

Distributed Deconfliction Algorithm for Unmanned Aerial Vehicles with Limited Range and Field of View Sensors

Steven Roelofsen^{1,2}, Alcherio Martinoli¹, Denis Gillet²

Abstract—This paper proposes a novel approach for collision avoidance between Unmanned Aerial Vehicles (UAVs) with limited range and field of view sensors. The algorithm is designed for unicycle vehicles that need to fly above a specific minimal speed to maintain flight. It uses a navigation function approach when the UAV is clear from conflict and smoothly switches to a collision avoidance maneuver when other UAVs are encountered. We show that the proposed avoidance algorithm can ensure a collision-free path. We also carry out a throughout quantitative analysis of the algorithm singularities and propose heuristic recipes for avoiding deadlock situations. Simulations are performed to show the effectiveness of the algorithm.

I. INTRODUCTION

In the last years numerous applications for UAVs appeared. Up to date, most applications still have a human in the loop but the trend is to have fully autonomous UAVs. Such autonomy requires the deployment of collision avoidance algorithms compatible with other flying entities. While UAVs can have a large variety of shapes and sizes, an important class of vehicles is represented by fixed-wing aircraft. Collision avoidance for fixed-wing aircraft is challenging because these vehicles can only safely operate above a certain minimal speed (i.e. no hovering), a feature that emphasizes their intrinsically nonholonomic nature. For that reason, avoidance maneuvers are typically performed by either a change in altitude or heading.

Several techniques have been proposed to solve this problem. For example, obstacles can be described in the velocity space, allowing for the computation of the speed that will best avoid collision [1]. Such technique has been applied to fixed-wing aircraft [2]. Another approach is to optimize a trajectory for a user-defined function. The advantage of such a method is that virtually all constraints can be met by just including them in the optimization problem definition. Such optimality is often resulting into a high computational cost that limits the use of this approach. An example can be found in [3]. A more complete review on the subject can be found in [4].

Studies also investigated the use of navigation functions for reciprocal collision avoidance. The navigation function is based on an analytic function whose gradient is the direction

This work has been financially supported by Honeywell, and has benefitted of the administrative and technical coordination of the EPFL Transportation Center.

¹ S. Roelofsen and A. Martinoli are with the Distributed Intelligent Systems and Algorithms Laboratory, School of Architecture, Civil and Environmental Engineering, École Polytechnique Fédérale de Lausanne.

² S. Roelofsen and D. Gillet is with the the Coordination and Interaction System Group, School of Engineering, École Polytechnique Fédérale de Lausanne.

to follow [5]. The advantage of an approach formulated using navigation functions is that analytical proofs can be obtained. In [6] navigation functions are used to avoid collision between autonomous vehicles under actuation and sensing constraints. Avoidance maneuvers for aircraft have been presented in [7] and validated for constrained sensing (both in range and in field of view), although they leverage more geometrical reasoning than navigation functions. In [8], the authors showed that a specific navigation function, known as dipolar function, guarantees collision avoidance for agents with fixed-wing dynamics. In [9] the authors introduce a gyroscopic force field to improve the collision avoidance capability of the navigation function for holonomic agents. In [10] a similar approach is used and applied to aircraft with limited turning rate and sensing range. The two latter studies rely on introducing a gyroscopic force to the potential field gradient to avoid stagnation points.

In this paper, we will show that the same effect can be obtained by overwriting the potential field output by an ad hoc turning rate. This will allow for collision avoidance with limited Field of View (FOV) sensors. We will show that our algorithm can guarantee collision-free paths and discuss the conditions for agents to separate from each other after the avoidance maneuver is terminated. Simulations will be provided to complete the demonstration.

II. PROBLEM FORMULATION

In this paper, a group of N non-holonomic homogeneous agents moving on a two dimensional plane are considered. Their motions are modeled as unicycles. The unicycle model has been chosen as an approximation for the UAV model because of its simplicity and because it captures the fixed wing aircraft's maneuvering capability well. The unicycle model is defined as:

$$\begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} v_i \cos \theta_i \\ v_i \sin \theta_i \\ u_i \end{bmatrix} \quad (1)$$

with u_i the turning rate command and $V_{min} \leq v_i \leq V_{max}$ the forward speed command of agent i . Note that this notation allows for constant speed agents with $V_{max} = V_{min}$, although it is not desirable as we will show in Section V. $\mathbf{q}_i = [x_i, y_i]^T$ is agent's position and θ_i is its heading. Together, \mathbf{q}_i and θ_i define the pose vector. In this paper all angles are defined on the interval $[-\pi, \pi]$ and all operations with angles return a value in that interval.

Each agent follows a path defined by a number of way-points. The navigation is done in two steps. First, the agent computes its desired heading using a navigation function.

