Collision avoidance effects on the mobility of a UAV swarm using chaotic ant colony with model predictive control.

DENTLER, J., ROSALIE, M., DANOY, G., BOUVRY, P., KANNAN, S., OLIVARES-MENDEZ, M.A. and VOOS, H.

2019

Collision avoidance effects on the mobility of a UAV swarm using Chaotic Ant Colony with Model Predictive Control

Jan Dentler, Martin Rosalie, Grégoire Danoy,
Pascal Bouvry, Somasundar Kannan, Miguel A.
Olivares-Mendez and Holger Voos

Abstract The recent development of compact and economic small Unmanned Aerial Vehicles (UAVs) permits the development of new UAV swarm applications. In order to enhance the area coverage of such UAV swarms, a novel mobility model has been presented in previous work, combining an Ant Colony algorithm with chaotic dynamics (CACOC). This work is extending CACOC by a Collision Avoidance (CA) mechanism and testing its efficiency in terms of area coverage by the UAV swarm. For this purpose, CACOC is used to compute UAV target waypoints which are tracked by model predictively controlled UAVs. The UAVs are represented by realistic motion models within the virtual robot experimentation platform (V-Rep). This environment is used to evaluate the performance of the proposed CACOC with CA algorithm in an area exploration scenario with 3 UAVs. Finally, its performance is analyzed using metrics.

1 Introduction

Over the last decade, the usage of UAVs has been strictly increasing in areas like the construction industry, agriculture, defense and security. The advantages of UAVs are their remote sensing capabilities as well as their mobility and related extended operational space, which are ideally suited for area exploration tasks. For instance, the response time in search and rescue scenarios within large areas can be reduced significantly by using multiple autonomously operating UAVs. However, the coverage of the surveyed area is dependent on the coordination of the UAVs. A comprehensive overview on mechanisms to coordinate and control such swarms are given in [12] and [3]. The detailed state of the art in multi-UAV area exploration from a perspective of optimization and artificial intelligence is summarized in

Jan Dentler, Martin Rosalie, Somasundar Kannan and Miguel A. Olivares-Mendez
Interdisciplinary Centre for Security, Reliability and Trust (SnT), University of Luxembourg, 29, avenue JF Kennedy, L-1855 Luxembourg, Luxembourg, E-mail: jan.dentler@uni.lu, E-mail: martin.rosalie@uni.lu, E-mail: somasundar.kannan@uni.lu and E-mail: miguel.olivaresmendez@uni.lu

Grégoire Danoy and Pascal Bouvry
FSTC–CSC–ILIAS, University of Luxembourg, 2, avenue de l'Université, Esch-sur-Alzette, Luxembourg, E-mail: gregoire.danoy@uni.lu and E-mail: pascal.bourvy@uni.lu

Holger Voos, SnT–RUES, University of Luxembroug, 2, avenue de l'Université, Esch-sur-Alzette, Luxembourg, E-mail: holger.voos@uni.lu
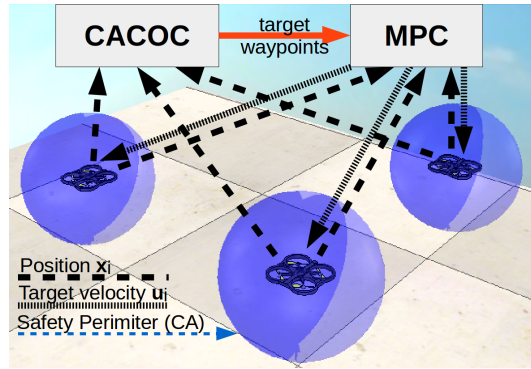
Fig. 1 Use case: Implementation of area exploration scenario. CACOC gives UAV waypoints to the MPC which controls UAV models in order to simulate a realistic closed-loop system.

[20]. A typical approach which is covered in all of these contributions is a cooperative area exploration control with potential fields over a receding horizon. The nature inspired interpretation of this method has been first introduced in [11] as Ant Colony Optimization with pheromones to coordinate a swarm of UAVs. This deterministic path planning is differing from Dynamic Data Driven Application Systems (DDDAS, see [19] for an example applied on a swarm of UAVs), where the UAVs coordinate according to the data they collected.

