



HAL
open science

Image Based Visual Servoing for Multi Aerial Robots Formation

Mark Bastourous, Jaafar Al-Tuwayyij, Francois Guérin, Frédéric Guinand

► **To cite this version:**

Mark Bastourous, Jaafar Al-Tuwayyij, Francois Guérin, Frédéric Guinand. Image Based Visual Servoing for Multi Aerial Robots Formation. 28th Mediterranean Conference on Control and Automation (MED 2020), Sep 2020, Saint-Raphaël, France. pp.115-120, 10.1109/MED48518.2020.9182942 . hal-03352607

HAL Id: hal-03352607

<https://hal.archives-ouvertes.fr/hal-03352607>

Submitted on 23 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Image Based Visual Servoing for Multi Aerial Robots Formation

Mark Bastourous¹, Jaafar Al-Tuwayyij, Francois Guérin², Frédéric Guinand^{1,3}

Abstract— There are numerous advantages of flying in group over using single robot in mission execution. However this implies solving a crucial issue: the coordination between drones. Moreover, according to the targeted application, it may be necessary or desirable that drones fly following a given geometric shape (line, diamond, etc.), a problem known as formation control. Building and maintaining a spatial geometric shape while evolving within the environment usually requires extensive communications between the robots for coordinating their movements. In this work we focus on the use of an Image-Based Visual Servoing (IBVS) technique for building and maintaining a Leader-Follower (LF) configuration of multi aerial vehicles (UAVs) without communication. While most IBVS techniques either require rigor camera calibration or can not regulate the error according to the three robot axes, our approach avoids the calibration phase by relying on image moments features to provide a vision-based predictive compensation method. The follower robot's solution works in GNSS-denied conditions and can run using only on-board sensors. The method is validated through simulations for a group of three quadrotors.

I. INTRODUCTION

Multi-Robot Systems (MRS) have been widely studied since the 90s [1], [2]. Many advantages can be expected from using a group of robots: increase and share of the payload [3], reduction of the time needed for the achievement of a task [4], fault tolerance and resilience of the system [5], [6], or use of simpler and cheaper robots for adaptability to the environment [7]. However, making a group of robots evolving together in the same environment entails the resolution of some problems. Coordination is one of the most critical ones, especially considering decentralized settings. Coordination between the members of the group can be achieved in various ways, from behavior-based or forces-based methods to Virtual Rigid Body (VRB) structure building, through Leader-Follower (LF) schemes.

In the present work we focus on the last approach. We consider a set of UAVs equipped with ubiquitously available sensors for the flight controller, a perspective camera, but, no possibility to communicate with each others. In addition to the absence of communications, the leader's velocity is considered unknown and no beforehand camera calibration is done.

While the question of building and maintaining LF formation control without communication has been addressed many times during the last two decades, the problem settings considered in our work are more restrictive than most of the known literature. Most of the time, for compensating the absence of communications, the proposed solutions rely on a sensor-based mechanism. Pan-controlled camera [8] or omnidirectional camera like in [9], [10], [11], or other types of sensors like the kinect [11] are often considered. Note first that the camera provides each follower the bearing angle with respect to the leader. In 2010 Fabio Mobidi and his colleagues show that an estimation of the distance between the follower and the leader, a.k.a. range estimation, can be obtained from an omnidirectional camera [9]. However, the proposed method requires communication between the leader and the followers. More recent works avoiding communications and considering cameras were proposed. From the data obtained by the sensors, each follower of the formation attempts to estimate the pose of the global or of its local leader. Unlike in our work, the design of the controller is often based on this estimation for deriving both the linear and angular speed for maintaining the configuration.

