

Collision Avoidance for Multiple Agents with Joint Utility Maximization

Javier Alonso-Mora^{1,2}, Martin Rufli¹, Roland Siegwart¹ and Paul Beardsley²

Abstract—In this paper a centralized method for collision avoidance among multiple agents is presented. It builds on the velocity obstacle (VO) concept and its extensions to arbitrary kino-dynamics and is applicable to heterogeneous groups of agents (with respect to size, kino-dynamics and aggressiveness) moving in 2D and 3D spaces. In addition, both static and dynamic obstacles can be considered in the framework.

The method maximizes a joint utility function and is formulated as a mixed-integer quadratic program, where online computation can be achieved as a trade-off with solution optimality. In experiments with groups of two to 50 agents the benefits of the joint utility optimization are shown. By construction, it's suboptimal variant is at least as good as comparable decentralized methods, while retaining online capability for small groups of agents. In its optimal variant, the proposed algorithm can provide a benchmark for distributed collision avoidance methods, in particular for those based on the VO concept that take interaction into account.

I. INTRODUCTION

The computation of global collision-free trajectories in a multi-robot setting remains challenging. Consequently, the overall problem is typically hierarchically decomposed into a global planning part (which may not consider robot constraints, nor interaction properties), and a local reactive component, which handles these unmodeled effects.

In this paper we explore this latter element from a centralized perspective in search for optimality and present a novel collision avoidance method for multiple agents. The method is based on the velocity obstacle (VO) paradigm and operates by maximizing a joint utility function. It shares similarities with the method presented in [1] but is not limited to the case of two agents. While centralization restricts the method's applicability to systems that are controlled from a single unit, many of the current multi-robot frameworks – such as the display presented in [2] – are operated in such a manner. Furthermore, the method does not require that all objects are under the control of the central unit, as both static obstacles and not-controlled agents can be seamlessly taken into account.

In addition, the proposed algorithm may serve as a benchmark for distributed, VO-based collision avoidance methods, such as [3] and [4]. In the aforementioned reciprocal methods, the space of feasible motions is significantly reduced in order to attain collision avoidance guarantees in the distributed case. This can be avoided via a centralized optimization, such as the one presented here. Furthermore,

the formulation presented in this work retains the properties of [4] which applies to heterogeneous groups of agents and respects their individual kino-dynamic properties.

The presented method formulates the joint optimization as a (mixed-integer) quadratic program (QP/MIQP), where online computation is achieved as a trade-off with optimality. While the MIQP formulation allows to compute truly optimal solutions, the QP formulation remains sub-optimal but enables online performance for very large groups of robots. Nonetheless, the QP formulation at all times dominates the distributed methods' solution set.

Related to our approach, centralized MIQP [5] and QP [6] optimizations have recently been developed to compute collision-free trajectories for groups of quad-rotor helicopters. In contrast to our formulation, these methods optimize intermediate states to reach a final destination and are therefore costly optimizations ideal for off-line trajectory generation. Instead, our method achieves online performance for *collision avoidance* by computing collision-free inputs in each time step, and can be combined with the distributed reciprocal collision avoidance method presented in [7] for aerial vehicles.

Besides the described main contributions, several improvements to the existing reciprocal collision avoidance framework are proposed. These include the use of repulsive forces to maintain a minimum inter-agent distance (in III-C), the use of per-agent weights to handle individual robots' aggressiveness (in V-A) and the use of an asymmetric cost function which penalizes an agent's changes in speed differently from changes in orientation (in V-A). Furthermore, the method is able to model avoidance preference on one side (in VI-A), which is a standard rule in our society.

For clarity of exposition, the method is described for kino-dynamically constrained agents moving in 2D Euclidean space. Nevertheless, it readily applies to agents moving in 3D Euclidean space – see Section VIII-C. The remainder of this paper is structured as follows. Sections II and III introduce the system and the basic concepts. Section IV formalizes the joint optimization, whereas Sections V and VI describe the optimization formulated as a low complexity QP and an optimal MIQP, respectively. In Section VII formal properties are given and in Section VIII extensions are discussed. Experimental results are presented in Section IX and Section X concludes the paper.

Throughout this paper scalars x are set in lower case italics, vectors \mathbf{x} in lower case bold and matrices X in uppercase italics. The sub-index i indicates agent identity, whilst super-index k indicates the time-step of the collision-avoidance control-loop.

