Collision Avoidance with Limited Field of View Sensing: A Velocity Obstacle Approach

Steven Roelofsen[1,2], Denis Gillet[2] and Alcherio Martinoli[1]

Abstract— **Collision avoidance, in particular between robots, is an important component for autonomous robots. It is a necessary component in numerous applications such as human-robot interaction, automotive or unmanned aerial vehicles. While many collision avoidance algorithms take into account actuation constraints, only a few consider sensing limitations. In this paper, we present a reciprocal collision avoidance algorithm based on the velocity obstacle approach that guarantees collision-free maneuvers even when the robots are only capable to sense their environment within a limited Field Of View (FOV). We also present the challenges associated to sensors with limited FOV, show the conditions under which maneuvering can be safely done, and the modifications that a velocity obstacle approach requires to satisfy such conditions. We provide simulations and real robot experiments to validate our approach.**

I. Introduction

Collision avoidance is of importance in robotics; many robotic applications rely, at least partially, on collision avoidance to safely navigate an environment. Examples of robotic systems that require a collision avoidance of some sort can be found in human-robot interaction [1], automotive [2] or unmanned aerial vehicles [3]. Over the years, several collision avoidance paradigms have emerged such as, for example, potential fields [4], the dynamic window approach [1], or the vector field histogram [5].

A successful approach to collision avoidance is based on the concept of Velocity Obstacle (VO), resulting in avoidance maneuvers that are efficient and proven safe [6], and is applicable to a large variety of systems [7], [8]. VO-based methods have the advantage to generate well-behaved, efficient maneuvers while keeping the computational cost low. Those methods have been successfully used on different robotic platforms [9], [10].

However, up to date the focus of collision avoidance algorithms has been on mitigating constraints on the actuation or improving the resulting trajectory, assuming that each robot is able to sense every other robot. The effect of a robot's Field Of View (FOV) on maneuvering is not well studied. A few notable exceptions are as follows. A study investigates the link between a small FOV and the probability of collision [11]. Other works are able to deal with a limited FOV [12],

[13], usually as a consequence of the control scheme rather than a design requirement.

In this paper, we present our approach to perform collision avoidance within the VO paradigm when the robots are only able to detect other robots in a limited FOV. We previously designed a collision avoidance algorithm based on potential fields that takes into account the FOV of the sensor [14], but the algorithm shows unwanted spinning behavior [15], [16]. By moving to a VO-based approach, we seek to produce more effective trajectories and, in particular, eliminate the spinning issue.

The contribution of this paper is three-fold. First we show the importance of FOV and the possible consequences when ignoring the constraints that a limited field of view imposes. Second, we formulate the conditions under which safe maneuvering can be performed with sensors that have limited FOV. Third, we present our Sensor-Aware Velocity Obstacle (SAVO) algorithm that guarantees no collision under limited FOV constraints. We leverage work found in the VO literature (e.g., [6]) as foundation of our work, and borrow many of the ideas throughout this paper.

The paper is organized as follows: Section I introduces the subject and presents related works. Section II presents the problem and describes the system of interest. Section III presents the relation between the FOV of a robot and its allowed motion. Section IV presents the algorithms. Results are presented in Section V. The paper is concluded in Section VI.

II. Problem Statement

In this paper we consider a set of N robots in the plane $\mathbb{R}^2$. A robot H has a position $\mathbf{p}_H$, a velocity $\mathbf{v}_H$ and a heading angle θ_H. In this paper, all angles are defined between $-\pi$ and π. The heading direction is defined as $\mathbf{d}_H = [\cos(\theta_H), \sin(\theta_H)]^T$. The robots are assumed to have a disk shape with radius r_H. The actuation is constrained by a maximum speed $||\mathbf{v}_H|| \leq v_H^{max}$, and a maximum speed change Δv_H^{max} that the robot can perform during a time step Δt. The robot is also assumed to be able to come to a full stop. The set of allowed velocities can be summarized by the following actuation set:

$$U_H = D(\mathbf{v}_H, \Delta v_H^{max}) \cap D(0, v_H^{max}) \tag{1}$$

with $D(\mathbf{c}, r)$ the closed disk of radius r centered at $\mathbf{c}$. The robot also has a maximum turning rate $u \leq \dot{\theta}_H^{max}$.