In 2015, Chen and Jia have presented an adaptive controller for a LF configuration without relying on communications [8]. An active vision sensor was used to track the leader, and they proposed both a controller for the active vision actuation and one for the tracking. However, accurate calibration of the camera was required for the distance measurement for leader's velocity estimation. This calibration is also a critical point for other methods relying on mutual pose estimation or relative pose estimation [12]. This makes a clear difference with our approach, which does not require camera calibration, that needs only four non co-linear points in the image plane to determine, in the follower's camera frame, the desired state of the leader.

More recently, in 2017, Guo and his colleagues proposed a method for LF formation control considering a robotic system in which communications are not allowed [10]. Leader's velocity is considered unknown and each follower is equipped with an uncalibrated omnidirectional or perspective camera, thus settings very similar to our current work. However, the described adaptive estimator, based on several feature points, failed when the movement of the leader is colinear to the follower's camera axis, a situation for which our method offers a solution as explained in Section II. The last notable work under this set of constraints is due to Liu and his colleagues who consider robots equipped with kinects [11]. This sensor allows an estimation of the pose (relative

*This work is funded by LHSM, and by "SolarFarm", project #280390, part of the BIONAER research program.

¹Normandie Univ. UNIHAVRE LITIS lab, Le Havre, France

²Normandie Univ. UNIHAVRE GREAH lab, Le Havre, France

³UKSW Faculty of Mathematics and Natural Sciences. Warsaw, Poland

orientation and position of the leader) and provides directly both angles and distances measurements but at the price of a very limited field of view, incompatible with many LF configurations.

In that context, our main contribution is a vision-based predictive compensation method for building and maintaining a Leader-Follower configuration under a set of restrictive constraints: no communications between the UAVs, no prior calibration of the camera and considering the leader's velocity as unknown. The method is based on the use of image moments as features for Image-Based Visual Servoing (IBVS), and unlike most of the existing works our approach does not use explicit distance measurement or camera calibration, thus can also be applied in a GNSS-denied environment.

The remaining of the paper is organized as follows. Section II presents the problem and the proposed resolution method. Validation of the approach through simulations, considering three aerial robots with a six Degrees of Freedom (DoF), is described in Section III. Concluding remarks followed by a discussion about going from simulation to real-world experiments are presented in the last Section.

II. PROBLEM STATEMENT

A. Problem description

The main task to be achieved by navigating a group of robots can be described as the ability to construct a geometrical shape sometimes known as Virtual Rigid Body (VRB). The virtual structure between several robots aims at facilitating the agile control of several multi-rotor formations. The goal of the follower's robot is to follow the target, the leader, by keeping a predefined separation distance d and bearing angle θ , as illustrated by Figure 1. Additionally we control the follower UAV without directly measuring distance and utilising the image plane of the camera that is mounted on the robot. The UAV leader is assumed to track perfectly a virtual target which moves along a defined trajectory (straight line, circle, etc) or with a human in the loop motion planning (description of leader's controller is out of the scope of this paper). Unlike other VRB methods, which rely on the group's barycenter to steer the formation [13], in our case each robot plans its own trajectory contributing to a distributed formation control, such that no positioning and tracking from a centralized point is required.

We adopt a multi-layer control scheme composed of low level control, which provides the control signals of the UAV, which is already implemented in the flight controller.

In the context of unavailable communication in between the robots, the formation control should surmise the value of leader's velocity by one way or another. In [14], we designed a high level controller which only requires simple tunings and rests on a predictive filtering algorithm and a first order dynamic model to recover an estimation of the leader's velocities and avoid the tracking errors. The proposition of this current paper is to develop and test a system that does