¹ J. Alonso-Mora, M. Rufli and R. Siegwart are with the Autonomous Systems Lab, ETH Zurich, 8092 Zurich, Switzerland {jalonso, rufli, rsiegwart}@ethz.ch

² J. Alonso-Mora, P. Beardsley are with Disney Research Zurich, 8092 Zurich, Switzerland {jalonso, pab}@disneyresearch.com

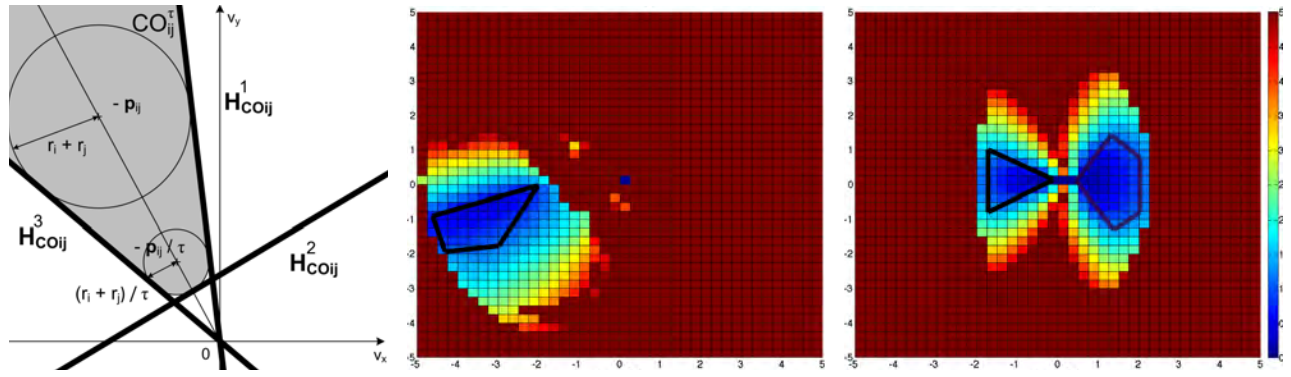


Fig. 1. *Left*: Reference control obstacle and its approximation by three half-planes (3). *Middle / right*: Tracking errors ε in [m] for sampled reference velocities \mathbf{u} (axis: $\mathbf{u}_x, \mathbf{u}_y$ [m]), saturated at 5m and for the kino-dynamics of the vehicle used in the experiments. For details we refer to [4]. The initial velocity and steering angle are -4 m/s / 0 m/s and -25° / 0° respectively. The reachable set for a given ε is approximated by one / two convex polygons.

II. OVERVIEW AND SYSTEM DEFINITION

Although the presented framework equally applies to alternative formulations, in this paper we employ the collision avoidance concept described in [4]: for each agent and at each time-step, a straight-line constant-velocity reference trajectory (given by $\mathbf{u}_i^k \in \mathbb{R}^2$) is obtained following the basic VO concept. The obtained *holonomic* control obstacle (III-A) represents the relative reference trajectories which would lead to a collision if agents were able to perfectly track it. In absence of perfect tracking capability, following our previous work [4], a variable parameter $\varepsilon_i \geq 0$ is introduced as an enlargement of the robots' radii. For given agent kinodynamics and reference tracking controller, the reachable set of reference velocities (III-B) then denotes the reference trajectories that can be tracked with an error below ε_i , thereby limiting the agent's maneuverability.

We denote the state of each agent by \mathbf{z}_i^k . In the case of a kino-dynamic bicycle, it includes position \mathbf{p}_i^k , steering angle and driving velocity. The state update for the agent is given by $f(\mathbf{z}_i^k, \mathbf{u}_i^k, t)$, with t time and $f(\cdot)$ a trajectory tracking controller continuous in \mathbf{z}_i . $f(\cdot)$ is described in [4] and [8] for the case of bicycle and unicycle kinematics respectively.

III. PRELIMINARY DEFINITIONS

A. Reference velocity obstacle

Given two robots with relative position $\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{p}_j$ and radii r_i and r_j , where $r_* = r_*^{real} + \varepsilon_*$, and considering straight-line constant-speed reference trajectories defined by the terminal velocities \mathbf{u}_i and \mathbf{u}_j , the set of relative reference velocities $\mathbf{u}_{ij} = \mathbf{u}_i - \mathbf{u}_j$ leading to collision within time τ (control obstacle CO_{ij}^τ) is given by [9]

$$CO_{ij}^\tau = \{\mathbf{u}_{ij} \mid \exists t \in [0, \tau], \|\mathbf{p}_{ij} + t\mathbf{u}_{ij}\|_2 \leq r_i + r_j\}. \quad (1)$$

This is equivalent to a truncated cone, as shown in Fig. 1 and is only computed if the distance between the two agents is below a threshold ($p_{ij} < K_d$).

