Multi-robot formation control and object transport in dynamic environments via constrained optimization

Alonso-Mora, Javier; Baker, Stuart; Rus, Daniela

DOI
10.1177/0278364917719333

Publication date
2017

Document Version
Final published version

Published in
The International Journal of Robotics Research

Article

The International Journal of Robotics Research
2017, Vol. 36(9) 1000–1021


Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/0278364917719333
journals.sagepub.com/home/ijr


Multi-robot formation control and object transport in dynamic environments via constrained optimization

Javier Alonso-Mora[1,2], Stuart Baker[1] and Daniela Rus[1]

Abstract

We present a constrained optimization method for multi-robot formation control in dynamic environments, where the robots adjust the parameters of the formation, such as size and three-dimensional orientation, to avoid collisions with static and moving obstacles, and to make progress towards their goal. We describe two variants of the algorithm, one for local *motion planning and one for* global *path planning. The* local *planner first computes a large obstacle-free convex region in a neighborhood of the robots, embedded in position-time space. Then, the parameters of the formation are optimized therein by solving a constrained optimization, via sequential convex programming. The robots navigate towards the optimized formation with individual controllers that account for their dynamics. The idea is extended to* global *path planning by sampling convex regions in free position space and connecting them if a transition in formation is possible - computed via the constrained optimization. The path of lowest cost to the goal is then found via graph search. The method applies to ground and aerial vehicles navigating in two- and three-dimensional environments among static and dynamic obstacles, allows for reconfiguration, and is efficient and scalable with the number of robots. In particular, we consider two applications, a team of aerial vehicles navigating in formation, and a small team of mobile manipulators that collaboratively carry an object. The approach is verified in experiments with a team of three mobile manipulators and in simulations with a team of up to sixteen Micro Air Vehicles (quadrotors).*

Keywords

Multi-robot systems, motion planning, formation control, constrained optimization, sequential convex programming, team of aerial vehicles, micro air vehicles, collaborative mobile manipulators, collaborative object transport

1. Introduction

Multi-robot teams can be employed for various tasks, such as surveillance, inspection, and automated factories. In these scenarios, robots may be required to navigate in formation, for example, to maintain a communication network, to collaboratively manipulate an object, or to survey an area. In this work we consider two motivating applications: formation flight for teams of unmanned aerial vehicles (UAVs) in tight spaces with static and moving obstacles, and collaborative transport of large objects by multiple mobile manipulators in automated factories and working side by side with humans and other robots.

Within the field of multi-robot navigation, formation control and reconfiguration in three-dimensional dynamic environments with moving obstacles remains challenging. In this work, we leverage efficient optimization techniques, namely quadratic programming, semi-definite programming, and (nonlinear) sequential quadratic programming to address this issue. Each one is employed at different stages of the proposed method for formation control among static and dynamic obstacles. These techniques provide good computational efficiency, local guarantees, and generality. Leveraging these tools, we introduce two centralized algorithms - a local motion planner and a global path planner - that enable a team of robots to navigate in formation in two-dimensional and three-dimensional environments with static and dynamic obstacles.

Given a set of target formation shapes, which serve as abstractions, our method optimizes the parameters (such as position, orientation, and size) of the multi-robot formation to avoid moving obstacles and make progress towards the goal. For local motion planning, an obstacle-free region,

[1]Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, USA
[2]Delft Center for Systems and Control, Delft University of Technology, Netherlands

Corresponding author:
Javier Alonso–Mora, Delft University of Technology, Mekelweg 2, 2628 CD Delft, Netherlands.
Email: j.alonsomora@tudelft.nl

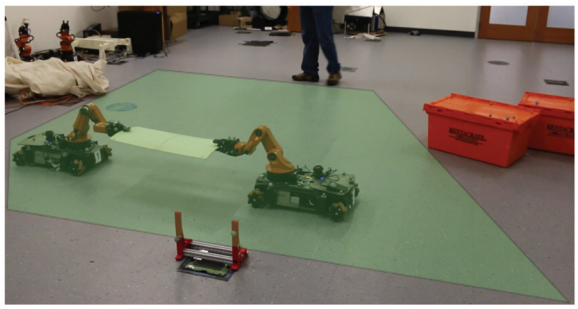


Fig. 1. Two mobile manipulators collaboratively carry a rigid object. A projection of the obstacle-free convex region is superimposed in green.

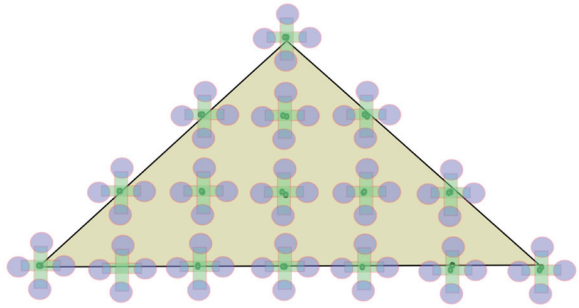


Fig. 2. A triangular formation with sixteen aerial robots can be abstracted by a triangle defined by three vertices. The formation can also be defined in three-dimensional space.

embedded in position-time space, is first grown in a neighborhood of the robots, and then the parameters of the formation are optimized, via a constrained optimization, to remain within this region. The formation optimization method guarantees that the team of robots remains collision-free and makes progress towards the goal. To make global progress towards a goal configuration, we also present a global path planner which builds a graph of feasible formations in the environment. The graph is created by a random sampling of convex regions in free space, which are kept if a valid formation exists within. A human may also provide the global path for the robots, or a desired velocity for the formation, and the robots will adapt their configuration automatically. An example of the method for mobile manipulators is shown in Figure 1. A video illustrating the results of this paper is available at https://youtu.be/sDNqdEPA7pE.

1.1. Contribution

The main contribution of this paper is a scalable and efficient method for navigation of a team of robots while reconfiguring their formation to avoid collisions with static and dynamic obstacles. The method applies to robots navigating in 2D and 3D workspaces and contributes the following.

1. *Locally optimal formation control.* The parameters of the group formation are optimized online within the neighborhood of the robots via a centralized sequential convex optimization with avoidance constraints in dynamic environments.
2. *A global path planner for navigating in formation.* A sampling based graph-search algorithm where convex regions in free space are sampled and connected if the intersections are traversable in formation. Sampling and nonlinear optimization are combined to find a safe global path.

This work provides a working solution to a difficult problem that has not been treated at this scale before. The main strength is its ability to handle dynamic obstacles in three-dimensional environments via constrained optimization, which automatically computes the parameters of the formation to avoid collisions. Furthermore, the formation control method scales well with the number of robots, since its complexity is independent from the number of robots in the team (see Figure 2 for an example of the abstraction of the formation by its outer vertices). Finally, we validate the approach in simulations with teams of aerial vehicles and in extensive experimental demonstrations with three mobile manipulators carrying a rigid object.

In an earlier conference version of this work (Alonso–Mora et al., 2015a), the local motion planner was introduced. In this paper, we describe the approach in detail, we extend it for global path planning, and we present additional experiments with a team of three mobile manipulators collaboratively carrying an object.

The geometrical and optimization ideas of the centralized method of this paper can be combined with consensus for distributed formation control. Recently, we presented an extension (Alonso–Mora et al., 2016) to the case where the robots have a reduced communication and visibility range and share information with their neighbors.

1.2. Related works

In the following we provide an overview of the related literature. In particular, we distinguish between methods for global path planning, which are typically off-line, and online methods for local motion planning and control.

1.2.1. Global path planning. Deadlock-free navigation in complex, yet *static*, environments can be achieved by computing a global path from the initial configuration to the goal configuration and a set of intermediate collision-free configurations for the team of robots. For example, Kushleyev et al. (2012) coined the problem as a mixed-integer quadratic optimization and Saha et al. (2014) relied on discretized linear temporal logic. Both methods provide global guarantees, but scale poorly with the number of robots and do not consider arbitrary formation definitions, instead they rely on squared formations.

An alternative is to randomly sample configurations in state-space to compute a set of safe configurations defining

the path for the team of robots. Barfoot and Clark (2004) computed a global path for the formation via Probabilistic Roadmaps (PRM) by considering a circle enclosing the formation and a leader. Krontiris et al. (2012) later computed a PRM directly for the formation, considering its real shape and a set of templates. Our global path planner is also sampling-based, yet it differs from PRM, or pure sampling-based strategies, in that we compute both feasible formations and traversable areas in free space, which we then use to focus the sampling in unexplored regions of the workspace.

The idea of computing convex regions in free space presents similarities with early work on cell decomposition (Latombe, 1991; LaValle, 2006). One way to decompose the workspace into cells is to triangulate the free space. Conner et al. (2003) and Kallem et al. (2011) used such a triangulation to synthesize controllers for single robot navigation in planar environments, Ayanian et al. (2011) combined a triangulation of the environment with navigation functions to achieve multi-robot control, and Derenick and Spletzer (2007) combined a triangulation of the planar environment with second order cone programming to compute a feasible path for a circular formation. Yet, these methods are limited to planar environments.

We do not compute a typical cell decomposition of the environment, but instead rely on intersections of large convex regions to guarantee collision-free navigation in formation and reconfiguration for the team of robots. Our method builds on the work by Deits and Tedrake (2015), where convex polytopes were used to compute trajectories for single quadrotors. Our approach for global path planning combines sampling-based and constrained optimization techniques to explore the large configuration space. In particular, we sample overlapping convex regions in free position space and rely on a nonlinear constrained optimization to compute the configuration of the robots that can occupy those spaces.

To handle moving obstacles online, we require a local motion planner which utilizes the global path for guidance and incorporates local modifications. In our local motion planner, like the global planner, we employ large convex regions, embedded in position-time space and computed in the neighborhood of the robots. The local planner computes safe motions for the team of robots in three-dimensional dynamic environments within this convex region, and is the main contribution of this work.

1.2.2. Local motion planning. A large part of formation control literature is devoted to the problem of maintaining a formation while respecting the kinematic and dynamic constraints of the robots. Examples of typical approaches for formation control include Lyapunov functions (Ogren et al., 2001), model predictive control (Dunbar and Murray, 2002), flocking (Dimarogonas and Kyriakopoulos, 2005) and leader-follower control (Ren and Sorensen, 2008), each one with its own advantages and disadvantages. For a short review on this topic we refer the reader to Chen and Wang (2005). In our work we do not intend to maintain a specific formation, but instead to adjust its parameters to achieve collision-free navigation in dynamic environments.

Many methods have been proposed for formation control in *obstacle-free* environments. Balch and Arkin (1998) employed a set of reactive behaviors. Other reactive approaches include potential fields (Olfati-Saber and Murray, 2002; Sabattini et al., 2011), and flocking (Dimarogonas and Johansson, 2008; Tanner et al., 2007). Two alternatives to reactive approaches are to use navigation functions (Michael et al., 2008) and to synthesize controllers (Hsieh et al., 2008). Other approaches exist, Desai et al. (2001) combined decentralized feedback laws with graph theory, Zhou and Schwager (2015) considered rigid formations, and Cheah et al. (2009) defined a formation via coverage. Shape stabilization in obstaclefree environments has also been analyzed by Fredslund and Mataric (2002), Fax and Murray (2004), and Cortés (2009).

Several of these approaches were also extended to *planar* environments with static obstacles. For example, social potentials were used by Balch and Hybinette (2000), control of rigid body formations by Egerstedt and Hu (2001), abstractions to enclosing shape by Belta and Kumar (2004) and by Michael and Kumar (2008), local planning in formation space by Kloder and Hutchinson (2006), and controller synthesis by Ayanian et al. (2009). Our method is conceptually similar to Belta and Kumar (2004) in that we also employ an abstraction of the formation, whose dimension is independent of the number of robots. Yet, we do not synthesize controllers, but instead formulate a constrained optimization to compute the parameters of a general formation of arbitrary shape. In contrast to these frameworks, which were limited to planar workspaces, our method achieves collision-free motion and reconfiguration in planar and *three-dimensional* dynamic environments with moving obstacles, and therefore it applies to teams of aerial vehicles.

We formulate the problem as a constrained optimization, which can be solved online via tools for Sequential Convex Programming (SCP). Constrained optimization, and in particular semidefinite programming, was employed by Derenick et al. (2010) for navigating a team of robots in environments with circular obstacles, yet limited to robots moving on the plane. Sequential Convex Programming has been recently employed by Augugliaro et al. (2012) and Chen et al. (2015) to compute collision-free trajectories for multiple UAVs, although they did not consider formation control. Morgan et al. (2016) combined goal assignment with sequential convex programming to optimize the trajectory for a team of robots to reach a target formation, but was limited to obstacle-free environments.

For efficiency, we abstract the robot dynamics when computing the parameters of the formation, but include them in the individual robot controllers. This abstraction is like that of our work on pattern formation for animation display

in obstacle-free environments (Alonso-Mora et al., 2012), where experiments were performed with 50 robots. Since the individual controllers do account for the robot kinematic and dynamic models, our method does apply to non-holonomic robots, as we show in simulations with teams of aerial vehicles.

1.2.3. Cooperative manipulation. Our approach for formation control applies to teams of ground and aerial robots, and it also extends to teams of cooperative manipulators collaboratively carrying an object.

One of the first approaches for collaborative object transport in *obstacle-free* environments was the use of virtual linkages by Khatib et al. (1996). The idea was later extended to decentralized control laws by Sugar and Kumar (2002) and by Tang et al. (2004), which enable a team of robots to accurately maintain a stable grasp in an obstacle-free environment. Static obstacles can be avoided by introducing potential functions that repel the robots from them, as shown by Tanner et al. (2003), but little control is then retained over the resulting configuration. Static and moving obstacles can also be avoided via constrained optimization, as shown by Alonso-Mora et al. (2015b) for the case of deformable objects. In these approaches, it is common to rely on force sensing to coordinate the robots.

In this work, we build on these ideas and present a general, although centralized, non-convex method to compute the parameters of the formation automatically and online via sequential convex programming, which includes both global path planning and local motion planning to avoid static and dynamics obstacles. The method applies to general scenarios, specific formation types, and an arbitrary number of robots. Yet, it enforces that the convex hull of the formation remains collision-free and is therefore best suited for robots carrying convex, or near-convex, objects.

1.3. Method overview

Motion planning for a formation of robots is an instance of planning for a high-dimensional system, which can be solved with sampling-based methods. We expand on this method with a two-step approach.

1. Computes convex obstacle-free regions in position space (global planner) or in position-time space (local planner), embedded in $\mathbb{R}^3$ or $\mathbb{R}^4$.
2. Executes an optimizer to compute the degrees of freedom of the formation, such as its position, size, and orientation, so that the robots remain within the convex regions in free space.

This method combines some of the benefits of sampling-based methods - namely exploring a non-convex workspace and improving the quality of the solution over time - with those of local optimization methods - namely efficiently finding a local optimum in continuous space. We describe two algorithms and one extension following this idea.

