On the Design of Outdoor Leader-Follower UAV-Formation Controllers From a Practical Point of View

LUCAS VAGO SANTANA\textsuperscript{1}, ALEXANDRE SANTOS BRANDÃO\textsuperscript{2}, AND MÁRIO SARCINELLI-FILHO\textsuperscript{3}

\textsuperscript{1}Department of Industrial Automation, Federal Institute of Espírito Santo, Linhares 29901-291, Brazil
\textsuperscript{2}Department of Electrical Engineering, Federal University of Viçosa, Viçosa 36570-900, Brazil
\textsuperscript{3}Department of Electrical Engineering, Federal University of Espírito Santo, Vitória 29075-910, Brazil

Corresponding author: Mário Sarcinelli-Filho (mario.sarcinelli@ufes.br)

This work was financed by CNPq–Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazilian Ministry of Science, Technology, Innovations and Communications, FAPES–Fundaçao de Amparo à Pesquisa e Inovação do Espírito Santo, and FAPEMIG–Fundaçao de Amparo à Pesquisa de Minas Gerais, Brazil.

ABSTRACT This work presents a control structure designed to autonomously guide two of-the-shelf quadrotors navigating in outdoor environments, as a leader-follower formation. The objective is to show how simple is to design controllers to guide the two vehicles when taking into account practical assumptions and simplifications in their model. This way, the main contribution of the paper is to show how practical knowledge obtained from real experiments using of-the-shelf quadrotors, translated to the assumptions and simplifications applied to its model, can help in the design of a suitable controller. The basic premise is that the quadrotor is driven by high-level commands that excite its degrees of freedom. The control system here designed following such reasoning is implemented in a ground station that allows manual or automatic control of the leader vehicle, with the follower one always automatically controlled. Experimental results in real outdoor scenarios, subject to unpredicted disturbances, validate the assumptions and simplifications applied to the model of the quadrotor, and the proposed control structure as well.

INDEX TERMS Multirobot systems, autonomous agents, motion control, unmanned aerial vehicle, nonlinear control systems.

I. INTRODUCTION The application of control techniques to unmanned aerial vehicles (UAVs) is an interesting research topic, and has been extensively explored. Thus, a lot of results regarding different control techniques adopted to guide the autonomous navigation of such vehicles have already been published. A few of them, illustrating different control approaches and results associated to quadrotors, are those reported in [1]–[6]. Nevertheless, such proposals are generally difficult to understand, and thus replicate, especially for novice readers, due to the complexity of their structure and mathematical description. As an extension, complex control techniques have also been proposed to guide groups of UAVs navigating in a coordinated way, thus generating what is nowadays called formation control or control of multi-robot systems. As one can see, for instance, in the works reported in [7]–[16], complexity is still a characteristic of the control techniques adopted.

This work also addresses the topic of guiding a multi-robot system, specifically a system composed by two quadrotors navigating as a leader-follower formation. The focus is on the design and implementation of a control structure for a quadrotor navigating as the follower UAV, in a leader-follower formation, in outdoor environments, under the incidence of wind disturbances. The main contribution of the paper is the proposal of a control architecture based on a simplified UAV model, to guide the vehicle when flying according to the leader-follower paradigm.

Hereinafter the paper discusses relevant aspects of the state of the art to give theoretical support to the methodology adopted to design the proposed controllers, as well
as presents and discusses results obtained from real flights using the proposed control structure, which validate the proposed control structure as well as the simplified model adopted to the quadrotor. Specifically, Section II presents some related works dealing with leader-follower formations of UAVs, whereas Section III briefly outlines the experimental setup used to run the validation experiments, and introduces a simplified model to represent the UAVs. In the sequel, Section IV presents the proposed automatic flight controllers, whereas Section V describes the software organization adopted to run the navigation experiments. Thereafter, Section VI shows some experimental results that validate the proposed control system, and, finally, Section VII highlights some remarks on the development of this work and stresses the performance of the control system designed with basis in the UAV simplified model in outdoor formation flights.

II. RELATED WORKS

Four classical paradigms are commonly adopted to address the problem of controlling a multi-robot system, also known as a multi-robot formation. They are the virtual structure, the behavior-based, the consensus-based and the leader-follower paradigms. This work focuses on the leader-follower control paradigm, proposing a control architecture to guide a leader-follower formation of two quadrotors.

The leader-follower (L-F) formation is a control strategy in which the leader is the agent responsible for conducting the follower. Therefore, the follower has the role of mimicking the movement of the leader. All information required for its navigation is referred to the leader. Thus, the leader is the most important agent in the formation, since its failure can compromise the mission accomplishment. There are several techniques to deal with leader malfunctioning and to assure that the formation remains running, but they are out of the scope of this work.

The concepts of the leader-follower approach are often used in other paradigms of formation control. The indication of a leader in a virtual structure or of an agent who dictates the meeting point in a consensus-based approach, for instance, are good examples [8], [17]–[19]. Besides, the L-F strategy allows that a follower vehicle be the leader of another agent, in a leader-follower formation involving three or more inline vehicles, thus configuring a cascade L-F approach. In this work, however, the L-F approach is used on its basic formulation, involving just two agents, a leader and one follower. This means that there is a relationship among the pose of the two agents and the formation itself, and the control strategy is designed according to this relationship.

