LiDAR-Based Navigation of Tethered Drone Formations in an Unknown Environment [*]

M. Bolognini and **L. Fagiano**

Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano, Piazza Leonardo da Vinci 32, Milano, Italy.
e-mail: michele.bolognini@polimi.it, lorenzo.fagiano@polimi.it

Abstract: The problem of navigating a formation of interconnected tethered drones, named STEM (System of TEthered Multicopters), in an unknown environment is considered. The tethers feed electrical power from a ground station to the drones and also serve as communication links. The presence of more than one interconnected drone provides enough degrees of freedom to navigate in a cluttered area. The leader drone in the formation must reach a given point of interest, while the followers must move accordingly, avoiding interference with the obstacles. The challenges are the uncertainty in the environment, with obstacles of unknown shape and position, the use of LiDAR (Light Detection And Ranging) sensors, providing only partial information of the surroundings of each drone, and the presence of the tethers, which must not impact with the obstacles and pose additional constraints to how the drones can move. To cope with these problems, a novel real-time planning algorithm based on numerical optimization is proposed: the reference position of each drone is chosen in a centralized way via a convex program, where the LiDAR scans are used to approximate the free space and the drones are moved towards suitably defined intermediate goals in order to eventually reach the point of interest. The approach is successfully tested in numerical simulations with a realistic model of the system.

Keywords: Autonomous vehicles, Robot navigation, Constrained control, Formation control, LiDAR sensors, Tethered drones.

1. INTRODUCTION

The drone market has been rapidly expanding in the last decade, and it is expected to keep growing in the coming years. Technological advancements and decreasing prices are driving the interest in the use of unmanned aerial vehicles (UAVs) for various purposes, such as mapping (Casella and Collin (2017)), emergency response (Nedjati and Vizvari (2016)) and building inspection (Rakha and Gorodetzky (2018)). Tethered drones are employed in applications where long operational time is needed and/or the system must be mechanically anchored for safety. Different aspects of tethered drones have been investigated (Tognon et al. (2016); Lee (2015); Lupashin and D'Andrea (2013); Oh et al. (2006); Choi et al. (2014)) and several commercial products exist as well (see, e.g., Elistair (2020)). Differently from the mentioned contributions, where a single tethered drone is considered, our research explores a novel tethered multi-drone system, called STEM (System of TEthered Multicopters), proposed in Fagiano (2017). STEM consists of a formation of two or more multicopter drones, tethered to each other and to a ground station. The tethers transfer electrical power from a ground power source to the drones, and also provide a power line commu-

nication network. Their length can be adjusted automatically by onboard winches, according to the formation's geometry. Each drone is stabilized by a local feedback controller, while a high level, centralized control algorithm computes the reference positions for all the units, according to the specific mission. The use of more than one tethered drone is justified by the gained flexibility: a chain of drones can reach places, e.g. behind obstacles, that a single tethered drone can not. A prototype STEM featuring two drones, each with 12 kg of mass and connected by 100-m-long cables, is currently under development at Politecnico di Milano.

In this paper, we study in particular the problem of autonomous navigation of STEM in a three-dimensional (3D) environment, with a-priori unknown obstacles. The goal is to reach a given point of interest with the first drone of the series. The other drones in the formation have to adapt their trajectories to allow the first one to reach its goal, while making all the units (and the tethers) avoid the obstacles. Since the latter are supposed to be a priori unknown, the centralized navigation algorithm can rely only on the partial information gathered by each unit, including, in addition to the usual position and attitude of each drone, the readings of planar LiDAR (Light Detection And Ranging) sensors installed on the UAVs.

This problem can be classified as a collision avoidance one: solution methods in the literature range from *planning* techniques, such as graph search (Hwangbo et al. (2007);
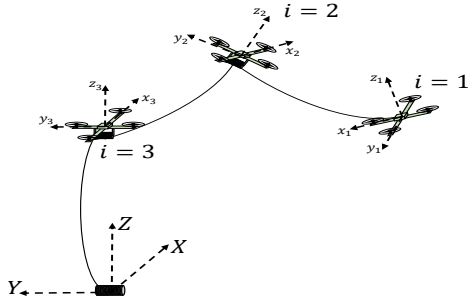
Fig. 1. Model of a considered STEM system with three drones.

Kitamura et al. (1996); Duchoň et al. (2014)), RRT (Li et al. (2016)), and potential fields (Khatib (1986)), to *reactive* techniques, e.g., Zapata and Lepinay (1999), Zufferey and Floreano (2006). However, to the best of the authors' knowledge, none of the existing approaches can deal with the combination of aspects in the problem at hand, i.e.: the presence of the tethers, the fact that the position and shape of the obstacles is assumed to be fully unknown a priori, and the use of LiDAR sensors, each one returning the distance of the closest obstacles along a finite number of directions.

