Auton Robot (2015) 39:101–121
DOI 10.1007/s10514-015-9429-0

Collision avoidance for aerial vehicles in multi-agent scenarios

Javier Alonso-Mora · Tobias Naegeli · Roland Siegwart · Paul Beardsley

Received: 24 October 2013 / Accepted: 3 January 2015 / Published online: 25 January 2015

Abstract This article describes an investigation of local motion planning, or collision avoidance, for a set of decision-making agents navigating in 3D space. The method is applicable to agents which are heterogeneous in size, dynamics and aggressiveness. It builds on the concept of velocity obstacles (VO), which characterizes the set of trajectories that lead to a collision between interacting agents. Motion continuity constraints are satisfied by using a trajectory tracking controller and constraining the set of available local trajectories in an optimization. Collision-free motion is obtained by selecting a feasible trajectory from the VO's complement, where reciprocity can also be encoded. Three algorithms for local motion planning are presented—(1) a centralized convex optimization in which a joint quadratic cost function is minimized subject to linear and quadratic constraints, (2) a distributed convex optimization derived from (1), and (3) a centralized non-convex optimization with binary variables in which the global optimum can be found, albeit at higher computational cost. A complete system integration is described and results are presented in experiments with up to four physical quadrotors flying in close proximity, and in experiments with two quadrotors avoiding a human.

Electronic supplementary material The online version of this article (doi:10.1007/s10514-015-9429-0) contains supplementary material, which is available to authorized users.

J. Alonso-Mora (✉)
ETH Zurich and Disney Research Zurich, Leonhardstrasse 21, 8092 Zurich, Switzerland
e-mail: jalonsom@csail.mit.edu

T. Naegeli · R. Siegwart
ETH Zurich, Leonhardstrasse 21, 8092 Zurich, Switzerland

P. Beardsley
Disney Research Zurich, Stampfenbachst. 48, 8006 Zurich, Switzerland

Keywords Collision avoidance · Reciprocal · Aerial vehicle · Quadrotor · Multi-robot · Multi-agent · Motion planning · Dynamic environment

1 Introduction

The last decade has seen significant interest in unmanned aerial vehicles (UAVs), including multi-rotor helicopters due to their mechanical simplicity and agility. Successful navigation builds on the interconnected competences of localization, mapping, and motion planning/control. Kumar and Michael (2012) and Mahony et al. (2012) provide comprehensive overviews of the field.

The computation of global collision-free trajectories in a multi-agent setting is currently challenging in real-time. One approach is to hierarchically decompose the problem into a global planner, which may omit motion constraints, and a local reactive component, which locally modifies the trajectory to incorporate all relevant constraints. The latter is the topic of this article—a real-time reactive local motion planner which takes into account motion constraints, static obstacles, and other agents. The important case where other agents are themselves decision-making is addressed.

Furthermore, a full system, including low level control, collision avoidance and basic high level planning is proposed end experimentally verified. This paper describes in detail algorithms for collision avoidance as well as their interaction with the multiple functioning parts.

1.1 Related work

In recent years, trajectory generation has been successfully demonstrated for a set of agents navigating in a controlled environment with an external tracking system that provides precise localization. See the testbeds in the GRASP lab (Michael et al. 2010) and the Flying Machine Arena Lupashin et al. (2011). In contrast, our goal is to handle dynamic obstacles, where real-time reactive collision avoidance is needed. We develop a method which is intended for on-board use, and which anticipates that the algorithmic development will be accompanied by the development of appropriate sensing capabilities, although the experiments in this article were done using a similar testbed to those above.

Global collision-free trajectories for a single agent can be obtained using randomized sampling (Frazzoli et al. 2002). This method takes into account the dynamics of the ego-agent and both static and dynamic agents, but it does not handle multi-agent scenarios with multiple decision-making agents. Global collision-free trajectories for a team of agents can be computed via a centralized mixed integer quadratic optimization (Mellinger et al. 2012), in which the state (the position and its derivatives) of the agents at time-spaced intervals is optimized. The method shows impressive results for trajectory generation (Kushleyev et al. 2012) but it lacks real-time performance. A related approach is to compute the state at time-spaced intervals via a sequential convex optimization (Augugliaro et al. 2012), leading to faster computations for trajectory generation, but still not in real-time. In contrast to that work, our goal is to compute a local trajectory for a short time horizon in a reactive way and at a high frequency.

Fast reactive local motion planning (or collision avoidance) is typically required to respond to unexpected events or errors in the environment model. The use of local motion planning in conjunction with a high-level controller for a multi-agent task, such as exploration and coverage, was described by Schwager et al. (2011), among others.

See-and-avoid approaches, such as the one described by Mcfadyen et al. (2012), can be used when no range information about the agent to be avoided is available. This is the case for instance for mid-air avoidance with vision-only sensing. In contrast, our work is concerned with close navigation, where relative position, velocity and size of the agents are known. We build on the concept of velocity obstacles (VO) by Fiorini and Shillert (1998), a method for characterizing the agent velocities that lead to a collision within a planning horizon.

Reactive collision avoidance methods in close proximity can typically be divided between rule-based and optimization-based. Rule-based methods, which include potential fields (Ogren et al. 2004) and optimal control laws (Hoffmann and Tomlin 2008), may work well in scenarios with low agent density and low speeds, but typically do not provide hard guarantees or must include restrictive rules. Optimization-based methods include a centralized nonlinear program (Raghunathan et al. 2004) which scales poorly with the number of agents and a mixed integer linear program (MILP) with constraints in dynamics (Kuwata and How 2007). Alternatively, decentralized nonlinear model predictive control using potential functions for collision avoidance (Shim et al. 2003) has been explored, but without guarantees. The method proposed in this work falls in the later category of optimization-based methods, thus providing hard guarantees. This is combined with concepts from potential fields, for increased safety.

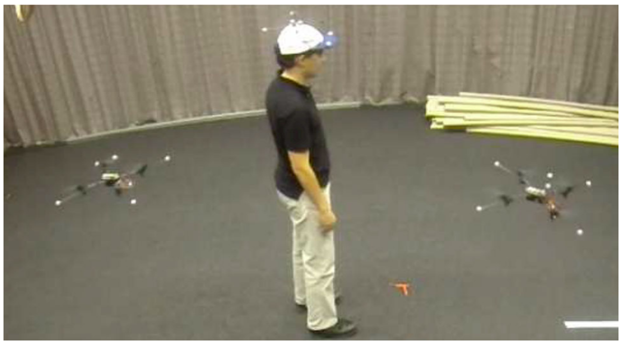


Fig. 1 Collision avoidance experiment with quadrotors plus a human

Slightly departing from pure reactive collision avoidance methods, a collision-free local motion can be obtained by using a set of motion primitives or forward simulating input commands and collision checking them. These approaches have been widely applied for single agents moving in 2D spaces by Fox et al. (1997), Knepper and Mason (2012) and Pivtoraiko and Kelly (2005) among others. Such an approach readily extends to 3D motion but with increase computational cost due to the larger set of motion primitives. In this work, we keep the idea of employing a set of local motions but do not collision check them individually, instead, we solve a convex optimization, which is more efficient (Fig. 1).

Reactive local planners typically ignore that other agents are decision-making entities and this may lead to suboptimal performance, especially in crowded multi-agent scenarios. The reciprocal velocity obstacles (RVO) method by van den Berg et al. (2009) addresses this and assumes pairwise collaboration of equal contribution, but it restricts robot action capabilities to a set of constant velocities. Under the assumption that all agents follow a straight line trajectory, future collisions can be characterized as a function of relative velocity alone. Extensions to linear dynamics have been developed by Bareiss and van den Berg (2013), which extends the method to LQR-obstacles, and by Rufli et al. (2013), which extends to systems that require any degree of continuity. But, both extensions apply only to homogeneous sets of agents, where all interacting agents present the same control parameters and their full state, including its higher order derivatives, is known. On the other hand, we build on the idea presented by Alonso-Mora et al. (2012b), which preserves motion continuity while applying to heterogeneous groups of agents with potentially different control parameters. We also build on the joint utility method presented by Alonso-Mora et al. (2013)

for centralized computation. Both methods build on previous work by van den Berg et al. (2009).

1.2 Contribution

Algorithmic

Our main contribution is a derivation of three methods for local motion planning, or collision avoidance, in 3D environments, based on recent extensions of the velocity obstacle concept.

- A *distributed convex* optimization in which a quadratic cost function is minimized subject to linear and quadratic constraints. Only knowledge of relative position, velocity and size of neighboring agents is required.
- A *centralized convex* optimization in which a joint quadratic cost function is minimized subject to quadratic and linear constraints. This variant scales well with the number of agents and presents real-time capability, but may provide a sub-optimal solution.
- A *centralized non-convex* optimization in which a joint quadratic cost function is minimized subject to linear, quadratic and non-convex constraints, formulated as a mixed integer quadratic program. This variant explores the entire solution space but scales poorly with the number of agents.

All methods can

- characterize agents as decision making agents that collaborate to achieve collision avoidance.
- incorporate motion continuity constraints.
- handle heterogeneous groups of agents.

A complexity analysis is provided, and a discussion of the advantages and disadvantages of each method.

System integration

The novel collision avoidance algorithms are integrated with the full control loop and implemented in an experimental platform formed by several quadrotor helicopters.

Experimental evaluation

Finally the article contains extensive results in experiments with up to four physical quadrotors, and in experiments with a human as a dynamic obstacle. We show the potential of the real-time approach and discuss future research.

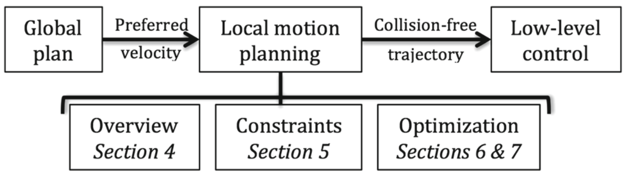


Fig. 2 High-level schema of the system

1.3 Organization

The problem statement is provided in Sect. 2, followed by an overview of the local motion planning framework in Sect. 3. The optimization framework is described in Sects. 4 and 5 formulates it as a convex optimization. Section 6 describes an extension where the problem is formulated as a non-convex optimization. The control framework for a quadrotor helicopter is then described in Sect. 7. Experimental results are described in Sects. 8, and 9 concludes this paper. A high-level block diagram of the proposed method is shown in Fig. 2.

1.4 Notation

Throughout this paper x denotes scalars, $\mathbf{x}$ vectors, X matrices and $\mathcal{X}$ sets. The Minkowski sum of two sets is denoted by $\mathcal{X} \oplus \mathcal{Y}$, $x = ||\mathbf{x}||$ denotes the euclidean norm of vector $\mathbf{x}$, $\mathbf{x}^H$ the projection of $\mathbf{x}$ onto the horizontal plane and x^3 its vertical component, thus $\mathbf{x} = [\mathbf{x}^H, x^3]$. The super index $\cdot^k$ indicates the value at time t^k, and $\tilde{t} = t - t^k$ the appropriate relative time. Subindex $\cdot_i$ indicates agent i and relative vectors are denoted by $\mathbf{x}_{ij} = \mathbf{x}_i - \mathbf{x}_j$. For ease of exposition, no distinction is made between estimated and real states.[1]

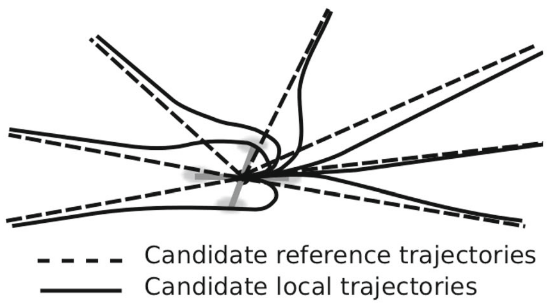
2 Problem statement

A group of n agents freely moving in 3D space is considered. For each agent i, $\mathbf{p}_i \in \mathbb{R}^3$ denotes its position, $\mathbf{v}_i = \dot{\mathbf{p}}_i$ its velocity and $\mathbf{a}_i = \ddot{\mathbf{p}}_i$ its acceleration.

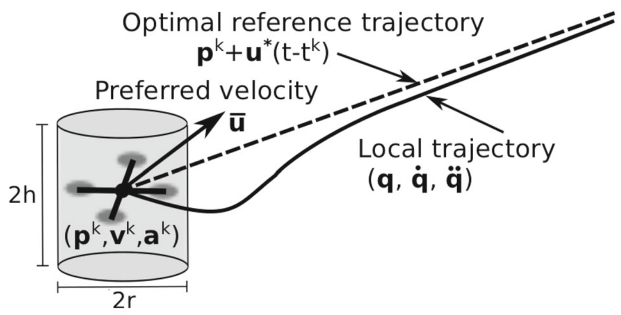
The goal is to compute for each agent, at high frequency (typically 10 Hz), a local motion that respects the kinematics and dynamic constraints of the ego agent and that is collision-free with respect to neighboring agents for a short time horizon (typically 2–10 s). This problem is also referred as collision avoidance.

Two scenarios are discussed, a *centralized* one where a central unit computes local motions for all agents simultaneously and a *distributed* one where each agent computes independently its local motion. For the second case, no communication between the agents is required but each agent must maintain an estimate of the position and velocity of its neighbors. For the distributed case, neighboring agents

[1] Estimated states with a Kalman filter, including time delay compensation, are used in our implementation.


Candidate reference trajectories
Candidate local trajectories

(a) Example of candidate local trajectories


Optimal reference trajectory
$\mathbf{p}^k+\mathbf{u}^*(t-t^k)$
Preferred velocity
$\bar{\mathbf{u}}$
Local trajectory
$(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$
2h
$(\mathbf{p}^k,\mathbf{v}^k,\mathbf{a}^k)$
2r

(b) Example of a local trajectory

Fig. 3 Schemas of local motion planning. Motion primitives given by candidate reference trajectories and local trajectory given by the optimal reference trajectory. **a** Example of candidate local trajectories. **b** Example of a local trajectory

are labelled as dynamic obstacles or decision-making agents which employ an identical algorithm to the ego-agent.

The local motion planning problem is first formulated as an efficient convex optimization (either centralized or distributed), followed by a non-convex optimization (centralized) that trades-off efficiency for richness of the solution space.

The shape of each agent is modeled by an enveloping cylinder of radius r_i and height $2h_i$ centered at the agent.[2] For the case of quadrotor helicopters, a cylindrical shape has been employed (to account for rotor downwash) for non-real time collision-free trajectory generation by Mellinger et al. (2012) and Augugliaro et al. (2012). This is in contrast to the sphere/ellipse simplification of previous work in real-time collision avoidance by Alonso-Mora et al. (2012b) and Bareiss and van den Berg (2013).