Regarding control techniques adopted to control multirobot systems, in [20] a sliding mode controller guides a group of quadrotors, which are stabilized and controlled by a non-singular fast terminal sliding mode technique. The strategy is numerically validated for a set of five UAVs, navigating longitudinally in a predefined altitude. In [17] an interactive L-F configuration states that the leaders react with respect to the followers, applying consensus-based concepts.

One or several UAVs lead, while other UAVs are followers, and the proposed interaction matrix establishes how a UAV interacts with its neighbors. In turn, an adjacency matrix is used in [8] to represent how quadrotors arranged as a group share information. A consensus-based approach drives the L-F formation, while an almost linear control law (obtained after performing a static feedback linearization on the altitude and attitude) ensures the simultaneous convergence of the UAVs. Similarly, graph theory models the interaction of the agents in [21], and a backstepping controller guarantees their convergence in a finite-time. The closed-loop stability, in the sense of Lyapunov, is proven.

In [19] simulations illustrate a L-F formation following a given set of waypoints. In such a case, the formation-control problem is decoupled into horizontal and vertical motions, and a model predictive control generates the reference for the leader UAV, while the follower one uses its flight information to predict the path and achieve the formation pattern.

Considering scenarios with external disturbances, a flatness-based finite time controller guides, in simulation, a leader follower formation in [22]. Assuming the existence of obstacles, the authors of [23] propose a rapidly-exploring random tree algorithm that computes a free-of-collision path for a UAV, which is followed by another one. The proposal runs in ROS-Gazebo environment, with obstacles modeled as round pillars. In such a case, there is no explicit mathematical relationship between the UAVs. Moreover, the follower UAV sees the leader one as an obstacle that should also be avoided during navigation.

To the best of our knowledge, most articles dealing with leader-follower formations are just numerically validated, or show real experiments run in a controlled and well-structured scenario. Thus, we understand that as our experimental results were gathered in outdoor environments, under unpredicted disturbances (such as wind gusts), they are also a valuable contribution. However, the main contribution is the design of the UAV controllers with basis on a simplified model.

III. THE UAV FORMATION SETUP

The hardware used in this work consists of two parts. The first one consists of a portable computer, hereinafter named Ground Station (GS), in which the control algorithms are run. The other part consists of two commercial quadrotors AR.Drone 2.0 of Parrot Drones SAS (see Figure 1). These quadrotors were chosen mainly because of its moderate acquisition cost (about US$450) and of the worldwide availability for buying. In addition, the AR.Drone quadrotor has important features that make easier the development of computer codes for its guidance, as described in [24], [25]. Moreover, its on-board computer has native codes that guides the quadrotor in important maneuvers, such as takeoff and landing, besides taking care of controlling the attitude of the vehicle.
A. GS-UAV COMMUNICATION

The communication between the ground station and an AR.Drone quadrotor is performed according to a set of protocols detailed in [25]. After turning on the quadrotor one can create a wi-fi link and establish an information sharing among the GS and the quadrotor itself, in such a way that the GS can require sensory data from the UAV and send control signals to it. In this work this is done for two quadrotors, the leader one and the follower one, such that we have two wi-fi links for the communication between the GS and the two UAVs. For each one the available sensory data is packed in the vector 

\[ q = [z, v_x, v_y, v_z, \phi, \theta, \psi, \text{lat}, \text{lng}]^T, \]

where \( z \) is the altitude (in m) in the global frame, \( v_x, v_y \) and \( v_z \) are the linear velocities (in \( \text{m/s} \)) in the body frame, and \( \phi, \theta \) and \( \psi \) are the Euler angles (in rad) in the global frame. As for \( \text{lat} \) and \( \text{lng} \), they are the latitude and longitude (in \( ^\circ \)), in the geographical frame.

To move the AR.Drone, its on-board computer receives and interprets four normalized control commands, arranged as the vector \( u = [u_\phi, u_\theta, u_\psi] \), where \( u_\phi \) and \( u_\theta \) represent the rolling and pitch commands, while \( u_\psi \) are the commands for altitude and yaw rates, all of them normalized to the interval \([-1.0, +1.0]\) [25].

B. THE SIMPLIFIED DYNAMIC MODEL

The mathematical model of quadrotor UAVs is well known in the literature. Usually such models consider the vehicles as rigid bodies in the 3-D space, controlled by high-level signals, such as forces and torques. The way chosen here to make easier to design controllers is to propose a simplified and effective representation (from a practical point of view) of the vehicle. Taking inspiration in [1], the UAV is modeled as a function of high-level control signals, generating the differential equations

\[
\begin{align*}
\ddot{x} &= F(\sin \psi \sin \phi + \cos \psi \cos \phi \sin \theta) / m \quad (1) \\
\ddot{y} &= F(-\cos \psi \sin \phi + \sin \psi \cos \phi \sin \theta) / m \quad (2) \\
\ddot{z} &= F(-\cos \theta \cos \phi) / m + g \\
\ddot{\phi} &= \tilde{r}_\phi \quad (5) \\
\ddot{\theta} &= \tilde{r}_\theta \quad (6) \\
\ddot{\psi} &= \tilde{r}_\psi \quad (6)
\end{align*}
\]

for the translational and rotational dynamics, in the global frame. In such equations \( m \) is the UAV mass, \( g \) is the gravity acceleration and the force \( F \) represents the total thrust generated by the four rotors. As for the variables \( \tilde{r}_\phi \), \( \tilde{r}_\theta \) and \( \tilde{r}_\psi \), the virtual inputs relative to the roll, pitch and yaw torques [26], they are the high-level control signals.