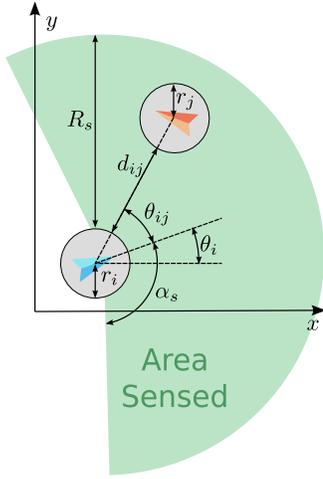


Fig. 1: Illustration of the angles and distances used in this paper. The green area represents the area where agent j (in orange) is sensed by agent i (in blue).

If the agent is on a collision course, it will perform an avoidance maneuver by turning right.

The collision zone of each agent i has a disk shape of radius r_i . The collision zone is a zone that should not be crossed by any obstacle or other agent. A collision is defined between agents i and j if the distance between the two agents is smaller than $r_i + r_j$.

An agent i is able to get range and bearing information of any other agent j that is in both sensing range and FOV of agent i . No occlusion effect is considered. The bearing angle with respect to agent i frame is defined as $\theta_{ij} = \text{atan2}(y_j - y_i, x_j - x_i) - \theta_i$ with θ_i being the heading of agent i . The FOV is defined as the set of bearing angles θ_{ij} that are in the interval $[-\alpha_s; \alpha_s]$ with $\frac{\pi}{2} \leq \alpha_s < \pi$. The measured distance between two agents is defined as $d_{ij} = \|\mathbf{q}_j - \mathbf{q}_i\| - r_i - r_j$ with r_i and r_j being the radius of agent i and j respectively. Agent j is not sensed if either the distance between the considered agents is above the sensing range R_s or the bearing angle is outside the FOV of agent i . Figure 1 illustrates the parameters presented in this section.

III. NAVIGATION FUNCTION

The navigation function is in charge to indicate the direction that brings the agent i from its initial state position \mathbf{q}_o to its destination \mathbf{q}_d . In the chosen navigation functions, this direction is obtained from the negative gradient of an analytical function called potential field. The potential field is defined in such a way that it has properties of interest such as convergence, guaranteed obstacle avoidance, and smooth trajectories.

The navigation function $V(\mathbf{q})$ used in this paper is inspired from [11]. Only the path planning function is kept as the collision avoidance is implemented using another method described below. Our navigation function is

$$V_i(\mathbf{q}_i) = \|\mathbf{q}_i - \mathbf{q}_{d_i}\|^2 + K_p \beta \left(\frac{\|\mathbf{q}_i - \mathbf{q}_{d_i}\|}{R_p} \right) e_i^2 \quad (2)$$

with K_p a constant gain and $e_i = \frac{\|(\mathbf{q}_{o_i} - \mathbf{q}_{d_i}) \times (\mathbf{q}_i - \mathbf{q}_{d_i})\|}{\|\mathbf{q}_{o_i} - \mathbf{q}_{d_i}\|}$. R_p is a radius that reduces the strength of the navigation function as the agent reaches its destination. The barrier function β is defined as follows:

$$\beta(a) = \begin{cases} 0 & \text{if } a < 0 \\ f(a) & \text{if } 0 \leq a < 1 \\ 1 & \text{otherwise} \end{cases} \quad (3)$$

with $f(a)$ a monotonic and smooth function that allows a smooth transition of β from 0 to 1. In this paper $f(a)$ is similar to the one used in [12] and is defined as:

$$f(a) = 3a^2 - 2a^3 \quad (4)$$

The gradient of the navigation function $[x_t, y_t]^T = -\nabla_{\mathbf{q}_i} V_i(\mathbf{q})$ defines the desired direction of motion. As the agent control input is heading rate and forward speed, a transformation is needed. An usual transformation is to compute a target heading $\theta_t = \text{atan2}(y_t, x_t)$. The control input for the agent is given by:

$$u_i = K(\theta_t - \theta_i) \quad (5)$$

$$v_i = V_{max} \quad (6)$$

Note that there are no changes in the forward speed. A variation in forward speed would only bring marginal performance gain when that same variation is needed for collision avoidance. This need for variable speed is explained in Section V.

IV. COLLISION AVOIDANCE

Because vehicles with unicycle type dynamics can not easily change their speed, the avoidance maneuver requires predominantly a change in the heading. Incorporating obstacles in the navigation function will lead to quick changes in target heading. This is because the gradient is often radial to the obstacle. If the destination and the obstacle are perfectly aligned, the agent will move towards the destination until the obstacle avoidance effect is larger, in which case the gradient points in the opposite direction.

