Distributed circular formation flight of fixed-wing aircraft with Paparazzi autopilot

Hector Garcia de Marina, Gautier Hattenberger

Distributed circular formation flight of fixed-wing aircraft with Paparazzi autopilot

Hector Garcia de Marina[*] Gautier Hattenberger[†]
ENAC, University of Toulouse, F-31055, France

ABSTRACT

In this paper we introduce the usage of guidance vector fields for the coordination and formation flight of fixed-wing aircraft. In particular, we describe in detail the technological implementation of the formation flight control for a fully distributed execution of the algorithm by employing the open-source project Paparazzi. In this context, distributed means that each aircraft executes the algorithm on board, each aircraft only needs information about its neighbors, and the implementation is straightforwardly scalable to an arbitrary number of vehicles, i.e., the needed resources such as memory or computational power not necessarily scale with the number of total aircraft. The coordination is based on commanding the aircraft to track circumferences with different radii but sharing the same center. Consequently, the vehicles will travel different distances but with the same speeds in order to control their relative angles in the circumference, i.e., their orbital velocities. We show the effectiveness of the proposed design with actual formation flights during the drone parade in IMAV2017.

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have emerged as powerful platforms for, among others, atmospheric research, intensive agriculture and surveillance. A key aspect of the usage of these vehicles is to make affordable and accessible certain tasks within these fields. In particular, the technologies around these aerial vehicles, such as batteries, propulsion systems, construction materials and their associated maintenance, have developed an interesting performance for the actual cost. This fact makes interesting the usage of several of these vehicles in a cooperative way in order to enhance tasks that were previously performed by a single aircraft. For example, we can perform missions more efficiently by splitting the payload of one aircraft, such as antennas [1] or different kind of surveillance cameras [2] to two or more aircraft. Furthermore, we can also employ the coordination of several aircraft for the estimation of the wind without employing expensive sensors [3]. Another interesting and different usage of fleets of UAVs are related with performance shows such as the suggested outdoor parade contest of IMAV2017[1].

There exist a large number of algorithms in literature related with the formation control of vehicles [4]. However, most of them consider the dynamics of the vehicle as kinematic points, which is a very restrictive requirement for fixed-wing aircraft that can be seen as an unicycle. Formation control algorithms dealing with this kind of dynamics can still be found in the literature [5, 6, 7]. However, most of them do not consider desirable requirements such as vehicles with constant speed, e.g., with motors operating constantly at their nominal conditions, or it is not clear how to combine such algorithms with trajectory tracking, which is quite important in order to guarantee the confinement of the UAVs in their allowed airspace.

This paper aims to show how the algorithm proposed in [8] has been implemented in a fully distributed way by employing the open-source project Paparazzi. The algorithm proposes a technique based on guidance vector fields for trajectory tracking [9] in order to achieve the coordination of multiple fixed-wing aircraft flying at constant (and equal) speeds. In particular, we focus on circular trajectories but it is also applicable to any other closed trajectories. The main idea relies on controlling the traveled distance by the aircraft by commanding them to track a smaller or bigger radius with respect to the desired circle. Consequently, the vehicles will travel different distances but with the same speeds in order to control their relative angles in the circumference, i.e., the algorithm controls their orbital velocities. The aircraft are not always placed on the commanded circles but far away from them, specially in the beginning of the algorithm where the initial positions of the vehicles could be arbitrary. Nevertheless, the proposed strategy is feasible because of the exponential convergence property to the desired trajectory of the guidance algorithm [9]. Therefore if the convergence of the formation flight algorithm is slow enough, then the whole cascaded system (trajectory tracking and formation control) will converge to the desired coordinated formation flight.

We organize this paper as follows. The Section 2 reviews the concept of vector field for tracking smooth trajectories. Then, we explain in Section 3 how to manipulate such vector fields in order to achieve the desired circular formation flight. We continue in Section 4 showing how the algorithm is fully distributed and executed in the open-source Paparazzi

[*]hector.garcia-de-marina@enac.fr
[†]gautier.hattenberger@enac.fr

[1]http://www.imavs.org/2017

autopilot [10]. In particular, in Section 5 we show as an example a simulation of the formation flight parade that will be performed during IMAV 2017. We end the paper with some conclusions in Section 6.

