

Decentralized High Level Controller for Formation Flight Control of UAVs

Mark Bastourous
Normandie Univ. UNIHAVRE
LITIS lab
Le Havre, France
mark.bastourous@univ-lehavre.fr

François Guérin
Normandie Univ. UNIHAVRE
GREAH lab
Le Havre, France
francois.guerin@univ-lehavre.fr

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

Eric Lemains
Normandie Univ. UNIHAVRE
GREAH lab
Le Havre, France
eric.lemains@univ-lehavre.fr

Abstract — The main contribution of this paper is the design of a decentralized and tuning-less high level controller able to maintain without tracking errors a Leader-Follower (LF) configuration in case of lack or degraded communications (latencies, loss...) between the leader and followers UAVs. The high level controller only requires simple tunings and rests on a predictive filtering algorithm and a first order dynamic model to recover an estimation of the leader UAV velocities and avoid the tracking errors.

Keywords: *Distributed control, Formation control, Robotics, Autonomous vehicles, UAVs*

I. INTRODUCTION

Multi-Robot Systems (MRS) have been widely studied since the 90s [1], [2]. Many advantages can be expected from using a group of robots: increase and share of the payload [3], reduction of the time needed for the achievement of a task [4], fault tolerance and resilience of the system [5], [6], or use of simpler and cheaper robots for adaptability to the environment [7]. However, making a group of robots evolving together in the same environment entails the resolution of some problems. Coordination is one of the most critical ones, especially considering decentralized settings. Coordination between the members of the group can be achieved through various ways, behavior or force-based methods, Virtual Rigid Body (VRB) structure building, or Leader-Follower (LF) schemes. In the present work, we focus on the last one and investigate the problem of maintaining such a configuration in case of degraded communications between the leader and the followers. The question of building and maintaining such a formation without communication has been addressed many times during the last two decades and most of the time the proposed solution relies on a sensor-based mechanism for compensating the absence of communication. Very often, works are considering cameras; pan controlled camera [8] or omnidirectional camera like in [9], [10], or other types of sensors like Kinect [11]. From the data obtained by the

sensors, each follower of the formation attempts to estimate the pose of the global or of its local leader. In [8], the authors consider this problem with a pan-controlled camera. Since no communication occurs, the leader's velocities are unknown. The goal of their work is the design of both an adaptive formation controller and an adaptive camera controller. The authors in [10] proposed a method for LF formation control considering a robotic system in which each follower is equipped with an uncalibrated omnidirectional or perspective camera. No communications are allowed and the velocity of the leader is considered unknown as well. In that work the authors present an adaptive estimator based on several feature points. In [11], the goal is to build and maintain a LF formation without communication between the robots. Each robot is equipped with only one type of sensor, a Kinect, for estimating the pose (relative orientation and position of the leader). The choice of the Kinect is justified by the fact that it provides color and depth information from which can be directly deduced angles and distances measurements but it suffers from a limited field of view. The controller depends on both the visibility maintenance and the minimization of position errors.

With respect to these previous works, our main contribution is the design of a decentralized and tuning-less high level controller able to maintain a LF configuration. The proposed controller combines an omnidirectional camera with filtered and predicted measurements and a first order dynamic model of the UAV to get the leader velocities (even in case of degraded communications (latencies, loss) between the leader and followers UAVs) and avoid the tracking errors.

II. PROBLEM STATEMENT

The goal is to construct a geometrical shape sometimes known as Virtual Rigid Body (VRB) in order to establish the desired formation. The virtual structure between several robots aims at facilitating the agile control of several multi-rotor formations. The goal of each follower UAV is to follow the leader by keeping predefined separation distances X_{SPi} ,

Y_{SP1} , Z_{SP1} as illustrated on Fig.1. The leader UAV is assumed to track perfectly a virtual target which moves along a defined trajectory (straight line, circle ...). The controller of the leader UAV is not described in this work. The description of the high level controller will be done for the follower UAV number 1. The vision-based system provides the distance (d_l) and the bearing angle (θ_l) of the leader UAV.