The local motion planning framework is applied to quadrotor helicopters, due to their agility and mechanical simplicity (Mahony et al. 2012). Implementation details, including an appropriate control framework are described in Sect. 7, followed by extensive experimental results with up to four quadrotors and a human.

3 Local motion planning: overview

In this work a fully integrated system is proposed, formed by several interconnected components. In this section first the terminology used is introduced, followed by the definition of the local trajectories and an overview of the optimization problem solved to obtain collision-free motions.

3.1 High level system description

Our system is formed by several interconnected modules, as shown in Fig. 2. In the absence of standard terminology, we use the following, illustrated in Fig. 3.

– *Global trajectory:* A trajectory between two configurations embedded in a cost field with simplified dynamics and constraints.
– *Preferred velocity:* The ideal velocity computed by a guidance system to track the global trajectory. It is denoted by $\bar{\mathbf{u}} \in \mathbb{R}^3$.
– *Local trajectory:* The trajectory executed by the agent, for a short time horizon, and selected from a set of *candidate local trajectories*, which respect the kinematics and dynamic constraints of the agent. Each one is computed from a *candidate reference trajectory* via a bijective mapping.
– *Candidate reference trajectory:* is given by a constant velocity (*candidate reference velocity*[3]) $\mathbf{u} \in \mathbb{R}^3$ which indicates the target direction and velocity of the agent. An *optimal reference trajectory*, parametrized by the *optimal reference velocity* $\mathbf{u}^* \in \mathbb{R}^3$, is selected from the set of *candidate reference velocities* and minimizes the deviation to $\bar{\mathbf{u}} \in \mathbb{R}^3$. It defines the *local trajectory*.

The focus of this work is on local motion planning, using a preferred velocity $\bar{\mathbf{u}}$ computed by a guidance system in order to track a global trajectory. Following the idea of Alonso-Mora et al. (2012a), an optimal *reference* trajectory is computed such that its associated local trajectory is collision-free. This provides a parametrization of local motions given by $\mathbf{u}$ and allows for an efficient optimization in candidate reference velocity space ($\mathbb{R}^3$) to achieve collision-free motions (Sect. 4).

3.2 Definition of candidate local trajectories

Each candidate reference trajectory is that of an omnidirectional agent moving at a constant velocity $\mathbf{u}$ and starting at the current position $\mathbf{p}^k$ of the agent

[2] In this formulation arbitrary object shapes can be considered, but with the assumption that they do not rotate during the local planning horizon (typically a few seconds).

[3] In previous works we have referred to it as holonomic trajectory and reference trajectory but, to hopefully increase clarity we now adopt the latter.

$$\mathbf{p}_{\mathrm{ref}}(t) = \mathbf{p}^k + \mathbf{u}(t - t^k), \quad t \in [t^k, \infty). \tag{1}$$

A trajectory tracking controller, respecting the kinematics and dynamic constraints of the agent, $\mathbf{q}(\tilde{t}) = f(\mathbf{z}^k, \mathbf{u}, \tilde{t})$ is considered, continuous in the initial state $\mathbf{z}^k = [\mathbf{p}^k, \dot{\mathbf{p}}^k, \ddot{\mathbf{p}}^k, \dots]$ of the agent and converging to the candidate reference trajectory. This defines the candidate local trajectories given by $\mathbf{q}(\tilde{t})$ and a bijective mapping (for given $\mathbf{z}^k$) from candidate reference velocities $\mathbf{u}$.

3.3 Overview of the optimization problem

A collision-free local trajectory (described in Sect. 3.2 and parametrized by $\mathbf{u}^*$) is obtained via an optimization in candidate reference velocity where the deviation to the preferred velocity $\bar{\mathbf{u}}$ is minimized. In Sect. 5 two alternative formulations are presented, a *centralized* and a *distributed* optimization, formulated as a convex optimization with quadratic cost and linearized constraints for

- *Motion continuity:* The position error between the candidate reference trajectory and the local trajectory must be below a limit ε to guarantee that the local trajectory is collision-free if the candidate reference trajectory is. Continuity in position, velocity, acceleration and further derivatives (optional) is guaranteed by construction. See Sect. 4.4.
- *Collision avoidance:* The candidate reference velocities leading to a collision are described by the velocity obstacle (Fiorini and Shillert 1998), a cone in relative candidate reference velocity space, for agents of ε-enlarged radius. See Sect. 4.5. Consider $\mathcal{C}_i$ the set of collision avoidance constraints for agent i with respect to its neighbors and $\mathcal{C} = \bigcup_{i \in \mathcal{A}} \mathcal{C}_i$.

These non-convex constraints can be linearized leading to a convex optimization with quadratic cost and linear and quadratic constraints. In Sect. 4 the optimization problem is formulated and Sect. 5 describes the convex optimization for the case of heterogenous groups of agents of unknown dynamics. An algorithm to solve the original non-convex optimization is described in Sect. 6, and an extension for homogeneous groups of agents (same dynamics and controller) is outlined in Appendix 1.

In the *centralized* case, a single convex optimization is solved where the optimal reference velocity of all agents $\mathbf{u}^*_{1:m} = [\mathbf{u}^*_1, \dots, \mathbf{u}^*_m]$ are jointly computed.

In the *distributed* case, each agent $i \in \mathcal{A}$ independently solves an optimization where its optimal reference velocity $\mathbf{u}^*_i$ is computed. In this case it can be considered that agents collaborate in the avoidance and apply the same algorithm, otherwise they are treated as dynamic obstacles. Only information about the neighbors' pairwise current position, velocity and shape is required.

4 Local motion planning: formulation

In this section the optimization cost and the optimization constraints are described.

4.1 Preferred velocity

In each local planning iteration, given the current position and velocity of the agent, a preferred velocity $\bar{\mathbf{u}}$ is computed to follow a global plan to achieve either:

- *Goal:* $\bar{\mathbf{u}}$ is given by a proportional controller towards the goal position $\mathbf{g}_i$, saturated at the agent's preferred speed $\bar{V} > 0$, and decreasing when in the neighborhood.

$$\bar{\mathbf{u}}_i = \bar{V} \min\left(1, \frac{||\mathbf{g}_i - \mathbf{p}_i||}{K_{\bar{v}}}\right) \frac{\mathbf{g}_i - \mathbf{p}_i}{||\mathbf{g}_i - \mathbf{p}_i||}, \tag{2}$$

  where $K_{\bar{v}} > 0$ is the distance to the goal from which the preferred velocity is reduced linearly. Alternatively, it can be given by an LQR controller as shown by Bareiss and van den Berg (2013).
- *Trajectory:* $\bar{\mathbf{u}}$ is given by a trajectory tracking controller for omnidirectional agents, such as the one by Hoffmann et al. (2008).

4.2 Repulsive velocity

The optimization algorithm described in this paper guarantees collision-free motion per se, as proved in Sect. 5.1. Nonetheless, optimality in our formulation implies that agents would be infinitely close to each other in the avoidance phase. In practical settings, with uncertainties and modeling errors, this can potentially lead to collisions.

To mitigate this problem, a small repulsive velocity, much lower than in traditional potential field approaches, is added to the preferred velocity $\bar{\mathbf{u}}$. This has the effect of pushing agents slightly away from each other when close together. At high speeds the small repulsive velocity alone is not enough to create collision-free motions. Collision-free guarantees arise from the optimization constraints.

A vector field like those of Fig. 4 gives the repulsive velocity $\mathring{\mathbf{u}}$ with finite support in a neighborhood of the agent. The equations for the repulsive velocity are described in Appendix 2.

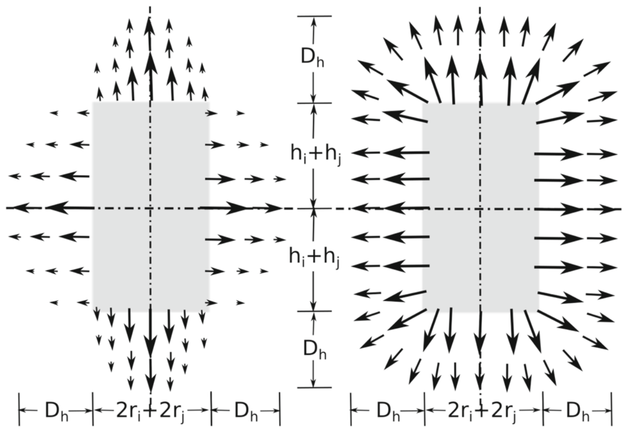


Fig. 4 Two examples of repulsive velocities

4.3 Optimization cost

The optimization cost is given by two parts, the first one a regularizing term penalizing changes in reference velocity and the second one minimizing the deviation to the preferred velocity $\bar{\mathbf{u}}_i$ corrected by the repulsive velocity $\mathring{\mathbf{u}}_i$.

Guy et al. (2010) showed that pedestrians prefer to maintain a constant velocity in order to minimize energy. Preference was given to turning instead of changing the speed. Furthermore, penalizing changes in speed leads to a reduction of deadlock situations. This idea is formalized as an elliptical cost, higher in the direction parallel to $\bar{\mathbf{u}}_i + \mathring{\mathbf{u}}_i$, and lower perpendicular to it. Let $\lambda > 0$ represent the relative weight and define $L = diag(\lambda, 1, 1)$ and D_i the rotation matrix of the transformation onto the desired reference frame.

In the *distributed* case, the quadratic cost is

$$C(\mathbf{u}_i) := K_o||\mathbf{u}_i - \mathbf{v}_i||^2 + ||L^{1/2}D_i(\mathbf{u}_i - (\bar{\mathbf{u}}_i + \mathring{\mathbf{u}}_i))||^2, \quad (3)$$

where, with the exception of the optimization variables $\mathbf{u}_i$, the values of all other variables are given for the current time instance and K_o is a design constant.

In the *centralized* case, the quadratic cost is

$$C(\mathbf{u}_{1:m}) := K_o \sum_{i=1}^{m} \bar{\omega}_i||\mathbf{u}_i - \mathbf{v}_i||^2 + \sum_{i=1}^{m} \bar{\omega}_i||L^{1/2}D_i(\mathbf{u}_i - (\bar{\mathbf{u}}_i + \mathring{\mathbf{u}}_i))||^2, \quad (4)$$

where $\bar{\omega}_i$ represent relative weights between different agents, to take into account avoidance preference between agents, which can be interpreted as aggressive (high $\bar{\omega}_i$) versus shy (low $\bar{\omega}_i$) behavior.

4.4 Motion continuity constraints

In order to guarantee collision-free motions it must be guaranteed that each agent remains within ε_i of its reference trajectory, with $\varepsilon_i < \min_j ||\mathbf{p}_j - \mathbf{p}_i||/2$.

Constraint 1 *(Motion continuity) For a maximum tracking error ε_i and current state $z_i = [\boldsymbol{p}_i, \dot{\boldsymbol{p}}_i, \ddot{\boldsymbol{p}}_i]$, the set of candidate reference velocities $\boldsymbol{u}_i$ that can be achieved with position error lower than ε_i is given by*

$$R_i = R(z_i, \varepsilon_i) = \{\boldsymbol{u}_i \mid ||(\boldsymbol{p}_i + \tilde{t}\boldsymbol{u}_i) - f(z_i, \boldsymbol{u}_i, \tilde{t})|| \le \varepsilon_i, \ \forall \tilde{t} > 0\}, \quad (5)$$

For arbitrary trajectory tracking controller $f(\mathbf{z}_i, \mathbf{u}_i, \tilde{t})$ the set R_i can be precomputed. For quadrotors freely moving in 3D space this concept was first introduced by Alonso-Mora et al. (2012b), where an LQR controller was used as the function $f(\mathbf{z}_i, \mathbf{u}_i, \tilde{t})$. In our implementation, position control and local trajectory are decoupled and $f(\mathbf{z}_i, \mathbf{u}_i, \tilde{t})$ is given by Eq. (22). See Fig. 5 for an example.

If the agents were omnidirectional, the set R_i would be given by a sphere, centered at zero and of radii $\mathbf{v}_{max}$ (no continuity in velocity). For the case of study of this paper, given ε_i and $\mathbf{z}_i$ (initial state), the limits are obtained from the precomputed data

$$\delta R_{i,\varepsilon_i} = \{\mathbf{u}_i \mid \max_{\tilde{t}>0} ||(\mathbf{p}_i + \tilde{t}\mathbf{u}_i) - f(\mathbf{z}_i, \mathbf{u}_i, \tilde{t})|| = \varepsilon_i\} \quad (6)$$

To reduce computational complexity, $\delta R_{i,\varepsilon_i}$ is then approximated by the largest ellipse (quadratic constraint of the form $\mathbf{u}_i^T Q_i \mathbf{u}_i + \mathbf{l}_i \cdot \mathbf{u}_i \le a_i$, with $Q_i \in \mathbb{R}^{3\times 3}$ positive semi-definite, $\mathbf{l}_i \in \mathbb{R}^3$ and $a_i \in \mathbb{R}$) fully contained inside $\delta R_{i,\varepsilon_i}$.

4.5 Collision avoidance constraints

Each agent must avoid colliding with static obstacles, other controlled agents and dynamic obstacles. A maximum time horizon τ is considered, for which collision-free motion is guaranteed.

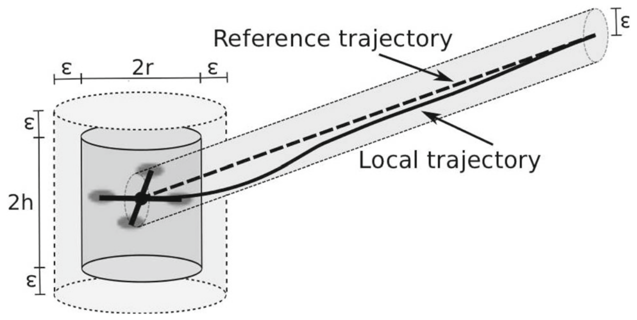


Fig. 5 Schema of reference and local trajectory with tracking error below ε. The radius and height of the agent are also enlarged by ε

A static obstacle is given by a region $\mathcal{O} \subset \mathbb{R}^3$. The agents are modeled by an enclosing vertical cylinder.[4] The disk of radius r centered at $\mathbf{c}$ is denoted by $D_{\mathbf{c},r} \subset \mathbb{R}^2$ and the cylinder of radius r and height $2h$ centered at $\mathbf{c}$ is denoted by $C_{\mathbf{c},r,h} \subset \mathbb{R}^3$. This section builds on the work on VO by van den Berg et al. (2009) and Fiorini and Shillert (1998).

