Safe and Robust Quadrotor Maneuvers Based on Reach Control

Marijan Vukosavljev, Ivo Jansen, Mireille E. Broucke, and Angela P. Schoellig

Abstract— **In this paper, we investigate the synthesis of piecewise affine feedback controllers to execute safe and robust quadrocopter maneuvers. The methodology is based on formulating the problem as a reach control problem on a polytopic state space. Reach control has so far only been developed in theory and has not been tested experimentally in a real system before. We demonstrate that these theoretical tools can achieve aggressive, albeit safe and robust, quadrocopter maneuvers without the need for a predefined open-loop reference trajectory. In a proof-of-concept demonstration, the reach controller is implemented in one translational direction while the other degrees of freedom are stabilized by separate controllers. Experimental results on a quadrocopter show the effectiveness and robustness of this control approach.**

I. Introduction

This paper proposes a novel framework for control of complex quadrocopter maneuvers that simultaneously impose requirements of safety, fast response, and a desired sequence of events. We apply the framework to a simple side-to-side maneuver in order to expose the main features of the framework. The framework is based on using hybrid systems, event-based switching, and solving a collection of so-called reach control problems (RCP). RCP has an extensive theoretical development, see [10]–[13], [17], but is completely lacking in experimental validation. Our primary goal is to illustrate, for the first time ever on a real system, the various strengths offered by the reach control approach. For our chosen maneuver we demonstrate the ability of our approach to carry out complex control specifications, and additionally show that it outperforms a standard tracking approach, particularly in the presence of unmodeled disturbances. Finally, the significance of the method is the ability to produce a feedback controller without the need to specify either a timed reference trajectory for tracking, or a spacial path to stabilize to in closed-loop (i.e. path following), as is required in the majority of the literature.

A significant effort in recent years has been devoted to aggressive manuevering, particularly of aerial vehicles. In [3], an impressive collection of aggressive quadrocopter maneuvers is featured. The typical approach is to use a cascaded set of loop-shaping feedback controls to track given reference position and yaw trajectories. In [1], maneuvers are first geometrically specified by splines, are then time parameterized so that the resulting reference trajectory is feasible, and are finally executed by tracking these references, achieving high precision due to an iterative learning algorithm that systematically rejects disturbances. While these methods can provide high performance maneuvers, due to the open-loop nature of the reference signals, any additional unaccounted disturbances can quickly deteriorate performance.

The other common approach to quadrocopter control involves path following, as in [6]–[8]. Under nominal conditions, timed trajectory tracking and path following perform comparably well. The difference is in how they respond to offsets. Trajectory tracking generates transients when the system tries to catch up to the reference; this may result in severe distortion from the original maneuver. Path following generates transients as the system converges to the nearest point on the path; this method better preserves the original maneuver but not its timing. We mention that point stabilization is the simplest type of path following, but can also have unpredictable transients if the system is too far away from the desired point. A more practical variation is shown in [5], where desired points are stabilized via LQR in succession, determined by working backwards from the goal state.

The typical approaches to path following are demonstrated in [6], [7], where small angle assumptions are imposed to achieve path following using linear PD or PID control. In [8], a path following approach using the full nonlinear geometric theory with dynamic compensation allows one to follow a prespecified Jordan curve along with a desired velocity and yaw profile. This is a powerful method, but still requires the design of the path itself, which may not be an easy task for more complex maneuvers. Our reach control approach offers similar benefits over timed trajectory tracking, but instead requires one to specify only the desired states of operation as polytopic regions in the state space and the overall direction of motion. The controller synthesis then yields a closed-loop system that effectively produces its own path that is by design within the prespecified operational envelope.

A similar method to the reach control approach is [9], incorporating the concepts of hybrid systems and guaranteed operational safety to achieve quadrocopter flips and attain collision avoidance between multiple vehicles. Safety is addressed via reachability analysis, based on numerically solving a Hamilton-Jacobi equation. This formulation is for nonlinear systems with disturbance inputs, which is more general than the reach control approach, but it is considerably more computationally involved. The reach control approach

Marijan Vukosavljev and Mireille E. Broucke are with the Dept. of Electrical and Computer Engineering, University of Toronto, Canada (e-mails: mario.vukosavljev@mail.utoronto.ca, broucke@control.utoronto.ca). Ivo Jansen was a visiting student from the Dept. of Mechanical Engineering, Eindhoven University of Technology, the Netherlands (email: i.h.m.jansen@student.tue.nl). Angela P. Schoellig is with the University of Toronto Institute for Aerospace Studies (UTIAS), Canada (email: schoellig@utias.utoronto.ca). Supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

Associated video at: http://tiny.cc/quadrotorRCPx.

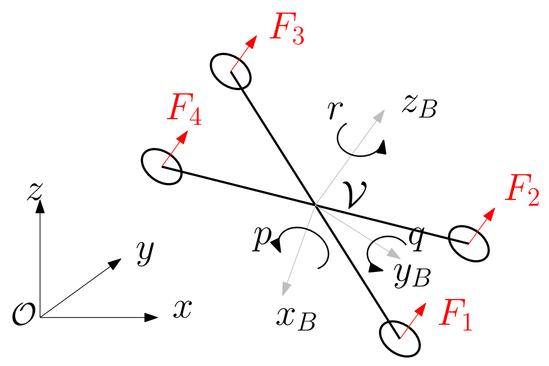


Fig. 1. The inertial and quadrocopter body-fixed frames $\mathcal{O}$ and $\mathcal{V}$. The quadrocopter is actuated by varying the thrusts F_i, $i \in \{1,2,3,4\}$ produced by each motor. This results in changes to its body rotation rates, (p,q,r) and vertical acceleration, which then causes a change to the quadrocopter's position and attitude.

works with a polytopic state space for affine systems, which for our maneuver results in a design that can essentially be computed by hand. Unfortunately both reach control and numerical reachability analysis quickly become intractable as the system dimensionality grows. While [9] includes successful experimental testing, reach control has never been applied on a real system before. This work presents a proof-of-concept of the practical applicability of reach control.

