2015 IEEE International Conference on Robotics and Automation (ICRA)
Washington State Convention Center
Seattle, Washington, May 26-30, 2015

Virtual Rigid Bodies for Coordinated Agile Maneuvering of Teams of Micro Aerial Vehicles

Dingjiang Zhou and Mac Schwager

***Abstract*— This paper proposes a method for controlling a team of quadrotor micro aerial vehicles to perform agile maneuvers while holding a fixed relative formation, as well as transitioning between a sequence of formations. The objective is to coordinate the quadrotors to fly in intricate interlaced patterns, similarly to an air show demonstration team. The paper proposes a new abstraction, called a *Virtual Rigid Body*, which allows the quadrotors to hold relative positions while executing agile maneuvers as a group. By planning trajectories for the Virtual Rigid Body in SE(3), trajectories for each quadrotor are obtained in order to maintain the desired formation during the maneuver. The paper also proposes a method for sequencing a series of Virtual Rigid Body formations, and automatically designing collision free transitions between successive formations, while the team simultaneously executes a trajectory in SE(3). The resulting sequence of formations and transitions gives trajectories that weave intricate designs while avoiding collisions. The method is demonstrated experimentally with three KMel K500 quadrotors flying in a motion capture environment.**

I. Introduction

In this paper, we propose a suite of trajectory design tools and control tools to control a team of quadrotor micro aerial vehicles through a sequence of agile, coordinated maneuvers. Our intention is to create trajectories for a team of quadrotors that are as rich and interleaved as one would see from a fighter jet demonstration team at an air show. We introduce an abstraction called a *Virtual Rigid Body* (VRB) in order to facilitate the agile control of multi-quadrotor formations, and transitions between formations. Our strategy builds upon differential flatness-based control techniques, which allow for controllers to be designed for a single quadrotor to execute agile trajectories.

Our trajectory design and control method is useful, for example, to maintain a team of quadrotors in formation while aggressively maneuvering the formation in a constrained environment, such as indoors or in dense urban canyons. It can also be used to transition between different formations to suit different purposes, for example to move in a triangle for aerodynamic efficiency, then convert to a line to fit through a tightly constrained opening. Our Virtual Rigid Body abstraction may also furnish a natural way for a human operator to control a large swarm of quadrotors as a single body.

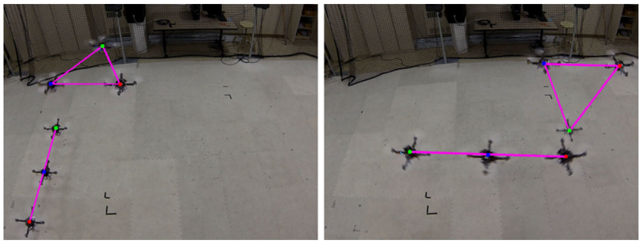

Fig. 1. The robots in the left frame dynamically reconfigure from the line to the horizontal triangle, while their Virtual Rigid Body takes off from ground and starts to perform a rolling maneuver. On the right, the robots transition from a vertical triangle back to a line, again while the formation simultaneously traverses a circle and performs a roll. Videos can be found at http://tinyurl.com/k5wnpmr.

Our key idea is to represent the group of quadrotors as a single body with a single reference frame, which we call a Virtual Rigid Body. Individual quadrotor occupy a single point on the VRB when holding a formation, or they traverse a trajectory on the VRB when transitioning between formations. Given a sufficiently smooth trajectory for the Virtual Rigid Body in SE(3), we propose a method to retrieve dynamically feasible SE(3) trajectories for the quadrotors by exploiting the differential flatness of the quadrotors' dynamics [8], [9]. We then design a feedback controller for the robots to robustly execute the specified trajectories and hold the desired formation despite disturbances and modeling errors. Furthermore, we present a method for sequencing multiple different formations, and automatically transitioning between each formation without collisions. This sequence of formations is executed in the moving reference frame of the VRB, yielding complex interleaved trajectories in the global reference frame. We demonstrate the effectiveness of our strategy in simulation, and in hardware experiments with three KMel K500 quadrotors in a motion capture environment. Fig. 1 shows still frames of three quadrotors transitioning between different formations with our method.

There exists an extensive literature in formation control, and some recent works have studied agile quadrotor formations in particular. For example, [15] proposes an agile formation control strategy in which a leader quadrotor flies an agile trajectory, while the other quadrotors control themselves to keep the formation relative to the leader, taking into account communication delays and failures. The effects of communication delays on the stability of quadrotor formations was further studied in [14]. In a non-agile regime, formation control for quadrotors was also considered in [2], which used a haptic interface for a human operator to control the formation, and in [10] which accomplished formation control for quadrotors using onboard vision, without a global reference frame. In [5] the authors describe control and

This work was supported in part by NSF grant IIS-1350904 and ONR grant N00014-12-1-1000. We are grateful for this support.

D. Zhou is with the Department of Mechanical Engineering, Boston University, Boston, MA 02215, USA, zdj@bu.edu.

M. Schwager is with the Department of Mechanical Engineering and the Division of Systems Engineering, Boston University, Boston, MA 02215, USA, schwager@bu.edu.