The collision avoidance constraints are computed with respect to an agent following the reference trajectory. Since the agents are not able to perfectly track the reference trajectory, their enveloping cylinder must by enlarged by the maximum tracking error ε_i. For ease of notation, through out this section the following is used: $\bar{r}_i := r_i + \varepsilon_i$ and $\bar{h}_i := h_i + \varepsilon_i$ the enlarged radius and cylinder height, and $\mathbf{p}_i$ the current position of agent i.

Constraint 2 *(Avoidance of static obstacles) For every agent $i \in \mathcal{A}$ and neighboring obstacle $\mathcal{O}$, the constraint is given by the reference velocities for which the intersection between $\mathcal{O}$ and the agent is empty for all future time instances, i.e. $\mathbf{p}_i + \mathbf{u}_i\tilde{t} \notin (\mathcal{O} \oplus C_{\mathbf{0},\bar{r}_i,\bar{h}_i})$, $\forall \tilde{t} \in [0, \tau]$.*

This constraint is given by a cone in the 3D candidate reference velocity space generated by $\mathcal{O} \oplus C_{\mathbf{0},\bar{r}_i,\bar{h}_i} - \mathbf{p}_i$ and truncated at $(\mathcal{O} \oplus C_{\mathbf{0},\bar{r}_i,\bar{h}_i} - \mathbf{p}_i)/\tau$. An example of this constraint for a rectangular object is shown in Fig. 6.

Boundary wall constraints are directly added, given by $\mathbf{n} \cdot \mathbf{u}_i \leq d(\text{wall}, \mathbf{p}_i)/\tau$, with $\mathbf{n}$ the normal vector to the wall and $d(\text{wall}, \mathbf{p}_i)$ the distance to it.

Constraint 3 *(Inter-agent collision avoidance) For every pair of neighboring agents $i, j \in \mathcal{A}$, the constraint is given by the relative reference velocities for which the agents' enveloping shape does not intersect within the time horizon, i.e. $\mathbf{u}_i - \mathbf{u}_j \notin \mathcal{VO}^\tau_{ij} = \bigcup_{\tilde{t}=0}^{\tau}((C_{\mathbf{p}_j,\bar{r}_j,\bar{h}_j} \oplus C_{-\mathbf{p}_i,\bar{r}_i,\bar{h}_i})/\tilde{t})$, $\forall \tilde{t} \in [0, \tau]$.*

For cylindrical-shape agents the constraint can be separated in the horizontal plane $\|\mathbf{p}^H_i - \mathbf{p}^H_j + (\mathbf{u}^H_i - \mathbf{u}^H_j)\tilde{t}\| \geq \bar{r}_i + \bar{r}_j$ and the vertical component $|p^3_i - p^3_j + (u^3_i - u^3_j)\tilde{t}| \geq \bar{h}_i + \bar{h}_j$ for all $\tilde{t} \in [0, \tau]$, where at least only one constraint needs to be satisfied. , where at least only one constraint needs to be satisfied. The constraint is a truncated cone, as shown in Fig. 7, left.

4.5.1 Approximation by linear constraints

Since the volume of reference velocities leading to a collision is convex, the aforementioned collision avoidance constraints are non-convex. To obtain a convex optimization,

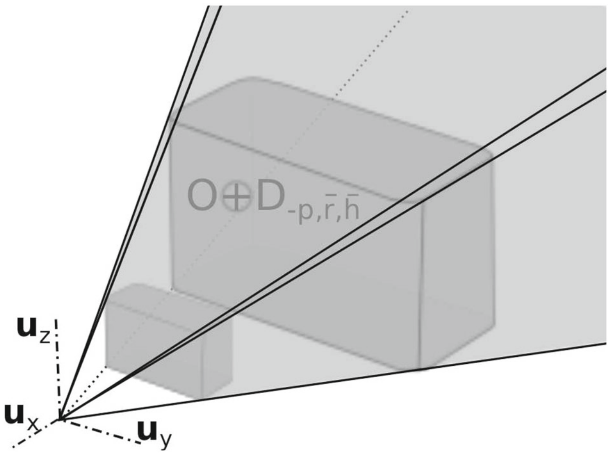


Fig. 6 Example of the collision avoidance constraint for a static rectangular obstacle $\mathcal{O}$. In *grey* the reference velocities $\mathbf{u}_i$ leading to a collision within the given time horizon. Its complement is non convex and is linearized

they must be linearized. For increased control over the avoidance behavior and topology, each non-convex constraint is approximated by five linear constraints, representing avoidance to the right, to the left, over and under the obstacle (or other agent) and a head-on maneuver, which remains collision-free up to $t = \tau$.

In particular, for the feasible space $\mathbb{R}^3 \setminus \mathcal{VO}^\tau_{ij}$ consider five ($l \in [1, 5]$) linear constraints (half-spaces) H^l_{ij} of the form $\mathbf{n}^l_{ij}(\mathbf{u}_i - \mathbf{u}_j) \leq b^l_{ij}$, with $\mathbf{n}^l_{ij} \in \mathbb{R}^3$ and $b^l_{ij} \in \mathbb{R}$, and verifying that if $\mathbf{u}_i - \mathbf{u}_j$ satisfies any of the linear constraints it is feasible with respect to the original constraint,

$$\bigcup_{l=1}^{5} H^l_{ij} \subset \mathbb{R}^3 \setminus \mathcal{VO}^\tau_{ij}. \tag{7}$$

An example is shown in Fig. 7 and the equations of the linear constraints are given in Appendix 3.

To linearize the original constraint, only one of the linear constraints is selected and added to the convex optimization.

4.5.2 Selection of a linear constraint

Each collision avoidance constraint is linearized by selecting one of the five linear constraints. Sensible choices include:

1. *Fixed side for avoidance.* If agents are moving towards each other ($\mathbf{v}_{ij} \cdot \mathbf{p}_{ij} < 0$), avoid on a predefined side, for example on the left ($l = 1$). If agents are not moving towards each other, the constraint perpendicular to the apex of the cone ($l = 5$) is selected to maximize maneuverability.
2. *Maximum constraint satisfaction for the current relative velocity:*

$$\underset{l}{\operatorname{argmin}}\,(\mathbf{n}^l_{ij} \cdot (\mathbf{v}_i - \mathbf{v}_j) - b^l_{ij}).$$

[4] To account for the downwash effect that does not allow for close operation in the vertical direction. The results readily extend to robots of arbitrary shape with the assumption of constant orientation for $t \in [0, \tau]$.

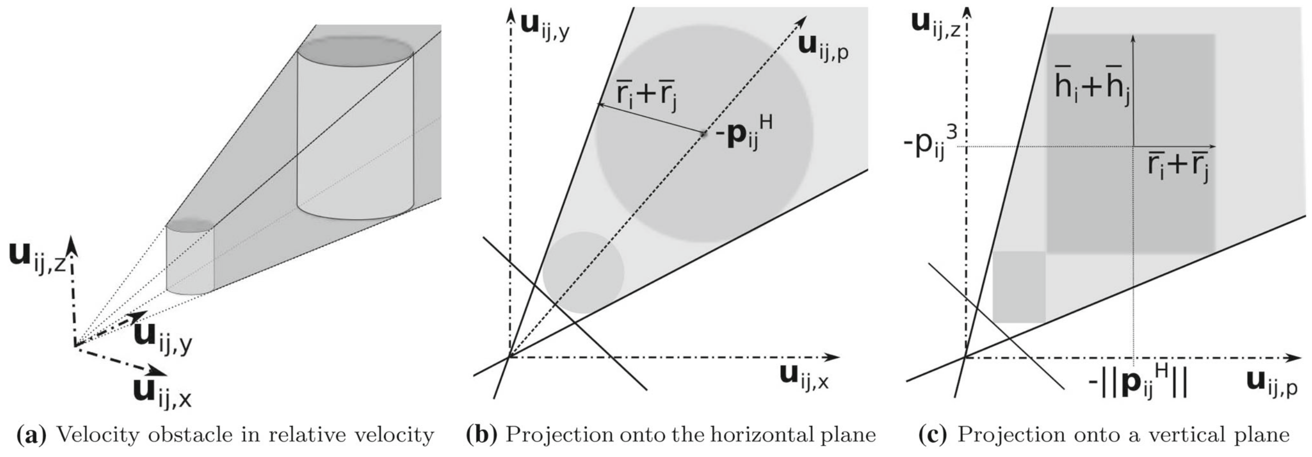


Fig. 7 Example of the collision avoidance constraint for a pair of interacting agents. The relative candidate reference velocities $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$ leading to a collision are displayed in *grey*. Its complement its non-convex and is linearized. *Left* Constraint in relative velocity space. *Middle/right* projection onto the *horizontal/vertical plane*, including the projection of three linearizations (or half-planes of collision-free relative velocities). **a** Velocity obstacle in relative velocity. **b** Projection onto the *horizontal plane*. **c** Projection onto a *vertical plane*

This selection maximizes the feasible area of the optimization, when taking into account the motion continuity constraints of Sect. 4.4.

3. *Maximum constraint satisfaction for the preferred velocity:*

$$\operatorname*{argmin}_l (\mathbf{n}^l_{ij} \cdot (\bar{\mathbf{u}}_i - \bar{\mathbf{u}}_j) - b^l_{ij}) \text{ if centralized.}$$
$$\operatorname*{argmin}_l (\mathbf{n}^l_{ij} \cdot (\bar{\mathbf{u}}_i - \mathbf{v}_j) - b^l_{ij}) \text{ if distributed.}$$

This selection may provide faster progress towards the goal position, but the optimization may become quickly infeasible if the agent greatly deviates from its preferred trajectory.

Option 1 provides the best coordination results as it incorporates a social rule, option 2 maximizes the feasible area of the optimization and option 3 may provide faster convergence to the ideal trajectory. An experimental evaluation of these options is given in Sect. 8.

Without loss of generality, consider $\mathbf{n}_{ij} \cdot \mathbf{u}_{ij} \leq b_{ij}$ the selected linearization of $\mathbb{R}^3 \setminus \mathcal{VO}^\tau_{ij}$, which can be directly added in the *centralized* optimization of Algorithm 1.

4.5.3 Partition: distributed case

In the *distributed* case, the reference velocity of agent j is typically unknown, only its current velocity $\mathbf{v}_i$ can be inferred. With the assumption that every agent follows the same algorithm, a partition must be found such that if both agents select new velocities independently, their relative reference velocity $\mathbf{u}_i - \mathbf{u}_j$ is collision-free. The idea of reciprocal avoidance first presented by van den Berg et al. (2009) is followed.

The change in velocity is denoted by $\Delta\mathbf{v}_i = \mathbf{u}_i - \mathbf{v}_i$ and the relative change of velocity by $\Delta\mathbf{v}_{ij} = \Delta\mathbf{v}_i - \Delta\mathbf{v}_j$. In the *distributed* case all agents are considered as independent decision-makers solving their independent optimizations. To globally maintain the constraint satisfaction and avoid collisions, an assumption on agent j's velocity is required.

Variable sharing of avoidance effort might be considered leading to $\Delta\mathbf{v}_i = \lambda\Delta\mathbf{v}_{ij}$ with the assumption that $\Delta\mathbf{v}_j = -(1-\lambda)\Delta\mathbf{v}_{ij}$. For collaborative agents that equally share the avoidance effort, $\lambda = 0.5$. If it is considered that agent j ignores agent i and continues with its current velocity, then $\lambda = 1$ and full avoidance is performed by agent i.

With the assumption that both agents share the avoidance effort (change in velocity), for $\lambda \in [0, 1]$ the linear constraint is rewritten as

$$\begin{aligned}
\mathbf{n}_{ij} \cdot \mathbf{u}_{ij} &= \mathbf{n}_{ij} \cdot (\mathbf{u}_i - \mathbf{v}_j - \Delta\mathbf{v}_j) \\
&= \mathbf{n}_{ij} \cdot (\mathbf{u}_i - \mathbf{v}_j + (1-\lambda)\Delta\mathbf{v}_{ij}) \\
&= \mathbf{n}_{ij} \cdot (\mathbf{u}_i - \mathbf{v}_j + (1-\lambda)(\mathbf{u}_i - \mathbf{v}_i)/\lambda) \\
&= \frac{\mathbf{n}_{ij}}{\lambda} \cdot (\mathbf{u}_i - (1-\lambda)\mathbf{v}_i - \lambda\mathbf{v}_j) \leq b_{ij}.
\end{aligned} \tag{8}$$

Leading to

$$\mathbf{n}_{ij} \cdot \mathbf{u}_i \leq b_i = \lambda b_{ij} + \mathbf{n}_{ij} \cdot ((1-\lambda)\mathbf{v}_i + \lambda\mathbf{v}_j), \tag{9}$$

which is a fair partition of the velocity space for both agents and provides avoidance guarantees.

5 Algorithm: convex optimization

To achieve low computational time the inherently non-convex optimization is approximated by a convex optimization with quadratic cost and linear and quadratic constraints.

Two algorithms are presented, one *centralized* and one *distributed*.

Algorithm 1 *(Centralized convex optimization) A single convex optimization is solved where the optimal reference velocity of all agents* $\boldsymbol{u}^*_{1:m} = [\boldsymbol{u}^*_1, \dots, \boldsymbol{u}^*_m]$ *are jointly computed.*

$$\begin{aligned} \boldsymbol{u}^*_{1:m} := \underset{\boldsymbol{u}_{1:m}}{argmin}\ & C(\boldsymbol{u}_{1:m}) \\ s.t.\ & \|\boldsymbol{u}_i\| \le v_{max},\ \forall i \in \mathcal{A} \\ & \textit{quadratic Constraint 1},\ \forall i \in \mathcal{A} \\ & \textit{linearized Constraint 2},\ \forall i \in \mathcal{A} \\ & \textit{linearized Constraint 3, of the form} \\ & \quad \boldsymbol{n}_{ij} \cdot (\boldsymbol{u}_i - \boldsymbol{u}_j) \le b_{ij}, \\ & \quad \forall i, j \in \mathcal{A}\ \textit{neighbors} \end{aligned} \tag{10}$$

where the optimization cost is given by Eq. (4).