2 Guidance vector field for tracking smooth trajectories

2.1 Fixed-wing aircraft's model

Consider for the *unit speed* fixed-wing aircraft the following nonholonomic model in 2D

$$\begin{cases} \dot{p} &= m(\psi) \\ \dot{\psi} &= u_\psi, \end{cases} \tag{1}$$

where $p = \begin{bmatrix} p_x & p_y \end{bmatrix}^T$ is the Cartesian position of the vehicle, $m = \begin{bmatrix} \cos(\psi) & \sin(\psi) \end{bmatrix}^T$ with $\psi \in (-\pi, \pi]$ being the attitude *yaw* angle[2] and u_ψ is the control action that will make the aircraft to turn. If we consider that the altitude of the vehicle is kept constant and its pitch angle is close to zero, then the control action u_ψ corresponds to the following bank angle ϕ in order to have a coordinated turn

$$\phi = \arctan \frac{u_\psi}{g}, \tag{2}$$

where g is the gravity acceleration.

2.2 Trajectory tracking

We have chosen the algorithm proposed in [9] for the task of tracking circular trajectories since it has been successfully validated in real flights [11]. One interesting property of the chosen algorithm is that the local exponential converge to the desired path is guaranteed. This property help us to support the convergence of the *high level* formation control algorithm under the argument of *slow-fast* dynamical systems [8].

Consider the circular path $\mathcal{P}$ described by

$$\mathcal{P} := \{p \,:\, \varphi(p) := p_x^2 + p_y^2 - r^2 = 0\}. \tag{3}$$

Clearly the function $\varphi : \mathbb{R}^2 \to \mathbb{R}$ belongs to the C^2 space and it is *regular* everywhere excepting at the origin, i.e.,

$$\nabla\varphi(p) = \begin{bmatrix} 2p_x \\ 2p_y \end{bmatrix} \neq 0 \iff p \in \mathbb{R}^2 \setminus 0. \tag{4}$$

The trajectory tracking algorithm employs the level sets $e(p) \triangleq \varphi(p)$ for the notion of *error distance* between the aircraft and $\mathcal{P}$. In particular, for circular trajectories a positive level set corresponds to an *expanded version* of the desired circle $\mathcal{P}$, while a negative level set corresponds to a *contracted version*. Note that the domain of the error distance is $e \in [-r^2, \infty)$. We define by $n(p) := \nabla\varphi(p)$ the normal vector to the curve corresponding to the level set $\varphi(p)$ and the tangent vector τ at the same point p is given by the rotation

$$\tau(p) = En(p) = \begin{bmatrix} 2p_y \\ -2p_x \end{bmatrix}, \quad E = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}.$$

Note that E will determine in which direction $\mathcal{P}$ will be tracked, which is done by following the direction at each point p given by of the vector field

$$\dot{p}_d(p) \triangleq \tau(p) - k_e e(p) n(p) = 2 \begin{bmatrix} p_y - k_e e p_x \\ p_x - k_e e p_y \end{bmatrix}, \tag{5}$$

where $k_e \in \mathbb{R}^+$ is a gain that defines how *aggressive* is the vector field. An example of the construction of the guidance vector field (5) is shown in Figure 1, and its visualization in Paparazzi for tracking an ellipsoidal trajectory is shown in Figure 2.

Let us define $\hat{x}$ as the unit vector constructed from the nonzero vector x. The vector field (5) is successfully tracked if we apply the following control action to (2) [9, 11]

$$u_\psi = -\left(E\hat{\dot{p}}_d\hat{\dot{p}}_d^T E\left((E - k_e e)H(\varphi)\dot{p} - k_e n^T \dot{p} n\right)\right)^T E\frac{\dot{p}_d}{||\dot{p}_d||^2} + k_d \hat{\dot{p}}^T E \hat{\dot{p}}_d, \tag{6}$$

where $H(\cdot)$ is the Hessian operator, i.e., from (4) we have that $H = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$ and $k_d \in \mathbb{R}^+$ is a positive gain that determines how fast the vehicle converges to the guidance vector field. We can clearly identify two terms in the addition in (6). The first term is a feedforward component and makes the aircraft to stay on the guidance vector field (5) while the second term makes the vehicle to converge to the guidance vector field in case that the vehicle is not aligned with it.