In the local motion planner, we rely on the notion of position-time space, where the time dimension is added to the workspace to account for moving obstacles. This is similar to the concept of configuration-time space introduced by Erdmann and Lozano-Perez (1987), but differs in that it is embedded in $\mathbb{R}^4$ instead of in the potentially large high-dimensional space - as would be the case for systems with many degrees of freedom. We make this natural choice explicit with the idea of planning traversable regions in position-time space and letting a non-convex optimizer compute the remaining degrees of freedom of the system to safely navigate within those traversable regions.

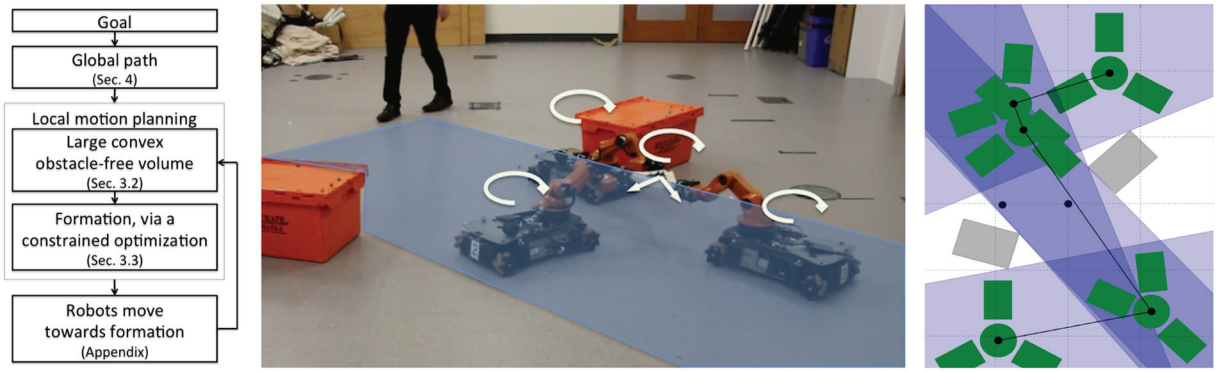
1.3.1. Local motion planning. Given a set of target formation shapes, our method, see Section 3, optimizes the parameters (such as position, orientation, and size) of the multi-robot formation. First, a convex obstacle-free region in position-time space is grown in a neighborhood of the robots. Second, the parameters of the formation are optimized within the convex region by solving a constrained optimization. The method guarantees that the team of robots remains collision-free and makes progress towards the goal. To make global progress towards a goal configuration, only waypoints for the formation center are required. A human may also provide the global path for the robots, or a desired velocity for the formation, and the robots will adapt their configuration automatically.

When individual robots navigate in formation, each robot independently progresses towards its assigned position in the optimal formation via a low-level planner. We employ the distributed convex optimization in velocity space by Alonso-Mora et al. (2015c), which avoids collisions and respects the dynamic constraints of the robot.

In Figure 3(a) we provide an overview of the method. In Figure 1 and in Figure 3(b) we show two examples of the method for mobile manipulators collaboratively carrying an object.

1.3.2. Global path planning. We also describe a method for global path planning from an initial configuration to a final configuration. The method, see Section 4, computes for the team of robots a path, and a set of safe intermediate formations. The robots avoid static obstacles and reconfigure their formation as required.

Our method presents similarities with sampling-based methods such as the Rapidly-Exploring Random Trees RRT approach by LaValle and Kuffner (2001). There, sampling was performed in configuration space and samples (i.e. configurations) and transitions were collision-checked with respect to obstacles. Here we describe an alternative, where sampling is performed in the low dimensional workspace, transitions between formations are guaranteed via convex polytopes, and safe configurations of the formation are obtained via a constrained optimization. The method introduced in this work explores the non-convex workspace and improves the quality of the solution over time thanks to



(a) Overview of the method (b) Local planner with degrees of freedom (c) Global path

Fig. 3. (a) Schematic overview of the method. Given a goal location for the team of robots, we first compute a global path from the start to the goal location, see Section 4. Then, the robots navigate along this path with continuous replanning via a local motion planner, which is described in Section 3. (b) Example with three mobile manipulators collaboratively carrying an object, see Section 5 for the extension of the method to cooperative object transport. In this case, the robots can rotate around their grasping point and a projection of the obstacle-free convex region is shown in blue. (c) Global path to navigate from the formation on the bottom left to the formation on the top right, see Section 4. Obstacles are shown in gray and the robots' formation in green. Obstacle-free convex regions (blue) connect the start with the goal configuration. Two regions are grown from the start and goal positions and the two intermediate regions are grown from random samples in the workspace (black dots). An optimized formation, in green, was computed for each of the two intersections between adjacent regions. The resulting path (solid black line) connects the start and the goal configurations and traverses through the convex regions.

sampling, while remaining efficient due to the constrained optimization and the use of convex regions, which provide a dimensionality reduction.

We create a graph of feasible formations, which connects the initial with the goal configuration. Each node in the graph is a valid configuration, which corresponds to a feasible formation embedded in free-space. Each edge between two configurations is associated with an obstacle-free convex region embedded in the workspace $\bar{\mathcal{F}}$. The graph is created by random sampling of positions in the workspace $\bar{\mathcal{F}}$ from which obstacle-free convex regions are grown. The parameters of valid formations within intersections of polytopes are computed via an efficient constrained optimization.

In Figure 3(c) we show an example of the method where three mobile manipulators carry an object from a start configuration (bottom left) to a goal configuration (top right). We display the first feasible path found by the algorithm, together with the sampled convex regions and the intermediate formations within the intersections.

1.3.3. Generality. We will first describe, in Sections 3 and 4, the method for a team of mobile robots navigating in a formation that can change shape via isomorphic transformations. We will then, in Section 5, describe an alternative formation definition for mobile manipulators collaboratively carrying objects. The method is general and can be adapted to other high-dimensional problems or formation definitions. The core idea is to generate convex, obstacle-free regions and then optimize the parameters of the formation (i.e. the degrees of freedom of the high-dimensional configuration) such that the robots are fully contained in the convex region. The only requirements to adapt the method are (a) a function $\mathcal{V}(\mathbf{z})$ that converts configurations $\mathbf{z}$ to the outer vertices $\mathbf{v}$ of the formation, and (b) a way to compute derivatives of the position of those outer vertices with respect to the configuration $\mathbf{z}$ (unless they are computed numerically).

In this paper, we describe two applications.

1. Formation control: The configuration of the robot team is given by the 3D position, size, and 3D orientation of the formation, i.e. $\mathbf{z} \in \mathbb{R}^3 \times \mathbb{R}_+ \times SO(3)$. Given a template formation, such as a square, the outer vertices are computed via an isomorphic transformation. This is our running example.
2. Collaborative transportation with mobile manipulators: The configuration of the robot team is given by the 2D position and orientation of the object that the robots carry, the orientation of the n robots around their grasping points and the length of their arms, i.e. $\mathbf{z} \in \mathbb{R}^2 \times SO(2)^{n+1} \times \mathbb{R}^n$. The outer vertices of the robots and object can be computed with their shapes and a set of rigid body transformations defined by the configuration. This is described as an extension of the method in Section 5.

An advantage of the method is that planning is decoupled into: (a) finding convex regions in the lower-dimensional free position-time space ($\mathbb{R}^4$) and (b) efficiently optimizing the configuration of the team of robots within those convex regions. This comes at the expense of completeness,

since in our approach we require that the robot team maintains a formation that does not intersect with obstacles, i.e. the robots can not maintain a formation while letting an obstacle pass through. In the event of dynamic obstacles, the team may break the formation to let a moving obstacle pass through, and come back to the original formation as soon as there is enough free room.

1.3.4. Organization. In Section 2 we introduce the notation and we describe the formation definition and the method to compute convex regions. The algorithm for local motion planning is detailed in Section 3, followed by the global path planner in Section 4. In Section 5 we introduce an extension of the method for transportation of an object by multiple manipulators. Section 6 presents experimental results with mobile manipulators and simulations with aerial vehicles. Finally, Section 7 provides a discussion of the method and Section 8 concludes this paper.

2. Preliminaries

In Table 1 we provide a list of the main symbols and variables employed in this paper.

2.1. Robots

Consider a team of robots navigating in formation. For each robot $i \in \mathcal{I} = \{1, \ldots, n\} \subset \mathbb{N}$, its position at time t is denoted by $\mathbf{p}_i(t) \in \mathbb{R}^3$. In the next two sections, we consider all robots to have the same dynamic model and cylindrical non-rotating shape of radius r and height $2h$ in the vertical dimension. Denote the volume occupied by a robot at position $\mathbf{p}$ by $\mathcal{A}(\mathbf{p}) \subset \mathbb{R}^3$.

For an alternative description of the robots, where cylindrical shape is not required, refer to the extension for rectangular mobile manipulators in Section 5.

2.2. Obstacles

Consider a set of static obstacles $\mathcal{O} \subset \mathbb{R}^3$ defining the global map. Denote by $\bar{\mathcal{O}}$ the set $\mathcal{O}$ dilated by half of the robot's volume, formally

$$\bar{\mathcal{O}} = \{\mathbf{p} \in \mathbb{R}^3 \quad | \quad \mathcal{A}(\mathbf{p}) \cap \mathcal{O} \neq \emptyset\} \tag{1}$$

Moving obstacles within the field of view of the robots can be accounted for. Denote by $\mathcal{I}_D = \{1, \ldots, n_d\} \subset \mathbb{N}$ the list of moving obstacles. For moving obstacle $j \in \mathcal{I}_D$ and time t, we denote by $\mathcal{D}_j(t) \subset \mathbb{R}^3$ the volume that it occupies, and

$$\bar{\mathcal{D}}_j(t) = \{\mathbf{p} \in \mathbb{R}^3 \mid \mathcal{A}(\mathbf{p}) \cap \mathcal{D}_j(t) \neq \emptyset\} \tag{2}$$

its dilation by half of robot's volume. For predicted future positions we employ the constant velocity assumption.

2.3. Obstacle-free workspace

The obstacle-free workspace, accounting only for static obstacles, is denoted by

$$\bar{\mathcal{F}} = \mathbb{R}^3 \setminus \bar{\mathcal{O}} \subset \mathbb{R}^3 \tag{3}$$

For current time t_o, and time horizon τ of the motion planner, denote the union of static and dynamic obstacles seen by the robots by

$$\bar{\mathcal{O}}^\tau(t_o) = \bar{\mathcal{O}} \times [0, \tau] \cup \bigcup_{\substack{t \in [0,\tau] \\ j \in \mathcal{I}_D}} \bar{\mathcal{D}}_j(t_o + t) \times t \subset \mathbb{R}^4 \tag{4}$$

where $\times$ denotes the Cartesian product of two sets, and with a slight abuse of notation we denote by variable $t \in \mathbb{R}$ the set $\{t\}$ containing a single point.

The position-time obstacle-free workspace is then

$$\bar{\mathcal{F}}^\tau(t_o) = \mathbb{R}^3 \times [0, \tau] \setminus \bar{\mathcal{O}}^\tau(t_o) \subset \mathbb{R}^4 \tag{5}$$

2.4. Directed obstacle-free convex region

The first building block of the proposed algorithm is to, given an obstacle map and a starting point, compute an obstacle-free convex polytope. We employ a fast iterative method, by Deits and Tedrake (2014), to compute large convex polytopes in free position space, i.e. $\mathcal{P}(\bar{\mathcal{F}}) \subset \bar{\mathcal{F}}$, or in free position-time space, i.e. $\mathcal{P}(\bar{\mathcal{F}}^\tau(t_o)) \subset \bar{\mathcal{F}}^\tau(t_o)$. With an abuse of notation, we may refer to this polytope by $\mathcal{P}$, and recall that for local motion planning (Section 3) we embed it in position-time space and for global path planning (Section 4) we embed it in position space. The method consists of two recurrent steps: (a) it computes the separating hyperplanes between an ellipsoid E and the obstacles $\bar{\mathcal{O}}$ via a quadratic optimization; and (b) it computes, via a semi-definite program, the largest ellipsoid E contained within the convex polytope $\mathcal{P}$. This polytope $\mathcal{P}$ can be described by the union of hyperplanes, see Figure 4 for an example. We extend the method by means of the following.

1. Considering a set of points In, potentially $\{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$, to be contained within $\mathcal{P}$. The iterative algorithm breaks at convergence or when $In \nsubseteq \mathcal{P}$.
2. Growing the region $\mathcal{P}$ towards a desired point $\mathbf{g}_{dir}$. This is achieved by initializing E to be the minimal ellipsoid such that $\{In, \mathbf{g}_{dir}\} \subset E$. The point $\mathbf{g}_{dir}$ is typically set to the goal position for the robot team $\mathbf{g}(t_f)$, and must also be contained within $\mathcal{P}$. If no solution exists, we evaluate alternative points via a linear search between $\mathbf{g}_{dir}$ and the centroid of the points in In.

If the polytope is embedded in position space, we denote by $\mathcal{P}_{In}^{\mathbf{g}_{dir}}(\bar{\mathcal{F}})$ the resulting convex polytope, which contains the points in In and does not intersect any of the obstacles, i.e. satisfies $In \subset \mathcal{P}_{In}^{\mathbf{g}_{dir}}(\bar{\mathcal{F}}) \subset \bar{\mathcal{F}}$ and which is grown in the direction of $\mathbf{g}_{dir}$ as described in the previous paragraph.

Table 1. List of symbols employed in the method.