978-1-4799-6922-7/15/$31.00 ©2015 European Union

trajectory generation tools for controlling swarms of micro-quadrotors between different formations. In a similar vein, a method for simultaneously solving the position assignment problem and controlling robots between two formations was presented in [16]. Position assignment to transition between two formations was also considered in [19], [18] for large teams of simple particle robots. Somewhat similar to our Virtual Rigid Body, the paper [11] defined a virtual structure representation, upon which the authors developed controllers for driving a group of ground robots to maneuver in SE(2). Formation control is also related to multi-agent consensus [12], [3], [17] which has enjoyed an explosion of research in recent years. Many formation control strategies have been designed based on consensus, for example as in [4], [13].

Our method is different from the above works in several ways. Firstly, our Virtual Rigid Body abstraction allows for the design of rich interleaved trajectories, rather than static formations. Secondly, we inherently incorporate the nonlinear, differentially flat dynamics of the quadrotors to produce trajectories that are dynamically feasible by design. Thirdly, our method allows for transitioning from one formation to another, while the entire virtual structure is rotating and translating in complex ways.

Quadrotor dynamics are known to be differentially flat [1], [7], which allows for efficient trajectory planning algorithms, as described in [8], [9]. We use differential flatness extensively in this work to generate dynamically feasible quadrotor trajectories. Also, in our previous work [20], we used differential flatness theory to design closed-loop controllers (not open-loop trajectories) for quadrotors to follow desired vector fields. In this paper we build on this work to design a formation control law based on vector fields.

The remainder of the paper is organized as follows. In Section II, we introduce notations and formally state the problems that we will solve in the paper. Section III presents our trajectory design method in detail. Section IV describes our experiments with three KMel K500 quadrotors, and our conclusions and future work are presented in Section V.

II. Notation and Problem Formulation

We consider a group of N robots labeled by $\{1,\ldots,N\}$ in a 3D environment. We let $\mathcal{F}_w$ be the fixed global reference frame for the group of robots and $\mathcal{F}_i$ the local frame of robot i, where $i \in \{1,\ldots,N\}$. We denote by $p_i(t) \in \mathbb{R}^3$ the position, and $R_i(t) \in SO(3)$ the orientation of robot i with respect to the global frame $\mathcal{F}_w$, where t is time.

Definition 1: (**Virtual Rigid Body**). A Virtual Rigid Body is a group of N robots defined with a local frame $\mathcal{F}_v$, in which the local positions of the robots are specified by $\{r_1(t), r_2(t), \cdots, r_N(t)\}$, where t is time. ■

Let $p_v(t) \in \mathbb{R}^3$ and $R_v(t) \in SO(3)$ denote the position and orientation of the origin of $\mathcal{F}_v$ expressed in the global frame $\mathcal{F}_w$ at time t, respectively. The relationship between $p_i(t)$, $r_i(t)$ of robot i and $p_v(t)$, $R_v(t)$ of the VRB is described as

$$p_i = p_v + R_v r_i, \tag{1}$$

$\forall i \in \{1,\cdots,N\}$, with "$(t)$" be dropped for simplicity.

A trajectory of a VRB is defined as a trajectory of the origin of its local frame $\mathcal{F}_v$ in $\mathcal{F}_w$, it can be given as a function of time t, or specified by a series of way points with assigned velocity, acceleration, etc., as well as the time intervals between any two successive way points. The orientation can also be parameterized with Euler angles for convenience. Generally, we denote the trajectory of a VRB as a combination of a position trajectory in $\mathbb{R}^3$ and an orientation trajectory in $SO(3)$ associated with time t:

$$\delta_v(t) \in \mathcal{C}^4 : \ \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^3 \times SO(3),$$

where $\mathcal{C}^4$ denotes the set of functions whose fourth derivative is continuous. This smoothness property is required in order to generate trajectories for the quadrotors using differential flatness tools, as will be discussed in detail in Section III. Correspondingly, we denote the trajectory of robot i in $\mathcal{F}_w$ as

$$\delta_i(t) \in \mathcal{C}^4 : \ \mathbb{R}_{\geq 0} \mapsto \mathbb{R}^3 \times SO(3),$$

$\forall i \in \{1,\cdots,N\}$. Again, $\delta_i(t)$ is a combination of $p_i(t) \in \mathbb{R}^3$ and $R_i(t) \in SO(3)$.

Before describing how we plan trajectories using the VRB abstraction, we first define some terms we use in this paper, and also layout the problems that lead to our solutions.

Definition 2: (**Formation**). A formation Π is a Virtual Rigid Body of a group of N robots with constant local positions $\{r_1, r_2, \cdots, r_N\}$ in $\mathcal{F}_v$ associated with a time duration of $T_\Pi > 0$. ■

We denote ${}^m r_i$ the local position of robot i in $\mathcal{F}_v$ in formation Π_m, and ${}^m p_i$ the global position of robot i in $\mathcal{F}_w$ when the Virtual Rigid Body is in formation Π_m.

