

Obstacle-aware UAV navigation based on a leader-follower consensus protocol

E.U. Rojo-Rodriguez¹, E.G. Rojo-Rodriguez¹, E. J. Ollervides-Vazquez^{1,2}, P. Castillo³ and O. Garcia-Salazar¹

Abstract—This work aims to present a navigation scheme whose topmost layer further considers the presence of obstacles and steers the aircraft towards a safe path by altering its navigation references. The complete framework is divided into two control layers carrying out the orientation and navigation, while a third obstacle-aware layer, which is based on a Virtual Leader-Follower coordination problem, generates the positioning references for the navigation controller. This outline allows designing the inner control layers independently, without obstacle awareness, giving flexibility to different combinations of UAVs and controller laws. To validate the proposed structure, numerical simulations are performed where moving obstacles are considered. In addition, experimental testing is carried out by implementing the three-layer navigation in a quadrotor aircraft, performing an independent trajectory while in the presence of a static obstacle. Both tests show acceptable results, and the obstacle collision avoidance is reached.

I. INTRODUCTION

There are numerous applications and fields where an Unmanned Aerial Vehicle (UAV) could be used. In the past decade, many researchers have focused their goals toward the solution or study of different application problems with specific criteria, such as navigation, obstacle avoidance, artificial vision, load carrying, among similar, [1], [2], [3].

Controlling the dynamics of the aircraft has proven to be the core problem to solve. Nonetheless, very successful developments have already proven the feasibility of their implementation and usage; different approaches ranging from traditional Proportional Integral Derivative (PID) controllers to more complex non-linear schemes have been tested and shown different advantages and benefits [4], [5]. These advances in the field allow the development of more sophisticated algorithms and applications that were not fully covered. Even though it may seem simple enough to implement an already existing scheme, different scenarios require a specific combination of UAV and corresponding control strategies, preventing from standardization of higher-level algorithms.

This drawback is considerable enough when security concerns arise as a consequence of taking the implementation

outside the controlled testing environment. In an actual UAV application, the complete mission must be performed, accounting for external disturbances, unexpected faults, and all possible unwanted interactions with the working environment, such as obstacles. The latter gains special attention as it represents a high percentage of UAV mission failures [6], so avoiding obstacle collisions is a navigation problem that has been heavily discussed in recent years. Having a different combination of aircraft types with even more diverse approaches of controller laws leads to designing obstacle avoidance strategies specifically for the implementation; as a consequence of this, it is challenging to come up with a general usage algorithm.

There are many schemes developed to get a suitable response against an obstacle. For instance, the most common approach is the anticipated path planning, where the trajectory followed by the UAV is designed with some degree of obstacle awareness. One example is the work presented in [7], where an offline path planning is carried out by means of a combination of a rapidly-exploring random tree (RRT) and potential fields to generate the most efficient path, this for a delivery quadrotor; although the authors reached their proposed goal, the main disadvantage become evident as all obstacles locations must be known before the algorithm is calculated. A similar development was presented in [8], in which an RRT algorithm is also used, in which the detection of static obstacles is also considered non-previously known obstacles, with the limitation of being static due to the nature of the path generation algorithm. This type of approach could also be implemented using different kinds of schemes, from evolutionary algorithms such as the work of [9], and [10], or even direct interaction algorithms, more focused on reaction time, and avoiding the collision with simple commands, [11].

While all these developments present a suitable way of solving the avoidance problem, they must be specifically designed or at least adapted to the specific mission and aircraft used. This has been addressed by defining working constraints; for instance, the dynamic obstacle avoidance solves the problem of moving obstacles, and in many cases, the algorithms are designed as velocity references most aircraft can follow; always considering its type constraints, such as turning radius for fixed-wing UAVs. Applications such as the presented in [12] or in [13], validate the usage of virtual fields to steer the UAV far from the obstacle, in which, if the navigation is independent, the field only needs to consider velocity constraints. On the other hand,

¹, E. U. Rojo-Rodriguez, E.G. Rojo-Rodriguez, E. J. Ollervides-Vazquez, and O. Garcia-Salazar are with Aeronautical Engineering Research and Innovation Center, CIIA-FIME-UANL, Monterrey Nuevo Leon, Mexico. edgar.rojordrg@uanl.edu.mx, erojor@uanl.edu.mx, edmundo.ollervidesvz@uanl.edu.mx, octavio.garciasal@uanl.edu.mx

²E. J. Ollervides-Vazquez is with La Laguna Institute of Technology-TecNM, Torreon Coahuila, Mexico. ejollervi@ieee.org

³P. Castillo is with Heudiasyc Laboratory CNRS UMR 7253, Université de Technologie de Compiègne, Compiègne France, castillo@hds.utc.fr

the most recent development is focused on mechanisms of self-adaptation and learning as a possible solution against obstacles, which can be considered uncertainties. As an example, authors in [14] developed a framework of avoiding dynamic obstacles by sensing and identifying the obstacle using a convolutional neural network, and with the relative positioning data, determine the necessary turning angle and relative velocity. More recent developments deal with the problem by applying the Reinforced Learning approach, as a way of altering the ideal reference the UAV follows, [15], [16].

The aim of this work is directed towards the proposal of a UAV navigation scheme that can be considered independent of the inner control layers of the aircraft while at the same time assuring the success of the navigation in an environment where static and dynamic obstacles could be present. A 3-layer control scheme is proposed, where the topmost layer corresponds to a *reference* generation, which in this sense can be separated from the remaining layers, so different configurations can be implemented. The algorithm is based on a multi-agent system (MAS) approach, where virtual coordination is defined between the ideal trajectory reference, considered as a virtual leader, and the actual navigation states of the UAV, while considering a virtual velocity vector in each obstacle, as added velocities in the coordination protocol.

