Agile Coordination and Assistive Collision Avoidance for Quadrotor Swarms Using Virtual Structures

Dingjiang Zhou, Zijian Wang, and Mac Schwager

***Abstract*—This paper presents a method for controlling a swarm of quadrotors to perform agile interleaved maneuvers while holding a fixed relative formation, or transitioning between different formations. The method prevents collisions within the swarm, as well as between the quadrotors and static obstacles in the environment. The method is built upon the existing notion of a Virtual Structure, which serves as a framework with which to plan and execute complex interleaved trajectories, and also gives a simple, intuitive interface for a single human operator to control an arbitrarily large aerial swarm in real time. The Virtual Structure concept is integrated with differential flatness-based feedback control to give an end-to-end integrated swarm tele-operation system. Collision avoidance is achieved by using multiple layered potential fields. Our method is demonstrated in hardware experiments with groups of 3–5 quadrotors tele-operated by a single human operator, and simulations of 200 quadrotors tele-operated by a single human operator.**

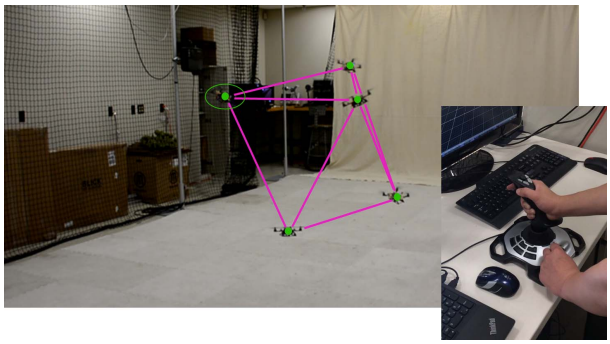


Fig. 1. Tele-operated flight of a swarm of five micro quadrotors. The pyramidal Virtual Rigid Body formation is rotating and translating, controlled by a human operator with a standard gaming joystick.

I. Introduction

In this paper, we propose an end-to-end system for tele-operated flight of an arbitrarily large swarm of micro aerial vehicles. By using a standard gaming joystick, we provide a natural interface for a human operator to guide a quadrotor swarm through rich, interleaved trajectories in an environment with obstacles. The operator controls the whole swarm as a single body, while each quadrotor autonomously attempts to hold the desired formation while avoiding collisions, hence the human operator can focus on the overall maneuver of the swam. Our approach can be useful in security, surveillance, and search and rescue applications, in which a human operator must maneuver a swarm through, e.g., a cluttered building, a forest, or a disaster site. It can also be used by a human operator to control artistic aerial swarm displays in real time.

We build upon the Virtual Structure concept [1]–[7], which has been previously applied primarily to satellite formations and ground vehicle formations. A set of static formations, as well as transitions between formations, is pre-stored. The human pilot flies the Virtual Structure directly as if it were a single aircraft, selecting between desired formations on-the-fly. Meanwhile, each quadrotor within the structure autonomously determines its own control action required to hold the formation, or transition between formations, throughout the maneuver. We include multiple potential fields applied both to the whole swarm and to individual quadrotors in order to avoid collisions among the quadrotors and with obstacles in the environment. Finally, a differential flatness-based feedback-feedforward controller [8], [9] allows each quadrotor to track the required trajectory.

More specifically, we propose an end-to-end system that incorporates three key elements to enable scalable quadrotor swarm tele-operation. Firstly, the swarm is represented as a Virtual Structure, a single body with a single reference frame. This Virtual Structure abstraction gives a simple, natural way for a single human operator to command trajectories for an arbitrary number of quadrotors, by representing the swarm as a single body. Secondly, individual quadrotors in the swarm use potential fields to hold the formation while deforming enough to avoid collisions with obstacles and with one another. Thirdly, the Virtual Structure and potential fields together give a desired trajectory for each quadrotor to follow. This trajectory is executed using a low-level differential flatness based feedforward-feedback controller.

The main challenge in integrating these three elements into a swarm control system is to ensure that the conditions for each element is compatible with the others. For example, we require that the trajectory for the Virtual Structure (whether tele-operated or pre-planned) is sufficiently smooth so as to give trajectories for the quadrotors that are four times continuously differentiable, thereby enabling the use of the differential flatness based low-level controller. Similarly, we require that the potential fields give desired trajectories that (i) avoid collision, but (ii) also give trajectories that are four

This work was supported in part by NSF grant IIS-1646921 and NSF grant CNS-1330008. We are grateful for this support.

D. Zhou is with Bito Robotics, Inc., Pittsburgh, PA 15203, USA, dingjiangzhou@bitorobotics.com.

M. Schwager and Z. Wang are with the Department of Aeronautics and Astronautics, Stanford University, Stanford, CA 94305, USA, {schwager, zjwang}@stanford.edu.

times continuously differentiable.

Our system enables scalability to swarms of arbitrary size by decoupling the control of the swarm into two sub-problems, (i) controlling the trajectory of the VRB in the global reference frame, and (ii) controlling the trajectory for each individual quadrotor within the VRB local reference frame. We let the human operator control the VRB in the global frame, and let the quadrotors autonomously compute their own control action in the VRB local frame, where collision avoidance is dealt with. Fig. 1 shows a tele-operated flight experiment (Section VI-B) in a motion capture environment. Preliminary results of some of the material from the paper has appeared in [10], [11]. Beyond these conference papers, this paper implements a formation library defining a set of formations and transformations for the quadrotor swarm, includes new simulation results with 200 quadrotors, and gives more technical details and more in-depth comparison with prior work.

A. Related Work

Our work builds on the existing work in Virtual Structures, which have been used extensively in the multi-agent formation control literature [1]–[7]. These works use a pre-defined virtual formation structure, combined with local control to maintain that structure throughout a maneuver. The focus has been on controlling mobile robots in the plane [1], [3], [7], controlling spacecraft formations [2], [4], controlling underwater vehicles for ocean monitoring [5], or controlling more abstract mobile agents [6]. We treat the Virtual Structure not as a rigid structure, but a dynamic reference frame in which quadrotors traverse trajectories within the frame when moving from one formation to another. Unlike previous work in virtual structures, we demonstrate our approach experimentally with a group of quadrotors in a motion-capture lab. Also, our method is integrated tightly with a differential flatness-based feed-forward, feed-back control architecture. Differential flatness [12], [13] allows for the use of highly efficient trajectory planning algorithms, a property that has been exploited to control single quadrotors through agile trajectories [8], [9], [14].

Many other methods have been proposed for formation control and trajectory planning. One popular approach adapts controllers from multi-agent consensus [15]–[17], to designing formation control strategies, for example [18], [19]. Much existing work has also focused on solving the position assignment problem in transitioning a swarm from one formation to another for quadrotors [20], or to concurrently plan trajectories while computing the minimal distance assignment [21]. In [22], [23], a sequential convex programming and distributed optimization approach is used to control a formation of quadrotors while reconfiguring the formation to avoid collision with the environment. The authors of [24] also considered formation control for quadrotors using only relative bearing information, and in [25], the authors controlled quadrotor formations using only on-board sensing from downward-facing cameras. In [20], tools for trajectory generation and micro quadrotor swarm control with different formations were described. Also in [26], the authors show methods to display 3D features, as well as animations, with multiple aerial vehicles. The authors in [27] proposed a multi-layer control architecture for semi-autonomous haptic teleoperation of UAVs.

