Using 2D ORCA restrictions in MPC to avoid downwash disturbance in multi-UAV path-following ⋆

Arthur da C. Vangasse * **Lívia de M. C. M. Soares** **
Glauber R. Leite ** **Guilherme V. Raffo** *** **Luciano C. A. Pimenta** ***

* *Graduate Program in Electrical Engineering, Universidade Federal de Minas Gerais, MG (e-mail: vangasse@ufmg.br)*
** *Instituto de Computação, Universidade Federal de Alagoas, AL, (e-mail: {lmcms,glauber}@ic.ufal.br).*
*** *Department of Electronic Engineering, Universidade Federal de Minas Gerais, MG (e-mail: {raffo,lucpim}@ufmg.br)*

Abstract: This work presents an adaptation of the MPC-ORCA (Model Predictive Control - Optimal Reciprocal Collision Avoidance) navigation technique for multi-UAVs, implementing 2D ORCA constraints in the MPC problem to address the problem of disturbance caused by downwash in small UAVs. As a proof of concept, we showed a downwash disturbance scenario with two micro quadrotors in the real world, motivating our choice of 2D ORCA restrictions. Then, to evaluate the applicability of the proposed approach, we implemented the algorithm in two simulation groups, a simpler numerical simulation using only the equations of motion and a more complete simulation, covering the flow of air particles. The results of the simulations show that the proposed method effectively prevented the overlap of the vehicles, while attempting to maintain the agents' performance of the path-following task within a shared curve.

Keywords: Multi-vehicle systems, Cooperative navigation, Motion control, Trajectory Tracking and Path Following, Flying robots

1. INTRODUCTION

The applications for multiple coordinated robotic systems are vast, including fields such as hunting Fan and Liu (2023), patrolling Xiang et al. (2023), and cooperative search Yang et al. (2007), among others. Moreover, as interest in autonomous multi-robot systems increases, so does their complexity. Having adopted a long list of tasks, two core ones stand out: navigation and collision avoidance.

Vector fields are efficient strategies for tackling path-following problems associated to the navigation of various mobile robots, especially UAVs. The previous literature used the technique in quadrotor control strategies (Rezende et al., 2020a, 2021; Miranda et al., 2022; Nunes et al., 2022, 2023). The method provides an intuitive manner for specifying the desired motion, which maps every robot configuration onto a velocity reference. The work in Rezende et al. (2020b) suggests a better performance of the path controllers in practical situations when comparing path and trajectory controllers. In Aguiar et al. (2008), the authors demonstrate that trajectory tracking problems are subjected to a minimal $\mathscr{L}_2$ tracking error norm equal to the least amount of control energy needed to stabilize the zero dynamics of the error system. Then, they show that this limitation does not apply to less restrictive path-following problems. However, ensuring safety through collision avoidance is still necessary to enable multi-agent integration in vector field guided scenarios.

Velocity Obstacles (VO) address dynamic obstacles and propose a widely disseminated solution for collision avoidance (Fiorini and Shiller, 1993; Van den Berg et al., 2008). The method produces constraints in the velocity domain that must be met for collision-free navigation. In Van den Berg et al. (2011), the authors present Optimal Reciprocal Collision Avoidance (ORCA), which covers the problem of considering collision avoidance between agents. Their proposal offers fast geometric solutions that are suitable for critical dynamic environments.

Finally, Model Predictive Control (MPC) enables simultaneous reference tracking and constraint compliance. Leite et al. (2021) implemented MPC with ORCA restrictions for multiple differential drive vehicle scenarios, addressing the convex optimization problem through quadratic programming. Efficient multi-robot safe time-varying path-following schemes were formulated in Freitas et al. (2023); Vangasse et al. (2023) where different collision avoidance

⋆ This work was carried out with the support of the Coordination for the Improvement of Higher Education Personnel - Brazil (CAPES) through the Academic Excellence Program (PROEX) and Finance Code 88887.136349/2017-00. This work was also supported in part by the project INCT (National Institute of Science and Technology) under the grant CNPq (Brazilian National Research Council) 465755/2014-3 and FAPESP (São Paulo Research Foundation) 2014/50851-0, CNPq (grant numbers 407063/2021-8 and 309925/2023-1), and Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG), under grant number APQ-00630-23.

methods were addressed. These works are based on the MPC-ORCA elaborated by Cheng et al. (2017), which takes advantage of the receding horizon to improve evasive movements during collision avoidance.

