Trajectory Design and Control for Aggressive Formation Flight with Quadrotors

Matthew Turpin, Nathan Michael, and Vijay Kumar

Abstract In this work we consider the problem of controlling a team of micro-aerial vehicles moving quickly through a three-dimensional environment while maintaining a tight formation. The formation is specified by a shape matrix that prescribes the relative separations and bearings between the robots. Each robot plans its trajectory independently based on its local information of other robot plans and estimates of states of other robots in the team to maintain the desired shape. We explore the interaction between nonlinear decentralized controllers, the fourth-order dynamics of the individual robots, the time delays in the network, and the effects of communication failures on system performance. An experimental evaluation of our approach on a team of quadrotors suggests that suitable performance is maintained as the formation motions become increasingly aggressive and as communication degrades.

1 Introduction

In this work we consider the problem of controlling a team of micro-aerial vehicles in a tight formation while performing aggressive maneuvers as an ensemble. There are several challenges that must be addressed as we consider experimental evaluation moving toward real-world application domains. In particular, we are interested in considering the effects of dynamics and control coupled with network time-delays and robot failures on overall system performance. To this end, we propose control laws that enable the system to control along trajectories based on a desired formation shape. The team of robots control as an evolving ensemble with respect to the states of neighboring robots. Individual robot control laws capture the fact that as the system

Matthew Turpin, Nathan Michael, and Vijay Kumar
GRASP Laboratory, University of Pennsylvania, Philadelphia, PA
e-mail: {mturpin,nmichael,kumar}@grasp.upenn.edu

evolves in time neighboring robots may provide varying quality of information for feedback control or disappear completely due to a critical failure. Further, we address the possibility of lossy communications with low-bandwidth and time-delays by requiring that neighboring robots exchange information about planned trajectories over finite time horizons. An experimental evaluation of our approach on team of quadrotors suggests that suitable performance is maintained as the formation motions become increasingly aggressive and communication degrades. Further, critical robots failures that are purposely introduced during experimentation do not significantly impact the capability of other robots to maintain the desired formation shape.

While our general approach is not specific to a particular robot platform, we mainly consider the control of a team of aerial robots. There is extensive literature using a leader-follower formation control approach in the area of multi-robot systems. The problem of controlling a formation of robots to follow a group motion while maintaining a reference shape or structure using only local information and the formulation using control graphs is analyzed in [3]. A distributed controller for trajectory tracking by a team of robots maintaining a rigid virtual structure is discussed in [4]. In [2], an architecture is proposed for precision spacecraft formation control applicable to space-based interferometry for imaging stars. In this approach, the authors detail the use of coordination variables determined by the system state as compared to a desired reference state. The authors of [16] develop stabilizing controllers for driving a formation of robots to rotate, translate, expand, or contract based upon a virtual structure representation. Control of aerial vehicles following the leader-follower framework is also discussed in [6], where experimental results are analyzed to consider performance of the controllers applied to unmanned aerial vehicles.

Our work builds on this literature but differs in many ways. First, we synthesize trajectory planners for each robot based on information on current state and planned trajectory from all neighbors, in addition to addressing feedback controllers as in the above papers. Second, we explicitly model noisy communication and sensing links across the team. Such a formulation permits a discussion of stability and convergence properties of the system to the desired formation shape given the communication and sensing graphs.

Our approach to modifying the underlying feedback control law based on performance shares many similarities to the rich literature where the formation communications and control are formulated as a weighted graph. In addition to considering stability and convergence properties, it is possible to derive conditions that ensure feasible formation structures based on individual robot capabilities [21]. Input-to-state stability of formations is discussed in [22], by considering the construction of graphs from primitives which provide known stability properties. The authors of [5, 17] consider information flow in feedback-based controllers and the consequential effects on the stability of the system to converge to the desired formation. Pertinent to our work is the fact that dynamic formation graphs pose interesting problems when

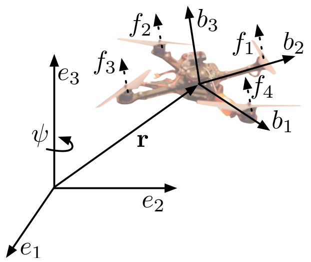


Fig. 1 The vehicle model. The position and orientation of the robot in the global frame are denoted by $\mathbf{r}$ and R, respectively. Control inputs (1) consider both position, $\mathbf{r}$, and the orientation, ψ, of the vehicle about the global z-axis (e_3). Each propeller generates a thrust f_i along the body-fixed z-axis (b_3).

dealing with the switching topology of the graph. In [12], the authors consider the controllability of state-dependent dynamic graphs. Consensus problems on formation graph structures with switching topologies and time-delays are discussed in [18].

The presentation of this work follows. We begin by describing the individual robot dynamic model and control for attitude stabilization and global position in Sect. 2. In Sect. 3, we define the formation as two-part representation that captures the desired ensemble shape and a confidence measure that dictates how each robot looks to other robots to inform their feedback control. Section 4 details control laws for formation control with a team of aerial robots and discusses considerations for improving controller performance given degrading network performance and robot failures. We conclude with an experimental study in Sect. 5 that evaluates performance under varying system conditions.