Several architectures have also been proposed for human-swarm interfaces for ground robots, including [28]–[31]. Tele-operation of ground robot swarms was considered in [32], where multiple non-holonomic vehicles were controlled in a leader-follower formation and the leader robot was remotely driven to a desired velocity. In [24], a haptic interface is used to control the formation of a group of quadrotors. Gesture based human-swarm interaction is seen in [33], in which authors deciphered human gestures to drive multiple ground mobile robots to display faces. Also in [34], a shared control of groups of quadrotor UAVs is proposed and a human assistance module allowed for human intervention when needed. In contrast, with our human-swarm interface the user steers the swarm as if it were a single aircraft, allowing the user to select between a library of formations on-the-fly to suit the situation.

Obstacle avoidance for single and multiple quadrotors is also heavily researched in the literature. A single quadrotor controlled through a haptic interface was demonstrated in [35], where the pilot received force feedback when obstacles were close. In [36], the authors provided a method to assist the operator to fly a quadrotor to perform collision avoidance by predicting its future trajectory. The paper [37] presented a method to assist the tele-operation of a quadrotor by sensing the nearby environment using sonar sensors.

Our work is different from the above works in several ways:

1) We describe a complete architecture from planning the trajectory for the whole swarm as a Virtual Structure to controlling individual quadrotors at a low-level using differential flatness.
2) We propose the Virtual Structure as a natural human-swarm interface.
3) We propose a simple, intuitive collision avoidance approach using multiple potential fields to allow the swarm to deform around obstacles, avoiding collisions among the quadrotors and with objects in the environment.
4) We demonstrate our algorithms in hardware experiments with groups of 3–5 quadrotors, and simulations with 200 quadrotors.

The remainder of this paper is organized as follows. In Section II, we introduce quadrotor dynamics, and define the Virtual Rigid Body and related concepts. We describe the core trajectory planning and control tools using the VRB abstraction in Section III, and our intuitive human-swarm interface in Section IV. Our assistive collision avoidance algorithms are illustrated in Section V. Then, we demonstrate experiments in Section VI. Finally, our conclusion is shown in Section VII.

II. Preliminary

In this section, we give the necessary background on quadrotor dynamics, and our Virtual Rigid Body abstraction.

A. Quadrotor Dynamics and Differential Flatness

A quadrotor is well-modeled as a rigid body with forces and torques applied from the four rotors and gravity [8]. The

relevant forces, moments, and coordinate frames are shown in Fig. 2.

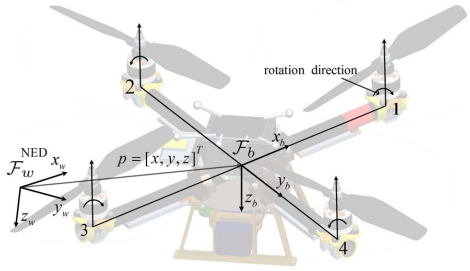


Fig. 2. Coordinate frames of a quadrotor. The global reference frame is denoted by $\mathcal{F}_w$, and the quadrotor body-fixed frame is denoted by $\mathcal{F}_b$.

The orientation of the quadrotor with respect to the global reference frame $\mathcal{F}_w$ is defined by a rotation matrix $\mathbf{R} = \mathbf{R}_{(z,\psi)}\mathbf{R}_{(y',\theta)}\mathbf{R}_{(x'',\phi)}$, which is parameterized with ZYX Euler angles. The angles ϕ, θ and ψ are the roll, pitch and yaw, respectively. The quadrotor dynamics is given by a nonlinear system of equations as follows,

$$\dot{\mathbf{v}} = \mathbf{g} + \frac{1}{m}\mathbf{R}\mathbf{f},\ \dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\Omega},\ \dot{\boldsymbol{\omega}}_b = \mathbf{J}^{-1}\boldsymbol{\tau} - \mathbf{J}^{-1}\boldsymbol{\Omega}\mathbf{J}\boldsymbol{\omega}_b,\ \dot{\mathbf{p}} = \mathbf{v} \tag{1}$$

where $\mathbf{v} = [v_x, v_y, v_z]^\top$ is the velocity in the global frame $\mathcal{F}_w$, $\mathbf{g} = [0, 0, g]^\top$ is the acceleration due to gravity, m is the quadrotor mass, $\mathbf{f} = [0, 0, f_z]^\top$ is the total thrust, where f_z is the summation of the four rotors' thrust. The angular velocity is donated as $\boldsymbol{\omega}_b = [\omega_x, \omega_y, \omega_z]^\top$, and $\boldsymbol{\Omega} = \boldsymbol{\omega}_b^\wedge = [0, -\omega_z, \omega_y; \omega_z, 0, -\omega_x; -\omega_y, \omega_x, 0]$ is the tensor form of $\boldsymbol{\omega}_b$. The torque generated from the rotors on the quadrotor is $\boldsymbol{\tau} = [\tau_x, \tau_y, \tau_z]^\top$, and $\mathbf{J}$ is the inertia matrix, and $\mathbf{p} = [x, y, z]^\top$ is the position of the quadrotor in the global frame $\mathcal{F}_w$. The quadrotor's inputs are the total thrust and the three torques $\boldsymbol{\mu} = [f_z, \tau_x, \tau_y, \tau_z]^\top$, which is in four dimensions, and the system has a 12 dimensional state vector, $\boldsymbol{\xi} = [x, y, z, v_x, v_y, v_z, \psi, \theta, \phi, \omega_x, \omega_y, \omega_z]^\top$. We also refer to the pose of a quadrotor as the position and orientation together $\boldsymbol{\delta} = (\mathbf{p}, \mathbf{R})$.

A *trajectory* refers to a signal over a given time interval, e.g., we may refer to a state trajectory, $\boldsymbol{\xi}(t)$, $t \in [t_1, t_2]$, a pose trajectory, $\boldsymbol{\delta}(t)$, $t \in [t_1, t_2]$, or an input trajectory $\boldsymbol{\mu}(t)$, $t \in [t_1, t_2]$. Planning a trajectory to be executed by a quadrotor is not trivial, as one must find the states and inputs that satisfy the dynamics (1). We call a state trajectory that satisfies the dynamics of a quadrotor for some input trajectory *dynamically feasible*, as defined formally as follows.

Definition 1: (**Dynamically Feasible Trajectory**) A quadrotor state trajectory $\boldsymbol{\xi}(t)$ is called dynamically feasible over a time interval $[t_1, t_2]$ if there exists a control input trajectory $\boldsymbol{\mu}(t)$ such that $\boldsymbol{\xi}(t)$ and $\boldsymbol{\mu}(t)$ satisfy (1) for all $t_1 \le t \le t_2$.