A robot H is able to detect another robot if it is in the FOV F_H of its own sensor:

$$F_H = \left\{ \mathbf{p} \mid ||\mathbf{p} - \mathbf{p}_H|| < r_H^s, |\angle(\mathbf{d}_H, \mathbf{p} - \mathbf{p}_H)| \leq \alpha_H^s \right\} \tag{2}$$

with r_H^s the sensor range and $\pi/2 < \alpha_H^s < \pi$ sensor's FOV half-angle. A robot H is able to estimate the position, velocity, and radius of a detected robot B if it is in the sensor's range and FOV of robot H. It is also able to identify the type of robot, and knows the actuation set U_B of robot B (or at least an estimation $\hat{U}_B$ such that $U_B \subset \hat{U}_B$).

The problem of interest is for a robot H to avoid collision with other robots despite the limited FOV of its sensing capability, while keeping its velocity $\mathbf{v}_H$ as close as possible to its prefered velocity $\mathbf{v}_H^{pref}$. State of the art VO-based algorithm will fail to avoid in some situation. This is because the sensor's limited FOV is in contradiction with the assumption that every robot has at least a noisy estimate of other robots' states. For scenarios illustrating the problem, see Section V.

III. Collisions and Field of View

The main concept of collision avoidance under FOV sensing constraints is to limit the motion of the robots in order to guarantee that they cannot collide with each other because of their intrinsic blind zones. In this paper we propose that the velocity remains in the following set, called the sensor-constraint set:

$$S_H = \left\{ \mathbf{v} \mid |\angle(\mathbf{v}, \mathbf{d}_H)| \leq \alpha_H^s - \frac{\pi}{2} \right\}. \tag{3}$$

We will now prove that constraining the velocity within the set defined by Equation 3 will be sufficient to ensure safe maneuvering. First, we show that robots that do not see each other will not get closer if they choose their velocity in their respective sensor-constraint sets.

Lemma 1: A robot A that chooses $\mathbf{v}_A \in S_A$ will not go towards the position of another robot B that robot A is unable to detect.

Proof: First note that because robot A does not detect robot B we have that $|\angle(\mathbf{d}_A, \mathbf{p}_B - \mathbf{p}_A)| > \alpha_H^s$. Define a line $\Gamma \perp (\mathbf{p}_B - \mathbf{p}_A)$ that passes somewhere between robots A and B, with a distance of at least r_A and r_B from robot A and B, respectively. Because $\mathbf{v}_A \in S_A$, it is possible to find a lower bound for the angle between $\mathbf{v}_A$ and the vector $\mathbf{p}_B - \mathbf{p}_A$:

$$|\angle(\mathbf{d}_A, \mathbf{p}_B - \mathbf{p}_A) + \angle(\mathbf{v}_A, \mathbf{d}_A)| \geq \tag{4}$$
$$|\angle(\mathbf{d}_A, \mathbf{p}_B - \mathbf{p}_A)| - |\angle(\mathbf{v}_A, \mathbf{d}_A)| > \alpha_H^s - (\alpha_H^s - \frac{\pi}{2}) = \frac{\pi}{2}$$

As a result the velocity $\mathbf{v}_A$ points away from the line Γ and moves away from the position of robot B. ∎

Using Lemma 1, we will now prove there is no collision possible between two robots that do not see each other. Figure 1 summarizes the principle behind the following proposition:

Proposition 1: Two robots A and B that do not detect each other are guaranteed to be collision free if and only if $\mathbf{v}_A \in S_A$ and $\mathbf{v}_B \in S_B$.

Proof: The proof that $\mathbf{v}_A \in S_A$ is a sufficient condition for no collision is a direct consequence of Lemma 1. If both robots do not see each other, then they both move away from the line Γ that divides the space between the two robots. They thus move away from each other.
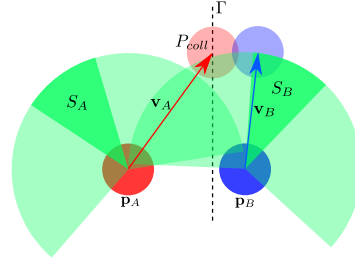

Fig. 1. Illustration of a situation for which choosing $\mathbf{v}_A \notin S_A$ leads to a collision. Robot A is the dark red disk, robot B is the dark blue disk. The FOV of the robots are shown as light green circular sectors. The dark green circular sectors are the sensor-constraint sets for each robot. The light red and blue disks represent position of robot A and B, respectively, that collided due to robot A choosing a velocity outside of S_A.

