Preprint version of:
Alejo D, Cobano JA, Heredia G and Ollero A (2016), A Reactive Method for Collision Avoidance in Industrial Environments, Journal of Intelligent & Robotic Systems, December 2016, Vol. 84(1), pp. 745–758.

Journal of Intelligent & Robotic Systems manuscript No.
(will be inserted by the editor)

A Reactive Method for Collision Avoidance in Industrial Environments

D. Alejo, J.A. Cobano, G. Heredia, A. Ollero

Received: date / Accepted: date

Abstract This paper presents a reactive method for collision avoidance with multiple aerial vehicles that has been applied in real time considering industrial environments. The proposed method is based on the 3D-Optimal Reciprocal Collision Avoidance algorithm. The main contribution of the proposed method is that it takes into consideration 3D modeled static obstacles. Therefore, it has been successfully applied in realistic industrial environments with the presence of complex static obstacles. Considerations of dynamic constraints of the aerial vehicles have been added. The algorithm has been integrated in ROS framework and tested in simulation. Several simulations with up to eight aerial vehicles have been performed, including long endurance cooperative missions. Finally, the second main contribution consists in the evaluation of several real experiments with up to four aerial vehicles which have been carried out in the testbed of the Center for Advanced Technologies (CATEC) facilities. The aerial vehicles flew in the presence of static obstacles and avoided potential collisions by modifying the planned trajectories in real-time.

Keywords Collision · Avoidance · Real-time · Reactive · Cooperative · Experimental · Safety · Reciprocal

D. Alejo, Robotics, Vision and Control Group Engineering School, University of Seville, 41092 Seville, Spain
Tel.: +34-954487343
Fax: +34-954487340
E-mail: dalejo@us.es
J. A. Cobano · G. Heredia · A. Ollero
Robotics, Vision and Control Group Engineering School, University of Seville, 41092 Seville, Spain

1 Introduction

Multiple-vehicle systems are being extensively studied in the last years in order to be applied in cooperative mission such as fire detection and monitoring [1] or surveillance [2]. The coordination and collision avoidance play a crucial role in these kind of applications. Particularly, reactive methods should compute solutions in real time whenever a potential collision is detected. Moreover, a good scalability of the methods is essential.

The ARCAS FP7 European Project is developing a cooperative free-flying robot system for assembly and structure construction [3]. The ARCAS system uses helicopters and quadrotors with multi-link manipulators for assembly tasks [4]. The aerial vehicles carry structure parts that will be assembled at the target destination. An important part in ARCAS is cooperative assembly planning and safe trajectory generation to perform the coordinated missions, assuring that neither the aerial vehicles nor the manipulators or the carried objects collide with each other.

Trajectory planning algorithms are used in these missions with multiple vehicles. Works published in the literature such as [5] [6] [7][8][9] are not able to compute a solution in real time for reactive purposes. In this case, the computational load should be very low as the algorithms should monitor the safety of the system several times per second. Many works have been also published related to the conflict resolution and detection problem but, in general, they present the same limitation [10] [11] [12][13].

A system for assembly and structure construction with multiple aerial vehicles which automatically identifies conflicts among them is presented in [14]. It computes collision-free trajectories whose quality improves when available computation time increases. Thus, a fea-

sible but suboptimal initial solution is quickly computed. This system presents deficiencies for reactive collision avoidance and reactions to unexpected situations.

Different reactive collision avoidance methods have been published in [15] [16] [17]. The method known as Reciprocal Collision Avoidance (RCA)[17] considers multiple vehicles navigating in a common environment and each vehicle takes half the responsibility of avoiding pairwise collisions. Most approaches solve the problem by considering zero order planning, that is, generate paths in the position space. On the other hand, RCA is formulated in the velocity space, so this is a first order planning procedure. Thus, RCA easily handles moving obstacles and also kinematic and dynamics constraints of the vehicle. The latter constraints can easily be taken into account by reducing the set of velocities that a vehicle can reach considering its current velocity. Different works on RCA have been applied to holonomic robots without considerations of the dynamics [18]. Recently, extensions for applying this algorithm in non-holonomic robots are proposed in [19].

The proposed method is based on the Optimal Reciprocal Collision Avoidance (ORCA) [18]. It is based on the work presented in [20] and it improves the RCA behavior by making the vehicles cooperate when performing collision avoidance maneuvers. Even though the method is decentralized, each robot needs to get information of the relative position and the relative velocity of the rest of robots. A centralized method by using ORCA for collision avoidance among multiple agents has been presented in [21].

The work presented in this paper has been carried out in the context of the ARCAS FP7 European project [3] to improve the system presented in [14]. Thus, a decentralized reactive method based on ORCA has been implemented in order to assure that neither the aerial vehicles nor the manipulators or the carried objects collide with each other nor with the static obstacles during the mission. This algorithm computes an optimal solution for the near future, adapting the quick solution computed by the system presented in [14] for unexpected events. Some improvements have been implemented. Most importantly, it considers both mobile and static obstacles in a 3D environment. Static obstacles have to be given in advance to the algorithm by means of a 3D mesh file which can be modified online. Other improvements include the handling of dynamic constraints and ellipsoid agent shapes. The algorithm has been integrated in ROS (Robot Operating System) framework. Realistic simulations have been performed in different scenarios of the ARCAS project with a dynamic quadrotor model based on the implemented in the Hector-quadrotor ROS package [22].

The paper is organized into seven sections. The description of the problem of collision avoidance with multiple aerial vehicles is presented in Section 2. The proposed decentralized reactive method is described in Section 3. Section 4 presents the simulations performed in ROS. Then, the experiments carried out in the multi-UAV testbed in the CATEC facilities are shown and analyzed in Section 5. Finally, the conclusions are detailed in Section 6.

2 Collision avoidance

In multi-UAV missions where several UAVs have to fly in the same area performing their tasks, usually optimized collision-free trajectories for each UAV are previously generated. However, deviations from the prescribed trajectories or unexpected events such as dynamic obstacles or perturbations may cause collisions with other UAVs or with obstacles, like pipes and other devices, present in the industrial environments considered in the ARCAS project. Although dynamic re-planning can be made with some trajectory planners at certain rates, these methods do not allow for a sufficiently fast response time to avoid the collision. In this case, a reactive method should compute a fast solution by ensuring that the separation between the UAVs is greater than a given safety distance. It is assumed that all possible velocity changes are allowed to solve the conflicts. That is, each vehicle can change its heading, airspeed and altitude from the original planned trajectory.