Approximate formulation: The non-convex space $\mathbb{R}^2 \setminus CO_{ij}^\tau$ can be approximated by n_{CO} half planes verifying

$$\bigcup_{l \in [1, n_{CO}]} H_{CO,ij}^l \subset \mathbb{R}^2 \setminus CO_{ij}^\tau. \quad (2)$$

For $n_{CO} = 3$, as in Fig. 1, the half-planes are

$$\begin{aligned} H_{CO,ij}^1 &= \{[\cos(\alpha + \beta), \sin(\alpha + \beta)]' \cdot \mathbf{u}_{ij} \leq 0\} \\ H_{CO,ij}^2 &= \{-\mathbf{p}_{ij}/p_{ij} \cdot \mathbf{u}_{ij} \leq ((p_{ij} - r_i - r_j)/\tau)\} \\ H_{CO,ij}^3 &= \{[\cos(\alpha - \beta), \sin(\alpha - \beta)]' \cdot \mathbf{u}_{ij} \leq 0\}, \end{aligned} \quad (3)$$

where $p_{ij} = \|\mathbf{p}_{ij}\|$, $\alpha = \text{atan2}(-\mathbf{p}_{ij})$ and $\beta = \text{acos}((r_i + r_j)/p_{ij})$. $H_{CO,ij}^1$ and $H_{CO,ij}^3$ represent avoidance to the right and to the left, respectively. $H_{CO,ij}^2$ represents a head-on maneuver, which remains collision-free up to $t = \tau$.

B. Reachable set of reference velocities

In order to guarantee collision-free motions it must be guaranteed that each agent remains within ε_i of its reference trajectory. For a maximum tracking error ε_i and current state \mathbf{z}_i^k , the set of reference velocities \mathbf{u}_i^k that can be achieved with position error lower than ε_i is given by

$$\begin{aligned} R_i &= R(\mathbf{z}_i^k, \varepsilon_i) = \\ &= \{\mathbf{u}_i^k \mid \|(\mathbf{p}_i^k + t\mathbf{u}_i^k) - f_p(\mathbf{z}_i^k, \mathbf{u}_i^k, t)\| \leq \varepsilon_i, \forall t > 0\}, \end{aligned} \quad (4)$$

where $f_p(\cdot)$ represents the position of the agent out of the state given by $f(\cdot)$.

The set R_i can be precomputed, in closed form for diff-drive vehicles [8] or by forward simulation for Ackerman vehicles [4]. Examples of the latter are displayed in Fig. 1.

Approximate formulation: If the agents are holonomic the set R_i represents a circle, centered either at zero (\mathbf{v}_{max} constraint, no continuity in velocity) or at the current velocity (\mathbf{a}_{max} constraint). In these cases, the constraints may be written as a quadratic constraint and used in the formulation described in the forthcoming sections. For general kinodynamics, if the set R_i is well behaved (this is the case for the studied diff-drive and Ackerman agents) it can be approximated by one or two convex polygons (intersections of half planes). The half-planes must verify

$$\bigcap_{l \in [1, n_{R,i}^1]} H_{R,i}^{1,l} \cup \bigcap_{l \in [1, n_{R,i}^2]} H_{R,i}^{2,l} \subset R_i, \quad (5)$$

with $n_{R,i}^2 = 0$ if R_i can be approximated by a single convex polygon. Fig. 1 depicts both cases.

The number and position of the half-planes can be computed from the set R_i by maximizing the covered area within a certain threshold. Values of $n_{R,i}^1$ range between three and five. These half-planes may also be pre-computed for different values of current state and tracking error.

C. Preferred reference velocity and inter-agent distance

For each agent independently, and at each time step a preferred reference velocity $\bar{\mathbf{u}}_i^k$ is computed, which can be given by a vector to a goal destination, a vector field, a global planner or a trajectory tracker.

In general, velocity or control obstacle methods tend to minimize inter-agent distance, bringing it to zero in the limit which may become unsafe in real scenarios. To approximately maintain a minimum inter-agent distance, $\bar{\mathbf{u}}_i^k$ is corrected by adding a repulsive velocity given by

$$\bar{\mathbf{u}}_{ij}^{\text{rep},k} = \max(0, V_r(D_r - p_{ij}^k)/(D_r - r_i - r_j)) \mathbf{p}_{ij}^k/p_{ij}^k, \quad (6)$$

where V_r is the maximum repulsive force and D_r the preferred minimal inter-agent distance.

IV. JOINT OPTIMIZATION

Let us now consider a group of N agents that move on a planar surface. Let them be controlled by a centralized entity. Then the joint set of reference velocities is denoted by $\mathbf{u}^k = [\mathbf{u}_1^k, \dots, \mathbf{u}_N^k] \in \mathbb{R}^{2N}$. Given a joint utility function to maximize at the current time step, or equivalently, a joint cost function $\mathcal{C}(\mathbf{u}^k)$ to minimize, the optimal collision-free reference velocities are given by

$$\begin{aligned} & \underset{\mathbf{u}^k}{\text{argmin}} && \mathcal{C}(\mathbf{u}^k) \\ \text{s. t.} && \mathbf{u}_i^k - \mathbf{u}_j^k &\notin CO_{ij}^{\tau} \quad \forall i < j \text{ with } p_{ij}^k < K_d \\ && \mathbf{u}_i^k &\in R_i \quad \forall i \end{aligned} \quad (7)$$

A naive solution strategy is to sample in \mathbb{R}^{2N} . In that case, constraint verification is very fast (evaluation of an equation for CO_{ij}^{τ} and a look-up in a table for R_i). However, the dimension of the sampling space becomes prohibitive for even moderate N . We thus propose two alternative strategies, where the cost function is quadratic and the constraints are half-planes. The first strategy approximates the problem via Quadratic Programming (QP). The second formulation consists of a Mixed Integer Quadratic Program (MIQP). The latter formulation is optimal with respect to (7).