Symbol	Definition	First appears
n / m	Number of robots / template formations	Section 2
$\mathbf{p}$	Position or point in the workspace (typically in $\mathbb{R}^3$)	Section 2
$\mathcal{A}(\mathbf{p})$	Volume occupied by a robot	Section 2
τ	Time horizon of the local motion planner	Section 2
$\mathcal{O}$ / $\bar{\mathcal{O}}$	Static obstacles / Dilated	Section 2
$\mathcal{D}_j$ / $\bar{\mathcal{D}}_j$	Dynamic obstacle / Dilated	Section 2
$\mathcal{O}^\tau(t_o)$ / $\bar{\mathcal{O}}^\tau(t_o)$	Union of static and dynamic obstacles for time $[t_o, t_o+\tau]$ / Dilated	Equation (4)
$\mathcal{F}$ / $\bar{\mathcal{F}}$	Obstacle-free static workspace (workspace minus static obstacles / Dilated)	Equation (3)
$\mathcal{F}^\tau(t_o)$ / $\bar{\mathcal{F}}^\tau(t_o)$	Obstacle-free workspace in position-time / Dilated	Equation (5)
$\mathcal{P}$	Obstacle-free convex polytope (several variants)	Section 2.4, 2.3
$\mathbf{r}^f_j$ / $\mathbf{r}^f_{0,j}$	Position of robot j in the optimized/template formation f	Section 2.5
$\mathbf{v}^f_j$ / $\mathbf{w}^f_j$	Outer vertex of the optimized/template formation f	Section 2.5
$\mathbf{t}$ / s / $\mathbf{q}$	Position (translation), size and orientation (quaternion) of a formation	Section 2.5
$\mathbf{z}$	Configuration, i.e. optimization parameters of a formation	Section 2.5
$\mathcal{V}(\mathbf{z}, f)$	Set of outer vertices of formation f with configuration $\mathbf{z}$	Equation (8)
$\mathbf{g}$	Goal position for the centroid of the robot team	Section 3
$\bar{s}$ / $\bar{\mathbf{q}}$	Desired size/orientation for the formation	Section 3
$G = \{V, E\}$	Graph containing convex regions and target formations	Section 4

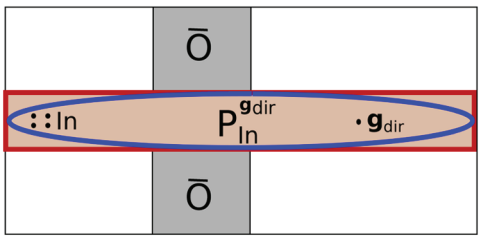
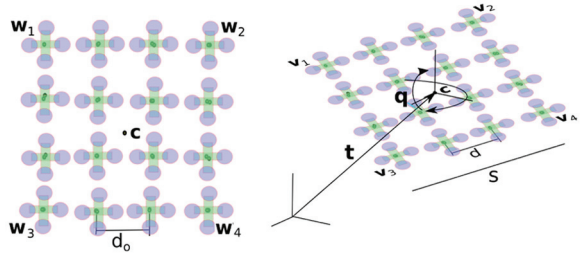


Fig. 4. Example of a convex directed polytope $\mathcal{P}_{In}^{\mathbf{g}_{dir}}(\bar{\mathcal{F}})$ (in red) and its associated ellipsoid (blue) in an environment with two static obstacles (gray). The polytope contains both the *In* points and the target point $\mathbf{g}_{dir}$.



(a) Template square formation (b) Transformed formation in 3D

Fig. 5. (a) Example of a template square formation with sixteen MAVs. The four outer vertexes define the convex hull. (b) The formation can be transformed with a translation $\mathbf{t}$, a 3D rotation $\mathbf{q}$ and a size s isomorphic transformation.

Definition 1 . (Directed polytope)**.** *We refer to a polytope* $\mathcal{P}_{In}^{\mathbf{g}_{dir}}(\bar{\mathcal{F}}) \subset \mathbb{R}^3$*, embedded in* $\bar{\mathcal{F}}$ *as a directed polytope, towards* $\mathbf{g}_{dir}$*. Analogously for a polytope* $\mathcal{P}_{In}^{\mathbf{g}_{dir}}(\bar{\mathcal{F}}^\tau(t_o)) \subset \mathbb{R}^4$*, embedded in* $\bar{\mathcal{F}}^\tau(t_o)$*.*

2.5. Definition of the formation

For an alternative description of the formation, refer to the extension for mobile manipulators in Section 5.

We consider a predefined set of $m \in \mathbb{N}$ template formations, such as square, line, or T. Each template formation $f \in \mathcal{I}_f = \{1, \ldots, m\}$ is given by a set of robot positions $\{\mathbf{r}^f_{0,1}, \ldots, \mathbf{r}^f_{0,n}\}$ and a set of outer vertices $\{\mathbf{w}^f_1, \ldots, \mathbf{w}^f_{n_f}\}$ relative to the center of rotation (typically the centroid) of the formation, where n_f denotes the number of outer vertices defining formation f. The set of vertices represents the convex hull of the robot's positions in the formation, thus reducing the complexity for formations with many robots. See Figure 5(a) for an example.

Further denote by d_f the minimum distance between any given pair of robots in the template formation f. Template formations can be defined by a human designer or automatically computed for optimal representation of a target shape as shown by Alonso–Mora et al. (2012).

A formation is then defined by an isomorphic transformation, which includes the size $s \in \mathbb{R}_+$, a translation $\mathbf{t} \in \mathbb{R}^3$, and a rotation $R(\mathbf{q})$ described by a unit quaternion $\mathbf{q} \in SO(3)$, its conjugate denoted by $\bar{\mathbf{q}}$. With this formation definition, the configuration for the team of robots is fully defined by $\mathbf{z} = [\mathbf{t}, s, \mathbf{q}] \in \mathbb{R}^3 \times \mathbb{R}_+ \times SO(3)$.

Given the configuration $\mathbf{z}$, and template formation ID f, the robot positions and outer vertices of the resulting formation are computed by

$$\begin{aligned} \mathbf{r}^f_i &= \mathbf{t} + s\, R(\mathbf{q})\, \mathbf{r}^f_{0,i}, \quad \forall i \in [1, n] \\ \mathbf{v}^f_j &= \mathbf{t} + s\, R(\mathbf{q})\, \mathbf{w}^f_j, \quad \forall j \in [1, n_f] \end{aligned} \tag{6}$$

where the rotation in $SO(3)$ is given by the quaternion operation

$$\left[0, \quad R(\mathbf{q})\,\mathbf{w}_j^f\right]^T = \mathbf{q} \times \left[0, \quad \mathbf{w}_j^f\right]^T \times \bar{\mathbf{q}} \tag{7}$$

For template formation f and configuration $\mathbf{z}$ we denote the set of outer vertices by

$$\mathcal{V}(\mathbf{z}, f) = [\mathbf{v}_1^f, \ldots, \mathbf{v}_{n_f}^f] \tag{8}$$

In the exposition of the method we rely on this definition for the formation, but the method is general and can be applied to alternative definitions, as shown in Section 5 for the case of several manipulators transporting a rigid object.

3. Local motion planning

The local motion planner computes the optimal parameters, i.e. the configuration, of the formation, in a neighborhood of the robots, via a constrained nonlinear optimization. For a given template formation $f \leq m$, the vector of optimization variables, i.e. the configuration, is denoted by $\mathbf{z} = [\mathbf{t}, s, \mathbf{q}] \in \mathbb{R}^3 \times \mathbb{R}_+ \times SO(3)$.

Denote by $\mathbf{g}(t) \in \mathbb{R}^3$ the goal position for the centroid of the formation at time t. This goal position, and a target orientation $\bar{\mathbf{q}}$ and size $\bar{s}$, can be given by a human operator or a global planner, as described in the forthcoming Section 4.

For an alternative formulation of the optimization, see the extension for mobile manipulators in Section 5.

3.1. Algorithm overview

To make progress towards the goal position while avoiding obstacles, the local planner computes a target formation and the required motion of the robots for a given time horizon $\tau > 0$, which must be longer than the required time to stop. Denote the current time by t_o and $t_1 = t_o + \tau$.

Our method consists of the following steps.

1. Compute a large convex polytope $\mathcal{P}$ contained in free position-time space, such that the robots are inside it, i.e. $\mathbf{p}_i(t_o) \in \mathcal{P} \subset \bar{\mathcal{F}}^\tau(t_o)$, $\forall i \in \mathcal{I}$, and that is directed towards the goal $\mathbf{g}(t_1)$. This is described in Section 3.2.
2. Compute the optimal formation f^* and configuration $\mathbf{z}^*$ such that the outer vertices $\mathcal{V}(\mathbf{z}^*, f^*)$ are contained within $\mathcal{P}$ and the distance between the formation's centroid and the goal $\mathbf{g}(t_1)$ is minimized. The parameters of the formation are optimized subject to a set of constraints via a centralized sequential convex optimization described in Section 3.3. In this computation, the robot's dynamics are abstracted.
3. In a faster loop, described in Section 3.5, the robots are optimally assigned to target positions in the formation and move towards them employing a low level local planner that generates collision-free inputs that respect the robot's dynamics. In particular, we build on a distributed convex optimization described by Alonso–Mora et al. (2015c), extended to account for static obstacles in a seamless way.
4. If no feasible formation exists in a neighborhood of the robots, we search for the parameters of a target formation near the goal position. In this case, the robot team splits and each robot navigates independently towards its assigned position in the target formation.

Algorithm 1 Local motion planning.

Given: Union of static and dynamic obstacles in position-time space $\bar{\mathcal{O}}^\tau(t_o) \subset \mathbb{R}^4$ at the initial time. The goal position $\mathbf{g}(t_1) \in \mathbb{R}^3$ for the centroid of the formation at time t_1 and desired size $\bar{s}$ and orientation $\bar{\mathbf{q}}$.

Compute: Target configuration $\mathbf{z}^*$ and formation f^*. Collision-free motion for the team of robots for up to the time horizon τ.

——————- Main process ——————

1: **while not** converged **do**
2: Compute large convex polytope $\mathcal{P} \subset \bar{\mathcal{F}}^\tau(t_o)$ in a neighborhood of the robots.
3: Compute optimal configuration $\mathbf{z}^*$ and formation f^*, such that the outer vertices $\mathcal{V}(\mathbf{z}^*, f^*)$ are contained within $\mathcal{P}$.
4: **end while**

——- Second parallel process, at high frequency ———

5: **while not** converged **do**
6: Assign robots to target positions in the formation.
7: Navigate towards the target formation.
8: **end while**

3.2. Obstacle-free convex region

First, the obstacle-free space in position-time $\mathbb{R}^3 \times [0, \tau] \subset \mathbb{R}^4$ is obtained, accounting for static and dynamic obstacles.

For the given time horizon τ consider the union of static and dynamic obstacles $\bar{\mathcal{O}}^\tau(t_o)$ and the associated position-time obstacle-free workspace $\bar{\mathcal{F}}^\tau(t_o)$, as described in equation (5).

Following Section 2.4, we compute two convex polytopes.

1. $\mathcal{P}_{fo \to g}$ in free position-time space, which contains all the robots at their current positions and initial time, i.e. $[\mathbf{p}_1(t_o), \ldots, \mathbf{p}_n(t_o)] \times 0$, and which is directed towards the formation's goal at the time horizon, i.e. $[\mathbf{g}(t_1) \times \tau] \in \mathbb{R}^4$. Formally

$$\mathcal{P}_{fo \to g} := \mathcal{P}^{[\mathbf{g}(t_1) \times \tau]}_{[\mathbf{p}_1(t_o), \ldots, \mathbf{p}_n(t_o)] \times 0}(\bar{\mathcal{F}}^\tau(t_o)) \tag{9}$$

2. $\mathcal{P}_{o \to g}$ in free position-time space, which contains the centroid of the robots' current positions and initial time,

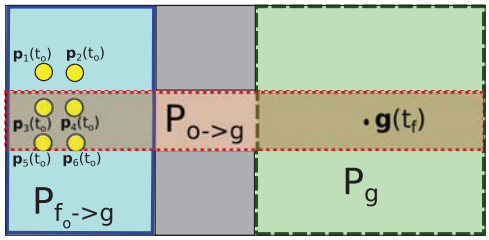


Fig. 6. Example of convex regions computed with the method described in Section 2.4 for a scenario with six yellow disk robots and two (gray) squared obstacles. The convex regions are: $\mathcal{P}_{f_o \to g}$ in blue solid border, $\mathcal{P}_{o \to g}$ in red dotted border and $\mathcal{P}_g$ (defined in Section 3.4) in green dashed border.

i.e. $[\sum_{i\in\mathcal{I}} \mathbf{p}_i(t_o)/n] \times 0 \in \mathbb{R}^4$, and which is directed towards the goal. Formally

$$\mathcal{P}_{o\to g} := \mathcal{P}^{[\mathbf{g}(t_1)\times\tau]}_{[\sum_{i\in\mathcal{I}} \mathbf{p}_i(t_o)/n]\times 0}(\bar{\mathcal{F}}^{\tau}(t_o)) \tag{10}$$

This polytope may not contain all the robots at their current positions.

A representative example of these regions (projected in $\mathbb{R}^2$) is shown in Figure 6. In general, we consider the convex polytope

$$\mathcal{P} = \mathcal{P}_{f_o\to g} \cap \mathcal{P}_{o\to g} \tag{11}$$

which:

- guarantees that the transition to the new formation will be collision-free, since $\mathcal{P} \subset \mathcal{P}_{f_o\to g}$ and all the robots are within the convex region $\mathcal{P}_{f_o\to g}$;
- is likely to make progress in future iterations, since $\mathcal{P} \subset \mathcal{P}_{o\to g}$, which is directed towards the goal.

Once the robots are within this intersection they can make progress towards the goal within $\mathcal{P}_{o\to g}$ in a collision-free manner. If $\mathcal{P} = \emptyset$, an alternative convex region is selected as described in the forthcoming Section 3.4.

We rely on a representation of the collision-free convex polytope $\mathcal{P}$ given by its equivalent set of linear constraints

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^4 \mid A\mathbf{x} \le \mathbf{b}, \text{ for } A \in \mathbb{R}^{n_l\times 4}, \mathbf{b} \in \mathbb{R}^{n_l}\} \tag{12}$$

where n_l denotes the number of faces of $\mathcal{P}$.

3.3. Nonlinear optimization

We formulate a constrained nonlinear optimization to compute a locally optimal formation f^* and the configuration $\mathbf{z}^*$ for the team of robots.

3.3.1. Optimization cost. We minimize a weighted sum of the deviation with respect to the formation's goal $\mathbf{g}(t_1)$, a desired size $\bar{s}$ and a desired rotation $\bar{\mathbf{q}}$. The cost term is then

$$J_f(\mathbf{z}) = w_t||\mathbf{t}-\mathbf{g}(t_1)||^2 + w_s||s-\bar{s}||^2 + w_q||\mathbf{q}-\bar{\mathbf{q}}||^2 + c_f \tag{13}$$

where w_t, w_s, w_q are design weights and c_f is the predefined cost for formation type $f \in \mathcal{I}_f$.

3.3.2. Constraints. Constraints are introduced to guarantee that all the robots in the formation are within the convex polytope (C_1) and to satisfy a minimum inter-robot distance (C_2) to avoid collisions between the robots in the team. Recalling Section 2.5 the constraints are then given by

$$\begin{aligned} C_1 &\equiv \mathcal{V}(\mathbf{z},f) \times t_1 \subset \mathcal{P} \equiv \\ &\bigcup_{j=1}^{n_f} \{A[[\mathbf{t} + s\,R(\mathbf{q})\,\mathbf{w}_j^f] \times t_1]^T \le \mathbf{b}\} \\ C_2 &\equiv \left\{ s \ge 2\frac{\max(r,h)}{d_f} \right\} \end{aligned} \tag{14}$$

where C_1 contains a constraint for each vertex $\mathbf{w}_j^f$ of the convex hull of the template formation f and implies that the robots do not collide with any obstacle. The constraint C_2 guarantees inter-robot collision avoidance since d_f is the minimum inter-robot distance for the template formation, recall Section 2.5, and the transformation applied to the formation is isomorphic.[1]

For planar formations, the additional constraints $\mathbf{q} \cdot [0,1,0,0]^T = 0$ and $\mathbf{q} \cdot [0,0,1,0]^T = 0$ may also be imposed to ensure rotation only occurs around the vertical axis.