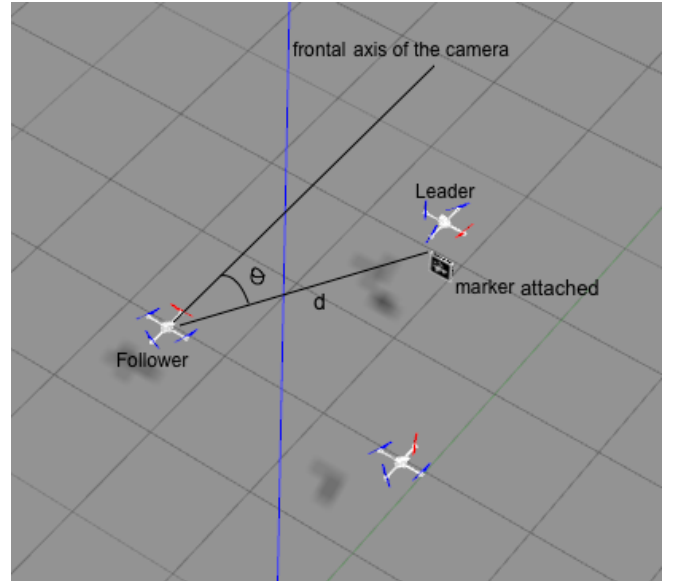


Fig. 1: Bird-eye view of the formation to create a triangle or V-shape with the leader in front carrying a fiducial marker. The follower is having a perspective camera viewing the leader and its marker. The camera image view is shown in figure 2.

not directly measure the distance to leader and at the same time preserves the virtual structure as concise as possible. For this constraint an Image-Based Visual Servoing (IBVS) has been used.

B. Approach description

Visual servoing is a technique that uses the visual information provided by the camera embedded on the robot, in a closed control loop. Early work of visual servoing started to emerge in 1980's. It became more mature then with the results of the work of adaptive visual control of a robot arm [15], then the work of [16], [17]. It can be categorized with Image Based Visual Servoing (IBVS), Position Based Visual Servoing (PBVS), and hybrid techniques. In our case eye-in-hand IBVS was implemented for some important reasons such as: monocular cameras available onboard of most aerial robots without calibration, directly control in image plane, special features representation that will require distance measurement.

First we have to define the features that are carefully selected for IBVS on underactuated system.

The features used for the algorithm are based on the popular image moments. They are represented starting from defining the object in the field of view as n points, each point $p_k = (x_k, y_k)$ represents the coordinates in the image plane. The moments m_{ij} and the centered moments μ_{ij} of this object are defined as:

$$m_{ij} = \sum_{k=1}^n x_k^i y_k^j \quad (1)$$

$$\mu_{ij} = \sum_{k=1}^n (x_k - x_g)^i (y_k - y_g)^j \quad (2)$$

Where $x_g = m_{10}/n$, $y_g = m_{01}/n$ and $n = m_{00}$. Since the considered shape is composed of a discrete set of points, then its area can be expressed as:

$$a = \mu_{12} + \mu_{21}, \quad a^* = \mu_{12}^* + \mu_{21}^* \quad (3)$$

Where a and a^* are the current and the desired areas of the shape. To choose visual features to control the translational Degrees Of Freedom (DOF), we can make use of the image moment point based features proposed in [18] and validated with more results in [19].

This major pitfall in the work of [10], as mentioned in the introduction, is the target's motion in the direction of the frontal axis of the center of the camera (z-axis), this can imply several singularities in their tracking.

For avoiding such a situation, normalized versions of the features are considered, and are defined such that:

$$a_n = Z^* \sqrt{\frac{a^*}{a}}, \quad x_n = a_n x_g, \quad y_n = a_n y_g \quad (4)$$

Where a^* and Z^* are the area and the depth of the target at the desired pose. And x_n, y_n are the normalized centroid of the shape.

The error vector (in the image plan) is:

$$e = s - s^* \quad (5)$$

where $s = (x_n, y_n, a_n)^T$ and $s^* = (x_n^*, y_n^*, a_n^*)^T$ be the current and the desired features' states respectively.