3 Guidance vector field as a coordinating tool

Consider a team of n aircraft traveling all of them at the same speed. The main objective of this paper is to show how to make them rendezvous without actuating on their traveling velocities. The rendezvous will happen at the same time that the team is traveling over a desired circular trajectory $\mathcal{P}$. We will see that this is possible to achieve by controlling the traveled distance of the aircraft around $\mathcal{P}$.

3.1 Circular trajectory

We summarize in this subsection the formation control algorithm presented in [8]. Consider that the center of $\mathcal{P}$ is at the origin as in (3). Let us define the phase of the aircraft as $\theta = \operatorname{atan2}(p_y, p_x)$ where

$$\operatorname{atan2}(y, x) = \begin{cases} \arctan(\frac{y}{x}) & \text{if } x > 0, \\ \arctan(\frac{y}{x}) + \pi & \text{if } x < 0 \text{ and } y \geq 0, \\ \arctan(\frac{y}{x}) - \pi & \text{if } x < 0 \text{ and } y < 0, \\ +\frac{\pi}{2} & \text{if } x = 0 \text{ and } y > 0, \\ -\frac{\pi}{2} & \text{if } x = 0 \text{ and } y < 0, \\ \text{undefined} & \text{if } x = 0 \text{ and } y = 0. \end{cases} \tag{7}$$

[2] For our setup, the yaw angle and heading angle can be considered equal due to the absence of wind.

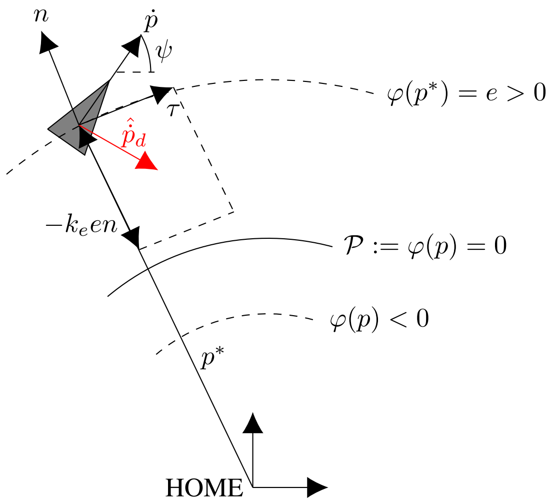


Figure 1: The direction to be followed by the UAV at the point p^* with respect to *HOME* for converging to $\mathcal{P}$ is given by $\hat{\dot{p}}_d$. The tangent and normal vectors τ and n are calculated from $\nabla\varphi(p^*)$. The error *distance* e is calculated as $\varphi(p^*)$.

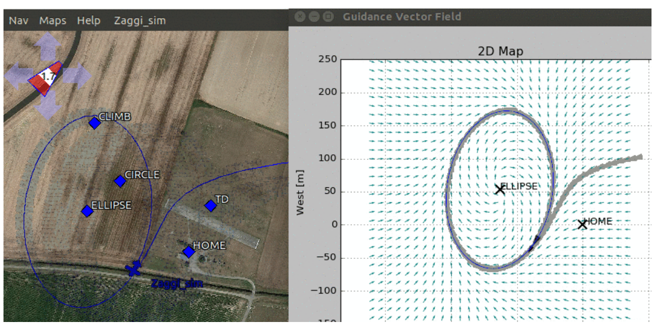


Figure 2: Example of the directions $\hat{\dot{p}}_d$ given by the vector field (5) for tracking an ellipsoidal track. Screenshot taken from the real time *gvf app* available in Paparazzi.