II. Quadrocopter Model

The quadrocopter model is ubiquitous in the literature; see, for example, [2], [3] or Chapter 4 of [4]; we refer the reader to those references for details. The dynamics are described by six degrees of freedom. The translational position (x,y,z) is measured in the inertial coordinate system $\mathcal{O}$ as shown in Figure 1. The vehicle attitude is defined by the body-fixed frame $\mathcal{V}$ and is represented by the ZYX-Euler angles, yaw, pitch, and roll, (ψ,θ,ϕ). The full state of the vehicle additionally includes the translational and rotational velocities of the body frame, $(\dot{x},\dot{y},\dot{z})$ represented in $\mathcal{O}$ and (p,q,r) represented in $\mathcal{V}$, respectively.

We employ a controller architecture depicted in Figure 2. We concentrate on the synthesis of the feedback controller for the x-direction, while a standard, nonlinear tracking controller (as, for example, proposed in [3]) is used for stabilizing the y- and z-coordinates of the vehicle as well as the yaw. In our control architecture (Figure 2), we assume that the full state of the vehicle is measured. An onboard controller takes desired pitch angle θ_d, roll angle ϕ_d, the angular body velocity around the body's z-axis r_d, and the vertical velocity of the vehicle $\dot{z}_d$ as inputs and calculates the required motor forces $F_{i,d}$, $i \in \{1,2,3,4\}$.

We assume that the nonlinear controller successfully stabilizes the vehicle at $y_d(t) = y_{des}$, $z_d(t) = z_{des}$, and $\psi_d(t) = 0$, $y_{des}, z_{des} \in \mathbb{R}$, and provides the onboard controller inputs $\phi_d(t)$, $r_d(t)$, and $\dot{z}_d(t)$. The equations governing the x- and

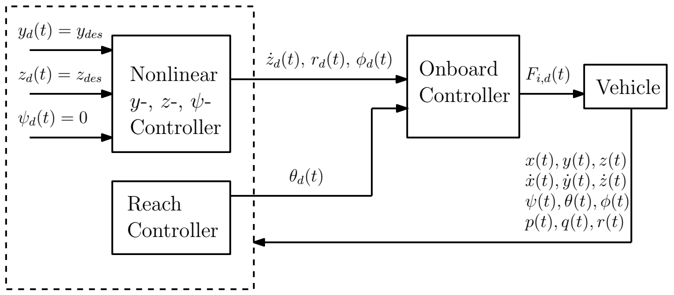


Fig. 2. The control architecture.

z-motion of the vehicle are then given by

$$\ddot{x}(t) = f(t)\sin(\theta(t)) \tag{1}$$
$$\ddot{z}(t) = f(t)\cos(\theta(t)) - g, \tag{2}$$

where g is the gravitational constant and $f(t)$ is the collective thrust normalized by the vehicle mass m,

$$f(t) = \frac{1}{m}\sum_{i=1}^{4} F_i(t), \tag{3}$$

with motor forces F_i, $i \in \{1,2,3,4\}$, see Figures 1 and 2.

Since $z(t) = z_{des}$ implies $\ddot{z}(t) = 0$, equation (2) gives $f(t) = g/\cos(\theta(t))$, and with (1) we have

$$\ddot{x}(t) = g\tan(\theta(t)) := u(t) \;\Leftrightarrow\; \theta(t) = \tan^{-1}\left(\frac{u(t)}{g}\right). \tag{4}$$

If we can define an x-acceleration profile, then with $u(t) = \ddot{x}(t)$ and (4) this will produce a desired pitch angle signal $\theta_d(t)$ for the onboard controller. The signal $u(t)$ will be constructed via the reach control and this will yield a feedback $u(t) = u(x(t), \dot{x}(t))$. The details are given in Section IV.

III. Control Specifications

The primary objective is to transport a quadrocopter between two ends of a room repeatedly. For simplicity of exposition, this side-to-side motion is along the x-axis of the inertial frame $\mathcal{O}$ and the yaw angle, ψ, is zero. This maneuver is decomposed into two opposing discrete modes: one in which the quadrocopter is moving from left to right, referred to as the L2R; and one in which the quadrocopter is moving from right to left, denoted by R2L. For each mode, we define a threshold distance that must be exceeded in order to transition to the opposite mode. During operation, the boundaries of the room are to be strictly adhered to in order to avoid collision. In addition, we specify a maximum speed to ensure the behavior is not too aggressive. Finally, a minimum speed is specified to encourage the quadrocopter to move with reasonable speed throughout each mode. In the y- and z-directions, the quadrocopter should remain stabilized at some nominal values.

The control specifications can be formalized in terms of the states of the system. Since this maneuver is executed in the x-direction, we formulate these specifications in terms of x and $\dot{x}$. Let the middle of the room be at the origin

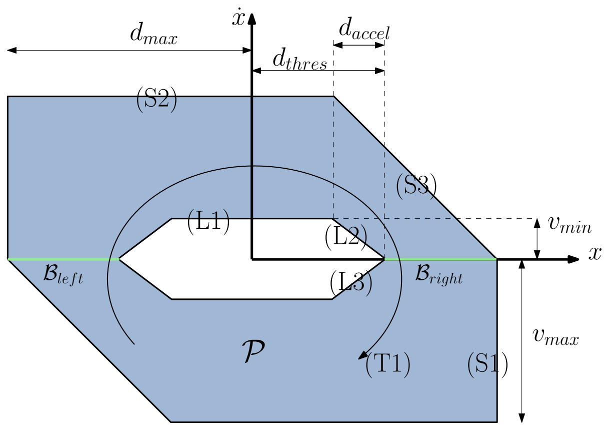


Fig. 3. Desired maneuver envelope (in blue). The specifications (S1-3), (L1-3), and (T1) are defined in Section III. The green lines represent the corresponding crossing sets that define when the system switches from the left-to-right to the right-to-left mode and vice versa.

of the inertial frame. We define a set of variables that completely parameterize the maneuver envelope, see Figure 3. For simplicity, the maneuver is symmetric about the origin so that the two modes are oddly symmetric. The distance from the origin to a room boundary is d_{max}. The maximum speed is v_{max}. The threshold distance as measured from the origin is d_{thres}. The encouraged minimum speed is v_{min}. Finally, a distance parameter d_{accel} is defined to help describe how quickly the quadrocopter should accelerate near the threshold point.

