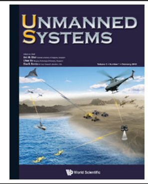
Unmanned Systems, Vol. 10, No. 2 (2022) 109–128

DOI: 10.1142/S2301385022500066

UNMANNED SYSTEMS
https://www.worldscientific.com/worldscinet/us




# Parametric and Implicit Features-Based UAV–UGVs Time-Varying Formation Tracking: Dynamic Approach

Ahmed Allam[*,‡], Abdelkrim Nemra[†,§], Mohamed Tadjine[*,¶]

[*]Laboratoire Commande des Processus,
Ecole Nationale Polytechnique10 Rue des Frères Oudek
El Harrach 16200 Algiers, Algeria

[†]VAI Laboratory, Ecole Militaire Polytechnique
Bordj El-bahri, Algiers, Algeria

Flexible and robust Time-Varying Formation (TVF) tracking of Unmanned Ground Vehicles (UGVs) guided by an Unmanned Aerial Vehicle (UAV) is considered in this paper. The UAV–UGVs system control model is based on leader-follower approach, where the control scheme consists of two consecutive tasks, namely, deployment task and TVF tracking. Accordingly, two novel nonlinear controllers are proposed for controlling the UGVs formation. First, unlike the classical frameworks on UGVs formation tracking, for which only particular shapes are handled (e.g. circle, square, ellipse), we propose a UGVs deployment-controller ensuring to reach free-formation shapes. The key feature is in using the estimated implicit representation of the desired formation shape as a potential function to generate the UGVs reference trajectory. Second, in the TVF tracking task, a robust cascaded velocity/torque controller for UGVs is proposed based on kinematic and dynamic models. Differently from the classical backstepping framework, the key idea is in introducing an auxiliary control input, in such a way that the overall UGV dynamics is converted into a simpler and modular control structure. As such, the auxiliary input is used to control indirectly the actual UGVs velocity vector. A signum term is added to the torque-input to compensate for the unknown external disturbances and unmodeled dynamics. Numerical simulation shows the effectiveness of the proposed formation controllers compared with the case when the perfect velocity-tracking assumption holds. Experimental results are further provided using three festos Robtino robots to show the validity of the proposed TVF tracking velocity-control scheme.

*Keywords*: UGVs formation tracking; robust control; Lyapunov stability; backstepping technique; quadrotor.

## 1. Introduction

Over the past two decades, Multi-Agent/Robot Systems (MAS/MRS) and their potential applications have attracted a lot of interest in many research fields of robotics [1]. Collaboration in an MRS offers valuable advantages comparing to the use of a single robot when accomplishing a given complex task, and increases the capabilities and the efficiency of execution. One of the promising research branches of MRS is UAV–UGV control and coordination.

Received 16 February 2021; Accepted 27 June 2021; Published 16 September 2021. This paper was recommended for publication in its revised form by editorial board member, Lu Liu.
Email Addresses: [‡]*ahmed.allam@g.enp.edu.dz*, [§]*karim_nemra@yahoo.fr*, [¶]*mohamed.tadjine@g.enp.edu.dz*

The researches on UAVs–UGVs control and coordination have attracted increasing attention. In particular with the rapid advances of communication, sensing, and embedded techniques. This is due to the broad and various applications of UAV–UGV coordination systems in both civilian and military fields such as exploration [2], surveillance and inspection [3], rescue and covering application [4].

The UAV–UGV coordination has demonstrated the capabilities in providing effectiveness, robustness, reliability, and practical solutions to the real-world that cannot be brought by other types of coordination. For instance, UAVs can serve as communication-relay or embed a GPS-receiver. Thus, the UGVs group becomes less prone to communication failures and the navigation–localization uncertainties may be reduced significantly [5]. Nonetheless, the applications relating to a formation of autonomous robots guided by UAVs such as target enclosing/tracking reveal the complementarities and the versatility brought to the overall system, where the UAVs can provide global, precise, and large aerial viewing and sensors measurements from the environment to the ground robots. As an example [6], an enemy-target can be observed, and localized by a UAV, and then its position is transmitted to ground robots to perform a task (e.g. track and enclose a target).

The motion control of MRS has received considerable attention from researchers in the robotic community. In the literature, the motion control is mainly based on three approaches. The most common one is leader-follower, in which some robots are assigned as leaders while the rest of the formation robots are considered followers. Each follower is controlled in order to maintain with its leader a given configuration, while the leader tracks a predefined reference trajectory. This approach presents the advantages of simplicity and efficiency. However, the main drawbacks are in error propagation, nonself-organization formation, and when one of the leaders fails to track its trajectory, its follower robots fail too [7]. To overcome these limitations, other alternative solutions are proposed by implementing leader reassignment technique or by adopting a strategy based on a virtual leader [8]. Another control approach is the virtual structure model, where the leader is virtual and the system is considered as a virtual rigid body with a fixed geometric model describing the special relationship between robots [9]. Therefore, the leader never fails and the stability of the whole system is not depending on the leader. The third motion control approach is behavioral-based method, in which we assign to each robot some desired local interactional sub-behaviors (i.e. robot-robot spacing, obstacle avoidance, goal-achieving), and the combination of these sub-behaviors results in a robot final behavior [10]. The method exhibits the advantages of decentralization aspect, scalability, robustness, and easy implementation. However, the method requires high computations gain. The previously introduced MRS control approaches can be unified within the general framework of consensus control [11]. The consensus protocols are based on the idea where all the vehicles update individually their information state relying on their local neighbors' information states. As a result, the entire vehicle's final information state converges to a common value. Therefore, an agreement is reached by all the agents on certain variables of interest [12].

Among the aspects relating to MRS control, we consider in this paper the TVF tracking of a UGVs group assisted by a UAV. A TVF control refers to the ability of MRS to change its formation shape (i.e. geometric relations between robots) in specific conditions, while tracking a reference trajectory. The formation's stability during the TVF tracking is supposed to be conserved. The formation shape changing can be required for many reasons such as

(i) Covering large parts of an area, wherein specific applications relating to environment-mapping, the ability of an MRS to spread out and to gather is essential.
(ii) Avoiding obstacles during the formation motion is critical for the MRS, where it is a practical way for the MRS to change the shape of the formation when facing obstacles to avoid collision.
(iii) Tracking and enclosing a target, where a group of robots needs to adjust the formation shape in such a way to track and surround a target for protection purposes, for instance.

Many frameworks have been reported in TVF tracking of UGVs group assisted by a UAV. In [13], the authors have considered a cooperative maneuver among a UAV–UGVs system, where the UGVs are guided by the UAV for obstacles avoidance purposes. In [14], a coherent TVF control of heterogeneous multi-agent system was considered. Lyapunov theory and synchronization method were used to design a decentralized controller to stabilize the swarming of UAV–UVGs system. In [15], a vision-based control method was presented for the guidance of a set of UGVs to reach a desired formation. The UVGs control relies on multiple cameras-equipped UAV as a control unit. In [16], a self-assembling of UGVs formation assisted by UAV was proposed, where the UAV uses the environment views to control and supervise the morphology formation of UGVs. In [17], a leader-follower-based approach was presented for heterogeneous UAV–UGV system control. The controller is based on kinematic models and relies on a centralized structure. Similar to [17], the authors in [18] proposed a virtual structure based approach to control a line formation of UGVs guided by a quadrotor. The method is based on kinematic models too. In [19], an improved expert PID target tracking control algorithm was proposed for the UAVs–UGVs system yielding an improvement of the system stability.

Almost all the frameworks mentioned above did not consider the dynamical models of the heterogeneous MRS. Furthermore, the handled formation shapes are specific forms such as (circle, line, rectangle, etc.). However, in practical applications, tracking free formation shapes by MRS is very practical, without losing the system stability. In [20], an operator named Elliptic Fourier Descriptors (EFDs) has been used for shape modeling and robots path planning. Image processing and computer vision are basically the fields that use the EFDs, where free-form shapes (i.e. closed contours) could be represented in digital images. In [20], the parametric and implicit models of the desired formation shape have been used to design the robots convergence controller. However, the controller is valid only for robots featured by first integrator model. Dynamic EFDs have been introduced in [21], and a formation controller has been derived to maintain a group of holonomic robots on a dynamic 2D curve. In our previous work [22], we extended the method proposed in [21] to design a controller for 3D planar formations shapes tracking using an extended dynamic version of EFDs.

Motivated by the aforementioned frameworks, and the potential applications of UAV–UGV systems, we propose an abstraction model for a UGVs group assisted by a UAV, where the UAV (i.e. quadrotor) acts as an eye in the sky and as leader that decides the UGVs formation shape (i.e. two-wheeled mobile robots). Two consecutive tasks are considered to be accomplished by the UGVs–UAV system. First, the UGVs will realize the deployment task, which consists of forming an initial free geometric configuration around the planar position of the quadrotor. In the second task, the UVGs will track a TVF shape while tracking the formation reference trajectory.

The main contributions of this paper are as follows:

- A novel deployment controller for the UGVs based on the kinematic model is proposed, where the nonholonomic constraints are considered. The key idea is in using the estimated implicit representation of the desired formation as a potential function to generate the UGVs reference trajectories.
- A novel robust cascaded velocity-torque controller based on the UGV kinematic and dynamic models is proposed to ensure the UGVs TVFs tracking. The key feature of the controller design is first in introducing a virtual auxiliary control input to control indirectly the actual UGVs velocity vector. Next, we added a switching term to the torque input, to compensate for the unknown external disturbance and the unmodeled dynamics.
- Also, we used the dynamic version of EFDs tool to model the motion of the desired UGVs formation shape. Thus, the dynamics of the formation is considered in the TVF tracking design.
- Finally, the validation of the proposed UGVs deployment and the UAV–UGVs TVF tracking controllers is carried out experimentally using three Festo's Robotino mobile robots, where the UAV is replaced by a virtual leader.

The rest of the paper is organized as follows. In Sec. 2, the EFDs formation modeling tool, the dynamical models of the mobile robot and quadrotor are introduced. Section 3 presents the UAV–UVGs system control structure, and Sec. 4 presents the design of the UGVs deployment control followed by the TVF tracking velocity-control design in Sec. 5. The UGVs formation torque-control design and the stability analysis are illustrated in Sec. 6, with some meaningful remarks. Then, the proposed formation controllers are validated through numerical simulation and experimental results in Sec. 7. Finally, the conclusion and perspectives are given in Sec. 8.

## 2. UAV–UGVs System Model

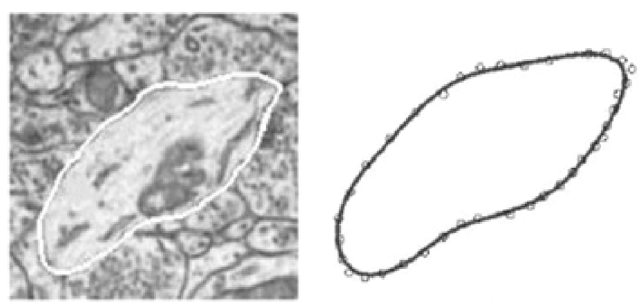
The control designing of a UGVs formation assisted by UAV is more challenging than the control of a homogeneous system because in the UAV–UGVs, each agent may have different dynamics and constraints. In this section, we highlight this challenge. First, it is described how to model parametrically a 2D time-varying formation shape using the EFDs and its corresponding implicit representation. Second, the dynamical model of the UAV (i.e. quadrotor) is presented, which will be considered as a leader of the ground mobile robots formation. The UAV decides the desired formation shape as well as the formation reference trajectory. Finally, the dynamical model of the UGV (i.e. two-wheeled mobile robot) is briefly presented.

### 2.1. *Parametric and implicit representation of planar curve*

The EFD is a mathematical tool originally introduced in [23]. It is used to define any planar free-form curve as a sum of ellipses. EFD operates as a transformation featured by translation, rotation, and scale. Biology and anatomy are the applications fields most relating to the use of EFDs (Fig. 1). The authors [20,21] show how to model a planar curve using the EFDs

$$\begin{cases} x(\delta) = a_0 + \sum_{k=1}^{n_h} (A_k \cos(k\delta) + B_k \sin(k\delta)), \\ y(\delta) = c_0 + \sum_{k=1}^{n_h} (C_k \cos(k\delta) + D_k \sin(k\delta)), \end{cases} \tag{1}$$

where $k$ is an index and $n_h$ denotes the number of harmonics used to represent the curve. $a_0$ and $c_0$ are the coordinates of



(a) (b)

Fig. 1. (a) Cell image, (b) reconstructed contour using EFDs [25].

February 11, 2022 2:21:17pm WSPC/284-US 2250006 ISSN: 2301-3850 2nd Reading



the curve center, and $A = [A_1, \ldots, A_{n_h}]^T \in \mathbb{R}^{n_h \times 1}$, $B \in \mathbb{R}^{n_h \times 1}$, $C \in \mathbb{R}^{n_h \times 1}$ and $D \in \mathbb{R}^{n_h \times 1}$ are vectors that represent the curve descriptors. The number of harmonics defines the representation accuracy, where the bigger is $n_h$, the more accurate representation the curve is $x(\delta)$ and $y(\delta)$ denote the coordinates of the points forming the curve contour. These coordinates are expressed as functions of a normalized parameter $\delta$ with $\delta \in [0, 2\pi]$. For example, the EFDs parameters of a circle are given as

$$[a_0 \ c_0]^T = [x_0 \ y_0]^T; A = R_c; B = 0; C = 0; D = R_c; n_h = 1,$$

where $(x_0, y_0)$ and $R_c$ are, respectively, the coordinates of the center of the circle and its radius.

It is further possible to estimate the EFDs vectors of any 2D closed curve defined by a set of points, with coordinates denoted by $(x_i, y_i), i \in [1, M_s]$. The EFDs vectors are estimated as in [21,24]

$$\begin{cases} A_k = \dfrac{1}{M_s}\displaystyle\sum_{i=1}^{M_s} x_i \cos(k\delta_i); & B_k = \dfrac{1}{M_s}\displaystyle\sum_{i=1}^{M_s} x_i \sin(k\delta_i), \\ C_k = \dfrac{1}{M_s}\displaystyle\sum_{i=1}^{M_s} y_i \cos(k\delta_i); & D_k = \dfrac{1}{M_s}\displaystyle\sum_{i=1}^{M_s} y_i \sin(k\delta_i), \\ a_0 = \dfrac{1}{M_s}\displaystyle\sum_{i=1}^{M_s} (x_i); & c_0 = \dfrac{1}{M_s}\displaystyle\sum_{i=1}^{M_s} (y_i); \ k \in \{1, n_h\}, \end{cases} \tag{2}$$

where $M_s$ is the number of points that define the curve and $\delta_i = i2\pi/M_s$. It is worth noting that the precision of estimation is dependent to the choice of the number of harmonics $n_h$.

To show how EFDs can be very useful for defining and modeling any 2D curve, Fig. 2 is depicted as an illustrative example, where the desired curve is defined in a dotted-line on an acquired image for surveillance purposes. This curve can be modeled using (2), and then reconstructed using (1).