2 Modeling, Controlling, and Trajectory Generation

In this section we focus on the development of an approach capable of enabling a single quadrotor robot to track a specified 3-D trajectory in $\mathbb{R}^3 \times SO(2)$. To this end, we first develop the dynamic model and propose onboard feedback control for attitude stabilization and position control in $SE(3)$. We continue by describing our approach to generate smooth trajectories and select feedback control inputs to track those trajectories using the onboard feedback control. Much of this presentation follows our previous work [10, 11] and the methods proposed in [8, 9]. As our controllers are nonlinear and do not rely on linearized models of the dynamics, the robots are able to follow trajectories that require large roll and pitch angles that produce high accelerations in the horizontal plane.

Consider the quadrotor robot model shown in Fig. 1 with mass m and rotational inertia $J \in \mathbb{R}^3$. Define the position and rotation of the vehicle in the inertial frame as $\mathbf{r} \in \mathbb{R}^3$ and $R \in SO(3)$, respectively. The angular velocity of the vehicle, $\Omega \in \mathbb{R}^3$, is defined as

$$\dot{R} = R\hat{\Omega}$$

where the operator $\hat{\cdot}$ is defined such that $\hat{x}y = x \times y$ for all $x,\, y \in \mathbb{R}^3$. Given that the i^{th} propeller generates the thrust output f_i as a function of propeller rotational speed, the dynamic model of the vehicle follows:

$$\begin{gathered} m\ddot{\mathbf{r}} = (fR - mg)e_3 \\ J\dot{\Omega} + \Omega \times J\Omega = M \end{gathered}$$

with $e_3 = [0,\, 0,\, 1]^{\mathrm{T}}$, $M = [M_1,\, M_2,\, M_3]^{\mathrm{T}}$, and

$$\begin{bmatrix} f \\ M_1 \\ M_2 \\ M_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ d & 0 & -d & 0 \\ 0 & d & 0 & -d \\ -c & c & -c & c \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

where d is the distance from the robot center of mass to the rotor and c is an aerodynamic drag term that relates differences in propeller speed to yawing moment about the body z-axis.

Due to the design of the system actuation, it is well-known that the quadrotor is underactuated and differentially flat [15]. The four inputs allow us to specify the force along the body-fixed z-axis (b_3) and the three moments in the body-fixed frame. We accordingly choose four output variables and specify the desired trajectory

$$\mathbf{x}_{\mathrm{d}}(t) : [t_0,\, t_f] \to \mathbb{R}^3 \times SO(2)$$

in the time interval $[t_0, t_f]$,

$$\mathbf{x}_{\mathrm{d}}(t) = \begin{bmatrix} \mathbf{r}_{\mathrm{d}}(t) \\ \psi_{\mathrm{d}}(t) \end{bmatrix} = \begin{bmatrix} x_{\mathrm{d}}(t) \\ y_{\mathrm{d}}(t) \\ z_{\mathrm{d}}(t) \\ \psi_{\mathrm{d}}(t) \end{bmatrix} \tag{1}$$

where ψ_{d} is the desired vehicle yaw (rotation about the inertial z-axis).

Following the attitude stabilization approach proposed in [9], we design the force and moment inputs, f and M_i, based on the desired input $\mathbf{x}_{\mathrm{d}}(t)$:

$$\begin{aligned} f &= (-k_{\mathbf{r}} e_{\mathbf{r}} - k_{\dot{\mathbf{r}}} e_{\dot{\mathbf{r}}} + mge_3 + m\ddot{\mathbf{r}}_{\mathrm{d}}) \cdot Re_3 \\ M &= -k_R e_R - k_\Omega e_\Omega + \Omega \times J\Omega \end{aligned} \tag{2}$$

with the error terms defined as follows:

$$
\begin{aligned}
e_{\mathbf{r}} &= \mathbf{r} - \mathbf{r}_{\mathrm{d}} \\
e_{\dot{\mathbf{r}}} &= \dot{\mathbf{r}} - \dot{\mathbf{r}}_{\mathrm{d}} \\
e_R &= \frac{1}{2\sqrt{1 + \mathrm{tr}\left[R_d^{\mathrm{T}} R\right]}} \left(R_{\mathrm{d}}^{\mathrm{T}} R - R^{\mathrm{T}} R_{\mathrm{d}}\right)^{\vee} \\
e_{\Omega} &= \Omega - R^{\mathrm{T}} R_{\mathrm{d}} \Omega_{\mathrm{d}}
\end{aligned}
$$

and dropping the dependence on time for clarity of presentation. The operator $(\cdot)^{\vee}$ is the inverse of the $\hat{\cdot}$ operator such that $(\cdot)^{\vee} : so(3) \to \mathbb{R}^3$. The gains $k_{\mathbf{r}}$, $k_{\dot{\mathbf{r}}}$, k_R, and k_{Ω} are selected to ensure stable performance. See [8, 9] for further explanation of the derivation of these error terms and proofs of stability and convergence of the control system to the desired inputs.