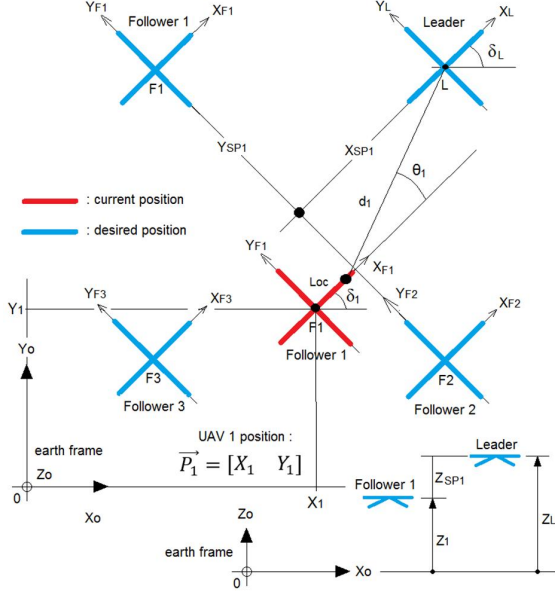


Figure 1. Problem statement

III. DESIGN OF THE HIGH LEVEL CONTROLLER

A. Double Exponential Smoothing (DES) algorithm

The proposed method assumes the availability of an estimation of both the distance (d_l) and of the bearing angle (θ_l) whose measurements are naturally degraded by noise and other uncertainties. A filtering stage to overcome these problems is mandatory. Such algorithm has to be fast, accurate, robust to different motions, and easy to implement. We assume that the leader UAV moves regularly over time with a trend. In [12],[13], a comparison is drawn between Double Exponential Smoothing (DES) algorithm and Kalman filter. The authors show that DES algorithm runs faster than Kalman filter with equivalent prediction performances and simpler implementations. That is why DES algorithm can be a good choice to generate smoothed estimates and forecast of distance (d_l) and bearing angle (θ_l). The DES algorithm at time instant n . T_e is the sampling period and n is the discrete-time index, implemented for both the distance (d_l) and bearing angle (θ_l) measurements is given as follows:

$$\hat{d}_{1n} = \alpha_{d1} \cdot d_{1n} + (1 - \alpha_{d1}) \cdot (\hat{d}_{1n-1} + b_{d1n-1}) \quad (1)$$

$$b_{d1n} = \lambda_{d1} \cdot (\hat{d}_{1n} - \hat{d}_{1n-1}) + (1 - \lambda_{d1}) \cdot b_{d1n-1} \quad (2)$$

$$\hat{\theta}_{1n} = \alpha_{\theta1} \cdot \theta_{1n} + (1 - \alpha_{\theta1}) \cdot (\hat{\theta}_{1n-1} + b_{\theta1n-1}) \quad (3)$$

$$b_{\theta1n} = \lambda_{\theta1} \cdot (\hat{\theta}_{1n} - \hat{\theta}_{1n-1}) + (1 - \lambda_{\theta1}) \cdot b_{\theta1n-1} \quad (4)$$

where: d_{1n} , θ_{1n} are the values of d_l and θ_l at n th sample.

\hat{d}_{1n} , $\hat{\theta}_{1n}$ are the smoothed values of d_{1n} and θ_{1n} .

b_{d1n} , $b_{\theta1n}$ are the trend values of d_{1n} and θ_{1n} .

Equations (1) and (3) smooth the values of the sequence of measurements by taking into account the trend, whilst (2) and (4) smooth and update the trend. The m ($m > 0$) steps ahead forecast of the next distance (\hat{d}_{1n+m}) and bearing angle ($\hat{\theta}_{1n+m}$) estimations are:

$$\hat{d}_{1n+m} = \hat{d}_{1n} + m \cdot b_{d1n} \quad \hat{\theta}_{1n+m} = \hat{\theta}_{1n} + m \cdot b_{\theta1n} \quad (5)$$

The initial values given to \hat{d}_{1n} , $\hat{\theta}_{1n}$, b_{d1n} and $b_{\theta1n}$ are:

$$\hat{d}_{11} = d_{11}, \hat{\theta}_{11} = \theta_{11}, b_{d11} = d_{12} - d_{11}, b_{\theta11} = \theta_{12} - \theta_{11} \quad (6)$$