In a circular trajectory we are interested in controlling the different inter-vehicle phases, e.g., between aircraft 1 and 2 we could define $z_1 := \theta_1 - \theta_2$. For a general case, the relationships between neighbors are described by an undirected graph $\mathbb{G} = (\mathcal{V}, \mathcal{E})$ with the vertex set $\mathcal{V} = \{1, \dots, n\}$ and the ordered edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. The set $\mathcal{N}_i$ of the neighbors of agent i is defined by $\mathcal{N}_i \stackrel{\Delta}{=} \{j \in \mathcal{V} : (i, j) \in \mathcal{E}\}$. We define the elements of the incidence matrix $B \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{E}|}$ for $\mathbb{G}$ by

$$b_{ik} \stackrel{\Delta}{=} \begin{cases} +1 & \text{if} \quad i = \mathcal{E}_k^{\text{tail}} \\ -1 & \text{if} \quad i = \mathcal{E}_k^{\text{head}} \\ 0 & \text{otherwise} \end{cases} . \tag{8}$$

Note that the i'th row of B denotes for the i'th vehicle and the k'th column denotes for the link $\mathcal{E}_k$. Define $\Theta \in \mathbb{R}^{|\mathcal{E}|}$ as the stacked vector of aircraft phases, then one can calculate

$$z = B^T \Theta, \tag{9}$$

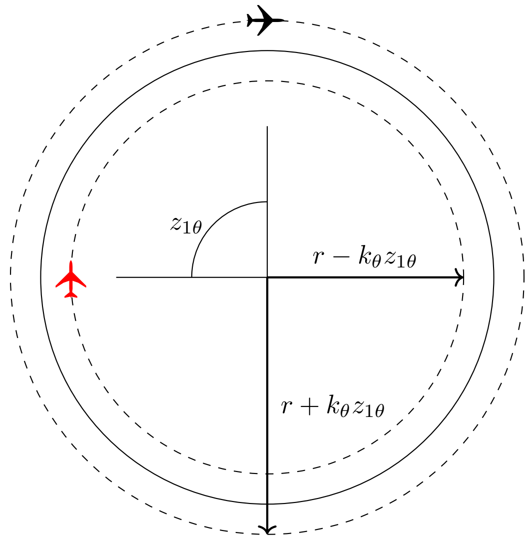


Figure 3: Red and black airplane want to fly over $\mathcal{P} := x^2 + y^2 - r^2 = 0$ (solid circle) with the same phase $\theta_1 = \theta_2$, i.e., $z_{1\theta} := \theta_1 - \theta_2 = 0$. Red airplane travels less distance if it tracks a smaller radius (negative level set with respect to $\mathcal{P}$), therefore its angular velocity with respect to the center of $\mathcal{P}$ is bigger than the angular velocity of black airplane. Once $z_{1\theta} = 0$, both aircraft are tracking $\mathcal{P}$. Setting the positive gain k_θ will tune the convergence time for the rendezvous.

as the stacked vector of all the available inter-vehicle phases $z_k(t)$ to be controlled. In particular, for rendezvous, we are interested in driving the signal $z(t)$ to zero. Depending on which level set of $\mathcal{P}$ an aircraft is following, it will travel with a different angular velocity $\dot{\theta}_i$ with respect to the center of $\mathcal{P}$, allowing us to control the evolution of the different z_k's. In order to track different level sets of $\mathcal{P}$ we introduce a control signal ${}^i u_r$ to (3) for the i'th aircraft as

$$\varphi(p) := p_x^2 + p_y^2 - (r + {}^i u_r)^2 = 0, \tag{10}$$

and we apply a consensus algorithm

$${}^i u_r = k_r B_i z, \tag{11}$$

where B_i stands for the i'th row of the incidence matrix B and $k_r \in \mathbb{R}^+$ is a positive gain. The results in [8] guarantees the exponential stability of the equilibrium at $\Theta = 0$ if $\mathcal{G}$ does not contain any cycles, e.g., a spanning tree topology. An example for two aircraft can be found in Figure 3.

4 Implementation in Paparazzi

We discuss in this section the implementation of the formation flight algorithm in the open-source project Paparazzi. We start by rewriting the control action (11) as follows

$${}^i u_r = k_r \sum_{j \in \mathcal{N}_i} (\theta_i - \theta_j), \tag{12}$$

$\mathcal{N}_i$	$\theta_{j\in\mathcal{N}_i}$	Timeout(ms)
2	π	1042
3	0.4π	426

Table 1: Example of table with neighbor's information.

therefore the i'th aircraft only needs to collect the phases θ_j from its neighbors. We do this process of collection by inter-vehicle communication, i.e., each aircraft calculates its phase θ_i and transmits it to its neighbors. Note that since we are controlling inter-vehicle phases, the aircraft can control their relative phases while following circumferences with different centers for $\mathcal{P}$, although obviously the rendezvous will not take place in such a situation.

