A distributed vision-based consensus model for aerial-robotic teams

Fabio Poiesi and Andrea Cavallaro

***Abstract*— We present a distributed model for a team of autonomous aerial robots to collaboratively track a target without external control. The model uses distributed consensus to coordinate actions and to maintain formation via geometric constraints. Each robot uses its ego-centric view of a target and the relative distance from its two closest neighbors to infer its steering commands. To account for noisy and missing target detections, the robots exchange their estimated target position and formation configuration through shared PID-controlled steering responses. We show that the proposed model enables the team to maintain the view of a maneuvering target with varying acceleration under noisy detections and failures up to situations when all robots but one lose the target from their field of view.**

I. Introduction

Cooperating robots are desirable for extended visual coverage [1]–[3], rapid search and rescue [4], [5] and collaborative grasping tasks [6]. Coordinated control toward a shared objective (e.g. following a target) while jointly planning and performing maneuvers (e.g. maintaining a formation) is challenging without an external localization system (e.g. in GPS-denied environments) [7]. Robots may rely on observing the relative position of their neighbors, which may be equipped with visual markers [8]. Robots in a formation need to account for noisy and missing detections caused by sensing failures or clutter as well as to coordinately adapt to unpredictable variations in target acceleration [9]–[11].

In this paper, we propose a distributed control model that allows aerial robots to coordinate within a formation using as a shared reference a moving target detected with their onboard camera (Fig. 1). The robots maintain a formation by continuously sensing the target and without a motion capture system or GPS. To infer the steering commands we exploit the orientation of each sensor with respect to the body of the robot. We considerably extend our previous work [12] and improve coordination by formulating a new PID controller and a new distributed information fusion algorithm that copes with target accelerations, decelerations and U-turns. We achieve a global agreement on the maneuvers of each robot using distributed consensus, which also enables the formation to avoid drifting when the target is not detected by one or more robots. We maintain the shape of the formation by constraining the robots' dynamics by their relative distance with their neighbors using geometrical constraints.

Unlike works that measure the 3D robot-target distance [13]–[15] using prior information about the size of the target [14], we infer control commands using only the position of the detected target on the image plane. Moreover, unlike [16] we localize the target with respect to the local reference system of each robot in order to allow the team to operate in absence of a global reference system.

Fabio Poiesi is with Technologies of Vision, Fondazione Bruno Kessler, Via Sommarive 18, Trento, IT, 38123, e-mail: <poiesi@fbk.eu>. He performed this work when he was with the Centre for Intelligent Sensing, Queen Mary University of London.

Andrea Cavallaro is with the Centre for Intelligent Sensing, Queen Mary University of London, Mile End Road, London, UK, E1 4NS, e-mail: <a.cavallaro@qmul.ac.uk>.

This work was supported by the Artemis JU and the UK Technology Strategy Board (Innovate UK) through project COPCAMS under Grant 332913. The support of the UK EPSRC through project NCNR (EP/R02572X/1) is also acknowledged.

II. Problem definition

Let a formation of N robots at time $k \in \mathbb{R}$ be defined as graph $F(k) = (\mathcal{C}(k), D(k))$, where the set $\mathcal{C}(k) = \{C_i(k)\}_{i=1}^{N}$ contains the state $C_i(k)$ of each robot i and the matrix $D(k) = [d_{i,j}(k)] \in \mathbb{R}^{N \times N}$ defines the distance $d_{i,j}(k)$ between robot i and j.

The robot state $C_i(k) = (x_i(k), R_i(k))$ consists of its position $x_i(k) \in \mathbb{R}^3$ in global coordinates and its orientation (attitude) $R_i(k) \in SO(3)$ [17], i.e. the rotation from the local (body) to the global coordinate system. The body of the robot is defined by three main body directions $b_{1,i}(k) = R_i(k)e_1$, $b_{2,i}(k) = R_i(k)e_2$ and $b_{3,i}(k) = -R_i(k)e_3$, where $e_1 = [1,0,0]^T$, $e_2 = [0,1,0]^T$ and $e_3 = [0,0,1]^T$. Let the subscript t indicate *target*. The target position in the global coordinate system is $x_t(k) \in \mathbb{R}^3$ and its velocity $v_t(k) = [v_{x,t}(k), v_{y,t}(k), v_{z,t}(k)]^T$.

Each robot i detects on its sensor plane the target with position $x_{t,i}(k) \in \mathbb{R}^2$. The 3D camera position corresponds to that of the robot. The objective is to maintain the target in the center of each sensor plane, and deviations from the center have to be mapped into steering commands to adjust the motion of the robots in order to achieve the objective.

III. Perception modelling

Let us consider a group of $N \geq 3$ aerial inertial robots capable of relative localization and target detection. The robots cannot rely on an external positioning systems, but can sense with their camera a target moving on the ground and with other sensors the relative position of their closest neighbors [18], [19].

Let the positions of all robots be initialized *in formation* and *on target*, with fixed body orientations. At $k = 0\ \forall i$, $C_i(0)$ is given, $v_t(0) = [0,0,0]^T$ and each sensing region has a different and fixed 3D orientation $R_{c,i}$ that points toward the target. $R_{c,i}$ is time invariant with respect to the body coordinate system $R_i(0)$ and the target is centered in each sensing region. The robot's inertial motion in the global coordinate system is determined by a controller that

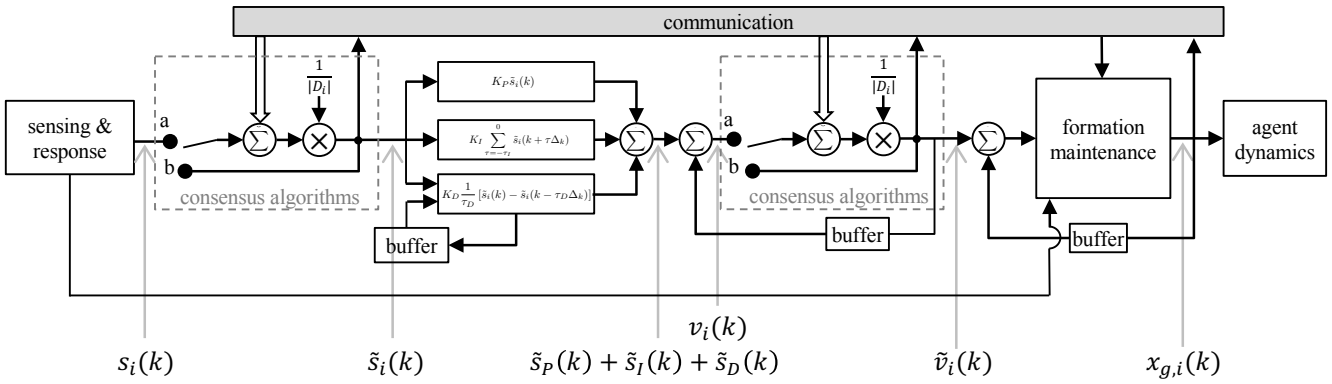

Fig. 1. Block diagram of the proposed control model. The position of the target on the sensor plane is used to determine the steering response, $s_i(k)$. Robots communicate to agree on a corrected steering response, $\tilde{s}_i(k)$, which is stabilized with a PID controller. Proportional, integral and derivative terms are indicated with $\tilde{s}_P(k)$, $\tilde{s}_I(k)$ and $\tilde{s}_D(k)$, respectively. The steering response proposes a velocity for each robot, which a consensus algorithm corrects ($\tilde{v}_i(k)$). Geometric constraints maintain the formation using the relative position of neighboring robots to compute the next goal position, $x_{g,i}(k)$, that each robot uses to determine its own dynamics.