Our distributed collision avoidance method adds a turning rate to the control law as well as a change in speed. To avoid that the navigation function interferes with the collision avoidance, they are both weighted as function of the distance d_{ij} and bearing angle θ_{ij} . The turning rate command u_i of agent i is defined as:

$$u_i = K \left(1 - \max_j (\beta_{a_j} \beta_{d_j}) \right) (\theta_t - \theta_i) + \sum_j \beta_{a_j} \beta_{d_j} \frac{-\pi V_{max}}{d_{ij}} \quad (7)$$

with $\max_j (\beta_{a_j} \beta_{d_j})$ the maximum value of $\beta_{a_j} \beta_{d_j}$ for all j . The forward speed v_i is defined as:

$$v_i = V_{max} - \Delta V \max_j (\beta_{a_j} \beta_{d_j}) \quad (8)$$

with $\Delta V = V_{max} - V_{min}$ agent's allowed speed range. The two barrier functions β_{a_j} and β_{d_j}

$$\beta_{a_j} = \beta \left(\frac{\alpha_s - |\theta_{ij}|}{\alpha_s - \frac{\pi}{2}} \right) \quad (9)$$

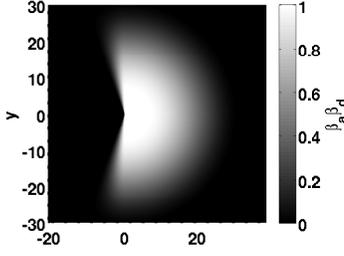


Fig. 2: Shape of the function $\beta_{a_j}\beta_{d_j}$. The function equals zero in the back of the agent, ignoring any other agents in that area as it is unable to sense them.

$$\beta_{d_j} = \beta\left(\frac{R_s - d_{ij}}{R_s - R_a}\right) \quad (10)$$

are used to limit the avoidance behavior in accordance with agent's sensing capability. R_s and α_s are parameters characterizing the sensing area (see Figure 1). R_a is the radius for which the agent will only do collision avoidance ($R_a < R_s$). The shape of the barrier function allows for a smooth transition between the navigation toward the goal and the collision avoidance. An example of the shape of $\beta_{a_j}\beta_{d_j}$ for $R_s = 30$, $R_a = 4$ and $\alpha_s = \frac{9\pi}{8}$ is shown in Figure 2. $\beta(\cdot)$ is the barrier function defined by Equation 3.

Proposition 1: An agent with dynamic model described by Equation 1 and with controller described in Equations 7 and 8 will not move in a way that will lead to collision with another agent.

Proof: Suppose that a collision occurs at time t_c between two agents and consider them earlier when they were apart from each other by a small distance $2R$. As they can go at a maximum speed of V_{max} , there is a small time interval $\Delta t = \frac{R}{V_{max}}$ for which they will not collide. Δt is thus a lower bound of the time required for them to collide. We want to prove by contradiction that *if an agent is moving towards another agent, there will not be a frontal collision between the two agents*. That is, an agent on a collision course will turn enough to not face the other agent and move away from the collision point in the time Δt , contradicting with the fact that t_c is the time of collision.

As they are on a collision course, at least one agent, for instance agent i , is moving towards the other during the time $t \in [t_c - \Delta t, t_c]$. Because it is moving towards the collision, agent i is sensing another agent j in front of it, meaning that $|\theta_{ij}| \leq \frac{\pi}{2}$. Thus according to Equation 9 $\beta_a = 1$. Moreover, the distance between the agents is small, meaning that $d_{ij} \leq R_a$ thus $\beta_{d_{ij}} = 1$ (from Equation 10). As a result we have $(1 - \max_j \beta_{a_j}\beta_{d_j}) = 0$ thus the control law is reduced to

$$u_i = \sum_j \beta_{a_j}\beta_{d_j} \frac{-\pi V_{max}}{d_{ij}} < \frac{-\pi V_{max}}{d_{ij}} < 0 \quad (11)$$

The change of heading during time Δt of that agent can be computed by integrating it:

$$|\Delta\theta_i| = \left| \int_{t_c - \Delta t}^{t_c} u_i dt \right| \geq \left| \int_{t_c - \Delta t}^{t_c} \frac{-\pi V_{max}}{d_{ij}} dt \right| \quad (12)$$

As the agent is going to collide, it will always move towards the other one during the time $t \in [t_c - \Delta t, t_c]$, the distance d_{ij} is continually decreasing from the distance R . Equation 12 can be rewritten as:

$$\begin{aligned} |\Delta\theta_i| &\geq \left| \int_{t_c - \Delta t}^{t_c} \frac{-\pi V_{max}}{d_{ij}} dt \right| \\ &\geq \left| \int_{t_c - \Delta t}^{t_c} \frac{-\pi V_{max}}{R} dt \right| \\ &= \left| \frac{-\pi V_{max}}{R} \Delta t \right| = \pi \end{aligned} \quad (13)$$

Thus the agent did more than a π turn during the time $t \in [t_c - \Delta t, t_c]$ contradicting the affirmation that the agent collides with the other agent in front of it. It will thus never reach the collision point because it will move away from it before t_c . ■

During the whole proof, the case of agent j was not considered. Only mattered the fact that agent i will not travel further than a distance R towards agent j . We will now discuss the possible behaviors of agent j .