**Remark 1:** Equations (3) and (6), related to the vertical displacement \( z \) and heading angle \( \psi \), respectively, are suppressed on the dynamic model, because a PD algorithm similar to the proposed in [3] regulates these variables to constant values.

For modern UAVs, such as the AR.Drone quadrotor, the set of equations (1)–(6) is enough to model and describe them, because they have an on-board computer responsible for performing low-level control of the engines, automatically adjusting its propulsion and orientations, according to the user commands. In summary, the low-level model can be suppressed in the control design. In such a context, it is possible to propose interesting simplifications to the model, considering that in most practical flights the maneuvers occur in near hovering situations (low velocities in the longitudinal and side directions). In other words, the roll and pitch angles are close to zero. Regarding the AR.Drone quadrotor, there indeed are configurable parameters on its on-board computer which allows limiting the maximum longitudinal and side inclinations.

**Remark 2:** During a hovering maneuver without disturbance or user commands, one can assume that the thrust generated by the propellers is equivalent to the vehicle weight, i.e.

\[ F \approx mg. \]

Assuming small angles \( \phi \) and \( \theta \) in (1) and (2), so that \( \cos \phi \approx 1, \cos \theta \approx 1, \sin \theta \approx \theta \) and \( \sin \phi \approx \phi \), and using (7) one gets

\[
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{z} \\
\dot{\phi} \\
\dot{\theta} \\
\dot{\psi}
\end{bmatrix} = g 
\begin{bmatrix}
\sin \psi & \cos \psi & 0 \\
-\cos \psi & \sin \psi & 0 \\
0 & 0 & 0 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix} 
\begin{bmatrix}
\phi \\
\theta \\
\psi
\end{bmatrix}. 
\]

**Remark 3:** Assuming a non-zero value for the yaw angle, notice that \( \phi \) and \( \theta \) modify the positions \( x \) and \( y \) of the UAV, showing that the translational and rotational dynamics are coupled.

Similar simplification can be done regarding the rotational dynamics, represented by (4), (5) and (6). It is important to stress the relationship among the virtual control signals \( \tilde{r}_\phi \) and \( \tilde{r}_\theta \), and the real orientation commands \( u_\phi \) and \( u_\theta \) is rather complex. From the literature perspective, it is clearly explained in [2], [26], [27] and many other works that establish formulas relating the electronic control of the motors and the forces and torques produced by them in the body of the vehicle. Oftentimes those equations involve physical parameters of the vehicle, such as moments of inertia, the axles of the engines, and so on.
However, it is difficult to use this method with the AR.Drone quadrotor, since it is not possible to individually control each engine. Actually, one sends a command \( u_\phi \), for the roll angle, or \( u_\theta \), for the pitch angle. Such commands are interpreted as a reference by the onboard computer, which is responsible for modifying the speeds of the engines, moving the UAV to a commanded orientation. Such observation indicates that there is a low level controller regulating the UAV attitude (i.e. the orientation angles), what is true when regarding the AR.Drone.

Therefore, this work proposes to consider that the dynamic response of the UAV to the orientation commands \( u_\phi \) and \( u_\theta \) is already controlled, such that its mathematical representation can be approximated by the linear second-order differential equations

\[
\begin{bmatrix}
\ddot{\phi} \\
\ddot{\theta}
\end{bmatrix} = \begin{bmatrix}
\omega_\phi^2 u_\phi - 2\zeta_\phi \omega_\phi \dot{\phi} - \omega_\phi^2 \phi \\
\omega_\theta^2 u_\theta - 2\zeta_\theta \omega_\theta \dot{\theta} - \omega_\theta^2 \theta
\end{bmatrix},
\]

where \( \omega_\phi \), \( \zeta_\phi \), \( \omega_\theta \) and \( \zeta_\theta \) are performance parameters of the attitude controller implemented onboard the UAV (the autopilot embedded in the firmware of the AR.Drone).

Some tests to identify such parameters were performed, but are not discussed here, since such topic is out of the scope of this work. Nonetheless, the experimental results presented in Section VI help to conclude the validity of this idea (at least for orientations with magnitude not greater than 0.175rad or 10°, maximum values adopted for \( \theta \) and \( \phi \)).

In summary, this discussion allows claiming that it is possible to control the translational dynamic response of the UAV according to (8) and the rotational one according to (9). Under the reported conditions, \( u_\phi \) and \( u_\theta \) become angle references to the controller represented in (9), reasoning that is adopted in the design of the controllers discussed hereinafter.

**IV. THE FLIGHT CONTROLLERS**

In this section flight controllers are proposed for the leader and the follower quadrotors, adopting the UAV model inspired in [1] and previously described. Figure 2 illustrates the formation setup composed by the leader and the follower UAVs, whose variables are related to the world frame. As for the variables involved, it is considered a subscript 1 for the leader and 2 for the follower, to refer to the vehicles.