The information that the system needs in order to solve the problem is the following:

1. Initial spatial trajectory of each aerial vehicle, which is described as a dense sequence of waypoints, usually with sample time of $0.01s$.
2. Parameters of the model of each aerial vehicle. They include maximum and minimum velocity and maximum allowed acceleration.
3. Location and velocity of each aerial vehicle in each instant.
4. Description of the static obstacles in the environment by means of a 3D-mesh file.

2.1 Basic coordination problem

Let the system be composed of two robots R_A and R_B, which are located on $\boldsymbol{p}_A$ and $\boldsymbol{p}_B$ and with radius r_A and r_B (see Figure 1(a)). Let $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$ be the velocity of robots A and B, respectively. These robots are on collision course, that is, if none of their velocities is changed a collision will take place before time τ. The

velocity obstacle, $VO^{\tau}_{A|B}$, is the set of relative velocities, $\boldsymbol{v}$, that will lead them to collision before τ for robot A, imposed by robot B.

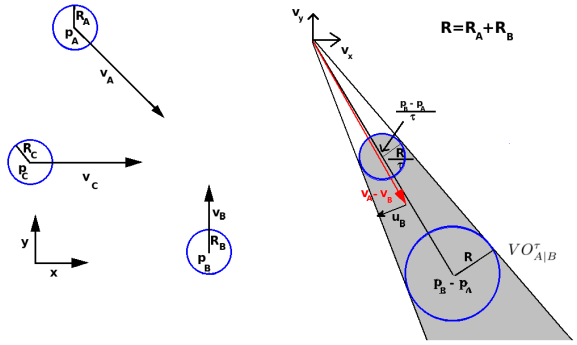

Fig. 1 a) On the left side, a scenario involving three robots (A, B and C) on collision course is represented. b) This scenario leads to $VO^{\tau}_{A|B}$ which is represented on the right side. The minimum reaction robots A and B have to perform in order to avoid their potential collision is represented by $\boldsymbol{u}_B$.

For convenience we will define the following variables and symbols:

$$R = r_A + r_B \tag{1}$$
$$\boldsymbol{p}_r = \boldsymbol{p}_B - \boldsymbol{p}_A \tag{2}$$
$$\boldsymbol{v}_r = \boldsymbol{v}_A - \boldsymbol{v}_B \tag{3}$$
$$D(\boldsymbol{p}, r) = \{\boldsymbol{q} | \|\boldsymbol{q} - \boldsymbol{p}\| < r\} \tag{4}$$

Note that the relative position for obtaining VO^{τ}_{AB} the relative position $\boldsymbol{p}_r$ is obtained by subtracting $\boldsymbol{p}_A$ from $\boldsymbol{p}_B$, this means how far is robot B from robot A. In contrast, the relative velocity $\boldsymbol{v}_r$ is calculated by subtracting $\boldsymbol{v}_A$ from $\boldsymbol{v}_B$; that can be seen as the rate at which robot B is getting closer to robot A. $D(\boldsymbol{p}, r)$ represents an open sphere of radius r centered at $\boldsymbol{p}$.

Then, $VO^{\tau}_{A|B}$, which is the velocity obstacle for A induced by B within time τ, can be defined as (see Figure 1(b)):

$$VO^{\tau}_{A|B} = \{\boldsymbol{v} | \exists t \in [0, \tau] :: t\boldsymbol{v} \in D(\boldsymbol{p}_r, R)\} \tag{5}$$

In order to get a collision-free situation, the relative velocity, $\boldsymbol{v}_r$, should be outside the velocity obstacle $VO^{\tau}_{A|B}$. There are a lot of pairs of sets of allowed velocities $\boldsymbol{v}^{free}_r$ but the pair that minimizes the differences between $\boldsymbol{v}_A$ and $\boldsymbol{v}_B$ with the preferred velocities, $\boldsymbol{v}^{pref}_A$ and $\boldsymbol{v}^{pref}_B$, should be chosen. These preferred velocities are given by the navigation modules of robots A and B, to encourage the minor deviation from the planned trajectories. A reaction takes place when the current velocities and the preferred velocities have to be different.

Let $\boldsymbol{u}_B$ be the vector from $\boldsymbol{v}^{pref}_r$ to the closest point on the boundary of the velocity obstacle (see Figure 1(b)), this represents the minimum reaction that robot A has to perform in order to avoid the potential collision with robot B if this robot does not perform any maneuver. Reciprocally, robot B should perform a reaction $-\boldsymbol{u}_B$ in order to avoid collision if robot A does not perform any maneuver.

As collaborative robots are considered, each one of them can take half of this reaction or this reaction can be divided for each robot as desired. For example, in heterogeneous systems where some vehicles have more maneuverability than others, the reaction should be carried out almost entirely by the first type of vehicles. In general we will define the reaction that robot A has to perform as: $\boldsymbol{u}^{ORCA}_B = \alpha_A \boldsymbol{u}_B$. In this paper, we impose that all the robots have equal reaction to the conflicts so $\alpha = 0.5$.

Once ORCA defines a half-space of collision-free velocities $ORCA^{\tau}_{A|B}$ as the set of velocities:

$$ORCA^{\tau}_{A|B} = \{\boldsymbol{v} | (\boldsymbol{v} - (\boldsymbol{v}_A + 0.5\boldsymbol{u})) \cdot \boldsymbol{u} \geq 0\} \tag{6}$$

Each robot computes the half-spaces of collision-free velocities taking into account the relative position and relative velocity of the rest of robots. Then, the intersection of all half-spaces is computed and a new collision-free velocity is selected that minimizes the next function (see Figure 2):

$$ORCA^{\tau}_A = D(0, v^{max}_A) \bigcap \left(\bigcap_{B \neq A} ORCA^{\tau}_{A|B} \right) \tag{7}$$
$$\boldsymbol{v}^{ORCA}_A = \min_{\boldsymbol{v} \in ORCA^{\tau}_A} \|\boldsymbol{v} - \boldsymbol{v}^{pref}_A\| \tag{8}$$

where v^{max}_A is the maximum velocity for robot A. This problem can be solved by using quadratic programming (QP).

Note that, in some densely packaged situations, this problem may become unfeasible. In these cases, a new problem relaxing the conditions of ORCA by decreasing the time τ in which the collision is ensured. Then, τ becomes a new variable in the optimization and the problem is to obtain the velocity that gives maximum τ (ensures collision free trajectories for maximum time, so is the safest velocity). Note that in this case, the optimization objective is not to be as close as possible to the preferred velocity, but rather be as safe as possible. For more details, please refer to [18].