The quadrotor dynamics (1) are differentially flat [8], [13], [38]. This property allows one to plan an arbitrary trajectory for a set of chosen flat outputs (as long as it is sufficiently smooth), and analytically find a dynamically feasible quadrotor state trajectory, and the open-loop input trajectory required to follow that state trajectory. The flat outputs for a quadrotor are commonly chosen to be $\boldsymbol{\sigma} = [x, y, z, \psi]^\top$. Given a flat output trajectory $\boldsymbol{\sigma}(t)$ that is four times continuously differentiable, the control input trajectory $\boldsymbol{\mu}(t)$ and the state trajectory $\boldsymbol{\xi}(t)$ can be found analytically from an *endogenous transformation* [38]. Formally, we write $\boldsymbol{\xi}(t) = \boldsymbol{\gamma}(\boldsymbol{\sigma}(t), \dot{\boldsymbol{\sigma}}(t), \ddot{\boldsymbol{\sigma}}(t), \dddot{\boldsymbol{\sigma}}(t), \ddddot{\boldsymbol{\sigma}}(t))$ and $\boldsymbol{\mu}(t) = \beta(\boldsymbol{\sigma}(t), \dot{\boldsymbol{\sigma}}(t), \ddot{\boldsymbol{\sigma}}(t), \dddot{\boldsymbol{\sigma}}(t), \ddddot{\boldsymbol{\sigma}}(t))$, where $\boldsymbol{\gamma}(\cdot)$ and $\boldsymbol{\beta}(\cdot)$ are known functions derived from the quadrotor dynamics. We do not explicitly give the form of these functions here, but they can be found in, e.g., [8], [9], [14].

B. Virtual Rigid Body

The Virtual Rigid Body is extended from the virtual structure [1] with a local reference frame rigidly attached among the group. Consider a swarm of N robots labeled by $\{1, 2, \ldots, N\}$, let $\mathcal{F}_i$ denote the local reference frame of robot i, and we denote by $\mathbf{p}_i(t) \in \mathbb{R}^3$ the position, and $\mathbf{R}_i(t) \in SO(3)$ the orientation of robot i with respect to the global reference frame $\mathcal{F}_w$, at time t.

*Definition 2 (**Virtual Rigid Body**):* A Virtual Rigid Body is a group of N robots and a local reference frame $\mathcal{F}_v$, in which the local positions of the robots are specified by a set of potentially time-varying vectors $\{\mathbf{r}_1(t), \mathbf{r}_2(t), \cdots, \mathbf{r}_N(t)\}$.

An example of a Virtual Rigid Body of three robots is shown in Fig. 3, in which the *formation* and *transformation* are defined as follows.

*Definition 3 (**Formation**):* A formation $\boldsymbol{\Pi}$ is a Virtual Rigid Body with constant local positions $\{\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_N\}$ in $\mathcal{F}_v$ for a group of N robots with a time duration $T_{\boldsymbol{\Pi}} > 0$.

*Definition 4 (**Transformation**):* A transformation $\boldsymbol{\Phi}$ is a Virtual Rigid Body with time varying local positions $\{\mathbf{r}_1(t), \mathbf{r}_2(t), \cdots, \mathbf{r}_N(t)\}$ in $\mathcal{F}_v$ for a group of N robots with a time duration $T_{\boldsymbol{\Phi}} > 0$.

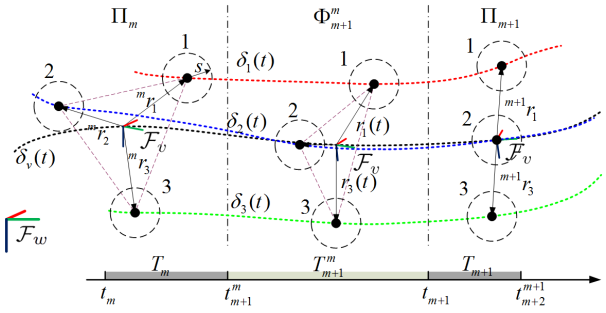


Fig. 3. The concept of a VRB with a group of three robots. The VRB is in a formation $\boldsymbol{\Pi}_m$ when $t_m < t \le t^m_{m+1}$, and $\boldsymbol{\Pi}_{m+1}$ when $t_{m+1} < t \le t^{m+1}_{m+2}$. It is in a transformation $\boldsymbol{\Phi}^m_{m+1}$ when $t^m_{m+1} \le t \le t_{m+1}$.

Let $\mathbf{p}_v(t) \in \mathbb{R}^3$ and $\mathbf{R}_v(t) \in SO(3)$ denote the position and orientation of the VRB frame $\mathcal{F}_v$ in the global reference frame $\mathcal{F}_w$ at time t, respectively. The pose trajectory of the VRB is then denoted $\boldsymbol{\delta}_v(t) = (\mathbf{p}_v(t), \mathbf{R}_v(t))$. Note that the VRB trajectory, together with the local position vectors for the quadrotors $\{\mathbf{r}_1(t), \mathbf{r}_2(t), \cdots, \mathbf{r}_N(t)\}$ in the VRB, kinematically define position trajectories $\mathbf{p}_i(t)$ for each quadrotor. Specifically, we have $\mathbf{p}_i = \mathbf{p}_v + \mathbf{R}_v\mathbf{r}_i$, $i \in \{1, 2, \cdots, N\}$, where "$(t)$" is dropped for simplicity. We require both $\boldsymbol{\delta}_v(t)$ and $\mathbf{r}_i(t)$ to be four times continuously differentiable, so

that we can apply differential flatness to extract dynamically feasible state trajectories for the quadrotors, as described in the next section.

III. Autonomous Flight Using VRB

Our methods work for any given VRB trajectory that is four times continuously differentiable. For example, we have employed the method [8] to give a $\boldsymbol{\delta}_v(t)$ based on keyframes and spline interpolations. We generate a VRB trajectory $\boldsymbol{\delta}_v(t)$ from the keyframes $\{K_1, K_2, \cdots\}$, on which velocity, acceleration, Euler angles, Euler angles' rates and so on, are assigned. Next, depending on whether the VRB is in a formation Π_m or a transformation Φ^m_{m+1}, we calculate the flat outputs $\boldsymbol{\sigma}_i(t) = [x_i, y_i, z_i, \psi_i]^\top$ and their time derivatives up to fourth order for quadrotor i. Finally, using the endogenous transformation, we find a dynamically feasible desired trajectory $\boldsymbol{\delta}_i(t)$ and open-loop control inputs $[f_{zi}(t), \boldsymbol{\tau}_i(t)^\top]^\top$. The inputs are used as feed-forward terms and the state trajectory is used as a reference trajectory in an $SE(3)$ controller [39] for quadrotor i to follow the desired trajectory.

Given a VRB trajectory and a formation with constant local position $\mathbf{r}_i$ for quadrotor i, i.e., when the quadrotor swarm is in a formation, the flat outputs $\boldsymbol{\sigma}_i(t) = [\mathbf{p}_i(t)^\top, \psi_i(t)]^\top$ and their derivatives for quadrotor i can be generated using the following simple result.

Proposition 1: Consider a VRB trajectory $\boldsymbol{\delta}_v(t) \in \mathcal{C}^4$ with constant local positions $\{\mathbf{r}_i, \cdots, \mathbf{r}_N\}$ in $\mathcal{F}_v$ of the N robots. The position $\mathbf{p}_i(t)$ of robot i in the global frame $\mathcal{F}_w$ is given by $\mathbf{p}_i = \mathbf{p}_v + \mathbf{R}_v\mathbf{r}_i$, and its derivatives can be calculated as

$$\begin{cases} \dot{\mathbf{p}}_i = \dot{\mathbf{p}}_v + \dot{\mathbf{R}}_v\mathbf{r}_i, \ \ddot{\mathbf{p}}_i = \ddot{\mathbf{p}}_v + \ddot{\mathbf{R}}_v\mathbf{r}_i, \\ \dddot{\mathbf{p}}_i = \dddot{\mathbf{p}}_v + \dddot{\mathbf{R}}_v\mathbf{r}_i, \ \ddddot{\mathbf{p}}_i = \ddddot{\mathbf{p}}_v + \ddddot{\mathbf{R}}_v\mathbf{r}_i, \end{cases}$$

while the yaw angle $\psi_i(t)$ and its derivatives can be chosen arbitrarily (e.g. it can be constant) $\forall i \in \{1, \cdots, N\}$.