However, another critical situation is encountered in multiple aerial robot scenarios. The creation of airflow below the drone characterizes the downwash phenomenon. This airflow produces disturbances that may compromise the stability of any neighbor crossing the affected area. Figure 1 is an example of this effect, where the lower robot should remain static in the center of the green sphere if it is not disturbed by the downwash caused by the upper vehicle.

The work in Arul and Manocha (2020) investigates the effects of downwash and the applicability of dynamic constraints in the MPC-ORCA problem. In that work, the robots' geometry is enveloped by an ellipsoid of adaptable height. However, with small UAVs sharing a cluttered workspace, the downwash can easily compromise their stability, so vertical alignment between UAVs should be strictly avoided.

This work proposes an adaptation of the MPC-ORCA formulation, where the optimization problem still attempts to solve a safe 3D path-following problem while implementing 2D ORCA constraints. This scheme should enable a group of robots to follow the same path and avoid the undesired conditions of collisions and downwash.

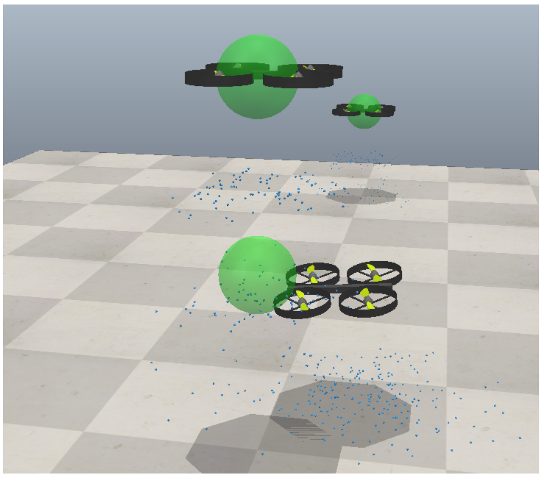


Figure 1. Demonstration of downwash produced by the upper UAV presenting disturbance to the bottom UAV.

2. TIME-VARYING VECTOR FIELDS

To compute a time-varying vector field $\Psi(\boldsymbol{p},t) : \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}^n$, where $\boldsymbol{p} \in \mathbb{R}^n$ is a point so that the trajectories of the system $\dot{\boldsymbol{p}} = \Psi(\boldsymbol{p},t)$ converge to and follow the curve $C(s,t)$, the work Rezende et al. (2021) proposes the vector field as follows

$$\begin{aligned}\Psi(\boldsymbol{p},t) = &-\eta(\boldsymbol{p},t)\hat{G}(\boldsymbol{p},t)\frac{\boldsymbol{D}(\boldsymbol{p},t)}{D(\boldsymbol{p},t)}\\ &+\eta(\boldsymbol{p},t)\hat{H}(\boldsymbol{p},t)\boldsymbol{T}^*(\boldsymbol{p},t)\\ &+\Pi_{\boldsymbol{T}}(\boldsymbol{p},t)\frac{\partial \boldsymbol{D}}{\partial t}(\boldsymbol{p},t),\end{aligned} \tag{1}$$

where $\eta(\boldsymbol{p},t) : \mathcal{U} \to \mathbb{R}$ is defined so that Ψ has a norm equal to the value of a desired velocity to follow the vector field, v_r. $D(\boldsymbol{p},t)$ is the smallest distance between the robot position $\boldsymbol{p}$ and the curve $C(s,t)$. The vector $\mathbf{D}(\boldsymbol{p},t) = \boldsymbol{p} - s^*(\boldsymbol{p},t)$ represents the distance vector between the robot and the nearest point on the curve $s^*(\boldsymbol{p},t)$. In turn, $\boldsymbol{T}^*(\boldsymbol{p},t)$ represents the tangent vector of the curve on the closest point, and the null space projection of $\boldsymbol{T}^*$, $\Pi_{\boldsymbol{T}}(\boldsymbol{p},t) : \mathcal{U} \to \mathbb{R}^{n\times n}$ is given by:

$$\Pi_{\boldsymbol{T}}(\boldsymbol{p},t) = \boldsymbol{I}_{n\times n} - \boldsymbol{T}^*(\boldsymbol{p},t)\boldsymbol{T}^*(\boldsymbol{p},t)^T, \tag{2}$$

where $\boldsymbol{I}_{n\times n}$ is the identity matrix of order n.