The above attitude stabilization approach operates in $SO(3)$ (as compared to the traditional Euler-angle parameterization approach [13]) and benefits from a stability basin of attraction that includes the full space of rotation matrices (excluding an exact inversion). Note that we assume our trajectory generation scheme plans trajectories with the constraint $f > 0$ and thus define R_{d} with respect to ψ_{d} such that $R_{\mathrm{d}} = [r_1,\, r_2,\, r_3]$ and

$$
\begin{aligned}
r_1 &= r_2 \times r_3 \\
r_2 &= \frac{r_3 \times [\cos\psi_{\mathrm{d}},\, \sin\psi_{\mathrm{d}},\, 0]}{\|r_3 \times [\cos\psi_{\mathrm{d}},\, \sin\psi_{\mathrm{d}},\, 0]\,\|} \\
r_3 &= \frac{-k_{\mathbf{r}} e_{\mathbf{r}} - k_{\dot{\mathbf{r}}} e_{\dot{\mathbf{r}}} + mge_3 + m\ddot{\mathbf{r}}_{\mathrm{d}}}{\| - k_{\mathbf{r}} e_{\mathbf{r}} - k_{\dot{\mathbf{r}}} e_{\dot{\mathbf{r}}} + mge_3 + m\ddot{\mathbf{r}}_{\mathrm{d}}\|}
\end{aligned}
$$

Additionally, the moment stabilization proposed in [9] includes higher-order inertial cancelation terms that we neglect because their effects are insignificant in our experiments.

We now wish to define smooth trajectories and feedback control laws that allow for tracking these trajectories given the inputs defined by (1) and the control in $SE(3)$ (2). Following our previous work [10], we propose an optimization-based methodology that seeks to find smooth trajectories that minimize the k^{th} derivative of the path. We choose to define the desired trajectory as a piecewise-smooth polynomial functions of order n over m time intervals:

$$
\mathbf{x}_{\mathrm{d}}(t) = \begin{cases}
\sum_{i=0}^{n} \alpha_i^1 t^i & t_0 \le t < t_1 \\
\sum_{i=0}^{n} \alpha_i^2 t^i & t_1 \le t < t_2 \\
\quad\vdots & \\
\sum_{i=0}^{n} \alpha_i^m t^i & t_{m-1} \le t < t_m
\end{cases}
$$

with α_i^m denoting the i^{th} polynomial coefficient in the trajectory over the m^{th} time interval. To find the trajectory that minimizes the k^{th} derivative, we consider the following optimization program:

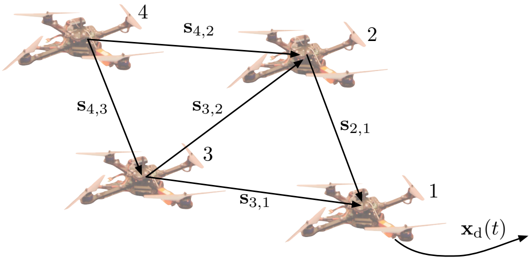


Fig. 2 The formation of quadrotors follow the leader (Robot 1) which tracks a desired trajectory $\mathbf{x}_{\mathrm{d}}(t)$. Each robot controls to maintain a desired shape defined by shape vectors and denoted by $\mathbf{s}_{i,j}$. Note that $\mathbf{s}_{i,j}$ also encodes a desired relative heading between robots (not pictured here).

$$\begin{array}{lll} \text{minimize} & \displaystyle\int_{t_0}^{t_m} \left[\mu_{\mathbb{R}^3} \left\| \frac{d^k \mathbf{r}}{dt^k} \right\|^2 + \mu_\psi \left(\frac{d^k \psi}{dt^k} \right)^2 \right] dt & \\ & \mathbf{x}_{\mathrm{d}}(t_i) = \sum_{j=0}^{n} \alpha_i^j t_i^j & i = 0, \ldots, m \\ \text{subject to} & \frac{d^p \mathbf{r}}{dt^p}\Big|_{t=t_j} = 0 \text{ or free} & j = \{0,\, m\},\, p = 1, \ldots, k \\ & \frac{d^p \psi}{dt^p}\Big|_{t=t_j} = 0 \text{ or free} & j = \{0,\, m\},\, p = 1, \ldots, k \end{array} \qquad (3)$$

where $\mu_{\mathbb{R}^3}$ and μ_ψ are constants that make the integrand non-dimensional. Note that there is some flexibility in the above program in designing the trajectory. For example, we may choose to consider different derivatives of interest for each of the desired inputs (in particular we may consider $k = 4$ for $\mathbf{r}$ while $k = 2$ for ψ). Additionally, we may ignore the endpoint derivative constraints for specific inputs if the application does not require smooth derivatives in those directions. State and input constraints such as limits on angular rates and propellor thrusts can be expressed as algebraic functions of $\mathbf{x}$ and its derivatives, and therefore can be incorporated in this formulation as additional constraints at each intermediate time, t_j. The above optimization is formulated as a quadratic program (QP) with initial conditions and derivative constraints defined as equality and inequality constraints as required. Further details of the methodology including the approach to determining the non-dimensional constants are available in [10].

The trajectory resulting from (3) is parameterized in terms of the coefficients of the sequence polynomial functions, α_i^m, and hence we only require knowledge of these values to fully specify the trajectory.