Proof: The result follows from a simple application of the chain rule. ∎

However, if the local positions $\{\mathbf{r}_1(t), \cdots, \mathbf{r}_N(t)\}$ in the Virtual Rigid Body frame $\mathcal{F}_v$ for all quadrotors are time varying, i.e., the quadrotor swarm is in a transformation, and we again assume that these local position vectors are planned such that there no collisions between quadrotors at those positions, then we have the following results to generate the flat outputs $\boldsymbol{\sigma}_i(t) = [\mathbf{p}_i(t)^\top, \psi_i(t)]^\top$ and their time derivatives for quadrotor i.

Proposition 2: Given a VRB trajectory $\boldsymbol{\delta}_v(t) \in \mathcal{C}^4$ with time varying local positions $\{\mathbf{r}_1(t), \cdots, \mathbf{r}_N(t)\}$ in $\mathcal{F}_v$, in which $0 < t^m_{m+1} \leq t \leq t_{m+1}$, of the N robots. The position $\mathbf{p}_i(t)$ of robot i in the global frame $\mathcal{F}_w$ is given by $\mathbf{p}_i = \mathbf{p}_v + \mathbf{R}_v\mathbf{r}_i$, and its derivatives can be calculated as

$$\begin{cases} \dot{\mathbf{p}}_i &= \dot{\mathbf{p}}_v + \dot{\mathbf{R}}_v\mathbf{r}_i + \mathbf{R}_v\dot{\mathbf{r}}_i, \\ \ddot{\mathbf{p}}_i &= \ddot{\mathbf{p}}_v + \ddot{\mathbf{R}}_v\mathbf{r}_i + 2\dot{\mathbf{R}}_v\dot{\mathbf{r}}_i + \mathbf{R}_v\ddot{\mathbf{r}}_i, \\ \dddot{\mathbf{p}}_i &= \dddot{\mathbf{p}}_v + \dddot{\mathbf{R}}_v\mathbf{r}_i + 3\ddot{\mathbf{R}}_v\dot{\mathbf{r}}_i + 3\dot{\mathbf{R}}_v\ddot{\mathbf{r}}_i + \mathbf{R}_v\dddot{\mathbf{r}}_i, \\ \ddddot{\mathbf{p}}_i &= \ddddot{\mathbf{p}}_v + \ddddot{\mathbf{R}}_v\mathbf{r}_i + 4\dddot{\mathbf{R}}_v\dot{\mathbf{r}}_i + 6\ddot{\mathbf{R}}_v\ddot{\mathbf{r}}_i + 4\dot{\mathbf{R}}_v\dddot{\mathbf{r}}_i + \mathbf{R}_v\ddddot{\mathbf{r}}_i, \end{cases}$$

for $t \in [t^m_{m+1}, t_{m+1}]$, while the yaw angle $\psi_i(t)$ can be chosen arbitrarily (e.g. it can be constant), $\forall\ i \in \{1, \cdots, N\}$.

Proof: Again, the expressions follow from a repeated application of the chain rule. ∎

A sequence of formations and transformations can be stitched together to form a choreographed dynamic trajectory for the swarm, as shown in Fig. 3. A more complicated sequence of formations and transformations can be seen in the experiments in Section VI-B.

IV. Intuitive Human Interface

Here we describe the human-swarm interface, with which the human operator is able to control an arbitrarily large quadrotor swarm as if the swarm were a single aircraft. Our Virtual Rigid Body, which abstracts the quadrotor swarm as a single body, decouples the trajectory generation for a group of robots into two subproblems, (i) generating the trajectory of the VRB in the global reference frame $\mathcal{F}_w$, and (ii) generating the trajectory for each individual quadrotor within the VRB local reference frame $\mathcal{F}_v$. Our human-swarm interface hence solves the first subproblem, by means of generating a trajectory for the Virtual Rigid Body from a standard off-the-shelf gaming joystick.

The joystick gives commands to our base computer, which interprets the commands as a state trajectory for the Virtual Rigid Body, then the commanded VRB state is broadcast to the quadrotors. This control method is intuitive, since the joystick is widely used in flight simulators and computer games. The goal of this section is to describe the map between the joystick signal, namely, Euler angles and thrust $(\phi_J, \theta_J, \psi_J, f_J)$ and their first derivatives $(\dot{\phi}_J, \dot{\theta}_J, \dot{\psi}_J, \dot{f}_J)$, to the Virtual Rigid Body trajectory $\boldsymbol{\delta}_v(t)$ in real time. The raw data from the joystick is noisy due to unavoidably jerky motion from the human hand, hence we apply a first order low pass filter to smooth this signal. Notice that the roll ϕ_J, pitch θ_J and yaw rate $\dot{\psi}_J$ are from the joystick directly, so that the roll rate $\dot{\phi}_J$ and the pitch rate $\dot{\theta}_J$ are obtained by numerical differentiation from ϕ_J and θ_J, while the yaw ψ_J is integrated from $\dot{\psi}_J$ numerically. Also, $\ddot{\psi}_J$ is from $\dot{\psi}_J$ by numerical differentiation.

We impose virtual dynamics on the Virtual Rigid Body to map the joystick commands to a VRB trajectory. The VRB dynamics can be as simple as single integrator dynamics, $\dot{\mathbf{x}}_v = \boldsymbol{\upsilon}$, where the state is $\mathbf{x}_v = [\mathbf{p}_v^\top, \psi_v]^\top$, e.g., we set roll ϕ_v and pitch θ_v to zero. With this model, the VRB trajectory $\boldsymbol{\delta}_v(t)$ is simplified as the position trajectory $\mathbf{p}_v(t)$ with the yaw trajectory $\psi_v(t)$. We apply again a first order filter to the input $\boldsymbol{\upsilon}$, e.g., $\dot{\boldsymbol{\upsilon}} = -a\boldsymbol{\upsilon} + \mathbf{u}_1$, where a is a positive parameter, and the input $\mathbf{u}_1$ is from the joystick by $\mathbf{u}_1 = [e_1\theta_J, e_1\phi_J, e_2 f_J, e_3\dot{\psi}_J]^\top$ in real time, where e_1, e_2 and e_3 are positive constant scaling factors, such that e_1 and e_2 scale the spatial speed, and e_3 scales the yaw rate. The three scaling factors, along with the positive parameter a, can be used as tuning variables by the system designer. These variables can be set to that the system is not over- or under-responsive to the human operator, and so that actuator saturation is avoided for the quadrotors under the trajectories typically commanded by the human operator.