The relation between the image dynamics and the velocity of the camera is usually defined as:

$$\dot{e} = L_s V_c \quad (6)$$

Since the velocities according to x,y, and z axes are the ones to be controlled, the interaction/Jacobian matrix becomes as follows:

$$L_s = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad (7)$$

Normally it is designed to control 6 DOF robot arm, but since the multi-rotor UAVs are considered underactuated systems, there are 2 DOF that can not be directly controlled. For a multi-rotor UAV, hover condition means the pitch or roll angle with respect to the world plane should be zeros. This will lead to omitting the forth and fifth columns and rows of the stack of the original interaction matrix leading to (7). Taking into account the fact that a kinematic model was considered. Three inputs delivered to the low level velocity controller such that $V_c = (\nu_x, \nu_y, \nu_z)^T$.

To make sure that the error is decreasing exponentially the well known error dynamics equation is defined as:

$$\dot{e} = -\lambda e \quad (8)$$

where λ is a positive control gain. Substituting in (6) and solving for V_c we get the velocity vector of the camera from the dynamics of the image:

$$V_c = -\lambda L_s^+ e \quad (9)$$

Since this camera is mounted on a UAV (in our case quadrotor), this velocity twist vector should be translated and rotated with respect to the mounting details to the center of mass of the robot. The used kinematic model of a quadrotor was taken into consideration unlike most of the other IBVS implementations for quadrotor that can be found in literature. That gives more freedom for the controller to be interchangeable to other types of quadrotors or even different multi-rotor vehicles without the need to redesign the gains, but rather fine tune them. Equation (6) is valid for static target. An additional aspect has to be included to that error equation compensating for the motion of the leader (10).

$$\dot{e} = \dot{s} - \frac{\partial e}{\partial t} \quad (10)$$

$\partial e / \partial t$ is considered to be the leader's velocity in image plane of the follower robot.

The full dynamics equation is given by substituting \dot{s} by it's motion equation leads to:

$$\dot{e} = L_s V_c - \frac{\partial e}{\partial t} \quad (11)$$

Then finally the velocity vector can be represented as:

$$V_c = \widehat{L_s^+} (-\lambda e - \frac{\partial e}{\partial t}) \quad (12)$$

Let us mention that the gains in the previous controller are adaptive, meaning that the gain value changes depending on the the error value:

$$\lambda = (\lambda_{\max} - \lambda_{\min}) \left(\frac{|e_t|}{|e_{\max}|} \right) + \lambda_{\min} \quad (13)$$

Where λ_{\max} , λ_{\min} are the upper and lower limits of the gain respectively, $|e_t|$ is the norm of the error vector at time t and $|e_{\max}|$ is the norm of the maximum error value at the first iteration of the control loop. According to [20], where authors show the effect and smoothness of adaptive gains on different visual servoing scenarios, which was adopted by default in our developed framework.

The velocity vector V_c is then passed to the low level controller of the flight controller (e.g., PX4 on Pixhawk) mounted on the UAV controlling its specific dynamics. Moreover to guarantee that the robot's velocity does not exceed the physical limit (in some cases security limit), a saturation function was introduced. This function will smooth and bound the velocities.

The $\widehat{\partial e/\partial t}$ is an estimation of the value of $\partial e/\partial t$. This term can be calculated in different methods dependant on the application. For instance if the target moves at a constant velocity, this value can be calculated as:

$$\widehat{\frac{\partial e}{\partial t}} = \gamma \sum_i e_i \quad (14)$$

Where γ is a positive gain. It should be adaptive as well as predictive to change online depending on the leader's velocity. A simplified version of that gain can be constant by tuning in the case where leader's velocities are constant, which is not our case.

Since there is no communication in between the robots, the follower robots should predict leader's velocity from the direct observation of the image moments. Building on the assumption that the robots are equipped with Attitude and Heading Reference System (AHRS) sensors, the robot can have good estimate of its own velocity in 3D space. This is an actual assumption that should hold for the robot to be able to stabilize itself using its low level flight controller. To get the prediction of the leader's velocity from the direct observation of the image moments, a common nonlinear Extended Kalman Filter (EKF) is implemented.