The main contribution of this paper is a new reactive approach to deal with these aspects. A real-time algorithm collects the information from the drones and employs it to build suitable polytopic constraints. The latter approximate locally the set of all directions that do not impact with an obstacle, while also ensuring that any two subsequent drones remain in line-of-sight. The presence of tether catenary is accounted for by a constraint tightening procedure. Then, a quadratic program (QP) is solved to update the reference position tracked by each drone, in order to eventually reach the point of interest. To formulate the approach and test it in a realistic simulation environment, additional contributions are presented, including a model of the LiDAR sensors and a model of the tether as a nonlinear multi-body dynamical system.

2. SYSTEM MODEL AND PROBLEM FORMULATION

The navigation algorithm we propose in this paper operates in discrete time with a chosen sampling period T_s. At each sampling time, it collects information from the drones and computes and sends back reference position values. Since it does not employ past information nor it features an internal state, for notational simplicity we drop the explicit time-dependence of all the involved variables.

2.1 Multicopter drones

To simulate the system, we employ six-degrees-of-freedom, nonlinear continuous-time models of the drones, as described by Fagiano (2017). The considered setup features a ground station and $N \in \mathbb{N}$ drones in series, identified by indexes $i = 1, 2, ..., N$. The drone with $i = 1$, named leader, is the farthest one in the series from the ground station, see Fig. 1. The position and velocity vectors of the i-th drone in a fixed, right-handed inertial reference frame (X, Y, Z), with Z pointing up, are denoted with $\boldsymbol{p}_i, \dot{\boldsymbol{p}}_i \in \mathbb{R}^3$, where $\boldsymbol{p}_i = [p_{X,i}, p_{Y,i}, p_{Z,i}]^T$ and $\cdot^T$ is the

matrix transpose operation. The origin of the reference frame coincides with the position of the ground station. Electro-actuated winches are installed on the ground station and on each drone, except the leader, and can adjust the length of the tethers connecting any two subsequent units in the series. We denote with L_i the length of the reeled-out tether connecting drone i to drone $i + 1$ (and drone N to the ground station). The drones and the winches are assumed to be automatically controlled with local feedback loops that make the UAVs track a reference position vector, denoted with $\boldsymbol{p}_{ref,i}$, while automatically adjusting the tether lengths. A possible design approach for such local controllers is provided in Fagiano (2017).

2.2 LiDAR sensors

Each drone i is equipped with two planar LiDAR sensors, each one spanning 360 degrees in one of two perpendicular planes. We assume the LiDARs to be installed on active gimbals that stabilize their attitude notwithstanding the drone's motion. Referring to Fig. 1, one plane, named "horizontal", is always parallel to (X, Y) and elevated at the drone's Z−coordinate, while the other one, the "vertical plane", is perpendicular to (X, Y) and can be oriented by adjusting the gimbal's yaw angle. This allows each drone to detect obstacles lying inside one of such planes, while no information is gathered outside them. The angular resolution of the LiDARs, in degrees, is θ_s. For each drone i, a sensor scan thus produces a vector $\boldsymbol{s}_i \in \mathbb{R}^M$ with $M = \frac{360}{\theta_s}$ entries $s_i(j)$, each one corresponding to the distance of the closest obstacle along the direction $j\theta_s$ (with $j = 0$ corresponding to a selected, fixed direction) for example in the horizontal plane, $j = 0$ corresponds to the X−axis:

$$\begin{aligned} X_{i,j}^{obs} &= s_i(j) \cos(j\theta_s) \\ Y_{i,j}^{obs} &= s_i(j) \sin(j\theta_s); \quad j = 0, \dots, M-1, \end{aligned} \tag{1}$$

see Fig. 2 for a visualization. In simulation, vector $\boldsymbol{s}$ is computed via a collision detection routine between the obstacles, defined by a series of possibly overlapping polytopes, and M segments originating from the drones and directed along the probed directions. The LiDARs also have a maximum detection range s_{max}: if in a given direction $j\theta_s$ no obstacle is detected within this range, then $s_i(j) = \infty$.

The choice of LiDARs for navigation is advantageous because they are fast, hence it is possible to use them in real-time scenarios, they are active sensors, thus they work in conditions where light and contrast are scarce or excessive, and they retain accuracy at a distance.

Tethers connecting the drones are modelled as a series of springs and dampers, the reader is referred to an extended version of this paper (Bolognini and Fagiano (2020)) for the detailed model.