In order to guarantee that the vector field has a constant norm v_r, i.e. $\|\Psi(\boldsymbol{p},t)\| = v_r$, it is required for the curve $C(s,t)$ to be slow-varying on time when compared to the robot speed v_r. As Rezende et al. (2021) suggests, the functions $\hat{G} = 2\pi\text{arctan}(k_f D(\boldsymbol{p}))$ and $\hat{H} = \sqrt{1-\hat{G}^2}$ guarantee asymptotic convergence and circulation of the curve, where $k_f > 0$.

3. OPTIMAL RECIPROCAL COLLISION AVOIDANCE

In a multi-agent scenario where $\boldsymbol{p}_i$ and $\boldsymbol{v}_i$, respectively, represent the position and velocity of the agent named $i \in \Upsilon = \{1,2,3...n\}$. At this point, it is worth highlighting that the original ORCA concept considers simple integrator dynamics. Therefore, in the context of this work, v_i can no longer be called the desired velocity but the instantaneous velocity. The velocity obstacle $VO^{\tau}_{i|j}$ induced by every other agent, named $j \neq i$, in a time frame $[0,\tau]$, is defined as

$$VO^{\tau}_{i|j} = \left\{\boldsymbol{v} \mid \exists t \in [0,\tau] :: \boldsymbol{v} \in R\left(\tfrac{\boldsymbol{p}_j - \boldsymbol{p}_i}{t}, \tfrac{r_i + r_j}{t}\right)\right\}, \tag{3}$$

in which $R(\boldsymbol{p},r)$ returns an open sphere centered in $\boldsymbol{p}$ of radius r in the $\mathbb{R}^3$ space Van den Berg et al. (2008).

At the occasion when a pair of agents produces a relative velocity $\boldsymbol{v}_{i|j}$ within their corresponding velocity obstacle, i.e. $\boldsymbol{v}_{i|j} \in VO^{\tau}_{i|j}$, a collision may happen in the given time interval τ. Then collisions can be avoided, if both agents deviate equally by imposing a relative velocity $\boldsymbol{w}_{i|j} + \boldsymbol{v}_{i|j} \notin VO^{\tau}_{i|j}$. Therefore, let $\boldsymbol{w}_{i|j}$ be the smallest vector that takes $\boldsymbol{v}_{i|j}$ to the closest point on the border of the velocity obstacle, $\partial VO^{\tau}_{i|j}$, obtained as

$$\boldsymbol{w}_{i|j} = \left(\underset{\boldsymbol{v}\in\partial VO^{\tau}_{i|j}}{\text{argmin}} \|\boldsymbol{v} - (\boldsymbol{v}_i - \boldsymbol{v}_j)\|\right) - (\boldsymbol{v}_i - \boldsymbol{v}_j), \tag{4}$$

which has a closed form geometric solution for disc-shaped agents. This solution is found by considering the conic shape truncated by a circle of $VO^{\tau}_{i|j}$, briefly discussed in Van den Berg et al. (2011). By checking if $\boldsymbol{v}_{i|j}$ lies closer to the spherical or conical shell, it is possible to deduce $\boldsymbol{w}_{i|j}$.

Then, let $\hat{\boldsymbol{n}}$ be the normal vector of the velocity obstacle border at $\boldsymbol{w}_{i|j} + \boldsymbol{v}_{i|j}$. Since both agents share the responsibility for collision avoidance (Van den Berg et al., 2011), the $ORCA^{\tau}_{i|j}$ space of admissible velocities that guarantee safe navigation is

$$ORCA^{\tau}_{i|j} = \left\{\boldsymbol{v} \middle| \left[\boldsymbol{v} - \left(\boldsymbol{v}_i + \frac{\boldsymbol{w}_{i|j}}{2}\right)\right] \cdot \hat{\boldsymbol{n}} \geq 0\right\}. \tag{5}$$

Based on the above constraint formulation, the overall set $ORCA^{\tau}_i$ of admissible velocities for an agent i, while considering multiple neighbors, is composed of the intersection of the $ORCA^{\tau}_{i|j}$ half-spaces,

$$ORCA^{\tau}_{i} = \bigcap_{j=1\dots n|j\neq i} ORCA^{\tau}_{i|j}. \tag{6}$$

It is worth emphasizing that the solution described above is capable of solving problems in three-dimensional spaces, usually related to aerial vehicles. However, our solution narrows the problem onto a two-dimensional plane, to avoid downwash problems.