V. QP FORMULATION: LOW COMPLEXITY SOLUTION

This method provides either a set of collision-free reference controls \mathbf{u}_i^k or an infeasible-problem flag with very low computational cost. It can be shown that the solution is, by construction, at least as good as that of the distributed VO-based reciprocal collision avoidance schemes presented in [3] and [4].

A. Joint utility function

The joint cost function is designed as a quadratic function. It's most intuitive form being the cumulated distance of each agent's reference velocity to its preferred one $\bar{\mathbf{u}}_i^k$.

In [10] it was shown that pedestrians prefer to maintain a constant velocity in order to minimize energy. Preference was given to turning instead of changing the speed. Furthermore, penalizing changes in speed leads to a reduction of deadlock situations. This idea can be formalized as an elliptical cost, higher in the direction parallel to $\bar{\mathbf{u}}_i^k$, and lower perpendicular to it. Let $\lambda > 0$ represent the relative weight.

To take into account avoidance preference between agents, which can be interpreted as aggressive vs. shy behavior, relative weights ω_i are added.

Let γ_i be the orientation of $\bar{\mathbf{u}}_i^k$. The joint cost is given by $\sum_{i=1}^N (0.5\mathbf{u}_i^{k,l} H_i^k \mathbf{u}_i^k + \mathbf{f}_i^k \mathbf{u}_i^k)$, with

$$\begin{aligned} H_i^k &= \omega_i \begin{bmatrix} \cos \gamma_i & -\sin \gamma_i \\ \sin \gamma_i & \cos \gamma_i \end{bmatrix} \begin{bmatrix} \lambda & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \gamma_i & \sin \gamma_i \\ -\sin \gamma_i & \cos \gamma_i \end{bmatrix} \\ \mathbf{f}_i^k &= -(H_i^k \bar{\mathbf{u}}_i^k)^T. \end{aligned} \quad (8)$$

Let H^k and \mathbf{f}^k be the $2N \times 2N$ matrix and the $2N$ vector formed by concatenation of the individual H_i^k and \mathbf{f}_i^k .

B. Constraints

The n_{CO} half-plane constraints (3) representing each of the control obstacle constraints are non additive. Therefore only one can be selected for each CO_{ij} . We denote the index of the selected half-plane by s_{ij} , which can be chosen following different criteria. Sensible choices include:

- 1) Fixed side for avoidance: $s_{ij} = 1$ for every CO
- 2) Maximum constraint satisfaction for the previous relative reference velocity $\mathbf{u}_i^{k-1} - \mathbf{u}_j^{k-1}$.
- 3) Maximum constraint satisfaction for the preferred relative reference velocity $\bar{\mathbf{u}}_i^k - \bar{\mathbf{u}}_j^k$.

The second option is the preferred criteria as it maximizes the feasible region of reference controls.

If the reachable set (5) is approximated by two convex regions, only one (t_i) is selected, either the closest one to the previous reference control or the closest one to the preferred reference control. Note that at high speeds the reachable set is formed by a single region.

The constraint given by each half-plane is written in lifted domain (\mathbb{R}^{2N}), with four non-zero terms (i, j) for each $H_{CO,ij}^{s_{ij}}$ and two (i) for each $H_{R,i}^{t_i,l}$. These constraints are denoted by $\mathbf{n}_*^* \mathbf{u}^k \leq c_*^*$.

C. Optimization

The optimum of the simplified problem is found by solving the following quadratic program with linear constraints

$$\begin{aligned} & \underset{\mathbf{u}^k}{\text{argmin}} && 0.5\mathbf{u}^{k,l} H^k \mathbf{u}^k + \mathbf{f}^k \mathbf{u}^k \\ \text{s. t.} && \mathbf{n}_{CO,ij}^{s_{ij}} \mathbf{u}^k &\leq c_{CO,ij}^{s_{ij}} \quad \forall i < j \mid p_{ij} < K_d \\ && \mathbf{n}_{R,i}^{t_i,l} \mathbf{u}^k &\leq c_{R,i}^{t_i,l} \quad \forall i, \forall l \in [1, n_{R,i}^l] \end{aligned}$$

Efficient algorithms, and off-the shelf libraries exist¹ to solve this QP in real time for large number of agents.

¹We use the IBM ILOG CPLEX optimizer.