2.3 Problem Formulation

The main focus of this work is the supervisory controller highlighted in Fig. 3. Its inputs are the position vectors $\boldsymbol{p}_i, i = 1, \dots, N$, the tether lengths $L_i, i = 1, \dots, N$, and the LiDAR readings, $\boldsymbol{s}_i, i = 1, \dots, N$ sent by the drones, while the outputs are the reference position vectors $\boldsymbol{p}_{ref,i}, i = 1, \dots, N$ to be computed and sent to the
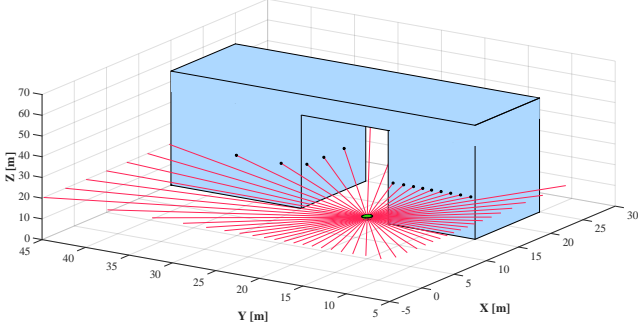
Fig. 2. Representation of a LiDAR scan along the horizontal plane in the simulation environment. The drone is represented as a green circle, while the light blue overarching structure represents an obstacle. Each line emanating from the drone is a laser beam. Similar scans are carried out along the vertical plane, at each time step.

drones in each sampling period. The problem we address is how to design such a supervisory controller to guide the leader drone to a point of interest with coordinates $\boldsymbol{p}_{poi}$ in the inertial reference (X, Y, Z), specified by an external agent, while moving the overall formation in order to avoid collisions between the drones and/or the tethers and the obstacles.

3. REAL-TIME NAVIGATION AND OBSTACLE-AVOIDANCE ALGORITHM FOR STEM

Let us collect the decision variables in a single vector $\boldsymbol{x} = [\boldsymbol{p}_{ref,1}^T, \ldots, \boldsymbol{p}_{ref,N}^T]^T \in \mathbb{R}^{3N}$, and let us define the matrix $\bar{\Lambda} = \mathrm{diag}(\Lambda_1, \ldots, \Lambda_N) \in \mathbb{R}^{3N \times 3N}$, where $\mathrm{diag}(\cdot)$ denotes a block-diagonal matrix and, for $i = 1, \ldots, N$, Λ_i are 3×3 diagonal matrices with the tuning parameters $\lambda_i \in (0, 1)$ on the diagonal. The proposed navigation algorithm consists of the following main steps:

(1) Collect the information $\boldsymbol{p}_i$, L_i, and $\boldsymbol{s}_i$, $i = 1, \ldots, N$;
(2) Using the collected information, formulate suitable *goals* for the drones, expressed as position vectors $\boldsymbol{p}_{goal,i} \in \mathbb{R}^3$ in the inertial frame. Collect the goals in vector $\boldsymbol{x}_{goal} = [\boldsymbol{p}_{goal,1}^T, \ldots, \boldsymbol{p}_{goal,N}^T]^T$;
(3) Also based on collected information, build a matrix $A \in \mathbb{R}^{p \times 3N}$ and vector $\boldsymbol{b} \in \mathbb{R}^p$ that define polytopic constraints on the drones' reference positions;
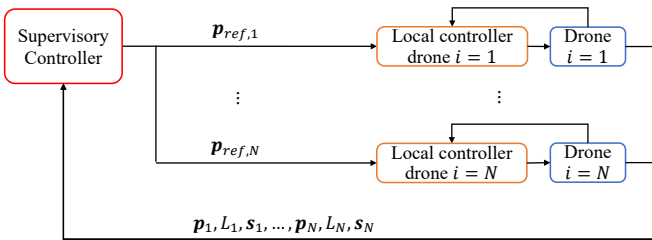(4) Compute $\boldsymbol{x}^*$ as:

Fig. 3. Layout of the considered control system. At each time step, the supervisory controller calculates position reference values for all drones, based on their current positions and LiDAR sensor readings.

$$\boldsymbol{x}^* = \arg \min_{\boldsymbol{x}} (\boldsymbol{x} - \boldsymbol{x}_{goal})^T \bar{\Lambda} (\boldsymbol{x} - \boldsymbol{x}_{goal})$$
$$\text{subject to} \quad (2)$$
$$A\boldsymbol{x} \leq \boldsymbol{b}$$

(5) Send the optimal position references contained in $\boldsymbol{x}^*$ to the corresponding drones, repeat from (1) at the next time step.