Thus, we obtain the VRB position trajectory $\mathbf{p}_v(t)$, yaw angle trajectory $\psi_v(t)$, as well as its velocity $\mathbf{v}_v(t)$. However,

we require the Virtual Rigid Body trajectory to be four times continuously differentiable (in $\mathcal{C}^4$) in order to use differential flatness. Therefore, the VRB acceleration $\mathbf{a}_v(t)$, jerk $\mathbf{j}_v(t)$ and snap $\mathbf{s}_v(t)$ must be found from the joystick commands. Similarly, we apply a first order filter to the acceleration, $\dot{\mathbf{a}}_v = -a\mathbf{a}_v + \mathbf{u}_2$, where the input $\mathbf{u}_2$ are from the joystick by $\mathbf{u}_2 = [e_1\dot{\theta}_J, e_1\dot{\phi}_J, e_2\dot{f}_J]^\top$. Finally, we obtain jerk and snap directly by $\mathbf{j}_v = \dot{\mathbf{a}}_v, \quad \mathbf{s}_v = \dot{\mathbf{j}}_v$. In conclusion, $\mathbf{x}_v$, $\boldsymbol{v}$, $\mathbf{a}_v$, $\mathbf{j}_v$ and $\mathbf{s}_v$ fully define the trajectory $\boldsymbol{\delta}_v(t)$ for the Virtual Rigid Body. With the VRB trajectory $\boldsymbol{\delta}_v(t)$, and if the formations and transformations of the VRB are defined, one can apply the methods described above in Sec. III to find the trajectory for each quadrotor.

As part of our human-swarm interface we propose the implementation of a formation library to store a set of formations as well as all transformations between any pair of formations for the Virtual Rigid Body.

*Definition 5 (**Formation Library**):* A *formation library* $\boldsymbol{\Xi}$ is a collection of formations $\{\boldsymbol{\Pi}_i | i \in \{1, \dots, M\}\}$ and all transformations $\{\boldsymbol{\Phi}_j^i | i \in \{1, \dots, M\}, j \in \{1, \dots, M\}, i \neq j\}$ from one formation to another for a given time duration T, for a Virtual Rigid Body of $N \geq 2$ quadrotor robots.

To demonstrate our concept, an example of a formation library for a VRB with five quadrotors is shown in Fig. 4, in which we have chosen five different formations. The VRB local frame $\mathcal{F}_v$ is assigned at the center of mass of each formation. In Fig. 4, all formations are in a plane except for the "Pyramid" formation. For these five formations, there are 20 transformations, considering that the transformation from $\boldsymbol{\Pi}_i$ to $\boldsymbol{\Pi}_j$ is different from the one from $\boldsymbol{\Pi}_j$ to $\boldsymbol{\Pi}_j$, where $i \neq j$. To generate the transformations of our formation library, the vector field method described in [14] is applied.

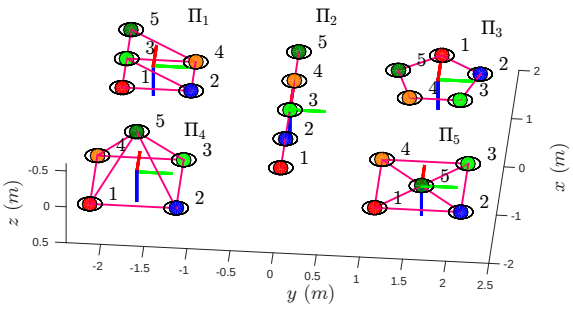


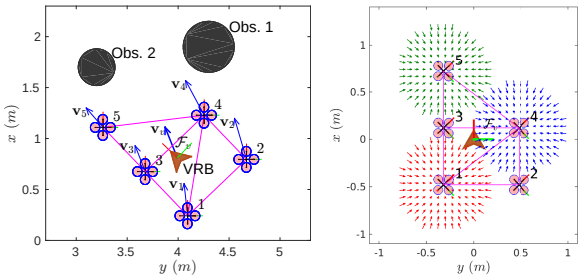
Fig. 4. Formations "Trapezoid", "Line", "Pentagon", "Pyramid" and "Square" for a Virtual Rigid Body of five quadrotors. Different colors donate different quadrotors, and the thin black circles present the safety diameters. The edges of neighborhood are plotted in magenta lines.

An application of our joystick interface and formation library methods for human control of a swarm of five quadrotors is illustrated in Section VI-B.

V. Assistive Collision Avoidance

In this section, we introduce a potential field based algorithm in our control architecture to assist the human operator by autonomously avoiding collisions with obstacles. Using multiple interacting vector fields, the algorithm allows for the formation to deform enough to avoid collisions with obstacles, and relax back into the formation when far away from obstacles. One possible draw back of potential field-based methods is that deadlocks may occur [32], however, since the swarm is teleoperated by a human operator in our case, this can be less of a problem. The operator can see when a deadlock is occurring, and reactively maneuver the swarm to dislodge the quadrotors that are stuck in a local minimum.

Our algorithm for collision avoidance is applied to environments like the scenario shown in Fig. 5(a), in which a swarm of five quadrotors is directed by a human operator, while the environment contains two obstacles. The obstacles are modeled as cylinders with different radii. We focus on formation control and collision avoidance, assuming that either the environment is instrumented with a motion capture system, or the robots have on-board perception to detect obstacles.



(a) Scenario with obstacles. (b) Formation vector fields.

Fig. 5. (a), Top view of a scenario with two obstacles in the environment. The magenta lines show the connectivity of the quadrotor swarm. The orange arrowhead shape in the middle denotes the VRB, and the $\mathbf{v}_v$ depicted by a thick blue arrow is the commanded velocity of the VRB from the joystick. The velocity of all quadrotors are denoted as $\mathbf{v}_1$ to $\mathbf{v}_5$, respectively. (b), Formation vector fields for quadrotor 1, 4 and 5 in the VRB frame $\mathcal{F}_v$.

Our control algorithm for collision avoidance runs in real time and is distributed among the quadrotor swarm, with the exception of the component that acts on the VRB, as described below. We apply the vector field method from our previous work [14], in which it was applied for single quadrotor obstacle avoidance. With this control algorithm, the resulting trajectories of each quadrotor remain dynamically feasible (Definition 1), and the smoothness requirement for our differential flatness based controller is satisfied.

A. Formation Vector Field

In our VRB framework, we apply a vector field for each quadrotor in the VRB local frame $\mathcal{F}_v$, such that when the quadrotor deviates from its desired position, a forcing vector will act on the quadrotor to pull it back to the desired position. Here we take advantage of our VRB abstraction, so that instead of considering the maneuver of the quadrotors in the global frame, we only consider their maneuver in the VRB local frame.

The vector in the vector field at position $\mathbf{r}$ for quadrotor i is defined as

$$\mathbf{V}_{0,i}(\mathbf{r}) = A(\mathbf{r}_i^d - \mathbf{r}), \quad i = 1, 2, \cdots, N,$$

where A is a positive constant parameter. As long as the quadrotor can follow its vector field, it will eventually con-

verge to its desired position in the VRB local frame, when the environment is obstacle free, as shown in Fig. 5(b).

B. Vector Fields from Obstacles

Our collision avoidance algorithm works at two levels: The first level is that we generate a vector field to force the Virtual Rigid Body to change its commanded velocity $\mathbf{v}_v$ to decrease the aggressiveness of the VRB, and the second level is that we generate another vector field from each obstacle to achieve collision avoidance for each quadrotor, while the VRB is still tracking its trajectory. The two levels can be unequally responsible in achieving collision avoidance. For example, one can give preference to adherence to the formation by increasing the magnitude of the VRB field and decreasing the individual quadrotor fields. Conversely one can give preference to deforming the formation by increasing the magnitude of the individual quadrotor vector fields and decreasing the VRB field. In this paper, for any obstacle, we consider a bounding cylinder that encompasses the obstacle in order to generate a repelling vector field. Fig. 6 shows our repelling vector field for the obstacles in the scenario of Fig. 5(a).