The remaining document is divided as follows: in Section II necessary mathematical tools are presented and defined to further develop the reference generator. Section III deals with the navigation carried out by the reference generation, which is based on a coordinated MAS, while validations through numerical simulations and a real-time experiment are described in Section IV. Conclusions are finally presented in Section V.

II. MATHEMATICAL TOOLS

A. Multi-rotor UAV dynamics

As an starting procedure, it is necessary to define the dynamic equation governing the aircraft, which for this case a quadrotor configuration is used. Different reference frames are considered while defining the equations of motion; for instance, a ground-fixed inertial framed defined as $\mathcal{I}_i = \{x_{\mathcal{I}_i}, y_{\mathcal{I}_i}, z_{\mathcal{I}_i}\}$ in which translational effects are modeled, a body frame attached to the center of gravity of the UAV as $\mathcal{B}_i = \{x_{\mathcal{B}_i}, y_{\mathcal{B}_i}, z_{\mathcal{B}_i}\}$, and an aerodynamic frame, $\mathcal{W}_i = \{x_{\mathcal{W}_i}, y_{\mathcal{W}_i}, z_{\mathcal{W}_i}\}$, that considers external forces such as wind gusts, changes in angle of attack and sideslip angle, and other aerodynamic effects, [17].

Newton-Euler formulation is used to describe the equations of motion of the UAV as follows

$$\dot{\xi}_i = V_i \quad (1)$$

$$m_i \dot{V}_i = (-T_i)R_i e_3 + m_i g e_3 + D_{\xi_i} \quad (2)$$

$$\dot{R}_i = R_i \hat{\Omega}_i \quad (3)$$

$$J_i \dot{\Omega}_i = -\Omega_i \times J_i \Omega_i + \tau_{a_i} + D_{\eta_i} \quad (4)$$

with $i = 1, \dots, N$ corresponding to an specific UAV if a formation of multiple aircrafts is defined, $\xi_i = (x_i, y_i, z_i)^\top \in \mathbb{R}^3$ are the position coordinates relative to the inertial fixed frame and $\eta_i = (\phi_i, \theta_i, \psi_i)^\top \in \mathbb{R}^3$ describes the rotation states of the i -th UAV, which are described by an orthogonal rotation matrix $R_i \in SO(3) : \mathcal{B}_i \rightarrow \mathcal{I}_i$ parametrized by the Euler angles ϕ_i roll, θ_i pitch and ψ_i yaw.

Angular velocities in the body frame \mathcal{B}_i are defined as $\Omega_i = (p_i, q_i, r_i)^\top \in \mathbb{R}^3$, while linear velocities in the inertial fixed frame \mathcal{I}_i as $V_i = (\dot{x}_i, \dot{y}_i, \dot{z}_i)^\top \in \mathbb{R}^3$. $T_i \in \mathbb{R}_{>0}$ is the total thrust of all the UAV actuator, and $\tau_{a_i} \in \mathbb{R}^3$ are the moments relative due the body frame, and as a result of actuators combinations.

Vectors of canonical basis of \mathbb{R}^3 are also considered, and are represented by e_1, e_2 , and e_3 . The term $m_i \in \mathbb{R}$ defines the mass of the i -th UAV, while $J_i \in \mathbb{R}^{3 \times 3}$ describes the moments of inertia. $\hat{\Omega}_i$ is the skew-symmetric matrix associated with the cross product (i.e. $\hat{a}b = a \times b \quad \forall a, b \in \mathbb{R}^3$). $D_{\xi_i} = (d_{\xi_{i1}}, d_{\xi_{i2}}, d_{\xi_{i3}})^\top \in \mathbb{R}^3$ and $D_{\eta_i} = (d_{\eta_{i1}}, d_{\eta_{i2}}, d_{\eta_{i3}})^\top \in \mathbb{R}^3$ are disturbances vectors, [18], [19], [20].

B. The General Consensus Protocol

Multi-agent systems are widely used in the coordinated flight of UAVs. Many approaches have been proposed; for instance, the *Consensus Protocol* scheme has been proven to be a reliable and robust approach to achieve coordinated motion.

On its most basic structure, a *Consensus protocol* gets the following form

$$u_{\zeta_{a_i}} = K_{P_i} \left(\sum_{j=1}^N a_{ij} (\zeta_j - \zeta_i) \right) + K_{D_i} \left(\sum_{j=1}^N a_{ij} (\dot{\zeta}_j - \dot{\zeta}_i) \right) \quad (5)$$

where K_{P_i} is a positive diagonal matrix whose elements guarantee the consensus on the position states, K_{D_i} represents a positive diagonal matrix containing terms for reaching consensus of velocity states. In addition, ζ_i represents the states of the agent, while ζ_j the state of its neighbor agent, with $i = 1, 2, 3, \dots, N$, $j = 1, 2, 3, \dots, N$, being N the total number of agents in the formation. It is worth to note that a_{ij} sets the communication strength from agent j to i .

C. Virtual-leader formation flight

One of the most common approaches to achieving a coordinated flight by using a multi-agent system is the so-called *Leader-Follower Consensus (LFC)*, whose main objective is to drive the i -th follower to the states of a time-varying reference, that can be a simulated agent or an ideal numerical reference. The advantages of this model rely on the simplicity of the protocol design due to the independence of the leader, which in this sense, partially solves some phenomena such as noise amplification through the multi-agent system communication topology.