To control the translational dynamics it is adopted a hierarchical control structure, similar to those in [26] and [28]. In such strategy, an external control loop is used to control the positions \( x \) and \( y \), generating desired orientation signals, which in turn represent angular references for an internal control loop for \( \phi \) and \( \theta \) (not transparent to the user).

**Remark 4 (Independent Controllers):** In closed-loop if the convergence of the orientation is faster than that of position, then the control system make the errors null in steady-state, as stated by the principle denominated *Time-Scale Separation* [28]. Therefore, it is feasible to design independent controllers for the translational and rotational dynamics of UAVs.

**A. THE CONTROLLER OF THE LEADER UAV**

Applying the approach of independent controllers, the leader UAV is first considered. Thus, the external control loop is established, which defines \( \phi_{d1} \) and \( \theta_{d1} \) as auxiliary control signals, representing the desired orientations for its internal loop. Then, (8) is rewritten as

\[
\begin{bmatrix}
\dot{x}_1 \\
\dot{y}_1
\end{bmatrix} = g \begin{bmatrix}
\sin \psi_1 & \cos \psi_1 \\
-\cos \psi_1 & \sin \psi_1
\end{bmatrix} \begin{bmatrix}
\phi_{d1} \\
\theta_{d1}
\end{bmatrix},
\]

where the subscript 1 identifies refers to the leader vehicle. Adopting matrix notation (10) becomes

\[
\dot{X}_1 = gR_{\psi_1} U_{d1},
\]

from where one can design the auxiliary control signals

\[
U_{d1} = \begin{bmatrix}
\phi_{d1} \\
\theta_{d1}
\end{bmatrix} = \frac{1}{g} R_{\psi_1}^{-1} (X_{1d} + \dot{\tilde{X}}_1 + \kappa_{D1} \tilde{X}_1 + \kappa_{P1} \tilde{X}_1),
\]

using the technique of inverse dynamics. In such control law \( X_{1d} = \begin{bmatrix} x_{1d} \\ y_{1d} \end{bmatrix} \) is the desired position for the leader UAV, \( X_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \) is its current position (estimated by the sensors onboard the UAV), and \( \tilde{X}_1 = X_{1d} - X_1 \) is the position error. As for \( \kappa_{D1} = \text{diag}(2.0, 2.0) \) and \( \kappa_{P1} = \text{diag}(2.0, 2.0) \), they are diagonal positive definite matrices correspondent to the derivative and proportional gains of the controller, respectively.

Introducing (12) in (11), the closed-loop dynamics of the translational error becomes

\[
\ddot{X}_1 + \kappa_{D1} \dot{X}_1 + \kappa_{P1} X_1 = 0,
\]

where \( \tilde{X}_1 = \tilde{X}_{d1} - \tilde{X}_1 \). The result is that this linear error dynamics has \( \tilde{X}_1 = 0 \) as equilibrium, which is asymptotically stable, since \( \kappa_{P1} \) and \( \kappa_{D1} \) are positive definite matrices, meaning that \( \tilde{X}_1 \to 0 \) when \( t \to \infty \).

In practice the control signals calculated in (12) should be limited, guaranteeing a bound in the required inclinations, thus preventing big inclinations due to great control errors. Such a limit is defined as

\[
\begin{bmatrix}
\phi_{d1} \\
\theta_{d1}
\end{bmatrix} = \begin{bmatrix}
sat(\phi_{d1}) \\
sat(\theta_{d1})
\end{bmatrix},
\]

where

\[
sat(\alpha) = \begin{cases}
\alpha, & \text{if } |\alpha| \leq 0.175\text{rad} \\
0.175 \text{ sign}(\alpha), & \text{otherwise},
\end{cases}
\]
is a saturation function, for which the value of 0.175rad was selected after running some trial experiments. In such condition it is assumed that the internal loop controlling the orientation of the vehicle is able to track any orientation given by (14). The practical effect of the saturation function is the truncation of the desired inclination, which assumes momentarily a constant inclination, moving towards the desired position.

Finally, the orientation references are sent to the internal control loop as

$$
\begin{bmatrix}
\dot{\phi}_1 \\
\dot{\theta}_1
\end{bmatrix} = \begin{bmatrix}
\omega_\phi^2 \phi_{d1} - 2 \xi_\phi \omega_\phi \dot{\phi}_1 - \omega_\theta^2 \phi_1 \\
\omega_\theta^2 \theta_{d1} - 2 \xi_\theta \omega_\theta \dot{\theta}_1 - \omega_\phi^2 \dot{\theta}_1
\end{bmatrix}.
\tag{15}
$$

In such case, introducing these signals in (9), the rotational dynamics becomes

$$
\begin{bmatrix}
\dot{\phi}_1 \\
\dot{\theta}_1
\end{bmatrix} = \begin{bmatrix}
\omega_\phi^2 \phi_{d1} - 2 \xi_\phi \omega_\phi \dot{\phi}_1 - \omega_\theta^2 \phi_1 \\
\omega_\theta^2 \theta_{d1} - 2 \xi_\theta \omega_\theta \dot{\theta}_1 - \omega_\phi^2 \dot{\theta}_1
\end{bmatrix},
\tag{16}
$$

which suggests that the internal loop controller implemented in the computer onboard the AR.Drone will be able to track a desired reference of orientation, with the performance determined by the parameters of the model.