Usually, α ($0 \leq \alpha \leq 1$) is called the data smoothing factor and λ ($0 \leq \lambda \leq 1$) is called the trend smoothing factor. A compromise has to be found for the values of α and λ . High values make the DES algorithm follow the trend more accurately whilst small values make it generate smoother results. For the follower UAV 1, the filtered (7) and predicted (8) distance errors (x, y axes) are the following ones (Fig.1):

$$\hat{e}_{1x} = \hat{d}_1 \cdot \cos(\hat{\theta}_1) - X_{SP1} \quad \hat{e}_{1y} = \hat{d}_1 \cdot \sin(\hat{\theta}_1) - Y_{SP1} \quad (7)$$

$$\tilde{e}_{1x} = \tilde{d}_1 \cdot \cos(\tilde{\theta}_1) - X_{SP1} \quad \tilde{e}_{1y} = \tilde{d}_1 \cdot \sin(\tilde{\theta}_1) - Y_{SP1} \quad (8)$$

B. Design of the tuning-less high level controller

We assume that the low level controllers (inner loop) implemented in the UAVs allow them to reach the desired references velocities (according to x and y axes) and the desired references positions (according to z axis (altitude, yaw angle)) with a short response time and without static errors, damping or oscillations. That means that: $\vec{V}_1 = \vec{U}_1$ with $\vec{V}_1 = [V_{1x} \ V_{1y}]$ the velocity vector of the follower UAV 1 and $\vec{U}_1 = [U_{1x} \ U_{1y}]$ its control inputs. All the UAVs are assumed to fly at the same altitude with the same yaw angle (altitude and yaw control can be controlled separately and are not detailed in this work). The control objective is to regulate the distance errors as follows:

$$\lim_{t \rightarrow \infty} \hat{e}_{1x}(t) = 0 \quad \lim_{t \rightarrow \infty} \hat{e}_{1y}(t) = 0 \quad (9)$$

To fulfill this control objective, we propose the following error dynamics equation:

$$\begin{bmatrix} \dot{\hat{e}}_{1x} \\ \dot{\hat{e}}_{1y} \end{bmatrix} = - \begin{bmatrix} A(\hat{e}_{1y}) \cdot f(\bar{D}_x, \sum_{i=1}^3 \beta_{xi}, \bar{V}_x - V_{Lx}, \bar{D}_x) \\ f(\bar{D}_y, \sum_{j=1}^3 \beta_{yj}, \bar{V}_y - V_{Ly}, \bar{D}_y) \end{bmatrix} = R \cdot \vec{V}_L^T - \vec{V}_1^T \quad (10)$$

$$R = \begin{bmatrix} \cos(\delta_L - \delta_1) & -\sin(\delta_L - \delta_1) \\ \sin(\delta_L - \delta_1) & \cos(\delta_L - \delta_1) \end{bmatrix} \quad A(\hat{e}_{1y}) = e^{-Ay \cdot |\hat{e}_{1y}|}$$

$\vec{V}_L = [V_{Lx} \ V_{Ly}]$ is the leader velocity vector. Ay is an attenuation parameter which makes decrease the effect of the linear control (x axis) when the lateral error (y axis) is not null.

β_{xi} ($\in [-1..+1]$) and β_{yj} ($\in [-1..+1]$) are defined as follows:

$$\begin{aligned} \beta_{x1} &= f\left(\frac{e_{x1}}{\bar{D}_X}, 1, 1\right) & \beta_{x2} &= f\left(\frac{e_{x2}}{\bar{D}_X}, 1, 1\right) & \beta_{x3} &= f\left(\frac{e_{x3}}{m \cdot Te \cdot \bar{V}_X}, 1, 1\right) \\ \text{with: } e_{x1} &= \hat{e}_{1x} & e_{x2} &= \frac{\hat{e}_{1x} + \hat{e}_{1x}}{2 \cdot m} & e_{x3} &= \bar{e}_{1x} - \hat{e}_{1x} \\ \beta_{y1} &= f\left(\frac{e_{y1}}{\bar{D}_Y}, 1, 1\right) & \beta_{y2} &= f\left(\frac{e_{y2}}{\bar{D}_Y}, 1, 1\right) & \beta_{y3} &= f\left(\frac{e_{y3}}{m \cdot Te \cdot \bar{V}_Y}, 1, 1\right) \\ \text{with: } e_{y1} &= \hat{e}_{1y} & e_{y2} &= \frac{\hat{e}_{1y} + \hat{e}_{1y}}{2 \cdot m} & e_{y3} &= \bar{e}_{1y} - \hat{e}_{1y} \end{aligned}$$