Note that (2) is a strictly convex QP of small size with Hessian $\bar{\Lambda}$, a diagonal matrix, which can be solved in very short time, see e.g. Wang and Boyd (2008); Bemporad (2016), and is thus suitable for real-time application. The weights γ_i are used to trade off, in the cost function, the distance of the different drones from their goals. Typically, a larger weight γ_1 is assigned to the leader's cost with respect to the other ones. The key points of the approach are the computation of the goals $\boldsymbol{p}_{goal,i}, i = 1, \ldots, N$ and of A and $\boldsymbol{b}$, described in the remainder of this section.

Regarding the goals, we initialize $\boldsymbol{p}_1^{goal} = \boldsymbol{p}_{poi}$ (i.e., the leader's goal is always equal to the wanted point of interest), and proceed sequentially, by setting the goal of drone $i + 1$ as:

$$\boldsymbol{p}_{goal,i+1} = \boldsymbol{p}_{goal,i} - \bar{d} \frac{\boldsymbol{p}_{goal,i} - \boldsymbol{p}_i}{\|\boldsymbol{p}_{goal,i} - \boldsymbol{p}_i\|}, \quad (3)$$

where $\bar{d}$ is a design parameter and $\boldsymbol{p}_i$ the current position of drone i. Equation (3) corresponds to the geometric construction presented in Fig. 4, whose aim is to make each drone $i + 1$ point a target "behind" drone i on the line connecting the latter to its goal, at a distance $\bar{d}$.
Regarding the constraints in (2), for simplicity we first consider a 2D case, and later extend the approach to 3D.
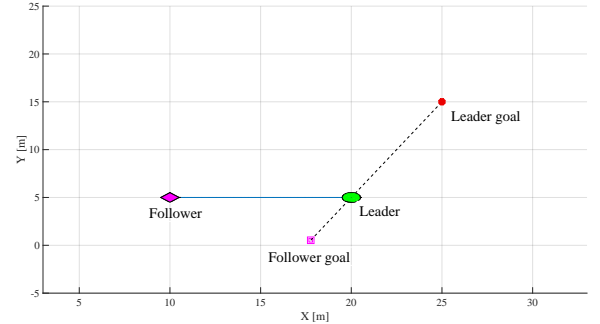

Fig. 4. Computation of goals on a plane parallel to (X, Y). The goal of drone $i + 1$ is computed as a point lying at a user-defined distance $\bar{d}$ (here 5 m) behind drone i on the line connecting $\boldsymbol{p}_{goal,i}$ and $\boldsymbol{p}_i$.

3.1 Computation of constraints in 2D

For the 2D case, let us assume that all drones have the same Z coordinate, so that their movements are bound to a plane parallel to (X, Y). We start by considering each drone i individually. Recall that the LiDAR readings $\boldsymbol{s}_i$ feature a finite entry if an obstacle is detected along the corresponding direction within the maximum distance, and infinite otherwise. It is then safe to state that if the drone moves along one of the directions with infinite distance in $\boldsymbol{s}_i$, no collisions can occur until the next sampling period. Moreover, if the scans are dense enough (*i.e.* if θ_s is small), then it is reasonable to assume that no

extremely thin obstacle lies between two consecutive scans, at least at a close distance. Figure 2 shows a graphical interpretation of this concept. Note that the maximum distance introduced in Section 2.2 can also be set to a value smaller than the LiDAR's limit, for example to navigate in narrow environments surrounded by obstacles, and that the intended sampling period (e.g., 0.1 s) is small compared to the drones' speed (e.g., 2 m/s). With these considerations in mind, let us define the following indexes:

$$\begin{aligned} \underline{j}_i &= \min_{j=1,\ldots,M} \quad j : s_i(j) = \infty \\ \bar{j}_i &= \max_{j=\underline{j}_i+1,\ldots,M} \quad j : s_i(k) = \infty, k = \underline{j}_i + 1, \ldots, j \end{aligned} \quad (4)$$

the pair $(\underline{j}_i, \bar{j}_i)$ thus identifies an angular sector spanning directions with no visible obstacles. The two lines that limit such a sector are given by:

$$\underbrace{\begin{bmatrix} -\tan(\underline{j}_i\theta_s) & 1 \\ -\tan(\bar{j}_i\theta_s) & 1 \end{bmatrix}}_{A_i\left(\underline{j}_i, \bar{j}_i\right)} \begin{bmatrix} p_X \\ p_Y \end{bmatrix} = \underbrace{\begin{bmatrix} -\tan(\underline{j}_i\theta_s)p_{X,i} + p_{Y,i} \\ -\tan(\bar{j}_i\theta_s)p_{X,i} + p_{Y,i} \end{bmatrix}}_{\boldsymbol{b}_i\left(\underline{j}_i, \bar{j}_i\right)} \quad (5)$$