From the point o view of the *Consensus protocol*, no changes are introduced in the basic structure, despite the fact that the neighbor agent is the virtual leader and thus is not subjected to the consensus. Nevertheless, it is important to

guarantee that the actual agent is reaching a consensus with the virtual leader, meaning the following definition can be addressed.

Definition 1: An agent is said to reach consensus with the virtual leader if there exist a Consensus protocol (5), such that the closed-loop system satisfies

$$\lim_{t \rightarrow \infty} \|(\zeta_j(t) - \zeta_i(t)) - \xi_i(0)\| = 0 \quad (6)$$

for all possible geometric formation constrains, considered as the initial condition vector $\xi_i(0)$.

For the case of this implementation, no formation distances are considered, so $\xi_i(0) = 0$.

III. NAVIGATION BASED ON A CONSENSUS PROTOCOL

As an alternative, to drive the states of the UAV to a secure position, considering its reference vector and the position of obstacles, a 3-layer control scheme is proposed. Rewriting equations (1)-(4), these are described as follows

$$\dot{\xi}_i = V_i \quad (7)$$

$$\dot{V}_i = u_{p_i} + d_{\xi_i} \quad (8)$$

$$\dot{R}_i = R_i \hat{\Omega}_i \quad (9)$$

$$\dot{\Omega}_i = u_{a_i} + d_{R_i} \quad (10)$$

with $u_{p_i} = ge_3 - \frac{T_{T_i}}{m_i}(R_i e_3)$, $\tau_{a_i} = Ju_{a_i}$, $d_{\xi_i} = \frac{D_{\xi_i}}{m_i}$ and $d_{R_i} = J_i^{-1}[-\Omega_i \times J_i \Omega_i + D_{\eta_i}]$. This leads to the definition of virtual control inputs $u_{p_i} \in \mathbb{R}^3$ and $u_{a_i} \in \mathbb{R}^3$ for the translational dynamics and rotational dynamics, respectively. As the most inner and fast layer u_{a_i} , a local closed-loop controller is designed, whose reference vector is the output of the following controller loop u_{p_i} , which deals with the local navigation of the aircraft. These first two layers are designed as saturated *Proportional Integral Derivative (PID)* controllers. For the in-detail development of these loops, see the previous work on [21].

The actual references for the navigation loop come from the topmost layer, which in this case is proposed as a modified LFC protocol whose configuration guarantees coordination between the virtual leader and the states of the UAV, but at the reference level. In this sense, the protocol acts as a reference generator for the navigation layer by coordinating the tracking errors between the aircraft and the virtual leader; this can be defined as

$$\begin{aligned} E_{\zeta_i} &= e_{\zeta_j} - e_{\zeta_i} \\ \dot{E}_{\zeta_i} &= \dot{e}_{\zeta_j} - \dot{e}_{\zeta_i} \end{aligned} \quad (11)$$

being e_{ζ_j} the tracking error for the j agent, or in this case the virtual leader, e_{ζ_i} , the tracking error for the UAV, \dot{e}_{ζ_j} and \dot{e}_{ζ_i} its corresponding time derivatives. It is worth to note that having a virtual leader configuration leaves the agent j errors at zero, being an ideal tracking.

For the actual LFC protocol acting as a reference generator, the following structure is proposed

$$\begin{aligned} u_{\zeta_{d_i}} &= K_S \zeta_i(t) + K_{CL} e_{\zeta_i} + K_{ob} \hat{e}_{\zeta_i} + \\ &K_{P_i} \left(\sum_{j=1}^N a_{ij} E_{\zeta_i} \right) + K_{D_i} \left(\sum_{j=1}^N a_{ij} \dot{E}_{\zeta_i} \right) \end{aligned} \quad (12)$$

where the terms on the first line correspond to a closed-loop designed to generate unit compatible references for the navigation layer; K_S is a matrix of gains that guarantee that the output is kept when consensus is reached, K_{CL} is a matrix of gains that drives the output to the actual desired states to reach, and K_{ob} is a matrix of gains that assures the generated reference will avoid the present obstacles. The second line of terms is designed to assure the tracking of the time-varying reference of the virtual leader, where K_{P_i} are the gains that assure position consensus while the K_{D_i} matrix the gains for the velocity consensus. The introduced term \hat{e}_{ζ_i} has the task to coordinate the rate at which the protocol change the output taken as navigation reference; and at the same time assures that, in the case of a possible collision, it would drive the output references out of a *detection zone*. This term is defined as

$$\hat{e}_{\zeta_i} = (\dot{\zeta}_{d_i}(t) + V_{ob_i}) - \dot{\zeta}_i(t) \quad (13)$$

where V_{ob_i} is and induced velocity depending on the position and orientation relative to the detection zone, further defined as the contribution of all obstacles present in the field of sensing of the UAV as

$$V_{ob_i} = \sum_{m=1}^M \|\mathcal{F}_m(x_i, y_i, z_i)\| \quad (14)$$

with $m = 1, 2, 3 \dots, M$ having M as the total number of obstacles, and $\mathcal{F}_m(x_i, y_i, z_i)$ as a vector space that defines the detection zone of each obstacle depending on its own euclidean position and the UAV state vector

$$\begin{aligned} \|\mathcal{F}_m\| &= \\ &\left(a_x e^{-\frac{(x_i - b_x(t))^2}{2c_x^2}} \right) \left(a_y e^{-\frac{(y_i - b_y(t))^2}{2c_y^2}} \right) \left(a_z e^{-\frac{(z_i - b_z(t))^2}{2c_z^2}} \right) \end{aligned} \quad (15)$$

having a_x, a_y, a_z as the magnitude of added velocities, $b_x(t), b_y(t), b_z(t)$ are the coordinates of the obstacle relative to the inertial fixed frame, and c_x, c_y, c_z the geometrical extension of the detection zone in each axis. As an example, Fig. 1, shows the plot for a velocity field with a coverage of 500mm in radius, and placed at the origin.