The main components of the EKF can be defined as follows:

Fundamental matrix \mathbf{F} can be defined as a nonlinear function $f(\mathbf{x}, \mathbf{u})$, and the linear expression $\mathbf{H}\mathbf{x}$ is replaced by a nonlinear function $h(\mathbf{x})$:

$$\begin{aligned} \dot{\mathbf{x}} &= f(\mathbf{x}, \mathbf{u}) + n_x \\ \mathbf{z} &= h(\mathbf{x}) + n_z \end{aligned}$$

where n_x and n_z are the noises.

\mathbf{x} is the state vector. It can be defined as the velocities according to x,y,z axes and their corresponding angular velocities w_x, w_y, w_z .

The main equation of the filter can be written as:

$$\bar{\mathbf{P}} = \mathbf{F}\mathbf{P}\mathbf{F}^T + \mathbf{Q}$$

\mathbf{F} is considered the state transition matrix, with values chosen imperically and can be defined as:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & 0 & 0.005 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.005 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0.005 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Then the Kalman gain matrix can be defined as:

$$\mathbf{K} = \bar{\mathbf{P}}\mathbf{H}^T(\mathbf{H}\bar{\mathbf{P}}\mathbf{H}^T + \mathbf{R})^{-1}$$

where \mathbf{R} is the measurement noise covariance matrix. It is considered the prior and defined as identity matrix of 6x6:

$$\mathbf{P} = (\mathbf{I} - \mathbf{K}\mathbf{H})\bar{\mathbf{P}}$$

Process and measurement uncertainty/noise matrix can be 6x6 identity matrix.

Measurement model which is presented in the matrix (\mathbf{H}) can be represented by two horizontally stacked 3x3 matrices: First matrix is the follower's robot self velocity measurement coming from a feedback sensor as optical flow or other techniques. The second matrix is represented as 3x3 zeros matrix, since yaw control is not considered in this paper. This will build the standard predict/update loop which will iterate to provide $\widehat{\frac{\partial e}{\partial t}}$.

Relating back to [14], we designed a controller able to compensate the communication loss by using a first order dynamic model and the double exponential smoothing algorithm.

III. SIMULATION AND RESULTS

A. Preliminaries

- Three UAVs are considered, one leader and two followers.
- The aggregation phase between the robots is not considered.
- UAVs are equipped with good low level controllers, e.x. open source PX4 stack. The control inputs of our concern can be summed up in vertical, lateral, linear and angular velocities with respect to its own reference frame.
- Each robot is equipped with embedded monocular perspective camera which can provide leader robot detection through tracking a fiducial marker like Apriltag¹, or ARtag.
- The algorithm should be initialized by posing the leader in the desired pose with regard to the follower. Then after capturing the image, the desired image moment will be deduced from it. That method can be considered as teaching by showing.

B. Simulation Design

In this section, we present the simulation results for both the Gazebo/ROS simulator. The model of the UAV is available to the scientific community as a ROS package [21], namely the AscTec Hummingbird model was used. We have mounted a perspective camera on the follower robots and fiducial marker as a tag on the leader to facilitate its detection. It has to be noted that, the algorithm is marker agnostic. It is only required to have 4 discrete points, that represent the shape that should be tracked (in our case for simplicity of experiments it was the corners of the marker). The desired image moments with respect to the detected one as seen from the follower UAV can be viewed in figure 2. The desired moments are in green and the detected ones are in red.