Agent j can have four different behaviors depending on its initial heading. In the first case, the agent j is also moving towards the collision point. In that case, its behavior will be the same as agent i ; it will perform a collision avoidance maneuver to the right. The second possible case is when agent j is moving away from agent i with an angle $|\theta_{d_{ji}}| > \alpha_s + \epsilon$ with

$$\begin{aligned} |\Delta\theta_j| &= \left| \int_{t_c - \Delta t}^{t_c} u_j ds \right| = \int_{t_c - \Delta t}^{t_c} |K(\theta_t - \theta_j)| ds \\ &= \epsilon \ll \pi - \alpha_s. \end{aligned} \quad (14)$$

$u_j = K(\theta_t - \theta_j)$ because the agent i is out of the field of view of agent j and thus $\beta_{a_i}\beta_{d_i} = 0$. The agent j will always have $|\theta_{d_{ji}}| > \alpha_s$ and will keep moving away from agent i while agent i is performing its collision avoidance maneuver. Third case is defined by $-\frac{\pi}{2} > \theta_{d_{ji}} \geq -\alpha_s - \epsilon$. Then the agent j will not rotate in a way that will lead to $\theta_{d_{ji}} > -\frac{\pi}{2}$ because at $\theta_{d_{ji}} = -\frac{\pi}{2}$ the avoidance term will bring it back in the region of $-\frac{\pi}{2} > \theta_{d_{ji}} \geq -\alpha_s - \epsilon$. The last possible case happens when $\frac{\pi}{2} < \theta_{d_{ji}} \leq \alpha_s + \epsilon$. Two outcomes are then possible. On the one hand, $\theta_{d_{ji}} > \frac{\pi}{2}$ during the time agent i performs its avoidance maneuver, agent j will move away from the collision during time Δt . On the other hand, $\theta_{d_{ji}}(t) < \frac{\pi}{2}$ during $t \in [t_c - \Delta t, t_c]$ in which case one can choose a new $\Delta t' \in]0, \Delta t[$ and reuse the proof proposed above with agent j instead of agent i . Note that the initial situation will always be classifiable as one of the first three scenarios because to get to the fourth one it needs to cross the region with $|\theta_{d_{ji}}| > \alpha_s$ for which the rotation during time $t \in [t_c - \Delta t', t_c]$ is very small.

This collision avoidance algorithm also works for more than two agents. If multiple agents are surrounding a given agent, that agent will spin until it finds a passage through which it can escape. This spinning effect can be seen as an emulation of a full stop as the agent remains in a small area.

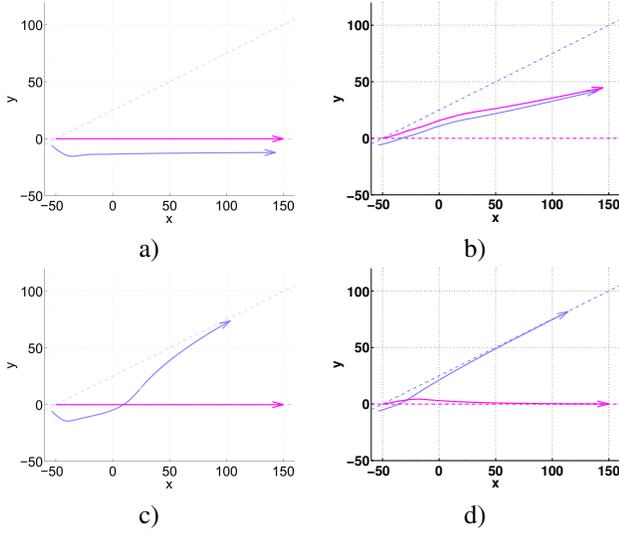


Fig. 3: Illustration of a scenario where two agents move in the same direction even if they do not have the same destination. The arrow heads show the direction of motion. The dashed lines show the desired courses (best seen in color). a) The proposed algorithm in the case the forward speed is constant. b) The same scenario for the algorithm presented in [10]. c) Same algorithm as in a) but the agents can decrease their speed. d) Same algorithm as in b) but the agents can decrease their speed.

V. AVOIDING LOCAL MINIMUM

In this section we show why the change in speed is necessary. The result will be discussed afterwards. Only the case of two agents interacting is considered.