4. MODEL PREDICTIVE CONTROL

In this work, we represent the robotic agents with double integrator dynamics. The state vector of each agent i is described as $\boldsymbol{x}_i(t) = [\boldsymbol{p}_i(t)\ \boldsymbol{v}_i(t)]^T$, in which $\boldsymbol{p}_i(t), \boldsymbol{v}_i(t) \in \mathbb{R}^3$ are respectively position and velocity in an instant t. Acceleration inputs are represented by $\boldsymbol{u}_i(t)$, considering the double integrator dynamic model:

$$\begin{aligned} &\ddot{\boldsymbol{x}}_i(t) = \boldsymbol{u}_i(t), \\ &\boldsymbol{u}_{min} \leq \boldsymbol{u}_i(t) \leq \boldsymbol{u}_{max}, \end{aligned} \tag{7}$$

where u_{min} and u_{max} denote the minimum and maximum acceleration input values, respectively. This is a suitable model to represent simplified dynamics of qadrotors assuming holonomy.

The Distributed MPC problem formulated in this work and solved in every iteration by each agent i, is given by

$$\begin{aligned} \boldsymbol{u}^* = \underset{\boldsymbol{u}_i}{\operatorname{argmin}} & \sum_{k=0}^{N-1} ||\boldsymbol{x}_i(k) - \hat{\boldsymbol{x}}_i(k)||^2_{\boldsymbol{Q}} + ||\boldsymbol{u}_i(k)||^2_{\boldsymbol{R}} \\ & + \sum_{k=1}^{N-1} ||\boldsymbol{u}_i(k) - \boldsymbol{u}_i(k-1)||^2_{\boldsymbol{\Delta R}} \\ \text{s.t. } & \boldsymbol{x}_i(k+1) = A\boldsymbol{x}_i(k) + B\boldsymbol{u}_i(k), \\ & \boldsymbol{u}_{min} \leq \boldsymbol{u}_i(k) \leq \boldsymbol{u}_{max}, \\ & \boldsymbol{v}_{min} \leq \boldsymbol{v}_i(k) \leq \boldsymbol{v}_{max}, \\ & \boldsymbol{x}_i(0) = \boldsymbol{x}_{i_0}, \\ & \boldsymbol{v}_{(x,y),i}(k+1) \in ORCA^{\tau}_i(k+1). \end{aligned} \tag{8}$$

where the proposed cost functional assumes a quadratic form of components $\boldsymbol{x}_i(k) - \hat{\boldsymbol{x}}_i(k)$ of error, $\boldsymbol{u}_i(k)$ the control action applied at instant k and $\boldsymbol{u}_i(k) - \boldsymbol{u}_i(k-1)$ the input variation over two instants. The variable $\boldsymbol{x}_i(k)$ represents the state of the system at step k of the prediction horizon of size N, $\hat{\boldsymbol{x}}_i(k)$ is the reference composed of the integration of the vector field Ψ in (1), and by the field itself, that is to say

$$\hat{\boldsymbol{x}}_i(k+1) = \begin{bmatrix} \hat{\boldsymbol{p}}(k+1) \\ \hat{\boldsymbol{v}}(k+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}(k) + \boldsymbol{\Psi}(\boldsymbol{p}, k)T_s \\ \boldsymbol{\Psi}(\boldsymbol{p}, k+1) \end{bmatrix}, \tag{9}$$

where $\hat{\boldsymbol{p}}$ and $\hat{\boldsymbol{v}}$ are reference positions and velocities respectively. The gain matrix $\boldsymbol{Q}$ is positive semi-definite. It punishes deviations from the reference, as the positive definite gain matrix $\boldsymbol{R}$ punishes large control efforts, and $\boldsymbol{\Delta R}$ significant variations on control actions.

The constraints imposed in the problem are described in the order of appearance. The first-order Euler's discretization of the double integrator dynamic model gives

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & 0 & T_s & 0 & 0 \\ 0 & 1 & 0 & 0 & T_s & 0 \\ 0 & 0 & 1 & 0 & 0 & T_s \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{B} = \begin{bmatrix} 0.5T_s^2 & 0 & 0 \\ 0 & 0.5T_s^2 & 0 \\ 0 & 0 & 0.5T_s^2 \\ T_s & 0 & 0 \\ 0 & T_s & 0 \\ 0 & 0 & T_s \end{bmatrix} \tag{10}$$

with the sampling time T_s. Admitted control actions are given by the interval $[\boldsymbol{u}_{min}, \boldsymbol{u}_{max}]$ and admitted velocities by $[\boldsymbol{v}_{min}, \boldsymbol{v}_{max}]$. The initial state, set to $\boldsymbol{x}_{i_0}$, is the current state of the system.