¹AprilTag 2: Efficient and robust fiducial detection, Wang et al, IROS 2016, <https://github.com/AprilRobotics/apriltag>

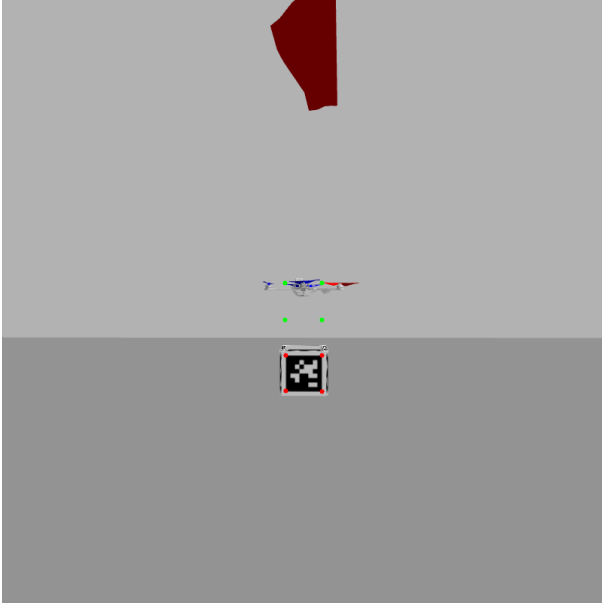


Fig. 2: Camera view of the follower UAV observing the leader UAV. Image moments visualization is emphasised. The desired features location is in green, and the detected features are in red which are the corners of the marker.

Gazebo/ROS Simulation: The values of the simulation used, are:

- Simulation time: 50s starting from 280s till 330s in figure 4, and from 420s till 470s in figure 3. The results were recorded to a rosbag file. Then the plots were generated continuously, that is the reason behind different time stamps which can be normalised to start from 0s and end at 50s.
- Camera mounted distance from the center of UAV according to z-axis = -0.15m according to x-axis = 0.15m
- Altitude: 7m
- Maximum allowed velocity: 0.5 m/s

The initial conditions of the quadrotors are:

- Take off positions (x,y,z) in meter: leader : 0,3,0; follower 1 : 0,1,0 ; follower 2 : 0,-1,0
- Relative positions (x-spacing,y-spacing, z-spacing) between the different followers and the leader are represented in meters as follows:
follower 1 at 2,1,0 ; follower 2 at 2,-1,0

which is done by posing the leader in the desired spacing then capturing the image of the follower's camera. Then image moments will be extracted and considered as features.

C. Simulation Results

Results obtained from the Gazebo simulation are reported in this section with two main analysis. The error according to the X,Y,Z axes of the two follower UAVs are shown in figures (3,4).

The distances being maintained at around the desired value, show consistent followers tracking which achieve consensus of the same desired commanded velocities eventually after their take off throughout the whole trajectory up till the landing phase.

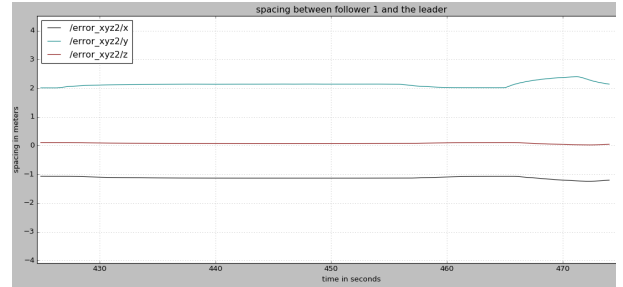


Fig. 3: Distances spacings according to the desired (X, Y, Z) between robot 2 "considered as follower 1" and the leader. The final phase starting at 465s is for landing

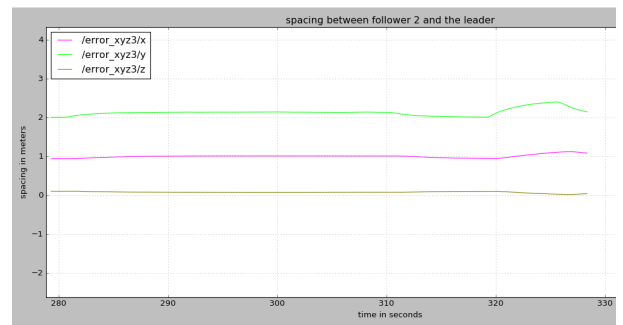


Fig. 4: Distances spacings according to the desired (X, Y, Z) between robot 3 "considered as follower 2" and the leader. The final phase starting at 465s is for landing.