Algorithm 2 *(Distributed convex optimization) Each agent* $i \in \mathcal{A}$ *independently solves an optimization where its optimal reference velocity* $\boldsymbol{u}^*_i$ *is computed.*

$$\begin{aligned} \boldsymbol{u}^*_i := \underset{\boldsymbol{u}_i}{argmin}\ & C(\boldsymbol{u}_i) \\ s.t.\ & \|\boldsymbol{u}_i\| \le v_{max}, \\ & \textit{quadratic Constraint 1, agent } i, \\ & \textit{linearized Constraint 2, agent } i, \\ & \textit{linearized Constraint 3, of the form} \\ & \quad \boldsymbol{n}_{ij} \cdot \boldsymbol{u}_i \le b_i,\ \ \forall j \in \mathcal{A}\ \textit{neighbor of } i, \end{aligned} \tag{11}$$

where the optimization cost is given by Eq. (3).

These $2m$ and 2-dimensional quadratic optimizations with linear and quadratic constraints can be solved efficiently using available solvers.[5]

If the optimization is feasible, a solution is found that guarantees collision-free motion up to the time horizon τ. If the optimization is infeasible, no solution exists for the linearized problem. In this case, the involved agents drive their last feasible trajectory (obtained at time t_f) with a time reparametrization given by

$$\begin{aligned} &\gamma(t) = (-t_f^2 + (2\tau + 2t_f)t - t^2)/(2\tau), && t \in [t_f, t_f + \tau], \\ &\gamma(t) = t_f + \tau/2, && t > t_f + \tau. \end{aligned} \tag{12}$$

This time reparametrization guarantees that the involved agents reach a stop on their previously computed paths within the time horizon of the collision avoidance algorithm. Since this computation is performed at a high frequency, each individual agent is able to adapt to changing situations.

[5] We use IBM ILOG CPLEX.

5.1 Theoretical guarantees

Remark 1 (Computational complexity) Linear in the number of variables and constraints, although the *centralized* is of higher complexity.

If *distributed*, for each agent the optimization consists of three variables, one quadratic constraint and a maximum of $m + n_o$ linear constraints for collision avoidance (in practice limited to a constant value K_c, plus n_o), where n_o is the number of static obstacles.

If *centralized*, the optimization consists of $3n$ variables, n quadratic constraints and less than $\frac{n(n-1)}{2} + n(m-n) + nn_o$ linear constraints (usually limited to $n(k_c + n_o)$).

Remark 2 (Safety guarantees) Safety is preserved in normal operation.

If *feasible*, collision-free motion is guaranteed for the local trajectory up to time τ (optimal reference trajectory is collision-free for agents of radii enlarged by ε and agents stay within ε of it) with the assumption that all interacting agents either continue with their previous velocity or compute a new one following the same algorithm and assumptions.

Let $\mathbf{p}_i(t) = [\mathbf{p}_i^H(t), p_i^{\%}(t)]$ denote the position of agent i at time $t \ge t^k$ and recall $\tilde{t} = t - t^k$. If the two agents are side by side then

$$\begin{aligned} \|\mathbf{p}_i^H(t) - \mathbf{p}_j^H(t)\| &= \|f(\mathbf{z}_i^k, \mathbf{u}_i, \tilde{t})^H - f(\mathbf{z}_j^k, \mathbf{u}_j, \tilde{t})^H\| \\ &\ge \|(\mathbf{p}_i^H(t^k) + \mathbf{u}_i^H\tilde{t}) - (\mathbf{p}_j^H(t^k) + \mathbf{u}_j^H\tilde{t})\| - \varepsilon_i - \varepsilon_j \\ &\ge r_i + \varepsilon_i + r_j + \varepsilon_j - \varepsilon_i - \varepsilon_j = r_i + r_j, \end{aligned} \tag{13}$$

where the first inequality holds from the triangular inequality and Constraint 1 ($\mathbf{u}_i \in R_i$ and $\mathbf{u}_j \in R_j$). The second inequality holds from Constraint 3 ($\mathbf{u}_i - \mathbf{u}_j \notin \mathcal{VO}^\tau_{ij}$). The proof proceeds analogously for the vertical component in the case where agents are on top of each other. Recall that at least one of the two constraints (projection onto the horizontal or vertical plane) must hold.

If *infeasible*, no collision-free solution exists that respects all the constraints. If the time horizon is longer than the required time to stop, passive safety is preserved if the last feasible trajectory of all agents are driven with a time reparametrization to reach stop before a collision arises (Eq. 12). This time reparametrization implies a slow down of the agent, which in turn may render the optimization feasible in a later time-step. Since this computation is performed at a high frequency, each individual agent is able to adapt to changing situations.

Remark 3 (Infeasibility) In some cases the constraints can be such that the optimization is over-constrained and infeasible.

The agent may reach a state where the optimization is over-constrained and therefore infeasible. This can be due to several causes, such as the following.

(a) Due to the limited local planning horizon together with over simplification of motion capabilities by reducing them to the set of local motions of Sect. 4.

(b) Due to differences between the precomputed motion primitives and the executed motion, arising from differences between the model and the real vehicle or uncertainty in localization and estimation of the agents' state.

(c) If *distributed*, given the use of pair-wise partitions of velocity space with either the assumption of equal effort in the avoidance or constant speed, not all world constraints are taken into account for the neighboring agents. Thus an agent may have conflicting partitions with respect to different neighbors rendering its optimization infeasible.

Remark 4 (Deadlock-free guarantees) As for other local methods, deadlocks may still appear.

A global planner for guidance is required in complex scenarios.

6 Algorithm: non-convex optimization

Section 4 described a method for local motion planning for heterogeneous groups of aerial vehicles which was formulated as a convex optimization problem in Sect. 5. This section presents an extension to a non-convex optimization where the global optimum can be found, albeit at an increased computational cost.

As shown in Sect. 4.5, each non-convex constraint $\mathcal{VO}$ is approximated by 5 linear constraints of which only one is introduced in the optimization, leading to a local optimum. The full solution space can be explored by the addition of binary variables, and reformulating the problem as a mixed integer quadratic program (MIQP).

One binary variable $\xi_c^l = \{0, 1\}$, is added for each linear constraint $l \in [1:5]$ approximation of a non-convex constraint $c \in \mathcal{C}$. Only one out of the 5 linear constraints is active, thus $\sum_{l=1}^{5} \xi_c^l = 4,\ \forall c \in \mathcal{C}$.

Algorithm 3 *(Centralized Mixed-Integer optimization) A single MIQP optimization is solved where the optimal reference velocity of all agents* $\boldsymbol{u}_{1:m}^* = [\boldsymbol{u}_1^*, \ldots, \boldsymbol{u}_m^*]$, *as well as the active constraints* $\bar{\xi} = \bigcup_{l\in[1,5],\, c\in\mathcal{C}} \xi_c^l$ *are jointly computed. Define* $\mathbf{f}_{\bar{\xi}}^T$ *a vector with (optional) weights for each binary variable, for instance to give preference to avoidance on one side.*

$$
\begin{aligned}
&\underset{\boldsymbol{u}_{1:m}}{argmin}\ C(\boldsymbol{u}_{1:m}) + \mathbf{f}_{\bar{\xi}}^T \bar{\xi} \\
&\quad s.t.\ ||\boldsymbol{u}_i|| \le u_{max},\ \forall i \in \mathcal{A} \\
&\qquad \textit{quadratic Constraint 1},\ \forall i \in \mathcal{A} \\
&\qquad \textit{linear Constraints 2, of the form} \\
&\qquad\quad \boldsymbol{n}_i^l \cdot \boldsymbol{u}_i - N\xi_i^l \le b_i^l,\quad \forall i \in \mathcal{A},\ \forall l \in [1,5] \\
&\qquad \textit{linear Constraints 3, of the form} \\
&\qquad\quad \boldsymbol{n}_{ij}^l \cdot (\boldsymbol{u}_i - \boldsymbol{u}_j) - N\xi_{ij}^l \le b_{ij}^l, \\
&\qquad\qquad \forall i, j \in \mathcal{A},\ \textit{neighbors},\ \forall l \in [1,5] \\
&\qquad \textstyle\sum_{l=1}^{5} \xi_c^l = 4,\ \forall c \in \mathcal{C},
\end{aligned}
\tag{14}
$$

where $N >> 0$ is a large constant of arbitrary value. The linear cost vector $\mathbf{f}_{\bar{\xi}}$ is a zero vector if no preference between linear constraints for each $\mathcal{VO}$ exists. Analogously to the work for 2D local planning by Alonso-Mora et al. (2013), different weights can be added for each linear constraint to include preference for a particular avoidance side.

This optimization can be solved via branch and bound using state of the art solvers.[6] Although the number of variables and constraints can be bounded to be linear with the number of agents, the number of branches to be explored increases exponentially. In practice a bound in the resolution time or number of explored branches (avoidance topologies) must be set.

A distributed MIQP optimization can also be formulated, but collision avoidance guarantees would be lost, since this can lead to a disagreement in the avoidance side and reciprocal dances appear, as observed by van den Berg et al. (2009) for the 2D case.

7 Control framework: quadrotor vehicle

The aforementioned algorithms are experimentally evaluated with quadrotor helicopters. A cascade control framework is implemented (see Fig. 8) which has several interconnected modules that work at different frequencies and are composed of multiple submodules.

Motion planning

– *Global motion planning:* In complex environments, a global planner is required for convergence. This is not the focus of this work, therefore a fixed trajectory or goal position is considered.
– *Local motion planning:* Computes (10 Hz) a collision-free local motion, given a desired global trajectory or a goal position.[7]
  - *Guidance system:* Outputs a preferred velocity $\bar{\mathbf{u}}$, given a desired global trajectory or a goal position, and the current state.

[6] We use IBM ILOG CPLEX.

[7] Although the local planning is designed for on-board performance, this is left as future work.

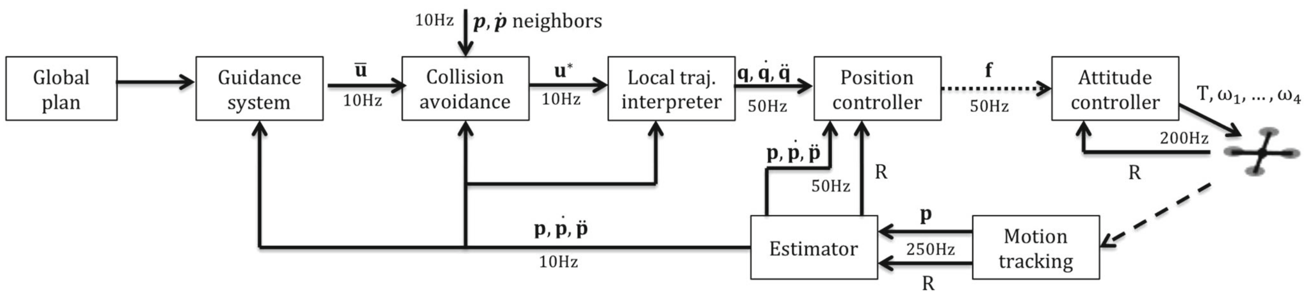


Fig. 8 Schema of control framework, with blocks, variables and frequencies

- *Collision avoidance:* Receives a preferred velocity $\bar{\mathbf{u}}$ and computes the optimal reference velocity $\mathbf{u}^*$, i.e. the closest collision-free candidate reference velocity that satisfies the motion continuity constraints. This is the topic of Sect. 4.

Control

– *Position control:* Controls the position of the quadrotor to the local trajectory and runs at 50 Hz off-board in a real-time computer[8].

  - *Local trajectory interpreter:* Receives the optimal reference velocity $\mathbf{u}^*$ at time instances t^k. Outputs the control states $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$, to satisfy motion continuity at time t^k and asymptotic convergence to the optimal reference trajectory given by $\mathbf{p}^k + \mathbf{u}^*\tilde{t}$.
  - *Position control:* Receives setpoints in position $\mathbf{q}$, velocity $\dot{\mathbf{q}}$ and acceleration $\ddot{\mathbf{q}}$ and outputs the desired force $\bar{F}$.

– *Low-level control:* Runs onboard (200 Hz) and provides accurate attitude control at high frequency. The low-level controller abstracts the nonlinear quadrotor model as a point-mass for the high-level control. It is composed of three submodules:

  - *Attitude control:* Receives a desired force $\bar{F}$ (collective thrust and desired orientation) and outputs desired angular rates $\bar{\omega}_x$, $\bar{\omega}_y$, $\bar{\omega}_z$.
  - *Body angular rate control:* Receives the desired body angular rates $\bar{\omega}_x$, $\bar{\omega}_y$, $\bar{\omega}_z$ and collective thrust T. The output is a rotation speed setpoint for each of the four motors $[\omega_1, \ldots, \omega_4]$.
  - *Motor control:* Receives the desired rotation speed of the motor $[\omega_1, \ldots, \omega_4]$. Off-the-shelf motor controllers are used to interface the four motors of the quadrotor.

[8] Position control and the local trajectory interpreter could be on-board given access to position sensing.

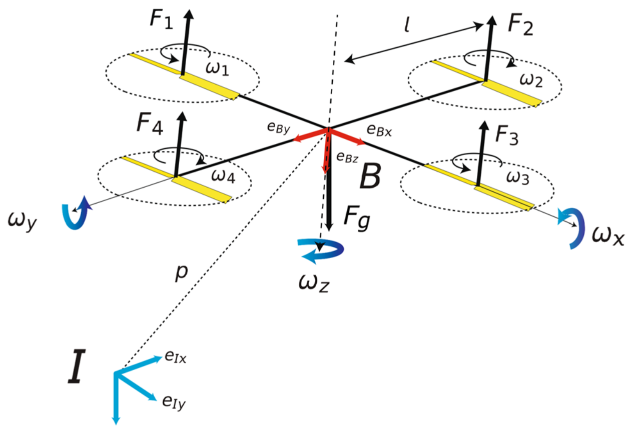


Fig. 9 The basic model of a quadrotor with applied forces

A quadrotor is a highly dynamic vehicle. To account for delays, the state used for local planning is not the current state of the vehicle, but a predicted one (by forward simulation of the previous local trajectory) at the time when the new local trajectory will be sent. For ease of exposition, this distinction is omitted in the following sections.

A quadrotor is an under-actuated system in which translation and rotation are coupled. Its orientation (attitude), is defined by the rotation matrix $R_I^B = [e_x, e_y, e_z] \in \mathcal{SO}(3)$, which describes the transformation from the inertial coordinate frame I to the body fixed coordinate frame B. For completeness, we provide an overview of the model and control. For details, refer to Mellinger and Kumar (2011), among others.

7.1 Quadrotor model

The four rotors of a quadrotor are mounted in fixed positions with respect to the body frame and their motor speeds ω_i can be controlled individually as indicated in Fig. 9. Each motor produces a force $F_i = c_t\omega_i^2$, and a moment $M_i = c_y\omega_i^2$, with c_t and c_y model specific constants.