Now, assuming that $\theta_s\left(\bar{j}_i - \underline{j}_i\right) \leq \pi$, the following inequality constraints (possibly with a change of sign to select the half-planes where no obstacles are present) identify the obstacle-free sector as a polyhedron:

$$A_i \begin{bmatrix} p_X \\ p_Y \end{bmatrix} \leq \boldsymbol{b}_i \quad (6)$$

Special cases are taken care of, such as when $\underline{j}_i\theta_s$ and/or $\bar{j}_i\theta_s$ are equal to $\frac{\pi}{2}$ or $\frac{3\pi}{2}$, when more non-contiguous, obstacle-free angular sectors are present, when the sector is non-convex (if $\theta_s\left(\bar{j}_i - \underline{j}_i\right) > \pi$) or when the point of interest lies between the leader drone and an obstacle. Further details are available in the extended version of the present paper, see Bolognini and Fagiano (2020).

Once A_i, $\boldsymbol{b}_i$, $i = 1, \ldots, N$ have been computed for all drones, we can build the following constraints on the reference position for each drone:

$$\bar{A}_i \begin{bmatrix} p_{X,ref,i} \\ p_{Y,ref,i} \end{bmatrix} \leq \bar{\boldsymbol{b}}_i \quad (7)$$

where $\bar{A}_N = A_i$, $\bar{\boldsymbol{b}}_N = \boldsymbol{b}_N$ and, for $i < N$:

$$\bar{A}_i = \text{diag}(A_i, A_{i+1}), \ \bar{\boldsymbol{b}}_i = \begin{bmatrix} \boldsymbol{b}_i \\ \boldsymbol{b}_{i+1} \end{bmatrix}. \quad (8)$$

Namely, for each drone we constrain the reference position to lie inside the intersection of the obstacle-free regions of the drone itself and of the next one in the series (i.e. its follower), in order to keep the two UAVs within line-of-sight of each other, thus ensuring that the $i-$th tether between them does not impact with any obstacle, see Fig. 5 for a visualization. This approach also favors recursive feasibility of the optimization problem at each time step, because drone $i + 1$ moves inside an obstacle-free area and drone i moves inside a subset of it. However, in some situations the presence of new, previously unseen obstacles might render constraints (7) unfeasible. In this case, one can either reduce the distance threshold above which a scanned direction is deemed obstacle-free, or consider a different obstacle free-angular sector (if more than one are present) for one of the two drones. As a matter of fact, in our tests recursive feasibility was always present. A formal
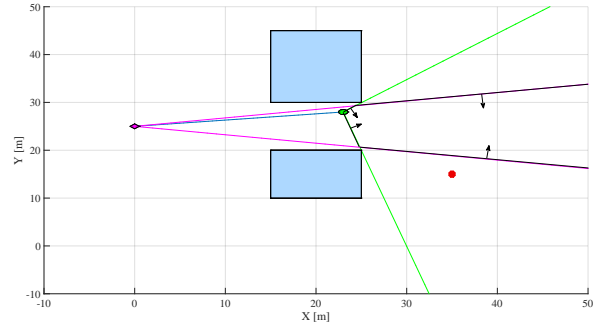

Fig. 5. The feasible space for the position reference of drone i (green circle) is limited by the one seen by drone $i+1$ (magenta square). If drone i were to travel within its own constraints only, the tether might impact with the obstacles.

proof of this property is subject of ongoing research.
The final step to build A and $\boldsymbol{b}$ in (2) is to consider constraints (7) altogether:

$$A = \text{diag}(\bar{A}_1, \ldots, \bar{A}_N), \ \boldsymbol{b} = \begin{bmatrix} \bar{\boldsymbol{b}}_1, \ \ldots, \bar{\boldsymbol{b}}_N \end{bmatrix}^T. \quad (9)$$

3.2 Computation of constraints in 3D

The main difficulty in extending the approach to the third dimension is the fact that each drone features two planar sensors, which is not equivalent to one 3D sensor. Thus, the points where obstacles are detected lie in the geometrical union of the two planes where the LiDAR scans take place, and it is not safe to make assumptions on the shape of the obstacles outside such region. For this reason, at each time step the algorithm sets the position reference for each drone in one of these planes and nowhere else. Therefore it must choose, for each drone, along which plane the movement will happen for the current sampling period. As mentioned in Section 2.2, we assume that the vertical plane where the LiDAR operates can be rotated via a gimbal. In particular, we assume that the leader carries out the vertical scan in the plane that contains the point of interest, while the other drones $i+1$, $i = 1, \ldots, N-1$, carry out the vertical scan in the plane that contains drone i. This is a sensible choice because, as a first approximation, if there is little wind and the speed of the drones is not high, the vertical plane containing drones i and $i + 1$ also contains the tether i with its catenary.