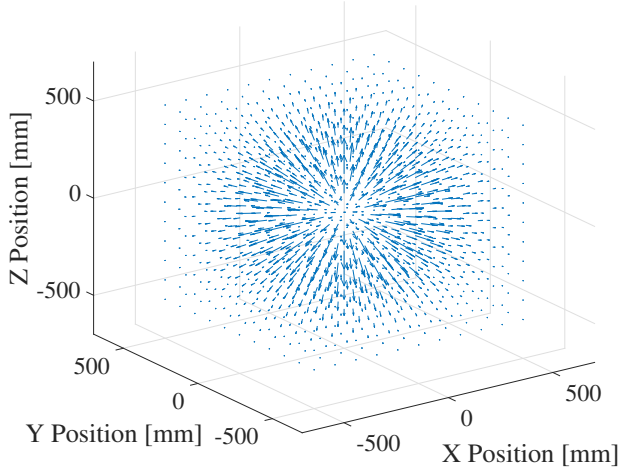


Fig. 1: Representation of the directions of a added velocities field with 500 radius, positioned at the origin.

To summarize the complete scheme, in Fig. 2, a graphical representation is shown. Note that the reference the UAV must perform is independent of the other algorithms, so it is possible to have collision scenarios. To deal with the previous situation, the actual references that the UAV must follow, are coordinated by the LFC protocol, which has the task of generate a compatible modified reference, considering the obstacles states, so the navigation controller assures its following, by means of the orientation controller.

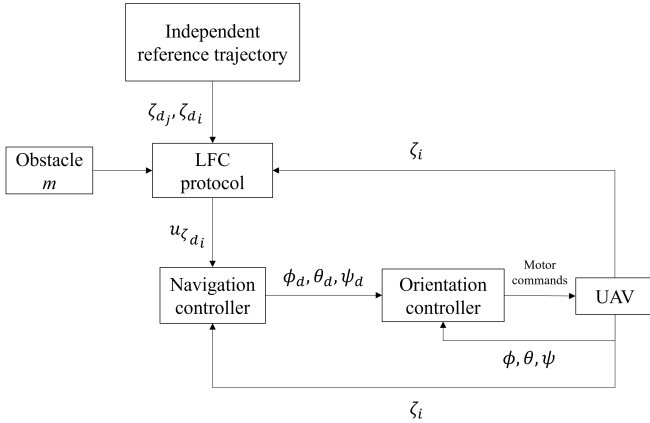


Fig. 2: Graphical representation of the complete navigation scheme.

IV. VALIDATION

For validation purposes, a set of simulations and an experimental test are carried out. Two scenarios are simulated, a hover flight with a moving obstacle and a circular trajectory tracking with a crossing obstacle. For the case of the experimental test, a circular trajectory with an obstacle in a collision position is tested.

A. Numerical simulations

Numerical simulations were performed in the C++ programming language, using the Euler integration method for solving the ODEs, and at 50Hz with a high priority in the OS scheduler. All algorithms ran in a computer with an Intel Core i7 9750H processor and 24 GB of RAM memory.

The simulated quadrotor is based on a Parrot AR. Drone 2.0, whose parameters and coefficients were proposed by [22].

1) *Hover Flight*: For the first numerical simulation scenario, the quadrotor aircraft is commanded with a fixed reference point at the origin, with a reference height of 1500mm. An obstacle with a detection zone of 750mm of radius follows a straight path starting at $X = 1750\text{mm}$ and 1500mm, with a velocity of 100mm/s., which allows the obstacle to interact directly with the aircraft, as its detection zone fully covers the original position of the UAV at some point.

In Fig. 3 the resultant trajectory is displayed in the XY plane, as well as the obstacle path and two instants of interaction. Note that at 19s, the detection zone of the obstacle gets close to the UAV, but the last does not move until they start to interact. At 26s, the detection zone is fully covering the original position of the aircraft; however, it is moved to a secure position, as depicted by the blue trajectory; meaning that, while the obstacle is getting close, the quadrotor is gradually moving by the effect of the added velocities.

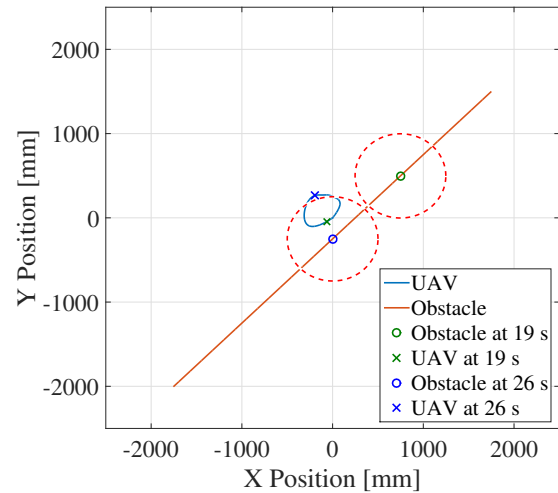


Fig. 3: Trajectories followed by the UAV and the obstacle in the hover flight scenario.

Figure 4 shows the previous behavior in the form of the individual $X - axis$ component. The same can be seen; this is, how the trajectory of the aircraft is disturbed from its reference in order to avoid the obstacle.