follows a goal (desired) trajectory point $x_{g,i}(k) \in \mathbb{R}^3$, and a goal direction of the first body direction $b_{1_g,i}(k)$ [17] (the subscript g stands for *goal*).

Target detection errors and target localization noise may introduce estimation errors. We therefore model target *detection errors* and target *localization noise*. Let $p(x_{t,i}(k))$ be the probability for the sensor of robot i to detect the target at time k. We model target detection as a time-independent Bernoulli distribution

$$p(x_{t,i}(k)) = \begin{cases} 1 - p_d & \text{if target detected} \\ p_d & \text{otherwise,} \end{cases} \tag{1}$$

where p_d is the miss-detection probability. An example of miss-detection probability of a state-of-the-art pedestrian detector [20] is $p_d \sim 0.23$.

Moreover, let $\psi_i : \mathbb{R}^3 \mapsto \mathbb{R}^2$ be the transformation that projects the target center of mass from the 3D world to the 2D sensor plane of robot i. We model target localization as

$$x_{t,i}(k) = \begin{cases} \psi_i(x_t(k)) + \omega_i(k) & \text{if target detected} \\ \textit{null} & \text{otherwise,} \end{cases} \tag{2}$$

where $\omega_i(k) = \mathcal{N}(0, \sigma_\omega)$ is the Gaussian noise that models the target detection noise, and *null* indicates no decision. The value of σ_ω models the noise on the position of the detection. A typical value of this noise, when the target is a pedestrian, is $\sim$10% of the size of the bounding box enclosing the target.

IV. Individual steering response

The objective of each robot is to maintain the target centered in its sensor plane. The *steering direction* and the *steering gain* depend on the target dynamics and its (detected) location on the image plane.

Let $s_i(k)$ be the steering response defined as

$$s_i(k) = \begin{bmatrix} s_{x,i} \\ s_{y,i} \end{bmatrix} = s_{m,i}(k) \circ sgn(s_{d,i}(k)), \tag{3}$$

where $s_{m,i}(k) = [s_{m_x,i}(k), s_{m_y,i}(k)]^T$ is the steering gain, $s_{d,i}(k) = [s_{d_x,i}(k), s_{d_y,i}(k)]^T$ is the steering direction and $\circ$ is the Hadamard product. If $s_{d_x,i}(k) > 0$, $sgn(s_{d_x,i}(k)) = 1$; if $s_{d_x,i}(k) < 0$, $sgn(s_{d_x,i}(k)) = -1$; if $s_{d_x,i}(k) = 0$, $sgn(s_{d_x,i}(k)) = 0$. The steering response of a robot when it does not detect the target (*null* in Eq. 2) is zero.

The altitude can be measured with an onboard barometer [8], [21], but for simplicity we consider that the robots move on a (virtual) plane at a non-zero altitude. We therefore do not consider in this work variations in altitude, i.e. $s_{d_z,i}(k) = 0$. Fluctuations due to altitude measurement errors [8], [22] can be addressed by measuring size changes of the detected target on the sensor plane.

We compute the *steering direction terms* by projecting $x_{t,i}(k)$ onto the two axes on the sensor plane. Because the sensor orientation can differ from that of the body frame of the robot, the two axes have to be rotated accordingly via $R_{c,i}$. The axes are defined as $e_{x,i} = \psi_i(R_i(0)R_{c,i}e_1)$ and $e_{y,i} = \psi_i(R_i(0)R_{c,i}e_2)$. The origin of the axes is the center of the sensor plane. The steering direction components are computed as

$$s_{d,i}(k) = [e_{x,i}, e_{y,i}]^T x_{t,i}(k). \tag{4}$$

The *steering gain terms* are designed such that the steering response becomes more vigorous as the distance between $x_{t,i}(k)$ and the center of the sensor plane increases. These terms are calculated via a function $M : \mathbb{R}^2 \mapsto \mathbb{R}^2$, where $s_{d,i}(k) \mapsto M(s_{d,i}(k)) = s_{m,i}(k)$ maps $x_{t,i}(k)$ into the steering gain terms of $s_{m,i}(k)$. No steering is performed when the target detection is in the center of the sensor plane. Therefore M is zero (minimum) when $x_{t,i}(k)$ coincides with the center of the sensor plane. We choose M to be a Gaussian function to produce a quicker steering response than a linear mapping when the target moves farther away from the center. M is regulated via the variance $\sigma_m^2 \in \mathbb{R}$ of the Gaussian as

$$s_{m,i}(k) = 1 - \exp\left(-\frac{1}{2} x_{d,i}(k)^T \begin{bmatrix} \sigma_m^2 & 0 \\ 0 & \sigma_m^2 \end{bmatrix}^{-1} x_{d,i}(k) \right). \tag{5}$$

The farther the target from the center of the sensor plane (with respect to both axes), the larger the terms of $s_{m,i}(k)$.

Due to detection noise and different perspectives, the steering response inferred by a robot via the target detection on the sensor plane may be different from that of other robots. Moreover, if the target changes its motion, some robots might be unable to react timely and to adapt their motion to keep the target in their sensing range. The solutions to these problems will be presented in the next section.

V. Collaborative reactive steering

A. Coordinated motion control

Robots iteratively receive from and send to neighbors the steering response, $\tilde{s}_i^l(k)$, where l is the iteration index and $\tilde{s}_i^0(k) = s_i(k)$, the steering response that acts on the first and second body directions, namely $b_{1,i}(k)$ and $b_{2,i}(k)$ (Eq. 3). The goal is to maintain over time a certain distance between robot i and robot j, $d_{i,j}(k)$, i.e. $d_{i,j}(k) \approx d_{i,j}(0)\ \forall\ k$.