The collision avoidance restrain the admissible space of the velocity states associated to the horizontal plane represented by $\boldsymbol{v}_{(x,y),i}(k+1)$. We resort to the approximation elaborated in Cheng et al. (2017). The propagation of the collision avoidance constraints over the prediction horizon is implemented by computing relative velocities and positions at the instant k using the propagated states $\bar{\boldsymbol{p}}_i(k)$ and $\bar{\boldsymbol{v}}_i(k)$ of all agents assuming constant velocity, written as:

$$\begin{aligned} &\bar{\boldsymbol{v}}_i(k) = \boldsymbol{v}_i(0) \\ &\bar{\boldsymbol{p}}_i(k) = \bar{\boldsymbol{p}}_i(k-1) + \bar{\boldsymbol{v}}_i(k-1)T_s \end{aligned} \tag{11}$$

with $\bar{\boldsymbol{p}}_i(0) = \boldsymbol{p}_i(0)$. Then, the resulting relative states provide the parameters required to compute Equation (6) at each instant k of the horizon.

5. IMPLEMENTATION

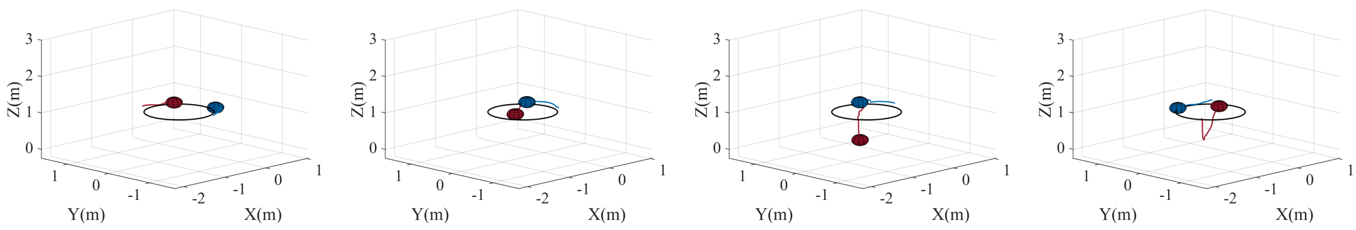
For this study, we developed three major implementation scenarios. The first is a scenario is reserved for a proof of concept and does not use the proposed solution, it reproduces the downwash problem in a real-world experiment. The scenario used the Crazyflie 2.1 micro quadcopters [1]. We also used Optitrack [2] for accurate motion tracking, Crazyradio 2.0 [3] for wireless communication, and CrazySwarm (Preiss* et al., 2017) within the Robot Operating System (ROS) Noetic (Quigley et al., 2009) framework. All of these components were hosted on a computer powered by an Intel i9 14900K processor.

To evaluate the application of the proposed technique, we implemented a set of simulations consisting of the multi-UAV coordinate motion. These simulations were based on the expected motion of the robots, which are defined as flying particles enveloped by a sphere with a defined radius in a shared workspace. We implemented simulations on multi-robot configurations with 8 UAVs, watching for situations of robot overlapping. The focus is on basic coordination and collision avoidance in a controlled environment.

Additionally, a physics simulation was conducted using CoppeliaSim through the ZeroMQ API and the Bullet 2.78 physics engine. This simulation focused on the physical interactions between UAVs and how they impact the surrounding environment, where the downwash problem could potentially affect the trajectory of other UAVs.

The developed algorithms were implemented in Python, using the CasADi library (Andersson et al., 2019) for

[1] https://www.bitcraze.io/products/crazyflie-2-1/
[2] https://optitrack.com/
[3] https://www.bitcraze.io/products/crazyradio-2-0/



(a) 19 seconds (b) 23 seconds (c) 25 seconds (d) 28 seconds

Figure 2. Downwash effect caused by the blue robot in a real-world scenario, affecting the motion of the red robot.

numerical optimization. Our implementation resorts to the Interior Point Optimizer (IPOPT) for solving constrained optimization problems. The simulations were conducted on a computer system equipped with an AMD Ryzen 9 5900HX processor.