VI. MIQP FORMULATION: OPTIMAL SOLUTION

Given that the feasible space is reduced by selecting a single half-plane constraint for each CO , the QP optimization of the previous section returns a suboptimal solution. The optimal reference velocities can be obtained by adding binary variables to the formulation. The optimization then becomes a Mixed Integer Quadratic Problem that can be solved via branch-and-bound, where the maximum number of explored nodes defines the ratio between optimality and computational time. To this end, n_{CO} binary variables need to be added for each control obstacle and another two for each reachable set approximated by two convex regions.

In particular, a constraint given by $\mathbf{n}_*^* \mathbf{u}_*^* \leq c_*^*$ is rewritten as $\mathbf{n}_*^* \mathbf{u}_*^* - Mb_*^* \leq c_*^*$, where M is a large enough constant and b_*^* the binary variable. $b_*^* = 0$ if the constraint is to be satisfied and $b_*^* = 1$ otherwise. Let \mathbf{b}^k be the vector of binary variables.

A. Joint utility function

The cost function with respect to the reference velocity \mathbf{u}^k is retained from the QP case. In contrast to the QP case where a single half-plane constraint was selected beforehand for each CO , now all of them are included in the optimization. Therefore, a side preference for the collision avoidance can now be added by appropriately selecting a penalty for the binary variables. Let us denote by $b_{ij,s}$ the n_{CO} binary variables for the half-plane constraints representing CO_{ij} .

For the case of $n_{CO} = 3$, the rule: "prefer to avoid on the right" is added by penalizing $b_{ij,1} = 1$ with a weight w_s . The vector of linear cost with respect to the binary variables is then given by

$$\mathbf{f}_b^k = \underbrace{[w_s \ 0 \ 0 \ \dots \ w_s \ 0 \ 0 \ 0 \ 0 \ \dots \ 0 \ 0]}_{3|CO|} \underbrace{[0 \ 0 \ \dots \ 0 \ 0]}_{2|I_2|},$$

where CO is the set of CO constraints, I_2 is the set of agents whose reachable set is approximated by two convex regions, $I_1 = [1, N] \setminus I_2$ and $|X|$ the cardinality of X .

B. Constraints

All half-plane constraints are included in the optimization, binary variables are added. Further constraints on the binary variables are added to impose that only one out of the n_{CO} $H_{CO,ij}^s$ constraints is active for each CO_{ij} and that only one out of the two convex regions is active for $i \in I_2$.

C. Optimization

At each time-step k , the joint optimal solution is found by solving the following Mixed Integer Quadratic Program with linear constraints and binary variables

$$\begin{aligned} \underset{\mathbf{u}^k, \mathbf{b}^k}{\text{argmin}} \quad & 0.5 \begin{bmatrix} \mathbf{u}^k & \mathbf{b}^k \end{bmatrix} \begin{bmatrix} H^k & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^k \\ \mathbf{b}^k \end{bmatrix} + \begin{bmatrix} \mathbf{f}^k & \mathbf{f}_b^k \end{bmatrix} \begin{bmatrix} \mathbf{u}^k \\ \mathbf{b}^k \end{bmatrix} \\ \text{s. t.} \quad & \mathbf{n}_{CO,ij}^s \mathbf{u}^k - Mb_{ij,s}^k \leq c_{CO,ij}^s \quad \forall ij, s \in \text{Set1} \\ & \sum_{s=1}^{n_{CO}} b_{ij,s}^k = n_{CO} - 1 \quad \forall i < j | p_{ij}^k < K_d \\ & \mathbf{n}_{R,i}^{1,l} \mathbf{u}^k \leq c_{R,i}^{1,l} \quad \forall i, l \in \text{Set2} \\ & \mathbf{n}_{R,i}^{t,l} \mathbf{u}^k - Mb_{i,l}^k \leq c_{R,i}^{t,l} \quad \forall i, t, l \in \text{Set3} \\ & b_{i,1}^k + b_{i,2}^k = 1 \quad \forall i \in I_2, \end{aligned}$$

where $\text{Set1} = \{i < j | p_{ij}^k < K_d, s \in [1, n_{CO}]\}$, $\text{Set2} = \{i \in I_1, l \in [1, n_{R,i}^1]\}$ and $\text{Set3} = \{i \in I_2, l \in [1, n_{R,i}^t], t = \{1, 2\}\}$. The number of binary variables increases fast with the number of agents, therefore a limit in the number of explored nodes is set to achieve good performance. Complexity and run-time analysis are given in Section IX. This optimization can be solved with off-the-shelf MIP solvers¹.

D. Initialization of the MIQP

To reduce the computational time of the algorithm an initial point may be specified. The QP solution of Section V with the constraints that a priori maximize the feasible set of reference velocities (i.e. selection criteria of 2 in Section V-B) is well suited for this purpose. This typically returns a feasible, although not optimal, solution with very low computational time. Optimality of the solution increases with time as more nodes are explored. Furthermore, the algorithm can exit at any time to cope with run-time constraints. This method thus features anytime properties.