The constraints in the vertical plane are derived with the same approach adopted for the horizontal plane, i.e. equations (4)-(9). Let us denote with $\bar{A}_{H,i}$, $\bar{\boldsymbol{b}}_{H,i}$ the terms in (7) pertaining to the horizontal plane, and with $\bar{A}_{V,i}$, $\bar{\boldsymbol{b}}_{V,i}$ those related to the vertical one. Then, after the constraint sets in both planes have been computed, in order to choose whether to place the reference position of each drone in the horizontal or vertical plane, the following quantities are derived:

$$c_H = \begin{cases} 0 & \text{if } \bar{A}_{H,i} \begin{bmatrix} p_{X,goal,i} \\ p_{Y,goal,i} \end{bmatrix} \leq \bar{\boldsymbol{b}}_{H,i} \\ d\left(\begin{bmatrix} p_{X,goal,i} \\ p_{Y,goal,i} \end{bmatrix}, \mathcal{C}_H\right) & \text{otherwise} \end{cases}$$

$$c_V = \begin{cases} 0 & \text{if } \bar{A}_{V,i} \begin{bmatrix} p_{XY,goal,i} \\ p_{Z,goal,i} \end{bmatrix} \leq \bar{\boldsymbol{b}}_{V,i} \\ d\left(\begin{bmatrix} p_{XY,goal,i} \\ p_{Z,goal,i} \end{bmatrix}, \mathcal{C}_V \right) & \text{otherwise} \end{cases}$$

where $p_{XY,goal,i} = \left\| \begin{bmatrix} p_{X,goal,i} - p_{X,i} \\ p_{Y,goal,i} - p_{Y,i} \end{bmatrix} \right\|$ is the relative position on plane (X,Y) of the goal with respect to the drone (recall that the vertical plane always contains both points), $d(\boldsymbol{v}, \mathcal{A})$ is the distance between point $\boldsymbol{v}$ and set $\mathcal{A}$, finally $\mathcal{C}_H = \left\{ \begin{bmatrix} p_X \\ p_Y \end{bmatrix} : \bar{A}_{H,i} \begin{bmatrix} p_X \\ p_Y \end{bmatrix} \leq \bar{\boldsymbol{b}}_{H,i} \right\}$ and $\mathcal{C}_V = \left\{ \begin{bmatrix} p_{XY} \\ p_Z \end{bmatrix} : \bar{A}_{V,i} \begin{bmatrix} p_{XY} \\ p_Z \end{bmatrix} \leq \bar{\boldsymbol{b}}_{V,i} \right\}$. If $c_H \leq c_V$, the horizontal plane is chosen, otherwise the vertical one. Intuitively, this approach chooses the plane in which the distance between the corresponding constraint set and the point of interest (possibly projected on (X,Y) for the horizontal plane) is smaller. When the goal is behind an obstacle, this strategy effectively chooses to circumvent it by moving in the plane where the required travel length appears to be smaller. Once the plane has been chosen, the corresponding constraints are included in the QP (2), with the addition of suitable linear constraints to force the position reference to lie on that plane. The constraints pertaining to the discarded plane are instead dropped.

Another aspect that must be included in the 3D case is the presence of the tether catenary. To avoid impact between the tether and an obstacle in the vertical plane, the following linear constraints are also included, exploiting the map $\underline{Z}$, which describes the vertical distance between a drone and the lowest point of the cable connecting it to its follower, as a function of the position of the follower. The reader is again referred to Bolognini and Fagiano (2020) for details on how the map is obtained.

$$\begin{aligned} p_{Z,ref,i} &\geq (\overline{Z}_{obs} + \Delta Z) + (p_{Z,i} - \underline{Z}_i) \\ p_{Z,ref,i+1} &\geq (\overline{Z}_{obs} + \Delta Z) + (p_{Z,i+1} - \underline{Z}_i), \end{aligned} \tag{10}$$

where $\overline{Z}_{obs}$ is the highest point detected by the vertical LiDAR sensor of drone $i+1$, oriented towards drone i, i.e., the height of the highest detected obstacle below the cable connecting drone $i+1$ to i, and ΔZ is a tunable safety margin. Note that (10) are enforced on both drones i and $i+1$: this fact might impact feasibility if, for one or both of the drones, the horizontal plane had been chosen. In these cases, if the problem is infeasible, the vertical plane is chosen instead of the horizontal one, and the related constraints are included in the QP, so that both drones are allowed to move in Z direction. This is the only exception to the plane selection approach described above.