Remark 1: In a formation, the local positions of the robots are constant, but the local orientations of the robots will not necessarily be constant. In fact, the quadrotors will have to rotate significantly in their local frame in order to maintain their relative positions in the VRB during a formation flight.

Definition 3: (**Transformation**). A transformation Φ is a Virtual Rigid Body of a group of N robots with time varying local positions $\{r_1(t), r_2(t), \cdots, r_N(t)\}$ in $\mathcal{F}_v$ associated with a time duration of $T_\Phi > 0$. ■

Definition 4: (**Switch**). A switch is an transitioning event of a VRB from a status of formation to a status of transformation, or vise versa. ■

Remark 2: A typical formation flight task of a VRB with N robots consists of M formations $\{\Pi_1, \cdots, \Pi_M\}$ and $M-1$ transformations $\{\Phi_2^1, \cdots, \Phi_M^{M-1}\}$ in a sequence as $\{\Pi_1, \Phi_2^1, \Pi_2, \cdots \Phi_M^{M-1}, \Pi_M\}$. The corresponding time durations associated with this formation flight task are in a sequence as $\{T_1, T_2^1, T_2, \cdots, T_M^{M-1}, T_M\}$. The total number of switches is $2(M-1)$. The first formation starts at t_1, the switch from Π_m to Φ_{m+1}^m and that from Φ_{m+1}^m to Π_{m+1} happen at t_{m+1}^m and t_{m+1}, respectively, and the final formation ends at t_{M+1}^M, in which $1 \leq m \leq M$ and $0 \leq t_1 < t_2^1 < t_2 \cdots < t_{M+1}^M$.

For safety and collision avoidance consideration, we define a constant clearance radius s_i for robot i, and the minimum safety distance between robot i and j is $s_{ij} = s_{ji} = s_i + s_j$.

As in Fig. 2, we have an example to show a section of a typical formation flight task of a Virtual Rigid Body of three robots with a safe radius $s_i = s, \forall i$. The Virtual

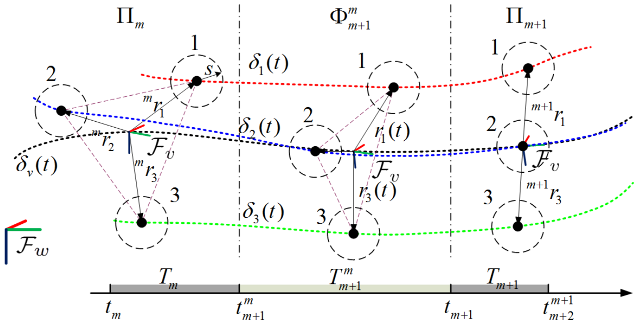

Fig. 2. A section of a typical formation flight task with two formations (a triangle Π_m on the left, and a straight line Π_{m+1} on the right), and one transformation (the smooth transition between them indicated by Φ^m_{m+1}).

Rigid Body is in formation Π_m with constant local positions $\{\ ^m r_1,\ ^m r_2,\ ^m r_3\}$ of the three robots when $t_m \leq t < t^m_{m+1}$, in a transformation Φ^m_{m+1} with time varying local positions $\{r_1(t), r_2(t), r_3(t)\}$ when $t^m_{m+1} \leq t < t_{m+1}$, and in a successive formation Π_{m+1} with constant local positions $\{\ ^{m+1}r_1,\ ^{m+1}r_2,\ ^{m+1}r_3\}$ when $t_{m+1} \leq t < t^{m+1}_{m+2}$. The switches happen at t^m_{m+1} and t_{m+1}, from Π_m to Φ^m_{m+1} and from Φ^m_{m+1} to Π_{m+1}, respectively. The VRB can be rotating and translating in the global frame $\mathcal{F}_w$ while these local formations and transformations occur.

Problem 1: (**Robots' Trajectories in a Formation**). Given a VRB trajectory $\delta_v(t) \in \mathcal{C}^4$, with constant local robot positions $\{r_1, \cdots, r_N\}$ in $\mathcal{F}_v$, find a dynamically feasible trajectory $\delta_i(t)$, for robot i, such that $\delta_i(t) \in \mathcal{C}^4$, $\forall i \in \{1, \cdots, N\}$. ■

Problem 2: (**Robots' Trajectories in a Transformation**). Given a VRB trajectory $\delta_v(t) \in \mathcal{C}^4$, with time varying local robot positions $\{r_1(t), \cdots, r_N(t)\}$ in $\mathcal{F}_v$, find a dynamically feasible trajectory $\delta_i(t)$, for robot i, such that $\delta_i(t) \in \mathcal{C}^4$, $\forall i \in \{1, \cdots, N\}$. ■