VII. FORMAL PROPERTIES

A. Complexity analysis

The maximum number of control obstacles equals $N(N-1)/2$, nonetheless, only those for which agents are closer than a constant K_d are considered, furthermore, a maximum number of control obstacles K_n per robot might be fixed.

1) *QP formulation*: The optimization is defined by $2N$ continuous variables. The number of constraints is given by

$$\begin{aligned} |CO| &\leq \min(N(N-1)/2, K_n N) \quad (\text{linear}) \\ \sum_{i=1}^N n_{R,i}^1 &\simeq 4N \quad (\text{linear}) \text{ or } N \quad (\text{quadratic}), \end{aligned}$$

depending on if the reachable set is approximated by a convex polygon or an ellipse. This optimization can be solved very efficiently even for large number of agents.

2) *MIQP formulation*: The optimization is defined by $2N$ continuous variables and $n_{CO}|CO| + 2|I_2|$ binary variables ($n_{CO} = 3$ here). The number of constraints is given by

$$\begin{aligned} n_{CO}|CO| &\leq n_{CO} \min(N(N-1)/2, K_n N) \quad (\text{linear}) \\ \sum_{i=1}^N n_{R,i}^1 + \sum_{i \in I_2} n_{R,i}^2 & \quad (\text{linear}) \text{ or } N + |I_2| \quad (\text{quadratic}) \\ |CO| + |I_2| & \quad (\text{linear}). \end{aligned}$$

Due to the relatively large number of binary variables, this optimization can only be solved inside a real-time control loop for a low number of agents. For larger groups of robots, a maximum number of explored nodes is set and a solution given by the QP is refined. Further details on computational time are given in Section IX.

B. Collision-avoidance guarantees

We distinguish between three cases.

1) *QP/MIQP feasible*: If the optimization program is feasible at the current time step, a reference velocity is found for every agent that guarantees collision-free motion up to at least time τ , typically a few seconds. Nonetheless, if all the agents are taken into account and none of the half-plane constraints $H_{CO,ij}^2$ are active for agent i , its reference velocity is collision-free to infinity.

2) *QP/MIQP unfeasible*: If the optimization is unfeasible, τ can be decreased (equivalent to a translation of the $H_{CO,ij}^2$ constraints), although the optimization may still be unfeasible and safety decreases. In order to prioritize safety, if the QP/MIQP is unfeasible, the speed of all vehicles is reduced at maximum deceleration rate till halt or the optimization becomes feasible. A per-vehicle strategy based on the slack variables remains as future work.

3) *Dynamic obstacles*: The feasibility of the optimization indicates if the dynamic obstacles can be avoided (assuming they adhere to their predicted reference velocity) or a collision is imminent. A fast control loop is able to handle small deviations in the prediction.

VIII. EXTENSIONS

A. Static obstacles

Static obstacles are added for each agent i by first computing the control obstacle with respect to obstacle \mathcal{O}

$$CO_{i\mathcal{O}}^\tau = \{\mathbf{u}_i \mid \exists t \in [0, \tau], \mathbf{p}_i + t\mathbf{u}_{ij} \in \mathcal{O} + B(r_i)\}, \quad (9)$$

where $\mathcal{O} + B(r_i)$ represents the Minkowski sum of the obstacle with a disk of radius r_i . Each $CO_{i\mathcal{O}}^\tau$ is approximated by one or more half-plane constraints that can be added in the QP or MIQP optimizations. The computation of this constraint is highly efficient if the obstacle is approximated by an ellipse or polygon. Due to the localness of the collision avoidance method, only the obstacles in the vicinity of the agent and in direct line of sight need to be added. To avoid deadlocks a global planner is required for guidance.

B. Uncontrolled dynamic obstacles

Consider d an agent of radius r_d that is not controlled by the system, and $\hat{\mathbf{u}}_d$ its estimated reference velocity, given for instance by a prediction based on its previous states. For any agent i in its vicinity, the control obstacle CO_{id}^τ and its approximation by half-planes are given by equations (1) and (3), where $\hat{\mathbf{u}}_d$ is known and not an optimization variable.

C. 3D Euclidean Space

For ease of explanation we introduced the method for agents moving on a 2D plane. Nevertheless, it readily extends to agents moving in 3D space.

- Reference velocities: $\mathbf{u}_i^k \in \mathbb{R}^3$.
- Control obstacles CO_{ij}^τ : truncated cones, given by Equation (1), which can be properly approximated by $n_{CO} = 5$ half-spaces.
- Reachable sets R_i : in [7] such a construction via an LQR trajectory tracking controller is described, where the reachable set is approximated by a sphere.

In this case, the QP optimization is characterized by $3N$ variables, $|\mathcal{CO}| \leq K_n N$ linear constraints and N quadratic constraints, which can be solved efficiently with low computational cost.