In the figure 5, to validate the algorithm and analyse its effectiveness, a logging of the absolute spacing distance ($\sqrt{x^2 + y^2 + z^2}$) between the leader and a follower UAV in the world frame is being studied. N.P: This distance spacing is not used by the controller, but only for analysis. The experiment starts with leader velocity of -0.3 m/s in the absolute y direction of the world. The follower will maintain a spacing distance of 1.75m. Then the leader will be commanded to stop, so a velocity of zero was commanded at around 142s. The follower UAV as shown will immediately react trying to achieve the same distance spacing. The prediction phase oscillates and then achieve the proper spacing. To tackle realistic scenarios, and test the robustness of the predictor. Even before it settle down to the appropriate spacing, the leader was commanded velocity of 0.3 m/s at around 150s. As shown the algorithm was able after some minor oscillation to achieve the desired spacing. Then at around 165s the leader was commanded with -0.3 m/s in the absolute Y of the world which according to 1 is in the same direction as the forward looking axis of the camera. Again and lastly the system was able to recover such variation in velocity and direction (from forward 0.3 m/s to

backward 0.3 m/s of the leader). As observed the oscillation is more apparent due to the toughness of such scenario. The follower robot has no knowledge of that velocity as there is no communication. As shown in figure 5, the distance is always around the desired 1.75m spacing. There exist oscillations at the time of changing the velocity of the leader

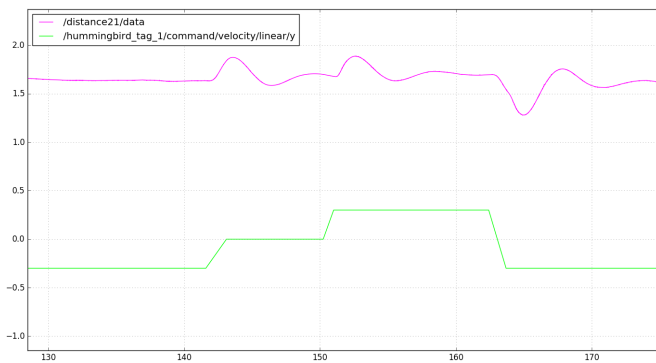


Fig. 5: Absolute distance spacing between robot 2 "considered as follower 1" and the leader is analysed over time. green : leader reference velocity - magenta: separating distance leader/follower

IV. CONCLUSION AND PERSPECTIVES

In this work we focused on the problem of building a model for formation of UAVs and maintaining this formation under the constraint of using only on-board local sensing measurements to detect image moments features. The use of such features, omits the need for explicit distance measurement, which can not naturally be done by monocular cameras without sophisticated algorithms. No infrastructure is required for the controller to be operational, such as GNSS outdoor or an indoor motion capture system. The proposed solution is based on a Leader-Follower scheme.

The reported results, obtained from the simulations of Gazebo are written in Python and C++ under ROS. They show that the formation is maintained during the navigation mission. The future perspectives are to perform real experiments in our laboratory in order to corroborate these simulated results. The hardware setup is designed based on DJI F450 quadrotor airframe equipped with PX4 low level controller stack on Pixhawk cube and perspective camera. The future work can tackle the problem of yaw control from tracking the image moments of the leader's robot.

REFERENCES

- [1] T. Arai, E. Pagello, and L. E. Parker, "Editorial: Advances in multi-robot systems," *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 655–661, 2002.
- [2] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng, "Cooperative mobile robotics: Antecedents and directions," *Autonomous Robots*, vol. 4, pp. 7–27, 1997.
- [3] D. Mellinger, M. Shomin, N. Michael, and V. Kumar, *Cooperative Grasping and Transport Using Multiple Quadrotors*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 545–558.