B. THE CONTROLLER OF THE FOLLOWER UAV

The controller adopted to guide the follower UAV comes from a formation controller, whose objective is to guide the follower vehicle so that the desired leader-follower formation is kept. First of all, the formation is defined as a Cartesian virtual structure. In this case it is characterized by $X_f = X_1 - X_2$, where $X_1 = [x_1 \ y_1]^T$ is the global position of the leader vehicle and $X_2 = [x_2 \ y_2]^T$ is the position of the follower one. An important detail is that the two UAVs are supposed to fly in the same altitude, so that just the coordinates $x$ and $y$ are considered.

Taking the second time derivative of such relationship it comes that $\ddot{X}_f = \ddot{X}_1 - \ddot{X}_2$, which can be written, using the same notation of (11), as

$$
\ddot{X}_f = g R_{\psi} U_{d1} - g R_{\psi} U_{d2},
\tag{17}
$$

from where one gets the formation controller as

$$
U_{d2} = \begin{bmatrix}
\phi_{d1} \\
\theta_{d1}
\end{bmatrix} = \begin{bmatrix}
\frac{1}{g} \omega_\phi^2 (-\mu + g R_{\psi} U_{d1})
\end{bmatrix},
\tag{18}
$$

with $\mu = \ddot{X}_{d2} + \kappa_{D\phi} \ddot{X}_f + \kappa_{P\phi} X_f$, being an auxiliary control signal, where $X_{d2}$ characterizes the desired formation, $X_f$ is the current formation (determined from the positions of the UAVs given by their sensors), $\ddot{X}_f = X_{d2} - X_f$ characterizes the formation error and $\kappa_{P\phi} = \text{diag}(2.0, 2.0)$ and $\kappa_{D\phi} = \text{diag}(2.0, 2.0)$ are positive definite diagonal matrices representing the derivative and proportional gains of the formation controller, respectively. Introducing the controller characterized in (18) in (17), one gets the dynamics of the formation error, which is

$$
\ddot{X}_f + \kappa_{D\phi} \dot{X}_f + \kappa_{P\phi} X_f = 0,
\tag{19}
$$

which allows characterizing the equilibrium $\ddot{X}_f = 0$ as asymptotically stable, since $\kappa_{D\phi}$ and $\kappa_{P\phi}$ are positive definite diagonal matrices. As a consequence, $\dot{X}_f \to 0$ when $t \to \infty$.

To conclude, the control signals obtained using (18) represents the desired orientations for the follower vehicle, what means to make

$$
\begin{bmatrix}
u_{d2} \\
u_{d2}
\end{bmatrix} = \begin{bmatrix}
\phi_{d1} \\
\theta_{d1}
\end{bmatrix} = \begin{bmatrix}
sat(\phi_{d1}) \\
sat(\theta_{d1})
\end{bmatrix}.
\tag{20}
$$

V. THE EXPERIMENTAL ARCHITECTURE

Figure 3 illustrates the flow-diagram of the architecture required to run the experiments. The algorithms correspondent to the blocks encapsulated in the dashed-line box were programmed in the ground station computer. The vehicles are represented as AR.Drone 1 (Leader) and AR.Drone 2 (Follower).

Regarding the leader vehicle one should inform the system whether it will be manually guided by a human operator or automatically guided by a flight controller. Case the manually guided option is chosen the commands $U_{d1}$ are directly sent to the AR.Drone 1 by the human operator through a smartphone or other control device. If the option is for the automatic control of the leader vehicle, the Trajectory Generator block provides the desired values for the navigation, which can be constant or time-varying values (respectively in the cases of positioning or trajectory-tracking tasks). The control is performed according to the hierarchical structure of Subsection IV-A, implementing (12) and (14) in the block Position Controller of Figure 3. The signals at the output of such block are sent to the ArDrone 1, which executes the internal orientation control loop in the code implemented in its onboard computer, which is not accessible to the designer.

There is a Kalman Filter block for each vehicle, which refers to an implementation based on [29], [30]. In this block the code correspondent to a state estimator is detailed, showing a method to feedback sensory data, such as positions and velocities of the UAV, in global coordinates, to the controllers. In the implementation of such block the calculation considers a data fusion involving raw data provided by a GPS module and by the inertial sensors, all of them provided by the onboard computer of the UAV in the data package $q$.

As for the follower UAV, in the Formation Generator block one should declare the desired structure for the formation, $X_{d2}$, which is controlled through the Formation Controller block, representing the implementation of (18) and (20). The signals at the output of such block are sent to the AR.Drone 2. It should be stressed that this last block also uses data correspondent to the AR.Drone 1. In such a case the GS shares the sensory information from the Kalman Filter 1 block and the computed (or manually set) control signals.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

In this section two experiments involving the ideas so far discussed are run, showing the practical validity of the assumptions considered. In both experiments the desired structure for
the formation is
\[
X_f = \begin{bmatrix} x_f - x_d \\ y_f - y_d \end{bmatrix} = \begin{bmatrix} 0.0 \\ 4.0 \end{bmatrix},
\]
which means that the follower vehicle should assume the same position of the leader, in the \( x \) direction, while staying 4 m apart of it, in the negative \( y \) direction. As for the \( z \) coordinate, the follower always stays at the same altitude of the leader, in this work.