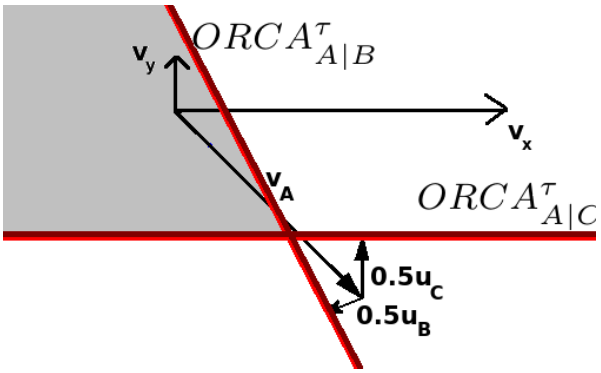

Fig. 2 The ORCA half-planes $ORCA^{\tau}_{A|B}$ and $ORCA^{\tau}_{A|C}$ that robots B, C induce in robot A are represented. The region of allowed velocities robot A can take is given by the intersection of these half-planes. This region is filled in light gray.

3 Decentralized Reactive Method

The decentralized reactive method proposed in this paper is based on ORCA. It is a good candidate to efficiently carry out the coordinated mission of the ARCAS project. The experimental scenario of the project is the multi-UAV aerial testbed of the Center for Advanced Aerospace Technologies (CATEC) which is equipped with a VICON localization system that provides estimation of the position of the robots with a precision of few millimeters in real-time. Therefore, all the robots can obtain information of the position of the rest of robots and the minimum separation distance could be defined as a sphere around of the robot because the uncertainties are not relevant.

However, several improvements to the ORCA algorithm are necessary in order to adapt them to the realistic environments as the ones proposed in the ARCAS project.

3.1 Dynamics constraints handling

The proposed method considers dynamics constraints. We use a similar approach as the one proposed in [23]. Another constraint is added considering the current velocity of the robot $\boldsymbol{v}(t)$ and the maximum acceleration $\boldsymbol{a}_{max}$. Let T_s be the sample rate of the algorithm, then the inequation relating $\boldsymbol{v}^{ORCA}$ and $\boldsymbol{v}(t)$ is given by:

$$\|\boldsymbol{v}^{ORCA} - \boldsymbol{v}(t)\| \leq a_{max} T_s \tag{9}$$

This will force the velocity given by ORCA module to be reachable by the controller on-board the quadrotors.

3.2 Considering 3D obstacles

Another important requirement of the project is that the navigation is performed in scenarios with the presence of static obstacles.

A 3D-map of the environment is assumed to be known. It can be loaded into the proposed method as a set of mesh files, which can be specified in any format compatible with the assimp library [1]. Note that in real scenarios unexpected or unmodeled obstacles might appear. For this reason, this information could be enriched with the inclusion of vision or range sensors in order to detect them. However, this is beyond the scope of this paper.

The Proximity Query Package [24] has been used in order to calculate the distance between the position of the aerial robot and the static obstacles. This library not only checks for collision between two 3D meshes with triangular faces, but also returns the distance vector between these meshes, $\boldsymbol{d}$.

When applying the algorithm in a determinate time-step, only each obstacle's closest point to the agent is considered. This is done for two main reasons: the first is to decrease computational load and the second is to not over-constrain the QP problem. Once this closest point to an obstacle is calculated, its velocity obstacle is calculated by only considering this closest point. In consecutive computations this point seems to be moving slowly (see Figure 3), allowing the algorithm to smoothly react to the shape of the obstacle. Besides, it is a natural approach that resembles the behavior of humans when piloting a vehicle in a scenario with complex obstacles.

However, concave obstacles can make the obstacle closest point to the quadrotor to jump between different parts of the obstacle as the quadrotor moves, and these discontinuities may mislead the algorithm (see Figure 4). For this reason, we have to include only convex obstacles in the 3D meshes file that represents the static obstacles. There exist many methods for splitting concave obstacles into multiple convex obstacles, such as the Hierarchical Approximate Convex Decomposition[25]. They can be applied offline in a preprocessing step to force the meshes to be composed of convex obstacles.

Note that the environment in the proposed application is dynamic, that is: new obstacles can be added whenever they are detected by the sensing system. In our implementation, each different obstacle is saved as an independent 3D-model. Actually each convex part of each obstacle is saved as an independent 3D-model. These models can be deleted or added online.

[1] Open Asset Import Library. `http://www.assimp.org/`

Figure 3 represents an obstacle, O, and a robot, R_A, which lies in position p_A and has radius r_A. The velocity obstacle is a cone constructed by the union of the position of the robot and the closest point from the robot to the obstacle. In a similar development as indicated in [18], the velocity obstacle can be defined as:

$$VO^{\tau}_{A|O} = \{\boldsymbol{v} | \exists t \in (0, \tau] :: t\boldsymbol{D}(\boldsymbol{p}_O, r_A)\} \quad (10)$$

where $\boldsymbol{p}_O$ is the closest point from the robot A to the obstacle. Once the velocity obstacle has been constructed, the minimum reaction can be calculated as indicated in section 2. Then, the ORCA half-plane is obtained taking into account that the robot should perform the whole reaction. This is indicated in the next equation.

$$ORCA^{\tau}_{A|O} = \{\boldsymbol{v} | (\boldsymbol{v} - (\boldsymbol{v}_A + \boldsymbol{u})) \cdot \boldsymbol{u} \geq 0\} \quad (11)$$

The last consideration is that the constraints due to static obstacles are not relaxed when an unfeasible problem is detected [18].

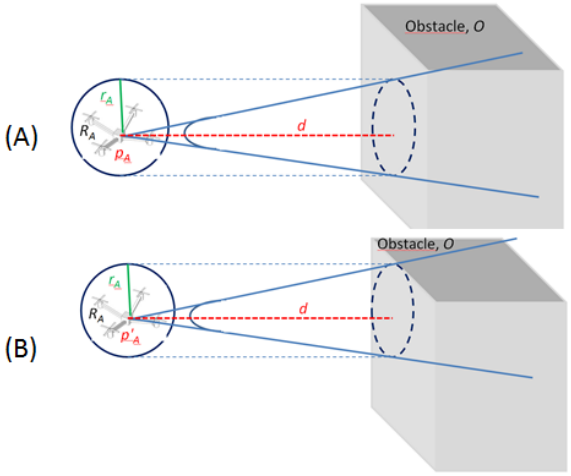

Fig. 3 Velocity obstacles induced by obstacle O to an agent in two different instants.