Problem 3: (**Smoothness at Switches**). Given a VRB trajectory $\delta_v(t) \in \mathcal{C}^4$, in which $t_m \leq t \leq t^{m+1}_{m+2}$, with constant local positions $\{^m r_1, \cdots, ^m r_N\}$ for $t_m \leq t \leq t^m_{m+1}$, and $\{^{m+1}r_1, \cdots, ^{m+1}r_N\}$ for $t_{m+1} \leq t \leq t^{m+1}_{m+2}$ in $\mathcal{F}_v$ of the N robots, where $0 \leq t_m < t^m_{m+1} < t_{m+1} < t^{m+1}_{m+2}$, find a transformation trajectory $r_i(t)$, in which $t^m_{m+1} \leq t \leq t_{m+1}$, and a trajectory $\delta_i(t)$, in which $t_m \leq t \leq t^{m+1}_{m+2}$, for robot i, such that $\delta_i(t) \in \mathcal{C}^4$, $\forall i \in \{1, \cdots, N\}$. ■

Problem 4: (**Collision Avoidance in Transformation**). In addition to Problem 3, we require that $\|r_i(t) - r_j(t)\| = \|p_i(t) - p_j(t)\| \geq s_i + s_j$, in which $t^m_{m+1} \leq t \leq t_{m+1}$, $\forall i, j \in \{1, \cdots, N\}$, $i \neq j$. ■

The solutions of the above problems will be shown in the following sections.

III. Trajectories Generation

Our method works for any given VRB trajectory that is four times continuously differentiable. For example, we have employed the method [8] to give a $\delta_v(t)$ based on keyframes and splines interpolations. As in Fig. 3, we generate a VRB trajectory $\delta_v(t)$ from the keyframes $\{K_1, K_2, \cdots\}$, on which velocity, accelration, Euler angles, Euler angles rates and so on, are assgined. Next, depending on whether the VRB is in a formation Π_m or a transformation Φ^m_{m+1}, we calculate the flat outputs $\sigma_i(t) = [x_i, y_i, z_i, \psi_i]^T$ and their time derivatives up to fourth order for quadrotor i. Finally, using the *endogenous transformation* for the quadrotor dynamics [8] based on the *differential flatness* theory, we find a dynamically feasible desired trajectory $\delta_i(t)$ and open-loop inputs $[f_{zi}(t), \tau_i(t)^T]^T$, that the inputs can be used as feed-forward terms in a SE(3) controller [6] to follow the desired trajectory for the quadrotor i.

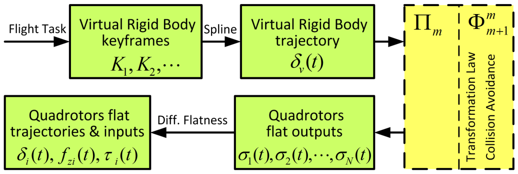
Flight Task
Virtual Rigid Body keyframes
$K_1, K_2, \cdots$
Spline
Virtual Rigid Body trajectory
$\delta_v(t)$
Π_m
Φ^m_{m+1}
Transformation Law
Collision Avoidance
Quadrotors flat outputs
$\sigma_1(t), \sigma_2(t), \cdots, \sigma_N(t)$
Diff. Flatness
Quadrotors flat trajectories & inputs
$\delta_i(t), f_{zi}(t), \tau_i(t)$

Fig. 3. The main procedure in our method to generate trajectories for the quadrotors in a Virtual Rigid Body.

Fig. 4 shows the first a few sections of a Virtual Rigid Body trajectory generated for our demostration. As seen from this trajectory, we want the quadrotors to take off slowly from the origin of $\mathcal{F}_w$ at time $t_1 = 0s$ to a point above it at $t_2 = 1.5s$, pass through the third keyframe at $t_3 = 6.5s$ with a certain upward speed and then tangent into the circle centered at the flight arena. The orientation trajectory of the VRB is not shown explicitly, but $\mathcal{F}_v$ is shown in each keyframe with an attached coordinate system.

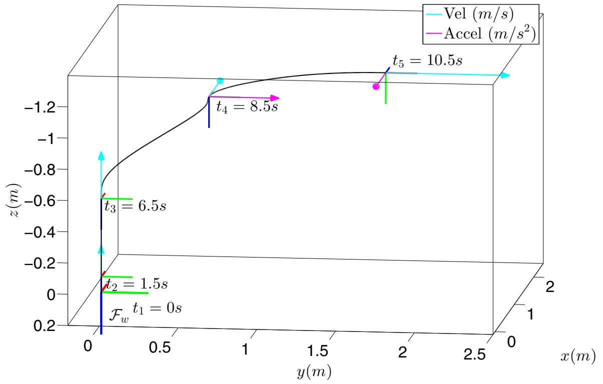

Fig. 4. Four sections of trajectories for a Virtual Rigid Body parameterized with five keyframes.

A. Trajectories for the Quadrotors in a Formation

In this section we solve Problem 1. We require the VRB trajectory, as well as a fixed set of local positions, defining the formation of the quadrotors in the VRB. In order to fly the quadrotors safely in a sequence of specified formations, we require that the formations satisfy the following assumption.

Assumption 1: (**Safety Configuration).** In a flight task with M formations $\{\Pi_1, \cdots, \Pi_M\}$, the desired local positions $\{^m r_1, \cdots, ^m r_N\}$ of the N robots in a VRB for formation Π_m, satisfy$\|\ ^m r_i -\ ^m r_j\| \geq s_i + s_j$, $\forall\ m \in \{1, \cdots, M\}$, $\forall\ i, j \in \{1, \cdots, N\}$, $i \neq j$. ■

We give six examples of safety configurations in Fig. 5 that we will use in our demonstration.