The saturation function (f) is defined below. It allows to take into account the UAV limitations (10). \bar{V}_X, \bar{V}_Y are the maximum allowed velocities according to x and y axes. The maximum allowed velocity of the follower is reached as soon as the components of its distance, to the leader, are greater or equal to \bar{D}_X, \bar{D}_Y . The saturation function (f) is also used to normalize the values of β_{xi} and β_{yj} between -1 and +1.

$$f(\text{eps}, Vm, Dm) = \begin{cases} -Vm & \text{if } \text{eps} < -Dm \\ \frac{Vm}{Dm} \cdot \text{eps} & \text{if } -Dm \leq \text{eps} \leq +Dm \\ +Vm & \text{if } \text{eps} > +Dm \end{cases}$$

From (10), we get the following control law:

$$\overrightarrow{U_1}^T = R \cdot \overrightarrow{V_L}^T + \begin{bmatrix} A(\hat{e}_{1y}) \cdot f(\bar{D}_X \cdot \sum_{i=1}^3 \beta_{xi}, \bar{V}_X - V_{Lx}, \bar{D}_X) \\ f(\bar{D}_Y \cdot \sum_{j=1}^3 \beta_{yj}, \bar{V}_Y - V_{Ly}, \bar{D}_Y) \end{bmatrix} \quad (11)$$

In steady state (10), the follower will fly at the speed of the leader without tracking errors ($\overrightarrow{V_1}^T = R \cdot \overrightarrow{V_L}^T$) only if it receives, through a communication network, the leader velocity ($\overrightarrow{V_L}$) without disturbances, latencies or losses.

C. Leader UAV velocities estimation

To compensate the tracking errors due to disturbances, latencies or loss of communication, the leader UAV velocity vector ($\overrightarrow{V_L}$) is estimated by using a first order dynamic model of the UAV (14):

$$\dot{\overrightarrow{V_{1m}}}^T = \begin{bmatrix} -\frac{1}{\tau_x} & 0 \\ 0 & -\frac{1}{\tau_y} \end{bmatrix} \cdot \overrightarrow{V_{1m}}^T + \begin{bmatrix} \frac{1}{\tau_x} & 0 \\ 0 & \frac{1}{\tau_y} \end{bmatrix} \cdot \overrightarrow{U_{1m}}^T \quad (14)$$

$\overrightarrow{V_{1m}} = [V_{1mx} \ V_{1my}]$ and $\overrightarrow{U_{1m}} = [U_{1mx} \ U_{1my}]$ are the velocity and control input vectors of the model respectively. In the earth frame (Fig.1), its position $\overrightarrow{P_{1m}} = [X_{1m} \ Y_{1m}]$ is:

$$\overrightarrow{P_{1m}}^T = R_o \cdot \overrightarrow{V_{1m}}^T \quad \text{with} \quad R_o = \begin{bmatrix} \cos(\delta_1) & -\sin(\delta_1) \\ \sin(\delta_1) & \cos(\delta_1) \end{bmatrix} \quad (15)$$

Let us consider the following filtered distance errors (Fig.1):

$$\hat{e}_{1m}^T = \begin{bmatrix} \hat{d}_1 * \cos(\hat{\theta}_1) \\ \hat{d}_1 * \sin(\hat{\theta}_1) \end{bmatrix} + R_o^{-1} \cdot (\overrightarrow{P_1}^T - \overrightarrow{P_{1m}}^T) - \begin{bmatrix} X_{SP1} \\ Y_{SP1} \end{bmatrix} \quad (16)$$

with: $\overrightarrow{e}_{1m} = [\hat{e}_{1mx} \ \hat{e}_{1my}]$

The control objective is to regulate the distance errors as follows:

$$\lim_{t \rightarrow \infty} \hat{e}_{1mx}(t) = 0 \quad \lim_{t \rightarrow \infty} \hat{e}_{1my}(t) = 0 \quad (17)$$