The proof of necessity goes as follows: suppose that there is a velocity $\mathbf{v}_A \notin S_A$ that does not lead to a collision for any $\mathbf{p}_B$, $\mathbf{d}_B$ and $\mathbf{v}_B \in S_B$. Choose $\mathbf{p}_B$ and a small ϵ such that $|\angle(\mathbf{d}_A, \mathbf{p}_B - \mathbf{p}_A)| = \alpha_H^s + \epsilon$. It follows that $\mathbf{p}_B \notin F_A$ and that robot A does not see robot B. Now take, without loss of generality, a velocity $\mathbf{v}_A$ such that $|\angle(\mathbf{v}_A, \mathbf{d}_A)| > \alpha_A^s - \pi/2 + \epsilon$, and that $\angle(\mathbf{v}_A, \mathbf{d}_A)$ has the same sign as $\angle(\mathbf{d}_A, \mathbf{p}_B - \mathbf{p}_A)$. Note that by definition $\mathbf{v}_A \notin S_A$. It follows that $|\angle(\mathbf{v}_A, \mathbf{p}_B - \mathbf{p}_A)| < \pi/2$. Define the line $\Gamma \perp (\mathbf{p}_B - \mathbf{p}_A)$ that goes between the two robots at a distance of $r_B + r_A/2$ from robot B. Because $\angle(\mathbf{v}_A, \mathbf{p}_B - \mathbf{p}_A) < \pi/2$, the center of robot A will cross Γ at point P_{coll} in some time in the future Δt_{coll}. It is now possible to choose $\mathbf{v}_B$ such that $\mathbf{v}_B \in S_B$ and that the distance between the robot B and the line Γ is less than $r_B + r_A$ when robot B is closest to P_{coll}. It is thus possible to find a triplet $\mathbf{p}_B$, $\mathbf{d}_B$ and $\mathbf{v}_B \in S_B$ that will lead to a collision between robots A and B. It is thus not possible to have $\mathbf{v}_A \notin S_A$ without risking a collision with B. ∎

Remark 1: The result remains true when more than two robots are involved because a single collision is essentially a two-robot problem.

Remark 2: The angle α_s has to be larger than $\pi/2$ for the sensor-constraint set to exist. If $\alpha_s < \pi/2$, then no guarantee about collision can be given. The risk of collision when $\alpha_s < \pi/2$ is studied in [11].

In fact, it is possible to derive Equation 3 directly from the VO framework. The argument goes as follows: consider two robots A and B that do not detect each other and are about to collide. As robot B is not responsible of the collision, the worst case scenario is that robot B stays stopped. Because the two robot are about to collide, their distance is exactly $r_A + r_B$. Robot B can be anywhere around robot A as long as $\angle(\mathbf{d}_A, \mathbf{p}_B - \mathbf{p}_A) > \alpha_A^s$. As a result, the VO of robot B that robot A has to avoid is a half plane at the collision point between robots A and B and tangent to robot B at that point. As a result, the set of velocities that can potentially lead robot A to collide with robot B because they cannot detect each other is:

$$I = \left\{ \mathbf{v} \mid \left(\angle(\mathbf{v}_A, \mathbf{d}_A) > \alpha_A^s - \frac{\pi}{2} \right) \vee \right.$$
$$\left. \left(\angle(\mathbf{v}_A, \mathbf{d}_A) < -\alpha_A^s + \frac{\pi}{2} \right) \right\} \tag{5}$$

To guarantee no collision, robot A has to pick a velocity in the set $\bar{I} = S_A$.

IV. VELOCITY OBSTACLE

Our approach starts from the classical definition of VO; a VO of robot B from the point of view of robot H is defined as:

$$VO_{H|B}^\tau = \{ \mathbf{v} \mid \exists t \in [0,\tau] :: t\mathbf{v} \in D(\mathbf{p}_B - \mathbf{p}_H, r_B + r_H + \sigma_r) \oplus \Sigma_p \}, \quad (6)$$

with Σ_p a set of possible position error and σ_r the error on the estimated radius. Care is taken to choose a value of τ large enough for the robot H to go from $||\mathbf{v}|| = v_H^{max}$ to $||\mathbf{v}|| = 0$ in less than time τ. In a similar fashion, the maximum velocity should remain low enough in order for the robot to come at a full stop in less than half its sensor range. Those conditions can be summarized as follows:

$$\begin{cases} \tau > \frac{v_H^{max}}{\Delta v_H^{max}} \Delta t \\ v_H^{max} < \sqrt{\frac{\Delta v_H^{max} r_H^s}{\Delta t}}, \end{cases} \quad (7)$$

with Δt the time step between two computations.