Let $\tilde{s}_i(k)$ be the steering response achieved via consensus[1] as

$$\tilde{s}_i(k) = \sum_{l=1}^{J_a} \frac{1}{|D_i|} \sum_{n \in D_i} R_i(k)^{-1} R_n(k) \tilde{s}_n^l(k), \qquad (6)$$

where J_a is the total number of iterations, $|.|$ is the cardinality of a set and D_i is the set of neighbors of robot i that have detected the target. The index n refers to a neighbor robot. We assume that robots share information without communication delays.

To achieve drive, tracking and fast response towards the desired steering response, unlike [12] we translate the steering response $\tilde{s}_i(k)$ inferred from the sensor plane into inertial controls for each robot using an integral and derivative (PID) controller [23]. The goal of our PID controller is to take the value of $\tilde{s}_i(k)$ (Eq. 6) to zero in order to achieve the objective of centering the target on each robot's sensor plane despite changes in target acceleration or vigorous turns. The *proportional value* of the PID acts directly on $\tilde{s}_i(k)$:

$$\tilde{s}_{P,i}(k) = K_P \tilde{s}_i(k), \qquad (7)$$

where K_P is the proportional gain. The *integral value* is obtained by accumulating the steering response over time as

$$\tilde{s}_{I,i}(k) = K_I \sum_{\tau=-\tau_I}^{0} \tilde{s}_i(k + \tau \Delta_k), \qquad (8)$$

where K_I is the integral gain, τ_I is the time interval for the integration (typically between the initialization of the system and the current time instant [24]) and Δ_k is the sampling time. The integration within the interval $[0, k]$ is unsuitable as it leads to a high rigidity of robot maneuvers. The reactivity of the robots increases by decreasing the integration interval. However, τ_I cannot be zero because this would lead to unstable maneuvers due to sharp responses of the controller to sudden motion variations of the tracked target. Finally, the *derivative value* is obtained by differentiating the steering response into two distinct time instants as

$$\tilde{s}_{D,i}(k) = \frac{K_D}{\tau_D} \left(\tilde{s}_i(k) - \tilde{s}_i(k - \tau_D \Delta_k) \right), \qquad (9)$$

where K_D is the derivative gain and τ_D defines the time interval to compute the differentiation. The larger τ_D, the more robust the steering response to target speed variations or to noisy measurements. The smaller τ_D, the quicker the reaction of a robot to target speed variations. A sensitive reaction to speed variations may cause oscillations due to sudden accelerations of the robots, with the possibility of losing the target from their fields of view; whereas a smooth reaction may not be sufficient to quickly adapt to the target motion variations and the target may be lost by the robots.

The proportional, integral and derivative values are combined to set a candidate PID controlled velocity as

$$v_{c,i}(k) = \tilde{v}_i(k - \Delta_k) + (\tilde{s}_{P,i}(k) + \tilde{s}_{I,i}(k) + \tilde{s}_{D,i}(k))\, \Delta_k, \qquad (10)$$

where $\tilde{v}_i(k - \Delta_k)$ is the agreed velocity of the ith camera.

Note that the PID controller acts on the individual steering response of each robot by enabling control stability via temporal integration. However, this integration step could magnify the marginal differences of the steering response computed in Eq. 6 as the number of iterations J_a might insufficient to make the steering response converge at the same value for all the robots. In order to ensure that the robots converge to the same velocity, we perform an additional consensus phase.

The $\tilde{v}_{c,i}(k)$ for robot i is the agreed velocity that is used in the next step to maintain the formation geometry:

$$\tilde{v}_{c,i}(k) = \sum_{l=1}^{J_b} \frac{1}{|D_i|} \sum_{n \in D_i} R_i(k)^{-1} R_n(k) \tilde{v}_{c,n}^l(k), \qquad (11)$$

where J_b is the total number of consensus iterations and $\tilde{v}_{c,n}^0(k) = v_{c,n}(k)$.

Our relative-positioning system relies on the local (body) reference system of each robot, not on a global coordinate system [25]. Unpredictable dynamics of the robots due to various factors, such as robots' inertia, noisy target detections and external disturbance (e.g. wind) can cause position variations. The two consensus phases used so far to agree on velocity do not include geometric knowledge that is necessary to maintain the shape of the formation. The inclusion of this knowledge will be discussed next.

B. Geometric constraints

We introduce geometric constraints to maintain, despite disturbances, the robots within their position in the formation. Each robot measures its relative distance from its neighbors and corrects its own velocity if the distance exceeds a certain tolerance.

We extend the idea proposed in [26] to cases when robots can sense the relative position of more than two neighbors and use non-hierarchical formation control to increase, compared to a leader-follower control scheme [26], robustness

[1]We chose the mean operator as it can fuse data for a 3-robot neighborhood. The median operator can be used when neighborhoods of at least four robots are available.

to speed variations of individual robots and to external disturbances (e.g. noisy target detections).

Given the relative and goal positions of neighbors with respect to the body reference of robot i and the candidate velocity $\tilde{v}_{c,i}(k)$, the inertia of the robot is computed via the desired trajectory point $x_{g,i}(k)$ to maintain itself in the formation. In the case of a quadrotor as robot, the inertial dynamics are determined by four identical propellers, which are equidistant from the center of the body and generate a thrust and torque normal to the body plane defined by $b_{1,i}(k)$ and $b_{2,i}(k)$ [17].

Given a goal trajectory point $x_{g,i}(k)$, the total thrust applied to a robot and the desired direction of the third-body axis $b_{3_g,i}(k)$ are selected to stabilize the translational dynamics[2]. In our case the robots' relative heading directions do not change during the flight and keep pointing to the direction set at initialization [12], [27].

To achieve coordinated motion control we first compute a candidate desired trajectory and then adjust it, if necessary, based on the circle intersection rule [26]. The candidate desired trajectory is calculated as

$$x_{c,g,i}(k) = x_i(k - \Delta_k) + R_i(k)\tilde{v}_{c,i}(k)\Delta_k. \quad (12)$$

To select $x_{c,g,i}(k)$ as the desired trajectory following the constraints of the formation shape, we initially compute the set of intersection points generated by all the neighbors of robot i as

$$\Theta_i(k) = \{\rho_m : \rho_m \in \Gamma(\hat{x}_{i,j}(k), d_{i,j}(0)) \cap \Gamma(\hat{x}_{i,q}(k), d_{i,q}(0)), \forall\, j, q \in D_i, j \neq q \neq i\}, \quad (13)$$

where $\Gamma(x, d)$ defines the circle with center x and radius d, $m \in \mathbb{N}$; $d_{i,j}(0)$ is the distance to be maintained between robot i and j, which is defined at initialization ($k = 0$); and $\hat{x}_{i,j}(k)$ and $\hat{x}_{i,q}(k)$ are computed by robot i using their measured positions and shared goal positions at $k - 1$, and the updated velocities received from robot j and q at k, respectively, such that

$$\hat{x}_{i,j}(k) = x_{g,j}(k - \Delta_k) + \eta_{i,j}(k) + R_j(k)\tilde{v}_{c,j}(k)\Delta_k, \quad (14)$$

where $\eta_{i,j}(k) = \mathcal{N}(0, \sigma_n)$ is an additive Gaussian noise on the measurement about the relative positions between two robots (i.e. [19]). σ_n models the localization error of the neighboring robots introduced by the proximity sensor.