Kuiper and Nadjm-Tehrani [15] define an area exploration scenario with ten UAVs using a pheromone based approach for path planning. Their study compares a random mobility model to a pheromone-based mobility model using metrics for coverage. Some recent studies propose to replace the random part of such metaheuristics by chaotic dynamics to increase their performance, e.g. for Particle Swarm [10,22] or Firefly Optimization [9]. Thus, previous work [26] focused on the usage of chaotic dynamics to create a Chaotic Ant Colony Optimization to Coverage (CACOC) mobility model in order to increase the area coverage performance of a standard Ant Colony algorithm. However, the high-level simulations used to evaluate CACOC do not take into account real quadrotor dynamics. For this reason in [27] CACOC has been evaluated in an area exploration scenario using ten realistic UAV models, which have been introduced in the authors previous work [7]. However, in [27] the UAVs are operating on different altitudes in order to avoid collisions. In reality this is not always feasible due to airspace and physical limitations and furthermore leads to inhomogeneous sensor perception area sizes and ground resolutions.

The main contribution of this work is addressing this limitation by extending the CACOC mobility model by a CA (CACOC+CA) mechanism. In addition, the paper analyses the performance of the proposed approach in a realistic use case exploration scenario with three UAVs, as illustrated in Fig.1. The realistic UAV behavior is modelled by using an identified model of real quadrotor dynamics in combination with a MPC position control, as introduced in the authors previous publication [7]. Furthermore, the in [7] presented CA technique for static obstacles is applied within this work to avoid collision between UAVs (MPC+CA). The result are realistic closed-loop UAV dynamics which reflect the position tracking with MPC of waypoints that are generated with the CACOC mobility model. Based on this realistic UAV behavior, this work is evaluating the efficiency of the proposed CACOC+CA with MPC+CA approach by means of coverage and CA metrics. For this purpose, new metrics are introduced to evaluate the CA capabilities of the utilized models in long-term simulations. These simulation serve as preliminary analysis steps before the deploy-

ment of real UAVs is possible. Additionally, this paper provides more details about chaotic dynamics and the CACOC mobility model.

This article is organized as follows. In section 2 the description of the CACOC UAV mobility model along with a CA mechanism is provided. This mobility model includes chaotic dynamics to obtain unpredictable and structured trajectories. Furthermore, the related state of the art and implementation details are given. Section 3 describes the MPC of UAVs including a CA mechanism as well as the underlying dynamic model of a UAV. The experimental scenario combining CACOC and MPC is given in section 4. This includes an evaluation of the proposed methods and a description of the metrics utilized for the statistical analysis. Section 5 is concluding the presented work and outlines future research directions.

2 UAV mobility models

This section first provides a brief description of the CACOC mobility model used to generate the target waypoints for the UAVs which lead to an optimal area exploration. This mobility model has been introduced in [26, 27]. Its algorithm and main properties are detailed within this section. Subsequently, we associate a theoretical CA method to the CACOC mobility model to enhance the capability of the swarm in order to allow the UAVs to fly at the same altitude.

2.1 Chaotic Ant Colony Optimization to Coverage (CACOC)

Typical applications of UAV swarms are area surveillance, search and rescue, etc. The advantage of swarms in such scenarios is the distributed sensing capability, which is further enhanced by the movements of each individual UAV. In order to fully exploit these advantages, the UAV movement has to be coordinated. A traditional method to optimize the area coverage is pheromone based Ant Colony Optimization (ACO). In order to improve the performance of this approach, previous work [26] has combined ACO with the chaotic behavior of a dynamical system resulting in the CACOC mobility model. More precisely, CACOC replaces the random parts of the ant colony optimization algorithm [8] by a chaotic behavior in such a way that the exploration capabilities of the UAV swarm is improved. The paper presented a high level simulation process in which CACOC has been used to describe the mobility of a swarm of ten UAVs. Its performance was compared to existing models using random, chaos and classical ant colony optimization algorithms [26]. The results demonstrate empirically that CACOC permits an increase in performance for several aspects of coverage (overall coverage, recent coverage and fairness of the coverage). The CACOC performance on the coverage problem was empirically demonstrated using metrics as described in Sec. 4.

This section provides a brief description of this mobility model and its extension regarding CA. For more details the interested reader should refer to [26]. We will first present a mobility model including chaotic dynamics (CROMM) before introducing CACOC that combines both chaotic dynamics with an Ant Colony exploration algorithm. One approach to generate chaotic dynamics is to use the Rössler system [28]. It is a well-known dynamical system that can exhibit various chaotic mechanisms [25].