a) Safety Specifications: To remain within the boundaries of the room, we require $|x| \leq d_{max}$, see Figure 3. The maximum speed limitation imposes $|\dot{x}| \leq v_{max}$. For a safe turnaround, we impose a deceleration requirement close to the room's boundaries. For compatibility with the RCP approach we use linear inequalities. We define a safe, minimum deceleration, a_{saf}, and obtain the following linear inequalities: $|x - \dot{x}/a_{saf}| \leq d_{max}$, where we choose $a_{saf} = -v_{max}/(d_{max} - d_{thres} + d_{accel}) < 0$, see Figure 3. For safety to be ensured, all three inequalities must be simultaneously satisfied at all times:

(S1) Position: $|x| \leq d_{max}$.
(S2) Speed: $|\dot{x}| \leq v_{max}$.
(S3) Deceleration: $|x - \dot{x}/a_{saf}| \leq d_{max}$.

b) Liveness Specifications: Next we describe the so-called liveness specifications. Their purpose is to make the maneuver more aggressive, so that around the middle of the room the quadrocopter moves with sufficient velocity. It is most easily visualized as a deleted region centered at the origin, see Figure 3. There are many ways to construct such a region. The only requirement is that it is contained strictly within the safety region described above. Using our parameters as defined in Figure 3, we first define the inequality $|\dot{x}| \geq v_{min}$, which says the speed should be above the minimum. Next we define $|x - \dot{x}/a_{liv}| \geq d_{thres}$, and $|x + \dot{x}/a_{liv}| \geq d_{thres}$, with $a_{liv} = -v_{min}/d_{accel} < 0$. These inequalities describe how the quadrocopter should accelerate from zero speed to the minimum speed and decelerate from the minimum speed to zero speed, respectively. For liveness to be ensured, any of the three inequalities must be satisfied at all times:

(L1) Speed: $|\dot{x}| \geq v_{min}$.
(L2) Acceleration: $|x - \dot{x}/a_{liv}| \geq d_{thres}$.
(L3) Deceleration: $|x + \dot{x}/a_{liv}| \geq d_{thres}$.

c) Desired Temporal Sequence: Finally we discuss the desired temporal sequence. For the side-to-side motion to be executed correctly, the system must transition to the opposite mode only under the correct transition criteria; namely, it reaches the specified threshold distance. More rigorously, we define the right- and left-side crossing sets as

$$\mathcal{B}_{right} = \{(x, \dot{x}) \mid x \in [d_{thres}, d_{max}],\ \dot{x} = 0\}, \tag{5}$$

$$\mathcal{B}_{left} = \{(x, \dot{x}) \mid x \in [-d_{max}, -d_{thres}],\ \dot{x} = 0\}. \tag{6}$$

The sets are shown in Figure 3 (green lines). Also, in the context of dynamical systems, we refer to the set of points in the state space that comprise a given system's trajectory in time as its *orbit*. Assuming that the initial mode be L2R, the system transitions to the R2L mode when the quadrocopter orbit crosses $\mathcal{B}_{right}$. Then the system transitions back to the L2R mode when the quadrocopter orbit crosses $\mathcal{B}_{left}$. This can be repeated indefinitely. The corresponding temporal specification is:

(T1) The quadrocopter's orbit must cross the sets $\mathcal{B}_{right}$ and $\mathcal{B}_{left}$ in exactly the following sequence: $\mathcal{B}_{right}$, $\mathcal{B}_{left}$, $\mathcal{B}_{right}$, $\mathcal{B}_{left}$, $\ldots$.

d) Summary: Collecting all the above inequalities, we can formally describe the region of the state space where the state must reside in order to simultaneously meet the safety and liveness requirements (blue shaded region in Figure 3. Since the inequalities are linear, this defines a (non-convex) polytope, denoted as $\mathcal{P}$. For all the specifications to be satisfied, we require the following logical statement to be true at all times:

$$[(\text{S1}) \wedge (\text{S2}) \wedge (\text{S3})] \wedge [(\text{L1}) \vee (\text{L2}) \vee (\text{L3})] \wedge (\text{T1}). \tag{7}$$

IV. Reach Control Concept

To satisfy the specifications from Section III, we use the reach control approach, see [11], [17], in order to determine a suitable $\theta_d(t) = \tan^{-1}(u(t)/g)$. We construct $u(t) = u(x(t), \dot{x}(t))$ as a feedback using the current x- and $\dot{x}$-measurement so that the closed-loop dynamics for the nominal equations of motion in the $(x\text{–}\dot{x})$-plane (4) guarantee safety, liveness, and the desired temporal sequence. From Section III, the set $\mathcal{P}$ determines the allowable $(x, \dot{x})$ states. We partition $\mathcal{P}$ into a set of regions for which we each specify a controller. Since $\mathcal{P}$ is a non-convex polytope, we employ a triangulation of $\mathcal{P}$ into simplices, cf. Figure 5.

An n-dimensional *simplex* $\mathcal{S} := \text{co}\{v_0, \ldots, v_n\}$ is the convex hull of $(n+1)$ affinely independent points in $\mathbb{R}^n$. For $n = 2$, as we consider it here, it is simply a triangle. A *facet* of a simplex is a boundary face of dimension $(n-1)$. A

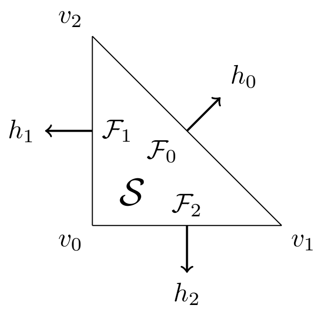


Fig. 4. Two-dimensional simplex and related terminology.

triangulation is a partition of a set $\mathcal{P} \subset \mathbb{R}^n$ into p simplices and is denoted as $\mathbb{T} = \{\mathcal{S}_1, \ldots, \mathcal{S}_p\}$, see [15]. Then $\mathbb{T}$ satisfies the properties:

(i) $\mathbb{T} = \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_p$ and
(ii) $\mathcal{S}_i \cap \mathcal{S}_j, i \neq j$, is a lower dimensional simplex of both $\mathcal{S}_i$ and $\mathcal{S}_j$ or the empty set $\forall i, j \in \{1, \ldots, p\}$.