3.3 Safety region

The original 3D-ORCA algorithm assumes that robots have spherical shapes. However, the shapes may vary among robots. For example, the shapes of the quadrotors that will be used in simulation is not covered uniformly with a sphere. In this case, the minimum horizontal separation distance should be greater than the vertical, which is better approximated by an ellipsoid

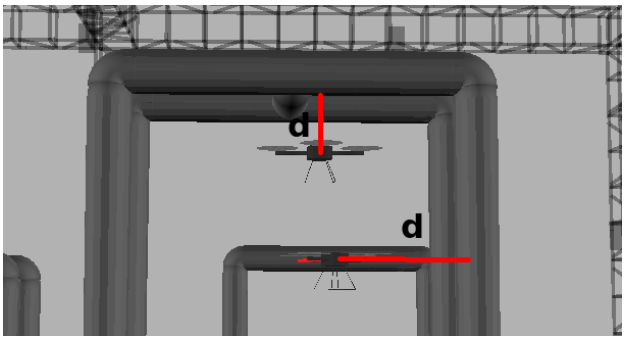

Fig. 4 Concave obstacles can break the time continuity of the distance vector. They have to be decomposed into several convex obstacles in a preprocessing step. d is the horizontal or vertical separation between a quadrotor and the closer static obstacle.

with the vertical semi-axis (r_z) smaller than the other two (r_{xy}). Therefore, a simple coordinate transform to the distances between robots and between robots and static obstacles is applied.

$$x' \leftarrow x \quad (12)$$
$$y' \leftarrow x \quad (13)$$
$$z' \leftarrow \alpha z \quad (14)$$

where $\alpha = \frac{r_{xy}}{r_z}$. These considerations will allow the quadrotor to get closer while performing vertical collision avoidance maneuver than when performing horizontal maneuvers. Note that the separation distances between robots can easily be obtained:

$$E_{xy} = 2r_{xy} \quad (15)$$
$$E_z = 2r_z \quad (16)$$

Last but not least, the dimensions of the safety region can be reconfigured in real time in order to model the aerial robot shape with the arm extended and contracted.

4 Simulations

Many simulations have been carried out in a realistic environment that simulates the multi-UAV testbed of the CATEC where the experiments will be carried out. The ROS framework is used to test the proposed method. The dimensions of the scenario are $15x15x4m^3$.

The proposed algorithm has been run in a laptop with an Intel^{TM}Core i5 processor and 4 GB of RAM. The operating system used was Kubuntu 12.04 Linux. The code was written in C++ language and integrated

with the ROS fuerte distribution. The dynamic quadrotor model used is based on the Hector-quadrotor ROS package [22].

The testbed of CATEC is integrated with ROS and the same ROS node architecture is used for both simulation and real experimentation. This diminishes the possible faults, reducing the efforts transition between simulation and real experimentation.

One scenario with static obstacles with up to eight quadrotors is considered (see Figure 5). The simulations have been performed in the same machine. The videos show the collision avoidance in real time. The videos of the different experiments are available at `http://www.youtube.com/0grvc0`.

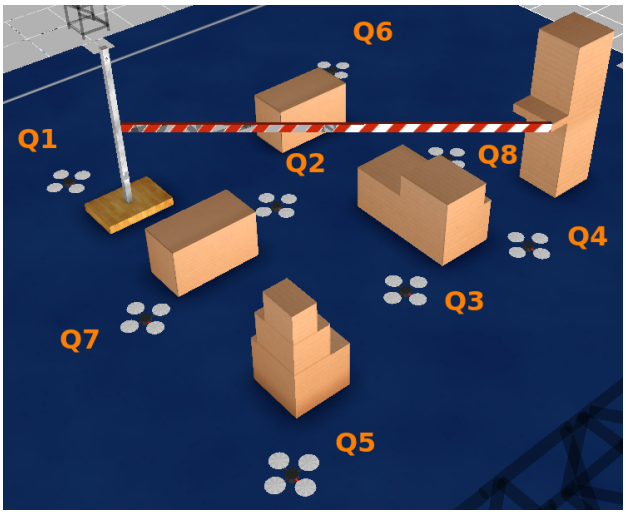

Fig. 5 Simulation scenario with up to eight quadrotors and static obstacles.

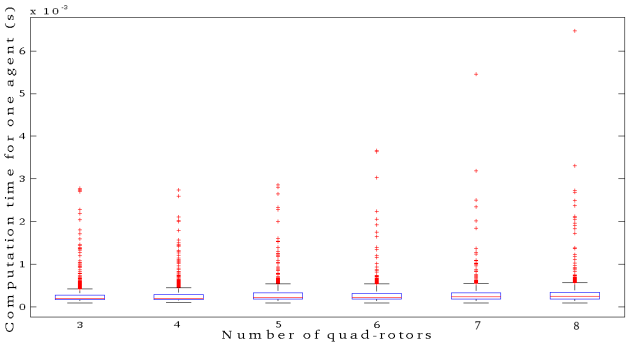

Fig. 6 Distribution of the execution computation time in proposed algorithm for one agent with the number of vehicles in the system. The median of each distribution is indicated in red, the blue box represent the 25th and 75th percentiles and the 3rd and 97th percentiles are indicated in black. Red marks represent the outliers.

4.1 Scalability

In this section we will study the scalability of the proposed method. Figure 6 shows the distribution of the computation time for calculating the collision-free velocity for one agent in the execution of simulations from 3 to 8 aerial vehicles shown in Figure 5. Note that each agent only takes into account the agents that are closer than the neighbouring distance. In this case, this distance was set to $4m$. Also, the preprocessing step is done offline, so its execution time has not been taken into account because it does not affect the real-time performance of the system.

These results show that the computation time in calculating the ORCA velocity for each agent was far below $1ms$ in more than the 97% of the cases. Moreover, the computation time grows very slowly with number of vehicles: it was confined between 0.3 and $0.5ms$ in the case of 3 vehicles and between 0.4 and $0.6ms$ with 8 vehicles.

The Collision Avoidance method was computing velocities at a rate of $20Hz$ for each quadrotor, which is one for each $50ms$. This has allowed us to perform simulations with up 8 quadrotors in real time and in the same machine. Taking into account these results, it is expected that we can raise this number to more than 50 without experiencing flaws. Furthermore, the computations can be easily distributed among several PCs thanks to the ROS integration. Thanks to this fact, there exist no theoretical limits in the number of robots this method can handle when considering the computational efforts.

4.2 Stability and reliability

Finally, a ten minutes long simulation in order to test the stability and reliability of the proposed method has been carried out with three vehicles (Q1, Q2 and Q3 in Figure 5). The main results of the simulation are detailed in table 1. The minimum separation distance between the vehicles and between pairs the vehicles and the scenario is listed. These minimum separation distances fulfill the requirements imposed in the first part of this section 4 demonstrating that even in long simulations the systems keeps being stable.