We write the software responsible of running the algorithm in Paparazzi as a module[3]. A module allows one to add new code in a flexible way with initialization, periodic and event functions without modifying the main autopilot loop. Each aircraft will run exactly the some code without any particular modification. This will allow the scaling up of the number of aircraft without any associated penalty. In order to run the algorithm each aircraft is required in their flight plan to be tracking a circle with the guidance vector field algorithm[4].

The module employs the new functionalities of Paparazzi Link v2.0[5]. This communication layer allows the inter-vehicle communication, i.e., air to air, without intervention of the Ground Control Station. Each aircraft in Paparazzi has an unique identification ID (uint8) that will be employed for each aircraft i in (12). The formation flight module runs two independent processes. The first process keeps updated a table once a message from a neighbor is received. This table shown in Table 1 has information about the IDs of the neighbors, their latest received θ_j and the time since this value was updated. An aircraft can always ask to be registered in or deleted from another aircraft's table in order to become neighbors or break the relationship. The second process is executed periodically with a frequency of 2Hz. It calculates ${}^iu_r$ in (12) from all the updated data not older than 2 seconds, so we avoid situations like an aircraft that abandoned the formation but it continues registered in its neighbors table. Then the aircraft updates the radius to be tracked by the guidance vector field. Finally, the aircraft updates its own θ_i and transmit it to its neighbors if and only if the GPS is reliable, e.g., it has *3D Fix*. Consequently, if the aircraft does not update its neighbors' tables, then it will not be taken into account in the formation after the timeout. This process is summarized in Algorithm 1.

[3]https://wiki.paparazziuav.org/wiki/Modules

[4]https://wiki.paparazziuav.org/wiki/Module/guidance_vector_field

[5]Paparazzi Link is a messages toolkit (message definition, code generators, libraries) to be used with Paparazzi and compatible systems

```
Result: Rendezvous of aircraft i with its neighbors.
Data: θ_i and Table 1.
while Formation Control == True do
    ^i u_r = 0;
    for All the rows of the table do
        if Timeout is not reached then
            ^i u_r = ^i u_r + (θ_i − θ_j) ;
        end
    end
    Set radius r^2 + ^i u_r in the guidance vector field;
    if GPS is reliable then
        Transmit θ_i to N_i;
    end
end
```

Algorithm 1: Algorithm executed at aircraft i once the formation control is activated. Note that is not only distributed but it does not need all the information from all the neighbors to be executed.

5 IMAV FORMATION FLIGHT PARADE

The three employed aircraft are custom 1.5m wingspan flying-wings equipped with the following electronics:

- 2x servos SAVOX SH-0257MG
- ESC Flyduino KISS 18A v1.2
- Brushless motor T-Motor MT2208-18 1100KV
- Propeller 8x6
- 3S Battery pack built from Panasonic NCR18650B (up to 40min endurance, 2h+ with 2x3S)
- Apogee board autopilot[6]
- Futaba receiver R6303SB
- Futaba transmitter FAAST
- 2x XBEE Pro S1 (on board and on ground)
- GPS M8
- Ground Control Station: Laptop running Paparazzi on Ubuntu 17.04

We perform a formation flight simulation of three aircraft planned for the parade show in IMAV2017. For real experiment results like the one from the Figure 5 we refer to [8]. Although the available space for maneuvering is tight, the simulation shows that it is possible to synchronize the aircraft flying at 11m/s (ground speed) in circumferences of radius 30 meters as it is shown in Figure 6.