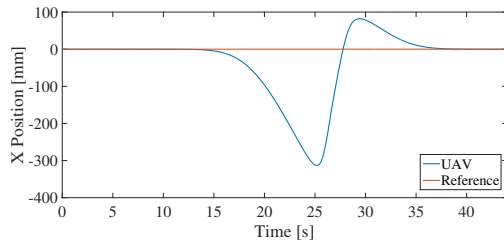


Fig. 4: X-axis trajectory followed by the UAV in the hover flight scenario.

In Fig. 5, the $Y - axis$ component is presented. Similar behavior is displayed, with the only difference being the sign and magnitude of the movement due to the direction of approximation of the obstacle.

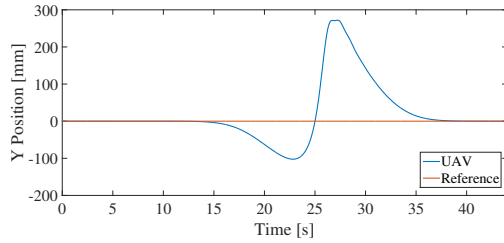


Fig. 5: Y-axis trajectory followed by the UAV in the hover flight scenario.

Finally, the $Z - axis$ component can be seen in Fig. 6, where no evident changes are presented, due to the fact that the obstacle is at the same height that the aircraft, so these do not interact in this axis.

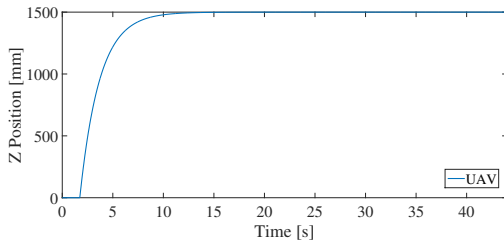


Fig. 6: Z-axis trajectory followed by the UAV in the hover flight scenario.

The core of the scheme is the actual references generated by the FLC protocol, which in essence have the task of dealing with external interactions; this is, the obstacles. As shown in Fig. 7, the behavior displayed by the aircraft is a consequence of the references generated by the protocol, whose outputs are the result of the avoidance of the obstacle passing by.

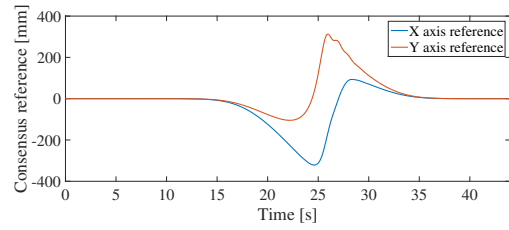


Fig. 7: Generated consensus references in the hover flight scenario.

Not less important, it is worth to display the trajectory of the obstacle, which for this case is graphed in Fig. 8, accounting for a constant height. As previously stated, both coordinates do not start at the same value; this is a way of preventing an ideal 45° approximation and having a better evaluation scenario.

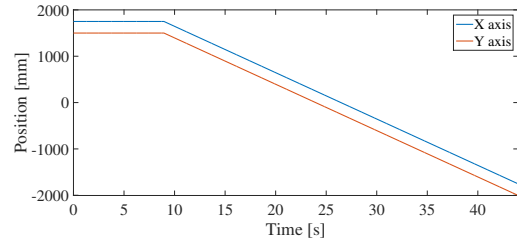


Fig. 8: Trajectories of the obstacle in the hover flight scenario.

2) *Trajectory tracking with obstacle intersection:* As another test scenario, a circular trajectory is performed. The radius of the circumference trajectory is $1750mm$, while the commanded speed is fixed at $0.03rad/s$. The trajectory for the obstacle remained at the same speed as the previous scenario, but with equal starting coordinates at $2100mm$, in order to assure two interactions with the actual aircraft. In Fig. 9, an XY plane representation of the simulation is shown, where two instants of the whole trajectory are extracted, the first at $11.4s$ and the second when the time is at $31.6s$. It is worth to note that in the first instant, the detection zone is interacting with the ideal reference; thus, the LFC protocol commanded a change of trajectory to the actual aircraft, allowing it to avoid the first encounter. When the quadrotor is at about $3/4$ of the cycle, the second interaction happened, and as seen in the Figure, now the LFC protocol commanded an avoidance getting ahead of the obstacle instead of retarding the ideal trajectory.

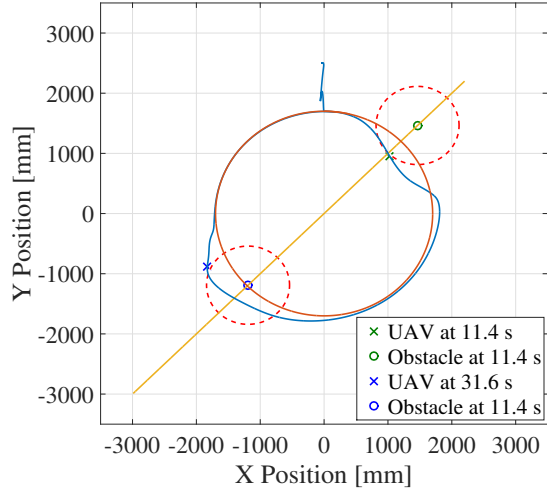


Fig. 9: Trajectories followed by the UAV and the obstacle in the trajectory tracking flight scenario.

Both interactions can also be seen in Fig. 10, and Fig. 11, for the individual X - *axis* and Y - *axis*, noting that at the two instants when the collision is occurred, a sudden deviation from the ideal reference happened, as a result of the LFC protocol changing the navigation reference.

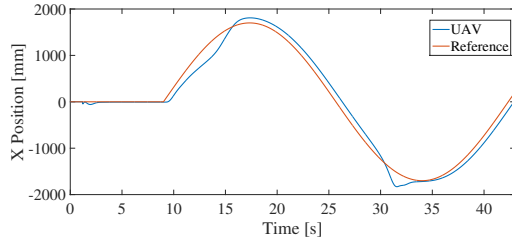


Fig. 10: X-axis trajectory followed by the UAV in the trajectory tracking flight scenario.