To fulfill this control objective, we propose the following error dynamics equation:

$$\dot{\hat{e}_{1m}}^T = - \begin{bmatrix} K_x & 0 \\ 0 & K_y \end{bmatrix} \cdot \hat{e}_{1m}^T = R \cdot \overrightarrow{V_L}^T - \overrightarrow{V_{1m}}^T \quad (18)$$

with: $K_x = \frac{\bar{V}_X}{\bar{D}_X}$ $K_y = \frac{\bar{V}_Y}{\bar{D}_Y}$

In steady state (14, 18), $\overrightarrow{U_{1m}} = \overrightarrow{V_{1m}}$ and $\overrightarrow{V_{1m}} = R \cdot \overrightarrow{V_L}$. The tracking errors (18) will be null only if the dynamic model get the leader UAV velocities without disturbances, latencies or losses. Otherwise, the tracking errors will depend on the values of the proportional gains K_x, K_y . Indeed, let us consider the following candidate Lyapunov function:

$$V = \frac{1}{2} \overrightarrow{\hat{e}_{1m}} \cdot \overrightarrow{\hat{e}_{1m}} = \frac{1}{2} \cdot (\hat{e}_{1mx}^2 + \hat{e}_{1my}^2) \quad (19)$$

Its time derivative can be written as:

$$\dot{V} = e_{1mx} \cdot \dot{e}_{1mx} + e_{1my} \cdot \dot{e}_{1my}$$

Thus, $\dot{V} < 0$ if $|\hat{e}_{1mx}| \geq \left| \frac{V_{Lx}}{K_x} \right|$ and $|\hat{e}_{1my}| \geq \left| \frac{V_{Ly}}{K_y} \right|$ so that the size of the tracking errors $|\hat{e}_{1mx}|$ and $|\hat{e}_{1my}|$ are uniformly and ultimately bounded by $\frac{\bar{V}_X}{K_x}$ and $\frac{\bar{V}_Y}{K_y}$ respectively, where \bar{V}_X and \bar{V}_Y represent bounds on the leader UAV velocity components satisfying $|V_{Lx}| < \bar{V}_X$ and $|V_{Ly}| < \bar{V}_Y$. These tracking errors don't have any effects since they concern the dynamic model. This interesting result allows to get the leader UAV velocities even in case of degraded (latencies, loss,...) communications. From (18), we get for the dynamic model the following control law:

$$\overrightarrow{U_{1m}}^T = R \cdot \overrightarrow{V_L}^T + \begin{bmatrix} K_x & 0 \\ 0 & K_y \end{bmatrix} \cdot \hat{e}_{1m}^T \quad (20)$$

Finally, by combining (11) and (20), we get:

$$\overrightarrow{U_1}^T = \overrightarrow{U_{1m}}^T + \begin{bmatrix} A(\hat{e}_{1y}) \cdot f(\bar{D}_X \cdot \sum_{i=1}^3 \beta_{xi}, \bar{V}_X - V_{Lx}, \bar{D}_X) \\ f(\bar{D}_Y \cdot \sum_{j=1}^3 \beta_{yj}, \bar{V}_Y - V_{Ly}, \bar{D}_Y) \end{bmatrix} \quad (21)$$

IV. SIMULATION RESULTS

The simulations have been carried out on Matlab-Simulink software. They are based upon a dynamic model of the UAV [14] combined with usual low level controllers (PID control for horizontal velocities and Phase Lead control for altitude and angular (yaw) positions). The simulation parameters are the following ones:

- Simulation time: $T=800s$ - Sampling period: $Te=0.01s$

- Altitude (after the take-off phase): $10m$
- Omnidirectional camera position: $Loc = 0.15m$
- Attenuation parameter : $Ay=0.3$
- Maximum velocities: $\vec{V}_X, \vec{V}_Y = 3 m/s$
- Maximum distances: $\vec{D}_X, \vec{D}_Y = 15 m$
- DES algorithm: $\alpha_{d,\theta} = 0.5, \lambda_{d,\theta} = 0.5, m = 10$
- Time constants (first order model): $\tau_x = 0.2s, \tau_y = 0.2s$
- Take-off conditions (m, deg) / linear-circle trajectories :
 - Leader $(X_L, Y_L, \delta_L) : [-82,-82,0^\circ] - [-95,0,90^\circ]$
 - Follower 1 $(X_1, Y_1, \delta_1) : [-90,-82,0^\circ] / [-95,-25,90^\circ]$
 - Follower 2 $(X_2, Y_2, \delta_2) : [-90,-74,0^\circ] / [-110,-15,90^\circ]$
 - Follower 3 $(X_3, Y_3, \delta_3) : [-90,-90,0^\circ] / [-80,-15,90^\circ]$