The MIQP is characterized by $3N$ continuous variables, $n_{CO}|\mathcal{CO}|$ binary variables, $(n_{CO} + 1)|\mathcal{CO}|$ linear constraints and N quadratic constraints. A detailed study of this case remains as future work.

IX. EXPERIMENTAL RESULTS

Experiments are performed in simulation for agents of $1.3m$ outer-radius with bicycle kinematics and maximum values of $5m/s$ (driving speed), $2m/s^2$ (driving acceleration), 30° (steering angle) and $30^\circ/s$ (steering velocity). The control law of [4] is used to track a reference trajectory and the remaining parameters of the optimization are set to $\varepsilon = 1m$, $\tau = 6s$ (reduced to $\tau = 3s$ if unfeasible), $K_d = 25m$, $D_r = 9.2m$, $V_p = V_r = 4m/s$ (equal to preferred speed), $\lambda = 2$, $\omega_s = 1.5$, a maximum of $10N$ CO_{ij} constraints are included in the optimization (sorted by p_{ij}) and a maximum of 200 nodes are explored in the MIQP. Position noise of $0.1m$ is added and the size of the scenario is $(15 + 1.5N)m$. Timings for the optimization are given for a Matlab-mex interface of CPLEX 11 running in a single thread (25% CPU) of a 2.66GHz laptop. In the accompanying video a representative run of the experiments is shown.

The experiments of this section are performed and compared for both the QP and the MIQP. For the QP, and for each CO , the $H_{CO,ij}^{s_{ij}}$ constraint is selected following the criteria of maximum constraint satisfaction with respect to the previous relative reference velocity (Section V-B-2). This prioritizes feasibility and gives good results in general, but ignores symmetries. For instance, if for Experiment 1 the constraints $H_{CO,ij}^{s_{ij}}$ are selected following Section V-B-1 a coordinated anti-clockwise rotation would be imposed and no deadlock would appear for the QP case, but this would reduce performance in the general case (i.e. Experiment 2).

Experiment 1: Two to 50 agents exchange antipodal positions on a circle, where $\bar{\mathbf{u}}_i^k$ is the vector to the goal of magnitude V_p . Time to convergence and computation time are shown in Fig. 2. All agents successfully reach their goal positions with the MIQP formulation. For large groups of agents a deadlock appears when the suboptimal QP is used without imposed anti-clockwise coordination. With the MIQP formulation coordination is always achieved leading to convergence for any number of agents.

Experiments 2 and 3: Two to 50 agents, divided in two groups of equal size, each tracks a given path as shown in Fig. 3. The preferred reference velocity $\bar{\mathbf{u}}_i^k$ is given by a trajectory tracker with $4m/s$ preferred speed on the curve. Each experiment is repeated five times with agents starting from a random configuration. Combined results of Experiments 2 and 3 are shown in the bottom row of Fig. 2 and in Fig. 3. The MIQP outperforms the QP (resulting in a lower tracking error) but requires higher computational time.

In both experiments computational time strongly depends on the number of neighbors considered in the collision avoidance. In Experiment 1 this is higher as all agents pass near the center of the circle. Experiment 2 is a more realistic scenario where only a few agents are in the neighborhood. Especially in this latter case, the centralized QP approach provides real time performance ($< 0.03s$) even for large groups of agents. On the other hand, the MIQP keeps real time capabilities for groups below eight agents.