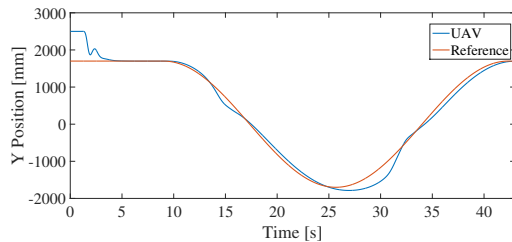


Fig. 11: Y-axis trajectory followed by the UAV in the trajectory tracking flight scenario.

Figure 12, displays the behavior in the Z - *axis*, with null interaction as both the obstacle and the UAV are at equal heights.

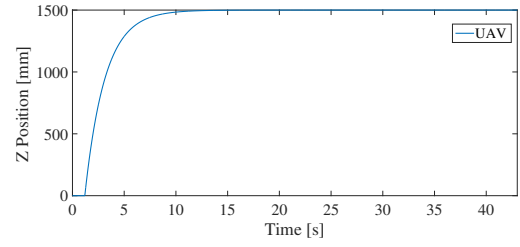


Fig. 12: Z-axis trajectory followed by the UAV in the trajectory tracking flight scenario.

Note the references trajectories generated by the LFC protocol, which are displayed in Fig. 13. These signals are responsible for the avoidance behavior described by the UAV.

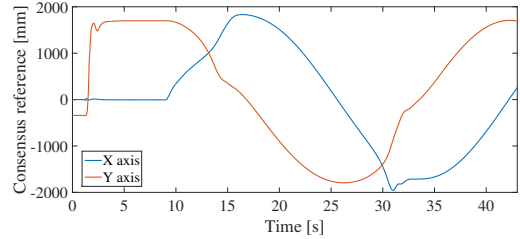


Fig. 13: Generated consensus references in the trajectory tracking flight scenario.

The trajectory of the obstacle remains equal in both movement axis, so they overlap in Fig. 14, meaning it moved at the same rates in both directions.

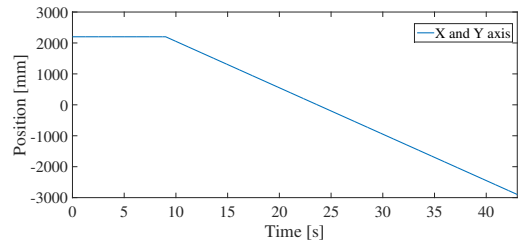


Fig. 14: Trajectories of the obstacle in the trajectory tracking flight scenario.

B. Real-time experiment

A real-time experimental test is also carried out to further validate the behavior of the whole scheme. A circular trajectory is commanded to the quadrotor aircraft, while a static obstacle is placed directly in the ideal path.

1) *Experimental platform:* As experimental platform a quadrotor aircraft is used, as well as a motion capture system. The testing facilities are located at the Navigation Laboratory, of the Aeronautical Engineering Research and Innovation Center (CIIRA), of the Autonomous University of Nuevo Leon (UANL).

The motion capture system consists of 16 VICON T40 cameras, streaming position and orientation data to a centralized computer at $100Hz$, with an accuracy of $0.1mm$ and 0.1° . The quadrotor aircraft used is designed and built for

testing purposes in the same facilities test, and it has basic characteristics shown in Table I. Fig. 15 shows the actual aircraft. The embedded avionics consists of a real-time dual core processor with up to 200 Mhz, where one core deals with communications and the remaining with the attitude estimation and controllers execution, an onboard 9-DOF inertial measurement unit used for attitude estimation, and 8 PWM outputs running at $50Hz$ of sample rate, of which 4 are used as motor outputs. FLC protocol and navigation control are processed in the same central computer, which deals with the position estimation, while in the onboard avionics, the attitude control is processed, using the estimated Euler angles.

Quadrotor specs	
Parameter	Specification
Type	X
Distance from C.G. to motor	12.5 cm
Weight	249 gr
Single motor maximum thrust	1.9 N

TABLE I: Quadrotor specifications and parameters.



Fig. 15: Quadrotor used for the experimental tests.

2) *Trajectory tracking with obstacle intersection:* The experimental circumference followed is defined with a radius of $1500mm$ and an angular velocity of $0.02rad/s$, with a fixed height at $1200mm$. The obstacle is placed at $X = -1425mm$ and $Y = -20mm$ with a virtual height of $1200mm$; this is, although the physical obstacle is short, a grater height is commanded as a security measure. The actual size of the obstacle is $350mm$ of radius, while the detection zone is established at $750mm$.

In Fig. 16 the ideal references and actual trajectory performed by the quadrotor is shown. The obstacle and its detection zone are also displayed as a blue circle and red dotted circle, respectively. It is possible to note that in this case, the aircraft actually crosses through the detection zone; despite this, no collision occurred due to the fact that the detection zone is established with a section that is greater than the actual obstacle. This behavior is expected knowing the fact that the actual quadrotor used for the physical experiment has a slower response time in comparison with the simulated

platform. Nonetheless, the performance exhibited is taken as acceptable, as the goal of avoiding the obstacle is met. It is important to note, that the velocity field not only affects in close proximity or within the detection zone; having an exponential decay of the field means that a small residue of velocity is going to be present, more notoriously in the experimental test.