The number of collision-avoidance maneuvers that the developed algorithm has performed during the simulation has been also listed. We consider that a vehicle is performing a maneuver whenever its desired velocity

and the ORCA velocity are distant enough. Mathematically this can be expressed as:

$$\|\boldsymbol{v}^{pref} - \boldsymbol{v}^{ORCA}\| \geq v_{thres} \quad (17)$$

where v_{thres} is set to $0.1\frac{m}{s}$. In this simulation, forty-one maneuvers involving from one to three quadrotors have been performed. The mean time a maneuver has lasted is $6.56s$.

Table 1 Results obtained in the reliability simulation.

Characteristics	Quadrotors	Value
# maneuvers	-	41
Avg Duration	-	$6.56s$
Avg Vehicles	-	1.75
Separation w. obstacles	QR1	0.60
	QR2	$0.62m$
	QR3	$0.53m$
Vertical separation	QR1-QR2	$0.77m$
	QR1-QR3	$0.84m$
	QR2-QR3	$0.75m$

5 Experimental Results

Three multi-UAV coordination experiments have been carried out in the multi-UAV testbed of the CATEC facilities. It has an usable volume of $10x10x3m$. The experiments are described and analyzed in this section.

Figure 7 shows one of the four autonomous quadrotors that have been used in the experiments. They are Hummingbird quadrotors from Ascending Technologies. The testbed is equipped with a VICON localization system which includes 20 infrared cameras distributed around the testbed. This system is capable of estimating the position and orientation of each quadrotor at a rate of $100Hz$ with millimeter and degree accuracy, respectively.

Experiment I tests the method in a simple scenario configuration. It involves two quadrotors which will fly trajectories to interchange their positions as shown in Figure 8. Static obstacles are not considered in the environment, so only maneuvers to avoid collision with other aerial robots are performed.

Table 2 shows the parameters considered in Experiment I: frequency (algorithm's rate of execution), minimum horizontal and vertical separation distances amongst robots (E_{xy} and E_z, respectively), and Time horizon (T), that represents the look-ahead time that each aerial robot considers to detect potential collisions with other robots). Obviously, the parameters related to static obstacles, Horizontal and Vertical Separation to Obstacles and Time Horizon Obstacles, have no effect in Experiment I.

Fig. 7 One of the Hummingbird quadrotors used in the experiments and one of the infrared cameras of the VICON system.

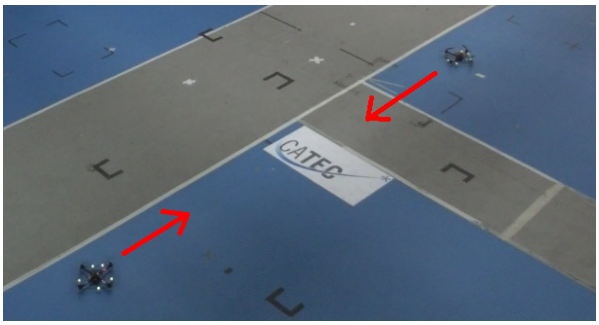

Fig. 8 Initial plans in Experiment I.

Table 2 Parameters of the proposed method in Experiments I, II and III.

Parameter	Value
Frequency	$20Hz$
Minimum Horizontal Separation (E_{xy})	$1.1m$
Minimum Vertical Separation (E_z)	$0.6m$
Time Horizon (T)	$8s$
Horizontal Separation to Obstacles (d_{xy}^{obs})	$0.9m$
Vertical Separation to Obstacles (d_z^{obs})	$0.6m$
Time Horizon Obstacles (T^{obs})	$2s$

Figure 9 shows the horizontal and vertical separation among the quadrotors during the flight. The vertical and horizontal separations were met during the whole experiment because the minimum values were not surpassed at the same time. Therefore, the flown trajectories were safe. Note that the vertical separation is always violated, while the horizontal separation is not, so the vehicles have performed a lateral maneuver

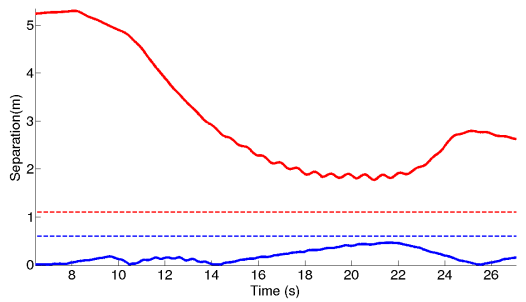

Fig. 9 Horizontal (red line) and vertical (blue line) separation between the aerial robots during Experiment 1. The minimum separation distances are shown in dashed line.

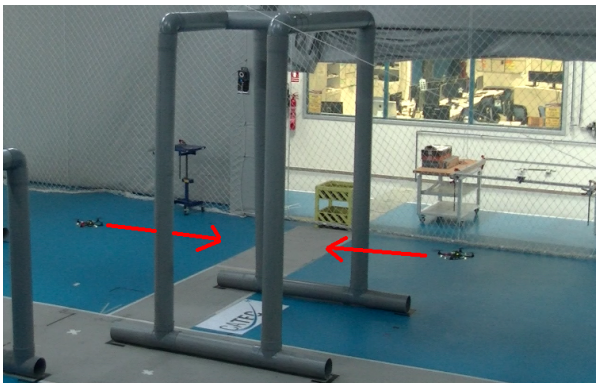

Fig. 10 Initial plans in Experiment II.

in order to avoid the collisions. Note that the horizontal separation plot is somewhat curly, this is a not desirable effect that where not found in simulation and that could be produced by some delay between the acquisition of the location of the VICON system and the reception of the new generated commands.

Experiment II is similar to Experiment I but with static obstacles. Figure 10 represents the initial trajectories of the quadrotors as well as the static obstacles of the scenario. In this case, the quadrotors are forced to perform different maneuvers in order to avoid collisions also with the static obstacles. A video of the experiment can be found at `https://youtu.be/pDLwemOi940`.

The configuration parameters of the static obstacles are considered in the Experiment II (see Table 2). Note that the minimum horizontal separation to obstacles is smaller than the minimum separation between quadrotors. This can be done because we are considering on the one hand the closest point from the static obstacle to the robot and on the other hand the center of the quadrotors. It is also remarkable that the time horizon in collision between quadrotors is much greater than the time horizon in collisions between a quadrotor and the static obstacles. Thus, the quadrotor maneuvers to avoid collisions with static obstacles take place only when they are sufficiently close to the obstacles, as pointed out in [18]. This parameter has to be carefully tuned taking into account the maximum allowed acceleration a_{max} in order to assure that no collisions with static obstacles can be produced.