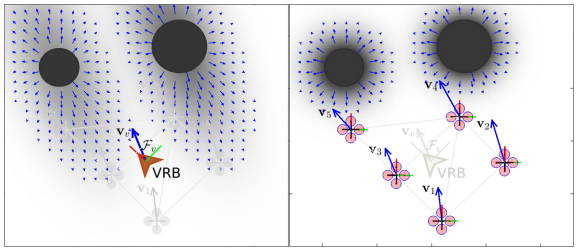


Fig. 6. Left: vector field for the VRB. Right: vector field for quadrotors. Each vector field is generated from (2), or (4), with different parameters. The darker color means larger magnitude and the lighter color means smaller magnitude. Intuitively, the VRB, or each single quadrotor, will be pushed toward the light region and avoid the dark region.

1) Vector Field for the Virtual Rigid Body: We define a repelling vector pointing from the position of an obstacle to the VRB at horizontal coordinate $\bar{\mathbf{p}}$, with its magnitude to be a Gaussian-like function related to the position and radius of this obstacle, as well as the VRB heading direction. Assuming that there are n obstacles in the environment, and we denote the horizontal position of obstacle k in the global frame $\mathcal{F}_w$ as $\bar{\mathbf{o}}_{w,k}$, then the overall repelling vector from the vector fields generated from the n obstacles at horizontal coordinate $\bar{\mathbf{p}}$ is

$$\mathbf{V}_1(\bar{\mathbf{p}}) = \sum_{k=1}^{n} f_k(\bar{\mathbf{o}}_{w,k}, r_k, \hat{\mathbf{v}}) \frac{\bar{\mathbf{p}} - \bar{\mathbf{o}}_{w,k}}{\|\bar{\mathbf{p}} - \bar{\mathbf{o}}_{w,k}\|}, \tag{2}$$

where the magnitude $f_k(\bar{\mathbf{o}}_{w,k}, r_k, \hat{\mathbf{v}})$ is a function of the obstacle position $\bar{\mathbf{o}}_{w,k}$, radius r_k and the VRB heading direction $\hat{\mathbf{v}} = \mathbf{v}_v / \|\mathbf{v}_v\|$:

$$f_k(\bar{\mathbf{o}}_{w,k}, r_k, \hat{\mathbf{v}}) = B_k \exp\Big(- (\bar{\mathbf{p}} - \bar{\mathbf{o}}_{w,k})^\top \mathbf{\Sigma}^{-1} (\bar{\mathbf{p}} - \bar{\mathbf{o}}_{w,k}) \Big),$$

where B_k is a positive scalar parameter related to the radius r_k of obstacle k. The value of B_k can be chosen so that the commanded velocity $\mathbf{v}_v$ for the VRB can still overcome the largest repelling vector, since the VRB is virtual. The 2×2 matrix $\mathbf{\Sigma}$ is positive definite and it defines the major and minor axes of the dynamic Gaussian-like function by

$$\mathbf{\Sigma}^{-1} = \Big[\hat{\mathbf{v}}, \hat{\mathbf{v}}^\perp\Big] \begin{bmatrix} \frac{1}{2\sigma_1^2} & 0 \\ 0 & \frac{1}{2\sigma_2^2} \end{bmatrix} \Big[\hat{\mathbf{v}}, \hat{\mathbf{v}}^\perp\Big]^\top, \tag{3}$$

where $\hat{\mathbf{v}}^\perp$ is the perpendicular unit vector of $\hat{\mathbf{v}}$, and σ_1, σ_2 are the covariance variables. From (3) and Fig. 6 (left), one can see that the VRB is repelled more from the obstacle if it is heading towards the obstacle directly, than if it is passing by the obstacles from the side, as shown in Fig. 6 (right).

2) Vector Field for Quadrotors: We design a stronger vector field for the individual quadrotors than that for the VRB, such that when a quadrotor gets too close to an obstacle, the repelling vector is powerful enough to "push" the quadrotor away,

$$\mathbf{V}_2(\bar{\mathbf{p}}) = \sum_{k=1}^{n} \left(C_k \exp\Big(- \frac{\|\bar{\mathbf{p}} - \bar{\mathbf{o}}_{w,k}\|^2}{2\sigma_3^2} \Big) \right) \frac{\bar{\mathbf{p}} - \bar{\mathbf{o}}_{w,k}}{\|\bar{\mathbf{p}} - \bar{\mathbf{o}}_{w,k}\|}, \tag{4}$$

where C_k is again a positive scalar parameter related to the radius of obstacle k. The value of C_k must be chosen large enough to ensure collision avoidance.

The vector field $\mathbf{V}_2$ is shown in Fig. 6 (right). Equation (4) is expressed in the global frame $\mathcal{F}_w$, while we express them in the VRB local frame $\mathcal{F}_v$ by

$$\mathbf{V}_2(\bar{\mathbf{r}}) = \sum_{k=1}^{n} \left(C_k \exp\Big(- \frac{\|\bar{\mathbf{r}} - \bar{\mathbf{o}}_{v,k}\|^2}{2\sigma_3^2} \Big) \right) \frac{\bar{\mathbf{r}} - \bar{\mathbf{o}}_{v,k}}{\|\bar{\mathbf{r}} - \bar{\mathbf{o}}_{v,k}\|}, \tag{5}$$

where $\bar{\mathbf{r}}$ denotes the horizontal coordinate in $\mathcal{F}_v$, and $\bar{\mathbf{o}}_{v,k}$ is the horizontal position of obstacle k in $\mathcal{F}_v$. Equation (5) is obvious since $\|\bar{\mathbf{p}} - \bar{\mathbf{o}}_{w,k}\| = \|\bar{\mathbf{r}} - \bar{\mathbf{o}}_{v,k}\|$.

C. Vector Fields from Quadrotors

Here we also consider inter-vehicle collision avoidance. With the two vector fields described in Section V-B, obstacle avoidance is achieved, as seen from our experiment in Section VI-C. To ensure inter-vehicle collision avoidance, and similarly to (5), we again apply the vector field method to each quadrotor in the VRB local frame $\mathcal{F}_v$,

$$\mathbf{V}_{3,i}(\mathbf{r}) = \sum_{j \in \mathcal{N}_i} \left(D \exp\Big(- \frac{\|\mathbf{r} - \mathbf{r}_j\|^2}{2\sigma_4^2} \Big) \right) \frac{\mathbf{r} - \mathbf{r}_j}{\|\mathbf{r} - \mathbf{r}_j\|},$$

where $\mathcal{N}_i$ denotes the neighbors of quadrotor i. The scale parameter D and covariance σ_4 can be tuned so that the vector field from each quadrotor will not effect the other quadrotors when the swarm is in a desired formation.

VI. Experiments

In this section, we describe hardware experiments and simulations with the proposed swarm control system. Video clips of these simulations and hardware experiments can be viewed in the accompanying video. In our hardware experiments, the positions and orientations for the quadrotors are obtained with a 16-camera OptiTrack motion capture system running at 120 Hz.