3 Formation Definition

The shape of a robot formation can be described by specifying a minimal set of relative position vectors and relative bearings. Implicitly this describes all relative positions and bearings. Since in Sect. 2, we specify trajectories in $\mathbb{R}^3 \times SO(2)$ including the three-dimensional position vector and yaw angle, we use a 4×1 *shape vector*:

$$\mathbf{s}_{i,j} = \mathbf{x}_j - \mathbf{x}_i = \begin{bmatrix} x_j - x_i \\ y_j - y_i \\ z_j - z_i \\ \psi_j - \psi_i \end{bmatrix}$$

to describe the relative position and relative heading for each pair of robots (see Fig. 2). In a team of N robots, there are $\frac{N(N-1)}{2}$ such shape vectors satisfying the following properties:

$$\begin{aligned} \mathbf{s}_{i,k} &= \mathbf{s}_{i,j} + \mathbf{s}_{j,k}, \quad \forall i,\, j,\, k \in 1, \ldots, N \\ \mathbf{s}_{i,i} &= [0,\, 0,\, 0]^{\mathrm{T}} \\ \mathbf{s}_{i,j} &= -\mathbf{s}_{j,i} \end{aligned}$$

Thus, if one of the robots (Robot 1) is specified as the leader, the overall shape can be prescribed by a $(N-1) \times (N-1)$ vector-valued, skew-symmetric matrix.

Unlike other lead-follower schemes proposed in the literature, a follower robot uses state information from all other robots that it can “see” or communicate with in its neighborhood. Thus robot i has the ability to use up to $(N-1)$ shape vectors, $\mathbf{s}_{i,j}$, $j \in \mathcal{N}_i$ where $\mathcal{N}_i$ is the neighborhood of robot i which includes at most $(N-1)$ other robots. We assume robot i has an estimate of the shape vector, $\tilde{\mathbf{s}}_{i,j}$, through sensing, or a direct estimate of the partial state of robot j, $\tilde{\mathbf{x}}_j^i$, via communication for all the robots in $\mathcal{N}_i$. Since some of these robots are further away than other robots and the quality of communication or sensing can be vary across pairs of robots, we allow each robot to choose a weight $c_{i,j}$ that reflects its confidence in the estimate $\tilde{\mathbf{s}}_{i,j}$, with the constraint

$$\sum_{j \in \mathcal{N}_i} c_{i,j} = 1 \quad \text{and} \quad c_{i,j} \geq 0 \tag{4}$$

This allows us to define a $N \times N$ *confidence matrix*, $\mathbf{C}$, whose entries are positive and the row sums are all 1. We would like the desired state of robot i to be defined by

$$\mathbf{x}_{i,\mathrm{d}} = \sum_{j \in \mathcal{N}_i} c_{i,j} \left(\tilde{\mathbf{x}}_j^i + \mathbf{s}_{i,j} \right) \tag{5}$$

and derive trajectories and controllers (as in Sect. 2) to drive $\mathbf{x}_i$ to $\mathbf{x}_{i,\mathrm{d}}$. The exception is for the lead robot (Robot 1) whose coefficients are chosen to be $c_{1,1} = 1$ and $c_{1,j} = 0$, $j \neq 1$. Also note that the leader can be a virtual robot or a reference trajectory for the formation thus eliminating a single point of failure.

One natural question is if (5) leads to a consistent specification of a formation. For the remainder of this work, we assume perfect estimation, *i.e.* $\tilde{\mathbf{x}}^i_j = \mathbf{x}_j$ for all i and j. With this assumption, it is straightforward to show by substitution that (3)–(5) yield a set of desired set points $\mathbf{x}_\mathrm{d}$ that are consistent with the specified $(N-1)$ shape vectors, $\mathbf{s}_{i,j}$. By extension, if the trajectory of the lead (and possibly virtual) robot is specified as a function of time, (5) leads to the specification of a consistent trajectory for all of the other robots. The main advantage of this formulation is that it allows each robot to choose a subset of the shape vectors it relies on to calculate its desired trajectory by varying the weights $c_{i,j}$. We note that $\mathbf{C}$ has the property that the sum of the rows equals 1. Schemes like this appear in the consensus literature where the desired heading, velocity or acceleration is obtained by a weighted average of information acquired from the neighbors [7, 24]. However, here our consensus-like rule (5) is used to define trajectories that each robot chooses as explained in the next section.

4 Formation Control

In the previous section we addressed the specification of the formation shape via shape vectors which fully specify the formation shape. Assuming that the shape vectors $\mathbf{s}_{i,j}$ are selected such that they respect the required separation distance between robots to avoid collisions or aerodynamic interactions (see [13] for a discussion of these interactions), we now discuss the computation of individual robot trajectories to drive the robots to the desired formation.

The lead robot (which can be a virtual robot) computes its desired trajectory according to (3) and will follow the trajectory using the methods described in Sect. 2. For the remaining robots, the error between the current system state and the desired state as defined by $\mathbf{s}_{i,j}$ for all i, j is:

$$\mathbf{e}_i(t) = \sum_{j \in \mathcal{N}_i} c_{i,j}(\mathbf{x}_j(t) - \mathbf{x}_i(t) - \mathbf{s}_{i,j}) \tag{6}$$

Note that (6) requires instantaneous knowledge of the current state of the neighboring robots as specified by $\mathbf{C}$. Unfortunately, this information will always be subject to delay due to the fact that it must be communicated over a network.

We propose a receding horizon control approach to address network delays and other sources of information latency. We assume that all robots operate with a synchronized system clock. The methodology outline follows.

- The lead robot computes its desired trajectory and broadcasts to all follower robots a message containing the polynomial coefficients and time intervals that fully specify its trajectory.