In order to study the LF formation and the global aspect of the VRB, linear and circle trajectories (radius of the circle: 90m - leader velocity according to x and y axes: 1m/s) have been chosen. The VRB requirements are the following ones:

- Circle trajectory (diamond):
 - $X_{SP1} = +20m - Y_{SP1} = 0m - Z_{SP1} = 0m$
 - $X_{SP2} = +10m - Y_{SP2} = -10m - Z_{SP2} = 0m$
 - $X_{SP3} = +10m - Y_{SP3} = +10m - Z_{SP3} = 0m$
 - UAVs yaw angle: $\delta_{1,2,3} = 90^\circ$
- Linear trajectory (line):
 - $X_{SP1} = +10m - Y_{SP1} = 0m - Z_{SP1} = 0m$
 - $X_{SP2} = +10m - Y_{SP2} = -10m - Z_{SP2} = 0m$
 - $X_{SP3} = +10m - Y_{SP3} = +10m - Z_{SP3} = 0m$
 - UAVs yaw angle: $\delta_{1,2,3} = 0^\circ$

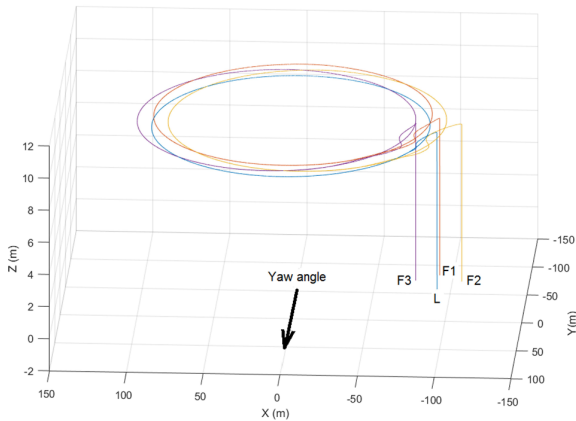


Figure 2. Circle trajectory

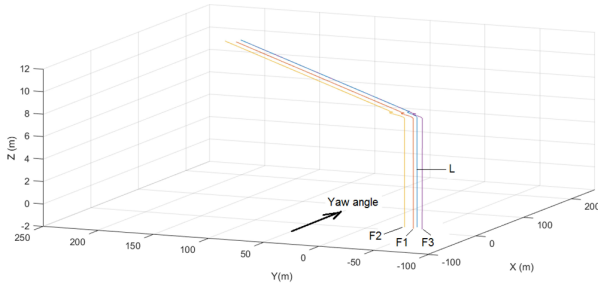


Figure 3. Linear trajectory

The simulation results in Fig.2,3 show that the UAVs follow properly (for different velocities and orientations) the circle and linear trajectories after the take-off phase and fly according to the desired VRB requirements (a triangle for the linear trajectory and a diamond for the circle trajectory). Then, to study separately and in details the effects of the leader velocity estimation with a first order dynamic model, the following scenario has been implemented for both trajectories:

Before $t=200s$, the leader UAV velocity is $\vec{V}_L = [0.3 \ 0.3]$ (in m/s). Between $t=200s$ and $t=600s$, the leader UAV accelerates $\vec{V}_L = [0.4 \ 0.4]$ (in m/s). After $t=600s$, the leader UAV recovers the previous velocity $\vec{V}_L = [0.3 \ 0.3]$ (in m/s). From $t=400s$ to $t=800s$, the communication between the UAVs isn't available. The simulation results correspond to the follower UAV 1. They are similar for the others.

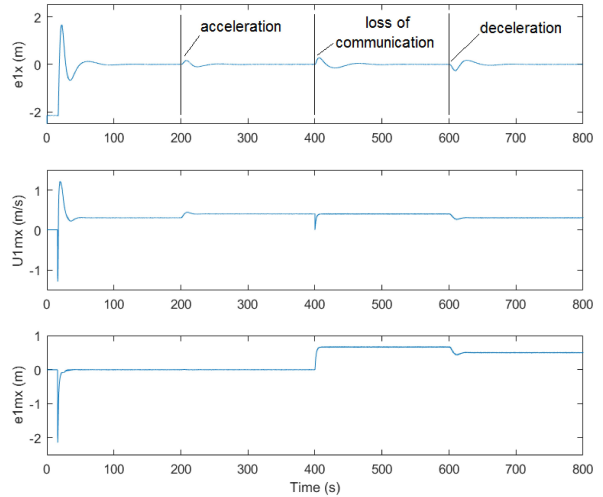


Figure 4. Follower 1 (linear trajectory)

x uav tracking error (e_{1x}) - model control input (U_{1mx}) - x model error (e_{1mx})