In the context of this work a local minimum means that two agents do not turn anymore although at least one of them is not following its desired direction. An example of this situation is shown in Figure 3. We will show that this configuration is stable. It is not a mathematical proof but rather an approximation to help understand what is happening. This deadlock happens when one agent is flying in the rear of another one and the avoidance turning rate equals the goal tracking turning rate, or differently stated:

$$\dot{\theta}_i = K(1 - \beta_{a_j}\beta_{d_j})(\theta_t - \theta_i) + \beta_{a_j}\beta_{d_j}\frac{-\pi V_{max}}{d_{ij}} = 0 \quad (15)$$

A first condition to have a local minimum can be derived directly from Equation 15. Indeed, writing it differently we obtain:

$$(\theta_t - \theta_i) = \frac{1}{K(1 - \beta_{a_j}\beta_{d_j})}\beta_{a_j}\beta_{d_j}\frac{\pi V_{max}}{d_{ij}} > 0 \quad (16)$$

As $K > 0$ and $0 \leq \beta_{a_j}\beta_{d_j} \leq 1$ the value of $(\theta_t - \theta_i)$ is always positive. Thus the local minimum forces the agent to deviate from its desired direction towards the right.

To show that the state resulting from Equation 15 is a local

minimum, consider the second time derivative of θ_i ,

$$\ddot{\theta}_i = \frac{d}{dt} \left[K(1 - \beta_{a_j}\beta_{d_j})(\theta_t - \theta_i) + \beta_{a_j}\beta_{d_j}\frac{-\pi V_{max}}{d_{ij}} \right] \quad (17)$$

To be able to get the results, the following assumptions were adopted:

Assumption 1: Both agents move at the same constant speed $v_i = v_j = V_{max}$. This assumption will be relaxed later in this section when the means to avoid this local minimum will be discussed. The control law in Section IV uses the result from this section.

Assumption 2: The desired heading θ_t does not vary in time. This is reasonable if the change in target heading is much slower than the time an agent takes to converge to the local minimum.

Assumption 3: The two agents start with similar headings so that $\Delta\theta = \theta_i - \theta_j$ is small and that the approximation $\Delta\theta^2 \approx 0$ holds.

Assumption 4: The agent j is not influenced by agent i because agent i is not in the FOV of agent j . As consequence $\dot{\theta}_j = 0$ and $\ddot{\theta}_j = 0$.

The derivation of the local minimum will go as follows. We will first get the time derivative of Equation 15. It will be followed by the time derivative of the barrier functions and the sensing data. Then the pieces will be put together to get a second order differential equation. By approximating the obtained differential equation by a linear system we will discuss the convergence of the local minimum depending on the spatial configuration of the two agents. Let's first consider the first term of Equation 17:

$$\begin{aligned} & \frac{d}{dt} [K(1 - \beta_{a_j}\beta_{d_j})(\theta_t - \theta_i)] \\ &= -K \left[\frac{d}{dt} [\beta_{a_j}\beta_{d_j}](\theta_t - \theta_i) + (1 - \beta_{a_j}\beta_{d_j})\dot{\theta}_i \right] \end{aligned} \quad (18)$$

The second term of Equation 17 is:

$$\begin{aligned} & \frac{d}{dt} \beta_{a_j}\beta_{d_j} \frac{-\pi V_{max}}{d_{ij}} \\ &= \frac{-\pi V_{max}}{d_{ij}} \left[\frac{d}{dt} [\beta_{a_j}\beta_{d_j}] - \beta_{a_j}\beta_{d_j} \frac{\dot{d}_{ij}}{d_{ij}} \right] \end{aligned} \quad (19)$$

The time derivative of the barrier function is obtained by using partial derivatives.

$$\begin{aligned} \frac{d}{dt} \beta_{a_j}\beta_{d_j} &= \beta_{a_j} \frac{\partial \beta_{d_j}}{\partial d_{ij}} \dot{d}_{ij} + \beta_{d_j} \frac{\partial \beta_{a_j}}{\partial \theta_{ij}} \dot{\theta}_{ij} \\ & \quad + \frac{\partial \beta_{a_j}\beta_{d_j}}{\partial t} \end{aligned} \quad (20)$$

Note that $\frac{\partial \beta_{a_j}\beta_{d_j}}{\partial t} = 0$ because both functions are time invariant. Because β is monotonic increasing we have:

$$\frac{\partial \beta_{a_j}}{\partial \theta_{ij}} \begin{cases} = 0 & \text{if } |\theta_{ij}| \leq \frac{\pi}{2} \\ < 0 & \text{if } \frac{\pi}{2} < \theta_{ij} \leq \alpha_s \\ > 0 & \text{if } -\frac{\pi}{2} > \theta_{ij} \geq -\alpha_s \\ = 0 & \text{otherwise} \end{cases} \quad (21)$$

$$\frac{\partial \beta_{d_j}}{\partial d_{ij}} \begin{cases} = 0 & \text{if } d_{ij} \leq R_a \\ < 0 & \text{if } R_a < d_{ij} \leq R_s \\ = 0 & \text{otherwise} \end{cases} \quad (22)$$