- Each follower robot j computes a trajectory for a finite time horizon, t_h, based on the desired shape vectors $\mathbf{s}_{i,j}$ and the information from the leader using the methods described below.
- After initialization, the neighboring robots begin to exchange their locally computed polynomial coefficients and time intervals. Any future control law updates take into account the intended trajectories of neighboring robots and seek to minimize the error between the current and desired formation shape (6).

As the polynomial representation of (3) is analytic and smooth, the robots have access to the intended trajectory of other robots and its derivatives. When communication performance degrades (as discussed in the experiments, Sect. 5), the robots assume that the last trajectory specification from each neighboring robot is still active and chooses controls accordingly. Additionally, when critical failures occur (determined by a long interval without communication), the robots simply modify $c_{i,j}$ accordingly. This approach allows for degrading communication performance such as dropped messages and latency as well as critical robot failures. While our present experimental testbed does not include on board sensing of shape vectors, our formulation allows us to incorporate such information.

To compute the local control of each robot based on the approach outlined above, we begin by redefining the error for robot i from (6) as:

$$\mathbf{e}_i(t) = \sum_{j=1}^{N} c_{i,j} \left(\sum_{k=1}^{n} \left(\alpha_j^k - \alpha_i^k\right) t^k - \mathbf{s}_{i,j} \right)$$

Note that instead of summing only $j \in \mathcal{N}_i$, we sum over all N robots since $c_{i,i} = 0$ and $c_{i,j} = 0$ for robots who are not neighbors. We can now minimize the integral of error squared from the current time t_c through a finite interval $[t_c, t_c + t_h]$:

$$\text{minimize} \quad \int_{t_c}^{t_h} \mathbf{e}_i(t)^{\mathrm{T}} \mathbf{e}_i(t) dt$$

However, since the quadrotor has a relative degree of 4 we choose to add error terms incorporating higher order derivatives to the cost functional:

$$\begin{aligned} &\text{minimize} && \int_{t_c}^{t_h} \left[\mathbf{e}_i(t)^{\mathrm{T}} \mathbf{e}_i(t) + \sum_{j=1}^{k} \kappa_j \mathbf{e}_i^{(j)}(t)^{\mathrm{T}} \mathbf{e}_i^{(j)}(t) \right] dt \\ &\text{subject to constraints on} && \mathbf{x}(t_c), \ldots, \mathbf{x}^{(k)}(t_c) \end{aligned} \tag{7}$$

where κ_i is the weight applied to the i^{th} derivative. This problem is now readily cast as a QP as in (3) and results in minimum error plans for robot i, and by extension, all the robots in the system. We solve (7) as a QP in real time to generate the control inputs

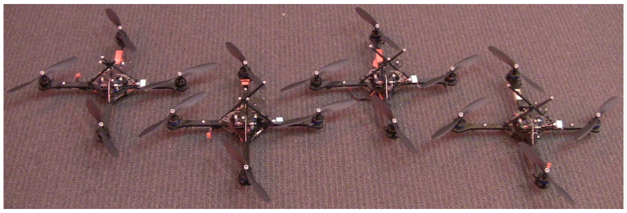


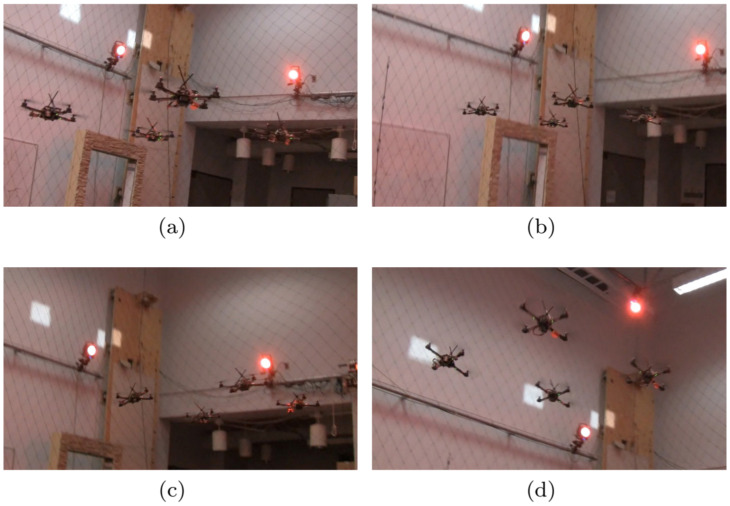
Fig. 3 The team of four quadrotors used in experimentation.

5 Results

In Sects. 2–4, we described our approach to generating trajectories and control for a team of quadrotors to follow a desired trajectory and converge to a desired shape. In this section, we consider the experimental evaluation of these methods. We look at three different studies: the first provides analysis of performance when the team controls along the same desired trajectory but with varying maximum velocities and accelerations (Sect. 5.2). The second study builds upon the first by considering the same system performance but given (emulated) degraded communication between robots (Sect. 5.3). The final study shows performance when a robot undergoes a critical failure such as complete network outage or hardware failure (Sect. 5.4). We begin by detailing the robots, software, and experimental infrastructure leveraged to complete the experiments.