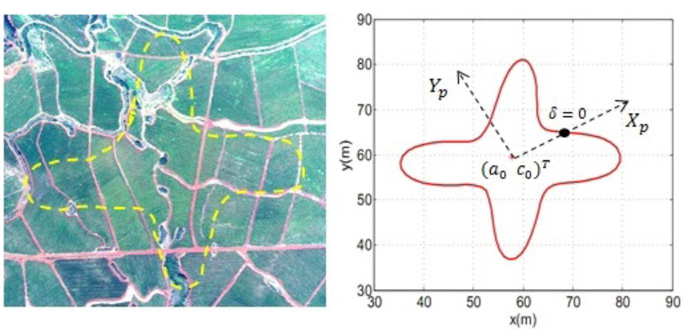



Fig. 2. An example of a 2D curve reconstructed after modeling using EFDs.

The authors in [21] proposed an approach to describe and to model the motion of a 2D free-form curve using dynamical EFDs as

$$\begin{aligned} x(\delta, t) &= a_0(t) + \sum_{k=1}^{n_h} (A_k(t)\cos(k\delta) + B_k(t)\sin(k\delta)), \\ y(\delta, t) &= c_0(t) + \sum_{k=1}^{n_h} (C_k(t)\cos(k\delta) + D_k(t)\sin(k\delta)), \end{aligned} \tag{3}$$

where $A \in \mathbb{R}^{n_h \times 1}$, $B(t)$, $C(t)$, and $D(t)$ are time-varying vectors and they are referred to as dynamical EFDs vectors and $a_0(t)$, $c_0(t)$, $x$, $y$, $\delta$, $n_h$ are already defined in (1). The parametric description (3) can be expressed as

$$\begin{bmatrix} x(\delta, t) \\ y(\delta, t) \end{bmatrix} = \begin{bmatrix} a_0(t) \\ c_0(t) \end{bmatrix} + \underline{\mathrm{EFDs}}(t), \tag{4}$$

where $\underline{\mathrm{EFDs}} \in \mathbb{R}^2$ is a relative offset vector relating the points of the curve contour with the curve center coordinates. Dynamical EFDs will be used to model the time-varying formation shape of a group of robots.

Further to the parametric representation (3), 2D curves could also be represented using an implicit polynomial function (IPF) denoted by $\mathcal{H}(x, y) = 0$. This implicit function of a closed curve is derived by the implicitization of the EFDs vectors. In this study, the IPF is obtained using the method detailed in [25]. The IPF that represents a closed curve modeled by EFDs takes the form

$$\mathcal{H}(x, y) = \sum_{0 \le i+j \le d_p} a_{ij} x^i y^j = 0, \tag{5}$$

where $a_{ij}$ are coefficients and $d_p$ is the polynomial degree with $d_p = 2n_h$ [25]. For example, the IPF of a circle is defined as $\mathcal{H}(x, y) = (x - x_0)^2 + (y - y_0)^2 - R_c^2$, where $(x_0, y_0)$ and $R_c$ are, respectively, its center coordinates and radius. The IPF of the formation shape can be considered as a potential fields function that could have control purposes.

### 2.2. *Dynamical Model of a Quadrotor UAV*

In this section, the dynamical model of a quadrotor used as the UAV is presented. The latter is considered as a rigid body. Let $\mathcal{F}_w = (O_W, X_W, Y_W, Z_W)$ be the global inertial frame, and let $\mathcal{F}_q = (O_q, X_q, Y_q, Z_q)$ be the body-fixed frame. Let $\eta = [\varphi_q, \theta_q, \psi_q]^T$ describe the orientation of the quadrotor (Euler angles) and $\chi_q = [x_q, y_q, z_q]^T$ denotes the position of its mass center with respect to the inertial frame.

A brief explanation of the classic process is given in order to derive the simplified dynamical model [26,27]]. The structure and the propellers are rigid and symmetric. The translational and rotational dynamics are expressed as

$$\begin{aligned} M_q \ddot{\chi}_q &= -M_q g \underline{e}_z + u_z \mathcal{R}_q \underline{e}_z, \\ I_q \dot{\bar{w}}_B &= -\bar{w}_B \times I_q \bar{w}_B + G_a + \tau_q, \end{aligned} \tag{6}$$

where $M_q$, $g$ and $u_z$ are the quadrotor mass, the gravitational acceleration, and the total thrust force developed by the four propellers, respectively. $\underline{e}_z = (0,0,1)^T$ denotes the $Z_W$-axis unit vector, and $\mathcal{R}_q$ is a rotation matrix relating the inertial frame $\mathcal{F}_w$ with the body-fixed frame $\mathcal{F}_q$. The vector $\bar{w}_B = [p,q,r]^T$ denotes the angular quadrotor rates measured in body frame $\mathcal{F}_q$, and $I_q = \mathrm{diag}(I_x, I_y, I_z)$ is the quadrotor's diagonal inertia matrix. $\tau_q = [u_\phi, u_\theta, u_\psi]^T$ are moments due to propellers forces acting on the quadrotor along the $x$, $y$ and $z$ body-fixed frame axes. $G_a = [J_r\dot{\theta}_q\Omega_r, J_r\dot{\phi}_q\Omega_r, 0]^T$ represents the gyroscopic moments due to the rotors' inertia $J_r$ with $\Omega_r$ being a mixture of propellers angular speeds $\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$ [26].

The angular quadrotor rates $\bar{w}_B$ are related to Euler angular rates $\dot{\eta}$ by the following relation:

$$\dot{\eta} = \begin{bmatrix} 1 & S_{\phi_q}\tan\theta_q & C_{\phi_q}\tan\theta_q \\ 0 & C_{\phi_q} & -S_{\phi_q} \\ 0 & S_{\phi_q}/C_{\theta_q} & C_{\phi_q}/C_{\theta_q} \end{bmatrix} \bar{w}_B, \tag{7}$$

where $S_{(\cdot)}$ and $C_{(\cdot)}$ are abbreviations for $\sin(\cdot)$ and $\cos(\cdot)$, respectively. The dynamical model of the quadrotor can be written using (6) and (7) as follows:

$$\ddot{\chi}_q = \begin{bmatrix} \ddot{x}_q \\ \ddot{y}_q \\ \ddot{z}_q \end{bmatrix} = \begin{bmatrix} u_z \dfrac{u_x}{M_q} \\ u_z \dfrac{u_y}{M_q} \\ -g + u_z \dfrac{C_{\theta_q} C_{\phi_q}}{M_q} \end{bmatrix}, \tag{8}$$

$$\ddot{\eta} = \begin{bmatrix} \ddot{\varphi}_q \\ \ddot{\theta}_q \\ \ddot{\psi}_q \end{bmatrix} = \begin{bmatrix} \dot{\theta}_q\dot{\psi}_q\left(\dfrac{I_y - I_z}{I_x}\right) - \dfrac{J_r\dot{\theta}_q\Omega_r}{I_x} + \dfrac{u_\phi}{I_x} \\ \dot{\phi}_q\dot{\psi}_q\left(\dfrac{I_z - I_x}{I_y}\right) + \dfrac{J_r\dot{\phi}_q\Omega_r}{I_y} + \dfrac{u_\theta}{I_y} \\ \dot{\phi}_q\dot{\theta}_q\left(\dfrac{I_x - I_y}{I_z}\right) + \dfrac{u_\psi}{I_z} \end{bmatrix}. \tag{9}$$

In (8), $u_x$ and $u_y$ are considered as virtual control inputs for $x_q$ and $y_q$ states, defined as [27]

$$u_x = C_{\psi_q}S_{\theta_q}C_{\phi_q} + S_{\psi_q}S_{\phi_q}; u_y = S_{\psi_q}S_{\theta_q}C_{\phi_q} - C_{\psi_q}S_{\phi_q}.$$

The quadrotor control inputs $u_\phi, u_\theta, u_\psi$ and $u_z$ are defined as

$$u_\phi = b\ell[\Omega_3^2 - \Omega_4^2]; u_\theta = b\ell[\Omega_1^2 - \Omega_2^2],$$

$$u_\psi = \sigma[\Omega_1^2 + \Omega_2^2 - \Omega_3^2 - \Omega_4^2]; u_z = \sum_{i=1}^{4} b\Omega_i^2,$$

where $\Omega_i, i = [1,4]$ is the $i$th propeller angular velocity, $\ell$ is the distance between the motor and the quadrotor center, $b$ and $\sigma$ are the propeller aerodynamic lift and drag coefficients, respectively.

## 2.3. *Dynamical Model of a Mobile robot UGV*

A group of $N_R$ homogeneous UGVs (i.e. nonholonomic two-wheeled mobile robots) is considered, where the generalized coordinates of each mobile robot are given by

$$q_j = [x_j, y_j, \theta_j]^T, \tag{10}$$

where $x_j$, $y_j$, $\theta_j$ are, respectively, the $x$, $y$ coordinates, and the orientation of the $j$th robot. In Fig. 3, the point $c$ is center of mass of the mobile robot.

The kinematic model of the mobile robot $j$ is written as [28,30]

$$\dot{q}_j = \begin{bmatrix} \dot{x}_j \\ \dot{y}_j \\ \dot{\theta}_j \end{bmatrix} = S^T \underline{v}_j = \begin{bmatrix} \cos(\theta_j) & -d_j\sin(\theta_j) \\ \sin(\theta_j) & d_j\cos(\theta_j) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_j \\ w_j \end{bmatrix}, \tag{11}$$

where $d_j$ is the distance from the rear axle to the mass center of robot $j$, $L_w$ is the half distance between the left and right wheels, $R_w$ is the robot wheel radius, $\underline{v}_j = [v_j \ w_j]^T$ with $v_j$ and $w_j$ are, respectively, the linear and the angular velocities of the $j$th mobile robot.

A Nonholonomic mobile robot characterized by $n$ generalized coordinates $(q_{j1}, \ldots, q_{jn})$, having $r$ control inputs and subject to $m$ constraints described in detail in [29,30], and mathematically after applying the transformation described in [29,30] to eliminate Lagrange multipliers yields the alternative dynamical model

$$\bar{M}_j(q_j)\dot{\underline{v}}_j + \bar{V}_{mj}(q_j, \dot{q}_j)\underline{v}_j + \bar{F}_j(\dot{q}_j) + \bar{\tau}_{dj} = \bar{B}_j\underline{\tau}_j, \tag{12}$$

where $\bar{M} \in \mathbb{R}^{r\times r}$ is a symmetric positive definite inertia matrix, $\bar{V}_{mj} \in \mathbb{R}^{r\times r}$ is the centripetal and coriolis matrix, $\bar{F}_j \in \mathbb{R}^{r\times 1}$ is the friction vector, $\bar{\tau}_{dj} \in \mathbb{R}^{r\times 1}$ represents an unknown bounded disturbances including unstructured unmodeled dynamics, and $\bar{B}_j \in \mathbb{R}^{r\times r}$ is the input matrix and $\underline{v}_j \in \mathbb{R}^{r\times 1}$, $\underline{\tau}_j \in \mathbb{R}^{r\times 1}$ are, respectively, the velocity and the torque vectors [29,30]. For the nonholonomic mobile robot

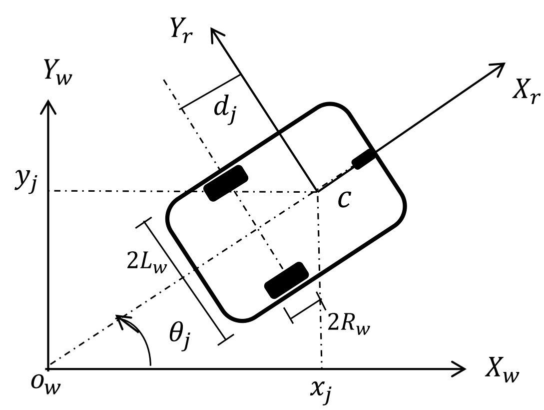



Fig. 3. Frames and parameters of a nonholonomic mobile robot.

described in Fig. 3, the complete dynamical model is expressed as follows [29,30]:

$$\left(m+\frac{2I_w}{R_w^2}\right)\dot{v}_j - m_c d_j (w_j)^2 + \bar{F}_{j,1} + \bar{\tau}_{dj,1} = \frac{(\tau_{R,j}+\tau_{L,j})}{R_w},$$
$$\left(I+\frac{2L_w^2}{R_w^2}I_w\right)\dot{w}_j + m_c d_j v_j w_j + \bar{F}_{j,2} + \bar{\tau}_{dj,2} = \frac{L_w(\tau_{R,j}-\tau_{L,j})}{R_w}, \tag{13}$$

where $m = m_c + 2m_w$ is the total mass of the mobile robot, $I = (I_c + m_c d_j^2 + 2m_w L_w^2 + 2I_m)$ is the total equivalent inertia, $m_c$ is the robot mass without the driving wheels and actuators (DC motors), $m_w$ is the mass of each driving wheel (with actuator), $I_c$ is the moment of inertia of the robot about the vertical axis through the center of mass, $I_w$ and $I_m$ are the moments of inertia of each driving wheel (with actuator) around the wheel axis, and the moment of inertia of each driving wheel with a motor about the wheel diameter, respectively. The matrices $\bar{M}_j$, $\bar{V}_{mj}$ and $\bar{B}_j$ in (12) are obtained from (13) as follows:

$$\bar{M}_j = \begin{bmatrix} m+\frac{2I_w}{R_w^2} & 0 \\ 0 & I+\frac{2L_w^2}{R_w^2}I_w \end{bmatrix},$$
$$\bar{V}_{mj} = \begin{bmatrix} 0 & -m_c d_j \dot{\theta}_j \\ m_c d_j \dot{\theta}_j & 0 \end{bmatrix}; \quad \bar{B}_j = \frac{1}{R_w}\begin{bmatrix} 1 & 1 \\ L_w & -L_w \end{bmatrix}.$$

**Remark 2.1.** Common to robotic systems, the *skew-symmetric property* that is defined as, $X^T(\dot{\bar{M}}_j - 2\bar{V}_{mj})X = 0$ for all vector $X$ [31], is an important feature that will be used later in the stability analysis section.

The complete mobile robot behavior can be described by (11) and (12). Let us define the nonlinear feedback control input as

$$\underline{\tau}_j = \bar{B}_j^{-1}[\bar{M}_j \underline{u}_j + \bar{V}_{mj}\underline{v}_j + \bar{F}_j], \tag{14}$$

where $\underline{u}_j \in \mathbb{R}^{2\times 1}$ is an auxiliary input. Applying the control law (14) to (12) allows the conversion of the dynamical control problem into the kinematic control problem such that

$$\begin{cases} \dot{q}_j = S^T \underline{v}_j, \\ \dot{\underline{v}}_j = \underline{u}_j, \end{cases} \tag{15}$$

where the matrix $S^T$ is defined in (11). The feedback control input (14) yields a form of cascaded kinematic and dynamic linearization structure. Thus, the alternative model (15) is convenient for the purpose of control.

## 3. The UAV–UGVs Control Structure

As depicted in Fig. 4, the MRS control scheme consists of two consecutive tasks. Accordingly, two controllers have been proposed, namely, deployment controller and TVF tracking controller.

The first task is UGVs deployment control, which consists of converging a group of mobile robots towards an initial formation shape (i.e. 2D curve). The latter is modeled by EFDs and its corresponding IPF. The quadrotor will be located at the center of the formation, being the leader that produces the formation shape. This scenario has a number of applications such as target enclosing and fire-monitoring.