As it is not always possible to know the behavior of other robot(s) due to the limited FOV, we take the conservative approach to build the set of velocities that will lead to a collision with robot B, named $C_{H|B}^\tau$, by taking into account all the possible future velocities of robot B. To take into account a possible uncertainty on the estimated velocity of the other robot B, the actuation set U_B is augmented by performing the Minkowski sum with the set of velocity uncertainty Σ_v. Furthermore, to guarantee that all robots have as solution the possibility to stop, the zero velocity point is added to the actuation set of another detected robot B. To keep a convex set, a convex hull operation is performed after the addition of the zero velocity, resulting in the augmented actuation set:

$$U_B^0 = \text{Conv}\Big(\{ U_B \oplus \Sigma_v, (0,0) \} \Big). \quad (8)$$

From the sets $VO_{H|B}^\tau$ and U_B^0 it is now possible to build $C_{H|B}^\tau$, the set containing all velocities of H that will lead to a collision with B in the next time τ, as follows:

$$C_{H|B}^\tau = VO_{H|B}^\tau \oplus U_B^0. \quad (9)$$

The set of velocities that avoids collision with all detected robots and respects the constraints imposed by both the sensing and the actuation of robot H is given by:

$$CA_H^\tau = \left\{ \mathbf{v} \mid \mathbf{v} \notin \bigcup_{B \neq H} C_{H|B}^\tau \wedge \mathbf{v} \in S_H \cap U_H \right\}. \quad (10)$$

The new velocity $\mathbf{v}_H^{new} \in CA_H^\tau$ to apply is then chosen to be the closest to some desired velocity $\mathbf{v}_H^{pref}$. If we choose a quadratic function as optimization function, the search for a solution is limited to the border of the collision avoidance set ∂CA_H^τ. Note that due to the conservative approach, it is possible that $CA_H^\tau = \emptyset$. If it is the case, the robot chooses $\mathbf{v}_H^{new} = [0,0]^T$ as a last resort avoidance maneuver:

$$\mathbf{v}_H^{new} = \begin{cases} \underset{\partial CA_H^\tau}{\arg\min} ||\mathbf{v} - \mathbf{v}_H^{pref}||, & \text{if } CA_H^\tau \neq \emptyset \\ [0,0]^T, & \text{otherwise.} \end{cases} \quad (11)$$

Proposition 2: N robots with velocity defined with Equation 11 will not collide.

Proof: If Equation 11 finds a solution, then by the construction of the velocity obstacle and because of Proposition 2, each robot will choose a velocity that will not collide with any other robot.

If no solution can be found (i.e. $CA_H^\tau = \emptyset$), a robot A will try to stop as quickly as possible (i.e. as fast as U_A allows for). The other robots fall into two categories. Either a robot B does not detect robot A, in which case it is of no threat to robot A because of Proposition 1. Or robot B does detect robot A, in which case, if robot B finds a solution, it will avoid robot A with the consideration that robot B will leave the possibility to robot A to stop.

Finally, if several robots are not able to find a solution, consider the moment where they first are not able to find a solution and start to brake. This can only happen if the sensors of the robots give detections of robots that were not in the FOV before. A robot B can enter the FOV of another robot in two ways. First, the robot B either comes from the rear, and was not seen by some robot A because $|\angle(\mathbf{d}_A, \mathbf{p}_B - \mathbf{p}_A)| > \alpha_A^s$. In this case, robot B is moving towards robot A (otherwise there would be no collision). As a consequence, robot B was able to detect robot A at previous time step and to find a solution to avoid A (as this is the first time step that robot B does not find a solution). If robot B could find a solution previously, the solution includes the possibility that robot A would suddenly brake and stop, even if robot B has to perform an emergency brake too because of the condition described by Equation 7. The second possibility is that robot A could not sense other robots previously because they were out of range. In which case all robots have, according to the conditions of Equation 7, enough time to come to a full stop before colliding with the other robots. ∎