[6]https://wiki.paparazziuav.org/wiki/Apogee/v1.00

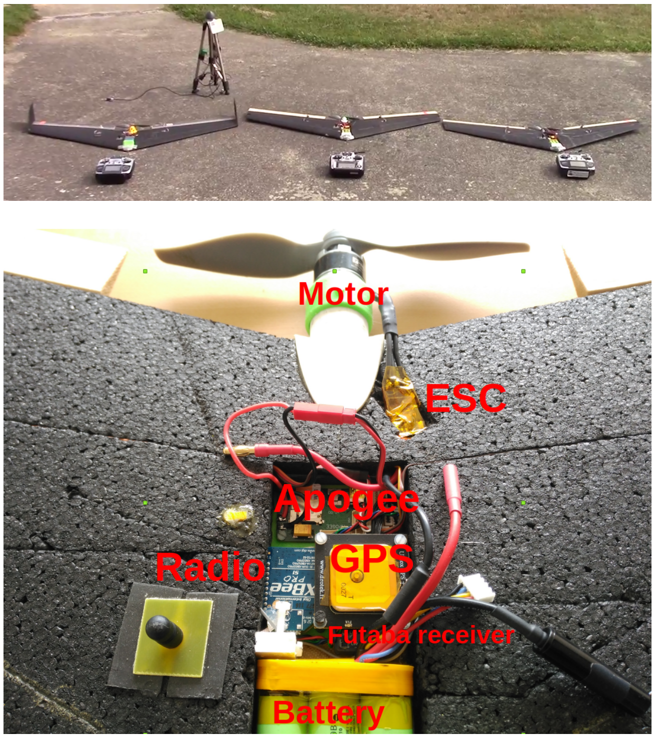


Figure 4: The three custom flying-wings employed for the formation flight parade equipped with Paparazzi autopilot.

Figure 5: Picture taken from an actual formation flight executing the algorithm described in this paper.

6 Conclusions

In this paper we have presented a fully distributed implementation of a flight formation controller for fixed-wing aircraft in the open-source autopilot Paparazzi. We manipulate guidance vector fields in order to control the inter-vehicle positions by changing the radius of the circumference to be eventually tracked. We finally show the effectiveness of the proposed strategies based on the rules for the IMAV2017 drone parade outdoor competition.

References

[1] Shuanggen Jin and Attila Komjathy. Gnss reflectometry and remote sensing: New objectives and results. *Advances in Space Research*, 46(2):111–117, 2010.

[2] Iván Maza, Fernando Caballero, Jesús Capitán, José Ramiro Martínez-de Dios, and Aníbal Ollero. Experimental results in multi-uav coordination for disaster management and civil security applications. *Journal of intelligent & robotic systems*, 61(1):563–585, 2011.

[3] Stephanie Mayer, Gautier Hattenberger, Pascal Brisset, Marius O Jonassen, and Joachim Reuder. A no-flow-sensorwind estimation algorithm for unmanned aerial systems. *International Journal of Micro Air Vehicles*, 4(1):15–29, 2012.

[4] Kwang-Kyo Oh, Myoung-Chul Park, and Hyo-Sung Ahn. A survey of multi-agent formation control. *Automatica*, 53:424–440, 2015.

[5] Joshua A Marshall, Mireille E Broucke, and Bruce A Francis. Pursuit formations of unicycles. *Automatica*, 42(1):3–12, 2006.

[6] Rodolphe Sepulchre, Derek A Paley, and Naomi Ehrich Leonard. Stabilization of planar collective motion with limited communication. 53(3):706–719, 2008.