As in the 2D case, LiDARs readings are manipulated if the goal lies between the drone and an obstacle. This is even more important in 3D, since the ground is considered an obstacle and if the goal is lower than the leader's position it is difficult to descend without modifying the sensor output. Finally, we include in the optimization program additional linear constraints pertaining to the reference position of drone N, $\boldsymbol{p}_{ref,N}$, which prevent it from moving too far away in the (X,Y) plane from the ground station.
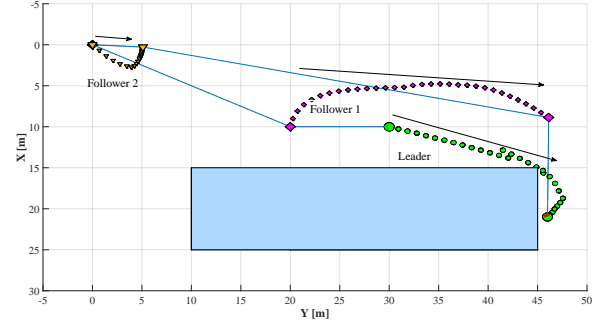

Fig. 6. In this simulation, the leader drone is temporarily prevented from traveling rapidly towards its goal, because its follower would then be behind a corner. Once the follower has moved past such corner, the leader can finally reach its destination.
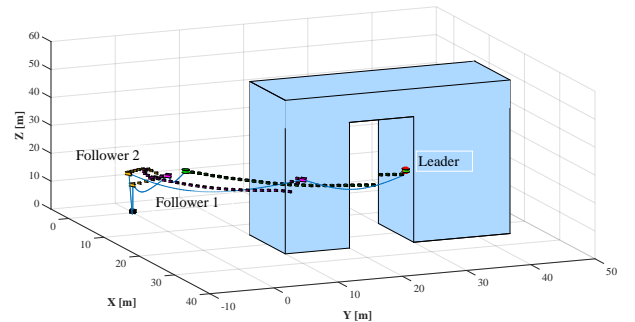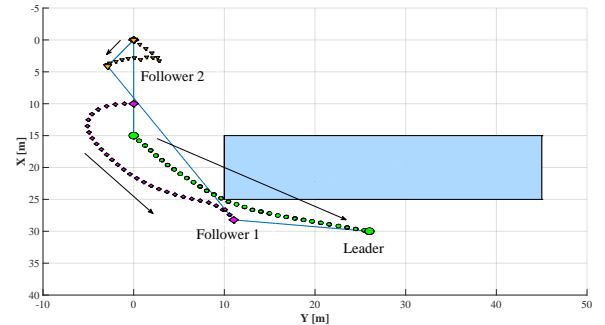

Fig. 7. A simulation of the whole system with three drones and a ground station (the drones move towards increasing x and y values). Note how the third drone (yellow triangle) is confined within a restricted area above the ground station.

4. SIMULATION RESULTS

All simulations were run with Matlab and Simulink. The latter was employed to simulate the continuous-time dynamics of the drones and the tethers and the discrete-time local controllers in between any two sampling instants of the supervisory controller, implemented in Matlab. The QP (2) has been solved with Matlab's `quadprog`. We carried out simulation tests with different shapes of obstacles and positions of the point of interest, covering many realistic application scenarios, and report here some of the obtained results. The simulations show that the proposed algorithm can manage a formation of three drones plus the ground station so as to guide the aircrafts in 3D space

avoiding the a-priori-unknown obstacles. Figure 6 highlights how the follower drone is guided around an obstacle and the leader is temporarily stopped from approaching the goal by the propagation of constraints introduced in section 3.1, in order to avoid a collision between the cable and the obstacle. The last movements of the leader also show that the condition where the goal lies between the drone and an obstacle is correctly recognized and taken care of, letting the drone approach the obstacle in order to get to its final destination. Figure 7 also shows that the leader can reach a goal initially behind an obstacle, furthermore it illustrates other aspects described in this paper:

- The third drone, directly connected to the ground station, hovers close to it in a confined region of space;
- The $i-$th drone tends to position itself on the line connecting drone $i-1$ to its goal;
- Each drone independently decides whether to move within the horizontal or vertical plane, depending on its goal and surroundings;
- The constraints on the horizontal plane are correctly propagated: in fact the second drone, towards the end of the simulation test, remains in a region of space where collision between obstacle and the second cable is avoided, letting its distance from the leader grow.