In the second task (see Fig. 4), once the initial formation shape is completed, the quadrotor will track a predefined reference trajectory, while each mobile robot will generate

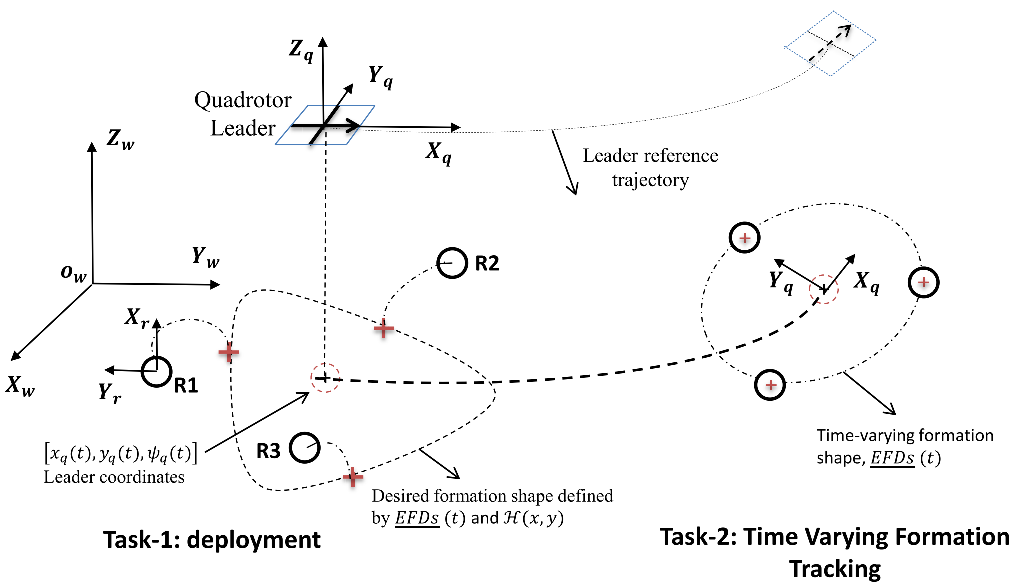



Fig. 4. Mobile robots formation control structure consisting of two tasks.

and tack its own trajectory and simultaneously maintains a desired time-varying formation shape. The desired shape is produced by the quadrotor depending on the mission requirements, circumstances, and the changes occurring in the operating environment. For example, the quadrotor can use an onboard camera sensor to acquire images from the operating environment, and then process these images to define the adequate formation shape. Both the quadrotor path planning and the process of the formation shape selection are not considered in this paper. Thus, the EFDs vectors relating to the desired formation shape and the quadrotor reference trajectory are predefined.

**Remark 3.1.** The control of single mobile robot can be developed by considering only the kinematic model (11), resulting in a pseudo velocity command input that we denote $\underline{v}_c(t) = [v_c(t)\ w_c(t)]^T$. However, the control performance of the mobile robot can be improved by incorporating its dynamics on the controller design. Consequently, a control torque $\underline{\tau}_j(t)$ can be designed to make the actual velocity $\underline{v}(t)$ follows the designed pseudo velocity command input $\underline{v}_c(t)$. Thus, in this paper, we seek to expand the framework developed for the control of a single mobile robot to the control of mobile robots formation-based Leader-Followers approach, where perfect velocity tracking assumption is removed. Therefore, we seek to develop a robust formation torque-controller to make the actual velocity follows the pseudo-velocity command $\underline{v}_c(t)$ in such a way the formation desired behavior is improved in terms of flexibility, tracking accuracy, and robustness, in contrast with the case when perfect velocity assumption is assumed.

## 4. UGVs Deployment Control (Kinematic-Based Approach)

The control aim in this task is to find a smooth velocity control input $\underline{v}_{jc}(t)$, $j \in [1, N_R]$ in such a way all the mobile robots converge to the initial desired formation shape. Unlike the classical frameworks on formation tracking that consider particular shapes such as circle, ellipse, and square. In this work, we propose a novel UGVs deployment controller, in which the UGVs can reach any free formation shape. As a result, the proposed UGVs deployment controller permits an efficient and flexible UGVs navigation, particularly in dynamic and unstructured environments. In addition, the method is suitable for real applications of UAV–UGVs systems, as the UAV can provide aerial images from which, the desired formation shape can be extracted.

The key idea is first to estimate the EFDs vectors of the formation shape. Next, we deduce the corresponding implicit polynomial function (IPF) denoted by $\mathcal{H}(x,y)$. The latter will be used to generate the UGVs reference trajectory leading to the desired formation. As detailed in Sec. 2.1, the position error function between the $j$th mobile robot and the formation shape (i.e. 2D closed curve) is expressed using the IPF as

$$e^j_{\mathrm{Form}} = \mathcal{H}(x_j, y_j), \tag{16}$$

where $(x_j, y_j)$ are the $j$th robot $x$ and $y$ coordinates, $e^j_{\mathrm{Form}}$ is an algebraic distance between the $j$th robot and the desired formation shape and $\mathcal{H}(x,y)$ is the IPF of the desired shape. By using (16), when the robot is exactly on the curve, the error is zero. While, the error is positive or negative if the robot is outside or inside the closed curve, respectively. We consider the reference trajectory for the $j$th mobile robot to be defined as

$$\dot{x}_{jr} = v_{jr}\cos(\theta_{jr}); \quad \dot{y}_{jr} = v_{jr}\sin(\theta_{jr}); \quad \dot{\theta}_{jr} = w_{jr}, \tag{17}$$

where $x_{jr}$, $y_{jr}$ and $\theta_{jr}$ are the positions and orientation of a virtual reference robot, mobile robot $j$ seeks to follow. $v_{jr}(t)$ and $w_{jr}(t)$ are, respectively, its reference linear and angular velocities. If $x_{jr}$, $y_{jr}$ and $\theta_{jr}$ are continuously differentiable and bounded when $t \to \infty$, and $(\dot{x}_{jr} + \dot{y}^2_{jr}) \neq 0$ it can be shown that

$$\begin{pmatrix} v_{jr} \\ w_{jr} \\ \theta_{jr} \end{pmatrix} = \begin{pmatrix} \dot{x}_{jr}\cos(\theta_{jr}) + \dot{y}_{jr}\sin(\theta_{jr}) \\ (\ddot{y}_{jr}\dot{x}_{jr} - \ddot{x}_{jr}\dot{y}_{jr})/(\dot{x}^2_{jr} + \dot{y}^2_{jr}) \\ \tan^{-1}(\dot{y}_{jr}/\dot{x}_{jr}) \end{pmatrix}. \tag{18}$$

The reference trajectory of the mobile robot $j$ is derived from (17) and (18) during the deployment stage, as is depicted in Fig. 5.

Refer to Fig. 5, in the first step, the mobile robot is approached to point-mass particle and exhibits the kinematic model given as, $[\dot{x}_j\ \dot{y}_j]^T = F_{a,j}$; $F_{a,j} = [F_{xj}\ F_{yj}]^T$, where $F_{xj}$ and $F_{yj}$ are the components of a virtual force. The 2D desired curve described by the IPF produces an external potential field (see Fig. 7) yielding an attractive force $F_{a,j}$ being exerted on the robot. By integrating the virtual force, the robot reference position $(x_{jr}, y_{jr})$ is obtained and smoothed out using a low-pass filter. Then, Eq. (18) is used to calculate the corresponding reference velocities and orientation of the virtual reference robot.

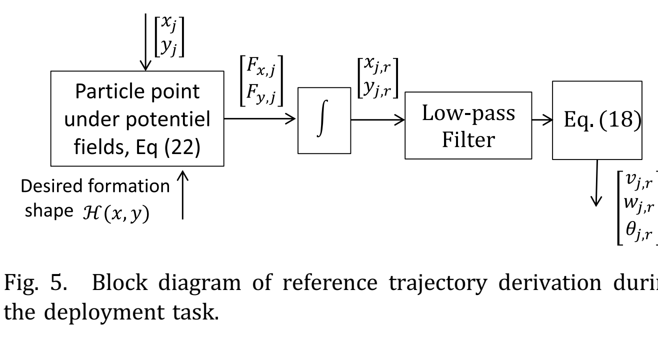



Fig. 5. Block diagram of reference trajectory derivation during the deployment task.

February 11, 2022 2:21:26pm WSPC/284-US 2250006 ISSN: 2301-3850 2nd Reading



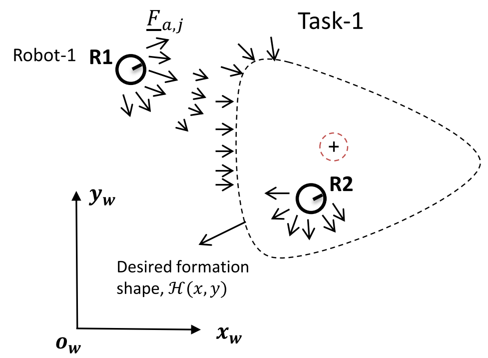



Fig. 6. Virtual attractive force exerted by the 2D curve on the mobile robots.

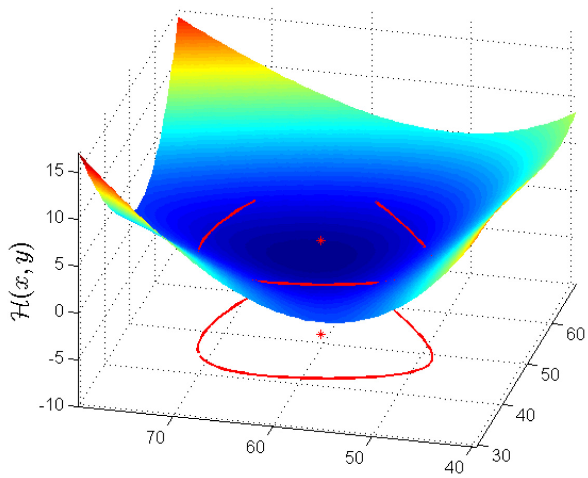



Fig. 7. 3D plot of the IPF, $\mathcal{H}(x,y)$.

Now, in the following, we explain how to derive the virtual force $F_{a,j}$ exerted by the 2D curve on the $j$th robot relying on Lyapunov theory. The attractive force will be designed to force the error (16) to exponentially decrease

$$\dot{e}^{j}_{\text{Form}} = -\lambda_1 e^{j}_{\text{Form}}, \tag{19}$$

where $\lambda_1$ is a positive number. Substituting (16) into (19) yields

$$\dot{\mathcal{H}}(x_j,y_j) = -\lambda_1 \mathcal{H}(x_j,y_j). \tag{20}$$

After deriving (16) with respect to time and substituting the particle model $(\dot{x}_j,\dot{y}_j)^T = (F_{x,j},F_{y,j})^T$ yields

$$(\mathcal{H}_x\ \mathcal{H}_y)\begin{pmatrix} F_{x,j} \\ F_{y,j} \end{pmatrix} = -\lambda_1 \mathcal{H}(x_j,y_j). \tag{21}$$

The virtual force $F_{a,j} = [F_{x,j},F_{y,j}]^T$ is determined from (21) for the mobile robot $j$ relying on pseudo inverse operator

$$\begin{pmatrix} F_{x,j} \\ F_{y,j} \end{pmatrix} = -\lambda_1 \frac{1}{\|\nabla\mathcal{H}(x_j,y_j)\|^2}\mathcal{H}(x_j,y_j)\begin{pmatrix} \mathcal{H}_x(x_j,y_j) \\ \mathcal{H}_y(x_j,y_j) \end{pmatrix}, \tag{22}$$

where $\mathcal{H}_x$ and $\mathcal{H}_y$ are the partial derivatives of $\mathcal{H}(x,y)$ with respect to $x$ and $y$, respectively, and $\nabla\mathcal{H} = [\mathcal{H}_x\ \mathcal{H}_y]^T$.

Once the reference trajectories of the mobile robots are derived, namely $(x_{jr},y_{jr},\theta_{jr},v_{jr},w_{jr}),\ j \in (1,N_R)$, the tracking error system $\underline{e}_j = [e^x_j,e^y_j,e^\theta_j]^T$ of the $j$th mobile robot can be given and expressed in the robot coordinate frame as

$$\begin{bmatrix} e^x_j \\ e^y_j \\ e^\theta_j \end{bmatrix} = \begin{bmatrix} \cos(\theta_j) & -\sin(\theta_j) & 0 \\ \sin(\theta_j) & \cos(\theta_j) & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x_{jr}-x_j \\ y_{jr}-y_j \\ \theta_{jr}-\theta_j \end{bmatrix}. \tag{23}$$

Thus, the problem of controlling a mobile robots formation may be expressed as the requirement to make the error dynamics of (23) globally asymptotically stable. By developing the first-order time derivative along (23), and using Lyapunov theory, a smooth velocity-control input $\underline{v}_{jc}(t)$ that guarantees the asymptotic stability can be selected as [28]

$$\begin{bmatrix} v_{jc} \\ w_{jc} \end{bmatrix} = \begin{bmatrix} v_{jr}\cos(e^\theta_j) + K_x e^x_j \\ w_{jr} + (v_{jr}+K_v)K_y e^y_j + (v_{jr}+K_v)K_\theta \sin(e^\theta_j) \end{bmatrix}, \tag{24}$$

where $K_x$, $K_y$, $K_v$ and $K_\theta$ are properly selected positive gains. Comparing the velocity controller proposed in [28] (Eq. (21)) with (24), we find that the term $K_v > 0$ is added to $v_{jr}$, thus, we ensure that the asymptotic stability holds even when $v_{jr} = 0$.

**Remark 4.1.** The followed method to generate the reference mobile robots trajectory leading to guarantee $\mathcal{H}(x_j,y_j) \to 0, j \in (1,N_R)$, as $t \to \infty$ is different from the classic Artificial Potential Fields (APFs) method, in the sense the controller is designed to converge the mobile robots to a 2D closed curve rather than to converge them to a goal point. Thus, the potential applications could be 2D obstacle avoidance, where the avoided obstacle curve can have any shape rather than being a circle or an ellipse as presented in many previous works on obstacle avoidance based behavioral approaches [10]. Thus, the mobile robots navigation space could be increased.

## 5. UGVs Time-Varying Formation Tracking Control (Task-2)

During the deployment task, the initial desired formation shape is reached by the mobile robots using (24), where the leader (i.e. quadrotor) is located in the center of the formation $[x_q\ y_q]^T = [a_0\ c_0]^T$. In the second task, the aim is to design a TVF tracking controller enabling the mobile robots to maintain the desired 2D curve, while tracking the formation reference trajectory (see Fig. 4).

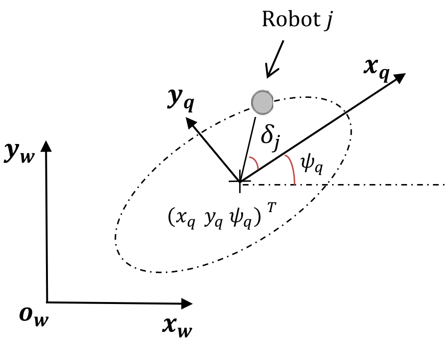



Fig. 8. Location of the mobile robot $j$ prior to start Task-2.