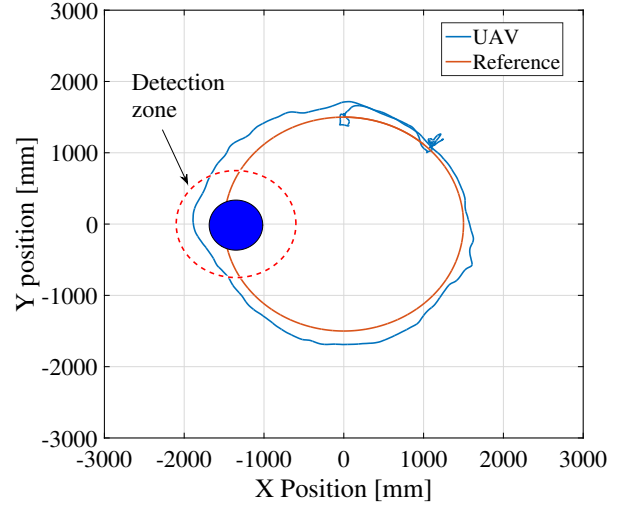


Fig. 16: Trajectories followed by the UAV and the obstacle in the experimental trajectory tracking flight scenario.

The same behavior is graphed in Fig. 17, in the 3 axes of movement. The velocity field is also shown, depicting the detection zone that affects the aircraft. Note how the UAV successfully avoids the obstacle.

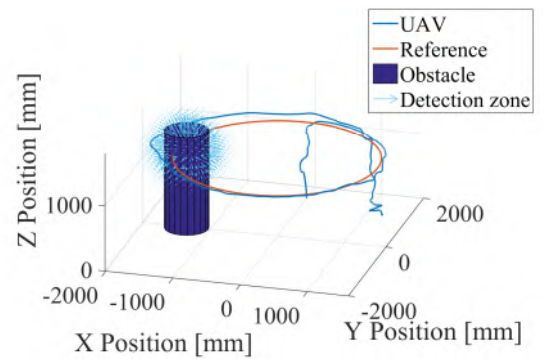


Fig. 17: 3D trajectories followed by the UAV and the obstacle in the experimental trajectory tracking flight scenario.

Individual axis signals are shown in Fig. 18 for the X - axis, in Fig. 19 for the Y - axis and for the Z - axis in Fig. 20. Obstacle interaction occurred at approximately at $55s$, which can be observed as the disturbance in the signal in comparison with the ideal reference.

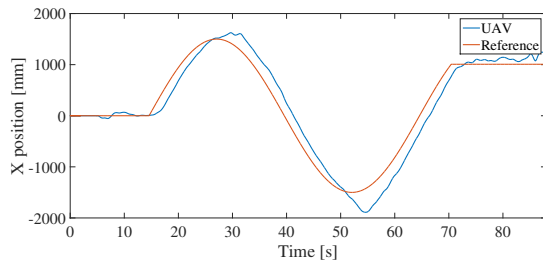


Fig. 18: X-axis trajectory followed by the UAV in the experimental trajectory tracking flight scenario.

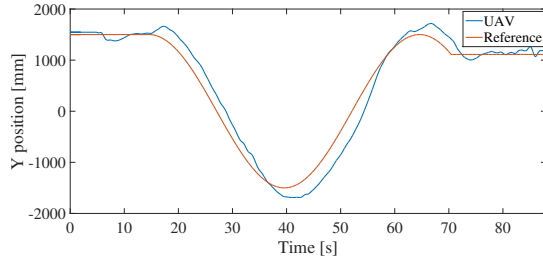


Fig. 19: Y-axis trajectory followed by the UAV in the experimental trajectory tracking flight scenario.

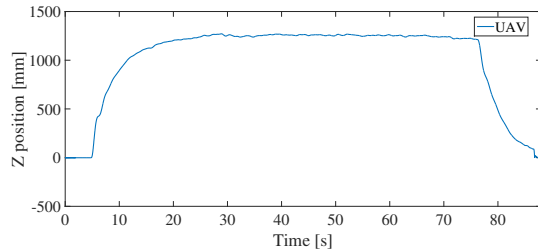


Fig. 20: Z-axis trajectory followed by the UAV in the experimental trajectory tracking flight scenario.

Previous performance is result of the actual navigation references generated by the FLC protocol, displayed in Fig. 21.

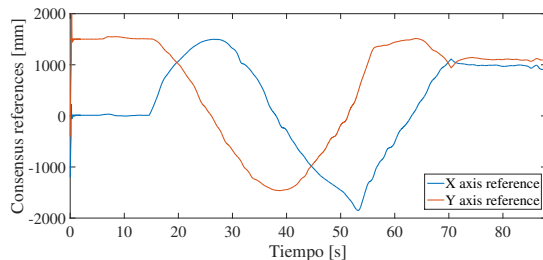


Fig. 21: Generated consensus references in the experimental trajectory tracking flight scenario.

This experimental test was captured on video and is available on the following link <https://youtu.be/IiYC9jNboEQ>.

V. CONCLUSIONS

In this work, a navigation scheme is presented, based on an implementation of the Leader-follower consensus, having the ideal reference as a virtual leader. In addition, the LFC is designed with awareness of obstacles, defining a virtual velocity field, so the consensus condition is changed during the fly. Numerical simulations are performed to validate the effectiveness of the approach, as well as an experimental scenario. Acceptable results are obtained, where the main goal is achieved, considering the limitations of the aircraft used and the necessary detection zones and time response.