5. CONCLUSION

An algorithm able to navigate a formation of tethered drones in an unknown environment, relying on LiDAR measurements, was presented. The algorithm operates in real-time using current information only, and is based on the solution of a convex QP. The algorithm does not employ maps nor it stores information about the visited zones, rather it drives the drone formation to incrementally explore the environment towards the final goal, thus it is suitable for navigation in time-varying environments as well. In the development of this approach, safety was prioritized over performance, because the obstacle-avoidance feature is a primary concern for the automated flight of systems such as STEM. This work was aimed at enabling the autonomous flight in a simulation environment of a system of tethered drones, and it represents a first step for further research and development activities. These include: the inclusion of additional performance indicators such as speed, energy consumption and repeatability; the addition of mapping and trajectory generation functions; the addition of vision-based sensors such as cameras; the realization of a real-world prototype to experimentally test the different approaches in a building monitoring task.

REFERENCES

Bemporad, A. (2016). A quadratic programming algorithm based on nonnegative least squares with applications to embedded model predictive control. *IEEE Transactions on Automatic Control*, 61(4), 1111–1116.

Bolognini, M. and Fagiano, L. (2020). Lidar-based navigation of tethered drone formations in an unknown environment. *available at https://arxiv.org/abs/2003.12981*.

Casella, E. and Collin, A. (2017). Mapping coral reefs using consumer-grade drones and structure from motion photogrammetry techniques. *Coral Reefs*, (36), 269–275.

Choi, S.Y., Choi, B.H., Jeong, S.Y., Gu, B.W., Yoo, S.J., and Rim, C.T. (2014). Tethered aerial robots using contactless power systems for extended mission time and range. In *IEEE Energy Conversion Congress and Exposition*, 912–916.

Duchoň, F., Babinec, A., and Kajan, M. (2014). Path planning with modified a* algorithm for a mobile robot. *Modeling of Mechanical and Mechatronic Systems*, 96, 59–69.

Elistair (2020). Company website: http://elistair.com/, last accessed september 2016.

Fagiano, L. (2017). Systems of tethered multicopters: Modeling and control design. *IFAC - PapersOnLine*, 50(1), 4610–4615.

Hwangbo, M., Kuffner, J., and Kanade, T. (2007). Efficient two-phase 3d motion planning for small fixed-wing uavs. 1035–1041. Rome, Italy.

Khatib, O. (1986). Real time obstacle avoidance for manipulators and mobile robots. pp. 500–505. St. Louis, MO, USA.

Kitamura, Y., Kishino, F., Tanaka, T., and Yachida, M. (1996). Real-time path planning in a dynamic 3-D environment. 2, 925–931. Osaka, Japan.

Lee, T. (2015). Geometric controls for a tethered quadrotor UAV. In *IEEE Conference on Decision and Control*, 2749–2754. Osaka, Japan.

Li, Y., Littlefield, Z., and Bekris, K. (2016). Asymptotically optimal sampling-based kinodynamic planning. *The International Journal of Robotics Research*, 35(5), 528–564.

Lupashin, S. and D'Andrea, R. (2013). Stabilization of a flying vehicle on a taut tether using inertial sensing. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2432–2438. Tokyo, Japan.

Nedjati, A. and Vizvari, B. (2016). Post-earthquake response by small uav helicopters. *Natural Hazards*, (80), 1669–1688.

Oh, S., Pathak, K., Agrawal, S.K., Pota, H.R., and Garrett, M. (2006). Approaches for a tether-guided landing of an autonomous helicopter. *IEEE Transactions on Robotics*, (22), 536–544.

Rakha, T. and Gorodetzky, A. (2018). Review of unmanned aerial system (uas) applications in the built environment: Towards automated building inspection procedures using drones. *Automation in Construction*, (93), 252–264.

Tognon, M., Dash, S.S., and Franchi, A. (2016). Observer-based control of position and tension for an aerial robot tethered to a moving platform. *IEEE Robotics and Automation Letters*, 732–737.

Wang, Y. and Boyd, S. (2008). Fast model predictive control using online optimization. In 17th *IFAC World Congress*, 6974–6979. Seoul, Korea.

Zapata, R. and Lepinay, P. (1999). Flying among obstacles. *Proceedings of the Third European Workshop on Advanced Mobile Robots*, 81–88.

Zufferey, J.C. and Floreano, D. (2006). Fly-Inspired Visual Steering of an Ultralight Indoor Aircraft. *IEEE Transactions on Robotics*, 22(1), 137–146.