The following scenarios assumed for the MPC problem in (8), $\boldsymbol{Q} = \boldsymbol{R} = I_{6\times 6}$. The sample time was set to $T_s =$ 50ms, the prediction horizon $N = 10$, the vector field's desired velocity $v_r = 0.1$m/s and the ORCA parameter $\tau = 0.5$s. Finally, in our simulations, robots follow the curve

$$C(s,t) = \begin{bmatrix} -\cos(s) \\ 0.2\sin(2s)\cdot\sin(0.002\pi t) \\ -\sin(s) \end{bmatrix}. \qquad (12)$$

6. RESULTS

In this section, we present three blocks of results. First, we contemplate the real-world experiment and the downwash problem detection. Then, we discuss the results of the simulation based on the equations of motion of the robots and the physics-based simulation, both of which use the proposed approach.

6.1 The Downwash Problem

The Crazyflie 2.1 are micro quadrotors that weigh 27 grams. During experiments using MPC-ORCA with spheric constraints as proposed in Freitas et al. (2023), downwash scenarios were frequently observed, especially during evasive movements that placed the robots on top of each other. In these scenarios, the resulting disturbances posed great risks to the system's stability.

Figure 2 illustrates a series of frames of a real-world experiment. A pair of robots navigate in opposite directions over the same curve, a challenging scenario with concurrent objectives. At 19 seconds, both agents have converged and navigated over the curve. Then, before collision imminence at 23 seconds, the evasive maneuver starts, Fig. 3 better illustrates the resulting movement which positions both robots one over the other. At 25 seconds, it is noticeable that the evasion caused a downwash effect. Fortunately, at the 28-second mark, both robots were able to recover the path-following behavior. A video is available in `https://youtu.be/S_1i6o11uK0?t=1` for more detail.

6.2 Motion Simulation Results

Our first results regarding the proposed solution seek to demonstrate its capabilities in crowded scenarios. We

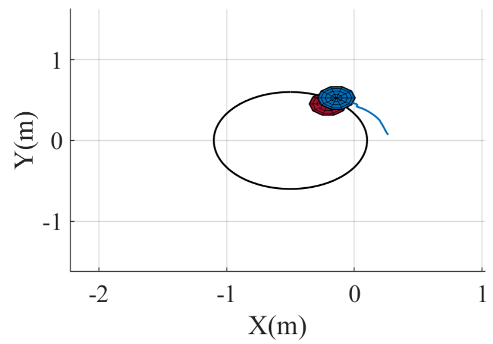
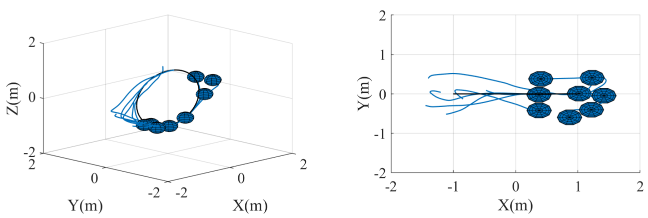


Figure 3. Top view of the moment where the red robot is affected by downwash in the real-world scenario.

proposed the same curve as a reference to 8 agents. Figure 4 shows the curve in black, a circle on the xz-plane, chosen due to its vertical movement-inducing shape. The agents in blue follow the curve closely during horizontal movements depicted at the 32-second mark. Then, when approaching the vertical movements, a bottleneck is observed, and the agents are forced to distance themselves from the curve. Finally, the agents were able to resume convergence to the curve, having avoided collisions and downwash-inducing positioning throughout the whole movement. For more detail, vist the viedo at: `https://youtu.be/S_1i6o11uK0?t=150`



(a) 32 seconds (b) 42 seconds

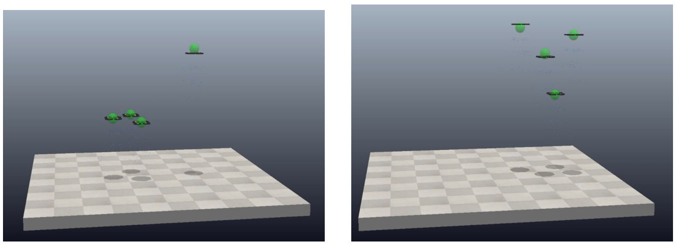
Figure 4. Perspective and top views of the 8 agents motion simulation, showing no overlap between robots.

6.3 Physics Simulation Results

To demonstrate the applicability of the method, we improved the scenario to simulate downwash effects. Figure 5 shows two pairs of drones assigned to the simulation. In this simulation, robots follow the same curve (12) while producing particles that can knock down their neighbors positioned below. It is worth mentioning that each pair runs in a different direction of the curve.