5.1 Implementation Details and Experiment Design

The robots used are sold commercially [1] and follow a standard four-propeller design (Fig. 3). The pose of the quadrotor is observed using a Vicon motion capture system at 100 Hz [23]. The pose is numerically differentiated to compute the linear and angular velocities of the robot. These values are available to Matlab via ROS [19] and a ROS-Matlab bridge [20]. All formation control commands are computed at 20 Hz in Matlab using the latest state estimate at the rate of the Vicon. The commands in Matlab are bridged to ROS where they are interpreted by a finite-state machine (FSM) wich aids in the experimental process [13]. The FSM manages the individual robots, places the robots at the appropriate initial conditions, then relinquishes control to Matlab where the desired trajectory controls are computed based on the methods of Sects. 3–4. The FSM computes the required inputs (Sect. 2) specified by the Matlab trajectory commands and transmits these values to the robot via Zigbee at a fixed rate of 100 Hz. This fixed rate is due to the limited bandwidth of Zigbee (57.6 kbps). These commands are interpreted by the attitude and body-fixed thrust controllers (2) operating on


(a) (b)

(c) (d)

Fig. 4 A team of four quadrotors fly in formation while achieving maximum velocities and accelerations exceeding 3 $\frac{m}{s}$ and 6 $\frac{m}{s^2}$, respectively. Videos of the experiments are available at http://mrsl.grasp.upenn.edu/mturpin/ISRR2011.mov.

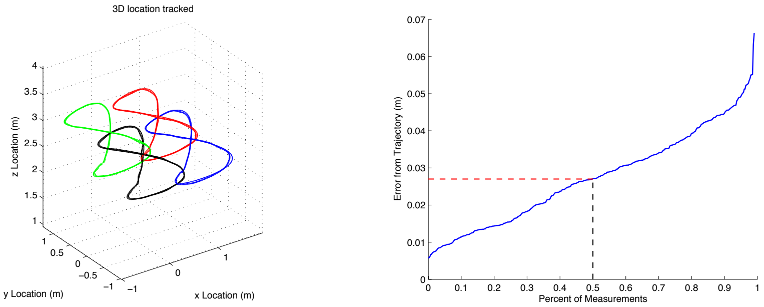
each robot's programmable embedded microprocessor and applied at a 1 kHz update rate.

The experiments emulate decentralized control by computing each robot's control trajectory in MATLAB separately in objects with no global knowledge and only information made available through emulated inter-robot communications. Additionally, imperfect communications are simulated such that individual robot-to-robot messages have time delays and dropped packets as described in Sect. 5.3.

All experimental studies were conducted using formations of four robots with specified constant shape vectors designed such that aerodynamic interaction would not be appreciable up to maximum planar desired accelerations of 6$\frac{m}{s^2}$. Confidence matrices ($\mathbf{C}$) were constructed so that each robot placed equal weight on estimates of any other robot with which it had recent communication.

5.2 Performance with Variable Maximum Velocity and Acceleration

To evaluate the performance of our methods controlling on demanding trajectories, the leader first optimized its trajectory over a specified set of 8 waypoints or key frames for each trial. These key frames in $\mathbb{R}^3 \times SO(2)$ were carefully chosen to give a variety of motions with high speeds and high accel-

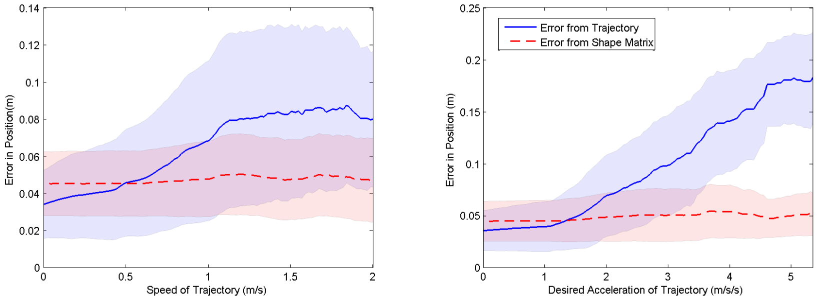


(a) (b)

Fig. 5 The trajectories traversed by each robot in the formation where thin lines show the desired trajectory and bold lines indicate the actual trajectory (Fig. 5(a)). Note that the desired and actual trajectories are sufficiently close that only the actual trajectory is readily visible. The inverse cumulative probability distribution function for the error in $\mathbb{R}^3$ for the lead robot while tracking the waypoints used for the performance study (Fig. 5(b)). Half of all measurements are within 2.7 cm of the desired trajectory.

erations in all dimensions. The minimum possible distance (in $\mathbb{R}^3$) traveled by the leader to reach all 8 waypoints is 5.0 meters, but the actual distance traveled is longer due to the smoothness conditions imposed. Time scaling from [10] was used to vary the time allowed between waypoints and indirectly affect the velocities and accelerations demanded of the quadrotors. Trials were run at 32, 24, 16, 12, and 10 seconds to reach all 8 waypoints. See Fig. 5(a) for a plot of the trajectories tracked by the quadrotors for the trial requiring 16 seconds. To understand how well each robot is tracking its desired trajectory, see Fig. 5(b) for the cumulative distributive function of the error in $\mathbb{R}^3$ for the lead robot. All follower robots have similar errors to the lead robot for every trajectory tracked in every trial run. Figure 4 displays snapshots of a fast trajectory (with accelerations of greater than 0.5 g) being tracked highlighting that even during large deviation from near hover operation, all robots remain in close formation.