When non-empty, the set $\Theta_i(k)$, contains at least $|D_i| - 2$ intersection points (two neighbors). Only the intersection points near the candidate $x_{c,g,i}(k)$ are considered. From the solution of Eq. 13, the points in $\Theta_i(k)$ can be divided into two sets: distant and near points from $x_{c,g,i}(k)$. We sort the near points in $\Theta_i(k)$ in ascending order based on their distance from $x_{c,g,i}(k)$ and we build a set $\Theta'_i(k)$ containing the first $\frac{|D_i|-2}{2}$ of the sorted points in $\Theta_i(k)$.

To confirm $x_{c,g,i}(k)$ as the desired trajectory that follows the constraints of the formation shape, we compute the point

$$\theta_{c,i}(k) = \frac{1}{|\Theta'_i(k)|} \sum_{m \in \Theta'_i(k)} \rho_m. \quad (15)$$

[2]For details on the control algorithm, please see [17].

The desired trajectory is then computed as

$$x_{g,i}(k) = \begin{cases} x_{c,g,i}(k) & \text{if } \Theta'_i(k) \notin \emptyset \wedge \epsilon_{i,j} > \varepsilon \\ \theta_{c,i}(k) & \text{otherwise,} \end{cases} \quad (16)$$

with $\epsilon_{i,j} = ||x_{c,g,i}(k) - \hat{x}_{i,j}(k)|| - d_{i,j}(0)$, $\forall i, j \in \Theta'_i(k)$, $i \neq j$, where $||.||$ is the ℓ-2 norm and ε is the separation tolerance term. The larger ε, the more constrained the robots in maintaining the inter-distances defined in $D(0)$.

VI. Validation

A. Simulation setup

We validate the proposed framework by analyzing the behavior of camera-equipped aerial robots that follow a moving target in a formation. We use the dynamics of a quadrotor to model the inertia of the robots [17].

Let each robot be equipped with a camera with focal length $f_L = 0.1$ and angle of view $\phi = \frac{\pi}{4}$. The resolution of the camera is $W = H = 600$ pixels. We evaluate the proposed method with a formation of up to $N = 12$ robots [28]. We set the altitude and the radius of the formation at $A = 5m$ and $L = 6m$, respectively, according to typical specifications of commercial quadrotors with a camera. The starting positions of the robots are

$$x_i(0) = \left(L\, cos((i-1)\frac{\pi}{N/2}),\ L\, sin((i-1)\frac{\pi}{N/2}),\ A \right)^T, \quad (17)$$

$\forall i \in \mathcal{C}(k)$. This configuration leads to an inter-distance among robots of about $3.1m$, that is the working distance of a realistic relative positioning system [18], [19].

Because the velocity of the target is unknown to the robots, we set the initial velocity equal to $v_t(0) = [0, 0, 0]^T\ \forall i$, with $\Delta_k = 0.04$ *sec* for all the experiments. The matrix $D(k)$ is defined such that each robot i is only aware of its closest neighbors and the distance $d_{i,j}$ to be maintained depends on the initial positions defined in Eq. 17. The following values are used: $\varepsilon = 0.1$ (Eq. 16) and $\sigma_m = \frac{W}{3}$ (Eq. 5); this value of σ_m allows the mapping function to be almost zero at the borders of the image plane. We chose the PID configuration with $K_P = 0.08$, $K_I = 0.00022$ and $K_D = 3$ based on a sensitivity analysis that we performed.

To analyze the reaction of the system to *noisy and missed target detections* we vary the success probability of detecting the target within the interval $p_d \in [0,\ 0.72]$ (Eq. 1) and the additive Gaussian noise (Eq. 2) with standard deviation $\sigma_\omega = p_\omega W$ within $p_\omega \in [0,\ 0.36]$. We perform 20 runs for each parameter and each target trajectory, and average the results. Fig. 2 shows the 10 target trajectories, $T_1, ..., T_{10}$, we employed for the evaluation that were chosen from a randomly generated set using the code from [29]. Each trajectory is characterized by different target dynamics with variations in speed and direction. Speed variations range from $1.04m/s$ to $9.20m/s$, which are comparable to a person walking and performing high intensity sport activities, respectively [30]. The target is $1.75m$ tall.

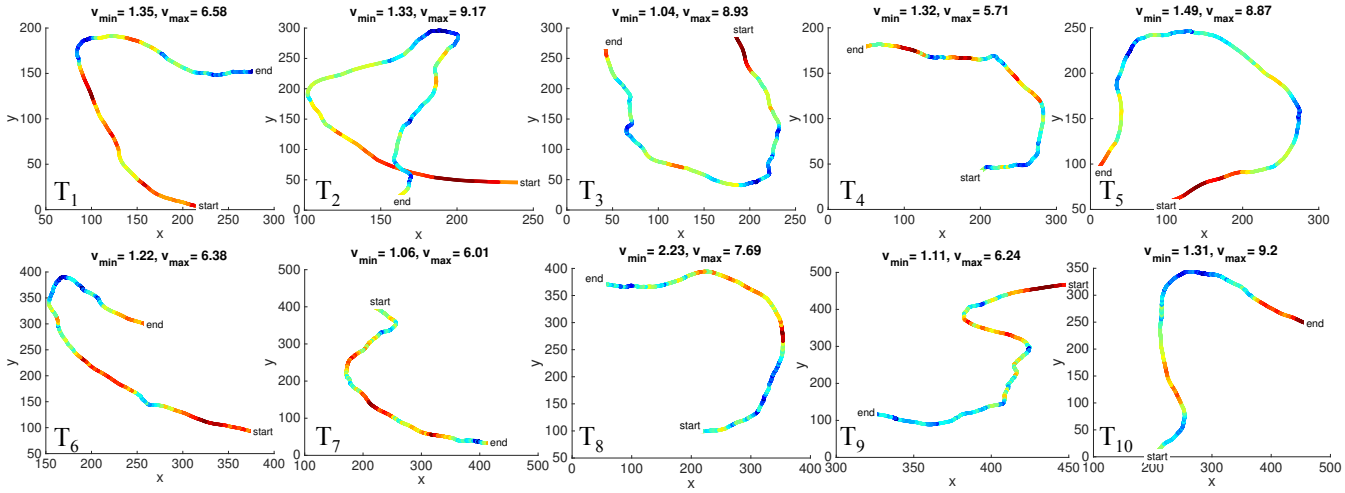

Fig. 2. The trajectory set. Speed variations are in dark blue for minimum velocity, v_{min}, and in dark red for maximum velocity, v_{max}. T_1: high velocity before a slow curve; T_2: steady acceleration that reaches a high velocity and a severe U-turn; T_3: start position at high velocity; T_4: alternated high and slow velocity; T_5 high initial velocity and large loop; T_6 alternated high and slow velocity plus U-turn with velocity variations; T_7: zig-zag start with acceleration just before a curve; T_8: curve at a high velocity; T_9: initial high velocity plus severe zig-zag; T_{10}: acceleration after the curve.