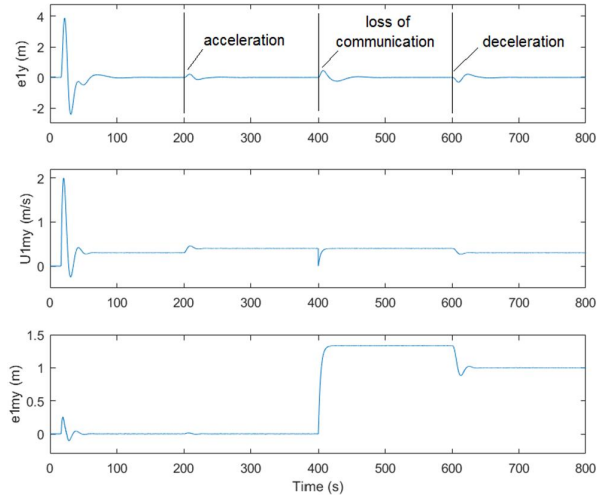


Figure 5. Follower 1 (linear trajectory)

y uav tracking error (e_{1y}) - model control input (U_{1my}) - y model error (e_{1my})

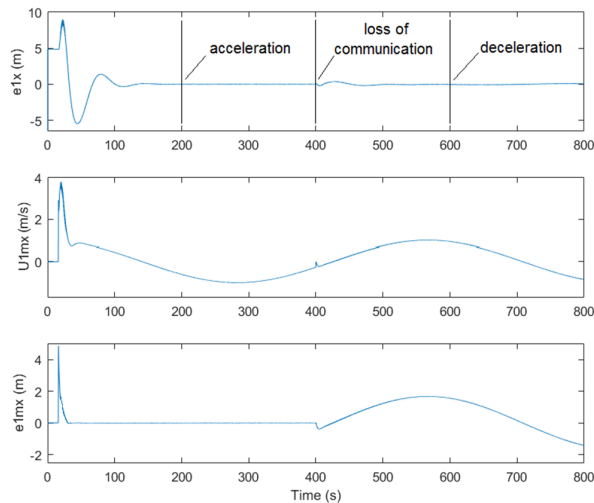


Figure 6. Follower 1 (circle trajectory)
x uav error (e_{1x}) – model control input (U_{1mx}) – y model error (e_{1mx})

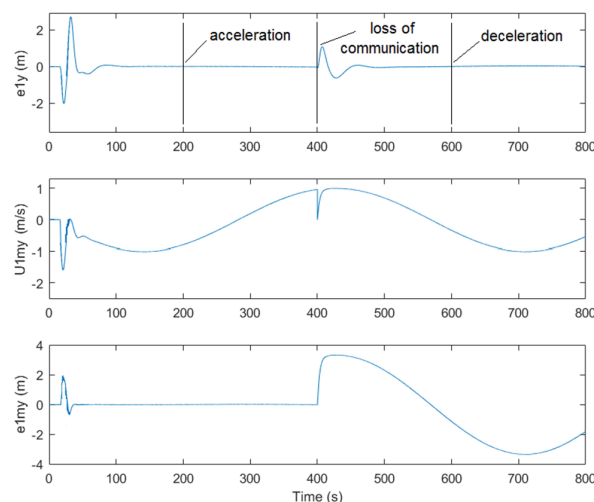


Figure 7. Follower 1 (circle trajectory)
y uav error (e_{1y}) – model control input (U_{1my}) – x model error (e_{1my})

On Fig.4,5,6,7 we can see that the tracking errors according to x and y axes converge toward zero when the leader UAV flies at different speeds and even when the communication between the leader and the followers is not available. As long as the communication is available, the model errors are null and the follower UAV 1 uses the speed information sent by the UAV leader. When the communication is lost, the model errors are not null and allows to recover the speed information.