Once a triangulation of $\mathcal{P}$ has been specified, the next step of the design is to identify a sequence of simplices to be visited by trajectories in order to "play out" the desired temporal sequence. Since the reduced nominal dynamics (4) are that of a double integrator, it is clear that the drift vector field mandates that the motion is clockwise on $\mathcal{P}$.

Using the sequence of simplices, *exit facets* for each simplex are designated. The trajectories starting in the given simplex may only exit the simplex through the exit facets, while the remaining facets act as *restricted facets*.

Finally, controllers are designed for each simplex based on the *reach control problem* (RCP). The reach control problem formulation, its theoretical developments, and conditions for solvability are discussed in [10]–[13], [17]. The main purpose of RCP is to guide trajectories of a dynamical system through a specific region of the state space defined by safety and performance requirements. Ultimately, the trajectories should reach a target set of states in finite time without violating the boundaries of the specified region.

Below we summarize the procedure for determining a controller over an arbitrary simplex in our triangulation. For the double integrator model, we have

$$\dot{s} = As + Bu = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} s + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u, \tag{8}$$

where $s := (x, \dot{x})$, whose components are the x-position and x-velocity of the quadrocopter, respectively. We consider an arbitrary simplex and define $\mathcal{I}_3 = \{0, 1, 2\}$ to be the index set for the three vertices of the simplex. The 2D coordinates of each vertex are denoted as $v_i \in \mathbb{R}^2$, $i \in \mathcal{I}_3$, see Figure 4. For each of the vertices, v_i, we must pick a corresponding $u_i \in \mathbb{R}$. In the sequel, we use the term *control value* to refer to the control assignments u_i specifically at the vertices v_i of $\mathcal{P}$. Each facet, $\mathcal{F}_i$, of the simplex is indexed by the vertex index that it does *not* contain, and each facet has an associated normal vector h_i to describe its orientation, see Figure 4. We assume that there is at least one restricted facet, and index the restricted facets using $\mathcal{I}_r \subset \mathcal{I}_3$.

To solve RCP over this simplex, we must pick the u_i to satisfy the so-called *invariance conditions* [17]; that is,

$$(\forall i \in \mathcal{I}_3)(\forall j \in \mathcal{I}_r \backslash \{i\}) \;\; h_j \cdot (Av_i + Bu_i) \leq 0. \tag{9}$$

This essentially encodes that with an appropriately chosen control at each vertex, the velocity vector at each vertex points in the right direction so that trajectories leave the simplex through an exit facet while avoiding crossing the restricted facets. If the inequalities in (9) are satisfied for each u_i, we can construct an affine feedback to be used over the entire simplex:

$$u(t) = K_c s(t) + g_c. \tag{10}$$

Upon choosing u_i that solve (9), the controller gains K_c and g_c are obtained using

$$\begin{bmatrix} K_c^\top \\ g_c \end{bmatrix} = \begin{bmatrix} v_0^\top & 1 \\ v_1^\top & 1 \\ v_2^\top & 1 \end{bmatrix}^{-1} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \end{bmatrix}. \tag{11}$$

The final step is to check that the closed-loop system, $\dot{s} = (A + BK_c)s + Bg_c$, contains no equilibrium in the simplex. Generally, this complicates the usage of RCP, but for our double integrator model, it is straightforward as equilibria can only occur along the x-axis.

The resulting control law is a piecewise affine feedback with switching between controllers occurring at the boundaries between contiguous simplices.

V. Reach Controller Design

In this section, we present our complete design (including numerical values) for the reach controllers on the polytope $\mathcal{P}$. To proceed, we fix the values of the motion parameters from Section III to: $d_{max} = 2.5$ m, $d_{thres} = 1.5$ m, $d_{accel} = 0.3$ m, $v_{max} = 2$ m/s, and $v_{min} = 0.6$ m/s.

A. Triangulation and Exit Facets

Our triangulation is shown in Figure 5. The vertices of the triangulation are uniquely labeled as $\hat{v}_i$, $i \in \mathcal{I}_v := \{1, \ldots, 16\}$, and their coordinates are chosen in terms of the motion parameters. The simplices are uniquely labeled as $\mathcal{S}_i$, $i \in \mathcal{I}_s := \{1, \ldots, 20\}$. The triangulation was naively generated by manually partitioning $\mathcal{P}$ into simplices and using intuition from a previous simulation design carried out in [16] on a similar polytope also for a double integrator system. Automated procedures for triangulating and solving RCP in conjunction are considered in [14], but are unnecessarily complicated for our 2D polytope $\mathcal{P}$.

The exit facets are normally specified so that the complete desired temporal sequence is enforced throughout $\mathcal{P}$ simultaneously; that is, the simplices above the x-axis cause the motion L2R and the simplices below cause the motion R2L. However, for practical purposes, we follow an unconventional approach and treat the discrete modes L2R and R2L individually, resulting in two separate closed-loop systems. Such a

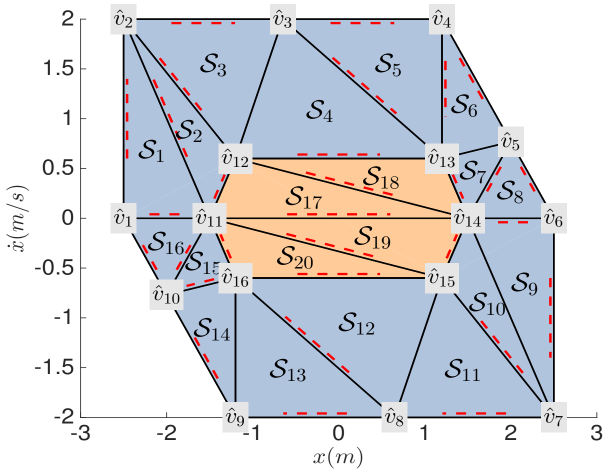


Fig. 5. The triangulation of $\mathcal{P}$, also showing exit facets. Red dashed lines represent the restricted facets of a given simplex, drawn inside of the respective simplex. The blue shaded area represents $\mathcal{P}$. The orange shaded area is guaranteeing the liveness of the system and must be avoided in nominal operating conditions.

design increases the robustness of the system as it enables a system to continue its L2R or R2L motion even under severe disturbances, which may move the system outside of $\mathcal{P}$ momentarily (for example, by increasing or decreasing the velocity). In technical terms, this means that the desired temporal sequence is preserved in the presence of unmodeled disturbances.