3.3.3. Nonlinear program. For a template formation $f \in \mathcal{I}_f$ the optimal configuration $\mathbf{z}_f^*$ is found by solving the nonlinear optimization

$$\begin{aligned} \mathbf{z}_f^* = \quad & \arg\min_{\mathbf{z}} J_f(\mathbf{z}) \\ s.t. \quad & \mathcal{V}(\mathbf{z},f) \times t_1 \subset \mathcal{P} \qquad (C_1) \\ & \left\{ s \ge 2\frac{\max(r,h)}{d_f} \right\} \qquad (C_2) \end{aligned} \tag{15}$$

We employ the nonlinear solver SNOPT by Gill et al. (2002), which internally executes a sparse Sequential Quadratic Program and converges to a feasible local minimum of the cost function.

The derivatives of the cost function (equation (13)) and constraints (equation (14)) are given by

$$\partial J_f(\mathbf{z})/\partial \mathbf{z} \equiv 2[w_t(\mathbf{t}-\mathbf{g}(t_1)),\, w_s(s-\bar{s}),\, w_q(\mathbf{q}-\bar{\mathbf{q}})]. \tag{16}$$

$$\begin{aligned} &\partial C_1/\partial \mathbf{z} \equiv \textstyle\bigcup_{j=1}^{n_f} [A,\, A\,R(\mathbf{q})\,\mathbf{w}_j^f,\, s\,A\,\partial R(\mathbf{q})\,\mathbf{w}_j^f/\partial \mathbf{q})] \\ &\partial C_2/\partial s \equiv d_f \end{aligned} \tag{17}$$

where $\partial R(\mathbf{q})\,\mathbf{w}_j^f/\partial \mathbf{q}$ is the Jacobian of equation (7).[2]

We set the initial point for the optimizer to

$$\mathbf{z}_{ini} = [\mathbf{g}(t_1), 2\max(r,h)/d_f, \mathbf{q}_{ini}] \tag{18}$$

where the initial quaternion is chosen to be equal to the quaternion addition of the desired orientation and a small random quaternion, i.e. $\mathbf{q}_{ini} = \bar{\mathbf{q}} + \mathbf{q}_{rand}$. The additional term is included to avoid singularities of the optimizer when some components of $\bar{\mathbf{q}}$ are zero.

If the constrained optimization of equation (15) is solved for each template formation, the index f^* of the locally optimal formation is then given by

$$f^* = \arg\min_f J_f(\mathbf{z}_f^*) \tag{19}$$

This formation definition and its associated nonlinear optimization are given as an example, but the framework is general and can be applied to other problem instances. In Section 5 we describe how to apply this method for the manipulation of rigid objects.

3.4. Iterations if problem is infeasible

The method, as described in the previous subsections, results in a formation and its configuration for the robots' team. It may occur though, that the robots make no progress towards the goal (deadlock) or that the optimization is infeasible, for example if the region $\mathcal{P}$ defined in equation (11) is too small and no feasible formation fits inside. In that case, one may search for a feasible formation using an alternative region. For a representative example see Figure 6.

First, we would repeat the optimization using the convex region $\mathcal{P}_{f_o \to g}$. If a valid target formation is found in this step, or in the original optimization with polytope $\mathcal{P}$, the transition is guaranteed to be collision-free, thanks to the convexity of the polytope $\mathcal{P}_{f_o \to g}$ which contains the current position of all the robots.

If this additional step is also unfeasible, then no formation may exist such that the robots can continue navigating in formation towards the goal. In that case the optimization can be repeated using the polytope $\mathcal{P}_{o \to g}$ or directly the polytope $\mathcal{P}_g := \mathcal{P}_{[\mathbf{g}(t_1),\tau]}(\bar{\mathcal{F}})$ that contains the formation's goal. Note though that these two polytopes do not contain the current robot positions. If a formation is found, the robots move individually, i.e. separately, towards their respective positions in the target formation. In this case, the formation is likely broken during the transition, but, this gives further flexibility to the method and the robots to navigate in formation whenever possible, via splitting and merging.

3.5. Individual planning towards target formation

The result of the computation of Section 3.3 is a target formation f^* and configuration $\mathbf{z}^*$. The associated set of target robot positions $\mathbf{r}_j$, for all $j \in \mathcal{I}$ can be computed with equation (6).

In this section, we describe the local planner that links the centralized formation optimization with the physical robots. At each step of the execution, at higher frequency than that of computing a new formation, the following steps are executed.

3.5.1. Goal assignment. Robots are optimally assigned to the target positions $\mathbf{r}_j$ with the objective of minimizing the sum of squared traveled distances. The optimal assignment $\sigma : \mathcal{I} \to \mathcal{I}$ is

$$\min_{\sigma} \quad \sum_{i \in \mathcal{I}} ||\mathbf{p}_i - \mathbf{r}_{\sigma(i)}||^2 \tag{20}$$

Following Alonso–Mora et al. (2012), this assignment can be centrally computed with the optimal Hungarian algorithm by Kuhn (1955), used in this work, or a suboptimal auction algorithm by Bertsekas (1988), which scales well with the number of robots.

3.5.2. Collision avoidance. To control the individual robots in the team and to avoid collisions between them, we employ the collision avoidance algorithm introduced by Alonso–Mora et al. (2015c). We employ the same constraints to avoid moving obstacles and to respect the kinematic model of the robots. To better handle environments with static obstacles, we include additional linear constraints defined by a convex polytope in free space, computed in a neighborhood of the robot. For details refer to the Appendix. This method, a convex optimization in velocity space, is well suited for our application. The formation control algorithms described in this paper are agnostic to the low-level controller and a different one could be employed.

4. Global path planning

Given an initial and a target configuration for the robot team, the global path planner computes a feasible path and intermediate formations to connect them. This is achieved by combining a sampling-based approach with constrained nonlinear optimization, the idea being to sample in a low dimensional space (workspace) and letting the optimizer compute the remaining degrees of freedom.

In particular, we create a graph where each node is a feasible formation and which contains the initial and the goal configuration. An edge between two nodes, or formations, is a convex region in free space, which contains both formations. An edge provides the means to transition between two nodes in the graph. An example was shown in Figure 3(c).

The approach can be applied to a single user-defined formation (i.e. square) or when multiple formations are given. In the latter, reconfiguration between formation shape would be allowed. In an abuse of notation, throughout this section we drop the subindex $f \in \mathcal{I}_f$, consider a single formation f and refer to a polytope $\mathcal{P}_{\mathbf{p}}(\bar{\mathcal{F}})$ embedded in the free position space, i.e. $\mathcal{P}(\bar{\mathcal{F}}) \subset \bar{\mathcal{F}}$, by $\mathcal{P}$. This is in contrast to the local planning approach, where we embedded the convex polytope in the free position-time space. Therefore, we do not consider moving obstacles in the global path planning.

4.1. Algorithm

Consider the obstacle-free workspace $\bar{\mathcal{F}}$ defined by equation (3), the start position $\mathbf{s}$ of the formation's centroid and its goal position $\mathbf{g} \in \mathbb{R}^3$. In Algorithm 2 we describe the proposed anytime method to compute a path for the robot team to navigate in formation from $\mathbf{s}$ to $\mathbf{g}$.

A graph $G = \{V, E\}$ is incrementally created. Each vertex in the vertices list V is given by the configuration $\mathbf{z}$

of a feasible formation. Each edge in the edge list E connects two nodes, i.e. valid configurations $\mathbf{z}_1, \mathbf{z}_2$ for the team of robots, if a convex region $\mathcal{P}$ exists such that the robots in both configurations are fully contained in the convex polytope.

We keep a list of existing polytopes P. And, for each polytope $\mathcal{P} \in P$ we keep a list $L_{\mathcal{P}}$ of configurations for which the team of robots is fully contained within the polytope. The node list is initialized with the initial $\mathbf{z}_s$ and final $\mathbf{z}_g$ configurations with centroids $\mathbf{s}$ and $\mathbf{g}$. Analogously, the polytopes list is initialized with the convex regions $\mathcal{P}_\mathbf{s}$ and $\mathcal{P}_\mathbf{g}$, which contain the initial and final configurations respectively.

The method proceeds by drawing random samples in the workspace ($\mathbb{R}^3$ for aerial vehicles). Each random sample $\mathbf{p} \in \mathbb{R}^3$ is rejected if it is inside an obstacle or one of the polytopes in the list P. If $\mathbf{p} \in \bar{\mathcal{F}} \setminus \bigcup_{\mathcal{P}\in P} \mathcal{P}$ then the following steps are executed.

1. A large convex polytope $\mathcal{P}_\mathbf{p}(\bar{\mathcal{F}}) \subset \bar{\mathcal{F}}$ is grown from $\mathbf{p}$ following the method of Section 2.4.
2. For each polytope $\mathcal{P} \in P$ that intersects $\mathcal{P}_\mathbf{p}$, we compute a configuration $\mathbf{z}$ and formation f for the team of robots such that the formation's vertices are fully contained within the intersection of both polytopes $\mathcal{V}(\mathbf{z}, f) \subset \mathcal{P}_\mathbf{p} \cap \mathcal{P}$ and that minimizes the squared distance from its centroid to $\mathbf{g}$. The configuration $\mathbf{z}$ and formation f are computed via a nonlinear optimization analogous to that of Section 3.3. For polytope $\mathcal{P}$ we denote this function by formation($\mathcal{P}$). If a valid configuration exists, it is added to the node list.
3. If a valid configuration $\mathbf{z}$ is added to the node list, then (a) an edge $\{\mathbf{z}, \mathbf{z}_i, \mathcal{P}_\mathbf{p}\}$ is added for all configurations $\mathbf{z}_i \in L_{\mathcal{P}_\mathbf{p}}$ and (b) an edge $\{\mathbf{z}, \mathbf{z}_i, \mathcal{P}\}$ is also added for all configurations $\mathbf{z}_i \in L_{\mathcal{P}}$. Recalling the previous section, it is guaranteed that the team of robots can navigate between both formations through the associated convex polytope.

A feasible solution is found as soon as a path (or sequence of connected vertexes in the graph G) is found which connects the initial position with the goal position of the formation's centroid. If we let the algorithm run longer, for example until most of the free space is covered by convex regions, the best path so far is found via graph search. This is, by computing the path of lowest cost in the graph G. For each edge E between two configurations $\mathbf{z}_1$ and $\mathbf{z}_2$ we define its cost by the distance between the centroids of $\mathbf{z}_1$ and $\mathbf{z}_2$.

Algorithm 2 Global path planning.

Given: obstacle field $\mathcal{O}$, start configuration $\mathbf{z}_s$ with centroid $\mathbf{s}$ and destination $\mathbf{g}$ for the formation's centroid.
Returns: a path $\mathcal{T}$ of feasible configurations (formations) from $\mathbf{s}$ to $\mathbf{g}$.
We describe a bidirectional graph search. The method can be adapted to a tree search.

Initialize empty graph $G = \{V, E\}$: $V = \emptyset$; $E = \emptyset$
Initialize empty polytope list $P = \emptyset$
Add the initial configuration to the node list $V \leftarrow \mathbf{z}_s$
Generate $\mathcal{P}_\mathbf{s}, \mathcal{P}_\mathbf{g} \subset \bar{\mathcal{F}}$ from $\mathbf{s}$ and $\mathbf{g}$
 # Add them to the polytope list
$P \leftarrow \mathcal{P}_\mathbf{s}$, $P \leftarrow \mathcal{P}_\mathbf{g}$
 # Compute a valid configuration in the goal region
$\mathbf{z}_g$ = formation($\mathcal{P}_\mathbf{g}$)
Add the goal configuration to the node list $V \leftarrow \mathbf{z}_g$
 # Create lists of valid configurations for both polytopes
$L_{\mathcal{P}_\mathbf{s}} = \{\mathbf{z}_s\}$, $L_{\mathcal{P}_\mathbf{g}} = \{\mathbf{z}_g\}$
 # Check if the start and goal can be connected
if $\exists \mathbf{z}$ = formation($\mathcal{P}_\mathbf{s} \cap \mathcal{P}_\mathbf{g}$) **then**
    $V \leftarrow \{\mathbf{z}\}$
    $E \leftarrow \{\mathbf{z}_s, \mathbf{z}, \mathcal{P}_\mathbf{s}\}$
    $E \leftarrow \{\mathbf{z}_g, \mathbf{z}, \mathcal{P}_\mathbf{g}\}$
end if
 # The following search loop can be executed until the first feasible path is found, until the whole space $\bar{\mathcal{F}}$ is explored or up to a maximum time bound.
while not end **do**
    Generate random sample $\mathbf{p} \in \bar{\mathcal{F}} \setminus (\cup_{i\in P} \mathcal{P}_v)$
    Generate polytope $\mathcal{P}_\mathbf{p} \subset \bar{\mathcal{F}}$ grown from $\mathbf{p}$
    # Try to create new node
    if $\exists \mathbf{z}$ = formation($\mathcal{P}_\mathbf{p}$) **then**
        $L_{\mathcal{P}_\mathbf{p}} = \emptyset$
        # Try to create new nodes and edges
        for $\mathcal{P} \in P$ **do**
            if $\exists \mathbf{z}_1$ = formation($\mathcal{P} \cap \mathcal{P}_\mathbf{p}$) **then**
                for $\mathbf{z}_i \in L_{\mathcal{P}}$ **do**
                    $E \leftarrow \{\mathbf{z}_1, \mathbf{z}_i, \mathcal{P}\}$
                end for
                for $\mathbf{z}_i \in L_{\mathcal{P}_\mathbf{p}}$ **do**
                    $E \leftarrow \{\mathbf{z}_1, \mathbf{z}_i, \mathcal{P}_\mathbf{p}\}$
                end for
                $V \leftarrow \mathbf{z}_1$
                $L_{\mathcal{P}_\mathbf{p}} \leftarrow \{\mathbf{z}_1\}$; $L_{\mathcal{P}} \leftarrow \{\mathbf{z}_1\}$
            end if
        end for
        $P \leftarrow \mathcal{P}_\mathbf{p}$
    end if
    return $\mathcal{T}$ =shortestPath(G)
end while

4.2. Execution in composition with the local motion planner

To navigate the team of robots from the initial to the goal configuration a global path consisting of a sequence $\mathcal{T} = \{\mathbf{z}_s, \ldots, \mathbf{z}_g\}$ of configurations is first obtained via the global path planning algorithm of the previous section. Each configuration $\mathbf{z}_w \in \mathcal{T}$ in the sequence, provides an intermediate setpoint for the team of robots. Denote its centroid by $\mathbf{w}$.