The UAVs are considered to have a constant speed and choose a movement direction at each discrete time step : A for ahead, R for 45° right and L for 45° left. Then, the next choice is given by the first return map (Fig. 2). This map underlines the dynamical signature

of the Rössler system giving ρ_{n+1} versus ρ_n. Thus, the next action depends on the previous one:

- if $\rho_n < 1/3$ then direction is right (R);
- if $1/3 \leqslant \rho_n < 2/3$ then direction is left (L);
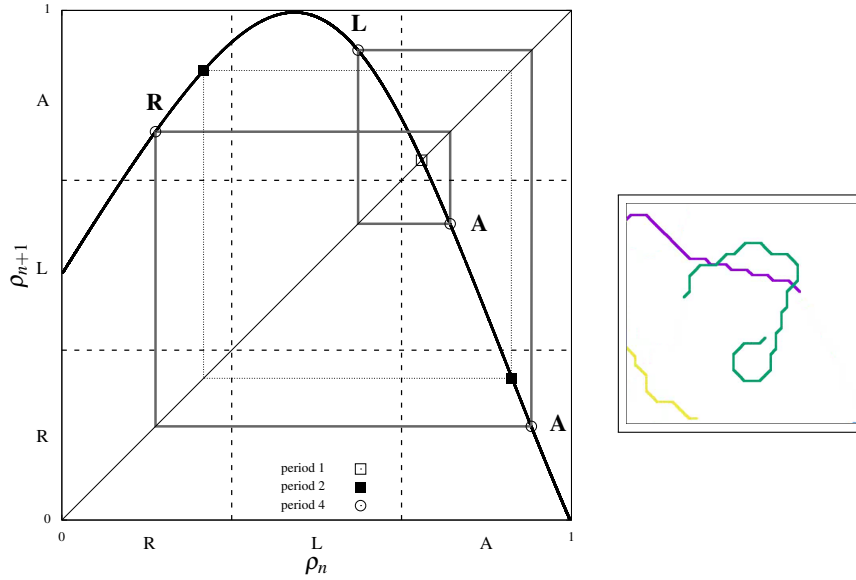- else the direction is ahead (A).

Fig. 2 Details of the CACOC mobility model with the first return maps from the Rössler system and the pattern it generates when there is no pheromone to guide the UAVs. The next choice depends on the previous one with: A for ahead, L for left and R for right. The period 1 orbits leads to symbols AAAAA... (strong violet straight line). The period 2 leads to the pattern ARARA... (lime green large circle). The period 4 highlighted in gray leads to the serpentine pattern RALARALA... (lime green and strong violet lines).

Algorithm 1 CROMM mobility model

1: **procedure** CROMM
2:     *current state* ← "ahead"
3: *loop*:
4:     ρ ← *next value in the first return map Fig. 2*
5:     **if** $\rho < \frac{1}{3}$ **then** *current state* ← "right"
6:     **else if** $\rho < \frac{2}{3}$ **then** *current state* ← "left"
7:     **else** *current state* ← "ahead"
8:     move according to the *current state*

This basic mobility model is named CROMM and its pseudo-code is detailed in Alg. 1. In that case, the good exploration performance of the UAVs is due to the periodic orbits of the system that lead to patterns (Fig. 2). The periodic orbits with low periods are considered as skeleton of the chaotic dynamics and are often visited during the simulation process using a

Table 1 Pheromone action table for ACO UAV mobility model. *left* is the amount of pheromones sensed on the left of the UAV, *ahead* is the amount of pheromones sensed in front of the UAV and *right* is the amount of pheromones sensed on the right of the UAV; $total = left + ahead + right$.

Probability of action with pheromones		
Left	Ahead	Right
$p_L = \frac{total-left}{2\times total}$	$p_A = \frac{total-ahead}{2\times total}$	$p_R = \frac{total-right}{2\times total}$

Runge-Kutta (4$^{\text{th}}$ order) algorithm. As illustrated Fig. 2, the period 1 orbits leads to symbols AAAAA... (straight line); the period 2 leads to the pattern ARARA... (large circle) and the period 4 leads to the serpentine pattern RALARALA...

For CACOC, we used an ant colony algorithm combined with chaotic dynamics. If there is no virtual pheromone to guide the UAV (pheromones are deposited by each UAV to indicate areas they already visited), CROMM is used. If there are pheromones, they have repulsive properties and the next choice of direction depends on the total amount of pheromones sensed around the UAV. Such a repulsive effect prevents the UAVs to reach recently visited areas since pheromones evaporate in time. We also used the first return map values (Fig. 2) to choose the next direction with the pheromones' perception instead of a random number. Consider that p_L, p_A and p_R are inversely proportional to the total amount of pheromones sensed respectively to the left, ahead and right of the UAV and that $p_R + p_L + p_A = 1$ (Tab. 1). Thus, with ρ_n taken from the first return map (Fig. 2), the next direction is chosen according to these rules:

- if $\rho_n < p_R$ next direction is right;
- if $p_R \leqslant \rho_n < p_R + p_L$ next direction is left;
- else the direction is ahead.