For the L2R mode, the exit facets for all simplices in $\mathcal{P}$ are chosen to cause the overall motion to repeatedly reach and cross the set $\mathcal{B}_{right}$. Then due to symmetry about the origin, the behavior for the R2L mode is chosen to be oddly symmetric of L2R. If the problem was not symmetric, we would need separate controller designs for each mode defined over $\mathcal{P}$, including perhaps separate triangulations. But because of symmetry, we only need one set of controllers defined over $\mathcal{P}$. We use odd symmetry on the L2R mode controllers to yield the R2L mode controllers. Referring to our triangulation in Figure 5, for the L2R mode, the nominal sequence of simplices is $\mathcal{S}_1$, $\mathcal{S}_2$, …, $\mathcal{S}_8$, $\mathcal{S}_9$, where in $\mathcal{S}_9$, we switch to the R2L mode and use the oddly reflected controller of $\mathcal{S}_1$ and so on. We ensure that the controllers give a continuous control law at the transition points between modes, which are at the sets $\mathcal{B}_{right}$ and $\mathcal{B}_{left}$. Given the symmetry, we focus only on the L2R mode for the remainder of this section.

To aid the robustness of our design, two additional features are implemented that are unconventional for reach control. Firstly, we define controllers in the non-liveness region such that if the state ends up in this region, the controllers defined there continue to enforce the L2R motion. The non-liveness region is triangulated into simplices $\mathcal{S}_i$, $i \in \{17, 18, 19, 20\}$, see the orange region in Figure 5. Secondly, by design several simplices of our triangulation permit two exit facets. This feature relaxes the associated invariance conditions (9). For example, a trajectory in the L2R mode that has reached the non-liveness region is guided back to the nominal path more directly by including more exit facets.The restricted facets for each simplex for the L2R mode are shown in Figure 5 by red dashed lines.

B. Controller Design Procedure

With the triangulation and exit facets presented, we are ready to construct the controllers on each simplex. For each simplex $\mathcal{S}_i$, $i \in \mathcal{I}_s$, we identify the three corresponding vertices $v_j = \hat{v}_{i_j}$, $j \in \mathcal{I}_3$, $i_j \in \mathcal{I}_v$, that form the simplex along with the corresponding exit facets $\mathcal{I}_r$. We design the three control values such that the invariance conditions (9) are satisfied, and finally use (11) to obtain the feedback law (10).

So far we have designed the controller on each simplex completely independently. However, this may result in discontinuities when transitioning between simplices. We recall that a discontinuous $u(t)$ implies a discontinuous θ_d, see (4), which translates into an infinite angular velocity and is therefore undesirable for our onboard controller. To address this problem, we match the control values at the vertices along shared facets between contiguous simplices, which results in a continuous control assignment. If the control values cannot be matched, because otherwise one of the simplices would fail to solve RCP, then the next best alternative is to minimize the size of the discontinuity. Another choice left to the control designer is the magnitude of the control values at the vertices, where higher values result in more aggressive controls.

Designing reach controllers that are continuous between simplices is tedious to carry out manually, but can be automated. We briefly outline our automated approach, which was carried out in MATLAB. For each triangulation vertex $\hat{v}_i$, $i \in \mathcal{I}_v$, we define a quadratic program with linear inequality constraints to obtain a control value $\hat{u}_{i_j}$ at $\hat{v}_i$ for each simplex $\mathcal{S}_{i_j}$, $i_j \in \mathcal{I}(\hat{v}_i)$, where $\mathcal{I}(\hat{v}_i) \subset \mathcal{I}_s$ denotes the simplex indices that share the vertex $\hat{v}_i$. Our quadratic objective is to minimize the sum of all the pairwise distances between the different simplex control values at that vertex; ideally, the value of the minimized objective function should be zero to ensure continuity between contiguous simplices. However, this is not always possible. The linear constraints on the control values $\hat{u}_{i_j}$, $i_j \in \mathcal{I}(\hat{v}_i)$, arise from the invariance conditions (9) involving only the vertex $\hat{v}_i$ on the corresponding simplices $\mathcal{S}_{i_j}$. This guarantees that upon completing this optimization procedure separately for all the triangulation vertices, all of the invariance conditions (9) associated with each simplex are satisfied. Lastly, we included a lower and upper bound on the control values for the quadratic program. These bounds define the maximum allowable pitch angle, see (4).

C. Resulting Controller Design

The resulting control values $\hat{u}_{i_j}$, $i_j \in \mathcal{I}(\hat{v}_i)$, for our specific example are listed in Table I. The actual feedback over each simplex can be constructed using (11), and the corresponding vertex and control values. A plot of the resulting closed-loop dynamics for the L2R mode over $\mathcal{P}$ and the non-liveness region is shown in Figure 6. As expected, trajectories above

TABLE I
DETERMINED CONTROL VALUES

Vertex	Control value	Vertex	Control value
$\hat{v}_1$	3.1421	$\hat{v}_2$	-0.5383
$\hat{v}_3$	-1.5135	$\hat{v}_4$	-3.1421
$\hat{v}_5$	-3.1421	$\hat{v}_6$	-3.1421
$\hat{v}_7$	3.1421	$\hat{v}_8$	3.1421
$\hat{v}_9$	3.1421	$\hat{v}_{10}$	3.1421
$\hat{v}_{11}$	1.3844	$\hat{v}_{12}$	2.4170
$\hat{v}_{13}$	1.0474	-	-
$\hat{v}_{14}$	0.3980 for $\mathcal{S}_7, \mathcal{S}_{17}, \mathcal{S}_{18}, \mathcal{S}_{19}$	$\hat{v}_{14}$	-1.3844 for $\mathcal{S}_8, \mathcal{S}_9, \mathcal{S}_{10}$
$\hat{v}_{15}$	3.1421	$\hat{v}_{16}$	3.1421