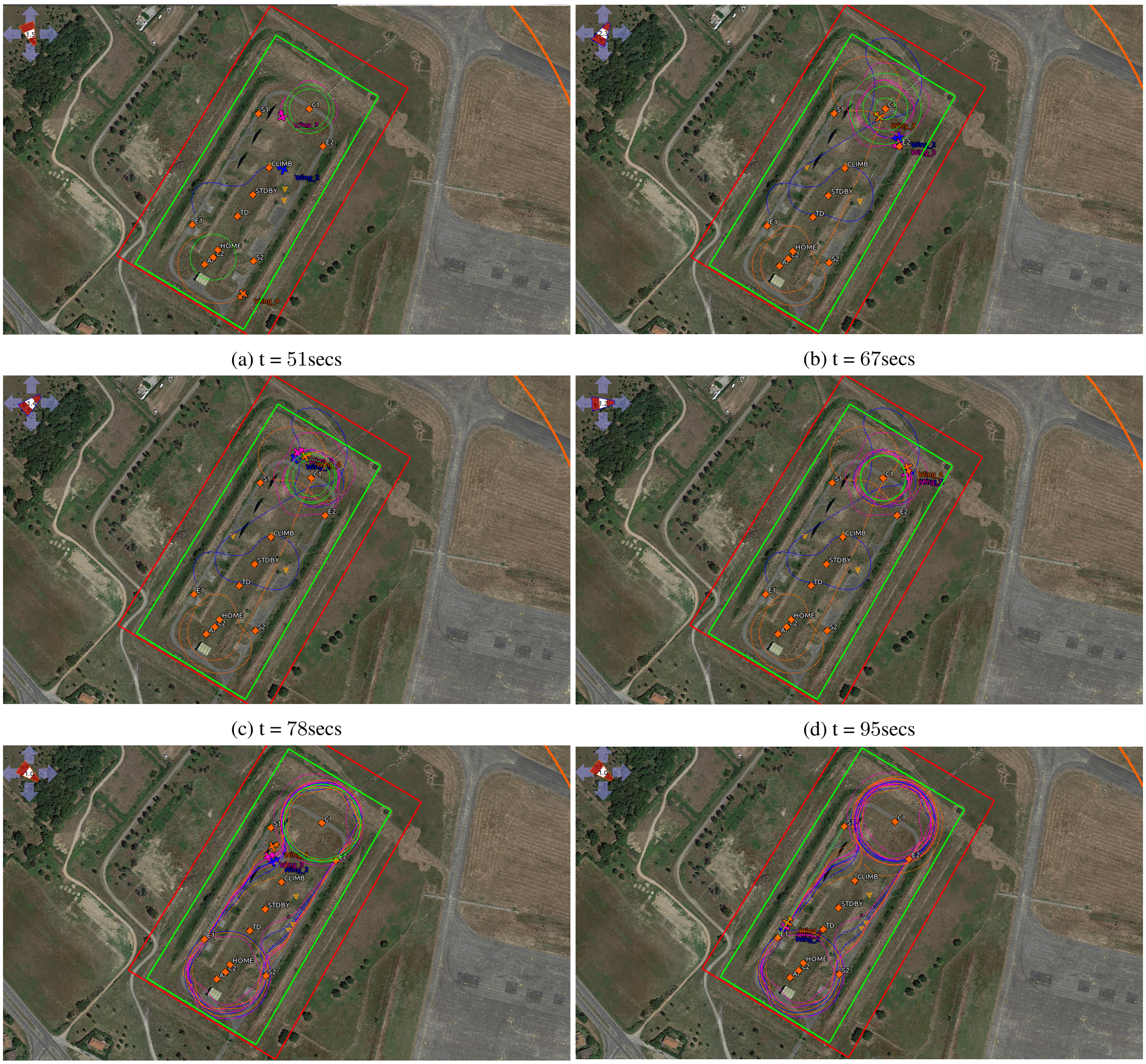
[7] Z. Sun, H. G. de Marina, G. S. Seyboth, B. D. O. Anderson, and C. Yu. Circular formation control of multiple unicycle-type agents with non-identical constant speeds. *IEEE Transactions on Control Systems Technology*, 2017.

[8] Hector G. de Marina, Zhiyong Sun, Murat Bronz, and Gautier Hattenberger. Circular formation control of fixed-wing uavs with constant speeds. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, 2017.

[9] Yuri A. Kapitanyuk, Anton V. Proskurnikov, and Ming Cao. A guiding vector field algorithm for path following control of nonholonomic mobile robots. *IEEE Transactions on Control System Technology*, 2016.

[10] Gautier Hattenberger, Murat Bronz, and Michel Gorraz. Using the paparazzi UAV system for scientific research. In *IMAV 2014, International Micro Air Vehicle Conference and Competition 2014*, pages pp–247, 2014.

[11] Hector Garcia de Marina, Yuri A Kapitanyuk, Murat Bronz, Gautier Hattenberger, and Ming Cao. Guidance algorithm for smooth trajectory tracking of a fixed wing uav flying in wind flows. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.



(a) t = 51secs

(b) t = 67secs

(c) t = 78secs

(d) t = 95secs

(e)

(f)

Figure 6: Screenshots from the Paparazzi ground control station showing the evolution of the circular formation. The first screenshot starts with the aircraft at arbitrary positions within the allowed flying area for the IMAV outdoor competition. The convergence to a synchronized formation flight takes place in around thirty seconds without leaving the allowed flying area. The circular synchronization can be used for *launching* the aircraft to travel together over the same segment.