First, we explain the process of generating the mobile robots reference trajectory relying on the knowledge of the desired TVF shape (i.e. EFDs vectors) and the leader's planar coordinates. Depending on the location of each robot on the desired curve, each mobile robot $j$ forms an angle $\delta_j$ relative to the $x$-axis of the quadrotor coordinate frame (Fig. 8).

The EFDs dynamical representation (4) is used to define the desired position of the mobile robots within the time-varying desired curve. The reference trajectory of the $j$th mobile robot, $q_{jr} = [x_{jr}, y_{jr}, \theta_{jr}]^T$ comes from the leader's output position $[x_q, y_q]^T$, yaw angle $\psi_q$ and the desired time-varying EFDs,

$$q_{jr}(t) = \begin{bmatrix} x_q(t) \\ y_q(t) \\ \theta_{jr}(t) \end{bmatrix} + R_q^w \begin{bmatrix} \underline{\mathrm{EFDs}}(\delta_j, t) \\ 0 \end{bmatrix}, \tag{25}$$

where $\theta_{jr} = \tan^{-1}(\dot{y}_{jr}, \dot{x}_{jr})$, $R_q^w(t)$ is a rotational matrix relating the quadrotor $x$–$y$ coordinate frame to the world $x$–$y$ coordinate frame (see Fig. 8)

$$R_q^w(t) = \begin{bmatrix} \cos(\psi_q) & -\sin(\psi_q) & 0 \\ \sin(\psi_q) & \cos(\psi_q) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{26}$$

From (4), we have $\underline{\mathrm{EFDs}}(\delta_j, t) = [\overline{AB}_j\ \overline{CD}_j]^T \in \mathbb{R}^{2\times 1}$ denotes the desired relative offset vector of the $j$th robot with respect to the leader coordinates given as

$$\underline{\mathrm{EFDs}}(\delta_j, t) = \begin{cases} \displaystyle\sum_{k=1}^{n_h} (A_k(t)\cos(k\delta_j) + B_k(t)\sin(k\delta_j)), \\ \displaystyle\sum_{k=1}^{n_h} (C_k(t)\cos(k\delta_j) + D_k(t)\sin(k\delta_j)), \end{cases} \tag{27}$$

where $A(t) \in \mathbb{R}^{n_h \times 1}$, $B(t)$, $C(t)$ and $D(t)$ are the desired time-varying EFDs vectors owing the same size. The desired orientation of each robot is defined in such a way that the mobile robot is always facing its desired position.

Now, in the following, we explain the design of the TVF tracking controller. In a single mobile robot control, a steering control input $\underline{v}_{jc}(t)$ is designed to solve three basic problems: path following, point stabilization, and trajectory following such that $\lim_{t\to\infty}(q_{jr} - q_j) = 0$ and $\lim_{t\to\infty}(\underline{v}_{jc} - \underline{v}_j) = 0$ [28]. If the mobile robot controller can successfully track a class of smooth pseudo-velocity command, then all three problems can be solved with the same controller. We will extend the stated three basic tracking control problems for a single mobile robot to the proposed formation control based leader-followers model.

**Definition 5.1.** The mobile robots formation system is said to achieve the output TVF tracking control if and only if for any given initial states $q_j(t=0)$, $\underline{v}_j(t=0)$, $j \in [1, N_R]$ there exists a control torque $\underline{\tau}_j$, such as

$$\begin{cases} \lim\limits_{t\to\infty} \|q_j(t) - q_{jr}(t)\| = 0, \\ \lim\limits_{t\to\infty} \|\underline{v}_j(t) - \underline{v}_{jc}(t)\| = 0, \end{cases} \quad j \in [1, N_R]. \tag{28}$$

In this section, the aim is limited on finding a smooth pseudo-velocity command $\underline{v}_{jc} = f(e_j, \underline{EFDs}(\delta_j, t), \dot{\chi}_q, \dot{\eta}_q, j)$, in such a way that $\lim_{t\to\infty}(q_j(t) - q_{jr}(t)) = 0, j = [1, 2, \ldots, N_R]$. In Sec. 6, we focus on designing a robust torque-control input $\underline{\tau}_j$ for each mobile robot with dynamic behavior described by (11) and (12), so that $\lim_{t\to\infty}(\underline{v}_j(t) - \underline{v}_{jc}(t)) = 0$. Achieving this for all the mobile robots $j = 1, 2, N_R$ guarantees that the mobile robots are tracking the formation reference trajectory, simultaneously, the formation is maintained and stabilized around the quadrotor planar coordinates $(x_q, y_q)$.

The formation time-varying tracking error expressed in the robot coordinate frame is written as in (23). The error dynamics system is determined by differentiating (23), substituting (11), (25), and using trigonometric relations yields (see (29)).

$$\begin{aligned} \dot{e}_j^x &= -v_j + v_q\cos(\theta_j - \rho_q) + w_j e_j^y \\ &\quad - \dot{\psi}_q\sqrt{(\overline{AB}_j)^2 + (\overline{CD}_j)^2}\,\sin(e_j^\theta + \psi_q + \beta_1) \\ &\quad + \sqrt{(\dot{\overline{AB}}_j)^2 + (\dot{\overline{CD}}_j)^2}\,\cos(e_j^{\theta_j,\psi_q} + \beta_2), \\ \dot{e}_j^y &= -w_j e_j^x + v_q\sin(\theta_j - \rho_q) + d_j w_j \\ &\quad + \dot{\psi}_q\sqrt{(\overline{AB}_j)^2 + (\overline{CD}_j)^2}\,\cos(e_j^\theta + \psi_q + \beta_1) \\ &\quad + \sqrt{(\dot{\overline{AB}}_j)^2 + (\dot{\overline{CD}}_j)^2}\,\sin(e_j^{\theta_j,\psi_q} + \beta_2), \\ \dot{e}_j^\theta &= \dot{\theta}_{jr} - w_j, \end{aligned} \tag{29}$$

In (29), $v_q = \sqrt{\dot{x}_q + \dot{y}_q}$, $\rho_q = \tan^{-1}(\dot{y}_q, \dot{x}_q)$ and $e_j^{\theta_j,\psi_q} = (\theta_j - \psi_q)$. Furthermore $(\dot{\overline{AB}}_j)$ and $(\dot{\overline{CD}}_j)$ are the derivatives of $(\overline{AB}_j)$ and $(\overline{CD}_j)$. The parameters $\beta_1$, $\beta_2$ are defined as

$$\begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} \tan^{-1}(\overline{CD}_j, \overline{AB}_j) \\ \tan^{-1}(\dot{\overline{CD}}_j, \dot{\overline{AB}}_j) \end{bmatrix}. \tag{30}$$

To stabilize the dynamics error (29), the pseudo-velocity command input (31)–(33) is proposed for mobile robot

$j, j = [1, N_R]$ to track the desired time-varying formation with respect to the quadrotor coordinates and yaw angle $(x_q, y_q, \psi_q)$

$$\begin{cases} v_{jc} = v_q \cos\left(\theta_j - \rho_q\right) + K_x e_j^x + \xi_{vj}, \\ w_{jc} = \begin{cases} \dot{\theta}_{jr} + \left(v_q + K_v\right) K_y e_j^y, \\ \quad + \left(v_q + K_v\right) K_\theta \sin\left(e_j^\theta\right) + \xi_{wj}, \end{cases} \end{cases} \tag{31}$$

where $\xi_{vj}$ and $\xi_{wj}$ are given as

$$\xi_{vj} = \begin{cases} -\dot{\psi}_q \sqrt{\overline{AB}_j^2 + \overline{CD}_j^2} \sin(e_j^\theta + \psi_q + \beta_1), \\ +\sqrt{(\dot{\overline{AB}}_j)^2 + (\dot{\overline{CD}}_j)^2} \sin(e_j^\theta + \psi_q + \beta_2), \end{cases} \tag{32}$$

$$\xi_{wj} = -\frac{|e_j^y|}{1/K_y + |e_j^y| d_j} [\dot{\psi}_q (d_j + \sqrt{\overline{\mathrm{AB}}_j^2 + \overline{\mathrm{CD}}_j^2}) + (v_q + K_v) K_\theta d_j + \sqrt{(\dot{\overline{AB}}_j)^2 + (\dot{\overline{CD}}_j)^2} + K_v]. \tag{33}$$

**Remark 5.1.** If we compare the proposed formation controller (31) with the robots deployment controller (24), we find that (32) and (33) are new terms. These new terms guarantee the stability of the overall mobile robots formation. Furthermore, the parameter $K_v > 0$ is introduced to ensure that the asymptotic stability holds even when the quadrotor velocity $v_q = 0$.

Before we proceed the stability analysis, the following assumptions are needed:

**Assumption 1.** Each mobile robot in the formation is wirelessly connected to the quadrotor, whose control inputs $u_z$ and $\tau_q$ are bounded $\forall t > 0$.

**Assumption 2.** The desired time-varying formation defined by the EFDs vectors (i.e. $A(t)$, $B(t)$, $C(t)$ and $D(t)$) as well as the leader's state and its control inputs are communicated to the whole formation mobile robots.

**Assumption 3.** The vectors $A(t)$, $B(t)$, $C(t)$ and $D(t)$ used in (27) are continuous, differentiable, and bounded. Furthermore, $v_q(t)$ and $\dot{\psi}_q(t)$ are bounded $\forall t > 0$.

**Assumption 4.** The perfect velocity tracking holds such as $\underline{v}_j = \underline{v}_{jc}$ (this assumption will be removed later).

**Theorem 5.1.** *Let a formation of $N_R$ nonholonomic mobile robots with dynamics described by (11) and (12), owning n generalized coordinates $q_j$, r actuators, and m independent constraints, guided by a quadrotor satisfying (28). Let Assumption $1 - 4$ hold. The jth mobile robot will be steered by a smooth pseudo-velocity command $\underline{v}_{jc}$ given by (31)–(33). Then, the origin $\underline{e}_j = 0$ consisting of the position and orientation error for the mobile robots formation $j = 1, \ldots, N_R$ are asymptotically stable.*

**Proof.** Consider the following candidate Lyapunov function:

$$\mathbb{Q}_j = \frac{1}{2}((e_j^x)^2 + (e_j^y)^2) + \frac{1 - \cos(e_j^\theta)}{K_y}. \tag{34}$$

We can notice that $\mathbb{Q}_j \geq 0$ and $\mathbb{Q}_j = 0$ only if $\underline{e}_j = 0$. The time derivative of (34) with the substitution of (31), (32), yields

$$\begin{aligned} \dot{\mathbb{Q}}_j = & -K_x (e_j^x)^2 - d_j K_y (v_q + K_v)(e_j^y)^2 \\ & - (v_q + K_v) \frac{K_\theta}{K_y} \sin^2(e_j^\theta) \\ & - d_j e_j^y \begin{bmatrix} \dot{\psi}_q + (v_q + K_v) K_\theta \sin(e_j^\theta) \\ -\dfrac{\dot{\psi}_q}{d_j} \sqrt{\overline{AB}_j^2 + \overline{CD}_j^2} \cos(e_j^\theta + \psi_q + \beta_2) \\ -\dfrac{\sqrt{(\dot{\overline{AB}}_j)^2 + (\dot{\overline{CD}}_j)^2}}{d_j} \sin(e_j^\theta + \psi_q + \beta_2) \\ + \dfrac{K_v \sin(e_j^\theta)}{d_j} \end{bmatrix} \\ & - \xi_{wj} \left[\frac{\sin(e_j^\theta)}{K_y} + e_j^y d_j\right]. \end{aligned} \tag{35}$$

Equation (35) can be expressed as an inequality

$$\begin{aligned} \dot{\mathbb{Q}}_j \leq & -K_x (e_j^x)^2 - d_j K_y (v_q + K_v)(e_j^y)^2 \\ & - (v_q + K_v) \frac{K_\theta}{K_y} \sin^2(e_j^\theta) \\ & + |e_j^y| \begin{pmatrix} \dot{\psi}_q (d_j + \sqrt{\overline{AB}_j^2 + \overline{CD}_j^2}) + (v_q + K_v) K_\theta d_j \\ + \sqrt{(\dot{\overline{AB}}_j)^2 + (\dot{\overline{CD}}_j)^2} + K_v \end{pmatrix} \\ & + \xi_{wj} \left[\frac{1}{K_y} + |e_j^y| d_j\right]. \end{aligned} \tag{36}$$

In (36), it is noticeable that the first three terms are negative whatever $\underline{e}_j \neq 0$. Moreover, substituting (33) into the terms remaining in (36) yields

$$\begin{aligned} & |e_j^y| \begin{bmatrix} \dot{\psi}_q (d_j + \sqrt{\overline{AB}_j^2 + \overline{CD}_j^2}) + (v_q + K_v) K_\theta d_j \\ + \sqrt{(\dot{\overline{AB}}_j)^2 + (\dot{\overline{CD}}_j)^2} + K_v \end{bmatrix} \\ & + \xi_{wj} \left[\frac{1}{K_y} + |e_j^y| d_j\right] \leq 0. \end{aligned} \tag{37}$$

Now, substituting (37) into (36) yields

$$\dot{\mathbb{Q}}_j \leq \begin{bmatrix} -K_x (e_j^x)^2 - d_j K_y (v_q + K_v)(e_j^y)^2 \\ -(v_q + K_v) \dfrac{K_\theta}{K_y} \sin^2(e_j^\theta) \end{bmatrix} = \Pi \leq 0. \tag{38}$$

It is clear that $\dot{\mathbb{Q}}_j \leq 0$, yields that $\mathbb{Q}_j$ is bounded $|\mathbb{Q}_j| < \infty$, and consequently $e_j^x$, $e_j^y$ and $e_j^\theta$ are bounded too.

Further, we have $\int_0^\infty \Pi dt \le \mathbb{Q}_j(\infty) - \mathbb{Q}_j(0)$ has a finite limit since $\dot{\mathbb{Q}}_j \le 0$ and $\mathbb{Q}_j(t) \ge 0$. From (29) and Assumption 3, we get $\|\underline{e}_j\|$ and $\|\dot{\underline{e}}_j\|$ are bounded, i.e. $\Pi$ is uniformly continuous. As we have $\int_0^\infty \Pi dt$ does not increase and converge to some constant value, then by Barbalat's lemma [32], $\Pi \to 0$ as $t \to \infty$, which implies from (38) that $\Pi \equiv 0 \Rightarrow \underline{e}_j = [e_j^x \; e_j^y \; e_j^\theta]^T = 0$. Consequently, the pseudo velocity control (31)–(33) guarantees that the error system (23) and (29) is stable and $\underline{e}_j \to 0$ as $t \to \infty$.