As the speeds of the trajectories are increased, we expect error from the desired trajectory to increase as time delays and modeling errors of quadrotor parameters become more significant. Datasets from all trials in this performance study were merged to analyze how well the robots maintained formation in demanding situations. Every data point collected was analyzed to relate speed and magnitude of desired acceleration to error from desired trajectory and error from formation with all other robots. The mean and variance of this data are plotted in Fig. 6. As one might expect, it is clear that as speed and acceleration increase, the error from the desired trajectory increases substantially. However, the shape error remains relatively constant across all tested speeds and accelerations. This resilience to relative forma-



(a) (b)

Fig. 6 As the robots' speed and acceleration increase, the shape errors (deviations from the entries in the specified shape vectors) remain relatively constant. Each plot depicts the mean squared error (solid and dashed lines) and standard deviation in error (filled regions) between the desired and actual robot trajectories (solid, blue) and shape (dashed, red) using data collected from many trials.

tion error during aggressive maneuvers confirms that our methodologies are sound and will ensure collision avoidance of robots in the formation.

5.3 Performance with Degraded Network Communication

This study is concerned with the practical complication of communication and sensing latencies, errors, and failures. To study how packet loss affects formation errors, we simulate various levels of failure of interagent communication and measure formation response. Each robot to robot communication is allowed to be transmitted successful with probability ρ. For example, a value of $\rho = 0.5$ signifies on average one out of every two attempted communications is successful. A large initial error trajectory was utilized to clearly demonstrate the decay to the desired shape. Figure 7 shows trials with four values of ρ ranging from 0.4 to 1.0. The slight initial offsets between trials are a result of different starting conditions between trials. As expected, the same decay rate is experienced regardless of the rate of communications failure. These results confirm that for even very unreliable networks, our controllers will converge to the desired shape.

5.4 Performance with Critical Robot Failures

To simulate a complete robot failure, at a prescribed time t_f, Robot 2 ceases all communications and safely moves away from the remaining formation.

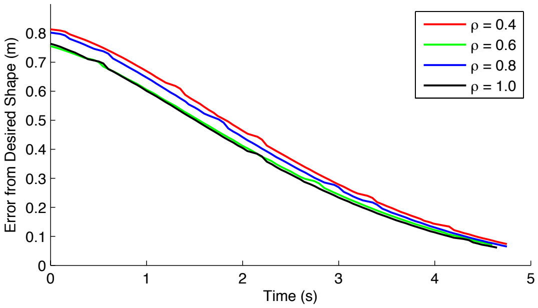
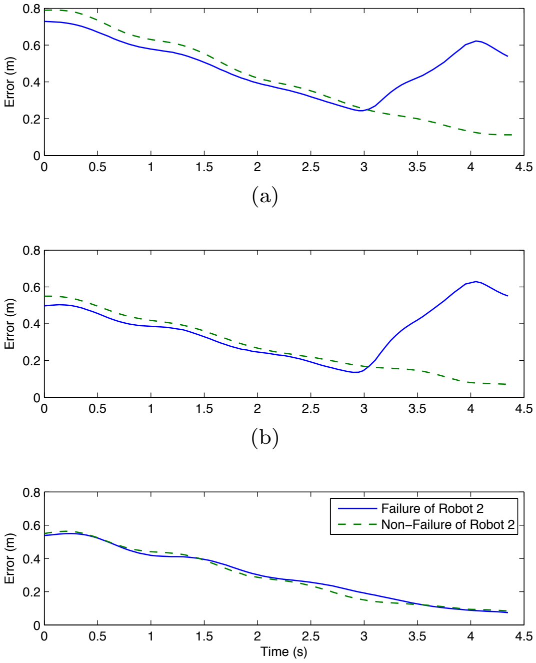


Fig. 7 Convergence of robots to the desired shape with simulated communication failures. ρ is the probability that communications are successful.

The remaining agents initially atribute the lack of new information from Robot 2 to a temporary communication failure and continue planning based on the most recent trajectory robot 2 transmitted. Over time, the remaining robots recognize that the data from Robot 2 is too outdated to continue planning with and update their respective entries in the confidence matrix to reflect this fact. The same trial was also conducted with all robots functional throughout for comparison. The complete robot failure and baseline non-failure cases are compared in Fig. 8. The interagent distances between all remaining robots continue to decay at the same rate whether or not there is a robot failure. As long as failing robots do not physically disable other robots in a crash, our controller is designed to recognize the failure and automatically compensate without any formation tracking quality loss.

6 Conclusion and Future Work

In this work we consider the problem of controlling a team of micro-aerial vehicles moving quickly through a three-dimensional environment while maintaining a tight formation. Our approach is based on the leader-follower control paradigm with the follower robots controlling to maintain a desired formation shape with respect to a leader that is possibly virtual. We define the formation using a set of shape vectors that describe the desired ensemble shape and a confidence matrix that dictates how much each robot relies on other robots' state information for its trajectory generation and control. Based on the current and desired formation state and plans according to neighboring robots, each robot computes a trajectory and controls along that trajectory to ensure that the system converges to the desired state. Unlike previous research on leader-follower control schemes or consensus algorithms, our work addresses the synthesis of trajectories for multiple robots satisfying dynamic



(a)

(b)

(c)

Fig. 8 A simulated failure case of Robot 2 at $t_s = 3$ seconds with non-trivial starting error from $\mathbf{s}$. The interagent error from $\mathbf{s}$ between robot pairs {1, 2}, {2, 3}, and {1, 3} are shown in Figs. 8(a)–8(c), respectively.

constraints and consensus across trajectories. An experimental evaluation of our approach on a team of quadrotors suggests that suitable performance is maintained as the group motions become increasingly aggressive even as communication degrades.