Algorithm 3 Function: $\mathbf{z}$ = formation($\mathcal{P}$)

Input: Convex polytope $\mathcal{P} \subset \bar{\mathcal{F}}$.
Output: A valid configuration $\mathbf{z}$ such that $\mathcal{V}(\mathbf{z},f) \subset \mathcal{P}$, or $\emptyset$.

1: **if** $\mathcal{P} = \emptyset$ **then**
2: **return** $\emptyset$
3: **else**
4: **return** result of a nonlinear optimization analogous to that of Section 3.3 with convex polytope $\mathcal{P}$ and minimizing $J(\mathbf{z})$ the deviation to a target configuration at $\mathbf{g}$, i.e.

$$\begin{aligned} \mathbf{z}^* = \quad & \arg\min_{\mathbf{z}} J(\mathbf{z}) \\ s.t. \quad & \mathcal{V}(\mathbf{z},f) \subset \mathcal{P} \\ & \left\{ s \geq 2\frac{\max(r,h)}{d_f} \right\} \end{aligned} \tag{21}$$

5: **end if**

Algorithm 4 Function: $\mathcal{T}$ =shortestPath(G)

Input: Graph G.
Output: A sequence of valid configurations $\mathcal{T} = \{\mathbf{z}_s, \ldots, \mathbf{z}_g\}$ and convex polytopes $\{\mathcal{P}_\mathbf{s}, \ldots, \mathcal{P}_\mathbf{g}\}$ such that the robot team can navigate through them from the start to the goal, or $\emptyset$.

1: **if** $\mathbf{z}_s$ and $\mathbf{z}_g$ are **not** connected in G **then**
2: **return** $\emptyset$
3: **else**
4: **return** result of graph search on G where the cost of traversing an edge $E = \{\mathbf{z}_1, \mathbf{z}_2, \mathcal{P}\}$ is given by $d(\mathbf{z}_1, \mathbf{z}_2)$, the Euclidean distance between the centroids of the two formations.
5: **end if**

To reach the final goal, the team of robots sequentially follows the intermediate setpoints in the path and the local planner minimizes the deviation towards the associated configuration $\mathbf{z}_w$ at every instance. To make progress towards the intermediate setpoints, and for improved performance, we slightly modify Algorithm 1 by selecting the convex region $\mathcal{P} \subset \hat{\mathcal{F}}^\tau(t_o)$ as follows.

We do not directly use the convex regions stored in the global path, since the robots need to account for dynamic obstacles in real-time.

First, a convex region $\mathcal{P}_{f_o \to w}$ containing all the robots in the team at their current positions, and directed towards $\mathbf{w}$ is computed. If the setpoint $\mathbf{w}$ is also inside the polytope, i.e. $\mathbf{z}_0$, $\mathbf{w} \in \mathcal{P}_{f_o \to w}$, then we use this polytope $\mathcal{P} := \mathcal{P}_{f_o \to w}$ for navigation.

Otherwise, we compute its intersection with a polytope generated with only the centroid of $\mathbf{z}_0$ and directed towards the waypoint $\mathbf{w}$, i.e. $\mathcal{P} := \mathcal{P}_{f_o \to w} \cap \mathcal{P}_{o \to w}$. In this case the robots may reconfigure to make progress towards the intermediate setpoint. See Section 2.4 for details on the computation of these polytopes.

5. Extension for mobile manipulators

In this section, we describe an extension where a team of mobile manipulators collaboratively carry an object in a dynamic environment. To achieve this, the robot shape, formation definition, and optimization equations are modified, and the derivations follow the same line of thought of the previous sections.

5.1. Robot and formation definition

The formation is defined by n mobile manipulators, each equipped with a robotic arm and grasping a rigid object at given points, see Figure 5(b) for an example with two mobile manipulators.

The position $\mathbf{t} \in \mathbb{R}^2$ and orientation $\theta_o \in SO(2)$ of the object can vary. Each manipulator $i = \{1, \ldots, n\}$ can rotate around the grasping point $\mathbf{g}_i$ by an angle $\theta_i \in [\theta_{min}, \theta_{max}] \subset [-\pi/2, \pi/2] \subset SO(2)$ relative to the direction pointing towards the center of the object. Each manipulator may change the arm length, denote $a_i \in [a_{min}, a_{max}] \subset \mathbb{R}$ the distance from the center of the robot to the grasping position.

The vertices of the object relative to its center and expressed in the object coordinate frame are denoted by $\{\mathbf{w}_1^0, \ldots, \mathbf{w}_{n_0}^0\}$. The vertices of manipulator i relative to its center and expressed in the robot coordinate frame are denoted by $\{\mathbf{w}_1^i, \ldots, \mathbf{w}_{n_i}^i\}$. Denote the grasping positions on the object, relative to its center, by $\mathbf{g}_0^i$. Without loss of generality, in this derivation we assume that the angle between the robot base and arm is constant (additional degrees of freedom could added to the set of optimization variables) and denote by $\mathbf{a}_0^i$ the vector from the center of the robot to the grasping point, of length a_i, expressed in the robot coordinate frame. An example with three robots grasping a triangular object is shown in Figure 7.

Given the configuration $\mathbf{z} = [\mathbf{t}, \theta_o, a_1, \ldots, a_n, \theta_1, \ldots, \theta_n] \in \mathbb{R}^{3+2n}$ of the formation, the vertices of the manipulator and object, expressed in the world coordinate frame, are

$$\begin{aligned} \mathbf{v}_j^0 &= [\mathbf{t}, 0]^T + R_{\theta_0} \mathbf{w}_j^0, && \forall j \in [1, n_0] \\ \mathbf{v}_j^i &= [\mathbf{t}, 0]^T + R_{\theta_0} \left(\mathbf{g}_0^i + R_{\theta_i} (\mathbf{w}_j^i - \mathbf{a}_0^i) \right), && \forall j \in [1, n_i] \end{aligned} \tag{22}$$

where R_θ is the rotation matrix $[\cos(\theta), -\sin(\theta), 0; \sin(\theta), \cos(\theta), 0; 0, 0, 1]$. We denote by

$$\mathcal{V}(\mathbf{z}, i) = [\mathbf{v}_1^i, \ldots, \mathbf{v}_{n_i}^i] \tag{23}$$

the set of n_i vertices for the object ($i = 0$) and each robot ($i > 0$) at configuration $\mathbf{z}$.

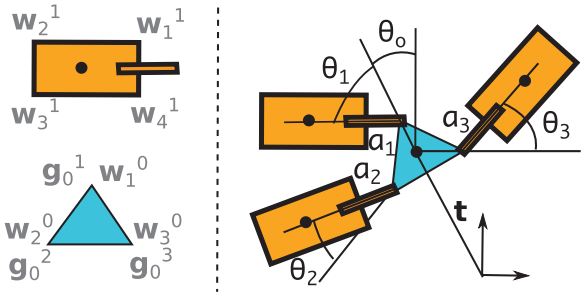


Fig. 7. Right: optimization variables for three mobile manipulators grasping a triangular object. Left: vertices of a mobile manipulator and the grasped object. For this triangular object, the vertices are equal to the grasping points.

5.2. Obstacle-free convex region

Recalling Section 2.3, the position-time obstacle-free workspace is given by

$$\begin{aligned} \mathcal{O}^{\tau}(t_o) &= \mathcal{O} \times [0,\tau] \cup \bigcup_{\substack{t\in[0,\tau] \\ j\in\mathcal{I}_D}} \mathcal{D}_j(t_o+t) \times t \subset \mathbb{R}^4 \\ \mathcal{F}^{\tau}(t_o) &= \mathbb{R}^3 \times [0,\tau] \setminus \mathcal{O}^{\tau}(t_o) \subset \mathbb{R}^4 \end{aligned} \tag{24}$$

where, now, the static and dynamic obstacles are *not* dilated.

The collision-free convex polytope containing the robots at their current state and directed towards the goal is

$$\mathcal{P}_{f_o\to g} := \mathcal{P}^{[\mathbf{g}(t_1)\times\tau]}_{\cup_{i=0:n}[\mathbf{v}^i_1(t_o),\ldots,\mathbf{v}^i_{n_i}(t_o)]\times 0}(\mathcal{F}^{\tau}(t_o)) \tag{25}$$

additional polytopes in free space are computed analogously.

5.3. Nonlinear optimization

Consider $\mathbf{z}_{ini} = [\mathbf{t}^{ini}, \theta_o^{ini}, a_1^{ini}, \ldots, a_n^{ini}, \theta_1^{ini}, \ldots, \theta_n^{ini}]$ the initial configuration of the robots and object at the current time.

Since the robots are rigidly attached to the object, we must explicitly impose that the transition between the current and the target configuration remains within the convex polytope. Consider $K > 0$ interpolation steps, and denote by $\mathbf{z}_\lambda$ the linearly interpolated configurations such that $\mathbf{z}_{\lambda=0} = \mathbf{z}_{ini}$ and $\mathbf{z}_{\lambda=K} = \mathbf{z}$. Angles are interpolated in the direction of minimum change and each interpolated configuration $\mathbf{z}_\lambda$ is expressed as a function of $\mathbf{z}_{ini}$ and $\mathbf{z}$, e.g. $\mathbf{t}_\lambda = \lambda(\mathbf{t} - \mathbf{t}^{ini})/K + \mathbf{t}^{ini}$.

Recalling equation (22) and the representation of the collision-free polytope $\mathcal{P}$ by a set of linear constraints, as in equation (12), the optimization is

$$\begin{aligned} \mathbf{z}^* = \quad & \arg\min_{\mathbf{z}} \; ||\mathbf{t} - \mathbf{g}(t_1)||^2 \\ s.t. \quad & \mathcal{V}(\mathbf{z}_\lambda, i) \times t_1 \subset \mathcal{P}_{f_o\to g} \\ & \theta_{min} \le \theta_i \le \theta_{max} \\ & a_{min} \le a_i \le a_{max} \\ & \forall j \in \{1,\ldots,n_i\},\ \forall i \in \{0,\ldots,n\} \\ & \forall \lambda \in \{1,\ldots,K\} \end{aligned} \tag{26}$$

The derivatives of the constraints with respect to the optimization variable $\mathbf{z}$ are computed analogously to equation (17).

6. Results

In this section, we present experiments with a team of three Kuka Youbot mobile manipulators collaboratively carrying an object and simulations with teams of quadrotor UAVs navigating in 3D environments. A video illustrating the results accompanies this paper and is also available at https://youtu.be/sDNqdEPA7pE.

The mobile manipulators are holonomic platforms. For the UAVs, we employ the nonlinear dynamical model and LQR controller used by Alonso–Mora et al. (2015c) with real quadrotors.

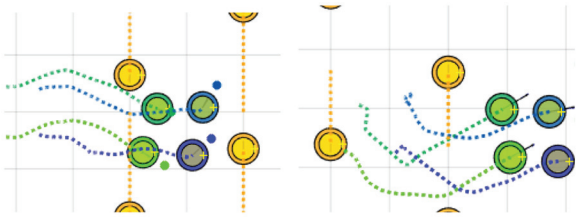
We use SNOPT by Gill et al. (2002) to solve the nonlinear program via Sequential Quadratic Programming, a goal-directed version of IRIS by Deits and Tedrake (2014) to compute the large convex regions and the Drake toolbox [3] from MIT to handle quaternions.

6.1. Multiple aerial vehicles in formation

To evaluate our approach in 3D environments with aerial vehicles we present experiments in three simulated scenarios. The first scenario consists of four controlled quadrotors and four dynamic obstacles. The second scenario consists of four controlled quadrotors flying in formation and avoiding several static obstacles and one dynamic obstacle. The last scenario involves sixteen quadrotors flying in formation through a narrow corridor. In our visualizations, we employ a cylinder since that is the shape we use for collision avoidance. Internally the quadrotors have an attitude controller and position controller and change their 3D pose within the enclosing cylinder, which is always kept vertical.

In all cases a new formation is computed every 2 s. The individual collision avoidance planners run at 5 Hz. The quadrotors move at speeds between 0.5 m/s and 1.5 m/s. In our simulations with four quadrotors a time horizon $\tau = 4$ s is considered. This is longer than the required time to reach a full stop. For the experiments with sixteen quadrotors a time horizon of $\tau = 10$ s is chosen, due to the large size of the formation and the scenario.

Consider the first scenario. Figure 8 shows the trajectories of four quadrotors (in green and blue) passing through two lanes of dynamic obstacles (in yellow). The dynamic obstacles in the left lane move downwards at 0.4 m/s and the ones in the right move upwards with the same speed. Two default formations are considered, square (which is preferred) and diamond. The goal for the formation follows a constant velocity trajectory along the middle horizontal line and the team successfully adapts the parameters of the formation to remain collision-free and pass in-between the obstacles. In this case, we imposed that the formation remains on the horizontal plane for illustrative purposes.



(a) Snapshot at 15 s, paths: 0-15 s (b) Snapshot at 25 s, paths 15-25 s

Fig. 8. Top view. Four robots (green-blue) navigate a 8 x 15 m^2 environment with two lanes of dynamic obstacles (orange). The four robots locally reconfigure the formation and make progress towards the right side.

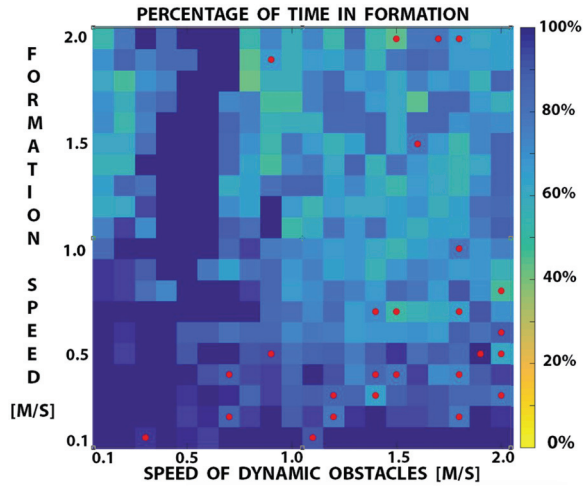


Fig. 9. Results for 3D navigation in the scenario of Figure 8 with four quadrotors and varying speeds of the dynamic obstacles and the formation. Fixed time horizon of $\tau = 5$ s is used in all experiments. The figure shows the percentage of time in which the target formation is within $\mathcal{P}_{f_o->g}$. This is, the percentage of time in which the formation is guaranteed to be maintained. If the optimization becomes unfeasible, due to the higher speed of the dynamic obstacles and the limited time horizon, the formation might break (in this case the target formation is found within $\mathcal{P}_{o->g}$ or $\mathcal{P}_g$, but not in $\mathcal{P}_{f_o->g}$). Scenarios for which a collision happened are marked with a red dot. As will be discussed in Section 7, collisions may arise mostly due to unforeseen changes of speed by moving obstacles or the limited time horizon.