The time derivatives of the sensor values \dot{d} and $\dot{\theta}_{ij}$ needs to be further developed. The time derivative \dot{d} is computed as follows :

$$\dot{d}_{ij} = \frac{1}{\|\Delta \mathbf{q}\|} (\Delta x \Delta \dot{x} + \Delta y \Delta \dot{y}) \quad (23)$$

with $\Delta \mathbf{q} = \mathbf{q}_j - \mathbf{q}_i$. Using Assumption 1 the position difference's change rate $\Delta \dot{\mathbf{q}} = [\Delta \dot{x}, \Delta \dot{y}]^T$ becomes:

$$\begin{aligned} \Delta \dot{x} &= V_{max} (\cos \theta_j - \cos \theta_i) \\ &= -2V_{max} \sin \left(\frac{\theta_j + \theta_i}{2} \right) \sin \left(\frac{\theta_j - \theta_i}{2} \right) \end{aligned} \quad (24)$$

$$\begin{aligned} \Delta \dot{y} &= V_{max} (\sin \theta_j - \sin \theta_i) \\ &= 2V_{max} \cos \left(\frac{\theta_j + \theta_i}{2} \right) \sin \left(\frac{\theta_j - \theta_i}{2} \right) \end{aligned} \quad (25)$$

Equations 24 and 25 can be approximated with $2 \sin \left(\frac{\theta_j - \theta_i}{2} \right) \approx -\Delta \theta$ using Assumption 3. Equation 23 can now be approximated by:

$$\dot{d}_{ij} \approx -\frac{1}{\|\Delta \mathbf{q}\|} V_{max} \Delta \theta (-\Delta x \sin \bar{\theta} + \Delta y \cos \bar{\theta}) \quad (26)$$

with $\bar{\theta} = \frac{\theta_i + \theta_j}{2}$. Because the measured bearing angle of agent i in the world frame is $\theta_{ij} + \theta_i$ we have that $\Delta \mathbf{q} = \|\Delta \mathbf{q}\| [\cos(\theta_{ij} + \theta_i), \sin(\theta_{ij} + \theta_i)]^T$ and thus:

$$\dot{d}_{ij} \approx V_{max} \begin{bmatrix} \sin \bar{\theta} \\ -\cos \bar{\theta} \end{bmatrix}^T \begin{bmatrix} \cos(\theta_{ij} + \theta_i) \\ \sin(\theta_{ij} + \theta_i) \end{bmatrix} \Delta \theta \quad (27)$$

$\dot{\theta}_{ij}$ is computed in a similar way:

$$\begin{aligned} \dot{\theta}_{ij} &= \frac{d}{dt} \text{atan2}(\Delta y, \Delta x) = \frac{-\Delta y \Delta \dot{x} + \Delta x \Delta \dot{y}}{\Delta x^2 + \Delta y^2} \\ &\approx -\frac{V_{max} (\Delta x \cos \bar{\theta} + \Delta y \sin \bar{\theta})}{\|\Delta \mathbf{q}\|^2} \Delta \theta \\ &= -\frac{V_{max}}{\|\Delta \mathbf{q}\|} \begin{bmatrix} \cos \bar{\theta} \\ \sin \bar{\theta} \end{bmatrix}^T \begin{bmatrix} \cos(\theta_{ij} + \theta_i) \\ \sin(\theta_{ij} + \theta_i) \end{bmatrix} \Delta \theta \end{aligned} \quad (28)$$

Using the results above the terms in Equations 18 and 19 can be rewritten as:

$$-\beta_{a_j} \beta_{d_j} \frac{\dot{d}_{ij}}{d_{ij}} \approx A_1 \Delta \theta \quad (29)$$

$$\beta_{a_j} \frac{\partial \beta_{d_j}}{\partial d_{ij}} \dot{d}_{ij} \approx A_2 \Delta \theta \quad (30)$$

$$\beta_{d_j} \frac{\partial \beta_{a_j}}{\partial \theta_{ij}} \dot{\theta}_{ij} \approx A_3 \Delta \theta \quad (31)$$

with A_1, A_2 and A_3 defined as:

$$A_1 = -\frac{V_{max}}{d_{ij}} \beta_{a_j} \beta_{d_j} \begin{bmatrix} \sin \bar{\theta} \\ -\cos \bar{\theta} \end{bmatrix}^T \begin{bmatrix} \cos(\theta_{ij} + \theta_i) \\ \sin(\theta_{ij} + \theta_i) \end{bmatrix} \quad (32)$$

$$A_2 = V_{max} \beta_{a_j} \frac{\partial \beta_{d_j}}{\partial d_{ij}} \begin{bmatrix} \sin \bar{\theta} \\ -\cos \bar{\theta} \end{bmatrix}^T \begin{bmatrix} \cos(\theta_{ij} + \theta_i) \\ \sin(\theta_{ij} + \theta_i) \end{bmatrix} \quad (33)$$

$$A_3 = -\frac{V_{max}}{\|\Delta \mathbf{q}\|} \beta_{d_j} \frac{\partial \beta_{a_j}}{\partial \theta_{ij}} \begin{bmatrix} \cos \bar{\theta} \\ \sin \bar{\theta} \end{bmatrix}^T \begin{bmatrix} \cos(\theta_{ij} + \theta_i) \\ \sin(\theta_{ij} + \theta_i) \end{bmatrix} \quad (34)$$