### A. Automatic Navigation of the Leader Throughout Waypoints

In this experiment the leader is automatically controlled by the controller designed in Subsection IV-A, whereas the formation control is performed through the controller designed to the follower vehicle, described in Subsection IV-B. The objective is to navigate throughout a set of four waypoints, characterized as the vertices of a square with 3 m of side. Along the experiment the values of \( x_{d1} \) and \( y_{d1} \) are modified each \( t = 10 \) s, to change from one to the next vertex of the square, according to the following sequence

\[
\begin{align*}
x_{d1} &= 0.0 \text{ m}, \quad y_{d1} = 0.0 \text{ m} \\
x_{d1} &= 0.0 \text{ m}, \quad y_{d1} = 3.0 \text{ m} \\
x_{d1} &= 3.0 \text{ m}, \quad y_{d1} = 3.0 \text{ m} \\
x_{d1} &= 3.0 \text{ m}, \quad y_{d1} = 0.0 \text{ m}.
\end{align*}
\]

In Figure 4(a) it is possible to observe the sensory measures correspondent to the global positions of both UAVs for the experiment. Regarding the \( x \) coordinate, one should observe that both vehicles converge to the desired value \( x_{d1} \), as characterized by the specification adopted for the formation. As for the \( y \) coordinate, while the leader UAV converges to the desired value \( y_{d1} \), the follower UAV performs a similar movement, but 4 m aside the leader, as expected from the formation specification. Such results confirm that the follower UAV mimics the movement of the leader one, preserving the specified formation, in this case just a difference of 4 m in the \( y \) coordinate. By its turn, Figure 4(b) shows a comparison of the desired values for the formation (which are constant) with the values obtained from the data acquired by the sensors along the experiment. As one can notice, in spite of the oscillations in the curve, mainly due to the numerical errors inherent to the inertial sensors and to the GPS module, there is a convergence to the desired values correspondent to the formation. Such remarks lead to the conclusion that the controllers of the external loop of the leader and follower UAVs are able to accomplish the proposed task.

As for the rotational dynamics, Figures 4(c) and 4(d) show the orientations of the two vehicles along the experiment. The values called reference values are those resulting from (14) and (20), which are compared with the sensory data. One should notice that at each change of \( x_{d1} \) or \( y_{d1} \), great position errors are introduced, generating a saturation of the actuators of the vehicles. Even under such situation, however, it is possible to perceive in the curves the capability of the system to recover the reference orientation values. Thus, it is possible to confirm the convergence of the orientations to the reference values, even facing tracking imperfections, associated to noisy sensors. Even so, the real responses of the vehicles.
follow what was expected, from the analysis of the previous sections, in terms of the operation of the internal loop of orientation control, showing that the theoretical assumptions and simplifications adopted in the control design are consistent with the practical flight exposed in Figure 4. Finally, to allow the reader to better understand the experiment, a video of it is available in https://youtu.be/BU5piIVdP7w.

B. MANUALLY CONTROLLED LEADER NAVIGATION

In this experiment a human operator guides the leader UAV, from the ground station, whereas the follower UAV mimics the movement of the leader, keeping the specified formation all the time. In this case just the controller designed for the follower vehicle runs in the computer at the ground station. Figure 5(a) shows the global positions correspondent to both vehicles. There one can see, as expected, that the profile correspondent to $x$ and $y$ is quite similar for both UAVs, with the displacement of 4 m in $y$, thus showing the capability of the formation controller of keeping the specified formation. In Figure 5(b) there is a comparison of the desired values for the formation (constant values) and the values correspondent to the data acquired by the sensors onboard the vehicles along the experiment. As it can be observed, quite low errors in the formation variables are present, thus allowing claiming that the controller designed for the follower vehicle is effective in guiding it to keep the desired formation.

Figures 5(c) and 5(d) show how the leader and follower UAVs orient themselves along the experiment. For the leader UAV the reference orientation values are the commands exhibited in the form of pulses, which were sent by the human operator. For the follower UAV, the reference orientation values are calculated by the control system, according to the necessity to keep the desired aspect of the formation. As one can see, all the variables converge along the curves to reach and keep the desired formation.

Finally, still observing Figure 5, one can confirm that the developed system also presents response coherent with the theoretical analysis of the previous sections. To better understand the experiment, a video of it is available in https://youtu.be/IuX6_bToJDA.
VII. CONCLUSION

This work developed an approach to model quadrotor UAVs and design automatic flight controllers for them, considering two of them navigating according to the leader-follower formation paradigm. The controllers thus designed were tested in outdoor environments, using data provided by inertial sensors and GPS module onboard the vehicles as feedback to close the control loop. The possibility of giving a more practical interpretation to a traditional equation of literature was verified, making the design of controllers a simplified task. The controllers were tested with the UAVs performing real flights, shown in videos, providing a better interpretation of the topics covered. The assumptions and simplifications made to the model of the vehicle were validated in the experiments, where the expected behaviors were in fact observed during the flights. As a final remark it should be stressed that the same control system can be used to guide the leader-follower formation in trajectory-tracking tasks, without any change.

