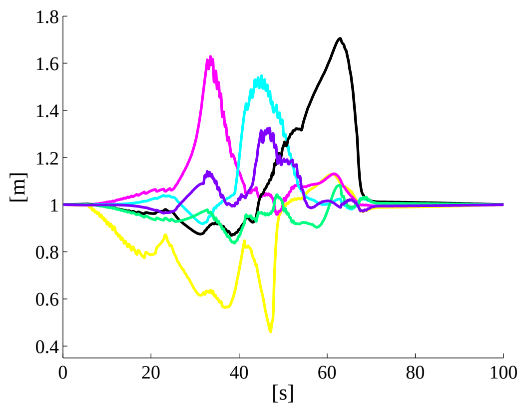

Fig. 3. Time history about the distance of each agent from the center of the swarm. Each color identifies an agent. The threshold σ_t over than an agent is considered lost has been not depicted in this plot.

4.4 Scenario 3. Corridor.

The flight environment is now different: a corridor has been simulated and the swarm has to flight inside it where two

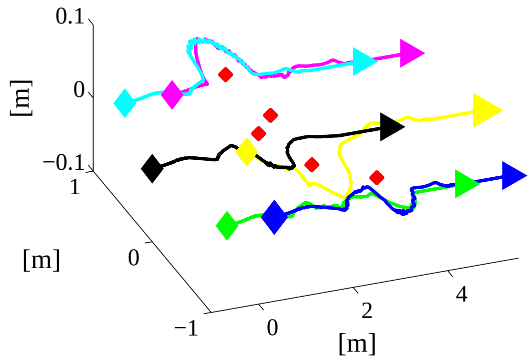

Fig. 4. Trajectory for each agent in the 3D space. Each color identifies an agent. The coloured diamonds are the initial positions of each agent's path, while the coloured triangles are the arrival points. Red diamonds are the considered obstacles.

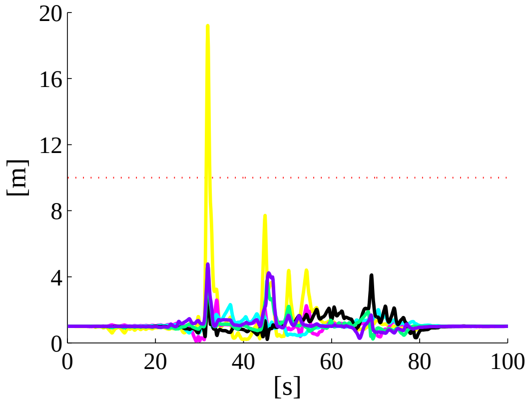

Fig. 5. Time history about the distance of each agent from the center of the swarm. Each color identifies an agent. A Gilbert model has been considered in such a case. The dashed red line represents the threshold σ_t.

Model	MAE [m]	MAC [m]	UAV Collision
Ideal	1.07	1.69	No
Bernulli	1.34	2.30	No
Gilbert	20.12	19.23	Yes
Gilbert n	1.68	2.04	No

Table 2. Simulation results about Scenario 2.

obstacles have been placed. The corridor's length is about 1m and it is passed through by the swarm of UAVs after about 2 meters from the beginning of the task, and thus when the platoon is already in a hexagonal formation.

The simulations, summarized in Table 3, show a clear difference between the case of perfect communication (see Fig. 6) and the implemented loss models. Even with simple models, the dependence from data loss is evident.

Model	MAE [m]	MAC [m]	UAV Collision
Ideal	0.47	1.03	No
Bernulli	53.2	53.35	Yes
Gilbert	197.06	197.9	Yes
Gilbert n	89.89	89.94	Yes

Table 3. Simulation results about Scenario 3.