A. Large-Scale Simulations

The scalability of the approach was investigated with simulations of 200 quadrotors being controlled by a single human operator. The swarm in this example has three formations, the letters "M", "S", and "L." Snapshots of the formations in a simulation are shown in Fig. 7. The simulation includes low-level dynamical models (1) of all 200 quadrotors incorporating the differential flatness based control, and was conducted with a real-time human joystick input. Collisions with two

cylindrical obstacles (black circles) are autonomously avoided by the quadrotors. The lower right frame in Fig. 7 shows the distance between each quadrotor and both obstacles over time. All curves are above the dark black line, indicating that there were no collisions with the obstacles.

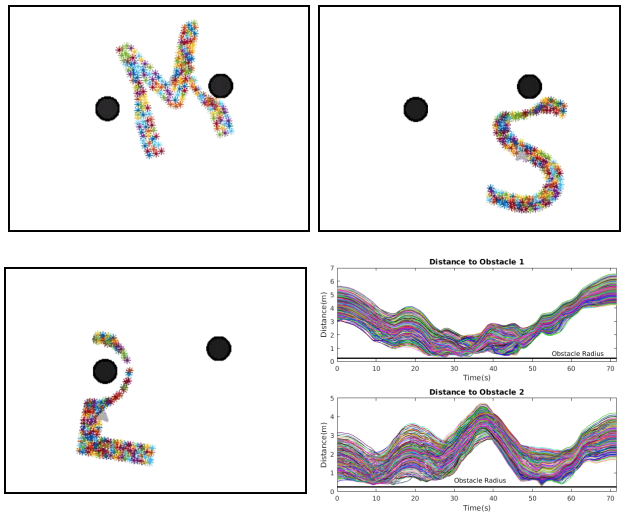


Fig. 7. Simulation results with 200 quadrotors tele-operated by a human operator with a joystick in real time. The swarm executes "M", "S", and "L" formations. Collisions with two cylindrical obstacles are autonomously avoided by the quadrotors using the proposed 3-layer vector field approach. The distance of all the quadrotors to the two obstacles are shown in the lower right plot, indicating that there were no collisions with the obstacles. Please see the accompanying video for a clip of the simulations.

B. Experiment I: Tele-operated Flight Using VRB

The system was implemented for a human operator to tele-operate a quadrotor swarm, which consists of five micro quadrotors, from a Logitech Extreme 3D Pro joystick. In this experiment, the operator selects a new formation from a selector button on the joystick, which initiates a transformation from the current formation to the new user-selected formation. The transformation occurs autonomously, and the quadrotors avoid collisions with one another, while the human operator steers the swarm as a single entity. An eight-frame sequence from the tele-operated flight experiment is shown in Fig. 8.

C. Experiment II: Assistive Collision Avoidance

The assistive collision avoidance capability of our system is validated experimentally with a group of micro quadrotors in an environment with two obstacles, similar to the setup shown in Fig. 5(a).

In this experiment, a human operator controls a swarm of five quadrotors flying a "Trapezoid" formation in the experimental arena with two cylinder obstacles. As in Experiment I, the human operator can choose a new formation by pressing a button on the joystick. A four-frame sequence of this experiment is shown in Fig. 9.

VII. Conclusions

In this paper, we proposed a suite of planning and control tools integrated into an end-to-end system for controlling the

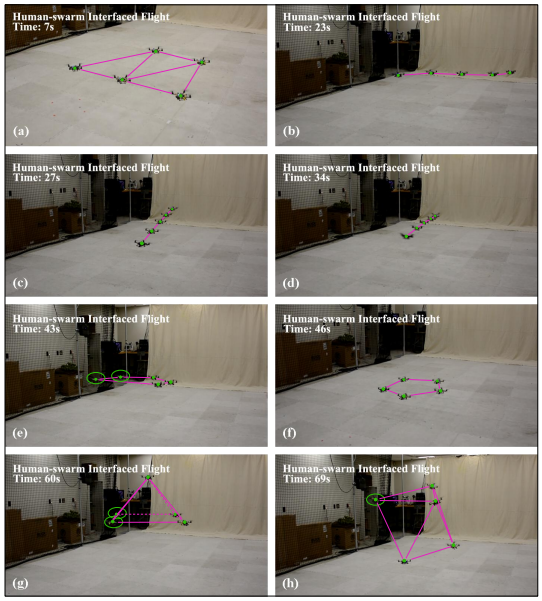


Fig. 8. Snapshots from the experimental video with five quadrotors controlled by a human operator from a joystick. The swarm starts in a "Trapezoid" formation (a), transforms (b) into a "Line" formation (c), rotates in place in the "Line" formation (d), transforms (e) into a "Pentagon" formation (f), then to a "Pyramid" formation (g), and finally rotates in place in the "Pyramid" formation (h). Quadrotors that are obscured by the dark background are highlighted with a green circle. The magenta lines between the quadrotors are used to show the shape of the formation only. See the accompanying video for clips of the experiment.

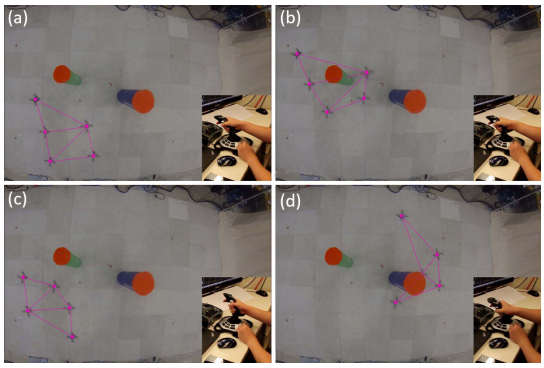


Fig. 9. A sequence of snapshots from an assistive collision avoidance experiment. (a), The quadrotor swarm is initialized as a "Trapezoid" formation; (b), The quadrotor swarm deforms while passing through obstacle 2 from two sides; (c), The corner of the "Trapezoid" formation is squeezed by obstacle 2; (d), One quadrotor comes close to obstacle 1, but with no collision. The magenta lines between the quadrotors are used to show the shape of the formation only. See the accompanying video for clips of the experiment.

flight of a swarm of quadrotors through agile, interleaved maneuvers. A Virtual Structure approach, together with differential flatness-based control, allows for the trajectory of the swarm to be decoupled from the trajectories of individual quadrotors within the swarm. Our methods enable a swarm of quadrotors to hold formations or to transition between formations in a local reference frame, while their Virtual Structure rotates and translates arbitrarily in the global fixed frame. The trajectory of the Virtual Structure can be choreographed by a base station computer, or it can be tele-operated by a single human operator from a standard gaming joystick, which provides

a simple, natural human-swarm interface. We apply multiple vector fields among the quadrotor swarm to avoid collisions among the quadrotors, and between quadrotors and obstacles in the environment, allowing the formation to compliantly deform. Our Virtual Structure based methods and assistive collision avoidance algorithms are validated experimentally with groups of 3–5 quadrotors in a motion capture system, and in simulations with 200 quadrotors.

REFERENCES

[1] M. A. Lewis and K.-H. Tan, "High precision formation control of mobile robots using virtual structures," *Autonomous Robots*, vol. 4, no. 4, pp. 387–403, 1997.