Yaw, pitch, roll and total thrust $T \in \mathbb{R}$ are controlled by differential control of the individual rotors. The system is under-actuated and the remaining degrees of freedom are controlled through the system dynamics.

Following (Mellinger and Kumar 2011), the total thrust T and the moments around the body axes $\mathbf{M}_a = [M_x, M_y, M_z] \in \mathbb{R}^3$ are given by

$$\begin{bmatrix} T \\ \mathbf{M}_a \end{bmatrix} = \underbrace{\begin{bmatrix} c_t & c_t & c_t & c_t \\ 0 & c_a l & 0 & -c_a l \\ -c_a l & 0 & c_a l & 0 \\ c_y & -c_y & c_y & -c_y \end{bmatrix}}_{U} \underbrace{\begin{bmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{bmatrix}}_{\omega_m}, \tag{15}$$

where l is the length from the center of gravity of the quadrotor to the rotor axis of the motor and c_a a model constant.

The angular velocity $\boldsymbol{\omega}$ and its associated skew anti-symmetric matrix $\breve{\omega}$ are denoted by

$$\breve{\omega} := \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_z & 0 \end{bmatrix}, \quad \boldsymbol{\omega} := \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}.$$

The angular acceleration around the center of gravity is given by

$$\dot{\boldsymbol{\omega}} = J^{-1}\left[-\boldsymbol{\omega} \times J\boldsymbol{\omega} + \mathbf{M}_a\right], \tag{16}$$

with J the moment of inertia matrix referenced to the center of mass along the body axes of the quadrotor. If R_B^I denotes the transformation from the body to the inertial frame, the rotational velocity of the quadrotor's fixed body frame with respect to the inertial frame is given by $\dot{R}_B^I = \breve{\omega} R_B^I$.

A point-mass model can be used for the position dynamics. The accelerations of the quadrotor's point-mass in the inertial frame $\ddot{p}_I$ are given by

$$\ddot{\mathbf{p}}_I = R_B^I \left[0,\ 0,\ -\tfrac{T}{m}\right]^T + \left[0,\ 0,\ g\right]^T, \tag{17}$$

where two forces are applied, gravity $F_G = mg$ in the z direction in the inertial frame and thrust T, in the negative z direction in the quadrotor body-frame.

7.2 Attitude control

The approach by Lee et al. (2010) that directly converts differences of rotation matrices into body angular rates is followed. A desired force setpoint $\bar{\mathbf{F}}$ and a desired yaw angle ψ given by a high-level controller are transformed to a desired rotation matrix $\bar{R} = [\bar{\mathbf{e}}_x, \bar{\mathbf{e}}_y, \bar{\mathbf{e}}_z]$ with

$$\bar{\mathbf{e}}_z = \frac{\bar{F}}{\|\bar{F}\|}, \quad \bar{\mathbf{e}}_y = \frac{\bar{\mathbf{x}}_\psi \times \bar{\mathbf{e}}_z}{\|\bar{\mathbf{x}}_\psi \times \bar{\mathbf{e}}_z\|}, \quad \bar{\mathbf{e}}_x = \bar{\mathbf{e}}_y \times \bar{\mathbf{e}}_z,$$

and $\bar{\mathbf{x}}_\psi = [\cos\psi, \sin\psi, 0]^T$ a vector pointing in the desired yaw direction. The desired body angular rates are controlled with a standard PD controller and given by

$$\bar{\boldsymbol{\omega}} = f\left(\frac{1}{2}(\bar{R}_I^B \hat{R}_B^I - \hat{R}_I^B \bar{R}_B^I)\right), \tag{18}$$

where $\hat{R}$ indicates the estimated rotation matrices.

7.3 Position control

According to the double integrator dynamics of Eq. (17), the discrete LTI model can be written as

$$\mathbf{x}_{k+1}^p = A\mathbf{x}_k^p + B\mathbf{u}_p, \quad \mathbf{y} = C\mathbf{x}_k^p, \tag{19}$$

with state vector $\mathbf{x}^p := [p_x, v_x, p_y, v_y, p_z, v_z]^T$, input vector $\mathbf{u}_p = [F_x, F_y, F_z]^T$ and system matrices

$$A := I_3 \otimes \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix}, \quad B := I_3 \otimes \begin{bmatrix} st \\ dt \end{bmatrix}, \quad C := I_3 \otimes \begin{bmatrix} 1 & 0 \end{bmatrix},$$

with I_3 the 3x3 identity matrix, $st = \frac{1}{2}dt^2$, dt the time step and $\otimes$ the Kronecker product. To have an integral action for the LQR controller, the system given in Eq. (19) is augmented with the integral term $\mathbf{x}_{pI} \in \mathbb{R}^3$ to the discrete LTI system

$$\tilde{\mathbf{x}^p} := \begin{bmatrix} \mathbf{x}^p \\ \mathbf{x}^{pI} \end{bmatrix}, \quad \tilde{A} := \begin{bmatrix} A & 0 \\ C & I \end{bmatrix}, \quad \tilde{B} := \begin{bmatrix} B \\ 0 \end{bmatrix}, \quad \tilde{C} := \begin{bmatrix} C & 0 \end{bmatrix}.$$

The control input $\mathbf{u}_p$ is then given with the state feedback law $\mathbf{u}_p = -K\tilde{x^p}$ with K the solution to the Discrete Algebraic Riccati equation (DARE) with state and input cost matrices.

7.4 Motion primitives

A point-mass M-order integrator model is considered, decoupled in each component,[9]

$$\mathbf{q}^{(M+1)}(\tilde{t}) = \nu \tag{20}$$

Let $\nu \in \mathbb{R}^3$ be piecewise continuous and $(.)^{(M+1)}$ represent the derivative of order $M+1$ with respect to time. The resulting solution $\mathbf{q}(\tilde{t})$ is $\mathcal{C}^M$-differentiable at the initial point.

For $M > 0$ the state feedback control law is given by

$$\begin{aligned} \nu(\tilde{t}) = &-\mathbf{a}_M \mathbf{q}^{(M)} - \cdots - \mathbf{a}_2\,\ddot{\mathbf{q}} \\ &+\mathbf{a}_1(\mathbf{u} - \dot{\mathbf{q}}) + \mathbf{a}_0(\mathbf{p}^k + \mathbf{u}\tilde{t} - \mathbf{q}). \end{aligned} \tag{21}$$

Mellinger and Kumar (2011) showed that the under-actuated model of a quadrotor can be written as a differentially flat (Van Nieuwstadt and Murray 1997) system, with its trajectory parametrized by its position and yaw angle $[\mathbf{p}, \psi]$ and their derivatives up to forth degree in position and second degree in yaw angle. In the following, the yaw angle ψ is

[9] Alternatively, any other controller with arbitrary constraints, or an LQR controller can be employed.

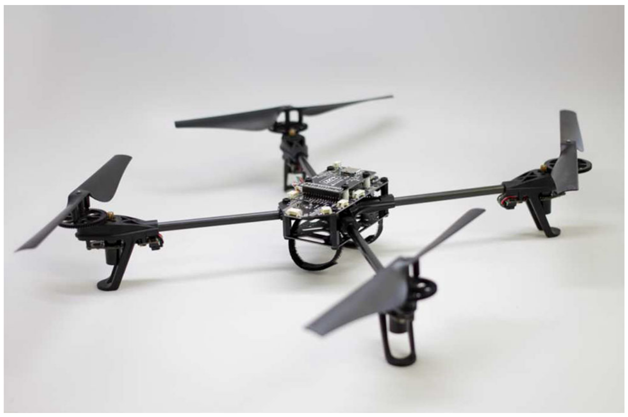


Fig. 10 The quadrotor used in the experiments consists of a PX4 IMU with an AR. Drone frame and motors

considered constant. Although the method presented in this paper extends to $M = 4$, in order to simplify the formulations, given the difficulty to accurately estimate the high order derivatives and their marginal effect, only continuity in acceleration ($M = 2$) is imposed in the remaining of the paper.

For $M = 2$ (continuity in initial acceleration), as derived in Rufli et al. (2013), the motion primitive is

$$\mathbf{q}(\tilde{t}) = \mathbf{p}^k + \mathbf{u}\tilde{t} + Fe^{-\omega_1 \tilde{t}} + ((\dot{\mathbf{p}}^k - \mathbf{u} + (\omega_1 - \omega_0)F)\tilde{t} - F)e^{-\omega_0 \tilde{t}}, \qquad (22)$$

with $F = (\ddot{\mathbf{p}}^k + 2\omega_0(\dot{\mathbf{p}}^k - \mathbf{v}_\infty))/(\omega_0 - \omega_1)^2$ and ω_0, ω_1 design parameters related to the decay of the initial velocity and acceleration and can potentially be different for each of the three axis.

8 Experimental results and discussion

Experiments were performed both in simulation and with physical quadrotors (Fig. 10). Simulation utilizes a model of the dynamics, and enables experiments for larger groups of vehicles. As an example, Fig. 11 shows a simulation for position exchange for ten vehicles. See the accompanying video for this experiment and other experiments in this section. All other results in this section are for physical quadrotors only.

8.1 Experimental setup

The experimental space is a rectangular area of approximately 5 m × 5.5 m and 2 m height. Quadrotor position is measured by an external motion tracking system[10] running at 250 Hz. Figure 10 shows the quadrotor used in the experiments. It is based on a PX4 IMU[11] with an AR.Drone frame and motors.[12] The vehicle is low-cost and easy to build, with a total weight of about 450g. Each vehicle is modeled with the following parameters. $\omega_0 = 3$, $\omega_1 = 3.5$, $r_i = 0.35$ m, $h_i = 0.5$ m, $v_{max} = 2$ m/s, $a_{max} = 2$ m/s^2, $\varepsilon = 0.1$ m and $\tau = 3$ s.

A system diagram was shown earlier in Fig. 8. A low-level attitude estimator and controller run on each quadrotor's IMU and the quadrotor is abstracted as a point-mass for the position controller. The position controller runs on a real-time computer and communication between the real-time computer and the quadrotors is done with a 50 Hz UART bridge.[13] The overlying collision avoidance runs in a normal operating system environment and sends a collision-free optimal reference trajectory to the position controller over a communication channel.

The position controller ensures that the quadrotors stay on their local trajectory as computed by the collision avoidance algorithm. This hard real-time control structure ensures the stability of the overall system and keeps the quadrotors on given position references.

8.2 Single quadrotor

The benchmark performance of the system is evaluated for a single quadrotor running the full framework, including local planning in which continuity up to acceleration is imposed. Figure 12 shows a single quadrotor tracking a circular motion given by a sinusoid reference signal in all three position components.

Figure 12 shows at top-left the projection of the position **p** of the quadrotor onto the horizontal plane, and at bottom-left the projection onto a vertical plane. The reference circular motion is not feasible due to the space limitation of the room, but the local planning successfully avoids colliding with the walls, causing the flattened edges in the y component. Figure 12 at top-right shows each component of the position **p** with respect to time as a continuous line, while the point-mass local trajectory **q** is shown by a dashed line. Good tracking performance is observed, although with a small delay due to unmodeled system delays and higher order dynamics. A steady-state error is further observed due to the use of an LQR controller without integral part in the experiments. Steady-state error is greatest in the vertical component, as a thrust calibration is not performed before flight, and its value varies therefore from quadrotor to quadrotor. To handle steady state error, continuity is imposed in the local trajectory instead of on the real state of the quadrotor. This approach adapts well to poorly calibrated systems, like the one in our experiments.

Figure 12 at bottom-right shows the measured velocity **v** of the quadrotor as a continuous line and the velocity of its point-mass local trajectory $\dot{\mathbf{q}}$ as a dashed line with respect to time. Good tracking performance is observed, although the delay is more apparent, especially for abrupt changes in

10 http://www.vicon.com/.

11 https://pixhawk.ethz.ch/px4/.

12 http://ardrone2.parrot.com/.

13 http://www.lairdtech.com/.

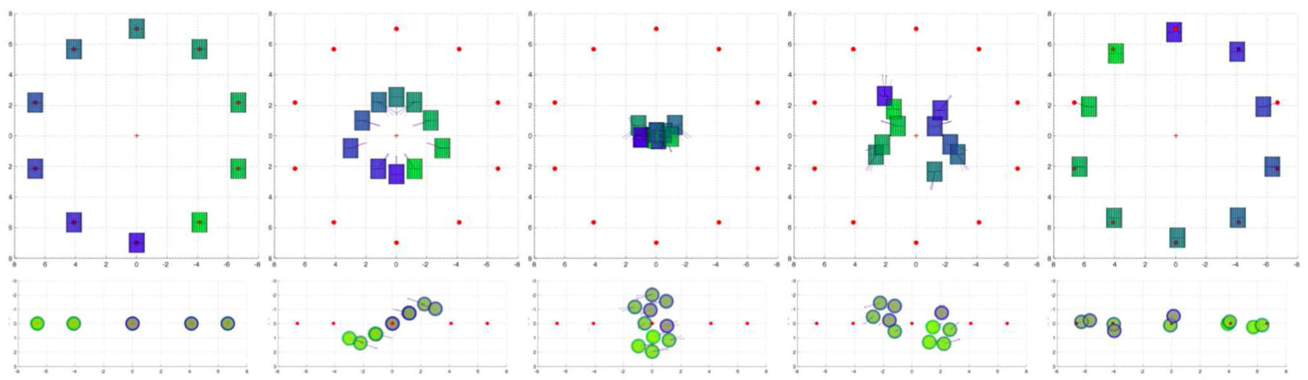


Fig. 11 Position exchange for ten quadrotors in simulation. *Top* a *horizontal*(x–y) view. *Bottom*, a *vertical* (x–z) view. Vehicle position is depicted for timestamps 0s:4s:6s:8s:12s in an arena of size 8 m × 4 m × 8 m. Convergence to the goal configuration is achieved in about 15 s