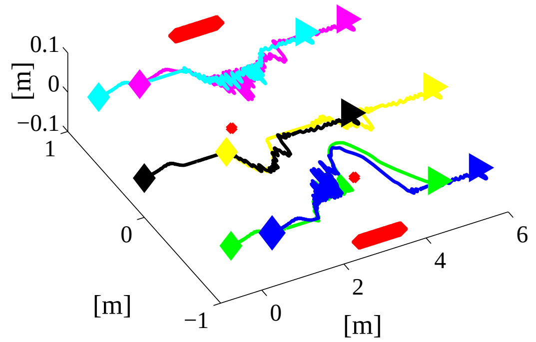

Fig. 6. Trajectory for each agent in the 3D space. Each color identifies an agent. The coloured diamonds are the initial positions of each agent's path, while the coloured triangles are the arrival points. Red spots are the considered obstacles. Red continuous lines represent the walls of the considered corridor. From the bevelled path of certain UAVs, it is evident how the obstacle avoidance constraint acts when an agent passes between two obstacles inside the same security threshold ρ_s.

5. CONCLUSION AND FUTURE WORK

In this paper, the inclusion of even simple loss models in centralized control algorithm to control agents' swarms leads to inevitably issues regarding the behavior of the agents and possible collisions between them. Although the performed analysis is just qualitative, due to the probabilistic nature of the loss models which generate random results, the manifestation of problems due to the loss of information in a network of agents coordinated by a central node has been highlighted. This is mainly due to the fact that the central node, e.g. the master, is subjected to a large amount of packages to manage, creating a bottleneck. Moreover, the possible loss of the master would lead to the conclusion of the mission.

In future work, the degradation of communications due to the traffic of individual agents will be addressed, considering a model of loss which also contains the collisions due the hight number of agents. Moreover, the alienation of a single agent from the swarm will be considered: in such a case, this agent should perform an emergency landing, while the swarm should be reconfigured to keep the reference trajectory. Future plans will also consider distributed algorithms.

REFERENCES

J.J. Acevedo, B.C. Arrue, J.M. Diaz-Banez, I. Ventura, I. Maza, and A. Ollero. A decentralized strategy to ensure information propagation in area monitoring missions with a team of UAVs under limited communications. In *International Conference on Unmanned Aircraft Systems*, pages 565–574, Atlanta, GA, USA, 2013a.

J.J. Acevedo, B.C. Arrue, I. Maza, and A. Ollero. A distributed approach for coverage and patrolling missions with a team of heterogeneous aerial robots under communication constraints. *International Journal of Advanced Robotic Systems*, 28(10):1–13, 2013b.

G. Antonelli. Stability analysis for prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. *IEEE Transactions on Robotics*, 25(5):985–994, 2009.

G. Antonelli, F. Arrichiello, and S. Chiaverini. The null-space-based behavioral control for autonomous robotic systems. *Journal of Intelligent Service Robotics*, 1(1): 27–39, 2008.

G. Antonelli, F. Arrichiello, and S. Chiaverini. Experiments of formation control with multirobot systems using the null-space-based behavioral control. *IEEE Transactions on Control Systems Technology*, 17(5): 1173–1182, 2009.

A. A. Belhoul, Y. A. Sekercioglu, and N. Mani. Mobility protocols and RSVP performance in wireless IPv6 networks: Shortcomings and solutions. *Wireless Communications and Mobile Computing*, 8(9):1183–1199, 2007.

A. Bicchi, A. Danesi, G. Dini, S. La Porta, L. Pallottino, I.M. Savino, and R. Schiavi. Heterogeneous wireless multirobot system. *IEEE Robotics and Automation Magazine*, 15(1):62–70, 2008.

F. Caccavale, V. Lippiello, G. Muscio, F. Pierri, F. Ruggiero, and L. Villani. Kinematic control with force feedback for a redundant bimanual manipulation system. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4194–4200, San Francisco, CA, 2011.

F. Caccavale, V. Lippiello, G. Muscio, F. Pierri, F. Ruggiero, and L. Villani. Grasp planning and parallel control of a redundant dual-arm/hand manipulation system. *Robotica*, 31(7):1169–1195, 2013.

C. Costanzo, V. Loscri, E. Natalizio, and T. Razafindralambo. Nodes self-deployment for coverage maximization in mobile robot networks using an evolving neural network. *Computer Communications*, 35(9):1047–1055, 2011.

S. Dixit, E. Yanmaz, and O.K. Tonguz. On the design of self-organized cellular wireless networks. *IEEE Communications Magazine, Feature Topic: Advances in Self-Organizing Networks*, 43(7):86–93, 2005.