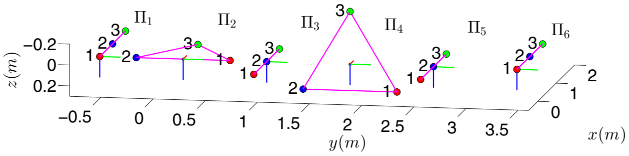

Fig. 5. Six example formations of a VRB with three quadrotors.

Given a VRB trajectory and a formation with constant local position r_i for quadrotor i, the flat outputs $\sigma_i(t) = [p_i(t)^T, \psi_i(t)]^T$ and their derivatives for quadrotor i can be generated using the following simple result.

Theorem 1: Given a VRB trajectory $\delta_v(t) \in \mathcal{C}^4$ with constant local positions $\{r_i, \cdots, r_N\}$ in $\mathcal{F}_v$ of the N robots. The position $p_i(t)$ of robot i in the global frame $\mathcal{F}_w$ is given by (1), and its derivatives can be calculated as

$$\begin{cases} \dot{p}_i &= \dot{p}_v + \dot{R}_v r_i \\ \ddot{p}_i &= \ddot{p}_v + \ddot{R}_v r_i \\ \dddot{p}_i &= \dddot{p}_v + \dddot{R}_v r_i \\ \ddddot{p}_i &= \ddddot{p}_v + \ddddot{R}_v r_i \end{cases}, \tag{2}$$

while the yaw angle $\psi_i(t)$ and its derivatives can be inherited from that of the Virtual Rigid Body directly, $\forall i \in \{1, \cdots, N\}$. ■

B. Trajectories for the Quadrotors in a Transformation

In this section we solve Problem 2. The procedure is similar to that in Section III-A, except here we have time varying local positions $\{r_1(t), \cdots, r_N(t)\}$ in the Virtual Rigid Body frame $\mathcal{F}_v$ for all quadrotors, so $\dot{r}_i(t) \neq 0$, $\ddot{r}_i(t) \neq 0$, and so on. In this case we have the following results to generate the flat outputs $\sigma_i(t) = [p_i(t)^T, \psi_i(t)]^T$ and their derivative for the quadrotors.

Theorem 2: Given a VRB trajectory $\delta_v(t) \in \mathcal{C}^4$ with time varying local positions $\{r_1(t), \cdots, r_N(t)\}$ in $\mathcal{F}_v$, in which $0 < t^m_{m+1} \leq t \leq t_{m+1}$, of the N robots. The position $p_i(t)$ of robot i in the global frame $\mathcal{F}_w$ is given by (1), and its derivatives can be calculated as

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

for $t \in [t^m_{m+1}, t_{m+1}]$, while the yaw angle $\psi_i(t)$ and its derivatives can be inherited from that of the Virtual Rigid Body directly, $\forall\ i \in \{1, \cdots, N\}$. ■

With a given VRB trajectory shown in Fig. 4, the trajectories for three quadrotors generated by Theorem 1 for our demonstration are shown in Fig. 6. We will let the team of quadrotor take off in a formation Π_1 and then transition to Π_2, with both formations illustrated in Fig. 5.

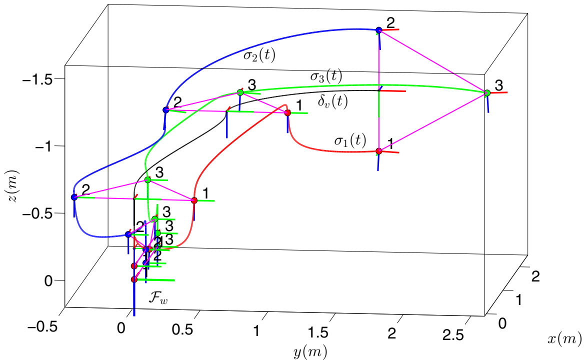

Fig. 6. Trajectories generated for three quadrotors with a given VRB trajectory shown in Fig. 4, two formations, Π_1 and Π_2 as shown in Fig. 5, and a transformation designed between them. Notice the rolling and turning of the VRB cause the formation to turn on its side in the global frame.

C. Smoothness and Collision Avoidance

Section III-A and III-B gave us smooth trajectories when the VRB is in a formation or a transformation, respectively. In this section we solve Problem 3 and 4, which consider the smoothness at the switches and collision avoidance in transformations. Designing a sufficiently smooth transformation law is critical to having good performance for a formation flight. From (3), we can make up the flat outputs $\sigma_i(t)$ and its derivatives for each quadrotor for $t \in [t^m_{m+1}, t_{m+1}]$, Remark 2 indicates that a transformation Φ^m_{m+1} only exists between two formations Π_m and Π_{m+1}. Intuitively, the position of the quadrotor i must be continuous in the $\mathcal{F}_v$ at time instances t^m_{m+1} and t_{m+1},