A. Heading Control

The control of the robot's heading, and thus the sensor's FOV, is essentially what allows SAVO to reach any point in the plane (as long as it is reachable) and not just the points in S_H. In this paper, we propose to steer the sensor heading based on the robot's velocity outputted by the collision avoidance algorithm. To avoid a deadlock due to no robot finding a solution, we also propose that when a robot uses its last resort avoidance maneuver, it spins in a defined direction at maximum turn rate:

$$u = \begin{cases} -k_a \angle(\mathbf{d}_H, \mathbf{v}_H), & \text{if } CA_H^\tau \neq \emptyset \\ \dot{\theta}_H^{max}, & \text{otherwise.} \end{cases} \quad (12)$$

To have a smooth transition between deadlock and normal maneuvering on one of the sides, and to use the full range

of turning rate, k_a should be set as follows:

$$k_a = \frac{\dot{\theta}_H^{max}}{\alpha_H^s - \frac{\pi}{2}}. \qquad (13)$$

If the gain k_a is found to be too aggressive, it is possible to use a smaller value for $\dot{\theta}_H^{max}$.

B. Reciprocity

Due to the limited FOV, it is not always possible for a robot to know if it is seen by an other robot, especially when the sensors are noisy. As a consequence, a robot is not able to rely on the fact another robot will cooperate, which is an important assumption for algorithms such as ORCA [6] or HRVO [10]. We propose the following to mitigate the oscillations common to non-reciprocal VO: the robot checks if only giving half of the change of velocity is also collision free:

$$\mathbf{v}_H^{half} = \frac{\mathbf{v}_H^{new} + \mathbf{v}_H^{prev}}{2} \notin \bigcup_{B \neq H} C_{H|B}^{\tau}, \qquad (14)$$

with $\mathbf{v}_H^{prev}$ the velocity computed at previous time step. The velocity $\mathbf{v}_H^{prev}$ is preferred over the actual velocity $\mathbf{v}_H$ because experiments with our quadrotors showed that the use of $\mathbf{v}_H$ amplifies the noise on the actuation. If $\mathbf{v}_H^{half}$ is collision free, it is used instead of $\mathbf{v}_H^{new}$, resulting in a maneuver aggressiveness similar to what is obtained with reciprocal algorithms. When robots meet, the second collision avoidance computation is usually enough to perform the avoidance with $\mathbf{v}_H^{half}$ instead of $\mathbf{v}_H^{new}$. Note that $\mathbf{v}_H^{half} \in S_H$ and $\mathbf{v}_H^{half} \in U_H$ because both sets are convex and both $\mathbf{v}_H^{new}$ and $\mathbf{v}_H^{prev}$ are in those two sets.

V. RESULTS

In this section, we present the results obtained with SAVO. First, the implementation differences with well-known state-of-the-art algorithms are explained. Second, simulation results are presented, followed by experimental results. The results are also presented in the supplementary video, in which the timings of the trajectories appear more clearly.[1]

A. Implementation

Due to the complex shapes that this method can create, it is not possible to apply the commonly used "triangle" approach for the computation on the velocity sets (e.g., [10]). It is not possible to simplify the problem into half-planes as in [6] due to the possible lack of reciprocity. To stay as close as possible to the mathematical formulation, the sets are implemented as convex polytopes. This has the drawback to be more computationally heavy and in principle scale with $\mathcal{O}(n^2)$, n being the number of robots observed. However, the computational cost is physically limited by the sensor's range. This method is thus well suited for robots with a small sensor range or that navigate in sparsely populated environments.
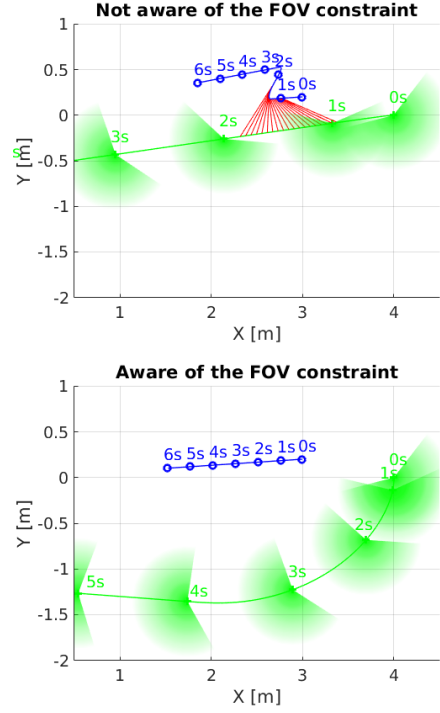
Fig. 2. Trajectories obtained with the first scenario. Top: the algorithm is not aware of the FOV. Bottom: The SAVO algorithm. The solid blue and green lines are robots' trajectories. The markers and numbers give an indication of the timing of the trajectories. The FOV of some robots at the timings indicated by numbers are represented by circular sectors. The color of the circular sectors match the color of the trajectories. The FOV range is not to scale. The red solid lines are distances between two robots that are less than the sum of their radii.