Figure 11 shows the vertical and horizontal separations between the quadrotors during the flight. The horizontal or the vertical separation are met during the whole flight. In contrast to Experiment I, there are time instants where the horizontal separation is not met but the vertical is. This indicates that the quadrotors have performed a vertical collision avoidance maneuver. This type of maneuver was imposed taking into account the scenario where the quadrotors were located. In this experiment, no noticeable oscillations were found in the behavior of the quadrotors.

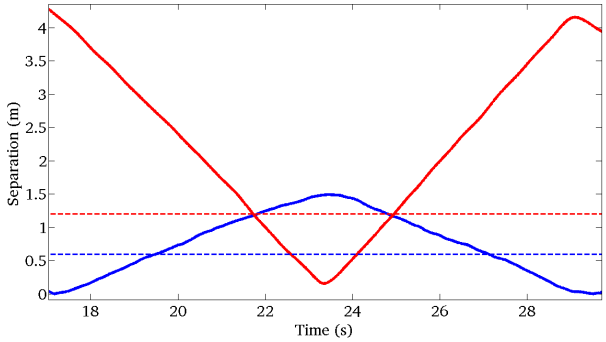

Fig. 11 Horizontal (red line) and vertical (blue line) separation between the aerial robots during Experiment II. The minimum separation distances are shown in dashed line.

Finally, Experiment III proposes a scenario with four quadrotors and several static obstacles. This scenario have tested the proposed method in the presence of more quadrotors and static obstacles. Thus, more complex maneuvers have to be performed. Figure 12 represents the initial trajectories of each quadrotor. The execution of the plan of quadrotor Q4 is delayed by approximately fifteen seconds with respect of the execution of quadrotors Q1-Q3.

The separations between pairs of quadrotors during the execution of the experiment are plotted in Figure 13. The trajectories are safe because the horizontal or vertical separation is met during the whole flight. However, the same oscillations that were found in Experiment I are found. Furthermore, in some situations were conflicts with more than two quadrotors were detected, the system evolved to an almost deadlock situation that lasted for almost ten seconds in some cases (see instants from 70s to 90s in separation between Q1-Q2, Q1-Q3

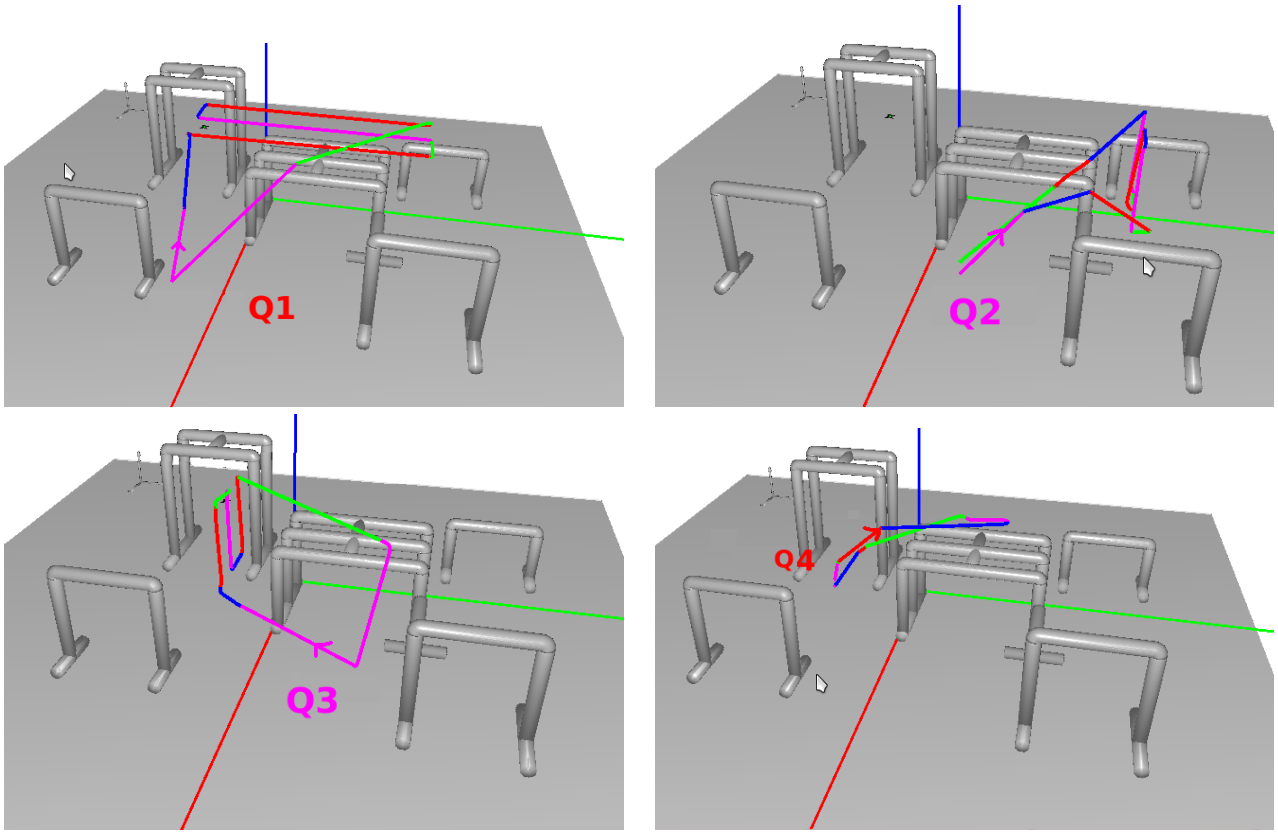

Fig. 12 Initial plans of Experiment III.

and Q2-Q3). Finally, some minor separation violations were found in some instants in the deadlocks (see instants from 60s to 80 s in separation between Q2-Q3) and without deadlocks (instants in the surroundings of 85s in separation between Q3-Q4). This situation, although brief, is not desirable for collision avoidance systems.

A video summary of the experiment can be found at `https://youtu.be/9NEGLpva4eg`.

The experiments performed were safe during the execution of the initial plans. However, the behavior of the proposed method when integrated into the real system is still far from the one desired and the one obtained in simulation. Some oscillations in roll were found in Experiments I and III. In addition, there were some states close to deadlocks at the end of Experiment III in which the quadrotors, although being static in their translational position, were oscillating in their roll angles. In fact, some of these oscillations did imply slight violations in the minimum allowed separations.