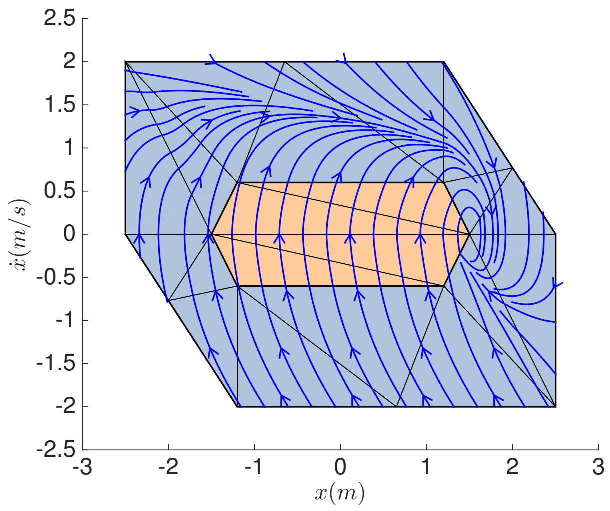


Fig. 6. The closed-loop vector field for the L2R mode, illustrated by the direction of the blue arrows.

the x-axis that start in $\mathcal{P}$ remain inside $\mathcal{P}$ and are guided to towards the set $\mathcal{B}_{right}$, while any other non-nominal trajectory (starting in $\mathcal{P}$ below the x-axis or in the non-liveness region) eventually recovers and crosses $\mathcal{B}_{right}$. We note that for most initial conditions in $\mathcal{S}_i$, $i \in \{14, 15, 16\}$, the trajectory would actually first cross $\mathcal{B}_{left}$ and then cross $\mathcal{B}_{right}$, violating the desired temporal sequence. However, to justify this behavior, these initial conditions are extremely non-nominal because they correspond to the quadrocopter moving left near or within the left threshold zone. As a result, avoiding crossing $\mathcal{B}_{left}$ is practically impossible under reasonable control effort.

We make some final remarks on the obtained control values. For all vertices but $\hat{v}_{14}$, we were able to use the same control value for adjacent simplex controllers, which pleasantly yields continuous closed-loop dynamics among contiguous simplices. Our optimization routine above provides a control value of 0 at $\hat{v}_{14}$, which yields an equilibrium at this vertex. Instead, we manually introduced a (fairly small) discontinuity at $\hat{v}_{14}$ to ensure that there is no equilibrium while still satisfying the associated invariance conditions (9) over the shared simplices. The price to pay is that the onboard controller described in Section II may give an unpredictable transient at the discontinuity, although this was not observed to be an issue in our experiment described in Section VI. It is

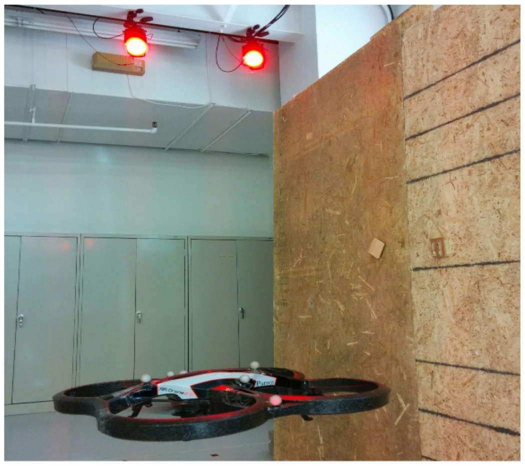


Fig. 7. Our quadrocopter vehicle close to the wall of the room with the motion capture camera system in the background.

easily seen that the discontinuity cannot be avoided except by making the control value at $\hat{v}_{14}$ zero, which is not permissible by the RCP framework. All other vertices on the x-axis have nonzero control values. Moreover, we can easily check that with our choice of control values, no equilibria appear in $\mathcal{P}$, which means that RCP is solved successfully on each simplex, guaranteeing our complex control specifications. We also note that the control values between $\hat{v}_1$ and $\hat{v}_6$ and between $\hat{v}_{11}$ and $\hat{v}_{14}$ (for $\mathcal{S}_8, \mathcal{S}_9, \mathcal{S}_{10}$) are negatives of each other. This yields a continuous transition between the controller on $\mathcal{S}_8$ and $\mathcal{S}_9$ (which uses the oddly reflected controller of $\mathcal{S}_1$) corresponding to the switch from the L2R to the R2L mode. Lastly, the largest control value converts to a desired pitch angle of 18 degrees, which we chose in order to demonstrate a reasonably fast side-to-side maneuver. This choice also resulted in larger controller gains in (10), which made our controller less sensitive to attitude disturbances and calibration offsets, as observed in our experiment.

VI. EXPERIMENTAL RESULTS

Our experimental platform is the Parrot AR.Drone 2.0 running firmware version 2.3.3. We interface with the AR.Drone through ROS, an open-source robot operating system [18]. More precisely, we used ROS Hydro, installed on a 64-bit 12.04 Ubuntu version. In addition, we used the ROS *ardrone autonomy* package [18], version 1.3.1. All experiments were conducted with the indoor hull shown in Figure 7, which protects the vehicle propellers.

We first demonstrate the successful execution of the desired side-to-side motion based on our RCP approach under nominal conditions, and compare it to the performance of a standard trajectory tracking controller [2], [3], which guides the vehicle along a predefined, timed side-to-side trajectory. Then to illustrate the robustness of our method, we introduce a disturbance not accounted for in the control design. We show that our control strategy still transitions between the two discrete modes correctly while the standard tracking control approach fails to do so. A video showing the experimental results can be found at: http://tiny.cc/quadrotorRCPx.