ACKNOWLEDGMENT

This research work is supported by the Office of Naval Research Global through the grant number N62909-20-1-2030

REFERENCES

- [1] A. A. Zhilenkov, S. G. Chernyi, S. S. Sokolov, and A. P. Nyrkov, "Intelligent autonomous navigation system for uav in randomly changing environmental conditions," *Journal of Intelligent & Fuzzy Systems*, vol. 38, no. 5, pp. 6619–6625, 2020.
- [2] M. Radmanesh, M. Kumar, P. H. Guentert, and M. Sarim, "Overview of path-planning and obstacle avoidance algorithms for uavs: a comparative study," *Unmanned systems*, vol. 6, no. 02, pp. 95–118, 2018.
- [3] M. E. Guerrero-Sánchez, R. Lozano, P. Castillo, O. Hernández-González, C. García-Beltrán, and G. Valencia-Palomo, "Nonlinear control strategies for a uav carrying a load with swing attenuation," *Applied Mathematical Modelling*, vol. 91, pp. 709–722, 2021.
- [4] V. M. Babu, K. Das, and S. Kumar, "Designing of self tuning pid controller for ar drone quadrotor," in *2017 18th international conference on advanced robotics (ICAR)*. IEEE, 2017, pp. 167–172.
- [5] I. Gonzalez-Hernandez, F. M. Palacios, S. S. Cruz, E. S. E. Quesada, and R. L. Leal, "Real-time altitude control for a quadrotor helicopter using a super-twisting controller based on high-order sliding mode observer," *International Journal of Advanced Robotic Systems*, vol. 14, no. 1, p. 1729881416687113, 2017.
- [6] X. Hu, B. Pang, F. Dai, and K. H. Low, "Risk assessment model for uav cost-effective path planning in urban environments," *IEEE Access*, vol. 8, pp. 150 162–150 173, 2020.
- [7] R. A. Krishnan, V. Jisha, and K. Gokulnath, "Path planning of an autonomous quadcopter based delivery system," in *2018 International Conference on Emerging Trends and Innovations In Engineering And Technological Research (ICETIETR)*. IEEE, 2018, pp. 1–5.
- [8] S. Bhandari and T. Srinivasan, "Path-planning around obstacles for a quadrotor uav using the rrt algorithm for indoor environments," in *AIAA Infotech@ Aerospace*, 2016, p. 2196.
- [9] M. Gutierrez-Martinez, E. Rojo-Rodriguez, L. Cabriaes-Ramirez, L. Reyes-Osorio, P. Castillo, and O. Garcia-Salazar, "Collision-free path planning based on a genetic algorithm for quadrotor uavs," in *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2020, pp. 948–957.
- [10] Z. Chen, F. Luo, and C. Zhai, "Obstacle avoidance strategy for quadrotor uav based on improved particle swarm optimization algorithm," in *2019 Chinese Control Conference (CCC)*. IEEE, 2019, pp. 8115–8120.
- [11] D. Falanga, K. Kleber, and D. Scaramuzza, "Dynamic obstacle avoidance for quadrotors with event cameras," *Science Robotics*, vol. 5, no. 40, 2020.
- [12] T. T. Mac, C. Copot, A. Hernandez, and R. De Keyser, "Improved potential field method for unknown obstacle avoidance using uav in indoor environment," in *2016 IEEE 14th International Symposium on Applied Machine Intelligence and Informatics (SAMII)*. IEEE, 2016, pp. 345–350.
- [13] R. A. Abdellatif and A. A. El-Badawy, "Artificial potential field for dynamic obstacle avoidance with mpc-based trajectory tracking for multiple quadrotors," in *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*. IEEE, 2020, pp. 497–502.

- [14] X. Dai, Y. Mao, T. Huang, N. Qin, D. Huang, and Y. Li, "Automatic obstacle avoidance of quadrotor uav via cnn-based learning," *Neuro-computing*, vol. 402, pp. 346–358, 2020.
- [15] B. Rubí, B. Morcego, and R. Pérez, "Quadrotor path following and reactive obstacle avoidance with deep reinforcement learning," *Journal of Intelligent & Robotic Systems*, vol. 103, no. 4, pp. 1–17, 2021.
- [16] B. Zheng, X. Guo, and J. Ou, "Policy-based monocular vision autonomous quadrotor obstacle avoidance method," in *Journal of Physics: Conference Series*, vol. 2083, no. 3. IOP Publishing, 2021, p. 032025.
- [17] R. F. Stengel, *Flight dynamics*. Princeton University Press, 2015.
- [18] R. Lozano, *Unmanned aerial vehicles: Embedded control*. John Wiley & Sons, 2013.
- [19] H. Ramirez-Rodriguez, V. Parra-Vega, A. Sanchez-Orta, and O. Garcia-Salazar, "Robust backstepping control based on integral sliding modes for tracking of quadrotors," *Journal of Intelligent & Robotic Systems*, vol. 73, no. 1, pp. 51–66, 2014.
- [20] A. Sanchez-Orta, V. Parra-Vega, C. Izaguirre-Espinosa, and O. Garcia, "Position–yaw tracking of quadrotors," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 6, p. 061011, 2015.
- [21] E. J. Ollervides-Vazquez, E. G. Rojo-Rodriguez, E. U. Rojo-Rodriguez, L. E. Cabriaes-Ramirez, and O. Garcia-Salazar, "Two-layer saturated pid controller for the trajectory tracking of a quadrotor uav," in *2020 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, 2020, pp. 85–91.
- [22] E. J. Ollervides-Vazquez, E. G. Rojo-Rodriguez, O. Garcia-Salazar, L. Amezcuita-Brooks, P. Castillo, and V. Santibanez, "A sectorial fuzzy consensus algorithm for the formation flight of multiple quadrotor unmanned aerial vehicles," *International Journal of Micro Air Vehicles*, vol. 12, p. 1756829320973579, 2020.