P.M.M. Fliess, J. Levine, and P. Rouchon. Flatness and defect of nonlinear systems: Introductory theory and examples. *International Journal of Control*, 61(6):1327–1361, 1995.

J.A. Guerrero, Y. Challal, and P. Castillo. *Impact of wireless medium access protocol on the quadrotor formation control*. John Wiley & Sons, Inc., Hoboken, NJ, USA, 2012.

O. Hohlfeld, R. Geib, and G. Hasslinger. Packet loss in real-time services: Markovian models generating QoE impairments. In *16th International Workshop on Quality of Service*, pages 239–248, Enschede, NL, 2008.

W. Jiang and H. Schulzrinne. Modelling of packet loss and delay and their effect on real-time multimedia service quality. In *10th International Workshop on Network and Operating System Support for Digital Audio and Video*, pages 1–9, Chapel Hill, NC, USA, 2000.

V. Lippiello and F. Ruggiero. Exploiting redundancy in Cartesian impedance control of UAVs equipped with a robotic arm. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3768–3773, Vilamoura, P, 2012a.

V. Lippiello and F. Ruggiero. Cartesian impedance control of a UAV with a robotic arm. In *10th International IFAC Symposium on Robot Control*, pages 704–709, Dubrovnik, HR, 2012b.

V. Lippiello, F. Ruggiero, and L. Villani. Exploiting redundancy in closed-loop inverse kinematics for dexterous object manipulation. In *IEEE International Conference on Advanced Robotics*, Munich, D, 2009.

S. Manfredi. Design of a multi-hop dynamic consensus algorithm over wireless sensor networks. *Control Engineering Practice*, 21(4):381–394, 2013a.

S. Manfredi. An algorithm for fast rendezvous seeking of wireless networked robotic systems. *Ad Hoc Networks*, 11(7):1942–1950, 2013b.

S. Manfredi. On global and local consensusability of multi-agent systems with input constraint and uncertain initial conditions. In *2013 American Control Conference*, pages 6102–6107, Washington, DC, USA, 2013c.

L. Marconi, F. Basile, G. Caprari, R. Carloni, P. Chiacchio, C. Huerzeler, V. Lippiello, R. Naldi, N. Janosch, B. Siciliano, S. Stramigioli, and E. Zwicker. Aerial service robotics: The AIRobots perspective. In *2nd International Conference on Applied Robotics for the Power Industry*, Zurich, CH, 2012.

G.L. Mariottini, F. Morbidi, D. Prattichizzo, N. Vander Valk, G.J. Pappas, and K. Daniilidis. Vision-based localization for leader-follower formation control. *IEEE Transactions on Robotics*, 25(6):1431–1438, 2009.

F. Morbidi, R.A. Freeman, and K.M. Lynch. Estimation and control of UAV swarms for distributed monitoring tasks. In *Proceedings of the American Control Conference*, pages 1069–1075, San Francisco, CA, 2011.

K. Nonami, F. Kendoul, S. Suzuki, and W. Wang. *Autonomous Flying Robots. Unmanned Aerial Vehicles and Micro Aerial Vehicles*. Springer-Verlag, Berlin Heidelberg, D, 2010.

P. Oh and W. Green. CQAR: Closed quarter aerial robot design for reconnaissance, surveillance and target acquisition tasks in urban areas. *International Journal of Computational Intelligence*, 1(4):353–360, 2004.

L.E. Parker. Multiple mobile robot systems. In B. Siciliano and O. Khatib, editors, *Springer Handbook of Robotics*, pages 921–941. Springer, 2008.

W. Ren and E. Atkins. Distributed multi-vehicle coordinated control via local information exchange. *International Journal of Robust and Nonlinear Control*, 17 (10-11):1002–1033, 2007.

B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo. *Robotics: Modelling, Planning and Control*. Springer, London, UK, 2008.

C. Tang and P.K. McKinley. Modeling multicast packet losses in wireless LANs. In *Proceedings of the 6th ACM international workshop on Modeling analysis and simulation of wireless and mobile systems*, pages 130–133, New York, NY, USA, 2003.

Y. Xunq, Modestino J.W., and X. Tian. The accuracy of Gilbert model in predicting packet-loss statistic for single-multiplexer network model. In *24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 4, pages 2602–2612, Miami, FL, USA, 2005.