TABLE I

TARGET INCLUSION PERFORMANCE (MEAN AND STANDARD DEVIATION) BY VARYING DETECTION NOISE, MISSED DETECTION PROBABILITY AND ROBOT RELATIVE LOCALIZATION NOISE IN THE CASE OF THE 12-ROBOT FORMATION.

	No noise	Detection noise (p_ω)				Missed detection (p_d)			Localisation noise (σ_n)			
		.06	.20	.28	.36	.24	.48	.72	.005	.010	.015	.021
T_1	.66 ± .36	.67 ± .34	.54 ± .38	.13 ± .13	.01 ± .08	.66 ± .35	.65 ± .35	.40 ± .26	.58 ± .23	.36 ± .13	.14 ± .08	.05 ± .08
T_2	.33 ± .33	.16 ± .15	.00 ± .06	.00 ± .06	.01 ± .06	.32 ± .31	.07 ± .12	.00 ± .06	.32 ± .25	.07 ± .07	.02 ± .06	.01 ± .06
T_3	.36 ± .37	.31 ± .32	.09 ± .12	.02 ± .06	.01 ± .06	.36 ± .38	.31 ± .32	.05 ± .07	.29 ± .26	.13 ± .11	.04 ± .06	.01 ± .06
T_4	.63 ± .37	.64 ± .37	.58 ± .36	.32 ± .29	.14 ± .11	.63 ± .37	.61 ± .36	.64 ± .34	.54 ± .28	.38 ± .15	.12 ± .08	.04 ± .08
T_5	.44 ± .38	.37 ± .36	.07 ± .09	.01 ± .06	.01 ± .06	.45 ± .39	.37 ± .34	.01 ± .07	.35 ± .27	.19 ± .14	.04 ± .06	.02 ± .06
T_6	.69 ± .36	.68 ± .36	.22 ± .21	.10 ± .10	.14 ± .14	.68 ± .36	.63 ± .36	.10 ± .12	.57 ± .26	.38 ± .15	.12 ± .08	.04 ± .07
T_7	.65 ± .34	.67 ± .33	.56 ± .32	.16 ± .12	.02 ± .07	.63 ± .35	.63 ± .35	.60 ± .32	.57 ± .23	.35 ± .12	.13 ± .09	.04 ± .07
T_8	.66 ± .36	.67 ± .36	.62 ± .37	.07 ± .08	.02 ± .07	.67 ± .35	.63 ± .35	.62 ± .35	.60 ± .25	.33 ± .14	.14 ± .08	.02 ± .07
T_9	.59 ± .40	.54 ± .38	.44 ± .34	.13 ± .13	.03 ± .07	.60 ± .39	.51 ± .39	.46 ± .34	.50 ± .27	.20 ± .11	.07 ± .07	.03 ± .07
T_{10}	.71 ± .37	.69 ± .37	.55 ± .37	.11 ± .16	.01 ± .07	.71 ± .37	.67 ± .37	.28 ± .31	.61 ± .27	.36 ± .15	.12 ± .08	.03 ± .07

B. Discussion

We quantify the performance in terms of average *target inclusion*, namely the percentage of the target included in the central area of the image plane, defined as a circle centered in $[0,0]^T$ with radius $\frac{W}{3}$, where W is the width of the image plane. When the target is centered in the image plane the target inclusion is equal to one. A *failure* occurs when the formation drifts from the target and the robots cannot generate steering responses to follow it. We evaluate the attitude of the formation by changing the parameters of the proposed vision-based controller, the two-phase distributed consensus and the use of multiple neighbors to achieve improved robustness in maintaining the formation shape.

Tab. I shows the target inclusion averaged over time and across the performance of robots in the formation under *detection noise* and *miss-detections*. The target inclusion performance does not reach 1 in the case of "No noise" because the target's motion variations are interpreted with a delay by robots when the target accelerates or decelerates. T_2 is the most challenging trajectory because of its initial vigorous target acceleration. Detection noise is what affects the performance the most. With some trajectories the formation can maintain track of the target, even when the standard deviation is a fifth of the image plane size ($p_\omega = 0.2$). The formation is robust also when the target is miss-detected with $p_d = 0.72$ (independently) on the image plane of each robot. This robustness is possible thanks to the redundancy introduced by multiple robots monitoring the target and because the consensus can effectively propagate the steering commands within the formation.

Relative localization noise affects the robots when they estimate the position of their neighbors. We model the relative localization noise within the interval $\sigma_n \in [0,\ 0.021]$ to analyze the performance up to system failure, using data collected with real devices [18] (summarized in [19]). In Tab. I we can observe that the robots can maintain the formation up to $\sigma_n = 0.01$, on average. The maximum localization error is about $3cm$ (on a distance of $3.1m$), which agrees with the specifications of the relative positioning system in [18]. T_2 is the only trajectory where the maximum localization error allowed before failure is about $2.1cm$. When the formation fails to track the target because of large noise in localization, the error $\epsilon_{i,j}$ is larger than the separation tolerance term. Therefore, each desired trajectory