In order to evaluate the robustness of the method, we performed additional experiments for this first scenario for varying speeds of the dynamic obstacles and the quadrotors flying in formation. The results are presented in Figure 9. We observe that most of the time the target formation for the robots is within $\mathcal{P}_{f_o->g}$, thus the formation is kept. But, at high speeds, in order to quickly progress towards the goal, the robots temporally break it and select a target one within $\mathcal{P}_{o->g}$ or $\mathcal{P}_g$. Good results, especially at lower dynamic obstacle speeds are observed. We believe that the results could be improved with an adaptive time horizon depending on the speed of both the moving obstacles and the formation. In scenarios with only static obstacles, the formation is maintained at all times.

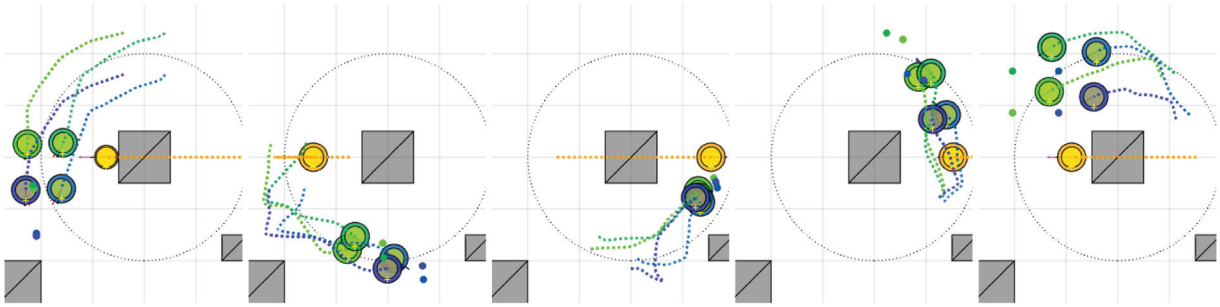
In this scenario, very few collisions arise when the target speed of the formation is higher or similar to that of the dynamic obstacles and our framework successfully drives the robots towards the goal while avoiding collisions. Some collisions arise when the speed of the dynamic obstacles is much higher than that of the formation. This is due to the local planning horizon and the robots being unable to escape on time due to their lower speed. Again, these results may be improved with an adaptive time horizon of the framework.

Next, we present experiments for the second scenario. Figure 10 shows snapshots and trajectories of four quadrotors tracking a circular trajectory while locally avoiding three static obstacles and a dynamic obstacle. Three default formations are considered here: square (1st preference), diamond (2nd preference), and line. The optimal parameters are computed with the nonlinear optimization allowing rotation in 3D (flat horizontal orientation preferred) and reconfiguration.

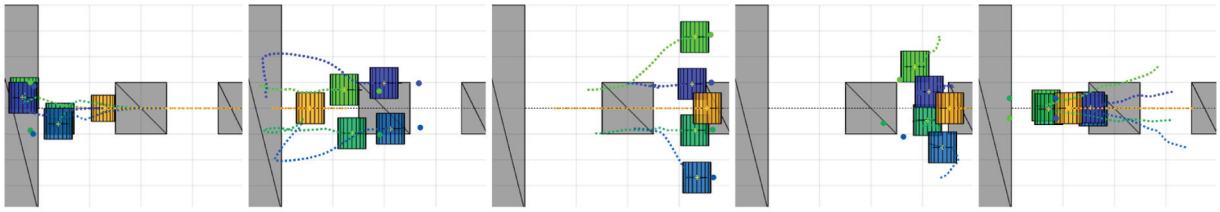
The four quadrotors start from the horizontal square and slightly tilt it (11 s) to avoid the incoming dynamic obstacle. To fully clear it while avoiding the obstacle in the lower corner, they shortly switch to a vertical line, and then back to the preferred square formation (20 s). To pass through the next narrow opening they switch back to the line formation (30 s) and then to the preferred square, tilted to avoid the dynamic obstacle (37 s). Once the obstacles are cleared they return to the preferred horizontal square formation (45 s).

Results of the third scenario, where sixteen quadrotors move along a corridor of three different widths are shown in Figure 11. Three default formations are considered: 4x4x1 defined by four vertices (preferred), 4x2x2 defined by eight vertices and 8x2x1 defined by four vertices. At each time step the method computes the optimal parameters for each of the three and selects the one of lowest cost. Between times 75 s and 110 s the method successfully rotates the formation by 90^o for it to be collision free (the default formations were horizontal, which is also preferred in the cost function).

Thanks to the abstraction of a formation by the vertices of its convex hull, see Section 2.5, the computation time of the nonlinear optimization is independent of the number of robots - as long as the same convex shape is maintained - and can be executed in real-time. It is worth noting that in this algorithm the dimension of the space where the robots move has little influence in the computational cost, which depends mostly on the number of variables defining the formation. In Table 2 we provide computational times for our implementation using a 2.6 GHz i7 laptop. The approach shows close to real time performance, typically below 300 milliseconds

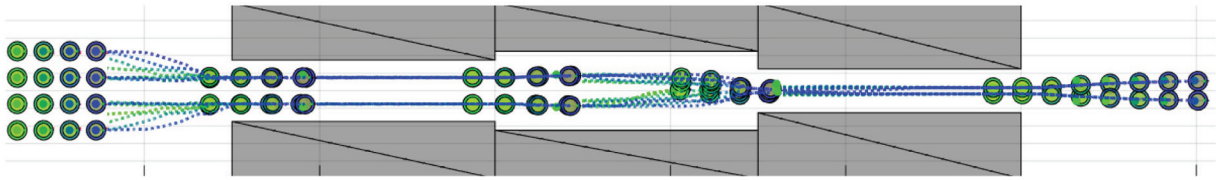


(a) Top view. From left to right, snapshots at 11 s, 20 s, 30 s, 37 s, and 45 s, and paths of the robots in-between.

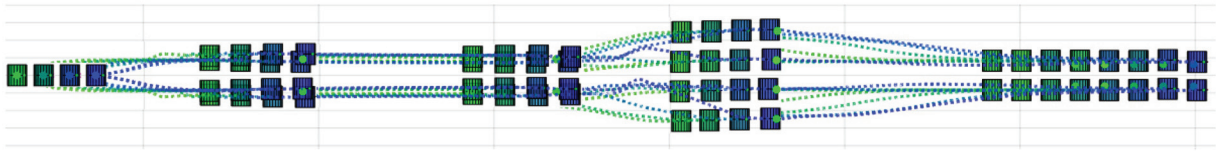


(b) Side view for the same snapshots of above. All five are obtained from a camera placed at the lower side of the top view. For clarity, the quadrotors are always displayed on top of obstacles even if they might be behind.

Fig. 10. Four quadrotors (green-blue cylinders) navigate in a 12 x 12 x 6 m^3 scenario with three static obstacles (grey) and a dynamic obstacle (yellow). The four quadrotors track a circular motion (black dots in top view) and locally reconfigure the formation to avoid collisions and make progress.



(a) Top view with robot paths and snapshots at time 0 s, 25 s, 50 s, 75 s, 110 s. Robots move from left to right.



(b) Side view with robot paths and snapshots at time 0 s, 25 s, 50 s, 75 s, 110 s. Robots move from left to right.

Fig. 11. 16 quadrotors navigate along a 70 x 10 x 10 m corridor, with obstacles shown in gray. The quadrotors locally adapt the formation to remain collision free. The following formations are observed: 4x4x1 - 4x2x2 - 4x4x1 (vertical) - 8x2x1 (vertical), finally transitioning towards horizontal 8x2x1.

Table 2. Computational time [ms] for our implementation.

Compute	Min	Mean	Max	Std deviation
Convex region	31.8	82.8	221.4	72.1
NL optimization	93.7	226.4	522.7	64.1

6.2. *Collaborative transport with two mobile manipulators*

We performed initial experiments with two mobile manipulators carrying a rigid object, as described in Section 5. In this first experiment the two robots are not allowed to change their orientation and distance with respect to the object ($\theta_{i=\{1,2\}} = 0$, $\mathbf{a}_{i=\{1,2\}} =$ constant). We optimize for the position and orientation of the object only.

Four snapshots are shown in Figure 12 of an experiment where the two mobile manipulators successfully carry the rigid object to the goal position behind the orange boxes while locally avoiding collisions with the human. In all our experiments, executed with external tracking, the robots successfully adapted their formation to avoid collisions. This assumes that the human cooperates, otherwise, collisions may still occur if the human moves faster than the robots or traps them against an obstacle.

Fig. 12. Four consecutive snapshots of an avoidance maneuver where two mobile manipulators collaboratively carry a rigid object and navigate it to the goal while adjusting their formation to avoid collisions with the orange boxes and the human. Robots and human are tracked by overhead cameras. This maneuver is performed in 1 minute.

A visualization with convex regions of another experiment is shown In Figure 13. For each snapshot the current (blue) and target (green) formation given by the optimization are displayed. The two robots successfully adapt their formation, rotating as required, to avoid both the dynamic obstacle (red) and static (grey) obstacles in this 8 m x 6 m scenario. Slices at t_o and $f_f = t_o + \tau$ of the convex region computed in position-time space are also shown for illustrative purposes. A time horizon $\tau = 2$ s was employed.

6.3. *Collaborative transport, three mobile manipulators*

We performed additional experiments with three mobile manipulators carrying an object, as described in Section 5. All three robots can change their orientation respect to the object, but their distance remains constant. We optimize for the position and orientation of the object and the orientation of all three manipulators. Given a goal for the formation, we first compute a global path with the algorithm of Section 4 considering only the two static obstacles. The local motion planner then runs at a frequency of approximately 5 Hz, accounts for the dynamic obstacle (person), and updates the parameters of the formation. A low-level controller is employed which, via high-frequency interpolation, drives the robots towards the desired formation.

We tested different configurations of the two boxes in our experimental space, covering all possible scenarios we thought of (some examples are shown in Figure 14). In all our experiments the robots could avoid collisions and reach the goal - as long as the human moved at a reasonable speed below that of the robots and did not aggressively push them against a wall.

Several configurations of the two boxes, with the computed global path, are shown in Figure 14. All of them were computed in the order of below ten seconds. The initial configuration is in the lower part of the images and the goal configuration in the upper part. In each figure, we display the samples (black dots), the convex regions (blue), the optimal formations within each intersection (green) and the path. We stop the construction of the graph as soon as the first solution is found. We observe that in general, very few iterations were required to find a feasible solution, which is also of good quality thanks to the optimizer. Navigation in all these scenarios was successfully achieved by the three mobile manipulators.

A representative experiment where the three robots navigate through the boxes and avoid a human is shown in Figure 15. For reference, in Figure 16 we show twelve different scenarios and the configuration of the three robots when navigating through the environment.

In these experiments, we employed a triangular object with foam exterior. The foam provides a small degree of deformability to compensate for the lack of compliance in the robot arms and low level controller of the mobile manipulators. Note that successful manipulation of a perfectly rigid body was shown in the previous experiments with two mobile manipulators, albeit at lower speeds.

6.4. *Global planning in large scenarios*

We also tested the approach in several larger scenarios. In Figure 17 we show two examples of the global planner in simulated 2D environments. In these two cases the global planner can run for several iterations, up to a fixed amount of time. We display all the computed convex regions and formations. After finding the first feasible path connecting the start with the goal position, we store (and display) the subsequent shorter paths found by the algorithm. An advantage of the method is that large areas in free space are explored by each convex polytope, which reduces the need for additional samples within.

7. Discussion

The method described in this paper showed good real-time performance and could successfully compute the optimal parameters for the multi-robot formation, while allowing for reconfiguration. The method provided collision-free navigation among static and dynamic obstacles in simulations with aerial vehicles and in experiments with mobile manipulators.

At least in part, the computational efficiency and the good scalability with respect to the number of robots in the formation is achieved by (a) not including the agent dynamics in the formation optimization but handling them in the individual local planners (this works well for robots with fast dynamics); and (b) considering the convex hull of the formation. In fact, the number of variables and

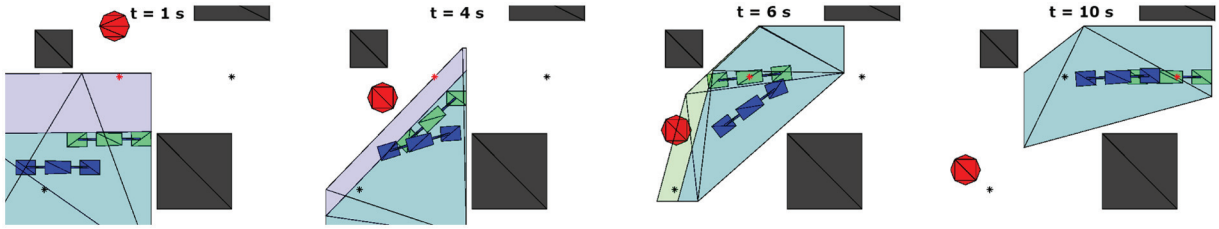


Fig. 13. Four consecutive snapshots of a 10 s avoidance maneuver where two mobile manipulators collaboratively carry a rigid object and navigate to two goals (crosses) while avoiding collisions with static (gray) and dynamic (red hexagon) obstacles. The current state of the two manipulators and the rigid object is displayed in blue and the target one (given by the optimization) in green. Two slices of the convex polytope are shown, purple for the current time $t_o = t$ (shown in the figure titles) and light green for time $t_1 = t_o + \tau$ (the intersection is the larger blueish region). The dynamic obstacle is shown at time t_o and it is moving at constant speed downwards. As displayed, the red dynamic obstacle may intersect with the light green slice of the convex polytope (at t_1), but not with the purple one (at t_o). The manipulators successfully navigate the rigid object through the two set points avoiding collisions. The initial, intermediate and final setpoints are shown with dots, the currently active one in red, the others in black.