From the analysis of the experiments, this behavior is mostly generated by communication delay, which was not modeled in simulation and thus not taken into account when designing the collision avoidance system. The total communications delay in the testbed experiments can be estimated as the sum of the delay from the VICON to the ORCA ROS node and the delay from the ORCA ROS node to the autopilot onboard the quadrotors. The first delay includes the delay due to the VICON processing, the communications between VICON and the ROS node and the delay due to the ROS middleware. The second delay includes the delay on processing the new velocity, the delay due to the ROS middleware and sending the commands to the quadrotor's autopilot via zigbee. In consequence, future efforts need to be performed in order to model the total delay and to estimate future states of the system taking the delay into account, in order to use the system in the ARCAS testbed. However, as the developed collision avoidance system is intended to be executed onboard each UAV in a distributed manner, and also each UAV will use its own sensors for positioning, in real applications this delay will be much lower and these effects will be much alleviated.

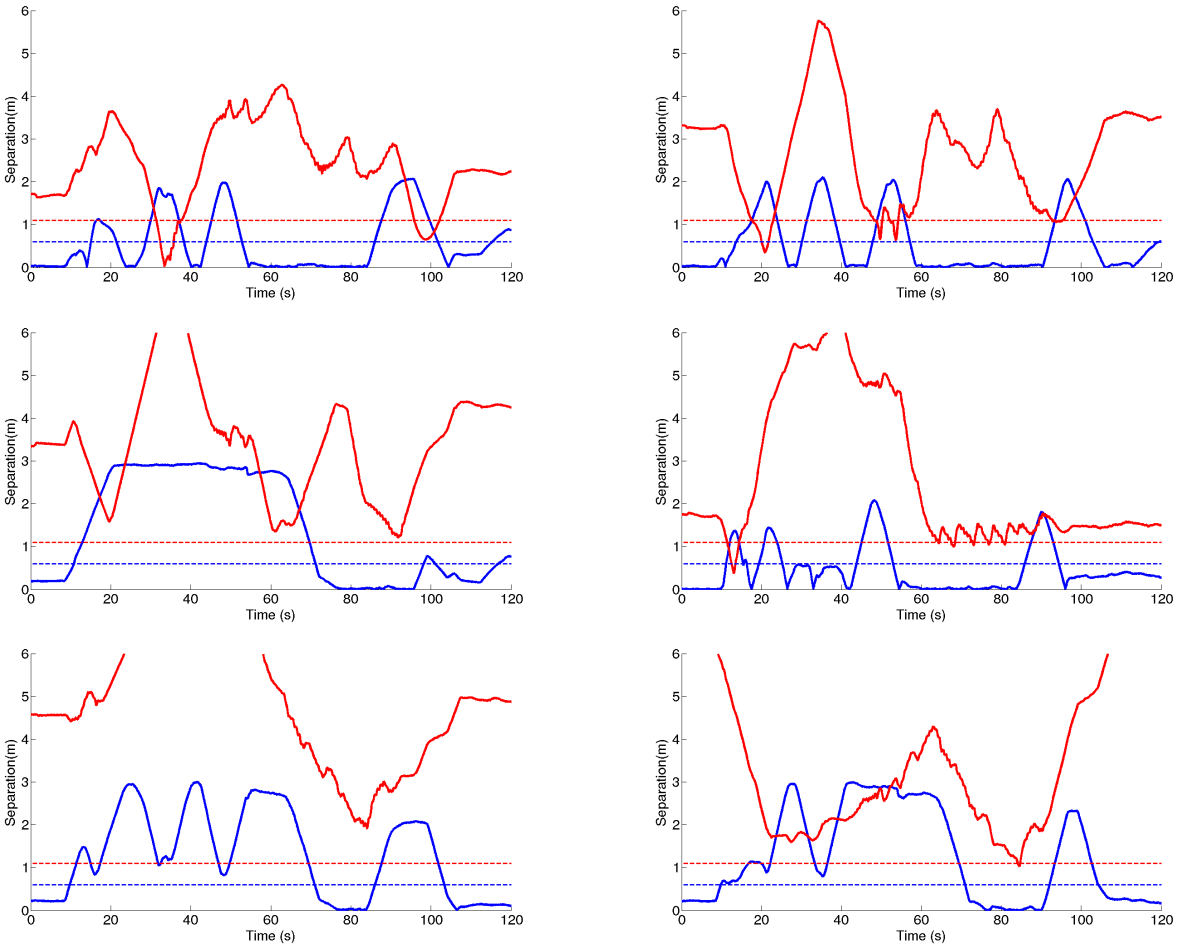

Fig. 13 Horizontal (red line) and vertical (blue line) separation between quadrotors during Experiment III. The minimum separation distances are shown in dashed line. These plots, from left to right and top to bottom represent the separations of quadrotors Q1-Q2, Q1-Q3, Q1-Q4, Q2-Q3, Q2-Q4, Q3-Q4.

6 Conclusions

A new real-time collision avoidance method based on the 3D-ORCA algorithm has been presented in this paper. The main contribution of the proposed algorithm with respect to the works published [18][26] [19] is that it considers complex 3D static obstacles. Also, the dynamics of the quadrotors has been considered and a new shape of the agents has been proposed. This was necessary in order to apply it in realistic environment as the proposed in ARCAS project [3]. Several simulations demonstrate both the safety and reliability of the method. Results show that more than 99% of the calculations were carried out in less than a millisecond. This shows that the proposed method can be run in real time in the same computer even in simulations with great number of aerial vehicles. On the other hand, due to the distributed nature of the algorithm, this would allow to apply the proposed algorithm onboard a UAV equipped with an unexpensive computer.

Moreover, the algorithm has been integrated in ROS framework with the same ROS node architecture used in the multi-UAV testbed of CATEC. This has allowed us to perform real experiments in the CATEC multi-UAV testbed with up to four quadrotors where the trajectories were modified in real time in order to perform collision avoidance among the quadrotors while not colliding with static obstacles. These tests are a remarkable contribution with respect to the work presented in [27]. However, some undesired behaviors appeared, mainly related with communication delays in the experimental setting. In any case, these effects would be much alleviated in real applications, when the collision avoidance algorithms are implemented onboard the UAVs and the onboard sensors are used for attitude and position estimation.

Future efforts will include modeling the communication delay of the testbed experimental setting. In this way, it can be simulated in order to reproduce the oscillations found in the experiments, so they can be corrected for the testbed experiments. Other aspects to be studied include taking into account non-holonomic constraints in order to apply this algorithm to fixed-wing UAVs.

Last but not least, the addition of cameras and/or range sensors to the UAV should be necessary in order to detect unmodeled obstacles in the environment. Furthermore, their measures can be used for obtaining the relative position between agents. One of the main difficulties in this case is to distinguish between static obstacles and cooperating agents. Some markers could be installed in the vehicles so they can be identified by cooperating agents. The uncertainties related to the measures should be taken into account. One conservative approach could be the expansion of the safety envelope of the vehicles taking into account these uncertainties.