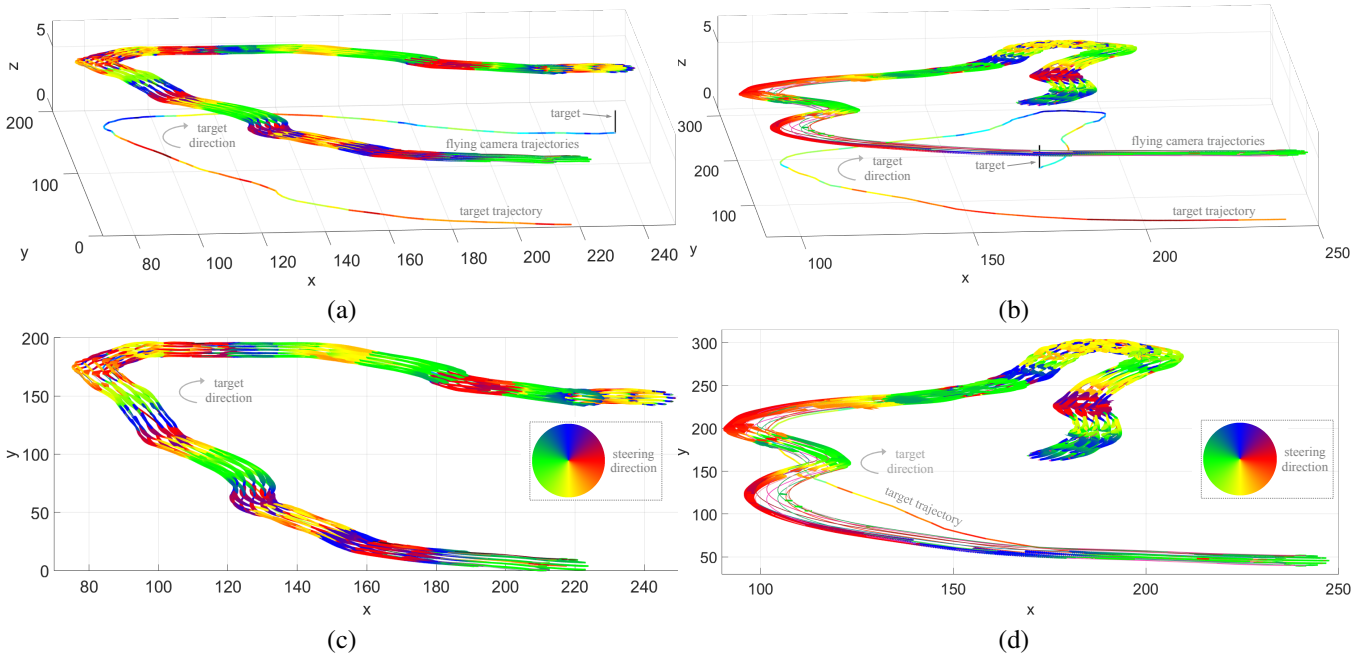

Fig. 3. Steering directions for the first 2500 time steps of (a,c) T_1 and (b,d) T_2 from (a,b) a 3D point of view and (c,d) top view. Note that the colored steering directions are generated before the distributed consensus step ($s_i(k)$) and the color of the target trajectory represents target acceleration and deceleration only (and not the steering direction as for the robots).

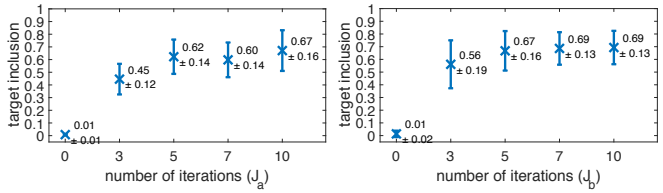

Fig. 4. Target inclusion performance (mean and standard deviation) for the distributed consensus steps in (a) Eq. 6 and (b) Eq. 11 obtained on T_1 under noisy conditions ($p_\omega = 0.06$, $p_d = 0.2$, $\sigma_n = 0$) and with communication failures during consensus ($p_f = 0.2$). (a) Variations of J_a with $J_b = 20$. (b) Variations of J_b with $J_a = 5$.

computed for the successive time step (Eq. 16) follows the candidate desired trajectory (Eq. 12), which *does not account for the geometric constraints of the formation, but only the velocity agreed via distributed consensus*.

Fig. 3 shows the steering directions of the first 2500 time steps of T_1 and T_2. A steering agreement is visible when the robots move together (visualized with the same color). The robots steering directions are generated before the distributed consensus step ($s_i(k)$). In Fig. 3a,c robots can follow the target despite steering disagreement. A steering disagreement may in fact emerge near to curves in the target trajectory, but it can be corrected using the distributed consensus. In Fig. 3b,d the robots drift from the target after the first part of the trajectory, but thanks to the external robots that can still view the target, the formation returns on target. During the returning phase (at about $x = 120$, $y = 150$) there is an oscillation and then the formation gets back on target. When the robots are back on target (at about $x = 140$, $y = 240$) the steering direction is opposite to the target motion direction as the target decelerates. Along this deceleration segment (green part) there is another disagreement about the steering direction. The robots generate steering responses with different directions due to the different target positions on their image plane.

Next, we analyze the method *with and without the distributed consensus* on T_1 to show the number of iterations used to achieve stable steering commands under noisy detections ($p_\omega = 0.06$, $p_d = 0.2$, $\sigma_n = 0$) and communication failures during consensus iterations. We simulate communication failures using the probability of failing, p_f, to send the steering data to the neighbor robot. We set $p_f = 0.2$. Fig. 4a shows that, without using distributed consensus, the formation is lost and that five consensus iterations of the steering response term led to steady performance. Fig. 4b shows the contribution of the consensus on the velocity term in achieving successful tracking.

Tab. II shows the temporal average of the inclusion performance using a single robot, and *formations composed of three, six and twelve robots* on T_1 and T_2 with ($p_\omega = 0.05$ $p_d = 0.1$ $\sigma_n = 0.005$) and without ($p_\omega = p_d = \sigma_n = 0$) noise. We can observe that the target inclusion improves

TABLE II

TARGET INCLUSION PERFORMANCE (MEAN AND STANDARD DEVIATION) WITH A SINGLE ROBOT AND WITH FORMATIONS COMPOSED OF THREE, SIX AND TWELVE ROBOTS UNDER DIFFERENT NOISE CONDITIONS.

		1	3	6	12
$p_\omega = p_d = \sigma_n = 0$	T_1	.03 ± .16	.58 ± .41	.69 ± .38	.70 ± .37
	T_2	.01 ± .09	.02 ± .09	.01 ± .06	.46 ± .42
$p_\omega = .05$ $p_d = .1$ $\sigma_n = .005$	T_1	.02 ± .11	.14 ± .09	.42 ± .17	.58 ± .25
	T_2	.01 ± .10	.03 ± .07	.04 ± .06	.24 ± .19

TABLE III

PERFORMANCE COMPARISON (MEAN AND STANDARD DEVIATION) BETWEEN [12] AND THE PROPOSED APPROACH (PRO) WITH $p_\omega = p_d = \sigma_n = 0$.