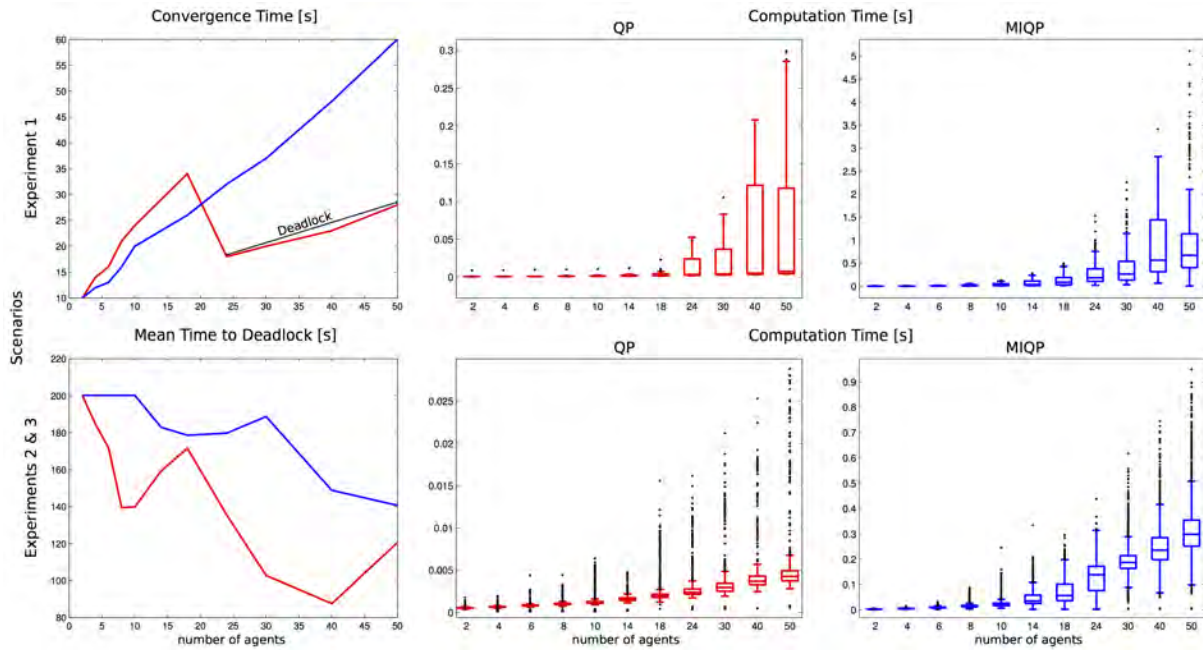


Fig. 2. *Top left:* Time to convergence / deadlock for Exp. 1 and changing number of agents, red: QP, blue: MIQP. *Top middle / right:* Computational time statistics for an optimization step (QP / MIQP respectively) of Exp. 1. *Bottom left:* Mean time to deadlock for five runs of Exp. 2 and 3 (200 s maximum simulated time). *Bottom middle / right:* Computational time statistics for an optimization step (QP / MIQP respectively) of Exp. 2 and 3 combined.

X. CONCLUSION

In this paper, a centralized method for collision avoidance among multiple agents has been presented that maximizes a joint utility function. The algorithm builds on the concept of velocity obstacles and its extension to arbitrary kinodynamics and applies to heterogeneous groups of agents moving in 2D or 3D environments. The joint optimization is formulated as a QP / MIQP where real-time computation can be achieved as a trade-off with optimality. In experiments with groups of two to 50 agents we showed that the joint MIQP outperforms a joint sub-optimal QP, which by construction is, at least, as good as equivalent distributed methods. Furthermore, the proposed algorithm, not only demonstrates the value of coordination, but may serve as

a benchmark for distributed collision avoidance methods, in particular for those based on the velocity obstacle paradigm and those that model interaction.

REFERENCES

- [1] Y. Abe and M. Yoshiki, "Collision avoidance method for multiple autonomous mobile agents by implicit cooperation," in *Intelligent Robots and Systems, 2001. Proc. 2001 IEEE/RSJ Int. Conf. on*, vol. 3, pp. 1207–1212 vol. 3, 2001.
- [2] J. Alonso-Mora, A. Breitenmoser, M. Ruffli, R. Siegwart, and P. Beardsley, "Image and animation display with multiple robots," *Int. Journal of Robotics Research*, vol. 31, pp. 753–773, May 2012.
- [3] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Int. Symp. on Robotics Research*, 2009.
- [4] J. Alonso-Mora, A. Breitenmoser, P. Beardsley, and R. Siegwart, "Reciprocal collision avoidance for multiple car-like robots," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2012.
- [5] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program (miqp) trajectory generation for heterogeneous quadrotor teams," in *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, May 2012.
- [6] F. Augugliaro, A. P. Schoellig, and R. D'Andrea, "Generation of collision-free trajectories for a quadcopter fleet: A sequential convex programming approach," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012.
- [7] J. Alonso-Mora, M. Schoch, A. Breitenmoser, R. Siegwart, and P. Beardsley, "Object and animation display with multiple aerial vehicles," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012.
- [8] J. Alonso-Mora, A. Breitenmoser, M. Ruffli, P. Beardsley, and R. Siegwart, "Optimal reciprocal collision avoidance for multiple non-holonomic robots," in *Proc. Int. Symp. on Distributed Autonomous Robotics Systems*, 2010.
- [9] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998.
- [10] S. J. Guy, J. Chhugani, S. Curtis, P. Dubej, M. Lin, and D. Manocha, "Pedestrians: A least-effort approach to crowd simulation," *Proc. ACM SIGGRAPH/Eurographics Symp. on Computer Animation*, pp. 119–128, 2010.

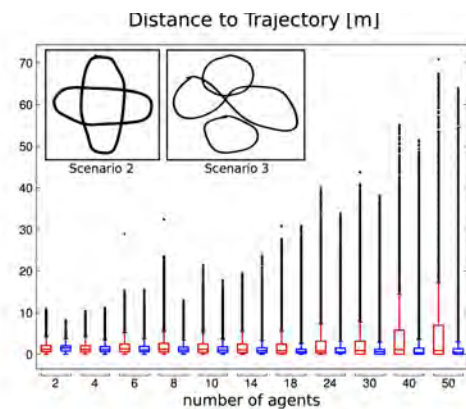


Fig. 3. Distance [m] to the desired path for varying number of agents for combined experiments 2 and 3. Red: QP, blue: MIQP, black: outliers.