## 6. UGVs Torque-Control (Backstepping-Like Feedback Linearization)

Hereafter, Assumption 4 is invalid. The proposed UGVs TVF tracking control task is based on robust cascaded velocity/torque control. In Sec. 5, a pseudo velocity control input $\underline{v}_{jc}(t)$, i.e. (31)–(33) was designed based on kinematic model. The aim in this section is to design a robust torque-control input $\underline{\tau}_j(t)$ based on the dynamic model for each mobile robot with dynamic behavior described by (12), such that $\lim_{t\to\infty}[\underline{v}_j(t) - \underline{v}_{jc}(t)] = 0$ (see Definition 5.1). Achieving this for $j = 1, 2, \ldots, N_R$ guarantees the mobile robots are tracking the formation reference trajectory, and simultaneously the formation is maintained and stabilized around the quadrotor coordinates $(x_q, y_q)$. In addition, we will illustrate the UAV-leader trajectory tracking control design. Before we start presenting the UGVs torque-control design, let us introduce the following assumption.

**Assumption 5.** The disturbance $\bar{\tau}_{dj}(t) = [\bar{\tau}_{dj,1}(t) \; \bar{\tau}_{dj,2}(t)]^T$ in (12) is unknown and satisfies $|\bar{\tau}_{dj,i}(t)|_{\text{abs}} \le \bar{\tau}_{d,\text{Max},i}$, $i = 1, 2$, $\forall t$, where $\bar{\tau}_{d,\text{Max}} \in \mathbb{R}^{2\times 1}$ is a vector with positive entries denoting the disturbance upper-bound. The $j$th mobile robot velocity tracking error is given

$$\underline{e}_{vjc} = \underline{v}_{jc} - \underline{v}_j. \tag{39}$$

In order to write the $j$th mobile robot dynamics as a function of $\underline{e}_{vjc}$ and $\dot{\underline{e}}_{vjc}$, we add and subtract $\bar{M}_j(q_j)\dot{\underline{v}}_{jc}$ and $\bar{V}_{mj}(q_j)\underline{v}_{jc}$ to (12), yields

$$\bar{M}_j(q_j)\dot{\underline{e}}_{vjc} = -\bar{V}_{mj}(q_j, \dot{q}_j)\underline{e}_{vjc} - \bar{B}_j\underline{\tau}_j + f_j(\varepsilon_j) + \bar{\tau}_{dj}, \tag{40}$$

where $f_j(\varepsilon_j)$ is given as

$$f_j(\varepsilon_j) = \bar{M}_j(q_j)\dot{\underline{v}}_{jc} + \bar{V}_{mj}(q_j, \dot{q}_j)\underline{v}_{jc} + \bar{F}_j(\underline{v}_j), \tag{41}$$

where $\varepsilon_j = [\dddot{\chi}_q, \dot{\chi}_q, \ddot{\eta}, \dot{\eta}, q_j, \underline{v}_j, \underline{e}_j, \dot{\underline{e}}_j]$. The quadrotor dynamics will be brought to (40) by the function $f_j(\varepsilon_j)$ through the term $\dot{\underline{v}}_{jc}$,

$$\dot{\underline{v}}_{jc} = f_{vjc}(\dddot{\chi}_q, \dot{\chi}_q, \ddot{\eta}, \dot{\eta}, q_j, \underline{e}_j, \dot{\underline{e}}_j). \tag{42}$$

From (31), (8) and (9), the quadrotor dynamics can be written in function of $\tau_q$, $u_z$, $\dot{\eta}$ and $\eta$. Substituting (8) and (9) into (42) results in the dynamics of the quadrotor to become a part of $\dot{\underline{v}}_{jc}$ as

$$\dot{\underline{v}}_{jc} = f_{vjc}(\tau_q, u_z, \dot{\eta}, \eta, \underline{e}_j, \dot{\underline{e}}_j). \tag{43}$$

The term $\dot{\underline{v}}_{jc}$ is bounded and constructible by the $j$th mobile robot relying on Assumptions 1 and 2. The function $f_{vjc}$ can be accurately approximated using a local neural network. We define the auxiliary control input $\underline{u}_j$ in (15) as

$$\underline{u}_j = \dot{\underline{v}}_{jc} + K_4\underline{e}_{vjc}, \tag{44}$$

where $K_4 \in \mathbb{R}^{2\times 2}$ is a diagonal gain matrix, with positive constants. Now, we propose the following robust torque-control input for the mobile robot $j$ by substituting (44) into (14) and adding a signum function

$$\underline{\tau}_j = \bar{B}_j^{-1}[\bar{M}_jK_4\underline{e}_{vjc} + f_j(\varepsilon_j) - \gamma \cdot \text{sign}(\underline{e}_{vjc})], \tag{45}$$

where $\gamma = [\gamma_1 \; \gamma_2]^T \in \mathbb{R}^{2\times 1}$ is a vector with positive constants that satisfies $\gamma_i > \bar{\tau}_{d,\text{Max},i}$, $i = 1, 2$ and the operator $(\cdot)$ refers to element by element matrix multiplication.

Substituting (45) into the dynamics of the $j$th mobile robot (12) and using (41) produces the closed loop error dynamics shown in (46),

$$\bar{M}_j\dot{\underline{e}}_{vjc} = -(\bar{M}_jK_4 + \bar{V}_{mj})\underline{e}_{vjc} + \bar{\tau}_{dj} - \gamma \cdot \text{sign}(\underline{e}_{vjc}). \tag{46}$$

A general TVF tracking control structure for an individual mobile robot $j, j \in [1, \ldots, N_R]$ is presented in Fig. 9.

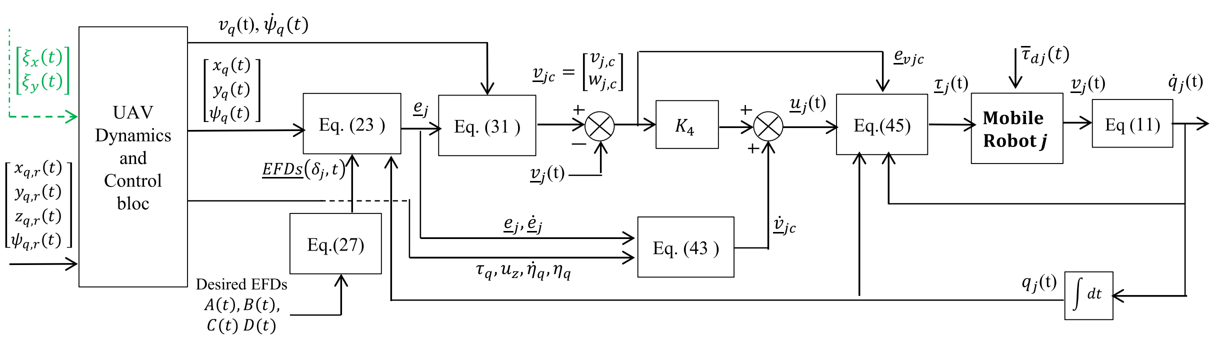



Fig. 9. TVF tracking control design architecture.

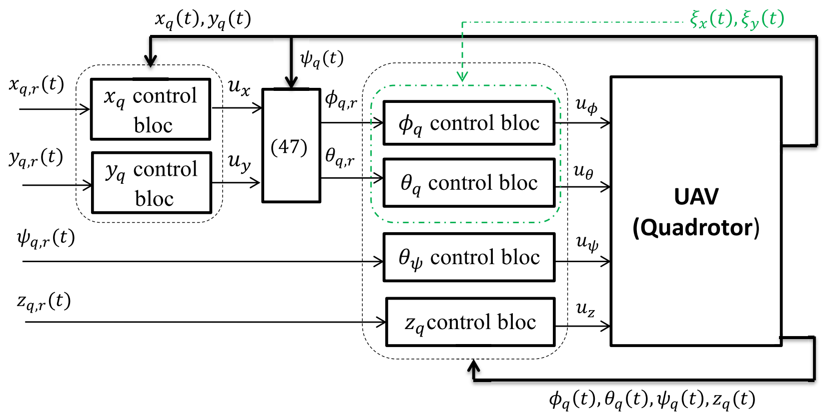



Fig. 10. UAV control system block diagram.

It is worth noting that the role of the leader-UAV is not only to provide its state information $[x_q, y_q, \psi_q]$ and the desired formation shape (i.e. EFDs vectors) for the UGVs followers, but also in providing its dynamics information and its control inputs $\ddot{\chi}_q, \dot{\chi}_q, \ddot{\eta}, \dot{\eta}, u_q$, with $u_q = [u_z, \tau_\phi, \tau_\theta, \tau_\psi]^T$. The UGVs dynamics and particularly the UAV dynamics have an important impact on the UGVs formation stability and tracking accuracy, especially when the UAV exhibits high dynamics. In the following, we will explain the UAV trajectory tracking control design.

It should be pointed out that the dynamic model of the quadrotor can be divided into two subsystems. An under-actuated subsystem, which consists of the position $x, y$, the roll and the pitch angles $\phi, \theta$ dynamics. A full actuated sub-system consists of the altitude $z$ and the yaw angle $\psi$ dynamics. Thus, the quadrotor $x, y$ states are controlled by the $\phi, \theta$ angles, respectively, while the $\phi, \theta$ states are controller by the control inputs $u_\phi, u_\theta$, respectively.

Refer to Fig. 10, the control strategy of the quadrotor system is divided into two parts in cascade. The first control part is the altitude $z$ and the yaw angle $\psi$ control, yielding the control inputs $u_z$ and $u_\psi$, respectively.

The second control part will be designed for the under-actuated sub-system to control in first step the quadrotor position $x, y$, yielding the virtual control input $u_x$ and $u_y$, respectively (see Sec. 2.2). By using (47), the reference roll and pitch angles $\phi_{q,r}$, $\theta_{q,r}$ are obtained. In the second step, the $\phi_{q,r}$, $\theta_{q,r}$ will feed-forward the roll and pitch control bloc to provide the control inputs $u_\phi, u_\theta$, respectively.

$$\begin{aligned}\phi_{q,r} &= \arcsin[u_x \sin(\psi_q) - u_y \cos(\psi_q)],\\ \theta_{q,r} &= \arcsin\left[\frac{u_x \cos(\psi_q) + u_y \sin(\psi_q)}{\cos(\phi_{q,r})}\right].\end{aligned} \tag{47}$$

All the quadrotor states are controlled by the simplest control law which is a PID controller. However, other robust controllers may be selected. The PID-controller takes many structures but the most common one has the following form:

$$u_k = K_p\left[e_k + \frac{1}{T_i}\int_0^t e_k dt + T_d \frac{de_k}{dt}\right], \tag{48}$$

where the subscript $k$ refers to the state being controlled (i.e. $x, y, z, \phi, \theta$, and $\psi$), and $K_p$, $T_i$, $T_d$ refer to the PID gains.

**Remark 6.1.** If we replace the UAV-leader with a UGV, basically, the same developments can be applied to align the new UGVs-formation tracking controller with UGV-leader dynamics under the following changes:

(i) In the formation deployment control (24), the control design will remain the same, because it is not dependent to the leader dynamics. The control design in this stage depends only on the IPF, i.e. $\mathcal{H}(x, y)$.

(ii) In the formation tracking control-based kinematic model (31)–(33), we replace $[x_q, y_q, \psi_q]$ by $[x_{ugv}, y_{ugv}, \theta_{ugv}]$ in (25)–(26). Consequentially, after redeveloping the derivative of the UGV formation tracking error (29), the parameters $v_q$, $\psi_q$, $\rho_q$ and $e_j^{(\theta_j, \psi_q)}$ in (29) and (31)–(33) will be replaced by $v_{ugv}$, $w_{ugv}$, $\theta_{ugv}$ and $e_j^{(\theta_j, \theta_{ugv})}$, respectively. The UGV-leader in this case has geometry as in Fig. 3, with behavior dynamics described by (11) and (12). In addition, $[x_{ugv}, y_{ugv}, \theta_{ugv}]$ and $v_{ugv}$, $w_{ugv}$ denote the leader-UGV generalized coordinates and its linear and angular velocities, respectively.

(iii) In the UGVs torque-control design-based dynamic model (45), the dynamics information of the leader-UGV (i.e. $v_{ugv}$, $w_{ugv}$, $\theta_{ugv}$, $\underline{\tau}_{ugv}$) will be brought to UGVs-torque-control input through the term $\dot{\underline{v}}_{jc} = f_{vjc}(v_{ugv}, w_{ugv}, \underline{\tau}_{ugv}, \underline{e}_j, \dot{\underline{e}}_j)$ in (43), with $\underline{\tau}_{ugv}$ denoting the UGV-leader torque control vector. As a result (45) will be updated accordingly.

**Remark 6.2.** In order to improve the UGVs formation stability and tracking accuracy, we can take into consideration the overall UGVs formation tracking error when designing the UAV control. Thus, we can feed-forward the UAV roll and pitch control blocs by the overall UGVs formation tracking error $\xi_x$, $\xi_y$ along the $x$ and $y$ axes, respectively (refer to Fig. 10), with $\xi_x = \sum_{j=1}^{N_R} [e_j^x]^2$ and $\xi_y = \sum_{j=1}^{N_R} [e_j^y]^2$. As such, when the UGVs formation is drifting from the desired shape, the terms $\xi_x$, $\xi_y$ will take place and will drive slightly the UAV by affecting $u_\phi$ and $u_\theta$ in such a way to minimize the values of $\xi_x$, $\xi_y$. However, this will come at the cost of temporary shifting the UAV from tracking its reference trajectory. When the UGVs progressively regain their desired formation, the terms $\xi_x$, $\xi_y$ would be decreasing so the UAV will get back to accurately track its reference trajectory. The insertion of the UGVs formation tracking error in the UAV control design yields a cross coupling between the UAV and the UGVs, which will result in the necessity to provide the UAV stability analysis.

**Remark 6.3.** In the TVF tracking control designing, it can be seen that a combined velocity/torque controller based on backstepping technique for the UGVs is proposed. In Sec. 5, first, the UGVs reference trajectories generation is detailed, then a UGVs pseudo-velocity command $\underline{v}_{jc}(t)$ is designed based on kinematic model, followed by the stability analysis. As we introduced an auxiliary control input $\underline{u}_j$ through a nonlinear feedback (14), the overall UGV system model is converted into a simpler and modular structure (14). In Sec. 6, the auxiliary control input $\underline{u}_j$ is designed to make the actual velocity vector $\underline{v}_j(t)$ follow its corresponding pseudo-velocity commands $\underline{v}_{jc}(t)$. After that, a robust torque-control input is designed for the UGVs by adding a signum term to compensate for the unknown external disturbance and the unmodeled dynamics. The proposed velocity-torque control (45) exhibits the particularities of incorporating the UVGs and the UAV dynamics information, which result in an improvement of the formation stability and robustness.