ACKNOWLEDGMENT

The authors thank CNPq–Conselho Nacional de Desenvolvimento Científico e Tecnológico, an agency of the Brazilian Ministry of Science, Technology, Innovations and Communications that supports scientific and technological development, and FAPES–Fundação de Amparo à Pesquisa e Inovação do Espírito Santo, an agency of the State of Espírito Santo, Brazil, that supports scientific and technological development, for the financial support granted to this research. They also thank UFES–Universidade Federal do Espírito Santo, IFES–Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo, UFV–Universidade Federal de Viçosa, and FAPEMIG–Fundação de Amparo à Pesquisa de Minas Gerais, for making possible to develop this work.

REFERENCES

[1] P. Castillo, A. Dzul, and R. Lozano, “Real-time stabilization and tracking of a four-rotor mini rotorcraft,” IEEE Trans. Control Syst. Technol., vol. 12, no. 4, pp. 510–516, Jul. 2004.
[2] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, “The GRASP multiple micro-UAV testbed,” IEEE Robot. Autom. Mag., vol. 17, no. 3, pp. 56–65, Sep. 2010.

[3] J. Engel, J. Sturm, and D. Cremers, “Scale-aware navigation of a low-cost quadcopter with a monocular camera,” Robot. Auton. Syst., vol. 62, no. 11, pp. 1646–1656, 201–6, 2017.

[4] M. C. P. Santos, C. D. Rosales, M. Sarcinelli-Filho, and R. Carelli, “A novel null-space-based UAV trajectory tracking controller with collision avoidance,” IEEE/ASME Trans. Mechatron., vol. 22, no. 6, pp. 2543–2553, Dec. 2017.

[5] M. C. P. Santos, C. D. Rosales, J. A. Sarapura, M. Sarcinelli-Filho, and R. Carelli, “An adaptive dynamic controller for quadrotor to perform trajectory tracking tasks,” J. Intell. Robot. Syst., vol. 93, nos. 1–2, pp. 5–16, Feb. 2019.

[6] D. K. D. Villa, A. S. Brandao, and M. Sarcinelli-Filho, “A survey on load transportation using multirotor uavs,” J. Intell. Robot. Syst., vol. 98, no. 2, pp. 1–30, 2020.

[7] M. A. Lewis and K.-H. Tan, “High precision formation control of mobile robots using virtual structures,” Auton. Robots, vol. 4, no. 4, pp. 387–403, 1997.

[8] A. Mahmood and Y. Kim, “Leader-following formation control of quadcopters with heading synchronization,” IEEE Trans. Robotics, vol. 48, no. 1, pp. 68–74, Dec. 2015.

[9] C. Rosales, P. Leica, M. Sarcinelli-Filho, G. Scaglia, and R. Carelli, “3D formation control of autonomous vehicles based on null-space,” J. Intell. Robot. Syst., vol. 84, nos. 1–4, pp. 453–467, Dec. 2016.

[10] J. Gimenez, D. C. Gandolfo, L. R. Salinas, C. Rosales, and R. Carelli, “Multi-objective control for cooperative payload transport with rotorcraft UAVs,” ISA Trans., vol. 89, pp. 491–502, Sep. 2018.

[11] X. Dong, Y. Hua, Y. Zhou, Z. Ren, and Y. Zhong, “Theory and experiment on formation-containment control of multirotor unmanned aerial vehicle systems,” IEEE Trans. Aerosp. Sci. Eng., vol. 16, no. 1, pp. 229–240, Jan. 2019.

[12] M. Aliehani and M. Inoue, “Performance evaluation of multi-UAV system in post-disaster application: Validated by HITL simulator,” IEEE Access, vol. 7, pp. 64386–64400, 2019.

[13] R. S. de Moraes and E. P. de Freitas, “Multi-UAV based crowd monitoring system,” IEEE Trans. Aerosp. Electron. Syst., vol. 56, no. 2, pp. 1332–1345, Apr. 2020.

[14] M. F. S. Rabelo, A. S. Brandao, and M. Sarcinelli-Filho, “Landing a UAV on static or moving platforms using a formation controller,” IEEE J. Robotic. Syst., vol. 15, no. 1, pp. 37–45, Mar. 2001.

[15] H. Wang, W. Chen, and J. Wang, “Coupled task scheduling for heterogeneous multi-robot system of two robot types performing complex schedule order fulfillment tasks,” Robot. Auton. Syst., vol. 131, Sep. 2020, Art. no. 103560.

[16] E. Olicay, F. Schuhmann, and B. Lohmann, “Collective navigation of a multi-robot system in an unknown environment,” Robot. Auton. Syst., vol. 132, Oct. 2020, Art. no. 103604.

[17] Z. Hou and I. Fantoni, “Interactive leader-follower consensus of multiple quadrotors based on composite nonlinear feedback control,” IEEE Trans. Control Syst. Technol., vol. 26, no. 5, pp. 1732–1743, Sep. 2018.