The tracking controller for starting at the left side in the L2R mode is based on tracking the following signal in the x-direction:

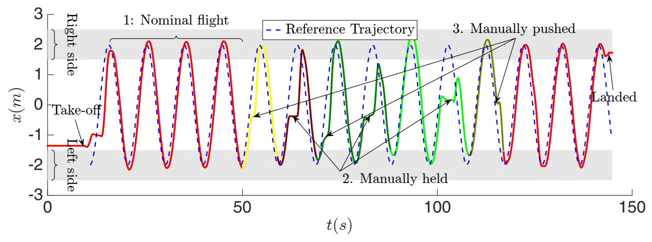
$$x_d(t) = 2\cos(0.65t + \pi). \tag{12}$$

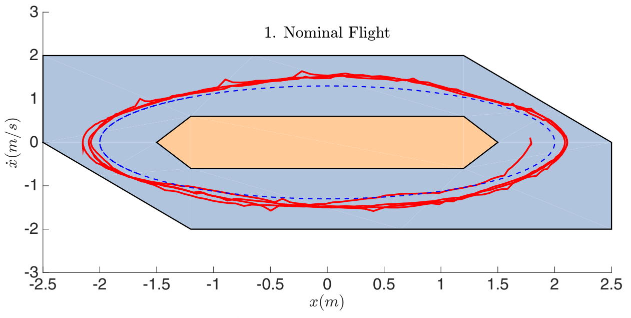
The resulting orbit $(x_d(t), \dot{x}_d(t))$, which is an ellipse, fits inside the polytope $\mathcal{P}$ for all time and executes the desired temporal sequence. The switches between modes occur automatically whenever the velocity $\dot{x}_d(t)$ crosses 0. The standard controller computes the desired pitch $\theta_d(t)$ based on the desired position $x_d(t)$ and the measured state of the vehicle and replaces the reach controller in Figure 2.

In the experiment, we lumped a sequence of actions that would test the quadrocopter's response to nominal and disturbance conditions, see Figures 8 and 9 for the tracking and RCP approaches, respectively. The following actions were performed for both control methods: (i) nominal flight consisting of a few cycles of the L2R and R2L modes, (ii) introducing a disturbance by manual holding the vehicle, and (iii) introducing a disturbance by pushing the vehicle. The manual holding involved grabbing the quadrocopter at various places during its flight, effectively reducing its velocity to zero, while the manual pushing involved mildly pushing the quadrocopter against its current direction of motion. During these disturbances, the state is likely to enter the non-liveness region, technically violating the control specifications (7); during these instances, we monitor the system's response of returning to nominal behavior and its adherence to safety and the desired temporal sequence.

We show the position $x(t)$ over time for the tracking and RCP approaches, see Figures 8 and 9, respectively. The colors differentiate the cycles where the different actions were executed, see the figure annotations. We highlight with gray the threshold zones, defined by d_{thres} and d_{max} (see Figure 3), on the left and right sides where the quadrocopter may turn around; the sets $\mathcal{B}_{left}$ and $\mathcal{B}_{right}$ are crossed in these highlighted zones when the velocity is zero. We also show the resulting orbit of the quadrocopter in the $(x-\dot{x})$-plane for selected cycles. On the left we show a few cycles of the nominal flight, and on the right we show cycles corresponding to select disturbances, where the colors of the orbits correspond to the colors in the above $x(t)$ signal. For the tracking approach, in Figure 8 we show the reference signal as a blue dashed line. For the RCP approach, in the polytopes of Figure 9 we show in gray the nominal vector field as in Figure 6. We show the R2L mode below the x-axis using odd symmetry of the L2R mode above the x-axis.

For the tracking method shown in Figure 8, the nominal flight portion (shown in red) tracks the reference signal well in time, and the corresponding nominal orbit shown on the upper polytope closely matches the reference ellipse which fits inside $\mathcal{P}$. Together, these plots show that the trajectory tracking approach meets all of the control specifications from Section III under nominal flight. In contrast, we can see that





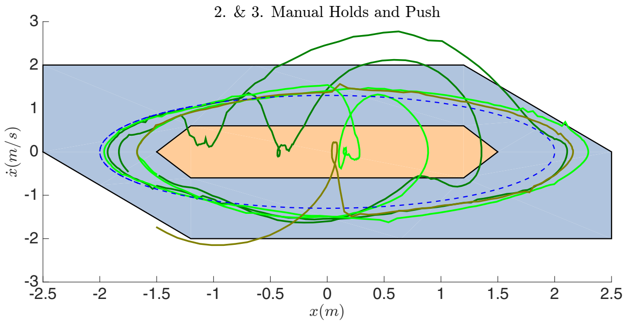


Fig. 8. Experimental results of the tracking approach.

under the manual hold, in two of the three attempts (shown in dark green and light green), the quadrocopter violated the desired temporal sequence. The hold, which lasted about 5 seconds, allowed the reference trajectory to pass towards the other side; upon release, the quadrotor simply followed the reference and crossed $\mathcal{B}_{left}$ twice in a row. The second manual push is shown in detail on the lower polytope (in olive green): we can see that as the quadrocopter attempted to catch up to the reference, the maximum speed limit was exceeded. Likewise, after the second manual hold (in dark green), upon release the quadrocopter sped up beyond the limit.

For the RCP method shown in Figure 9, the nominal flight portion (shown in red) also satisfies the control specifications. Interestingly, the orbit on the upper polytope shows a non-trivial limit cycle, matching fairly closely to the background vector field. In contrast to the tracking approach, it is clear from the $x(t)$ plot that no matter which disturbance was presented, the desired temporal sequence was always maintained. Furthermore, the safety constraints were always respected, as can be seen in the various cycles on the lower polytope.

The main weakness of the tracking approach is that the reference signal imposes an unnatural timing to the behavior of the system. Under any disturbance of the types considered here, upon removal of that disturbance the system attempts to catch up to the reference. During this transient response, the