$$\begin{cases} r_i(t^m_{m+1}) &= {}^m r_i \\ r_i(t_{m+1}) &= {}^{m+1} r_i \end{cases}, \ \forall\ i \in \{1, \cdots, N\}, \tag{4}$$

we also require that the velocity, acceleration, jerk and snap be continuous at the time of switches,

$$\begin{cases} \dot{p}_i(t^m_{m+1}) &= {}^m \dot{p}_i \\ &\vdots \\ \ddddot{p}_i(t^m_{m+1}) &= {}^m \ddddot{p}_i \\ \dot{p}_i(t_{m+1}) &= {}^{m+1} \dot{p}_i \\ &\vdots \\ \ddddot{p}_i(t_{m+1}) &= {}^{m+1} \ddddot{p}_i \end{cases}, \tag{5}$$

which gives us

$$\begin{cases} \dot{r}_i(t) &= 0 \\ \quad \vdots \\ \ddddot{r}_i(t) &= 0 \end{cases}, \ \forall\ t \in \{t^m_{m+1}, t_{m+1}\}. \tag{6}$$

Furthermore, the minimum safety distance should be satisfied throughout a transformation, as to solve Problem 4.

The first element in designing a transformation law is to solve a *target assignment* problem which can be accomplished with the well-known *Hungarian algorithm.* A number of methods can be used, for example those in [16], [18], [19]. Straight line trajectories generated in this way (by naively matching the initial position in one formation with the final position in another) may yield trajectories that violate the minimum safety distance requirement. To modify the trajectories to prevent collision, we employ the vector field based method from our previous work [20], in which we present an algorithm to give a dynamically feasible trajectory for a quadrotor to avoid colliding with a static obstacle. Here we modify this algorithm so that all quadrotors move to avoid colliding with all others.

To this end, we create a joint vector field, which is a combination of a *transformation vector field* and a *repelling vector field*, for each quadrotor, as in Fig. 7.

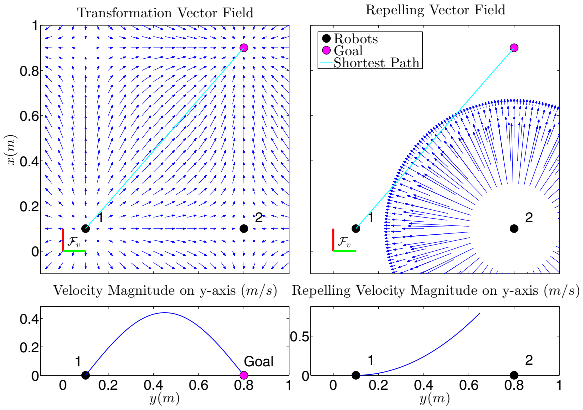

Fig. 7. The transformation law is genreated by two vector fields. We show only two robots and only the goal of robot 1 for simplicity. The velocity magnitude on the y axis of robot 1 is designed to be a sine function such that the initial and final velocity are zero, as well as their derivatives. The repelling vector magnitude created by robot 2 is a parabola function such that only when the robot comes close enough to its neighbor will it be deformed by a repelling force.

The transformation vector field considers only the initial and final position of a quadrotor to drive it from one place to another, while the repelling vector field considers the repelling effect from all other nearby quadrotors to avoid collision. The transformation velocity v_{Ti} in the transformation vector field at $r_i(t)$ for quadrotor i is given by

$$v_{Ti} = A_i \sin(a_i r_i(t) + c_i), \ \forall i \in \{1, \cdots, N\}, \tag{7}$$

in which A_i is a constant coefficient, a_i and c_i are parameters that make sure when $r_i(t^m_{m+1}) = {}^m r_i$ and $r_i(t_{m+1}) = {}^{m+1} r_i$, we have $v_{Ti} = \dot{r}_i(t) = 0$, as in (6), at these two time instances. The repelling velocity v_{Ri} of quadrotor i repelled from the repelling vector field of quadrotor j is given by

$$v^j_{Ri} = \begin{cases} B_j \left(\frac{D - \|r_i - r_j\|}{D - s_i - s_j}\right)^2 \frac{r_i - r_j}{\|r_i - r_j\|}, & \text{if } j \in \mathcal{N}_i \\ 0, & \text{if } j \notin \mathcal{N}_i \end{cases}, \tag{8}$$

$\forall i \in \{1, \cdots, N\}$, where $\mathcal{N}_i$ denotes the collection of indices of nearby quadrotors of quadrotor i. B_j is a constant coefficient, and D is a constant threshold defining $\mathcal{N}_i$ by $j \in \mathcal{N}_i$, if $\|r_j(t) - r_i(t)\| \leq D, j \neq i$. Furthermore, we have

$$v_i = v_{Ti} + \sum_{j \in \mathcal{N}_i} v^j_{Ri}, \ \forall i \in \{1, \cdots, N\}, \tag{9}$$

and by Theorem 2 in [20], the flat outputs and their derivatives are obtained to produce a dynamically feasible trajectory for quadrotor i.

Fig. 8 shows the transformation law from our combined vector field method, robot 1 and robot 2 “bend” their trajectories so that the minimum safety distance is guaranteed.