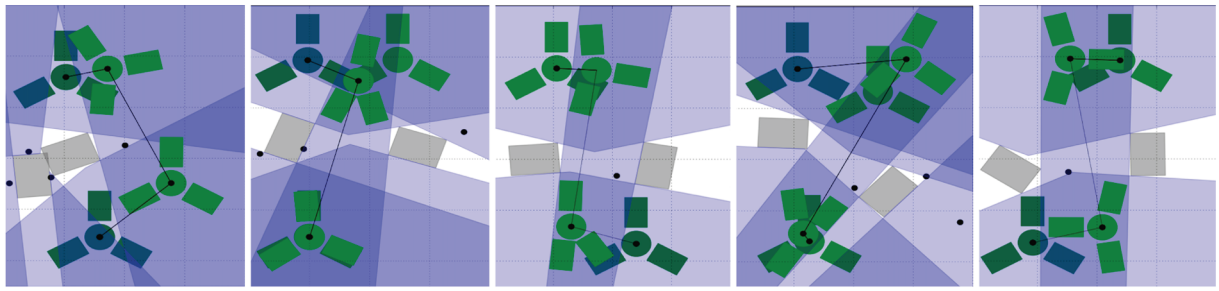
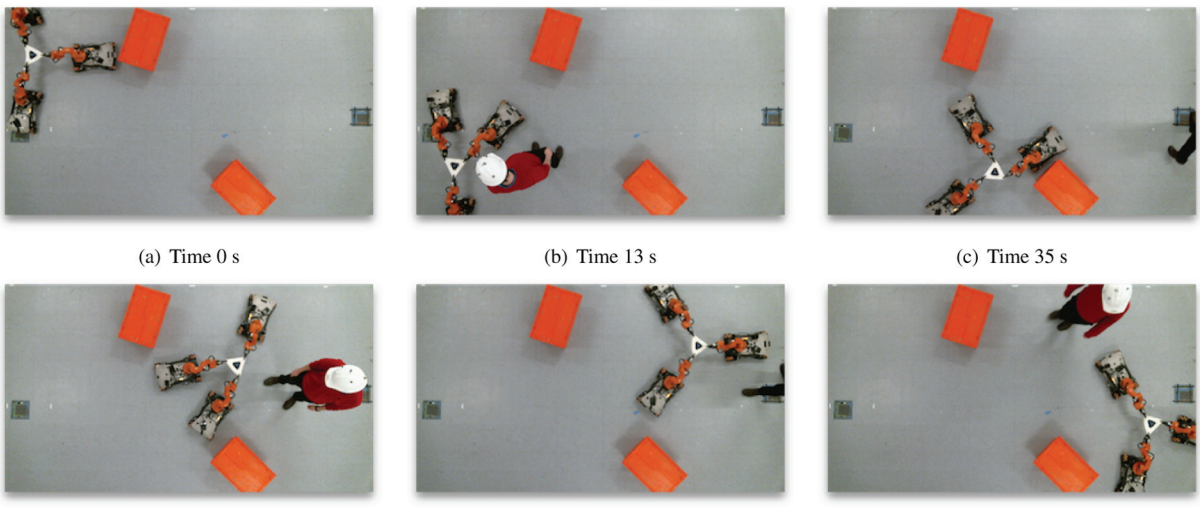


Fig. 14. Five examples of the global path planner for different scenarios with two static obstacles. The three mobile manipulators carry an object and can rotate and translate while grasping. The initial and final formations are displayed in dark green. Light green formations are additional nodes of the graph. The first feasible path is displayed with a solid black line. All samples (black dots), polytopes (blue) and optimized formations (green) within the intersections of polytopes are shown. Typically, a feasible path is found with very few samples.



(a) Time 0 s (b) Time 13 s (c) Time 35 s

(d) Time 48 s (e) Time 60 s (f) Time 80 s

Fig. 15. Three mobile manipulators collaboratively carry an object and navigate to a goal position in the other side of the room. The global path planner guides them through the two static obstacles and they locally avoid the walking human. The robots successfully adapt their formation to pass through the narrow opening and avoid the human.

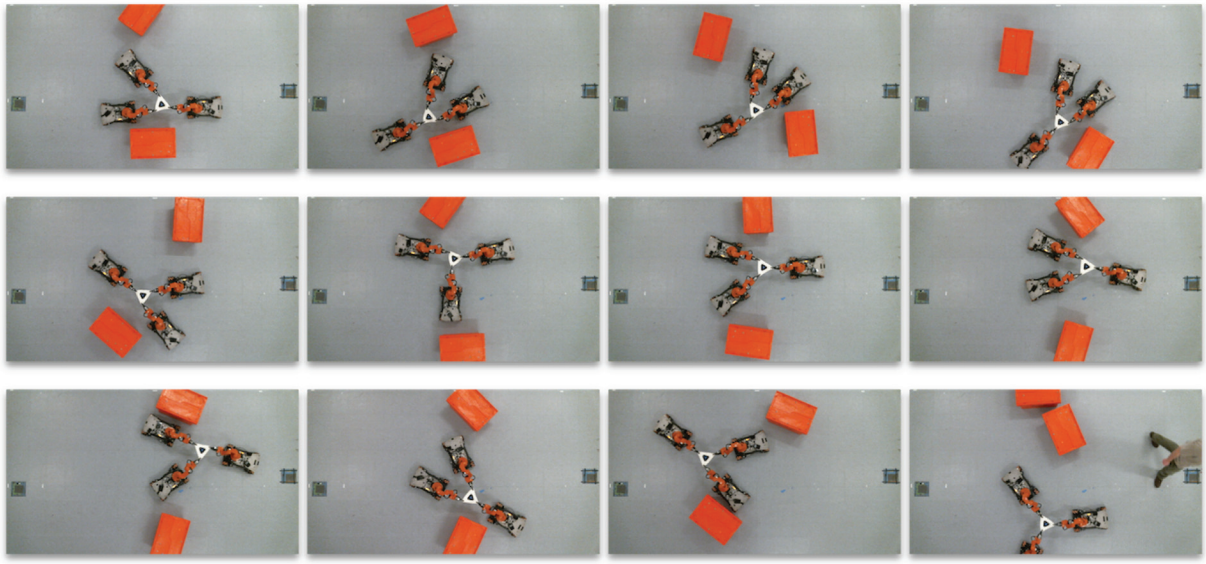
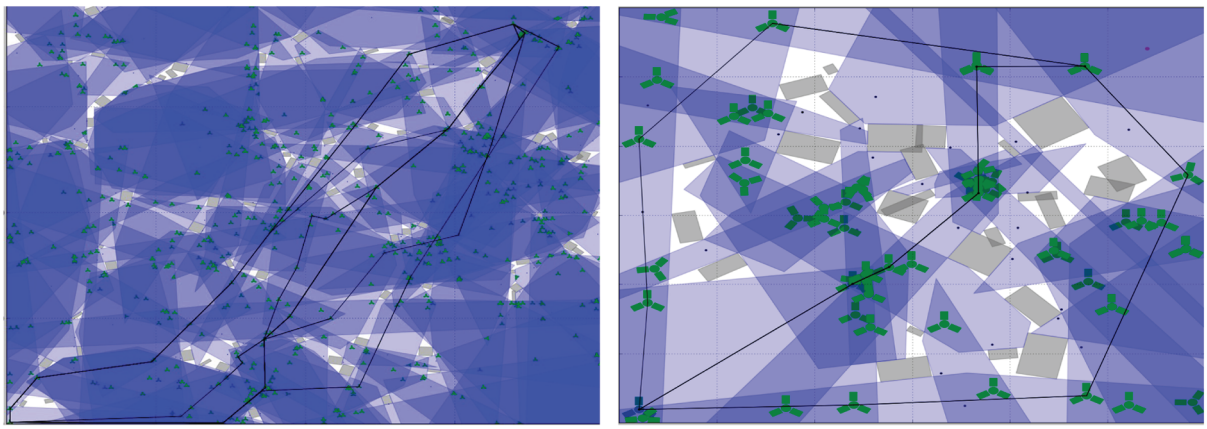


Fig. 16. Twelve different experiments where the three mobile manipulators collaboratively carry an object and navigate to a goal position in the other side of the room. For each one of them, a snapshot while they traverse through the two obstacles is shown. This experiment shows robustness to the location of the obstacles and that the robot formations vary across different execution runs. If the obstacles leave enough free space, as is the case in the lower right corner experiment, the team of robots maintain the preferred formation and do not need to rotate around their grasping points. Otherwise, they successfully adapt the configuration.



(a) Large 100 m x 100 m scenario with rectangular obstacles of size between 1 m^2 and 25 m^2. The algorithm run for 100 s.

(b) Medium 15 m x 15 m scenario with rectangular obstacles of size between 0.25 m^2 and 9 m^2. The algorithm run for 15 s.

Fig. 17. Two examples of the global path planner connecting a start (lower left corner) with a goal (upper right corner) for two large scenarios with many static obstacles. The three mobile manipulators carry an object and can rotate and translate while grasping. The algorithm runs for a fixed amount of time. The first feasible path, and the feasible paths found in subsequent iterations that decrease the cost, are displayed with a solid black line. All samples (black dots), polytopes (blue), and optimized formations (green) within the intersections of polytopes are shown.

constraints of the formation. In fact, the number of variables and constraints of the formation control method is independent from the number of robots. The optimization problem in Equation (15) has eight variables for 3D motion and $n_i \cdot n_l + 2$ constraints, where n_i is the number of vertices of the convex hull of robot positions in formation i and n_l is the number of sides in the convex polytope. We recall that the number of vertices of the convex hull depends on the shape of the formation, e.g. a square formation has four vertices, independently of the number of robots therein. For collaborative manipulators, the number of variables $(3+2n)$, and constraints $(2n+m \cdot n_l \cdot (n_o+n_i \cdot n))$, scale linearly with the number of robots. In this case, n_o and n_i are the number of vertices of the object and each robot, respectively.

In our experiments, the computation of a large obstacle-free convex polytope following Section 2.4 showed very good results, but no guarantees exist that the best volume

will be obtained. In fact, the method will converge to a local optimum of the cost function, which is guaranteed to be fully contained in free space. Searching over several regions might prove advantageous. One may also consider employing a faster, albeit suboptimal algorithm to quickly compute a convex region.

To compute the parameters of the multi-robot formation our method solves a nonlinear optimization via Sequential Convex Programming. This method converges to a local optimum of the non-convex problem. Global optimality can only be guaranteed if the original optimization problem is convex, which is typically not the case. For the non-convex case, the number of iterations required to find a locally optimal, or even feasible, solution is not defined. In practice, the method performed very well, quickly returning good parameters for the formation in all cases where a valid formation could be fitted within the convex polytope.

These observations also apply to the case of the global planner and no strong guarantees can be given for the general non-convex optimization case. Thanks to the sampling of convex regions, the method will successfully explore the whole workspace. For speed-up, as described in Algorithm 2, we limit the sampling of regions to points outside of the union of current convex regions in the graph. In most scenarios this heuristic works well, but it can potentially miss narrow openings, since, although the whole space is covered by convex regions, the intersection might not be traversable. Two advantages of this method are (a) that sampling is performed in a low dimension space - the workspace - instead of in the high-dimensional configuration space and (b) that large areas of the free space are explored/covered at once when contained within a convex polytope. This has the potential to speed-up global path planning for formations of robots.

If the optimizations are feasible and a solution is found, the motion is guaranteed to be collision-free up to the time horizon of the local planner, under the assumption that the moving obstacles maintain a constant speed. This is true because: (a) the convex region is fully contained in free position-time space; (b) the robots at their initial position and at the positions in the target formation are fully contained in the convex region and (c) the motion in between the two formations as well, if the robots move in a straight line (the linear combination of two points lies within the convex polytope). For mobile manipulators collaboratively carrying an object, this is satisfied up to the interpolation.

Nonetheless, collisions with moving obstacles can still arise if the assumptions are not met. For instance, if the moving obstacles change the direction of motion quicker than the robots can react, if the moving obstacles move too fast, if the planning horizon is not long enough, or if the team of robots are trapped in a corner from where they cannot feasibly escape.

An advantage of the method is that planning is decoupled into: (a) finding convex regions in the lower dimensional free position-time space ($\mathbb{R}^4$) and (b) efficiently optimizing the configuration of the team of robots within those convex regions. This comes at the expense of completeness, since in our approach we require that the robot team maintains a formation that does not intersect with obstacles, i.e. the robots cannot maintain a formation while letting an obstacle pass through. In the event of dynamic obstacles, the team may break the formation to let a moving obstacle pass through, and come back to the original formation as soon as there is enough free room.

Lastly, the method is general and can be adapted to other high-dimensional problems or formation definitions. The core idea of the algorithms is to generate convex obstacle-free regions and then optimize the parameters of the formation (i.e. the degrees of freedom of the high-dimensional configuration) such that the robots are fully contained in the convex region. The only requirements to adapt the method are (a) a function that converts configurations $\mathbf{z}$ to the outer vertices of the formation $\mathcal{V}(\mathbf{z}, f)$ or high-dimensional system, and (b) a way to compute derivatives with respect to the configuration $\mathbf{z}$ (unless they are computed numerically).

8. Conclusion

In this paper, we showed that navigation of teams of robots in formation among arbitrary static and dynamic obstacles can be achieved via a constrained nonlinear optimization. By first computing a large obstacle-free convex polytope and then optimizing the formation parameters, low computational cost is achieved together with good navigation results. In several simulations with aerial vehicles navigating in 3D environments we showed successful navigation in formation where robots may reconfigure the formation as required to avoid collisions and make progress.

Our method can be applied both for real-time local navigation in a dynamic environment and to compute global paths in static environments. The global planner successfully combines a sampling-based method in the workspace with nonlinear optimization for the remaining degrees of freedom of the formation, thus reducing the dimensionality of the sampling problem.

For formation control, the approach scales to teams of robots of arbitrary size, since only the convex hull of the formation is employed in the constrained optimization. Simulations with sixteen quadrotors -although more could be used - demonstrate this. The approach is general and can also be adapted to other formation definitions and applications, as showed in our experiments with three mobile manipulators collaboratively carrying an object,

In this work, we did allow for splitting and merging of robots, from/to a joint formation to/from individual navigation. An interesting avenue for future work is that of splitting and merging of the group formation into smaller sub-formations, or to maintain the formation while letting dynamic obstacles through, which is currently not possible.

Additional avenues of future research include incorporating the dynamic constraints of the robots in the nonlinear optimization problem and accounting for uncertainties in the prediction of the movement of the dynamic obstacles. In this work, the nonlinear dynamics of the robots were decoupled from the formation control and accounted for by the individual controllers locally.

Acknowledgements

The authors are grateful to Robin Deits and Hongkai Dai from MIT for their help with IRIS, Drake and SNOPT.

Funding

This work was supported in part by the MIT Lincoln Laboratory, SMARTS N00014-09-1051, pDOT ONR N00014-12-1-1000 and the Boeing Company.

Notes

1. In our implementation we represent quaternions by vectors in $\mathbb{R}^4$ of unit norm and consider the additional constraint $C_3 = \{||\mathbf{q}||^2 = 1\}$.
2. We employ the implementation within the Drake toolbox, available in RobotLocomotion/drake/.
3. http://drake.mit.edu

References

Alonso–Mora J, Baker S and Rus D (2015a) Multi-robot navigation in formation via sequential convex programming. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, Hamburg, 2015, pp. 4634–4641. DOI: 10.1109/IROS.2015.7354037.