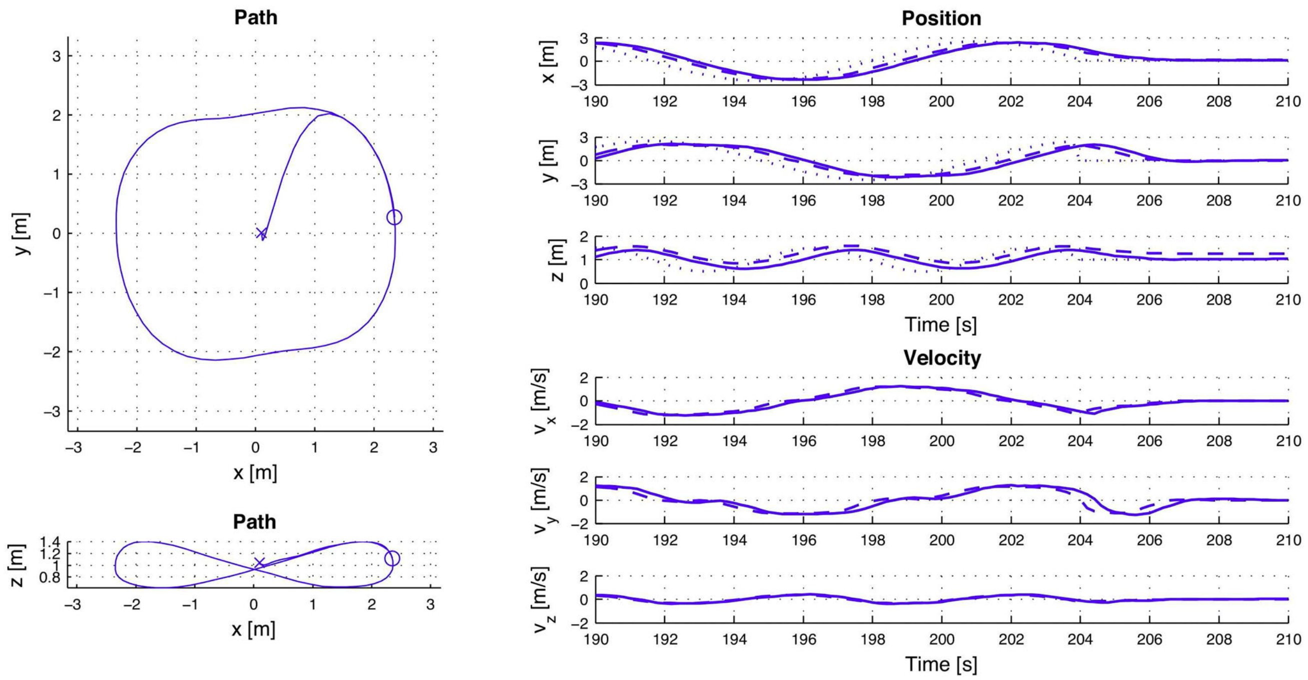


Fig. 12 Experiment with a single quadrotor. *Left* Horizontal (x–y) and vertical (x–z) views of the path followed by the quadrotor when tracking a sinusoid signal in each position component. The flattened edges in the y component are due to the wall constraints in the experimental space. *Right* Position and velocity of the quadrotor ($\mathbf{p}$, $\mathbf{v}$) with respect to time are shown as a *solid line*, and position and velocity of the local trajectory ($\mathbf{q}$, $\dot{\mathbf{q}}$) are shown as a *dashed line*

velocity, partially due to the unmodeled high-order dynamics of the quadrotor. The smoothness of the velocity profile can be adjusted by the parameters ω_0 and parameter ω_1 in Eq. (22) which can be tuned for overall good performance. The compensation of this and other system delays is a topic for future work.

8.3 Position swap

This section describes position swap experiments with sets of two or four quadrotors, with about fifty transitions to different goal configurations, both antipodal and random. The centralized (Algorithm 1) and distributed (Algorithm 2) algorithms are both tested, with the three different linearization options for the collision avoidance constraints described in Sect. 4.5.2. The joint MIQP optimization (Algorithm 3) was also tested, but proved too slow for real time performance at 10Hz in our implementation. In simulation, the approach provided good results.

Figure 13 shows a representative position exchange for two quadrotors. Figure 13 at left shows the projection of their paths on the horizontal and vertical planes. The slight change in height is due to the dynamics of the quadrotors. Figure 13 at right shows the velocity profile for both quadrotors. The preferred speed of the quadrotors is 2 m/s, and the position

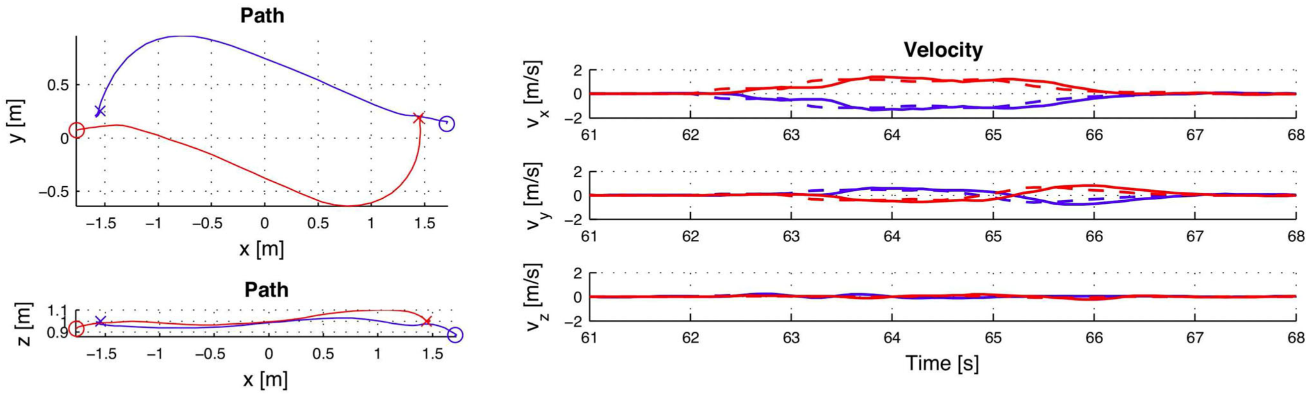


Fig. 13 Position exchange for two quadrotors. *Left horizontal* (x–y) and *vertical* (x–z) plots of the path followed by each quadrotor, with initial position marked by "o" and final position by "x". *Right* Velocity profile with respect to time for both quadrotors. The measured velocity $\mathbf{v}$ is shown by a *continuous line*, and the control velocity $\dot{\mathbf{q}}$ is shown by a *dashed line*)

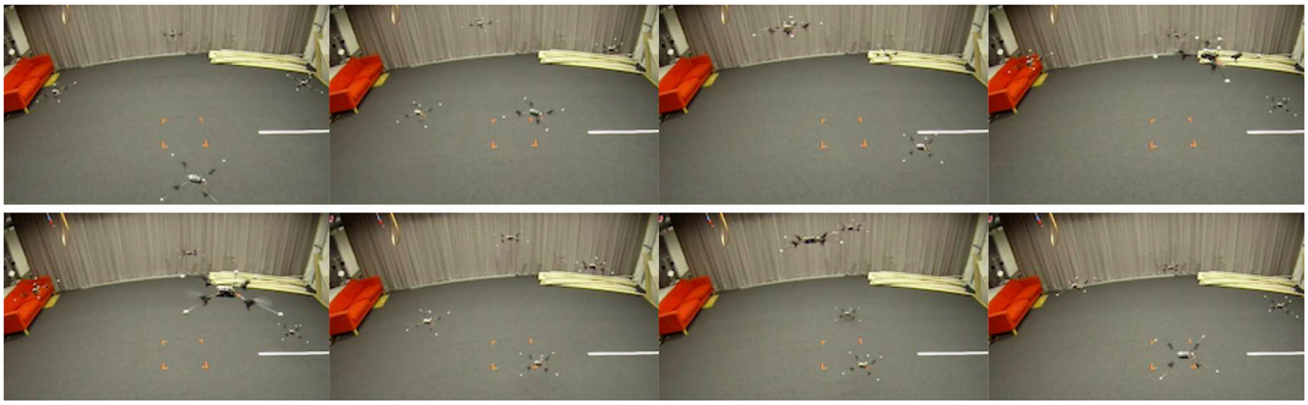


Fig. 14 Position exchange of four quadrotors, with video frames approximately every 2 s. *Top* Linearization of collision avoidance constraints enforcing that all agents avoid to the right. *Bottom* Linearization of collision avoidance constraints with respect to the current velocity

exchange takes a few seconds. Very similar performance was obtained in the distributed and centralized case and with the different linearization options.

Figure 14 shows two experiments in which four quadrotors transition to antipodal positions whilst changing height. Two different linearization strategies are employed—Fig. 14 at top shows the case when avoidance to the right is imposed (Sect. 4.5.2, case (1), while Fig. 14 at bottom shows the case when collision avoidance constraints are linearized with respect to the measured velocities (Sect. 4.5.2, case (2). Figure 15 shows the data for the two experiments, again with the different linearization strategies at top and bottom. Figure 15 at left-and-center columns shows the projection of the position of all quadrotors onto the horizontal plane and a vertical plane. Figure 15 at right shows the velocity profile for all quadrotors. The preferred speed is 2 m/s. The position exchanges were collision-free thanks to the local planning algorithm even for this complex case in which the preferred velocity always points towards the goal positions and therefore is not collision-free, and the dimensions of the room only allow the quadrotors to fly at a height between 0 m and 2 m.

For the case when avoidance to the right is imposed, the behavior is highly predictable and smooth, as the rotation of all vehicles to one side is imposed in the linearization, which showed good results both in the centralized and distributed cases. Although this characteristic is effective for cases of symmetry, in arbitrary configurations it can lead to longer trajectories. For the case of linearization with respect to the current velocity, the behavior can be different for different runs and emerges from non-deterministic characteristics of the motion. It is observed that in this particular example two quadrotors chose to pass over each other. Although this linearization option presents very good performance in not too crowded environments, its performance degrades in the case of high symmetry and very crowded scenarios (for example due to disagreements in avoidance side, especially in the distributed case). A detailed analysis of the best linearization option for each case is outside of the scope of this paper and remains as future work.

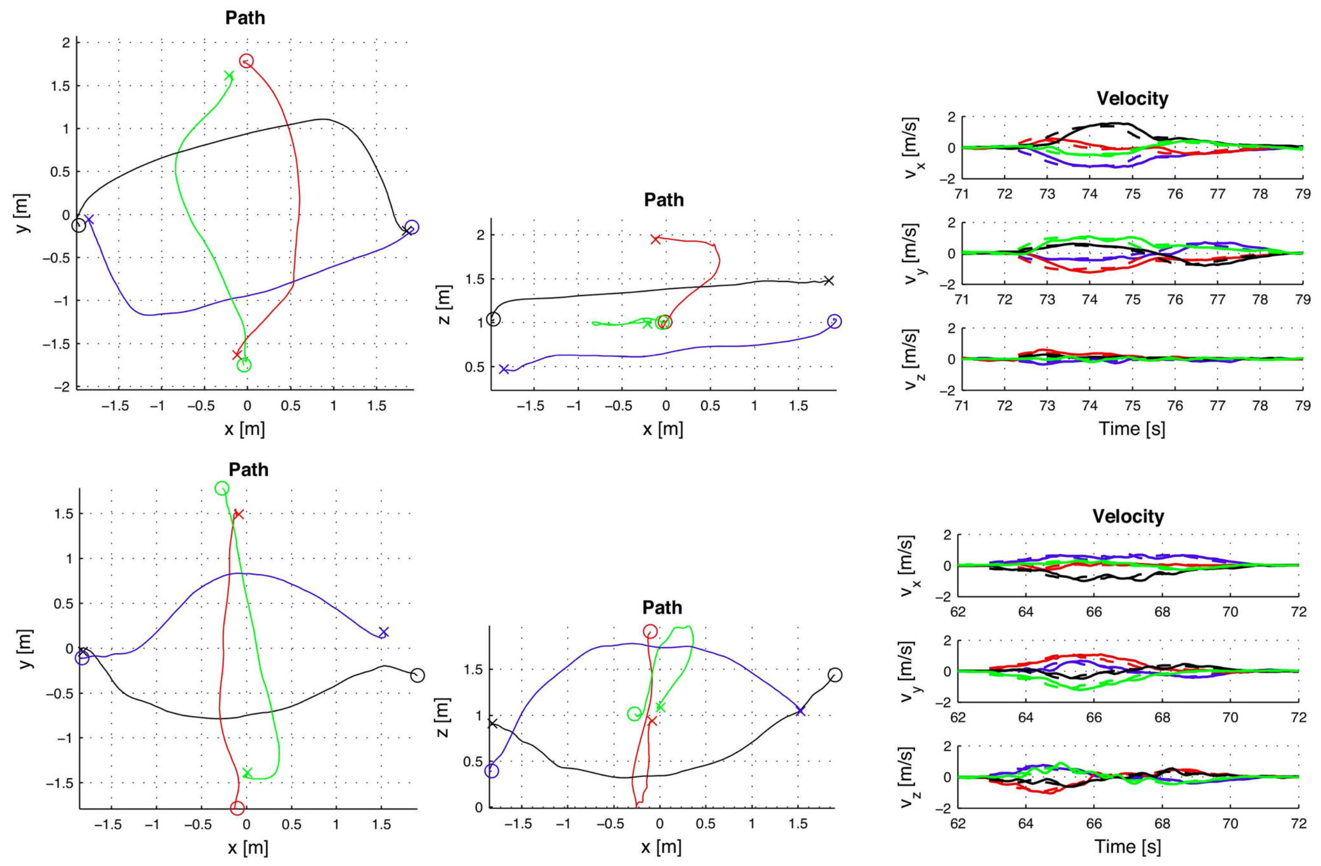


Fig. 15 Position exchange of four quadrotors, *Top* Linearization of collision avoidance constraints enforcing that all agents avoid to the right. *Bottom* Linearization of collision avoidance constraints with respect to the current velocity. *Left and middle column Horizontal* (x–y) and *vertical* (x–z) plots of the path followed by each quadrotor, with initial position marked by "o" and final position by "x". *Right* Velocity profile with respect to time for both quadrotors. The measured velocity $\mathbf{v}$ given by a *continuous line*, and the control velocity $\dot{\mathbf{q}}$ by a *dashed line*)

8.4 Trajectory following

This section describes experiments to test the method for trajectory tracking i.e. the case of a moving goal. Two representative scenarios are described. Vertical motion is not displayed in the results because, without loss of generality, motion was approximately in a horizontal plane.

Figure 16 shows two quadrotors that are tracking intersecting trajectories while avoiding collision locally. The blue quadrotor follows a circular motion and the red quadorotor follows a diagonal motion such that there are two intersection points in position and time. The red quadrotor finally transitions to a rest position. Figure 16 at left shows that both quadrotors must deviate from their ideal trajectories in order to avoid collision. Figure 16 at right is the velocity plot which shows that no deadlock appeared, and both quadrotors were moving at a velocity of about 1.5 m/s, except towards the end of the experiment where the red quadrotor approaches a static goal position. The trajectory for the blue quadrotor then becomes feasible and it is well tracked as shown by the velocity values after 100 s.