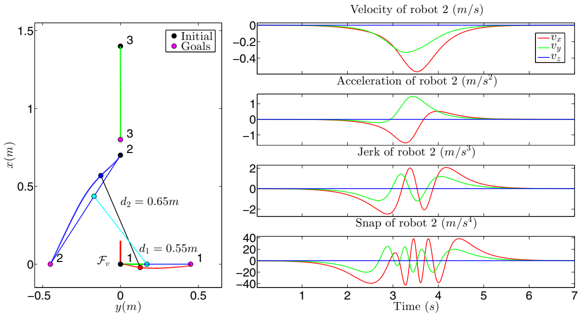

Fig. 8. Left: trajectories of the three robots in transformation Φ^1_2, which is a transitioning from Π_1 to Π_2 as shown in Fig. 5. The safety distance is set as $s = s_i + s_j = 0.6m$, and the threshold is set as $D = 0.7m$. $d_1 = 0.55m$ and $d_2 = 0.65m$ are the minimum distance when repelling vector field be applied or not, respectively. Right: Velocity, acceleration, jerk and snap of robot 2 in the VRB local frame $\mathcal{F}_v$.

IV. Experiments

Our experiments use three KMel K500 quadrotors flying through a sequence of the six formations shown in Fig. 5, with five transformations generated between each two successive formations, while following a Virtual Rigid Body trajectory. Due to a small flight volume in our experiment arena, the VRB trajectory is designed to move in a tight circle, while spinning with a constant angular rate about the roll axes. A minimum relative distance of $0.7m$ is set in our formations such that collision would easily happen if no transformation law is applied since the actual diameter of a KMel K500 quadrotor is $0.55m$. Including the five keyframes shown in Fig. 4, we have designed 30 keyframes in 71 seconds for the entire flight demonstration and Fig. 9 shows the three trajectories for the corresponding quadrotors. The three quadrotors circle in the flight arena while transitioning between different formations within a rotating VRB.

Our algorithm were implemented in C/MEX in MATLAB. Position and orientation for the quadrotors are obtained with a 16 camera OptiTrack motion capture system running at 120Hz. Our optimized C/MEX code is capable of generating a segment of trajectory in less than 5 ms for the three quadrotors, fast enough for online implementation—planning a trajectory segment at switch time instances, without downgrade the OptiTrack system. A video of our experiment is submitted along with this paper, and can also be watched at http://tinyurl.com/k5wnpmr. An eight-frame sequence of the video is shown in Fig. 10, but of course the motion of the quadrotors is difficult to appreciate from the still frames.

V. Conclusions and Future Work

In this paper, we proposed a method for generating complex interleaved trajectories for a group of quadrotor micro

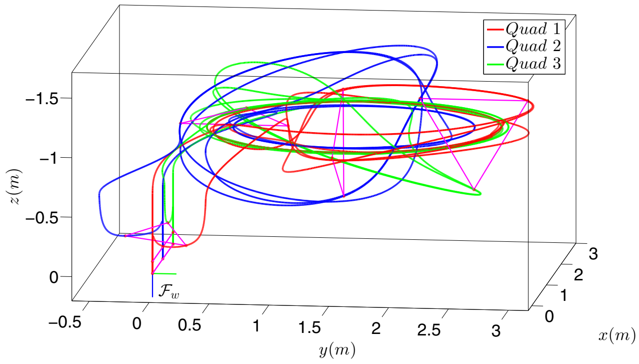

Fig. 9. Trajectories for three quadrotors. Several formations are depicted in magenta.