[18] S. Xingling, T. Biao, Y. Wei, and Z. Wendong, “Estimator-based MLP neural-adaptive dynamic surface containment control with prescribed performance for multiple quadrotors,” Aerosp. Sci. Technol., vol. 97, Feb. 2020, Art. no. 105620.

[19] C.-W. Chang and J.-K. Shiau, “Quadrotor formation strategies based on distributed consensus and model predictive controls,” Appl. Sci., vol. 8, no. 11, p. 2246, Nov. 2018.

[20] Z. Zhao and J. Wang, “Leader-follower formation control of multiple quadrotors,” in Proc. IEEE CSSA Guid., Navigat. Control Conf. (CGNCC), Oct. 2018, pp. 1–6.

[21] H. Du, W. Zhu, G. Wen, and D. Wu, “Finite-time formation control for a group of quadrotor aircraft,” Aerosp. Sci. Technol., vol. 69, pp. 609–616, Oct. 2017.

[22] X. Ai and J. Yu, “Flatness-based finite-time leader-follower formation control of multiple quadrotors with external disturbances,” Aerosp. Sci. Technol., vol. 92, pp. 20–33, Sep. 2019.

[23] Y. Dong, C. Fu, and E. Kayacan, “RRT-based 3D path planning for formation landing of quadrotor UAVs,” in Proc. 14th Int. Conf. Control. Autom., Robot. Vis. (ICARCV), Nov. 2016, pp. 1–6.

[24] P.-J. Briseau, F. Callou, D. Vissière, and N. Petit, “The navigation and control technology inside the AR.Drone micro UAV,” IFAC Proc. Volumes, vol. 44, no. 1, pp. 1477–1481, Jan. 2011.

[25] S. Piskorski, N. Brulez, P. Eline, and F. D’Haeyer, AR.Drone Developer Guide SDK Version 2.0. Paris, France: Parrot, Dec. 2012.

[26] F. Kendoul, Z. Yu, and K. Nonami, “Guidance and nonlinear control system for autonomous flight of minirotorcraft unmanned aerial vehicles,” J. Field Robot., vol. 27, no. 3, pp. 311–334, May 2010.

[27] A. S. Brandao, M. S. Filho, and R. Carelli, “High-level underactuated nonlinear control for rotorcraft machines,” in Proc. IEEE Int. Conf. Mechatronics (ICM), Feb. 2013, pp. 279–285.

[28] S. Betrand, N. Guérard, T. Hamel, H. Piet-Lahanier, and L. Eck, “A hierarchical controller for miniature VTOL UAVs: Design and stability analysis using singular perturbation theory,” Control Eng. Pract., vol. 19, no. 10, pp. 1099–1108, Oct. 2011.

[29] L. V. Santana, A. S. Brandao, and M. Sarcinelli-Filho, “An automatic flight control system for the AR.Drone quadrotor in outdoor environments,” in Proc. Workshop Res., Educ. Develop. Unmanned Aerial Syst. (RED-UAS), Nov. 2015, pp. 401–410.

[30] L. V. Santana, A. S. Brandao, and M. Sarcinelli-Filho, “An open-source testbed for outdoor navigation with the AR.Drone quadrotor,” IEEE Syst. J., early access, Sep. 9, 2020, doi: 10.1109/JSYST.2020.3020012.

**LUCAS VAGO SANTANA** received the degree in control engineering and automation from the Federal University of Ouro Preto, and the master’s and Ph.D. degrees in electrical engineering from the Federal University of Espírito Santo. He has published 3 journal articles, 1 book chapter, and 18 conference papers. He has experience in the electronic automation of electrical and industrial processes, working mainly in the subjects of electronics, robotics, and control.

**ALEXANDRE SANTOS BRANDÃO** received the B.S. degree in electrical engineering from the Federal University of Viçosa, Minas Gerais, Brazil, in 2006, the M.Sc. and Ph.D. degrees in electrical engineering from the Federal University of Espírito Santo, Brazil, in 2008 and 2013, respectively, and the Ph.D. degree in control system engineering from the National University of San Juan, San Juan, Argentina, in 2014. He has coauthored more than 20 journal articles, 2 book chapters, and more than 120 conference papers, besides having advised 8 M.Sc. and 4 Ph.D. students. His research interest includes nonlinear control of aerial and terrestrial mobile vehicles.

**MÁRIO SARCINELLI-FILOHO** received the B.S. degree in electrical engineering from the Federal University of Espírito Santo, Brazil, in 1979, and the M.Sc. and Ph.D. degrees electrical engineering from the Federal University of Rio de Janeiro, Brazil, in 1983 and 1990, respectively. He is currently a Professor with the Department of Electrical Engineering, Federal University of Espírito Santo, and a Researcher of the Brazilian National Council for Scientific and Technological Development (CNPq). He has coauthored more than 60 journal articles, more than 340 conference papers, and 17 book chapters, besides having advised 20 Ph.D. and 25 M.Sc. students. His research interests include nonlinear control, mobile robot navigation, multi robot systems, coordinated control of mobile robots, unmanned aerial vehicles, and coordinated control of ground and aerial robots. He is a Senior Editor of the Journal of Intelligent and Robotic Systems.