**Theorem 6.1.** *Let $K_4$ in (45) be a diagonal matrix with positive constants, $\gamma \in \mathbb{R}^{2\times 1}$ is vector that satisfies $\gamma > \bar{\tau}_{d,\mathrm{Max}}$, and suppose Assumptions 1–3 and 5 are valid. We consider the pseudo velocity-command input $\underline{v}_{jc}(t)$ for the jth mobile robot as defined by (31)–(33). We apply the torque-control input given in (45) to the formation of mobile robots $j = 1, 2, \ldots, N_R$, with dynamics described by (12). Therefore, the position, orientation and velocity tracking errors ($\underline{e}_j$ and $\underline{e}_{vjc}$, $j = [1, 2, \ldots, N_R]$) are stable and ultimately bounded around the origin.*

**Proof.** Let $\tilde{\mathbb{Q}}_j$ be a Lyapunov candidate function defined as

$$\tilde{\mathbb{Q}}_j = \mathbb{Q}_j + \frac{1}{2}\underline{e}_{vjc}^T \bar{M}_j \underline{e}_{vjc}, \tag{49}$$

where $\mathbb{Q}_j$ is already defined in (34). $\tilde{\mathbb{Q}}_j \geq 0$ and $\tilde{\mathbb{Q}}_j = 0$ only if $\underline{e}_j = 0$ and $\underline{e}_{vjc} = 0$. Differentiating (49) yields

$$\dot{\tilde{\mathbb{Q}}}_j = \dot{\mathbb{Q}}_j + (\underline{e}_{vjc})^T \bar{M}_j \dot{\underline{e}}_{vjc} + \frac{1}{2}(\underline{e}_{vjc})^T \dot{\bar{M}}_j \underline{e}_{vjc}. \tag{50}$$

It has been shown that $\dot{\mathbb{Q}}_j \leq 0$ from the proof of Theorem-5.1. Substituting (46) into (50) yields

$$\begin{aligned}\dot{\tilde{\mathbb{Q}}}_j = {} & \dot{\mathbb{Q}}_j - \underline{e}_{vjc}^T \bar{M}_j K_4 \underline{e}_{vjc} + \frac{1}{2}\underline{e}_{vjc}^T(\dot{\bar{M}}_j - 2\bar{V}_{mj})\underline{e}_{vjc} \\ & + \underline{e}_{vjc}^T \bar{\tau}_{dj}(t) - \underline{e}_{vjc}^T \gamma \cdot \mathrm{sign}(\underline{e}_{vjc}).\end{aligned} \tag{51}$$

After applying the *skew-symmetric property*, given by $X^T(\dot{\bar{M}}_j - 2\bar{V}_{mj})X = 0$, $\forall X \in \mathbb{R}^2$ [31], and relying on Assumption 5, (51) can be expressed as

$$\dot{\tilde{\mathbb{Q}}}_j \leq \begin{bmatrix} \dot{\mathbb{Q}}_j - \underline{e}_{vjc}^T \bar{M}_j K_4 \underline{e}_{vjc} \\ -|e_{vjc}^T|_{\mathrm{abs}}(\gamma - \bar{\tau}_{d,\mathrm{Max}}) \end{bmatrix} = \tilde{\Pi} \leq 0, \tag{52}$$

where $|\cdot|_{abs}$ denotes the absolute value of some vector. Examining (52), it is obvious that $\dot{\tilde{\mathbb{Q}}}_j \leq 0$ since $\bar{M}_j$ and $K_4$ are definite positive matrices, and $\gamma$ satisfies $\gamma_i > \bar{\tau}_{d,\mathrm{Max},i}$, $i = 1, 2$. Similar to the proof of Theorem 1, knowing that $\dot{\tilde{\mathbb{Q}}}_j \leq 0$, yields that $\tilde{\mathbb{Q}}_j$ is bounded $\|\tilde{\mathbb{Q}}_j\| \leq \infty$, and consequently $\underline{e}_j$ and $\underline{e}_{vjc}$ are bounded too. On the other side, $\int_0^\infty \tilde{\Pi} dt \leq \tilde{\mathbb{Q}}_j(\infty) - \tilde{\mathbb{Q}}_j(0)$ has a finite limit since $\dot{\tilde{\mathbb{Q}}}_j \leq 0$ and $\tilde{\mathbb{Q}}_j(t) > 0$, i.e. $\tilde{\Pi}$ is uniformly continuous, by Barbalat's lemma [32], $\tilde{\Pi} \to 0$ as $t \to \infty$, which implies from (52) that $\tilde{\Pi} \equiv 0 \Rightarrow [e_j \ e_{vjc}]^T = 0$. Then, from (38) and (52), we get the velocity tracking error $\underline{e}_{vjc}$ followed by the position and orientation tracking error $\underline{e}_j$ are uniformly stable. However, due to the signum term in (45), the tracking errors $\underline{e}_{vjc}$ and $\underline{e}_j$ will not exactly converge to zero, instead they will be ultimately bounded and close to a neighborhood of the origin, as it is demonstrated in simulation section, and the ultimate bounds are dependent on the selected parameter $\gamma$, which in its turn depends on the estimation of the disturbance upper-bound $\bar{\tau}_{d,\mathrm{Max}}$. Further, we can replace the signum term in (45) by a saturation function in order to reduce the chattering effect. However in that case, the system stability needs to be restudied. □

## 7. Simulation and Experimental Results

### 7.1. *Simulation results*

In this section, simulation studies are conducted to show the results of applying the proposed UAV–UGVs formation controls. The aim of the simulation results is (i) to confirm the theoretical conjecture and to show the effectiveness of the proposed deployment and UGVs TVF tracking controllers, (ii) to highlight and to demonstrate the improvement brought to UGVs formation in terms of stability, tracking accuracy and robustness when UGVs dynamics and particularly UAV (dynamics and control inputs) information are considered in the torque control design, in the presence of unknown disturbance and unmodeled dynamics. The obtained simulation results were carried out using three identical mobile robots guided by a quadrotor in MATLAB environment. The quadrotors trajectory planning and the process of producing the desired formation shape are not within the scope of the paper. Thus, the EFDs vectors and the quadrotors reference trajectory $(x_{q,r}, y_{q,r}, z_{q,r}, \psi_{q,r})$ are predefined. The quadrotor is tracking its predefined reference trajectory using a PID controller (refer to Fig. 10).

To respond the simulation aims, two simulation cases were considered. In Case 1, we used the controllers (24) and (31)–(33) with perfect velocity tracking assumption (i.e. $\underline{v}_j = \underline{v}_{jc}$ and $\dot{\underline{v}}_j = \dot{\underline{v}}_{jc}$) which means that the UGVs and the UAV dynamics are ignored. In Case 2, the controllers (24), (31)–(33) and (45) are used, implying that the UGVs and the UAV dynamics/control information are incorporated in the UGVs torque-control design. The obtained results of Case 2 are compared with the ones of Case 1. In both cases, two tasks are performed by the UGVs formation, namely, deployment task and TVF tracking task.

Hereafter, the position states $x_j, y_j$ and the heading $\theta_j$ are expressed in Meter and Radian units, respectively. The mobile robots initial posture are chosen as, $q_1 = [-5.4\ 15.96\ 0]^T$, $q_2 = [4.76\ 30.71\ 0.05]^T$ and $q_3 = [11.52\ 7.36\ 1.58]^T$. In deployment task, we consider that the initial desired shape (i.e. EFDs vectors) is given

$$x_q(t=0) = 10m; \quad y_q(t=0) = 20m; \quad n_h = 2;$$
$$A_1 = \begin{bmatrix} 7 \\ 0.5 \end{bmatrix}; \quad B_1 = \begin{bmatrix} 3 \\ 0 \end{bmatrix}; \quad C_1 = \begin{bmatrix} 1 \\ 0.2 \end{bmatrix}; \quad D_1 = \begin{bmatrix} 7 \\ 1 \end{bmatrix},$$

where the above EFDs vectors are expressed in Meter. The corresponding IPF is denoted by $\mathcal{H}(x, y)$

$$\begin{aligned}\mathcal{H}(x,y) = {} & 10^{-3}x^4 - 1/2600x^3y - 9\times10^{-3}x^3 \\ & + 5\times10^{-4}x^2y^2 - 0.026x^2y + 0.39x^2 \\ & - 10^{-4}xy^3 + 4\times10^{-3}xy^2 - 0.1xy \\ & - 0.57x + 2.5\times10^{-3}y^3 - 0.04y^2 - 1.8y + 27.3.\end{aligned}$$

The quadrotor's mechanical parameters were taken from [27, Table E]. During Task-2, we assume that the quadrotor reference trajectory is given as in Table 1, and the formation shape is described by the following desired time-varying EFDs vectors:

$$A_2(t) = A_1; B_2(t) = B_1 - 3\sin(0.05t\pi);$$
$$C_2(t) = C_1 - \sin(0.05t\pi); D_2(t) = D_1.$$

Table 1. Quadrotor reference trajectory.

| | $t \in [0, 4s]$ | $t \in [4, 16s]$ | $t \in [16, 20s]$ |
|---|---|---|---|
| $x_{q,r}(m)$ | $10 + 5t$ | $30 + 15\cos(0.26t - 2.61)$ | $30–5t$ |
| $y_{q,r}(m)$ | 20 | $30 + 15\sin(0.26t - 2.61)$ | 50 |
| $z_{q,r}(m)$ | 5 | 7 | 6 |
| $\psi_{q,r}(\text{Rad})$ | 0 | $0.26t$ | $\pi$ |

The following gains and parameters were utilized for the controllers:

$$K_x = 1.2\text{s}^{-1}; K_y = 1.2\,\frac{\text{Rad}}{\text{m}^2}; K_\theta = 1.5\,\frac{\text{Rad}}{\text{m}}; K_v = 5\,\frac{\text{m}}{\text{s}};$$
$$\lambda_1 = 0.5\text{s}^{-1}; K_4 = 1.5\times \text{I}_{2\times2}\text{s}^{-1}; M_s = 50; dt = 0.02\,\text{s}.$$

We used the following parameters of the mobile robots in the simulation, $m = 15\,\text{kg}$, $I = 0.71\,\text{kg.m}^2$, $R_w = 0.1\,\text{m}$, $L_w = 0.32\,\text{m}$ and $d_j = 0.05\,\text{m}$. The added friction to the mobile robots dynamics and the unknown external disturbance are given as

$$\bar{\tau}_{dj} = [\,0.7e^{-0.5t} + \delta 0.3\text{cos2t}\ \ 0.3e^{-0.3\delta t} + 0.12\text{sin3t}\,]^T,$$
$$\bar{F}_j = [\,\delta 0.1v_j\ \ (1-\delta)0.12w_j\,]^T; j \in [1,3],$$

where $\delta$ is random umber between 0 and 1 with uniform distribution. We choose $\gamma = [1.15\ 0.435]^T > \bar{\tau}_{d,\text{Max}}$. Figure 11 depicts the obtained mobile robots trajectories during the deployment process followed by the TVF tracking task for Case 2, using the formation controllers (24), (31)–(33) and (45). It can be seen that the three mobile robots have successfully reached the initial desired 2D desired curve, where the curve center $(a_0, c_0)$ corresponds to the quadrotor's initial coordinates $[x_q(t=0), y_q(t=0)]$. After the deployment task was completed, the mobile robots have tracked the quadrotor trajectory while preserving accurately the desired time-varying formation shape.

In Fig. 12, we see the evolution of the IPF during the deployment task for each mobile robot in Case 2 using (24). The distance between the desired 2D curve and each robot' position is expressed by the IPF. It is obvious that the IPFs are decreasing exponentially during the robots convergence process towards the desired 2D curve perimeter.

The UAV-leader position and angles tracking errors $[e_q^x, e_q^y, e_q^z]$, $[e_q^\phi, e_q^\theta, e_q^\psi]$ and its corresponding control inputs $u_q = [u_\phi, u_\theta, u_\psi]^T$ are depicted in Fig. 13. In Fig. 14, we show the trajectory tracking errors of the three mobile robots along the $x$-axis, the $y$-axis and the heading error for two the cases. The comparison shows that the TVF is seemed to be greatly degraded and not maintained in Case 1, where we notice unsteady tracking-errors state, in contrast with Case 2, in which, the settling time and the overshot

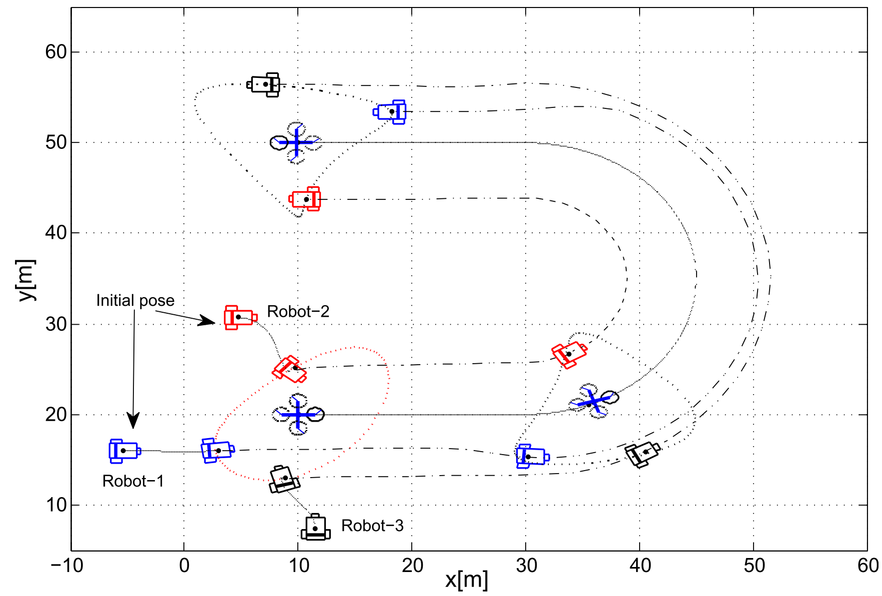



Fig. 11. Formation deployment and TVF tracking by three mobile robots (Case 2 using (24), (31)–(33) and (45)).