For both simulations and real experiments, the robots had a goal position $\mathbf{p}_{goal}$ from which the preferred velocity was derived:

$$\mathbf{v}_H^{pref} = s_H^{pref} \frac{\mathbf{p}_H^{goal} - \mathbf{p}_H}{||\mathbf{p}_H^{goal} - \mathbf{p}_H|| + r^g} \qquad (15)$$

with s_H^{pref} the preferred speed of robot H and r^g a term that will slow down the robot as it reaches its goal position.

B. Simulations

To illustrate the importance of FOV constraints in collision avoidance, we designed two scenarios where an implementation of VO that does not take the constraints into account fails to safely maneuver. The canonical VO implementation differs from the SAVO algorithm in only two points. First, it does not constrain the velocity of a robot H to stay in S_H. Second, it does not include the zero velocity in what would be U_B^0 for the SAVO algorithm. It uses, instead of U_B^0, the set $U_B^\Sigma = U_B \oplus \Sigma_v$. This conservative implementation of VO allows us to demonstrate that the problem is indeed due to violating the constraints imposed by the limited FOV of the sensor. The parameters of our simulations are $r = 0.4\ m$, $r^s = 2.5\ m$, $\alpha^s = 110°$, $\Delta v^{max} = 0.12\ m/s$, $v^{max} = 4\ m/s$, $\dot{\theta}^{max} = 0.5\ rad/s$ and $\tau = 1\ s$. The simulations are performed with a time step $\Delta t = 0.05\ s$. The same parameters are used for both SAVO and the normal VO
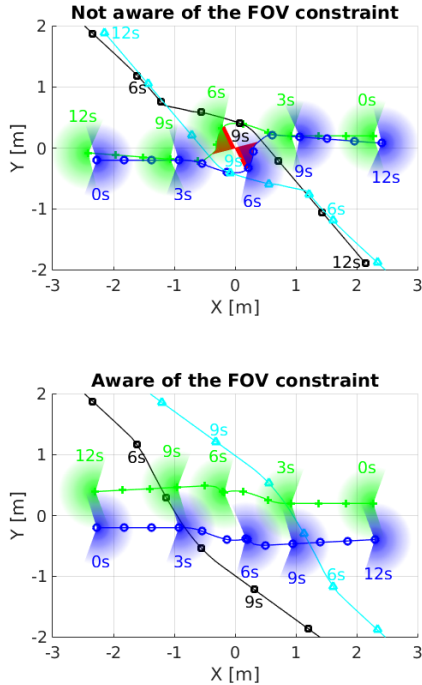
Fig. 3. Trajectories obtained with the second scenario. Top: the algorithm is not aware of the FOV. Bottom: The SAVO algorithm. The solid blue, black, green and cyan lines are robots' trajectories. The markers and numbers give an indication of the timing of the trajectories. The FOV of some robots at the timings indicated by numbers are represented by circular sectors. The color of the circular sectors match the color of the trajectories. The FOV range is not to scale. The red solid lines are distances between two robots that are less than the sum of their radii.

implementations. If one algorithm gives a command that the robot is not able to follow, the actual actuation saturate.

The first scenario involves two robots that start back-to-back. Both robots have their destination on the left side of the horizontal axis. One slower robot has the right orientation and simply has to go forward to reach its destination. The other, faster, robot faces away from its destination. The fastest trajectory is for the robot to go in a straight line, which leads to a collision in the case the algorithm is not aware of the each other constraints, as shown in the top plot of Figure 2. When the SAVO algorithm is used, the faster robot stays at the same position until it has turned enough for the sensor to see a clear path towards the destination, resulting in the trajectories shown in the bottom plot of Figure 2.