The signs of A_1, A_2 and A_3 are determined by the result of the vector multiplication. As the scalar product can be related to the cosine of the angles between those vectors, the sign of A_1, A_2 and A_3 is determined by the angle $\theta_{ij} + \theta_i - \bar{\theta} \approx \theta_{ij}$. This is illustrated in Figure 4. Using Equations 33 and 34, Equation 18 can be rewritten as:

$$\begin{aligned} &\frac{d}{dt} [K (1 - \beta_{a_j} \beta_{d_j}) (\theta_t - \theta_i)] \\ &= -K \left[(1 - \beta_{a_j} \beta_{d_j}) \dot{\theta}_i + (A_2 + A_3) (\theta_t - \theta_i) \Delta \theta \right] \end{aligned} \quad (35)$$

Because θ_j is constant the term $(\theta_t - \theta_i)$ of Equation 35 can be rewritten as $(\theta_t - \theta_j + \theta_j - \theta_i) = (\theta_t - \theta_j - \Delta \theta)$. Using Assumption 3 as in Equation 26, the term $(A_2 + A_3) (\theta_t - \theta_i) \Delta \theta$ of Equation 35 can be written as:

$$\begin{aligned} &(A_2 + A_3) (\theta_t - \theta_i) \Delta \theta \\ &= (A_2 + A_3) (\theta_t - \theta_j - \Delta \theta) \Delta \theta \\ &= (A_2 + A_3) ((\theta_t - \theta_j) \Delta \theta - \Delta \theta^2) \\ &\approx (A_2 + A_3) (\theta_t - \theta_j) \Delta \theta \end{aligned} \quad (36)$$

because $\Delta \theta^2$ is small. We can now rewrite Equation 17 using the Assumption 4 that $\dot{\theta}_j = 0$ leading to $\Delta \dot{\theta} = \dot{\theta}$ and $\Delta \ddot{\theta} = \ddot{\theta}$:

$$\Delta \ddot{\theta} + a_1 \Delta \dot{\theta} + a_2 \Delta \theta \approx 0 \quad (37)$$

with a_1 and a_2 defined as:

$$a_1 = K (1 - \beta_{a_j} \beta_{d_j}) \quad (38)$$

$$a_2 = \left[(A_2 + A_3) BK + (A_1 + A_2 + A_3) \frac{\pi V_{max}}{d_{ij}} \right] \quad (39)$$

$$B = \theta_t - \theta_j \quad (40)$$

which has a stable solution if $\frac{-a_1 \pm \sqrt{a_1^2 - 4a_2}}{2} < 0$ which is true if both $a_1 > 0$ and $a_2 > 0$. The system is unstable otherwise. As a_1 is always positive, the stability of local minimum comes essentially from a_2 . The condition $a_2 > 0$ happens when $A_1, A_2 > 0, A_3 = 0$ and $B > 0$ which corresponds to the case where agent i sees the other agent on its front left and agent j is moving to the right with respect to the goal. Note that such analysis is not possible for agent j (in the front) because the sign of A_3 is different from A_1 and A_2 . This local minimum results from the combination of a limited field of view and both agents moving at the same speed. Indeed when near the local minimum, agent i has only little correction to perform to align with its desired heading. But to align, agent i needs to get closer to agent j , which would increase its avoidance reaction. As both agents have the same speed, the distance stays constant. Agent i is thus in a deadlock. The limited field of view leads to agent j (front one) to ignore agent i (in the rear of agent j). The solution will thus not come from agent j either. In the case of [10] the local minimum arises from their swirling effect reduction diminishing the radius of avoidance area at the rear of the aircraft. This is analogue to our algorithm where the front aircraft ignores the rear one, although for different reasons. This local minimum problem is solved by adding a change

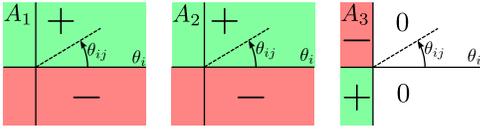


Fig. 4: Signs of A_1, A_2 and A_3 as function of θ_{ij} angles if $\theta_i = \theta_j$.