V. CONCLUSION AND PERSPECTIVES

In conclusion, the formation control can work in a fully decentralized way. Indeed, unlike other VRB methods, which rely on the group barycenter to steer the formation [15], our approach requires neither positioning nor tracking from a centralized point. The high level controller will be

implemented in C++ language in a companion computer (Odroid XU4) embedded in DJI F450 quadrotors. The static and dynamic performances of the formation control will be evaluated for different shapes (lines, triangle, diamond,...) and trajectories (straight lines, circles,...), and the impact of communication losses will be analyzed.

ACKNOWLEDGMENT

The authors would like to thank Le Havre Town Council (LHSM) for their support under research grants.

REFERENCES

- [1] Arai.T, Pagello.E, Parker.L.E. – “Editorial: Advances in multi-robot systems” - IEEE Transactions on Robotics and Automation - 18(5), 655/661 – 2002.
- [2] Y. U. Cao, A. S. Fukunaga, A. B. Kahng – “Cooperative mobile robotics: Antecedents and directions” - Autonomous Robots, vol. 4, pp. 7–27 - 1997.
- [3] Mellinger.D, Shomin.M, Michael.N, Kumar.V – “Cooperative Grasping and Transport Using Multiple Quadrotors” - Springer Berlin Heidelberg 545/558 - 2013.
- [4] Khamis.A, Hussein.A, Elmogy.A – “Multi-robot Task Allocation: A Review of the State-of-the-Art” – Springer International Publishing 31/51 – 2015.
- [5] Zeng.Y, Zhang.R, Lim.T.J – “Wireless communications with unmanned aerial vehicles: opportunities and challenges” - IEEE Communications Magazine - 54(5), 36/42 - 2016.
- [6] Tuna.G, Nefzi.B, Conte.G – “Unmanned aerial vehicle-aided communications system for disaster recovery” - Journal of Network and Computer Applications - 41, 27/36 - 2014.
- [7] Tosun.T, Daudelin.J, Jing.G, Kress-Gazit.H, Campbell.M, Yim.M – “Perception-informed autonomous environment augmentation with modular robots” - IEEE International Conference on Robotics and Automation (ICRA) - doi:10.1109/icra.2018.8463155 - 2018.
- [8] Chen.X, Jia.Y – “Adaptive leader-follower formation control of non-holonomic mobile robots using active vision” - IET Control Theory & Applications - 9(8), 1302/1311 - doi:10.1049/iet-cta.2014.0019 - 2015.
- [9] Morbidi.F, Mariottini.G.L, Praticchizzo.D – “Observer design via immersion and invariance for vision based leader follower formation control” - Automatica,46(1),148/154 - doi:10.1016/j.automatica.2009.10.016 - 2010.
- [10] Guo.D, Wang.H, Chen.W, Liu.M, Xia.Z, Leang.K.K – “A unified leader-follower scheme for mobile robots with uncalibrated on-board camera” - IEEE International Conference on Robotics and Automation (ICRA) - doi:10.1109/icra.2017.7989438. – 2017.
- [11] Liu.X, Ge.S.S, Goh.C.H - “Vision-based Leader/Follower formation control of multiagents with visibility constraints” - IEEE Transactions on Control Systems Technology - 27(3), 1326/1333 - doi:10.1109/tcst.2018.2790966 – 2019.
- [12] Joseph J.LaViola Jr, “Double Exponential Smoothing: An Alternative to Kalman Filter-Based Predictive Tracking”, Proceedings of EGVE '03 Workshop on Virtual Environments, Pages 199 –206, ISBN:1-58113-686-2 - New York (USA), 2003.
- [13] Joseph J.LaViola Jr, “An Experiment Comparing Double Exponential Smoothing and Kalman Filter - Based Predictive Tracking Algorithms”, IEEE Virtual Reality Conference, ISBN: 0-7695-1882-6, Los Angeles (USA) – 2003.
- [14] David Escobar Sanabria – “ARDrone Simulink Development Kit Version 1.0” - University of Minnesota, Minneapolis, MN - Pieter J. Mosterman, MathWorks, Natick, MA - September 2, 2013
- [15] Brandao.A.S, Sarcinelli-Filho.M – “On the guidance of multiple uav using a centralized formation control scheme and delaunay triangulation” - Journal of Intelligent & Robotic Systems - 84(1-4), 397/413 - 2016.