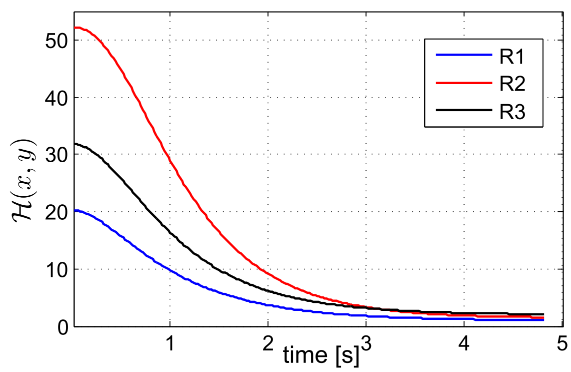



Fig. 12. IPF, i.e. $\mathcal{H}(x_j, y_j)$, $j \in [1, 3]$ in deployment task (Case 2 using the controller (24)).

decreased significantly and further the tracking-errors state are bounded around zero. The comparison demonstrates the improvement brought to the UGVs formation tracking behavior in terms of smooth convergence and accurate tracking, when considering the UGVs and the UAV dynamics and control inputs information (i.e. Case 2). The improvement can be explained further by the fact that the control law (45) includes an inner velocity tracking-loop and a discontinuous term introduced to attenuate the influence of the external disturbance. In Fig. 14(b), the errors tend to decrease till becoming almost nil at time $t = 4.9$ s, which corresponds to the end of the deployment task. From $t = 4.9$ s, we notice an increase of the $x$-axis errors and the heading errors for the three robots, this is due to the robots initial heading at the beginning of the second task (see Fig. 11), as well as the nonholonomic constraints that prevent the robots from moving along $y$-axis, then quickly the tracking-errors tend to almost a certain bound, so the time-varying formation is nearly maintained. In Table 2, a numerical comparison in terms of the tracking-error overshot and the absolute tracking-error mean values along the $x$ and $y$ axes, respectively, is provided. It is straightforward to conclude that the UGVs formation tracking performance

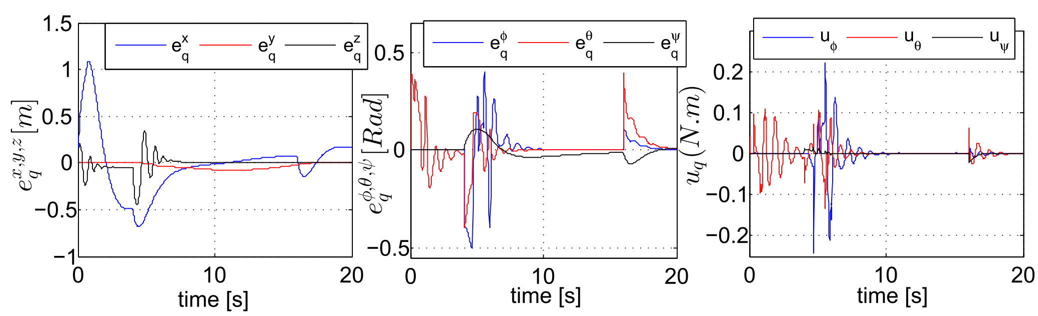
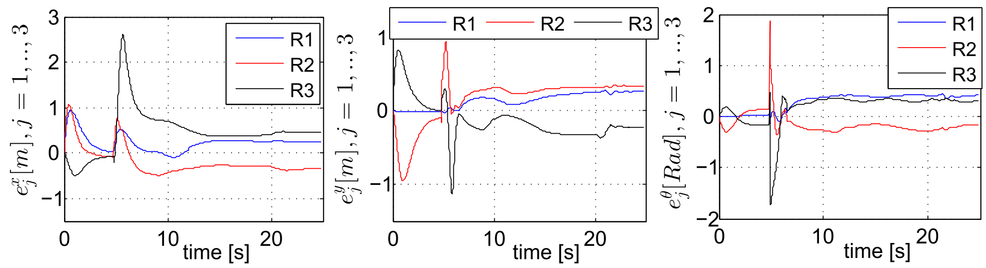
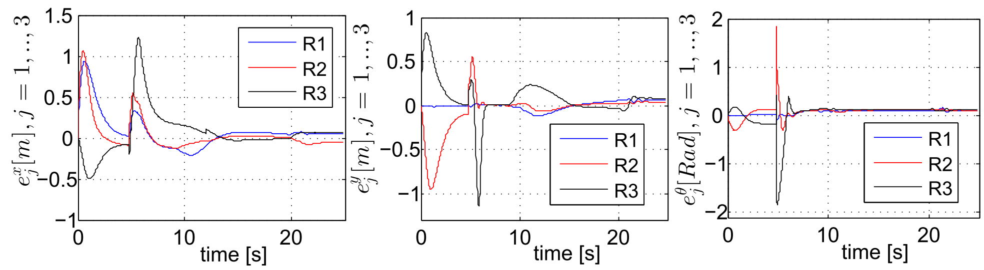



Fig. 13. UAV position and angles tracking error $[e_q^x, e_q^y, e_q^z]$, $[e_q^\phi, e_q^\theta, e_q^\psi]$, respectively, and the corresponding control inputs $u_q = [u_\phi, u_\theta, u_\psi]^T$.




(a)




(b)

Fig. 14. Formation tracking errors, (a) Case 1 using (24), (31)–(33) controllers, (b) Case 2 using (24), (31)–(33) and (45) controllers.

Table 2. Comparison between the control performance obtained in Cases 1 and 2 (TVF tracking task).

| | Case 1 | | | Case 2 | | |
|---|---|---|---|---|---|---|
| | *R*1 | *R*2 | *R*3 | *R*1 | *R*2 | *R*3 |
| $e^x_j(m)$ overshot | 0.51 | 0.76 | 2.62 | 0.32 | 0.55 | 1.23 |
| $e^y_j(m)$ overshot | 0.18 | 0.93 | 0.31 | −0.11 | −0.05 | 0.23 |
| Mean$[\lvert e^x_j(t)\rvert]$(m) | 0.24 | 0.33 | 0.54 | 0.14 | 0.11 | 0.16 |
| Mean$[\lvert e^y_j(t)\rvert]$(m) | 0.14 | 0.32 | 0.24 | 0.06 | 0.11 | 0.13 |

in terms of accurate formation tracking, stability, and disturbance attenuation have been significantly improved in Case 2, compared with Case 1, which confirms the theoretical conjecture and the effectiveness of the UGVs robust torque-control.

Figures 15 and 16 show the actual velocity and the torque control inputs $\underline{v}_j$, $\underline{\tau}_j$ of the three mobile robots, respectively, in Case 2, where some oscillations are observed due to the selection of the control gains and the effect of the signum term in $\underline{\tau}_j$ that produces the chattering phenomenon.

### 7.2. *Experimental results*

In this section, the aim is to demonstrate experimentally the feasibility and the effectiveness of the proposed deployment-controller (24) and the TVF tracking-controller (31)–(33), respectively. The validation was carried out using a group of three mobile robots type festo's Robotino$^{(\mathbb{R})}$ [33], in an area of $4\,\text{m} \times 6\,\text{m}$. Motion capture system (OptiTrack system) has been used to provide the positions and orientations of the mobile robots with high precision. It consists of a set of four cameras model Prime$^x$13 able to track in 3D a cloud of points in the defined workspace. The festo's Robotino is an omnidirectional mobile platform with three drive units (three omni-wheels) (Fig. 17). However, in this experiment, the Robotino will be controlled as if it is a nonholonomic mobile robot. Thus, a transformation between the three wheels angular velocities $(\dot{\varphi}_1, \dot{\varphi}_2, \dot{\varphi}_3)$ and

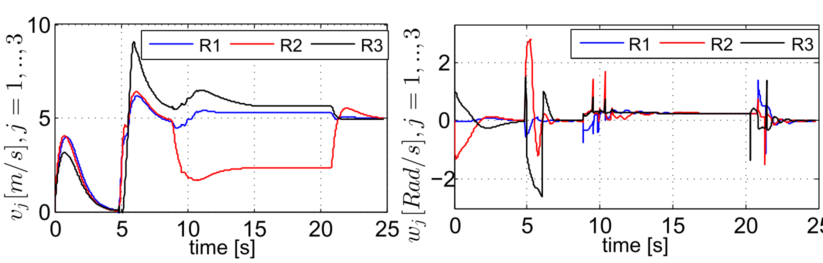



Fig. 15. Actual linear and angular velocities in Case 2 using (24), (31)–(33) and (45) controllers.

February 11, 2022 2:21:52pm WSPC/284-US 2250006 ISSN: 2301-3850 2nd Reading



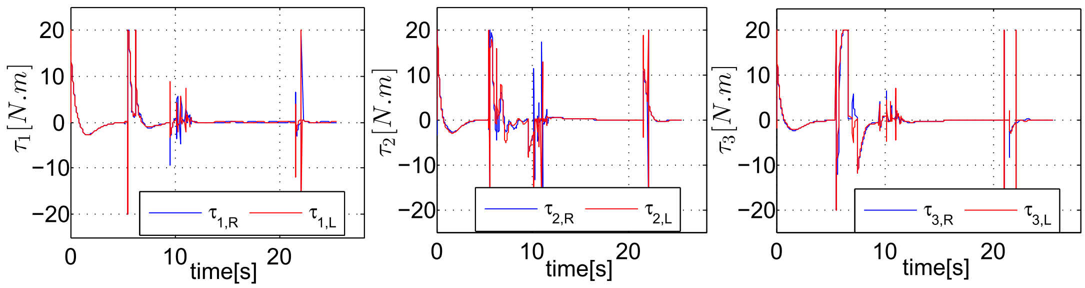



Fig. 16. Applied torque control inputs (right and left wheel), in Case 2 using (24), (31)–(33) and (45) controllers.

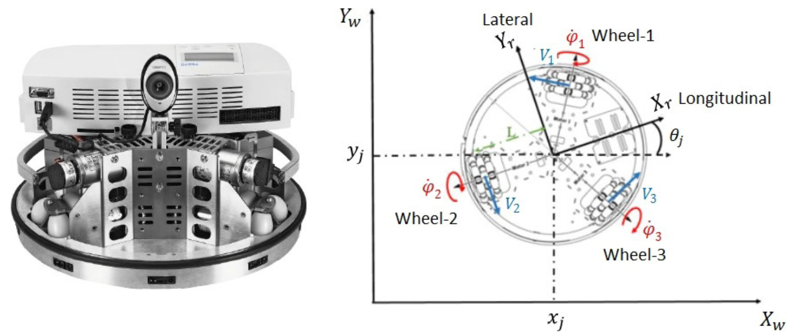



Fig. 17. Front view of Robotino with its geometry plan.

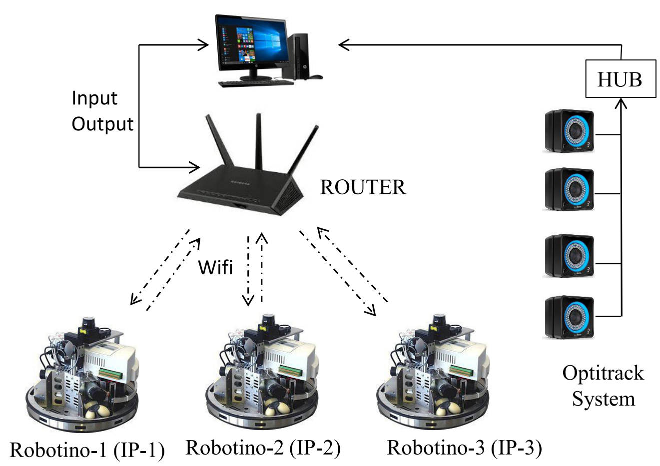



Fig. 18. Experimental platform.

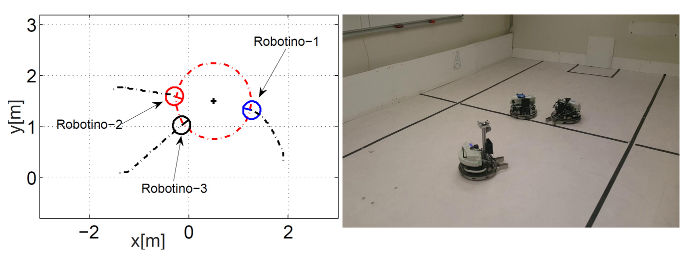



Fig. 19. Convergence process of three Robotino robots (Experiment 1).

the platform longitudinal and rotational velocities is used (53). As a result, the Robotino will be controlled by its longitudinal and rotational velocities $v_x$, $w$, respectively, ignoring its lateral velocity $v_y = 0$ [34].

$$\begin{bmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \dot{\varphi}_3 \end{bmatrix} = \frac{1}{R_w} \begin{bmatrix} -\dfrac{2\cos(\pi/6)}{3} & 0 & \dfrac{2\cos(\pi/6)}{3} \\ \dfrac{2\sin(\pi/6)}{3} & -\dfrac{2}{3} & \dfrac{2\sin(\pi/6)}{3} \\ \dfrac{1}{3L} & \dfrac{1}{3L} & \dfrac{1}{3L} \end{bmatrix}^{-1} \begin{bmatrix} v_x \\ v_y \\ w \end{bmatrix}. \tag{53}$$

In (53), $R_w$ is the identical wheels radius and $L$ is the distance from the Robotino mass center to the wheel mass center [34]. The main control program runs on a ground

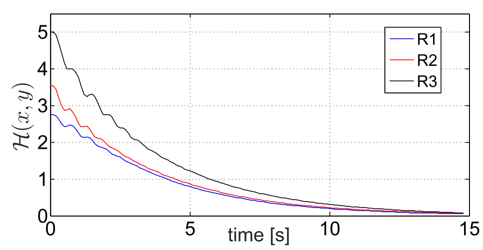



Fig. 20. Evolution of the IPF of the three Robotino during the deployment task (Experiment 1).

February 11, 2022 2:22:06pm WSPC/284-US 2250006 ISSN: 2301-3850 2nd Reading



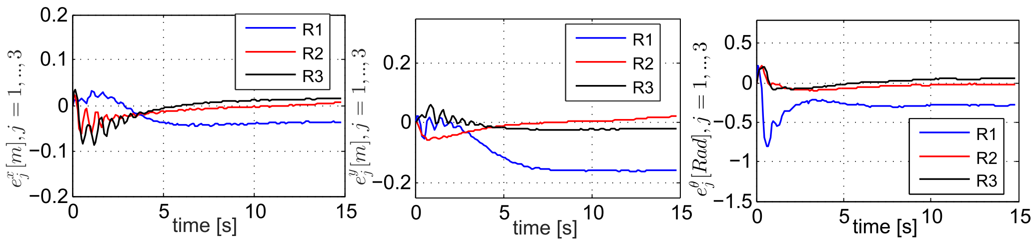



Fig. 21. Tracking errors of the three Robotino robots in Experiment 1 (i.e. deployment task).

Table 3. Reference trajectory of the virtual leader.

| | $t \in [0, 10\,\mathrm{s}]$ | $t \in [10, 20\,\mathrm{s}]$ |
|---|---|---|
| $x_L(\mathrm{m})$ | $0.1t - 1$ | $0.85 + 1.92\cos(0.1\pi t)$ |
| $y_L(\mathrm{m})$ | 1.225 | $3 + 1.92\sin(0.1\pi t)$ |
| $\psi_L(\mathrm{Rad})$ | 0 | $0.1(t - 10) - \pi/6$ |

station (PC), which communicates with the three Robotino robots via a Wi-Fi network (Fig. 18), where an external Router is configured as a point access that initiates a Wi-Fi network. Each of the Robotino robots is equipped with an embedded PC and a WLAN interface set with a unique IP address, which is configured to get integrating to the Wi-Fi initiated by the Router in mode (Clients). On the other side, the ground control PC (central unit which runs the control algorithms) is configured to connect to the initiated router Wi-Fi. Then, specific functions of a Wi-Fi-library within C++ environment are used to handle the communication network, sensory data, and the control inputs sent from the ground PC to the three Robotino robots. The state information of the three Robotino is obtained from the Optitrack processing unit and sent to the ground control PC via an ethernet port. In the case of poor communication, some UAV–UGVs connection links may get sometimes broken along time, which may result in formation tracking degradation, due to the fact of not receiving the leader's information (i.e. state, dynamics and control inputs). As a solution, the UAV–UGVs formation tracking may be explored under the constraint of partial access to UAV-leader's information (i.e. only some UGVs have access to UAV's information).

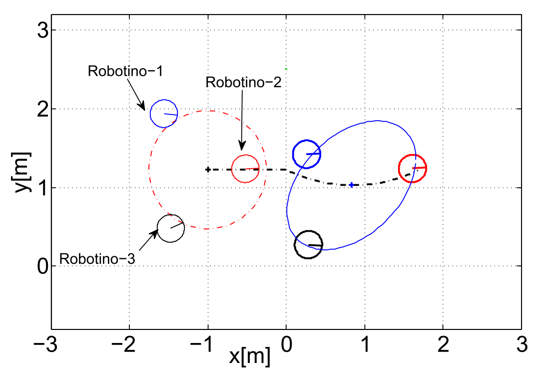



Fig. 22. TVF tracking by three Robotino robots (Experiment-2).