ISSN: 2525-8311

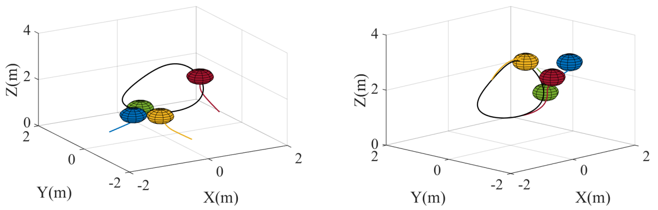
DOI: 10.20906/CBA2024/4478



(a) 25 seconds (b) 130 seconds

Figure 5. Timestamps of the path-following task in the physics-based simulation.

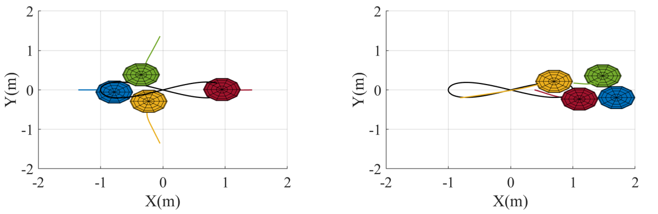
Figure 6 shows the robots positioning around the curve in black; the pair of blue and red follows the curve in a counterclockwise direction, and the remaining agents in the opposite direction. The images show the bottlenecks during vertical movements. The movement is recorded in video at: `https://youtu.be/S_1i6o11uK0?t=30`.



(a) 25 seconds (b) 130 seconds

Figure 6. Perspective view of the path-following task in the physics-based simulation.

In Figure 7 it is evident that no overlap occurred. In general, the agents were able to follow the curve closely, except during times of jamming when avoiding collisions and overlaps. A video of the complete simulation is available at: `https://youtu.be/S_1i6o11uK0?t=210`.



(a) 25 seconds (b) 130 seconds

Figure 7. Top view of the path-following task in the physics-based simulation.

7. CONCLUSION

This work addresses the multi-robot safe path-following problem. To offer safety, the solution tackles downwash and collision avoidance simultaneously by implementing 2D ORCA constraints in the Distributed MPC problem. Path-following in its turn is accomplished simultaneously by many robots by leveraging the Artificial Vector Fields from Rezende et al. (2021).

Moving forth from Freitas et al. (2023), the downwash problem was elucidated through a real-world experiment using two micro quadrotors. The Crazyflie 2.1 lightweight structure showed susceptibility to disturbances motivating our choice of ORCA constraints based on disc-shaped robots reduced to the horizontal plane.

Our solution offers simplicity, having ruled out the possibility of benefiting from the advantages of more complex solutions such as Arul and Manocha (2020). On the other hand, we studied the behavior of the constraints simulated by Cheng et al. (2017), having extrapolated the path following by one dimension.

Although simple and effective, the proposed measure has limitations. Since the occupancy area of a UAV in the computation of the 2D ORCA constraints can be interpreted as a cylinder with infinite height, dense regions that require more vertical motion may lead to conservative movements and push the robots away from the target curve.

For further research, we aim to experiment the proposed solution in our real robot environment. Also, this work can be expanded and scaled for swarms of heterogeneous robots, presenting different sizes and mobility capabilities, which may require adaptations in the ORCA shared collision avoidance responsibility policy.

REFERENCES

Aguiar, A.P., Hespanha, J.P., and Kokotović, P.V. (2008). Performance limitations in reference tracking and path following for nonlinear systems. *Automatica*, 44(3), 598–610. doi:https://doi.org/10.1016/j.automatica.2007.06.030. URL `https://www.sciencedirect.com/science/article/pii/S0005109807003482`.

Andersson, J.A., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). Casadi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11, 1–36.

Arul, S.H. and Manocha, D. (2020). Dcad: Decentralized collision avoidance with dynamics constraints for agile quadrotor swarms. *IEEE Robotics and Automation Letters*, 5(2), 1191–1198. doi:10.1109/LRA.2020.2967281.

Cheng, H., Zhu, Q., Liu, Z., Xu, T., and Lin, L. (2017). Decentralized navigation of multiple agents based on orca and model predictive control. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 3446–3451. doi:10.1109/IROS.2017.8206184.

Fan, S. and Liu, H.H.T. (2023). Multi-uav cooperative hunting in cluttered environments considering downwash effects. *Guidance, Navigation and Control*, 3(01), 2350004.

Fiorini, P. and Shiller, Z. (1993). Motion planning in dynamic environments using the relative velocity paradigm. *Proceedings IEEE International Conference on Robotics and Automation*, 560–565. doi:10.1109/robot.1993.292038.