We are interested in pursuing this research further particularly as we wish to apply these methods on teams of flying robots outdoors. At present, we are unable to make strong comments on the convergence and stability of the formation with a time-varying desired shape, $\mathbf{s}(t)$, or confidence matrix, $\mathbf{C}(t)$. However, the ability to smoothly change shapes suggests that we can enable robot teams to switch between proximity graphs that are non isomorphic in a smooth fashion. Further, we wish to consider how to design shape vectors to achieve desired formations based on the quality of network communications, state estimation, and vehicle capabilities.

We are actively moving our implementation from MATLAB to C++ to run locally on each robot. We are also pursuing estimation methods for outdoor operation to eliminate the requirement of the VICON system for experimentation [14].

References

1. Ascending Technologies, GmbH. `http://www.asctec.de`
2. Beard, R.W., Lawton, J., Hadaegh, F.Y.: A coordination architecture for spacecraft formation control. IEEE Trans. Control Syst. Technology **9**(6), 777–790 (2001)
3. Desai, J.P., Ostrowski, J.P., Kumar, V.: Modeling and control of formations of nonholonomic mobile robots. IEEE Trans. Robot. **17**(6), 905–908 (2001)
4. Egerstedt, M., Hu, X.: Formation constrained multi-agent control. IEEE Trans. Robot. Autom. **17**(6), 947–951 (2001)
5. Fax, J.A., Murray, R.M.: Information flow and cooperative control of vehicle formations. IEEE Trans. Autom. Control **49**(9), 1465–1476 (2004)
6. Gu, Y., Seanor, B., Campa, G., Napolitano, M.R., Rowe, L., Gururajan, S., Wan, S.: Design and flight testing evaluation of formation control laws. IEEE Trans. Control Syst. Technol. **14**(6), 1105–1112 (2006)
7. Jadbabaie, A., Lin, J., Morse, A.S.: Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Trans. Autom. Control **48**(6), 988–1001 (2003)
8. Lee, T.: Geometric tracking control of the attitude dynamics of a rigid body on SO(3). In: Proc. of the Amer. Control Conf. San Francisco, CA (2011)
9. Lee, T., Leok, M., McClamroch, N.H.: Geometric tracking control of a quadrotor UAV on SE(3). In: Proc. of the IEEE Conf. on Decision and Control. Atlanta, GA (2010)
10. Mellinger, D., Kumar, V.: Minimum snap trajectory generation and control for quadrotors. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom. Shanghai, China (2011). To Appear
11. Mellinger, D., Michael, N., Kumar, V.: Trajectory generation and control for precise aggressive maneuvers with quadrotors. In: Proc. of the Intl. Sym. on Exp. Robot. Delhi, India (2010)
12. Mesbahi, M.: On state-dependent dynamic graphs and their controllability properties. IEEE Trans. Autom. Control **50**(3), 387–392 (2005)
13. Michael, N., Mellinger, D., Lindsey, Q., Kumar, V.: The GRASP multiple micro UAV testbed. IEEE Robot. Autom. Mag. **17**(3), 56–65 (2010)
14. Michael, N., Stump, E., Mohta, K.: Persistent surveillance with a team of MAVs. In: Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. San Francisco, CA (2011). Submitted
15. Nieuwstadt, M.J.V., Murray, R.M.: Real-time trajectory generation for differentially flat systems. Intl. J. Robust and Nonlinear Control **8**(11), 995–1020 (1998)
16. Ogren, P., Fiorelli, E., Leonard, N.: Formations with a mission: stable coordination of vehicle group maneuvers. In: Proc. of Intl. Sym. on Mathematical Theory Networks and Syst. Notre Dame, IN (2002)
17. Olfati-Saber, R., Murray, R.M.: Distributed cooperative control of multiple vehicle formations using structural potential functions. In: Proc. of the IFAC World Congress. Barcelona, Spain (2002)
18. Olfati-Saber, R., Murray, R.M.: Consensus problems in networks of agents with switching topology and time-delays. IEEE Trans. Autom. Control **49**(9), 1520–1533 (2004)
19. Robot Operating System (ROS). `http://www.ros.org`
20. ROS-Matlab Bridge. `http://github.com/nmichael/ipc-bridge`
21. Tabuada, P., Pappas, G.J., Lima, P.: Feasible formations of multi-agent systems. In: Proc. of the Amer. Control Conf., pp. 56–61. Arlington, VA (2001)
22. Tanner, H., Pappas, G.J., Kumar, V.: Input-to-state stability on formation graphs. In: Proc. of the IEEE Intl. Conf. on Robot. and Autom., pp. 2439–2444. Las Vegas, NV (2002)
23. Vicon Motion Systems, Inc. `http://www.vicon.com`
24. Vicsek, T., Czirók, A., Ben-Jacob, E., Cohen, I., Shochet, O.: Novel type of phase transition in a system of self-driven particles. Physical Review Letters **75**(6), 1226–1229 (1995)