Figure 17 shows two quadrotors (blue and red) that are tracking non-colliding circular trajectories with 180 degrees phase shift, while a human (black) moves in the arena. The human is considered as a dynamic obstacle. The two quadrotors avoid collisions locally whilst tracking, as closely as possible, their ideal trajectories. Figure 17 at left shows the trajectories and Fig. 17 at right shows the measured velocity for all agents. No collision or deadlock was observed, although in some cases a quadrotor had to stop as the optimization became infeasible. In this experiments the human must move at a speed similar to that of the quadrotors (<2 m/s) and be slightly predictable, due to the constant velocity assumption.

8.5 Overall behavior and algorithm evaluation

This section discusses overall behavior based on statistics collected over all experiments.

Deadlock was not observed in our experiments, but it is possible from a theoretical standpoint and can occur

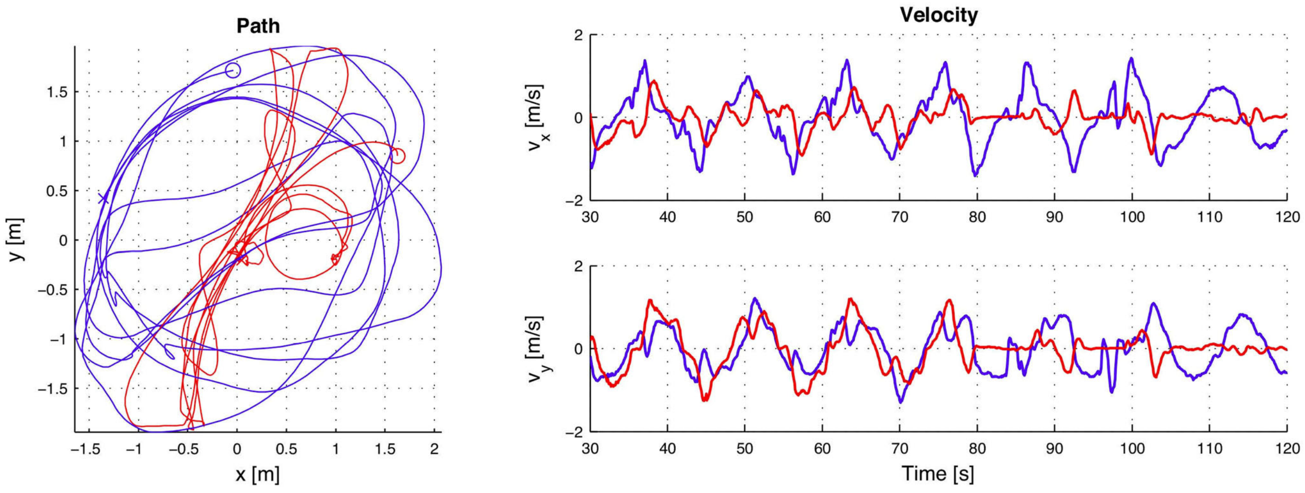


Fig. 16 Two quadrotors track intersecting trajectories while avoiding collision locally. The *blue* quadrotor follows a *circular* motion and the *red* quadrotor follows a *diagonal* motion such that there are two intersection points in position and time. The *red* quadrotor finally transitions to a rest position. *Left Horizontal* (x–y) plot of the path followed by each agent, with initial position marked by "o" and final position by "x". *Right Horizontal* components of the measured velocity **v** with respect to time for all agents. *Vertical* motion is approximately zero (Color figure online)

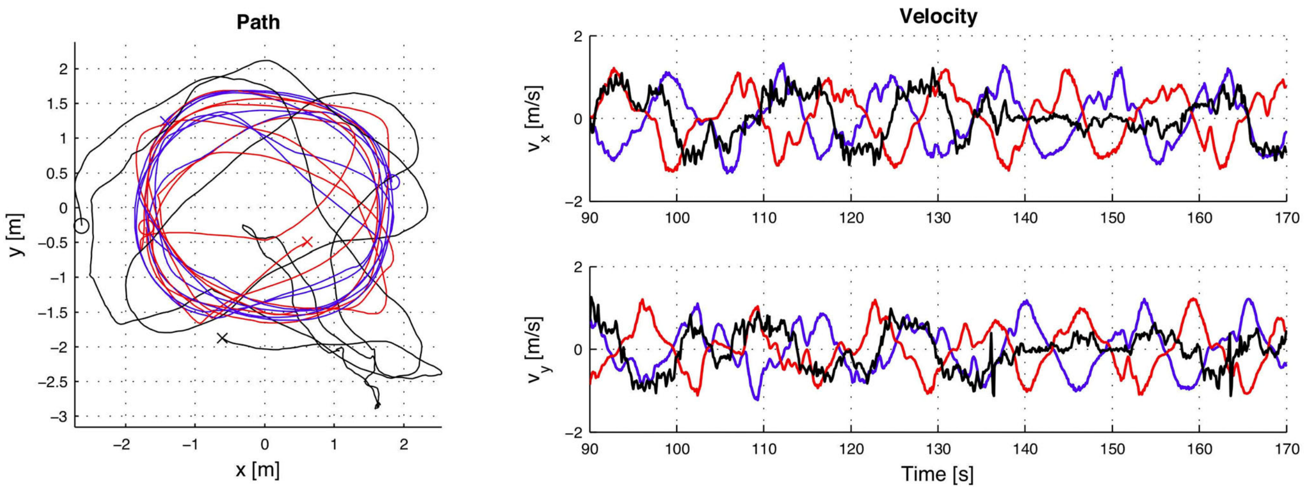


Fig. 17 Two quadrotors (*blue* and *red*) track non-colliding circular trajectories, while a human (*black*) moves randomly in the arena. The two quadrotors avoid collisions locally. *Left Horizontal* (x–y) plot of the path followed by each agent, with initial position marked by "o" and final position by "x". *Right Horizontal* components of the measured velocity **v** with respect to time for all agents. *Vertical* motion is approximately zero (Color figure online)

either due to a stop condition being the optimal solution with respect to the optimization objective, or due to the optimization problem being infeasible. The constraints can be such that the optimization is infeasible (no solution is found) for two main reasons. Firstly, for experimental reasons such as the unmodeled high-order dynamics of the vehicles and other uncalibrated parameters such as delays, leading to differences between the precomputed motion primitives and the executed ones. Secondly, for reasons inherent in the method (and the trade-off between richness of planning versus low computational time), such as the limited horizon of the local planning or the convexification of the optimization via the linearization of the collision avoidance constraint.

The local planning optimization, computed at 10 Hz, was infeasible in under 10 % of the cumulated time-steps over all experiments (about 40,000 data points at 10 Hz over more than 50 experimental runs). On encountering this condition, the speed of the quadrotor is reduced at a higher than normal rate following Eq. (12) and the quadrotor can (if necessary) be brought to a halt before a collision thanks to the finite local planning horizon. In all cases, and thanks to the reduction in

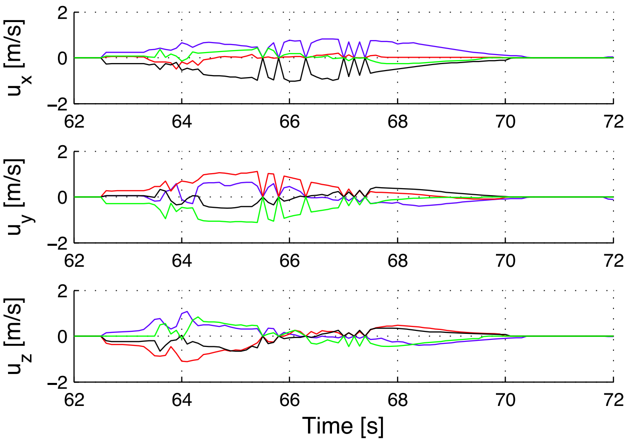


Fig. 18 Optimal reference velocity $\mathbf{u}^*$ for the experiment of Fig. 15, bottom with four quadrotors. There are six timesteps where the optimization becomes infeasible between times 64s and 68s. No collision was observed (see Fig. 15 and the accompanying video)

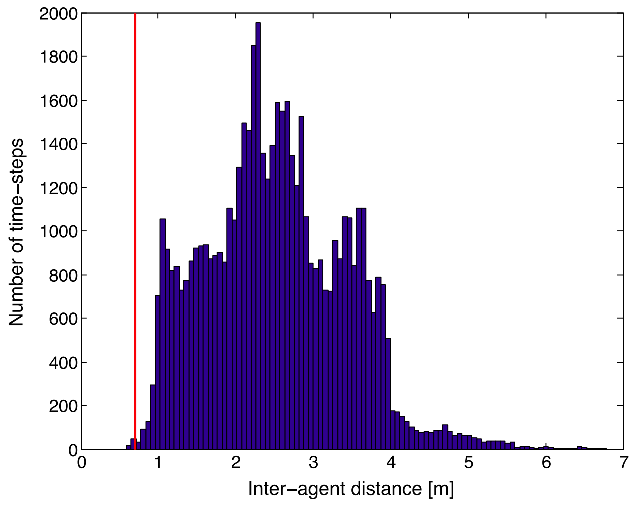


Fig. 19 Histogram of distance between all agents cumulated over all the experiments (about 40,000 data points sampled at 10 Hz). Only a few samples are below the sum of radii (0.7 m)

speed, the optimization became feasible after a further one or more time-steps, typically below 0.5 seconds.[14]

Figure 18 shows the optimal reference velocity $\mathbf{u}^*$ resulting from the optimization with respect to time for one of the experiments with four quadrotors (Fig. 15, bottom). Six instances in which the optimization is infeasible are observed between the times 65 and 68 seconds. These are points where the optimal reference velocity becomes zero before the quadrotors have converged to their goal positions. The effect of these infeasible time-steps is to slow down the quadrotors, which in turn has the effect of rendering the local planning feasible in subsequent time-steps. This extreme situation was mostly not observed in experiments with only two quadrotors.

Figure 19 shows accumulated statistics for the inter-agent distance over all experiments. There is a step decrease in the number of data points with distance below one meter. This is due to the fact that quadrotors are considered to have a radius of 0.35 m, enlarged by $\varepsilon = 0.1$ m to account for the dynamics, as discussed in Sect. 4, and which can decrease towards zero when in close proximity.

A few data-points appear to be in collision (<0.7 m), which can happen in cases where the optimization becomes infeasible, especially when a human is walking in the workspace. In the fully controlled case, no collision was visually observed and all situations were successfully resolved. In the case with a human present, the quadrotor halted before a collision in a few cases - this happens for example when the velocity estimation is inaccurate or the human moves too fast or with an abrupt change in direction.

Algorithm evaluation: Good performance is observed especially for the centralized convex optimization. The distributed optimization performs well in simulation and in experiments with two quadrotors. If the collision avoidance constraints are linearized with respect to the current velocity,[15] the distributed approach suffers from inaccuracies in the velocity estimation, which can lead to performance degradation due to disagreements in the linearization.[16] The centralized, non-convex optimization performs well in simulation but was too slow for real-time performance.

9 Conclusion

Quadrotors are increasing in autonomy, acquiring more sophisticated onboard sensing, and being deployed in groups and around humans. These developments motivate the work here on real-time reactive local motion planning for a set of decision-making agents moving in 3D space. Building on the concept of VO, three approaches are described and compared—a centralized convex optimization, a distributed convex optimization, and a centralized non-convex optimization. The methods can be applied to a group of agents which is heterogeneous in size, dynamics and aggressiveness. Successful performance was shown in extensive experiments with up to four quadrotors in close proximity, and including humans.

The algorithms have low computational complexity so that local motion planning can be on-board, provided

[14] This might not always be the case, mostly in scenarios with fast dynamic obstacles, and if a collision can not be avoided. In that case, the quadrotor stops to guarantee passive safety.

[15] If they are linearized following a given strategy such as avoid to the right, coordination is always achieved, but the solutions can be suboptimal.

[16] For example both agents try to avoid each other on the same side.

that appropriate sensing capabilities are present. In this work several linearization options for the non-convex constraints have been discussed and experimentally evaluated. A detailed analysis of the best linearization option for each case is nonetheless outside of the scope of this paper and remains as future work. Additional future works also include the modeling of system delays for accurate tracking of the motion primitives, which would considerably improve performance. Last but not least, asynchronous operation should be evaluated. We expect the distributed algorithm to perform well in that case thanks to the fast replanning cycle and the low (in comparison) change in position and velocity.

Appendix 1: Extension to homogenous group of agents

This appendix describes an extension of the local motion planning method of Sect. 4 towards an homogeneous group of agents (having the same control parameters).

With the assumption that all agents have the same control parameters (ω_0, ω_1) for the local trajectories (Sec. 7.4), the ε enlargement of the agents is not required. This is achieved by substituting the VO constraint (Constraint 3 of Sec. 4.5) by a 3D extension of the control obstacle $C^M - CO$ introduced by Rufli et al. (2013). The $C^M - CO$ characterizes the (n-differentiable) control trajectories in collision and is computed by formulating in relative candidate reference velocity space the full trajectories (Eq. (22) for $M = 2$). Linearization of the constraint is still required and is done with respect to the current velocity. The algorithms described in this paper can be applied thereafter.

Relying on the concept of differential flatness for a quadrotor vehicle (Mellinger and Kumar 2011), if $M = 5$ is used, the quadrator would, in theory, be able to perfectly track the control trajectory. In this case the full state (up to the fifth derivative) shall be known for all agents.

Appendix 2: Equations repulsive velocity

For the repulsive velocity field of Fig. 4, left, the repulsive velocity for agent $i \in \mathcal{A}$ is given by

```
**for** $j = 1 : m,\ j \neq i$ **do**
  **if** $p^H_{ij} < r_i + r_j$ **and** $|p^3_{ji}| < (h_i + h_j + D_h)$ **then**
    $\mathring{u}^3_{ij} = V_r \left(\frac{r_i+r_j-p^H_{ij}}{r_i+r_j}\right)\left(\frac{h_i+h_j+D_h-|p^3_j-p^3_i|}{D_h}\right)\frac{p^3_{ji}}{|p^3_{ij}|}$
  **end if**
  **if** $p^H_{ij} < r_i + r_j + D_r$ **and** $|p^3_{ji}| < (h_i + h_j)$ **then**
    $\mathring{\mathbf{u}}^H_{ij} = V_r \left(\frac{h_i+h_j-|p^3_{ij}|}{h_i+h_j}\right)\left(\frac{r_i+r_j+D_r-p^H_{ij}}{D_r}\right)\frac{\mathbf{p}^H_{ji}}{p^H_{ji}}$
  **end if**
**end for**
$\mathring{\mathbf{u}}_i := \sum_{j=1}^{m} \mathring{\mathbf{u}}_{ij};\quad \mathring{\mathbf{u}}_i = min(V_r/||\mathring{\mathbf{u}}_i||, 1)\mathring{\mathbf{u}}_i$
```

where V_r is the maximum repulsive force and D_r, D_h the preferred minimal inter-agent distance in the X–Y plane and in the Z component respectively.