The second scenario involves four robots. Two slow robots avoid each other and thus get close. Just when the two slow robots successfully avoid the collision and go out of each other's detection range, each one sees a fast robot approaching from the other side they performed the avoidance. In case where the algorithm is not respecting the constraints imposed by the sensor, the slow robots will perform an aggressive maneuver towards each other to avoid the faster robots. That maneuver will lead them to collide (top plot of Figure 3). With the SAVO algorithm, the two slow robots will stop instead and remain safe (bottom plot of Figure 3). Notice how SAVO, although more constrained, generates

Fig. 4. The AscTec Hummingbird used in the experiments.

smoother and faster trajectories compared than the normal VO implementation.

C. Experiments

To further validate our approach, we performed experiments with real quadrotors. Although they do move in a three dimensional space, avoidance with quadrotors should be performed horizontally as it is hazardous to maneuver in the down-wash of another quadrotor. Furthermore, quadrotors typically are more agile at translational compared to yaw rotation, making it an ideal case study for the SAVO algorithm.

Our experimental setup is composed of two Hummingbirds quadrotors from Ascending Technologies (see Figure 4). Each quadrotor is equipped with an ARM-based CPU that runs a Linux distribution with the Robotic Operating System (ROS) installed. The link between ROS and the quadrotor is done using [17]. The quadrotors fly in a room with dimensions 11.8 *m* by 9.8 *m* and 5 *m* high, and equipped with a Motion Capture System (MCS) that tracks the quadrotors with a millimetric accuracy. Each quadrotor receives the pose of both quadrotors through WiFi at 100 *Hz*. The sensor range and FOV is emulated before the MCS data is used in the SAVO algorithm.

Because the motion of a quadrotor is disturbed by non-deterministic air flows, small accelerations performed by the quadrotor have little effect on its motion. As a result the quadrotor is not able to properly accelerate and maneuver if Δv^{max} is too small. As a result, we chose $\Delta v^{max} = 0.3$ *m/s*. The desired speed was set to $s^{pref} = 0.6$ *m/s* The other parameters have the same values as used in simulation.

To test the SAVO algorithm, two scenarios were designed. Those scenarios are to demonstrate that the algorithm is able to run on a real platform rather than to show the consequences of a limited FOV. The first scenario involves two quadrotors in a head-on configuration. The first quadrotor starts at the position $[3.5, 0]$ *m* and has its goal position at $[-3.5, 0]$ *m*. The other robot does the opposite: it goes from the position $[-3.5, 0]$ *m* and tries to reach position $[3.5, 0]$ *m*. The second scenario illustrates a crossing scenario. The first quadrotor starts at $[-3.5, 2]$ *m* and has its goal at $[3.5, -2]$ *m*. The second robot starts at $[3.5, 2]$ *m* and has its goal at $[-3.5, -2]$ *m*. In both scenarios, the robots start oriented facing the goal. The resulting trajectories are shown in Figure 5. Notice that the robot move back at some points of the trajectory. This is an overshoot of the braking maneuver and is inherent to the propulsion method of the

Fig. 6. Boxplot of the minimum distance between the quadrotors during the avoidance maneuvers. The red dashed line represents the distance at with the quadrotors would collide.
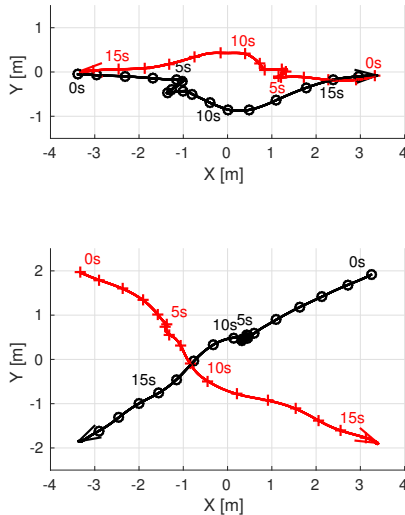
Fig. 5. Trajectories obtained for the experiments with quadrotors. Top: Head on collision. Bottom: Cross intersection. The black and red solid lines represent the trajectory of each quadrotor. The arrow ends show the direction of motion. The markers and numbers give an indication of the timing of the trajectory.

quadrotor. It can be accounted for by increasing the radius of the quadrotors in the SAVO algorithm. This issue should not be present for ground robots because when they brake, their velocity goes to zero with respect to the ground. For each scenario, the minimum distance between the robots during the avoidance maneuver was recorded and is shown in Figure 6. Each scenario was run at least 19 times and at no moment did the distance between the robots drop below $0.8\ m$, the distance at which they would collide. In fact, the robots stayed at a safe distance. This is a consequence of the conservative approach required by the limited FOV of the robots.