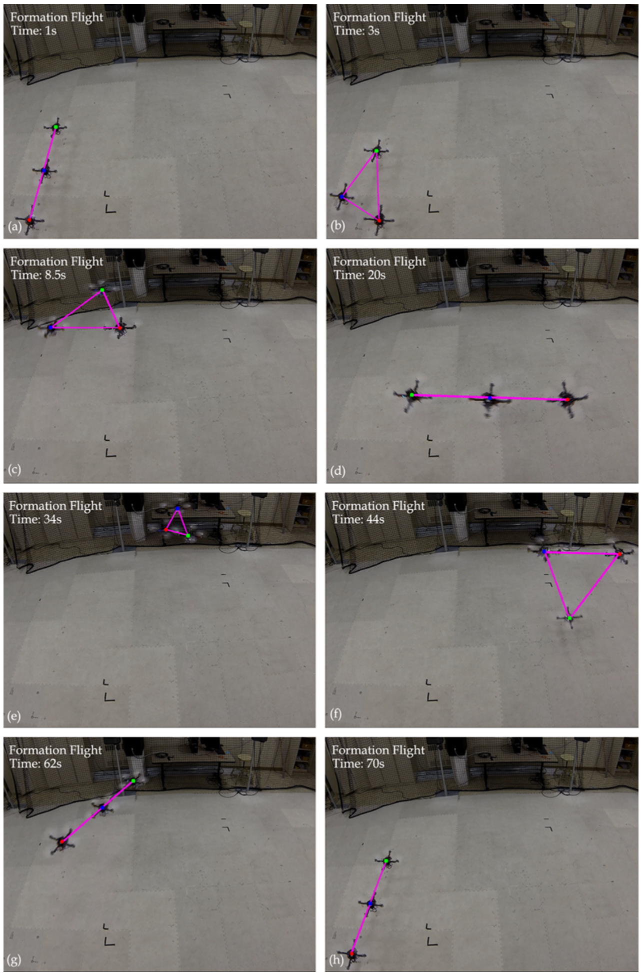

Fig. 10. Sequence of images from the experimental video with the VRB of three quadrotors circling in the flight arena. The VRB is in (a) Π_1 while taking off from ground; (b) Φ_2^1; (c) the beginning of Π_2; (d) Π_3; (e) Φ_4^3; (f) Π_4; (g) Π_5 and (h) Π_6 while landing.

aerial vehicles using a new abstraction, called a *Virtual Rigid Body*. This allows for quadrotors to hold formations and to transition between formations in a local reference frame, while their Virtual Rigid Body rotates and translates arbitrarily in the global fixed frame. Differential flatness tools are used to obtain state trajectories and open-loop inputs for the quadrotors to execute the planned trajectories. Our future work is to implement an online, semi-distributed version of our algorithm, and conduct experiments outdoors with more quadrotors.

References

[1] M. Fliess, J. Lévine, P. Martin, and P. Rouchon. Flatness and defect of non-linear systems: Introductory theory and examples. *International journal of control*, 61(6):1327–1361, 1995.

[2] Antonio Franchi, Carlo Masone, Volker Grabe, Markus Ryll, Heinrich H Bülthoff, and Paolo Robuffo Giordano. Modeling and control of uav bearing formations with bilateral high-level steering. *The International Journal of Robotics Research*, 31(12):1504–1525, 2012.

[3] Ali Jadbabaie, Jie Lin, and A Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *Automatic Control, IEEE Transactions on*, 48(6):988–1001, 2003.

[4] Meng Ji, Abubakr Muhammad, and Magnus Egerstedt. Leader-based multi-agent coordination: Controllability and optimal control. In *American control conference*, pages 1358–1363, 2006.

[5] Alex Kushleyev, Daniel Mellinger, Caitlin Powers, and Vijay Kumar. Towards a swarm of agile micro quadrotors. *Autonomous Robots*, 35(4):287–300, 2013.

[6] Taeyoung Lee, M Leoky, and N Harris McClamroch. Geometric tracking control of a quadrotor uav on se (3). In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5420–5425. IEEE, 2010.

[7] Philippe Martin, Pierre Rouchon, RM Murray, et al. Flat systems, equivalence and trajectory generation. 2006.

[8] Daniel Mellinger and Vijay Kumar. Minimum snap trajectory generation and control for quadrotors. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2520–2525. IEEE, 2011.

[9] Daniel Mellinger, Nathan Michael, and Vijay Kumar. Trajectory generation and control for precise aggressive maneuvers with quadrotors. *The International Journal of Robotics Research*, 31(5):664–674, 2012.

[10] Eduardo Montijano, Dingjiang Zhou, Eric Cristofalo, Mac Schwager, and Carlos Sagues. Vision-based distributed formation control without a global reference frame. *International Journal of Robotics Research*, 2014.

[11] Reza Olfati-Saber and Richard M Murray. Distributed cooperative control of multiple vehicle formations using structural potential functions. In *IFAC World Congress*, pages 346–352, 2002.

[12] Reza Olfati-Saber and Richard M Murray. Consensus problems in networks of agents with switching topology and time-delays. *Automatic Control, IEEE Transactions on*, 49(9):1520–1533, 2004.

[13] Wei Ren and Nathan Sorensen. Distributed coordination architecture for multi-robot formation control. *Robotics and Autonomous Systems*, 56(4):324–333, 2008.

[14] Mac Schwager, Nathan Michael, Vijay Kumar, and Daniela Rus. Time scales and stability in networked multi-robot systems. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3855–3862. IEEE, 2011.

[15] Matthew Turpin, Nathan Michael, and Vijay Kumar. Trajectory design and control for aggressive formation flight with quadrotors. *Autonomous Robots*, 33(1-2):143–156, 2012.

[16] Matthew Turpin, Nathan Michael, and Vijay Kumar. Capt: Concurrent assignment and planning of trajectories for multiple robots. *The International Journal of Robotics Research*, 33(1):98–112, 2014.

[17] Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-driven particles. *Physical review letters*, 75(6):1226, 1995.

[18] Jingjin Yu and Steven M LaValle. Distance optimal formation control on graphs with a tight convergence time guarantee. *arXiv preprint arXiv:1204.3820*, 2012.

[19] Jingjin Yu and Steven M LaValle. Shortest path set induced vertex ordering and its application to distributed distance optimal formation path planning and control on graphs. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 2775–2780. IEEE, 2013.

[20] Dingjiang Zhou and Mac Schwager. Vector field following for quadrotors using differential flatness. In *Proc. of the International Conference on Robotics and Automation (ICRA)*, 2014.