Alonso–Mora J, Knepper RA, Siegwart R, et al. (2015b) Local Motion planning for collaborative multi-robot manipulation of deformable objects. In: *IEEE international conference robotics and automation*, Seattle, WA, 2015, pp. 5495–5502. DOI: 10.1109/ICRA.2015.7139967.

Alonso–Mora J, Montijano E, Schwager M, et al. (2016) Distributed multi-robot navigation in formation among obstacles: A geometric and optimization approach with consensus. In: *IEEE international conference on robotics and automation (ICRA)*, Stockholm, 2016, pp. 5356–5363. DOI: 10.1109/ICRA.2016.7487747

Alonso–Mora J, Naegeli T, Siegwart R, et al. (2015c) Collision avoidance for aerial vehicles in multi-agent scenarios. *Autonomous Robots* 39(1): 101–121.

Alonso–Mora J, Schoch M, Breitenmoser A, et al. (2012) Object and animation display with multiple aerial vehicles. In: *IEEE/RSJ international conference on intelligent robots and systems*, Vilamoura, 2012, pp. 1078–1083. DOI: 10.1109/IROS.2012.6385551

Augugliaro F, Schoellig AP and 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*, Vilamoura, 2012, pp. 1917–1922. DOI: 10.1109/IROS.2012.6385823.

Ayanian N, Kallem V and Kumar V (2011) Synthesis of feedback controllers for multiple aerial robots with geometric constraints. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, San Francisco, CA, 2011, pp. 3126–3131. DOI: 10.1109/IROS.2011.6094943.

Ayanian N, Kumar V and Koditschek DE (2009) Synthesis of controllers to create, maintain, and reconfigure robot formations with communication constraints. In: *International symposium on robotics research (ISRR)* 2009, Zurich, pp.625–642.

Balch T and Arkin RC (1998) Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation* 14(6): 926–939.

Balch T and Hybinette M (2000) Social potentials for scalable multi-robot formations. In: *IEEE international conference on robotics and automation (ICRA) Symposia proceedings*, San Francisco, CA, 2000, pp. 73–80 vol. 1.DOI: 10.1109/ROBOT.2000.844042.

Barfoot TD and Clark CM (2004) Motion planning for formations of mobile robots. *Robotics and Autonomous Systems* 46: 65–78.

Belta C and Kumar V (2004) Abstraction and control for groups of robots. *IEEE Transactions on Robotics* 20(5): 865–875.

Bento J, Derbinsky N, Alonso–Mora J, et al. (2013) A message-passing algorithm for multi-agent trajectory planning. In: *Advances in neural information processing systems NIPS*. pp. 521–529.

Bertsekas DP (1988) The auction algorithm: A distributed relaxation method for the assignment problem. *Annals of Operations Research* 14(1): 105–123.

Cheah CC, Hou SP and Slotine J (2009) Region-based shape control for a swarm of robots. *Automatica* 45: 2406–2411.

Chen Y, Cutler M and How JP (2015) Decoupled multiagent path planning via incremental sequential convex programming. In: *IEEE international conference on robotics and automation (ICRA)*, Seattle, WA, 2015, pp. 5954–5961. DOI: 10.1109/ICRA.2015.7140034.

Chen YQ and Wang Z (2005) Formation control: A review and a new consideration. In: *IEEE/RSJ international conference on intelligent robots and systems*, 2005, pp. 3181–3186. DOI: 10.1109/IROS.2005.1545539.

Conner DC, Rizzi AA and Choset H (2003) Composition of local potential functions for global robot control and navigation. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)* (Cat. No. 03CH37453), 2003, pp. 3546–3551 vol. 3. DOI: 10.1109/IROS.2003.1249705.

Cortés J (2009) Global and robust formation-shape stabilization of relative sensing networks. *Automatica* 45(12): 2754–2762.

Deits R and Tedrake R (2014) Computing large convex regions of obstacle-free space through semidefinite programming. In: *workshop on the algorithmic fundamentals of robotics*, Cham: Springer International Publishing, pp. 109–124, DOI: 10.1007/978-3-319-16595-0_7.

Deits R and Tedrake R (2015) Efficient mixed-integer planning for UAVs in cluttered environments. In: *IEEE international conference on robotics and automation (ICRA)*, Seattle, WA, 2015, pp. 42–49. DOI: 10.1109/ICRA.2015.7138978.

Derenick J, Spletzer J and Kumar V (2010) A semidefinite programming framework for controlling multi-robot systems in dynamic environments. In: *49th IEEE conference on decision and control* (CDC), Atlanta, GA, 2010, pp. 7172–7177. DOI: 10.1109/CDC.2010.5717711.

Derenick JC and Spletzer JR (2007) Convex optimization strategies for coordinating large-scale robot formations. *47th IEEE Transactions on Robotics* 23: 1252–1259.

Desai JP, Ostrowski JP and Kumar V (2001) Modeling and control of formations of nonholonomic mobile robots. *IEEE Transactions on Robotics and Automation* 17(6): 905–908.

Dimarogonas DV and Johansson KH (2008) On the stability of distance-based formation control. In: *IEEE conference on decision and control (CDC)*, Cancun, 2008, pp. 1200–1205. DOI: 10.1109/CDC.2008.4739215.

Dimarogonas DV and Kyriakopoulos KJ (2005) Formation control and collision avoidance for multi-agent systems and a connection between formation infeasibility and flocking behavior. In: *Proceedings of the 44th IEEE conference on decision and control (CDC)*, 2005, Sevilla, pp. 84–89. DOI: 10.1109/CDC.2005.1582135.

Dunbar WB and Murray RM (2002) Model predictive control of coordinated multi-vehicle formations. In: *Proceedings of the 41st IEEE conference on decision and control*2002, vol. 4, pp. 4631–4636. DOI: 10.1109/CDC.2002.1185108.

Egerstedt M and Hu X (2001) Formation constrained multi-agent control. *IEEE Transactions on Robotics and Automation* 17(6): 947–951.

Erdmann M and Lozano–Perez T (1987) On multiple moving objects. *Algorithmica* 2: 477–521.

Fax JA and Murray RM (2004) Information flow and cooperative control of vehicle formations. *IEEE Transactions on Automatic Control* 49(9): 1465–1476.

Fredslund J and Mataric MJ (2002) A general algorithm for robot formations using local sensing and minimal communication. *IEEE Transactions on Robotics and Automation* 18(5): 837–846.

Gill PE, Murray W and Saunders MA (2002) SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM journal on optimization* 12(4): 979–1006.

Hsieh A, Kumar V and Chaimowicz L (2008) Decentralized controllers for shape generation with robotic swarms. *Robotica* 26: 691–701.

Kallem V, Komoroski AT and Kumar V (2011) Sequential composition for navigating a nonholonomic cart in the presence of obstacles. *IEEE Transactions on Robotics* 27(6): 1152–1159.

Khatib O, Yokoi K, Chang K, et al. (1996) Vehicle/arm coordination and multiple mobile manipulator decentralized cooperation. In: *Proceedings of the 1996 IEEE/RSJ international conference on intelligent robots and systems* Osaka, 1996, vol. 2, pp. 546–553. DOI: 10.1109/IROS.1996.570849.

Kloder S and Hutchinson S (2006) Path planning for permutation-invariant multirobot formations. *IEEE Transactions on Robotics* 22(4): 650–665.

Krontiris A, Louis S and Bekris KE (2012) Multi-level formation roadmaps for collision-free dynamic shape changes with non-holonomic teams. In: *IEEE international conference on robotics and automation (ICRA)* Saint Paul, MN, 2012, pp. 1570–1575. DOI: 10.1109/ICRA.2012.6225372.

Kuhn HW (1955) The Hungarian method for the assignment problem. In: *Naval Research Logistics*. pp. 83–97 Wiley Subscription Services, Inc. DOI: 10.1002/nav.3800020109.

Kushleyev A, Mellinger D and Kumar V (2012) Towards a swarm of agile micro quadrotors. *Autonomous Robots* 35(4): 287–300.

Latombe JC (1991) *Robot Motion Planning*. Boston: Kluver.

LaValle SM (2006) *Planning Algorithms*. Cambridge: Cambridge University Press.

LaValle SM and Kuffner JJ (2001) Randomized kinodynamic planning. *The International Journal of Robotics Research* 20(5): 378–400.

Michael N and Kumar V (2008) Controlling shapes of ensembles of robots of finite size with nonholonomic constraints. *Proceedings of Robotics: Science and Systems IV*, June 2008.

Michael N, Zavlanos MM, Kumar V, et al. (2008) Distributed multi-robot task assignment and formation control. In: *Proceedings of the IEEE international conference on robotics and automation*, Pasadena, CA, 2008, pp. 128–133. DOI: 10.1109/ROBOT.2008.4543197.

Morgan D, Subramanian GP, Chung SJ, et al. (2016) Swarm assignment and trajectory optimization using variable-swarm, distributed auction assignment and sequential convex programming. *The International Journal of Robotics Research* 35(10): 1261–1285.

Ogren P, Egerstedt M and Hu X (2001) A control Lyapunov function approach to multi-agent coordination. In: *Proceedings of the 40th IEEE conference on decision and control*, Orlando, FL, 2001, pp. 1150–1155, vol.2. DOI: 10.1109/.2001.981040.

Olfati–Saber R and Murray RM (2002) Distributed cooperative control of multiple vehicle formations using structural potential functions. In: *IFAC world congress*, vol. 35.1, pp. 495–500, Barcelona.

Ren W and Sorensen N (2008) Distributed coordination architecture for multi-robot formation control. *Robotics and Autonomous Systems* 56(4): 324–333.

Sabattini L, Secchi C and Fantuzzi C (2011) Arbitrarily shaped formations of mobile robots: Artificial potential fields and coordinate transformation. *Autonomous Robots* 30: 385–397.

Saha I, Ramaithitima R, Kumar V, et al. (2014) Automated composition of motion primitives for multi-robot systems from safe LTL specifications. In: *IEEE/RSJ international conference on intelligent robots and systems*, Chicago, IL, 2014, pp. 1525–1532. DOI: 10.1109/IROS.2014.6942758.

Sugar TG and Kumar V (2002) Control of cooperating mobile manipulators. *IEEE Transactions on Robotics and Automation* 18(1): 94–103.

Tang CP, Bhatt R and Krovi VN (2004) Decentralized kinematic control of payload transport by a system of mobile manipulators. In: *IEEE international conference on robotics and automation* 2004, pp. 2462–2467, vol. 3, New Orleans. DOI: 10.1109/ROBOT.2004.1307430.

Tanner HG, Jadbabaie A and Pappas GJ (2007) Flocking in fixed and switching networks. *IEEE Transactions on Automatic Control* 52(5): 863–868.

Tanner HG, Loizou SG and Kyriakopoulos KJ (2003) Nonholonomic navigation and control of cooperating mobile manipulators. *IEEE Transactions on Robotics and Automation* 19(1): 53–64.

Yu J and LaValle SM (2013) Fast, near-optimal computation for multi-robot path planning on graphs. In: *AAAI conference on artificial intelligence, late breaking papers*, July, Washington.

Zhou D and Schwager M (2015) Virtual rigid bodies for coordinated agile maneuvering of teams of micro aerial vehicles. In: *IEEE international conference on robotics and automation (ICRA)*, Seattle, WA, 2015, pp. 1737–1742. DOI: 10.1109/ICRA.2015.7139422.

Appendix: Collision avoidance

In this appendix, we provide a description of the method for collision avoidance employed with the aerial vehicles. We implement the convex optimization introduced by Alonso–Mora et al. (2015c) with identical motion constraints and constraints for avoidance of other agents. This approach adapts to changes in the environment, avoids moving obstacles, and respects the dynamics of the robot via a set of motion primitives. Each motion primitive was defined to track a constant reference velocity with a robot-specific controller.

We extend the method towards environments with complex static obstacles. In particular, using the convex polytope computation described in Section 2.4, we add a new constraint to guarantee that the motion of the robot is within the obstacle-free workspace $\mathcal{F}$. Following the notation of Alonso–Mora et al. (2015c), the additional constraint for avoidance of static obstacles is computed as follows.

Denote by $\bar{\mathcal{O}}^{\epsilon} = \{\mathbf{p} \in \mathbb{R}^3 \mid \mathcal{A}_{\epsilon}(\mathbf{p}) \cap \mathcal{O} \neq \emptyset\}$ the set of static obstacles dilated by the robot volume plus a small value $\varepsilon > 0$. A convex polytope $\mathcal{P}_{\mathbf{p}_i}^{\mathbf{r}_{\sigma(i)}}(\mathbb{R}^3 \setminus \bar{\mathcal{O}}^{\epsilon}) \subset \mathbb{R}^3$ is computed following Section 2.4. This polytope is in obstacle-free space, contains the initial position $\mathbf{p}_i$ of the robot, and is directed towards the robot's goal position $\mathbf{r}_{\sigma(i)}$ in the new formation.

In Alonso–Mora et al. (2015c) the collision avoidance algorithm was formulated as a constrained optimization in velocity space. Therefore, the convex region needs to be converted to an equivalent region in velocity space. Given the time horizon τ of the planner, this is formally

$$\mathcal{P}^{u}(\mathbf{p}, \varepsilon) := (\mathcal{P}_{\mathbf{p}_i}^{\mathbf{r}_{\sigma(i)}}(\mathbb{R}^3 \setminus \bar{\mathcal{O}}^{\epsilon}) - \mathbf{p}_i)/\tau \tag{27}$$

where each linear constraint defining $\mathcal{P}_{\mathbf{p}_i}^{\mathbf{r}_{\sigma(i)}}(\mathbb{R}^3 \setminus \bar{\mathcal{O}}^{\epsilon})$ is expressed relative to the current position of the robot and is divided by the time horizon. In particular, if the robot selects a reference velocity that satisfies this constraint, i.e. $\mathbf{u} \in \mathcal{P}^{u}(\mathbf{p}, \varepsilon)$, then all future positions up to the time horizon τ are within $\mathcal{P}_{\mathbf{p}_i}^{\mathbf{r}_{\sigma(i)}}(\mathbb{R}^3 \setminus \bar{\mathcal{O}}^{\epsilon})$. This polytope is then included in the distributed convex optimization of Alonso–Mora et al. (2015c).

If the target position $\mathbf{r}_{\sigma(i)}$ of robot i is within its line of sight (this is the case if $\mathbf{z} \subset \mathcal{P}_{f_o \to g}$), then the collision avoidance algorithm successfully drives the robot towards it. Otherwise, a global planner, such as the ones proposed by Bento et al. (2013) or Yu and LaValle (2013), can be used for guidance.

Index to multimedia extension

Archives of IJRR multimedia extensions published prior to 2014 can be found at http://www.ijrr.org, after 2014 all videos are available on the IJRR YouTube channel at http://www.youtube.com/user/ijrrmultimedia