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
[12]	.08 ± .22	.01 ± .08	.01 ± .06	.03 ± .14	.03 ± .12	.06 ± .20	.02 ± .11	.08 ± .20	.05 ± .18	.06 ± .16
Pro	.70 ± .38	.41 ± .42	.30 ± .38	.61 ± .39	.47 ± .43	.68 ± .38	.60 ± .38	.62 ± .39	.60 ± .42	.72 ± .38

as the number of robots increases. In T_2 a single robot and formations with three and six robots are unable to follow the target. By analysing the qualitative results we observed that the single robot fails a few time steps after the initialization because when the target goes out of the field view the steering responses cannot be generated sufficiently quick and accurately to maintain track of the target. The redundancy introduced with a three-robot formation allows us to cope with targets going outside the field of view of one or two robots as it is still possible to generate steering responses, communicate them to the other robots and drive the formation back on target. A higher redundancy, e.g., with six and twelve robots makes it possible to achieve even better target inclusion performance.

Tab. III shows the results of the comparison between [12] and the proposed approach. Results show that the more realistic challenges introduced with the new trajectories make [12] fail in all cases. The small values of target inclusion in [12] are generated because the initialisation is performed on target. With the proposed approach, after a few time steps the robots can adapt their velocity and follow the target, whereas [12] fails to adapt. Also, note that the proposed approach, as opposed to [12], does not use a prior velocity and starts with speed zero. There are two main reasons for the superior performance of the proposed approach. Firstly, the PID controller allows the robots to rapidly correct their velocity when the target performs motion variations. Secondly, the distributed consensus stages allow the robots that see the target to achieve an agreement on the steering commands and to communicate this information to the others.

C. Execution times

The value of the steering response shared among robots is a double ($8bytes$). The communication channel is unicast between each pair of robots using IEEE 802.11a with throughput up to $54Mbps$. We assume that the communication is TCP, so for each data packet transmitted (DATA), the receiver replies with an acknowledgement (ACK) [31]. The time to transmit a packet encapsulating a double value is

$$T_{DATA} = T_{preamble} + T_{header} + \left\lceil \frac{30.75 + L}{BpS(m)} \right\rceil T_{symbol}$$
$$= 16\mu s + 4\mu s + \left\lceil \frac{30.75 + 8}{27} \right\rceil 4\mu s = 28\mu s, \quad (18)$$

where $T_{preamble}$ is the preamble duration, T_{header} is the header duration, T_{symbol} is the symbol interval, L is the payload in *bytes* and $BpS(m) = m/2$ is the number of bytes in a symbol at m Mbps. The time to receive the ACK is

$$T_{ACK} = T_{preamble} + T_{header} + \left\lceil \frac{16.75}{BpS(m)} \right\rceil T_{symbol}$$
$$= 16\mu s + 4\mu s + \left\lceil \frac{16.75}{27} \right\rceil 4\mu s = 24\mu s. \quad (19)$$

When a packet is sent, the robot needs to wait $16\mu s$ (SIFS) to receive the ACK. To send another packet the robot needs to wait $34\mu s$ (DIFS), in order to avoid chances of collisions. The time the consensus takes to perform 5 iterations and communicate with 2 neighbors is

$$4 \times 5 \times (T_{DATA} + 16\mu s + T_{ACK} + 34\mu s) = 2.04ms, \quad (20)$$

where 4 considers the transmission and reception of one packet to and from each neighbor.

The cost for formation maintenance can be estimated similarly to the consensus algorithm above, but with one iteration only:

$$4 \times (T_{DATA} + 16\mu s + T_{ACK} + 34\mu s) = 0.41ms. \quad (21)$$

Finally, we consider $0.01ms$ for PID and robot dynamics as the PID can be computed instantly and the transmission of commands to the motors (robot dynamics) can be performed at $250Hz$ [32].

TABLE IV

EXECUTION TIMES FOR THE MAJOR STEPS.

Steps	**Time (ms)**
Consensus algorithm (2.04×2)	4.08
PID controller	0.01
Formation maintenance	0.41
Robot dynamics	0.01
Interval between two frames ($25Hz$)	**40.00**
Interval for robot perception & response	35.49

Table IV summarizes the time required for each step of the proposed framework. A 25Hz camera allows for $40ms$ to process the data, make decisions and actuate the robot dynamics. Based on the above analysis, an object detector (e.g. [33]) running onboard faster than $35ms$ would therefore allow the proposed framework to operate in real time. Current state-of-the-art pedestrian detectors working in unconstrained scenarios can run at $37ms$ per frame [34] and therefore we are expecting to be able to soon run the proposed framework in real time using a standard 25Hz camera.

VII. CONCLUSION

We presented a closed-loop distributed control model that enables a formation of aerial robots to follow a moving target without relying on an external positioning system. The target is detected independently on each robot's sensor plane and

the target position is mapped into steering controls. To account for noisy detections, the inferred steering controls are corrected via distributed consensus to achieve an agreement on the maneuvers to accomplish. A PID controller helps each robot within the formation to achieve flight stability. The formation is maintained over time via geometric constraints based on the relative position of neighbors. The formation can handle noisy and missed target detections occurring for each robot independently, and is also robust to relative localization errors.

We are working towards the development of the physical collaborative robot network that embodies the model described in this paper. To this end, we will also address the problems caused by communication delays and the on-the-fly correction of the PID parameters to adapt to the target dynamics.

REFERENCES

[1] M. Schwager, B. Julian, M. Angermann, and D. Rus, "Eyes in the sky: Decentralized control for the deployment of robotic camera networks," *Proceedings of IEEE*, vol. 99, no. 9, pp. 1541–1561, Sep. 2011.

[2] L. Doitsidis, S. Weiss, A. Renzaglia, M. Achtelik, E. Kosmatopoulos, R. Siegwart, and D. Scaramuzza, "Optimal surveillance coverage for teams of micro aerial vehicles in GPS-denied environments using onboard vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 173–188, Aug. 2012.

[3] A. Adaldo, S. Mansouri, C. Kanellakis, D. Dimarogonas, K. Johansson, and G. Nikolakopoulos, "Cooperative coverage for surveillance of 3d structures," in *Proc. of IROS*, Vancouver, CA, Sep. 2017.

[4] A. Macwan, J. Vilela, G. Nejat, and B. Benhabib, "A multirobot path-planning strategy for autonomous wilderness search and rescue," *Trans. on Cybernetics*, vol. 45, no. 9, pp. 1784–1797, Sep. 2015.

[5] M. Liu, K. Sivakumar, S. Omidshafiei, C. Amato, and J. How, "Learning for multi-robot cooperation in partially observable stochastic environments with macro-actions," in *Proc. of IROS*, Vancouver, CA, Sep. 2017.

[6] M. Saska, T. Baca, V. Spurny, G. Loianno, J. Thomas, T. Krajnik, P. Stepan, and V. Kumar, "Vision-based high-speed autonomous landing and cooperative objects grasping - towards the mbzirc competition," in *Workshop on Vision-based High Speed Autonomous Navigation of UAVs (IROS)*, Daejeon, KO, Oct. 2016.