Acknowledgement

This work was supported by the European Commission FP7 ICT Programme under the project ARCAS 287617 and the RANCOM project (P11-TIC-7066), which has been funded by the Junta de Andalucia(Spain). David Alejo is granted with a FPU Spanish fellowship from the Spanish Ministerio de Educación, Cultura y Deporte. Authors would like to thank M. A. Trujillo, J. Ruiz and Y. Rodríguez of the Center for Advanced Aerospace Technologies (CATEC) for their support in the experimental work and for the development of the simulation and experimentation platforms.

References

1. L. Merino, F. Caballero, J. M. de Dios, I. Maza, and A. Ollero, "An unmanned aircraft system for automatic forest fire monitoring and measurement," *Journal of Intelligent and Robotic Systems*, vol. 65, no. 1, pp. 533–548, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10846-011-9560-x
2. R. W. Beard, T. McLain, D. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature UAVs ," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1306–1324, 2006.
3. A. Ollero, "Aerial robotics cooperative assembly system (ARCAS): First results," in *Aerial Physically Acting Robots (AIRPHARO) workshop, IROS 2012*, Vilamoura, Portugal, October 7-12 2012.
4. A. E. Jimenez-Cano, J. Martin, G. Heredia, R. Cano, and A. Ollero, "Control of an aerial robot with multi-link arm for assembly tasks," in *IEEE Int. Conf. Robotics and Automation (ICRA)*, Karlsruhe, Germany, MAy 6-10 2013.
5. S. M. Lavalle, J. J. Kuffner, and Jr., "Rapidly-Exploring Random Trees: Progress and Prospects," in *Algorithmic and Computational Robotics: New Directions*, 2000, pp. 293–308.
6. J. A. Cobano, R. Conde, D. Alejo, and A. Ollero, "Path planning based on genetic algorithms and the monte-carlo method to avoid aerial vehicle collisions under uncertainties," in *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, 2011, pp. 4429–4434.
7. R. Vivona, D. Karr, and D.Roscoe, "Pattern-based genetic algorithm for airborne conflict resolution," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, Keystone, Colorado, August 2006.
8. A. Stentz and I. C. Mellon, "Optimal and Efficient Path Planning for Unknown and Dynamic Environments," *International Journal of Robotics and Automation*, vol. 10, pp. 89–100, 1993.
9. M. Pontani and B. A. Conway, "Particle Swarm Optimization Applied to Space Trajectories," *Journal of Guidance Control and Dynamics*, vol. 33, pp. 1429–1441, 2010.
10. A. Vela, S. Solak, W. Singhose, and J.-P. Clarke, "A mixed integer program for flight-level assignment and speed control for conflict resolution," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, dec. 2009, pp. 5219 –5226.
11. I. Hwang and C. J. Tomlin, "C.: Protocol-based conflict resolution for air traffic control. air traffic control quarterly 15(1," 2007.
12. N. Durand and J. Alliot, "Ant colony optimization for air traffic conflict resolution," in *Proceedings of the Eighth USA/Europe Air Traffic Management Research and Development Seminar (ATM2009)*, Napa, (CA, USA), 2009.
13. J. A. Cobano, D. Alejo, A. Ollero, and A. Viguria, "Efficient conflict resolution method in air traffic management based on the speed assignment," in *Proceedings of the 2nd International Conference on Application and Theory of Automation in Command and Control Systems*, ser. ATACCS '12. Toulouse, France, France: IRIT Press, 2012, pp. 54–61. [Online]. Available: http://dl.acm.org/citation.cfm?id=2325676.2325684
14. D. Alejo, J. A. Cobano, G. Heredia, and A. Ollero, "Collision-free 4D trajectory planning in Unmanned Aerial Vehicles for assembly and structure construction," *Journal of Intelligent and Robotic Systems*, vol. 73, pp. 783–795, 2014.
15. E. Lalish and K. A. Morgansen, in *Proceedings of the 47nd IEEE Conference on Decisionn and Control*, Cancun, Mexico.
16. G. Roussos, G. Chaloulos, K. Kyriakopoulos, and J. Lygeros, in *Proceedings of the 47nd IEEE Conference on Decisionn and Control*, Cancun, Mexico.
17. J. P. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation." in *ICRA*. IEEE, 2008, pp. 1928–1935. [Online]. Available: http://dblp.uni-trier.de/db/conf/icra/icra2008.html#BergLM08
18. J. van den Berg, S. J. Guy, M. C. Lin, and D. Manocha, "Reciprocal n-body Collision Avoidance," in *INTERNATIONAL SYMPOSIUM ON ROBOTICS RESEARCH*, 2009.
19. J. Alonso-Mora, A. Breitenmoser, P. Beardsley, and P. Siegwart, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Saint Paul, MN, USA.

20. P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments using Velocity Obstacles," *International Journal of Robotics Research*, vol. 17, pp. 760–772, 1998.
21. J. Alonso-Mora, M. Rufli, P. Siegwart, and P. Beardsley, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany.
22. J. Meyer, "Hector quadrotor ros package website," 2014, [Accessed 5-February-2014]. [Online]. Available: http://wiki.ros.org/hector_quadrotor
23. J. van den Berg, J. Snape, S. Guy, and D. Manocha, "Reciprocal collision avoidance with acceleration-velocity obstacles," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May 2011, pp. 3475–3482.
24. M. C. L. Eric Larsen, Stefan Gottschalk and D. Manocha., "Proximity query package website," 2014, [Accessed 5-February-2014]. [Online]. Available: http://gamma.cs.unc.edu/SSV/
25. M. Ghosh, N. M. Amato, Y. Lu, and J.-M. Lien, "Fast approximate convex decomposition using relative concavity," *Computer-Aided Design*, in press 2012, also appear in Proc. of Symposium on Solid and Physical Modeling, Dijon, France, Oct. 2012.
26. J. Alonso-Mora, A. Breitenmoser, M. Rufli, P. Beardsley, and R. Siegwart, "Optimal Reciprocal Collision Avoidance for Multiple Non-Holonomic Robots," in *Proc. of the 10th International Symposium on Distributed Autonomous Robotic Systems (DARS)*, A. Martinoli and F. Mondada, Eds. Berlin: Springer Press, November 2010.
27. D. Alejo, J. Cobano, G. Heredia, and A. Ollero, "Optimal reciprocal collision avoidance with mobile and static obstacles for multi-uav systems," in *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*, May 2014, pp. 1259–1266.