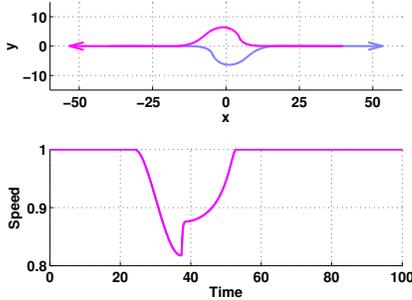


Fig. 5: (Top) Generated trajectories for proposed collision avoidance algorithm for two agents with limited FOV sensing. The arrow heads show the direction of motion. (Bottom) The speed curves for both agents. The two speed curves overlap.

in speed. As the agent behind is slower, the distance between the two will increase leading to their separation. For example a decrease in speed of 20% allows for a quick escape out of the local minimum as shown in Figure 3. The same decrease in speed can also be used with the algorithm presented in [10] to avoid the same problem. If both are exactly side-by-side and have their heading in the same direction, they will slow down equally. But because one agent is always on the left side of the other one (thus sees the later on its right), at least one of them is in an unstable situation ($A_1, A_2 \leq 0$ and $A_3 = 0$) and will separate itself if the sensors are noisy.

VI. SIMULATIONS

We performed microscopic, point-mass simulations with noiseless sensors to validate the results presented in this paper. Figure 5 shows the collision of two agents on a head on collision course. Even if the algorithm is designed with a complete U-turn in mind, the resulting trajectory is smooth. The parameters used are $R_p = 1$, $K_p = 20$ and $K = 0.6$. The agents are traveling at a speed included in the interval $[0.8; 1]$. Their radius is 1. The sensor range R_s is 30 and its FOV in $[-\frac{5\pi}{8}; \frac{5\pi}{8}]$. We also set $R_a = 4$.

Our algorithm is able to perform collision avoidance for more than two agents. Figure 6 shows an example where five agents want to cross the same point simultaneously.

VII. CONCLUSION

In this paper a collision avoidance algorithm for vehicles with unicycle dynamics and limited sensing capability is presented and proven effective. To the best of our knowledge, this is the first algorithm designed for agents with limited FOV sensors. We also show that such algorithm has a

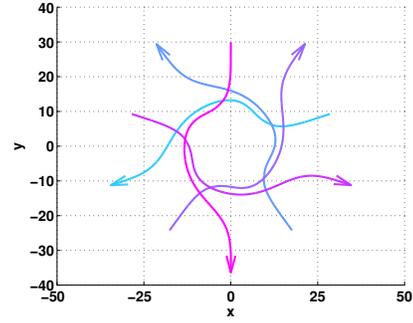


Fig. 6: Generated trajectories for proposed collision avoidance algorithm for five agents with limited FOV sensing. The arrow heads show the direction of motion.

possible local minimum that arises from the combination of a limited field of view and identical speed for both agents. We show that this effect is also present in at least one other algorithm found in the literature. A slight change in speed is shown to be an effective method to avoid this local minimum for both algorithms. Another possibility would be to consider the third dimension to avoid the local minimum. The effect of noise on the algorithm's is currently investigated with real UAVs.

REFERENCES

- [1] J. van den Berg, S. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Robotics Research*, ser. Springer Tracts in Advanced Robotics, 2011, vol. 70, pp. 3–19.
- [2] Y. I. Jenie, E.-J. Van Kampen, C. C. de Visser, and Q.-P. Chu, "Selective velocity obstacle method for cooperative autonomous collision avoidance system for UAVs," in *AIAA Guidance, Navigation, and Control Conference*, 2013, DOI:10.2514/6.2014-1472.
- [3] C.-K. Lai, M. Lone, P. Thomas, J. Whidborne, and A. Cooke, "On-board trajectory generation for collision avoidance in unmanned aerial vehicles," in *IEEE Aerospace Conference*, 2011, DOI:10.1109/AERO.2011.5747526.
- [4] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Trans. on Intelligent Transportation Systems*, vol. 1, no. 4, pp. 179–189, 2000.
- [5] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potential functions," *IEEE Trans. on Robotics and Automation*, vol. 8, no. 5, pp. 501–518, 1992.
- [6] 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*, Sept 2011, pp. 392–397.
- [7] C. Carbone, U. Ciniglio, F. Corraro, and S. Luongo, "A novel 3D geometric algorithm for aircraft autonomous collision avoidance," in *IEEE Conference on Decision and Control*, 2006, pp. 1580–1585.
- [8] G. Roussos, D. V. Dimarogonas, and K. J. Kyriakopoulos, "3d navigation and collision avoidance for nonholonomic aircraft-like vehicles," *International Journal of Adaptive Control and Signal Processing*, vol. 24, no. 10, pp. 900–920, 2010.
- [9] 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.
- [10] P. Panyakeow and M. Mesbahi, "Decentralized deconfliction algorithms for unicycle UAVs," in *American Control Conference*, June 2010, pp. 794–799.
- [11] —, "Deconfliction algorithms for a pair of constant speed unmanned aerial vehicles," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 50, no. 1, pp. 456–476, 2014.
- [12] 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.