In the experiment, we used a virtual leader instead of a physical quadrotor due to the limitation of the flight space and the requirement to perform the experiments in the Optitrack visual range, where its reference trajectory is predefined and shared among the formation robots.

In the first experiment, deployment task control using (24) is carried out using three Robotino robots, where the initial desired EFDs vectors are given

$$x_L(t = 0) = 0.5\,\mathrm{m}; \quad y_L(t = 0) = 1.5\,\mathrm{m}; \quad n_h = 1;$$
$$\tilde{A}_1 = [0.75]\,\mathrm{m}; \quad \tilde{B}_1 = [0]\,\mathrm{m}; \quad \tilde{C}_1 = [0]\,\mathrm{m};$$
$$\tilde{D}_1 = [0.75]\,\mathrm{m}; \quad K_x = 1.5\,\mathrm{s}^{-1}; \quad K_y = 1.5\,\frac{\mathrm{Rad}}{\mathrm{m}^2};$$
$$K_\theta = 0.3\,\frac{\mathrm{Rad}}{\mathrm{m}}; \quad K_v = 0.9\,\frac{\mathrm{m}}{\mathrm{s}};$$
$$\lambda_1 = 0.2\,\mathrm{s}^{-1}; \quad M_s = 70; \quad dt = 0.1\,\mathrm{s}.$$

Figure 19 depicts the successful convergence process of the three robots Robotino towards the 2D initial desired curve starting from their initial position. Figure 20 shows the evolution of the IPF of the three Robotino robots, where it can be seen that they decrease to zero with some weak oscillations which confirm the convergence process of the mobile robots. In Fig. 21, we see the tracking errors along the $x$, $y$ axes and the heading errors. Some oscillations are observed at the beginning of the convergence while the error magnitude is bounded.

In the second experiment, three Robotino robots are used to validate the second task (TVF tracking). Once the deployment task is realized, the three robots will track a reference trajectory while maintaining a TVF shape (see Fig. 22). The leader's reference trajectory is given as in Table 3.

The used parameters, as well the EFDs vectors are given

$$\tilde{A}_2(t) = \tilde{A}_1 - 0.2\sin(0.1\pi t); \quad \tilde{B}_2(t) = \tilde{B}_1; \quad \tilde{C}_2(t) = \tilde{C}_1;$$
$$\tilde{D}_2(t) = \tilde{D}_1 + 0.3\cos(0.1\pi t);$$
$$K_x = 8; \quad K_y = 11; \quad K_\theta = 4; \quad K_v = 0.8; \quad dt = 0.1\,\mathrm{s}.$$

Figure 23 shows the tracking errors along the $x$ and $y$ axes as well as the heading errors, where we can notice that

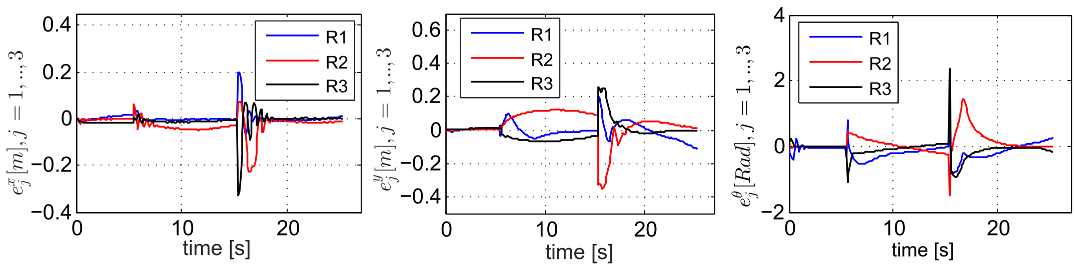



Fig. 23. Tracking errors of the three Robotino robots in Experiment 2.

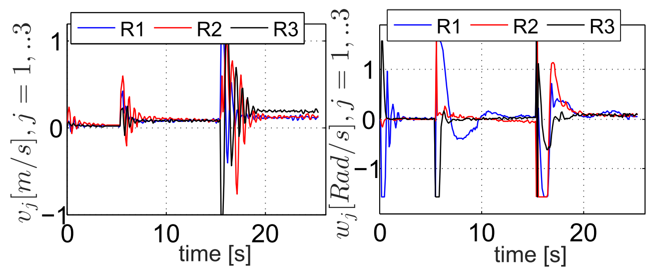



Fig. 24. Velocity-control inputs of the three Robotino robots (Experiment 2).

the error magnitude is weak and bounded, however, at $t = 15.2$ s, we observe some errors increase due to the change of the heading reference trajectory. Finally, in Fig. 24, the actual velocities control signals of the three Robotino robots are shown, in which weak chattering effect is visible on the linear velocities signals $v_j[\mathrm{m/s}], j = [1,3]$. This can be explained by backward and forward movements of the robots, due to the difficulty in choosing the suitable control parameters $K_x$ and $K_y$ during the experiment.

## 8. Conclusion and Perspective

In this paper, two novel nonlinear controllers have been proposed for controlling a UGVs system led by a UAV. First, in deployment-task control, the key feature is to use the implicit representation of the desired formation shape. The controller exhibits flexibility and offers smooth convergence to any free-formation shape. Second, in the TVF tracking control, a combined robust velocity/torque controller-based backstepping is proposed, for which the UGVs-formation tracking accuracy and robustness have been improved in regards to modeling errors and external disturbance. Lyapunov theory is used to prove the stability and the boundedness of the UGVs formation tracking errors. The effectiveness of the proposed controllers was demonstrated through numerical simulation, in which the importance of considering the UGVs and the UAV dynamics information through the insertion of an inner velocity-tracking loop is illustrated in the presence of modeling errors and unknown external disturbance. The simulation results revealed good performance in terms of smooth convergence towards the desired TVF shape, accurate tracking, and further in terms of robustness with respect to disturbance. The UGVs-formation control using the proposed pseudo velocity-control $\underline{v}_{jc}$ is validated experimentally using three robots type festo's Robotino$^{(\mathbb{R})}$. As a perspective, we plan to use a neural network to estimate the information dynamics as it is difficult to get the exact knowledge of them. In addition, we intend to consider the UGVs-group feedback information when designing the control law of the UAV-leader.

## Acknowledgments

The authors would like to express their gratitude to Prof. Rochdi Merzouki from Lille-Polytech, University of Lille, France for his advices and support to perform this work as well as for his guidance to conduct the experiments.

## References

[1] K. K. Oh, M. C. Park and H. S. Ahn, A survey of multi-agent formation control, *Automatica* **53** (2015) 424–440.
[2] Y. Ding, B. Xin and J. Chen, A review of recent advances in coordination between unmanned aerial and ground vehicles, *Unmanned Syst.* **9**(2) (2020) 97–117.
[3] Y. Liu, Z. Luo *et al.*, Cooperative routing problem for ground vehicle and unmanned aerial vehicle: The application on intelligence, surveillance, and reconnaissance missions, *IEEE Access* **7** (2019) 63504–63518.
[4] S. Huang, S. Rodney, T. Huat, W. Wai, N. Martinel, G. Luca Forest and C. Micheloni, Coverage control of multiple unmanned aerial vehicles: A short review, *Unmanned Syst.* **6**(2) (2018) 131–144.
[5] D. H. Choi, B. H. Jung and D. K. Sung, Energy aware path planning of an unmanned aerial vehicle acting as a communication relay for mobile ground nodes, *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **233** (3) (2019) 1124–1132.
[6] H. Duan and S. Liu, Unmanned air/ground vehicles heterogeneous cooperative techniques: Current status and prospects, *Sci. China Technol. Sci.* **53**(5) (2010) 1349–1355.
[7] A. Brandao, M. Sarcinelli, R. Carellit and T. Bastos-Filho, Decentralized control of leader–follower formations of mobile robots with obstacle avoidance, in *IEEE Int. Conf. Mechatronics*, Spain, 2009, pp. 1–6.
[8] Z. Hou and I. Fantoni, Distributed leader-follower formation control for multiple quadrotors with weighted topology, in *IEEE Proc. 10th Conf. System of Systems Engineering*, USA, 2015, pp. 256–261.

February 11, 2022 2:22:10pm WSPC/284-US 2250006 ISSN: 2301-3850 2nd Reading




[9] X. X. Yang, G. Y. Tang *et al.*, Formation control for multiple autonomous agents based on virtual leader structure, in *24th Chinese Control and Decision Conf.*, IEEE, Taiyuan, China, 2012, pp. 2833–2837.

[10] H. Rezaee and A. A. Farzaneh, Decentralized cooperative control scheme with obstacle avoidance for a team of mobile robots, *IEEE Trans. Ind. Electron.* **61**(1) (2014) 347–354.

[11] W. Ren, Consensus strategies for cooperative control of vehicle formations, *IET Control Theory Appl.* **1**(2) (2007) 505–512.

[12] X. Wei, J. Yu, R. Wang, X. Dong, Q. Li and Z. Ren, Time-varying formation control for time-delayed multi-agent systems with general linear dynamics and switching topologies, *Unmanned Syst.* **7**(1) (2019) 3–13.

[13] A. Aghaeeyan, F. Abdollahi and H. A. Talebi, Robust cooperative control in the presence of obstacles, in *IEEE 21st Conf. Electrical Engineering (ICEE)*, Iran, 2013, pp. 1–6.

[14] R. Rahimi, A. Farzaneh and N. Karo, Time-varying formation control of a collaborative heterogeneous multi agent system, *Rob. Auton. Syst.* **62**(12) (2014) 1799–1805.

[15] M. Aranda, Y. Mezouar, G. Nicolas and C. Sagues, Scale-free vision-bsed aerial control of a ground formation with hybrid topology, *IEEE Trans. Control. Syst. Technol* **27**(4) (2018) 1703–1711.

[16] N. Mathews, A. L. Christensen, A. Stranieri, A. Scheidler and M. Dorigo, Supervised morphogenesis: exploiting morphological flexibility of self-assembling multi-robot systems through cooperation with aerial robots, *Robot. Auton. Syst.* **112** (2019) 154–167.

[17] L. V. Santana, A. S. Brandao and M. Sarcinelli-Filho, Heterogeneous leader-follower formation based on kinematic models, in *Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, Arlington, USA, 2016, pp. 342–346.

[18] M. F. Rabelo, A. S. Brandao and M. Sarcinelli-Filho, Centralized control for an heterogeneous line formation using virtual structure approach, in *IEEE Proc. 2018 Latin American Rob. Symp.*, Brazil, 2018, pp. 135–140.

[19] W. Qize, J. Qi, Chong Wu and M. Wang, Design of UGV trajectory tracking controller in UGV–UAV cooperation, in *IEEE 39th Chinese Control Conf. (CCC)*, 2020, pp. 3689–3694.

[20] Y. Esin, M. Unel and M. Yildiz, Formation control of multiple robots using parametric and implicit representations, in *ICIC Int. Conf. Intelligent Computing*, Berlin, 2008, pp. 558–565.

[21] S. Evren and M. Unel, Planar formation control of swarm robots using dynamical elliptic Fourier descriptors, *Trans. Inst. Meas. Control.* **37**(5) (2015) 661–671.

[22] A. Allam, M. Tadjine and A. Nemra, 3D Robots formation cooperative control using EFDs, in *PAIS 3rd Int. Conf. Pattern Analysis and Intelligent System*, IEEE, Algeria, 2018, pp. 1–6.

[23] F. P. Kuhl and C. R. Giardina, Elliptic fourier features of a closed contour, *Comput. Gr. Image Process.* **18** (1982) 236–258.

[24] M. Nixon and A. Aguado *Feature Extraction and Image Processing* (Elsevier, London, 2008).

[25] H. Yalcin, M. Unel and W. Wolovich, Implicitization of parametric curves by matrix annihilation, *Int. J. Comput. Vis.* **54**(1) (2003) 105–115.

[26] B. Wang and Z. Youmin, Adaptive sliding mode fault-tolerant control for an unmanned aerial vehicle, *Unmanned Syst.* **5**(4) (2017) 209–221.

[27] S. Bouabdallah, Design and control of quadrotors with application to autonomous flying, PhD. thesis, Ecole Polytechnique Federale de Lausanne (2007).

[28] R. Fierro and F. L. Lewis, Control of a nonholomic mobile robot: Backstepping kinematics into dynamics, *J. Robot. Syst.* **14**(3) (1997) 149–163.

[29] R. Dhaouadi and A. Abu-Hatab, Dynamic modeling of differential-drive mobile robots using Lagrange and Newton–Euler methodologies: A unified framework, *Adv. Robot. Autom.* **2** (2013) 1–7.

[30] Y. Saidi, A. Nemra and M. Tadjine, Robust mobile robot navigation using fuzzy type 2 with wheel slip dynamic modeling and parameters uncertainties, *Int. J. Model. Simul.* **40**(6) (2020) 397–420.

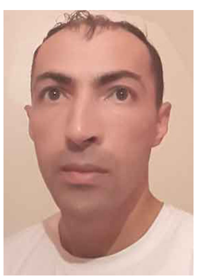
[31] F. L. Lewis and A. Yesildirak, *Neural Network Control of Robot Manipulators and Nonlinear Systems* (Taylor and Francis, London, UK, 1999).

[32] H. K. Khalil, *Nonlinear systems* (Englewood Cliffs, NJ: Prentice Hall, 2002).

[33] Website of the Festo subsidiary, https://www.festo-didactic.com.

[34] B. Kellalib, A. Nouara, C. Vincent and A. Nemra, Towards simultaneous localization and mapping tolerant to sensors and software faults: Application to omnidirectional mobile robot, *Proc. Inst. Mech. Eng. Pt. I: J. Syst. Contr. Eng.* **235**(2) (2020) 269–288.


**Ahmed Allam** received his engineering degree from the Ecole Militaire Polytechnique (EMP), Algiers, Algeria in Electrical engineering and the Magistere degree in control and automation from (EMP), Algeria, in 2010 and 2015, respectively. He is now pursuing Ph.D. degree in Automatic in the laboratory of Commande des processus (LCP) of National Polytechnic School (ENP), Algiers. His current research interests include Multi-Agents/Vehicles Systems control, Consensus protocols, Nonlinear and Cooperative control.



**Abdelkrim Nemra** received his engineering degree in Electrical Engineering from the Military Polytechnic School of Algiers (EMP), Algeria, and his Ph.D. degree in Electrical Engineering from Cranfield University, United Kingdom (UK), in 2004 and 2010, respectively. Since 2011, he is within the Department of Intelligent Autonomous Vehicles, EMP, Algeria, where he is currently. His main research interests are in Robotic, Control and Computer vision.

**Mohamed Tadjine** received his engineering degree in Electrical Engineering from the National Polytechnic School of Algiers (ENP), Algeria, and his Ph.D. degree in Automatic Control from National Polytechnic Institute of Grenoble (INPG), France, in 1990 and 1994, respectively. From 1995 to 1997, he was a researcher at the Automatic Systems Laboratory, Amiens, France. Since 1997, professor Tadjine is within the Department of Automatic Control, ENP, Algeria, where he is currently. His main research interests are in Robust and Nonlinear control, Estimation and Fault diagnosis.