[2] R. W. Beard, J. Lawton, and F. Y. Hadaegh, "A coordination architecture for formation control," *IEEE Transactions on Control Systems Technology*, vol. 9, pp. 777–790, 2001.

[3] R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," in *IFAC World Congress*, 2002, pp. 346–352.

[4] W. Ren and R. W. Beard, "Formation feedback control for multiple spacecraft via virtual structures," in *Control Theory and Applications, IEE Proceedings-*, vol. 151, no. 3. IET, 2004, pp. 357–368.

[5] P. Ogren, E. Fiorelli, and N. E. Leonard, "Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment," *Automatic Control, IEEE Transactions on*, vol. 49, no. 8, pp. 1292–1302, 2004.

[6] E. Lalish, K. A. Morgansen, and T. Tsukamaki, "Formation tracking control using virtual structures and deconfliction," in *Decision and Control, 2006 45th IEEE Conference on*. IEEE, 2006, pp. 5699–5705.

[7] W. Ren and N. Sorensen, "Distributed coordination architecture for multi-robot formation control," *Robotics and Autonomous Systems*, vol. 56, no. 4, pp. 324–333, 2008.

[8] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2520–2525.

[9] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012.

[10] D. Zhou and M. Schwager, "Virtual rigid bodies for coordinated agile maneuvering of teams of micro aerial vehicles," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1737–1742.

[11] ——, "Assistive collision avoidance for quadrotor swarm teleoperation," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA 16)*, Stockholm, Sweden, May 2016, pp. 1249–1254.

[12] M. Fliess, J. Lévine, P. Martin, and P. Rouchon, "Flatness and defect of non-linear systems: introductory theory and examples," *International journal of control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[13] P. Martin, P. Rouchon, R. Murray *et al.*, "Flat systems, equivalence and trajectory generation," 2006.

[14] D. Zhou and M. Schwager, "Vector field following for quadrotors using differential flatness," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6567–6572.

[15] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1520–1533, 2004.

[16] A. Jadbabaie, J. Lin, and A. S. Morse, "Coordination of groups of mobile autonomous agents using nearest neighbor rules," *Automatic Control, IEEE Transactions on*, vol. 48, no. 6, pp. 988–1001, 2003.

[17] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel type of phase transition in a system of self-driven particles," *Physical review letters*, vol. 75, no. 6, p. 1226, 1995.

[18] J. Cortés, "Global and robust formation-shape stabilization of relative sensing networks," *Automatica*, vol. 45, no. 12, pp. 2754–2762, 2009.

[19] M. Ji, A. Muhammad, and M. Egerstedt, "Leader-based multi-agent coordination: Controllability and optimal control," in *American control conference*, 2006, pp. 1358–1363.

[20] A. Kushleyev, D. Mellinger, C. Powers, and V. Kumar, "Towards a swarm of agile micro quadrotors," *Autonomous Robots*, vol. 35, no. 4, pp. 287–300, 2013.

[21] M. Turpin, N. Michael, and V. Kumar, "Capt: Concurrent assignment and planning of trajectories for multiple robots," *The International Journal of Robotics Research*, vol. 33, no. 1, pp. 98–112, 2014.

[22] J. Alonso-Mora, S. Baker, and D. Rus, "Multi-robot formation control in dynamic environments via constrained optimization," *International Journal of Robotics Research*, 2017, Accepted.

[23] J. Alonso-Mora, E. Montijano, M. Schwager, and D. Rus, "Distributed multi-robot navigation in formation among obstacles: A geometric and optimization approach with consensus," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA 16)*, Stockholm, Sweden, May 2016, pp. 5356–5363.

[24] A. Franchi, C. Masone, V. Grabe, M. Ryll, H. H. Bülthoff, and P. R. Giordano, "Modeling and control of uav bearing-formations with bilateral high-level steering," *The International Journal of Robotics Research*, vol. 31, no. 12, pp. 1504–1525, 2012.

[25] E. Montijano, E. Cristofalo, D. Zhou, M. Schwager, and C. Saguees, "Vision-based distributed formation control without an external positioning system," *IEEE Transactions on Robotics*, vol. 32, no. 2, pp. 339–351, 2016.

[26] J. Alonso-Mora, M. Schoch, A. Breitenmoser, R. Siegwart, and P. Beardsley, "Object and animation display with multiple aerial vehicles," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 1078–1083.

[27] D. Lee, A. Franchi, H. I. Son, C. Ha, H. H. Bülthoff, and P. R. Giordano, "Semiautonomous haptic teleoperation control architecture of multiple unmanned aerial vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 4, pp. 1334–1345, 2013.

[28] R. Arkin and K. Ali, "Integration of reactive and telerobotic control in multi-agent robotic systems," *Proceedings of the International Conference on Simulation of Adaptive Behavior*, 1994.

[29] J. McLurkin, J. Smith, J. Frankel, D. Sotkowitz, D. Blau, and B. Schmidt, "Speaking swarmish: Human-robot interface design for large swarms of autonomous mobile robots," in *AAAI Spring Symposium: To Boldly Go Where No Human-Robot Team Has Gone Before*, 2006, pp. 72–75.

[30] C. Vasile, A. Pavel, and C. Buiu, "Integrating human swarm interaction in a distributed robotic control system," in *2011 IEEE Conference on Automation Science and Engineering (CASE)*. IEEE, 2011, pp. 743–748.

[31] E. Olson, J. Strom, R. Morton, A. Richardson, P. Ranganathan, R. Goeddel, M. Bulic, J. Crossman, and B. Marinier, "Progress toward multi-robot reconnaissance and the magic 2010 competition," *Journal of Field Robotics*, vol. 29, no. 5, pp. 762–792, 2012.

[32] S. Mastellone, D. M. Stipanović, C. R. Graunke, K. A. Intlekofer, and M. W. Spong, "Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments," *The International Journal of Robotics Research*, vol. 27, no. 1, pp. 107–126, 2008.

[33] J. Alonso-Mora, S. Haegeli Lohaus, P. Leemann, R. Siegwart, and P. Beardsley, "Gesture based human-multi-robot swarm interaction and its application to an interactive display," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5948–5953.

[34] A. Franchi, C. Secchi, M. Ryll, H. H. Bulthoff, and P. R. Giordano, "Shared control: Balancing autonomy and human assistance with a group of quadrotor uavs," *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 57–68, 2012.

[35] A. M. Brandt and M. B. Colton, "Haptic collision avoidance for a remotely operated quadrotor uav in indoor environments," in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*. IEEE, 2010, pp. 2724–2731.

[36] J. Israelsen, M. Beall, D. Bareiss, D. Stuart, E. Keeney, and J. van den Berg, "Automatic collision avoidance for manually tele-operated unmanned aerial vehicles," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6638–6643.

[37] J. Mendes and R. Ventura, "Assisted teleoperation of quadcopters using obstacle avoidance," *Journal of Automation Mobile Robotics and Intelligent Systems*, vol. 7, no. 1, pp. 54–58, 2013.

[38] R. M. Murray, "Optimization-based control," *California Institute of Technology, CA*, 2009.

[39] T. Lee, M. Leoky, and N. H. McClamroch, "Geometric tracking control of a quadrotor uav on se (3)," in *Decision and Control (CDC), 2010 49th IEEE Conference on*. IEEE, 2010, pp. 5420–5425.