VI. CONCLUSION

In this paper, we have presented a VO-based collision avoidance that remains safe when the robot's sensors only detect other robots in a limited FOV. The conditions and constraints to satisfy safe maneuvering are presented and scenarios where ignoring the FOV constraints lead to collision are shown. We validate our approach with simulations and real robot experiments.

As future work, the relation between reciprocity and FOV should be investigated, as for a significant part of possible scenario all robots involved are able to detect each other. In that case, the presented algorithm can be improved by applying the concept of reciprocity. Other future work would extend this work to the third dimension for aerial applications. Finally, the possibility of deadlocks and means to mitigate them should be investigated.

REFERENCES

[1] S. Thrun, D. Fox, and W. Burgard, "The dynamic window approach to collision avoidance," *IEEE Transactions on Robotics and Automation*, vol. 4, p. 1, 1997.
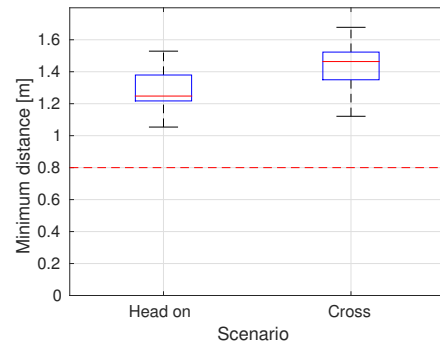
[2] L. Makarem and D. Gillet, "Decentralized coordination of autonomous vehicles at intersections," in *IFAC World Congress*, vol. 18, no. 1, 2011, pp. 13 046–13 051.

[3] T. Zsedrovits, Á. Zarándy, B. Vanek, T. Péni, J. Bokor, and T. Roska, "Collision avoidance for UAV using visual detection," in *IEEE International Symposium on Circuits and Systems*, 2011, pp. 2173–2176.

[4] D. E. Chang and J. E. Marsden, "Gyroscopic forces and collision avoidance with convex obstacles," in *New trends in nonlinear dynamics and control and their applications*, 2003, pp. 145–159.

[5] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991.

[6] J. van den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Proc. Int. Symp. on Robotics Research 2009, ser. Springer Tracts in Advanced Robotics*. Springer, 2011, pp. 3–19.

[7] D. Bareiss and J. van den Berg, "Generalized reciprocal collision avoidance," *The International Journal of Robotics Research*, vol. 34, no. 12, pp. 1501–1514, 2015.

[8] J. Alonso-Mora, A. Breitenmoser, P. Beardsley, and R. Siegwart, "Reciprocal collision avoidance for multiple car-like robots," in *IEEE International Conference on Robotics and Automation*, 2012, pp. 360–366.

[9] P. Conroy, D. Bareiss, M. Beall, and J. van den Berg, "3-D reciprocal collision avoidance on physical quadrotor helicopters with on-board sensing for relative positioning," *arXiv preprint arXiv:1411.3794*, 2014.

[10] J. Snape, J. van den Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 696–706, 2011.

[11] Q. Zhang, G. Leng, and V. Govindaraju, "Duration of collision-free motion of unmanned vehicles in a confined area," *Robotica*, vol. 34, no. 02, pp. 347–360, 2016.

[12] B. Fankhauser, L. Makarem, and D. Gillet, "Collision-free intersection crossing of mobile robots using decentralized navigation functions on predefined paths," in *IEEE International Conference on Cybernetics and Intelligent Systems*, 2011, pp. 392–397.

[13] A. Mcfadyen, L. Mejias, P. Corke, and C. Pradalier, "Aircraft collision avoidance using spherical visual predictive control and single point features," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 50–56.

[14] S. Roelofsen, A. Martinoli, and D. Gillet, "Distributed deconfliction algorithm for unmanned aerial vehicles with limited range and field of view sensors," in *American Control Conference*, 2015, pp. 4356–4361.

[15] S. Roelofsen, D. Gillet, and A. Martinoli, "Reciprocal collision avoidance for quadrotors using on-board visual detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2015, pp. 4810–4817.

[16] S. Roelofsen, A. Martinoli, and D. Gillet, "3D collision avoidance algorithm for unmanned aerial vehicles with limited field of view constraints," in *IEEE 55th Conference on Decision and Control*, 2016, pp. 2555–2560.

[17] "asctec_hl_framework ROS package," http:http://wiki.ros.org/asctec_mav_framework, 2015-02-04.