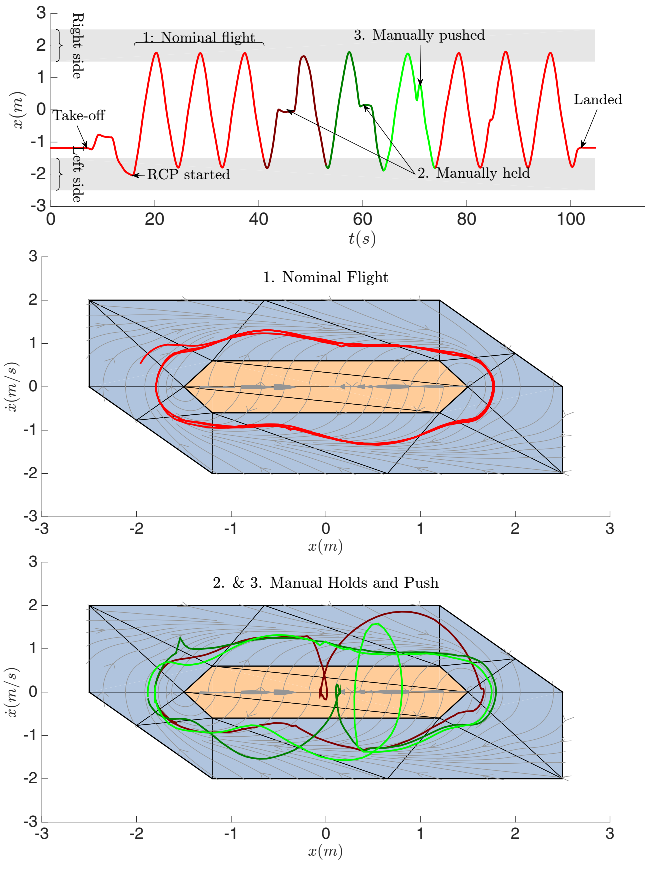


Fig. 9. Experimental results of the RCP approach.

safety constraints can be easily violated because the tracking controller has no encoded information regarding safety. Similarly, the desired temporal sequence can be violated under a disturbance because the tracking signal, which is defined *a priori*, cannot change its signal or remember which side the quadrocopter last crossed. Rather than accommodating the off-course quadrocopter, the tracking controller only forces the quadrocopter to catch up as fast as the controller gains permit without regard to anything else. In contrast, the RCP approach puts at foremost the control specifications. Its feedback nature means that at any point in $\mathcal{P}$ union the non-liveness region, the quadrocopter can respond intelligently to return back to nominal behavior while respecting safety and the desired temporal sequence after a disturbance is removed. The compromise is that the reach control structure and design are considerably more complex compared to a simple timed trajectory, but that is the price to pay to have such a richly detailed closed-loop vector field as shown in Figure 6.

VII. Conclusion

We succeeded in experimentally demonstrating the first ever implementation of reach controllers on a real system. The result is a logically complex quadrocopter maneuver. The main advantages of the reach control approach are that it permits the incorporation of safety constraints, different modes of operation, aggressive speed of execution, and robustness to unmodeled disturbances, as shown in our comparison between the reach control and standard tracking approaches. The side-to-side maneuver shown here was mainly chosen to demonstrate the proof-of-concept. A first extension would be to incorporate reach control also in the y-direction or, by yawing the vehicle, use a hybrid-based framework to switch between oriented side-to-side motions to achieve more complex motions. Another extension is to explore the possibility of eliminating the remaining standard tracking controllers with appropriate reach controllers to be able to logically control all the degrees of freedom of the quadrocopter.

References

[1] A. P. Schoellig, F. Mueller, and R. D'Andrea, "Optimization-based iterative learning for precise quadrocopter trajectory tracking," *Autonomous Robots*, vol. 33, nos. 1-2, pp. 103-127, 2012.
[2] A. P. Schoellig, M. Hehn, S. Lupashin, and R. D'Andrea, "Feasibility of motion primitives for choreographed quadrocopter flight", in *Proceedings of the American Control Conference (ACC)*, 2011, pp. 3843-3849.
[3] S. Lupashin, M. Hehn, M. Mueller, A. P. Schoellig, M. Sherback, and R. D'Andrea, "A platform for aerial robotics research and demonstration: the Flying Machine Area," *Mechatronics*, vol. 24, no. 1, pp. 41-54, 2014.
[4] A. LaViers and M. Egerstedt, "Controls and art: inquires at the intersection of the subjective and the objective," Springer, 2014.
[5] R. Tedrake, I. Manchester, M. Tobenkin, and J. Roberts, "LQR-trees: Feedback motion planning via sums-of-squares verification," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1038-1052, 2010.
[6] 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.
[7] V. Cichella, I. Kaminer, E. Xargay, V. Dobrokhodov, N. Hovakimyan, A. P. Aguiar, and A. M. Pascoal, "A Lyapunov-based approach for time-coordinated 3D path-following of multiple quadrotors," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2012, pp. 1776-1781.
[8] A. Roza and M. Maggiore, "Path following controller for a quadrotor helicopter," in *Proceedings of the American Control Conference (ACC)*, 2012, pp. 4655 - 4660.
[9] J. Gillula, G. Hoffmann, H. Huang, M. Vitus, and C. Tomlin, "Applications of hybrid reachability analysis to robotic aerial vehicles," *The International Journal of Robotics Research*, vol. 30, no. 3, pp. 335-354, 2011.
[10] M.E. Broucke, "Reach control on simplices by continuous state feedback," *SIAM Journal on Control and Optimization*, vol. 48, no. 5, pp. 3482-3500, 2010.
[11] M.E. Broucke and M. Ganness, "Reach control on simplices by piecewise affine feedback," *SIAM Journal on Control and Optimization*, vol. 52, no. 5, pp. 3261 - 3286, 2014.
[12] L.C.G.J.M. Habets and J.H. van Schuppen, "A control problem for affine dynamical systems on a full-dimensional polytope," *Automatica*, vol. 40, no. 1, pp. 21-35, 2004.
[13] L.C.G.J.M. Habets, P.J. Collins, and J.H. van Schuppen, "Reachability and control synthesis for piecewise-affine hybrid systems on simplices," *IEEE Transactions on Automatic Control*, vol. 51, no. 6, pp. 938-948, 2006.
[14] M. Kloetzer and C. Belta, "A fully automated framework for control of linear systems from temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 287-297, 2008.
[15] C. W. Lee, "Subdivisions and triangulations of polytopes," *Handbook of Discrete and Computational Geometry*, CRC Press Series Discrete Math. Appl., pp. 271-290, 1997.
[16] G. Ashford, "A Time-Varying Feedback Approach to Reach Control on a Simplex", Master's Thesis at the University of Toronto, 2011.
[17] B. Roszak and M. E. Broucke, "Necessary and sufficient conditions for reachability on a simplex," *Automatica*, vol. 42, no. 11, pp. 1913-1918, 2006.
[18] Ardrone autonomy package, ver. 1.3.1, available at www.ros.org.