- [4] A. Khamis, A. Hussein, and A. Elmoogy, *Multi-robot Task Allocation: A Review of the State-of-the-Art*. Springer International Publishing, 2015, pp. 31–51.
- [5] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: opportunities and challenges," *IEEE Communications Magazine*, vol. 54, no. 5, pp. 36–42, 2016.
- [6] G. Tuna, B. Nefzi, and G. Conte, "Unmanned aerial vehicle-aided communications system for disaster recovery," *Journal of Network and Computer Applications*, vol. 41, pp. 27–36, 2014.
- [7] T. Tosun, J. Daudelin, G. Jing, H. Kress-Gazit, M. Campbell, and M. Yim, "Perception-informed autonomous environment augmentation with modular robots," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2018. [Online]. Available: <https://doi.org/10.1109/icra.2018.8463155>
- [8] X. Chen and Y. Jia, "Adaptive leader-follower formation control of non-holonomic mobile robots using active vision," *IET Control Theory & Applications*, vol. 9, no. 8, pp. 1302–1311, May 2015. [Online]. Available: <https://doi.org/10.1049/iet-cta.2014.0019>
- [9] F. Morbidi, G. L. Mariottini, and D. Prattichizzo, "Observer design via immersion and invariance for vision-based leader-follower formation control," *Automatica*, vol. 46, no. 1, pp. 148–154, Jan. 2010. [Online]. Available: <https://doi.org/10.1016/j.automatica.2009.10.016>
- [10] D. Guo, H. Wang, W. Chen, M. Liu, Z. Xia, and K. K. Leang, "A unified leader-follower scheme for mobile robots with uncalibrated on-board camera," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2017. [Online]. Available: <https://doi.org/10.1109/icra.2017.7989438>
- [11] X. Liu, S. S. Ge, and C.-H. Goh, "Vision-based leader-follower formation control of multiagents with visibility constraints," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 3, pp. 1326–1333, May 2019. [Online]. Available: <https://doi.org/10.1109/tcst.2018.2790966>
- [12] L. Teixeira, F. Maffra, M. Moos, and M. Chli, "Vi-rpe: Visual-inertial relative pose estimation for aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2770–2777, Oct 2018.
- [13] A. S. Brandão and M. Sarcinelli-Filho, "On the guidance of multiple uav using a centralized formation control scheme and delaunay triangulation," *Journal of Intelligent & Robotic Systems*, vol. 84, no. 1-4, pp. 397–413, 2016.
- [14] M. Bastourous, F. Guérin, F. Guinand, and E. Lemains, "Decentralized high level controller for formation flight control of uavs," in *2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, 2020, pp. 226–231.
- [15] A. C. Sanderson and L. E. Weiss, "Adaptive visual servo control of robots," in *Robot vision*. Springer, 1983, pp. 107–116.
- [16] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control," *IEEE transactions on robotics and automation*, vol. 12, no. 5, pp. 651–670, 1996.
- [17] F. Chaumette, "Visual servoing," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Ed. Springer, 2014, pp. 869–874.
- [18] O. Tahri and F. Chaumette, "Point-based and region-based image moments for visual servoing of planar objects," *IEEE Transactions on Robotics*, vol. 21, no. 6, pp. 1116–1127, 2005.
- [19] P. A. Karasev, M. M. Serrano, P. A. Vela, and A. Tannenbaum, "Depth invariant visual servoing," in *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE, 2011, pp. 4992–4998.
- [20] O. Kermorgant and F. Chaumette, "Dealing with constraints in sensor-based robot control," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 244–257, 2013.
- [21] F. Furrer, M. Burri, M. Achtelik, and R. Siegwart, *Robot Operating System (ROS): The Complete Reference (Volume 1)*. Cham: Springer International Publishing, 2016, ch. RotorS—A Modular Gazebo MAV Simulator Framework, pp. 595–625. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-26054-9_23