[7] A. Khan, B. Rinner, and A. Cavallaro, "Cooperative robots to observe moving targets: Review," *IEEE Trans. on Cybernetics*, vol. 48, no. 1, pp. 187–198, Jan. 2018.

[8] T. Nageli, C. Conte, A. Domahidi, M. Morari, and O. Hilliges, "Environment-independent formation flight for micro aerial vehicles," in *Proc. of IROS*, Chicago, IL, USA, Sep. 2014, pp. 1141–1146.

[9] M. Saska, T. Baca, J. Thomas, J. Chudoba, L. Preucil, T. Krajnik, J. Faigl, G. Loianno, and V. Kumar, "System for deployment of groups of unmanned micro aerial vehicles in GPS-denied environments using onboard visual relative localization," *Autonomous Robots*, pp. 1–26, Apr. DOI: 10.1007/s10514-016-9567-z, 2016.

[10] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV," in *Proc. of ICRA*, Hong Kong, CN, May 2014, pp. 4974–4981.

[11] C. Teulire, E. Marchand, and L. Eck, "3-d model-based tracking for uav indoor localization," *IEEE Trans. on Cybernetics*, vol. 45, no. 5, pp. 869–879, May 2015.

[12] F. Poiesi and A. Cavallaro, "Distributed vision-based flying cameras to film a moving target," in *Proc. of IROS*, Hamburg, DE, Sep. 2015, pp. 2453–2459.

[13] F. Morbidi and G. Mariottini, "On active target tracking and cooperative localization for multiple aerial vehicles," in *Proc. of IROS*, San Francisco, CA, USA, Sep. 2011, pp. 2229–2234.

[14] B. Fidan, V. Gazi, , S. Zhai, N. Cen, and E. Karatas, "Single-view distance-estimation-based formation control of robotic swarms," *IEEE Trans. on Industrial Electronics*, vol. 60, no. 12, pp. 5781–5791, Dec. 2013.

[15] M. Aranda, G. Lopez-Nicolas, C. Sagues, and M. Zavlanos, "Three-dimensional multirobot formation control for target enclosing," in *Proc. of IROS*, Chicago, IL, USA, Sep. 2014, pp. 357–362.

[16] S. H. Semnani and O. Basir, "Semi-flocking algorithm for motion control of mobile sensors in large-scale surveillance systems," *Trans. on Cybernetics*, vol. 45, no. 1, pp. 129–137, Jan. 2015.

[17] T. Lee, M. Leok, and N. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," in *Proc. of ICRA*, Atlanta, GA, USA, Dec. 2010, pp. 5420–5425.

[18] F. Rivard, J. Bisson, F. Michaud, and D. Letourneau, "Ultrasonic relative positioning for multi-robot systems," in *Proc. of ICRA*, Pasadena, CA, USA, May 2008, pp. 323–328.

[19] J. Roberts, T. Stirling, J.-C. Zufferey, and D. Floreano, "3-D relative positioning sensor for indoor flying robots," *Autonomous Robots*, vol. 33, no. 1-2, pp. 5–20, Aug. 2012.

[20] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?" in *Proc. of ECCV*, Zurich, CH, Sep. 2014, pp. 613–627.

[21] X. Zhang, B. Xian, B. Zhao, and Y. Zhang, "Autonomous flight control of a nano quadrotor helicopter in a GPS-denied environment using on-board vision," *IEEE Trans. on Industrial Electronics*, vol. 62, no. 10, pp. 6392–6403, Sep. 2015.

[22] M. Tanveer, S. Ahmed, D. Hazry, F. Warsi, and M. Joyo, "Stabilized controller design for attitude and altitude controlling of quad-rotor under disturbance and noisy conditions," *American Journal of Applied Sciences*, vol. 10, no. 8, pp. 819–831, 2013.

[23] K. Ang, G. Chong, and Y. Li, "PID control system analysis, design, and technology," *IEEE Trans. on Control Systems Technology*, vol. 13, no. 4, pp. 559–576, Jul. 2005.

[24] I. Sa and P. Corke, "System identification, estimation and control for a cost effective open-source quadcopter," in *Proc. of ICRA*, Saint Paul, MN, USA, May 2012, pp. 2202–2209.

[25] M. Bartels and H. Werner, "Cooperative and consensus-based approaches to formation control of autonomous vehicles," in *Proc. of International Federation of Automatic Control*, Cape Town, South Africa, Aug. 2014, pp. 8079–8084.

[26] B. Anderson, B. Fidan, C. Yu, and D. Walle, "UAV formation control: Theory and application," in *Recent Advances in Learning and Control*. Springer, 2008, vol. 371, pp. 15–33.

[27] M. Saska, J. Vakula, and L. Preucil, "Swarms of micro aerial vehicles stabilized under a visual relative localization," in *Proc. of ICRA*, Hong Kong, CN, May 2014, pp. 3570–3575.

[28] G. Vasarhelyi, C. Viragh, G. Somorjai, N. Tarcai, T. Szorenyi, T. Nepusz, and T. Vicsek, "Outdoor flocking and formation flight with autonomous aerial robots," in *Proc. of IROS*, Chicago, USA, Sep. 2014, pp. 3866–3873.

[29] S. Oh, S. Russel, and S. Sastry, "Markov Chain Monte Carlo data association for multi-target tracking," *IEEE Trans. on Automatic Control*, vol. 54, no. 3, pp. 481–497, Mar. 2009.

[30] A. Ferro, J. Villacieros, P. Floria, and J. Graupera, "Analysis of speed performance in soccer by a playing position and a sports level using a laser system," *Journal of Human Kinetics*, vol. 44, pp. 143–153, Dec. 2014.

[31] Y. Kim, S. Choi, K. Jang, and H. Hwang, "Throughput enhancement of IEEE 802.11 WLAN via frame aggregation," in *Proc. of Vehicular Technology Conference*, Los Angeles, CA, USA, Sep. 2004, pp. 3030 –3034.

[32] J. Witt, B. Annighofer, O. Falkenberg, and U. Weltin, "Design of a high performance quad-rotor robot based on a layered real-time system architecture," in *Proc. of Intelligent Robotics and Applications*, Aachen, GE, Dec. 2011, pp. 312–323.

[33] D. Ribeiro, A. Mateus, J. Nascimento, and P. Miraldo, "A real-time pedestrian detector using deep learning for human-aware navigation," *arXiv:1607.04441 [cs.RO]*, Jul. 2016.

[34] F. D. Smedt, D. Hulens, and T. Goedeme, "On-board real-time tracking of pedestrians on a uav," in *Proc. of CVPR Workshops*, Boston, MA, USA, Jun. 2015.