Freitas, E.J.R., Vangasse, A.D.C., Raffo, G.V., and Pimenta, L.C.A. (2023). Decentralized multi-robot collision-free path following based on time-varying artificial vector fields and mpc-orca. In *2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE)*, 212–217. doi:10.1109/LARS/SBR/WRE59448.2023.10333004.

Leite, G.R., da C. Vangasse, A., Ícaro B. Q. Araújo, and Savino, H.J. (2021). Collision avoidance with differential drive robots using mpc-orca. In *Procedings do XV Sim-*

pósio Brasileiro de Automação Inteligente, SBAI2021. SBA Sociedade Brasileira de Automática. doi:10.20906/sbai.v1i1.2774. URL `http://dx.doi.org/10.20906/sbai.v1i1.2774`.

Miranda, V.R., Rezende, A., Rocha, T.L., Azpúrua, H., Pimenta, L.C., and Freitas, G.M. (2022). Autonomous navigation system for a delivery drone. *Journal of Control, Automation and Electrical Systems*, 33(1), 141–155. doi:https://doi.org/10.1007/s40313-021-00828-4.

Nunes, A.H.D., Raffo, G.V., and Pimenta, L.C.A. (2023). Integrated vector field and backstepping control for quadcopters. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 1256–1262. doi:10.1109/ICRA48891.2023.10160824.

Nunes, A.H.D., Rezende, A.M.C., Cruz, G.P., Freitas, G.M., Gonçalves, V.M., and Pimenta, L.C.A. (2022). Vector field for curve tracking with obstacle avoidance. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, 2031–2038. doi:10.1109/CDC51059.2022.9992435.

Preiss*, J.A., Hönig*, W., Sukhatme, G.S., and Ayanian, N. (2017). Crazyswarm: A large nano-quadcopter swarm. In *IEEE International Conference on Robotics and Automation (ICRA)*, 3299–3304. IEEE. doi:10.1109/ICRA.2017.7989376. URL `https://doi.org/10.1109/ICRA.2017.7989376`. Software available at `https://github.com/USC-ACTLab/crazyswarm`.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A.Y., et al. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, 5. Kobe, Japan.

Rezende, A.M.C., Goncalves, V.M., and Pimenta, L.C.A. (2021). Constructive time-varying vector fields for robot navigation. *IEEE Transactions on Robotics*, 38(2), 852–867. doi:10.1109/TRO.2021.3093674.

Rezende, A.M.C., Gonçalves, V.M., Nunes, A.H.D., and Pimenta, L.C.A. (2020a). Robust quadcopter control with artificial vector fields. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 6381–6387. IEEE. doi:10.1109/ICRA40945.2020.9196605.

Rezende, A.M.C., Júnior, G.P.C., Fernandes, R., Miranda, V.R.F., Azpúrua, H., Pessin, G., and Freitas, G.M. (2020b). Indoor localization and navigation control strategies for a mobile robot designed to inspect confined environments. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, 1427–1433. doi:10.1109/CASE48305.2020.9217005.

Van den Berg, J., Guy, S.J., Lin, M., and Manocha, D. (2011). Reciprocal n-body collision avoidance. In C. Pradalier, R. Siegwart, and G. Hirzinger (eds.), *Robotics Research*, 3–19. Springer Berlin Heidelberg, Berlin, Heidelberg. doi:10.1007/978-3-642-19457-3_1.

Van den Berg, J., Ming Lin, and Manocha, D. (2008). Reciprocal Velocity Obstacles for real-time multi-agent navigation. *2008 IEEE International Conference on Robotics and Automation*, 23, 1928–1935. doi:10.1109/ROBOT.2008.4543489. URL `http://ieeexplore.ieee.org/document/4543489/`.

Vangasse, A.D.C., Raffo, G.V., and Pimenta, L.C.A. (2023). Mpc-cbf strategy for multi-robot collision-free path-following. In *2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE)*, 284–289. doi:10.1109/LARS/SBR/WRE59448.2023.10332984.

Xiang, H., Han, Y., Pan, N., Zhang, M., and Wang, Z. (2023). Study on multi-uav cooperative path planning for complex patrol tasks in large cities. *Drones*, 7(6), 367.

Yang, Y., Polycarpou, M.M., and Minai, A.A. (2007). Multi-uav cooperative search using an opportunistic learning method. *Journal of Dynamic Systems, Measurement, and Control*, 129(5), 716–728.