Appendix 3: Equations linearization of VO

Denote $\bar{h}_{ij} = \bar{h}_i + \bar{h}_j$ and $\bar{r}_{ij} = \bar{r}_i + \bar{r}_j$.

The non-convex constraint $\mathbb{R}^3 \setminus \mathcal{VO}^\tau_{ij}$ is linearized to obtain a convex problem. For an approximation with five linear constraints, as in Fig. 7, the linear constraints are given by

```
**if** $p^H_{ij} > \bar{r}_{ij}$ **then**
  $\alpha = \text{atan2}(\mathbf{p}^H_{ij}), \quad \alpha_n = \text{acos}(\bar{r}_{ij}/p^H_{ij})$
  **if** $|p^3_{ij}| > \bar{h}_{ij}$ **then**
    $\alpha_1 = \text{atan}\left(\frac{p^3_{ij}+\bar{h}_{ij}}{p^H_{ij}-\text{sign}(p^3_{ij})\bar{r}_{ij}} - \frac{\pi}{2}\right)$
    $\alpha_2 = \text{atan}\left(\frac{p^3_{ij}-\bar{h}_{ij}}{p^H_{ij}+\text{sign}(p^3_{ij})\bar{r}_{ij}} + \frac{\pi}{2}\right)$
    $\mathbf{O} = [p^H_{ij} - \bar{r}_{ij},\ p^3_{ij} - \text{sign}(p^3_{ij})\bar{h}_{ij}]$
  **else**
    $\alpha_1 = \text{atan}\left(\frac{p^3_{ij}+\text{sign}(p^3_{ij})\bar{h}_{ij}}{p^H_{ij}-\bar{r}_{ij}} - \text{sign}(p^3_{ij})\frac{\pi}{2}\right)$
    $\alpha_2 = \text{atan}\left(\frac{p^3_{ij}-\text{sign}(p^3_{ij})\bar{h}_{ij}}{p^H_{ij}-\bar{r}_{ij}} + \text{sign}(p^3_{ij})\frac{\pi}{2}\right)$
    $\mathbf{O} = [p^H_{ij} - \bar{r}_{ij},\ 0]$
  **end if**
  $\alpha_O = \text{atan2}(\mathbf{O})$
  $\mathbf{n}^1_{ij} = [\cos(\alpha - \alpha_n),\ \sin(\alpha - \alpha_n),\ 0]$
  $\mathbf{n}^2_{ij} = [\cos(\alpha + \alpha_n),\ \sin(\alpha + \alpha_n),\ 0]$
  $\mathbf{n}^3_{ij} = [\mathbf{p}^H_{ij}\cos(\alpha_1)/p^H_{ij},\ \sin(\alpha_1)]$
  $\mathbf{n}^4_{ij} = [\mathbf{p}^H_{ij}\cos(\alpha_2)/p^H_{ij},\ \sin(\alpha_2)]$
  $\mathbf{n}^5_{ij} = [\mathbf{p}^H_{ij}\cos(\alpha_O)/p^H_{ij},\ \sin(\alpha_O)]$
  $c^1_{ij} = c^2_{ij} = c^3_{ij} = c^4_{ij} = 0;\ c^5_{ij} = ||\mathbf{O}||/\tau$
**else if** $|p^3_{ij}| > \bar{h}_{ij}$ **then**
  $\alpha = \text{atan2}(\mathbf{p}^H_{ij}), \quad \beta = \text{atan}(\frac{|p^3_{ij}|-\bar{h}_{ij}}{\bar{r}_{ij}})$
  $\alpha_1 = \frac{\pi}{2} - \text{atan}(\frac{|p^3_{ij}|-\bar{h}_{ij}}{\bar{r}_{ij}-p^H_{ij}})$
  $\alpha_2 = \frac{\pi}{2} - \text{atan}(\frac{|p^3_{ij}|-\bar{h}_{ij}}{\bar{r}_{ij}+p^H_{ij}})$
  $\mathbf{n}^1_{ij} = [\cos\alpha\cos\alpha_1,\ \sin\alpha\cos\alpha_1,\ \sin\alpha_1]$
  $\mathbf{n}^2_{ij} = [\cos\alpha\cos\alpha_2,\ \sin\alpha\cos\alpha_2,\ \sin\alpha_2]$
  $\mathbf{n}^3_{ij} = [\cos(\alpha + \frac{\pi}{2})\sin\beta,\ \sin(\alpha + \frac{\pi}{2})\sin\beta,\ \cos\beta]$
  $\mathbf{n}^4_{ij} = [-\cos(\alpha + \frac{\pi}{2})\sin\beta,\ -\sin(\alpha + \frac{\pi}{2})\sin\beta,\ \cos\beta]$
```

```
    $\mathbf{n}_{ij}^5 = [0,\ 0,\ 1]$
    $c_{ij}^1 = c_{ij}^2 = c_{ij}^3 = c_{ij}^4 = 0;\ c_{ij}^5 = (|p_{ij}^3| - \bar{h}_{ij})/\tau$
    **if** $p_{ij}^3 < 0$ **then**
      $\mathbf{n}_{ij}^s[3] = -\mathbf{n}_{ij}^s[3] \quad \forall s$
    **end if**
  **else**
    Agents $i$ and $j$ are in collision
    $\mathbf{n}_{ij}^l = \mathbf{p}_{ij}/p_{ij}$
    $c_{ij}^l = -(\bar{r}_{ij} - p_{ij}^H + \bar{h}_{ij} - |p_j^3 - p_i^3|/\tau$
  **end if**
```

where H_{ij}^1 and H_{ij}^2 represent avoidance to the right / left, H_{ij}^3 and H_{ij}^4 above / below and H_{ij}^5 represents a head-on maneuver, which remains collision-free up to $t = \tau$.

References

Alonso-Mora, J., Breitenmoser, A., Beardsley, P., & Siegwart, R. (2012a). Reciprocal collision avoidance for multiple car-like robots. In *2012 IEEE International conference on robotics and automation (ICRA)* (pp. 360–366).

Alonso-Mora, J., Schoch, M., Breitenmoser, A., Siegwart, R., & Beardsley, P. (2012b). Object and animation display with multiple aerial vehicles. In *2012 IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1078–1083).

Alonso-Mora, J., Rufli, M., Siegwart, R., & Beardsley, P. (2013). Collision avoidance for multiple agents with joint utility maximization. *ICRA*, *2013*, 1–6.

Augugliaro, F., Schoellig, A. P., & D'Andrea, R. (2012). Generation of collision-free trajectories for a quadrocopter fleet: A sequential convex programming approach. In *IEEE/RSJ international conference on intelligent robots and systems* (pp. 1–6).

Bareiss, D., & van den Berg, J. (2013). Reciprocal collision avoidance for robots with linear dynamics using LQR-obstacles. In *IEEE international conference robotics and automation*.

Fiorini, P., & Shillert, Z. (1998). Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, *17*(7), 760–772.

Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, *4*(1), 23–33.

Frazzoli, E., Dahleh, M. A., & Feron, E. (2002). Real-time motion planning for agile autonomous vehicles. *Journal of Guidance, Control, and Dynamics*, *25*(1), 116–129.

Guy, S. J., Chhugani, J., Curtis, S., Dubey, P., Lin, M., & Manocha, D. (2010). Pledestrians: A least-effort approach to crowd simulation. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics symposium on computer animation* (pp. 119–128).

Hoffmann, G. M., & Tomlin, C. J. (2008). Decentralized cooperative collision avoidance for acceleration constrained vehicles. In *47th IEEE conference on decision and control (CDC)* (pp. 4357–4363).

Hoffmann, G. M., Waslander, S. L., & Tomlin, C. J. (2008). Quadrotor helicopter trajectory tracking control. In *AIAA guidance, navigation and control conference and exhibit* (pp. 1–14).

Knepper, R. A., & Mason, M. T. (2012). Real-time informed path sampling for motion planning search. *The International Journal of Robotics Research*, *31*, 1231–1250.

Kumar, V., & Michael, N. (2012). Opportunities and challenges with autonomous micro aerial vehicles. *The International Journal of Robotics Research*, *31*, 1279–1291.

Kushleyev, A., Mellinger, D., & Kumar, V. (2012). Towards a swarm of agile micro quadrotors. In *Robotics: Science and systems (RSS)*.

Kuwata, Y., & How, J. P. (2007). Robust cooperative decentralized trajectory optimization using receding horizon MILP. In *Proceedings of the 2007 American control conference* (pp. 11–13).

Lee, T., Leoky, M., & McClamroch, N. H. (2010). Geometric tracking control of a quadrotor UAV on SE (3). In *49th IEEE conference on decision and control (CDC), 2010* (pp. 5420–5425).

Lupashin, S., Schöllig, A., Hehn, M., & D'Andrea, R. (2011). The Flying Machine Arena as of 2010. In: *2011 IEEE international conference on robotics and automation (ICRA)* (pp. 2970–2971).

Mahony, R., Kumar, V., & Corke, P. (2012). Multirotor aerial vehicles: modeling, estimation, and control of quadrotor. *IEEE Robotics & Automation Magazine*, *19*(3), 20–32.

Mcfadyen, A., Corke, P., & Mejias, L. (2012). Rotorcraft collision avoidance using spherical image-based visual servoing and single point features. In: *2012 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, IEEE (pp. 1199–1205).

Mellinger, D., & Kumar, V. (2011). Minimum snap trajectory generation and control for quadrotors. In *2011 IEEE international conference on robotics and automation (ICRA)* (pp. 2520–2525).

Mellinger, D., Kushleyev, A., & Kumar, V. (2012). Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams. In: *2012 IEEE international conference on robotics and automation (ICRA)* (pp. 477–483).

Michael, N., Mellinger, D., Lindsey, Q., & Kumar, V. (2010). The GRASP multiple micro-UAV testbed. *IEEE Robotics & Automation Magazine*, *17*(3), 56–65.

Ogren, P., Fiorelli, E., & Leonard, N. E. (2004). Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment. *IEEE Transactions on Automatic Control*, *49*(8), 1292–1302.

Pivtoraiko, M., & Kelly, A. (2005). Generating near minimal spanning control sets for constrained motion planning in discrete state spaces. In *2005 IEEE/RSJ international conference on intelligent robots and systems, 2005 (IROS 2005)* (pp. 3231–3237).

Raghunathan, A. U., Gopal, V., Subramanian, D., Biegler, L. T., & Samad, T. (2004). Dynamic optimization strategies for three-dimensional conflict resolution of multiple aircraft. *Journal of Guidance, Control, and Dynamics*, *27*(4), 586–594.

Rufli, M., Alonso-Mora, J., & Siegwart, R. (2013). Reciprocal collision avoidance with motion continuity constraints. *IEEE Transactions on Robotics*, *29*, 899–912.

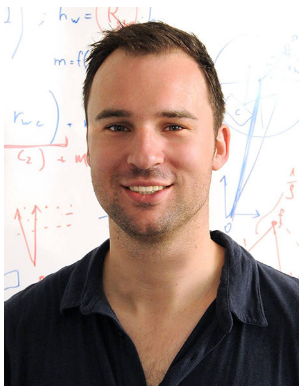
Schwager, M., Julian, B. J., Angermann, M., & Rus, D. (2011). Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, *99*(9), 1541–1561.

Shim, D. H., Kim, H. J., & Sastry, S. (2003). Decentralized nonlinear model predictive control of multiple flying robots. In *Proceedings of the 42nd IEEE conference on decision and control, 2003* (pp. 3621–3626).

van den Berg, J., Guy, S. J., Lin, M., & Manocha, D. (2009). Reciprocal n-body collision avoidance. In *International symposium on robotics research (ISRR)*.

Van Nieuwstadt, M. J., & Murray, R. M. (1997). Real time trajectory generation for differentially flat systems. *International Journal of Robust and Nonlinear Control*, *8*, 995–1020.

Javier Alonso-Mora is currently a Postdoctoral Associate at MIT. He received the M.Sc. and Ph.D. degrees in robotics from ETH Zurich, Zurich, Switzerland in 2010 and 2014 respectively, and the Diploma degree in mathematics in 2008 and in engineering in 2010, both from the Universitat Politecnica de Catalunya (UPC), Barcelona, Spain. During his Ph.D. he was a joint researcher at Disney Research Zurich. His current research focuses on multi-robot systems with two primary goals. First, motion planning algorithms that model inter-agent decision-making. And, second, intuitive methods for controlling and interacting with large teams of mobile robots. Dr. Alonso-Mora received the Willi-Studer Award for the best Master Diploma in robotics at ETH Zurich. He has received excellence scholarships from the UPC, the Spanish Research Council, and the Swiss Government.



Tobias Naegeli received his M.Sc. degree in electrical engineering from ETH Zurich, Switzerland in 2013. He is now a Ph.D. student in the Advanced Interactive Technologies Lab also at ETH Zurich, working on MAV flight in human environments.

Roland Siegwart received the M.Sc. and Ph.D. degrees in mechanical engineering from ETH Zurich, Zurich, Switzerland. He has been a Full Professor for Autonomous Systems with ETH Zurich, Zurich, Switzerland, since 2006 and the Vice President Research and Corporate Relations since 2010. From 1996 to 2006, he was an Associate and later a Full Professor for Autonomous Microsystems and Robotics with the Ecole Polytechnique Federale de Lausanne, Lausanne, Switzerland. He leads a research group of around 30 people working on several aspects of robotics. Dr. Siegwart is a member of the Swiss Academy of Engineering Sciences and the Officer of the International Federation of Robotics Research. He has served as the Vice President for Technical Activities from 2004 to 2005, a Distinguished Lecturer from 2006 to 2007, and an AdCom member from 2007 to 2009 of the IEEE Robotics and Automation Society.

Paul Beardsley leads the Computer Vision Group at Disney Research, Zurich, working on new technologies for Walt Disney Company. He has a Ph.D. from the Robotics Group at the University of Oxford, and experience in industrial research in the US, Japan and Europe. He has been with Disney Research in Zurich since 2008, where he works on mobile robots and vision systems. Recent projects include robotic images for entertainment, and autonomous robots for capturing 3D models of environments.