The pseudo-code of CACOC is detailed in Alg. 2.

Algorithm 2 CACOC mobility model

1: **procedure** CACOC
2:     *current state* ← "ahead"
3: *loop*:
4:     ρ ← *next value in the first return map Fig. 2*
5:     **if** no pheromone sensed in the neighborhood **then**
6:         *current state* ← $CROMM(\rho)$ # see Alg. 1
7:     **else**
8:         **if** $\rho < p_R$ **then** *current state* ← "right" # see Tab. 1 for p_R
9:         **else if** $\rho < p_R + p_L$ **then** *current state* ← "left" # see Tab. 1 for p_R and p_L
10:         **else** *current state* ← "ahead"
11:     move according to the *current state*

2.2 CACOC and Collision Avoidance (CACOC+CA)

One of the drawbacks of the previously introduced CACOC mobility model is that it does not prevent UAVs from colliding. This can be avoided by flying the UAVs at different altitudes, which however might not be possible due to limited airspace and also leads to differing sensor perception areas and ground resolutions. For these reasons, it is preferable that UAVs are able to fly at the same altitude which requires CA mechanisms. Therefore, this work aims to include CA in the CACOC mobility model (CACOC+CA).

2.2.1 State of the Art

In the literature, several surveys detail the main methods used to implement CA for UAVs [1, 2, 18]. One of this article provides a classification of the CA system (see Fig. 3 [1]). In [13], authors used evolutionary algorithms to find an optimal solution for path planing including CA constraints. However, this family of methods is not suitable for our problem because we are aiming to obtain unpredictable trajectories (without repetition of the same path). The same issue comes with methods proposing CA for static objects using a collision cone philosophy to predict the collision (see [31] for instance). Most of the literature uses a safety cone to detect the potential collision and initiate evasive maneuvers (see [17] for instance). In 2012, Lalish & Morgansen [16] introduced the Distributed Reactive Collision Avoidance (DRCA) algorithm which is a two-step process. It consists of an optimization-based deconfliction maneuver, followed by the longer-term deconfliction maintenance phase. Its reactive and force-field type approach has been recently extended for swarms of quadrotor helicopters [17]. Authors of [14] use the proportional navigation (PN) guidance law to build a CA system. This guidance law was originally designed for missile engagement scenarios and require persistent trajectories.

All of these algorithms handle various situations including heterogeneity and priority between UAVs in order to permit the UAVs to reach their destination close to the nominal trajectory. However, the CACOC mobility model is typically using low waypoint and direction update intervals. Accordingly, the chosen direction of a UAV is not persistent wherefore the cone evasion method or proportional navigation cannot be used efficiently. Furthermore, the goal of the cooperating UAVs is maximum area coverage which is not considered in the previously mentioned techniques. For example for several UAVs in close proximity, the goal is to spread them out by using a repelling mechanism. To the best of our knowledge, no current algorithm is available to suffice chaotic fast direction changes combined with the coverage problem capable of avoiding collisions of UAVs.

2.2.2 Collision Avoidance for CACOC

As previously mentioned, the CACOC mobility model is based on CROMM (Alg. 1). The latter one uses the properties of periodic orbits to obtain exploratory patterns: straight line, right turn and serpentine movements (Fig. 2). In order to avoid collision with CACOC, a collision evasion maneuver is chosen complementarily to these patterns: a large turn to the left. The evasion maneuver is initiated when one or more UAVs appear in a UAV neighborhood. This left turn is combined with the average direction of the neighbors (UAVs in the neighborhood) and lasts until the other UAVs leave this neighborhood. We use the velocities:

- previous direction ($\mathbf{p}$)
- velocity to the left ($\mathbf{l}$)
- average velocity of the neighbors ($0.8 \times \mathbf{n}$) computed from $\mathbf{p}$ of the other UAVs
⇒ net velocity ($\mathbf{d} = \mathbf{l} + 0.8 \times \mathbf{n}$)

As illustrated by Fig. 3, the net velocity gives the direction of the UAVs. Then this direction vector is normalized to obtain the CA vector with the same norm as CACOC vectors:

$$\mathbf{d}_s = s \frac{\mathbf{l} + 0.8 \times \mathbf{n}}{||\mathbf{l} + 0.8 \times \mathbf{n}||} \tag{1}$$

with s the UAVs speed as a parameter of our model. For the sake of clarity, the mobility model is referred to as CACOC+CA in the remainder of this article. The algorithm of CACOC+CA is given by Alg. 3.
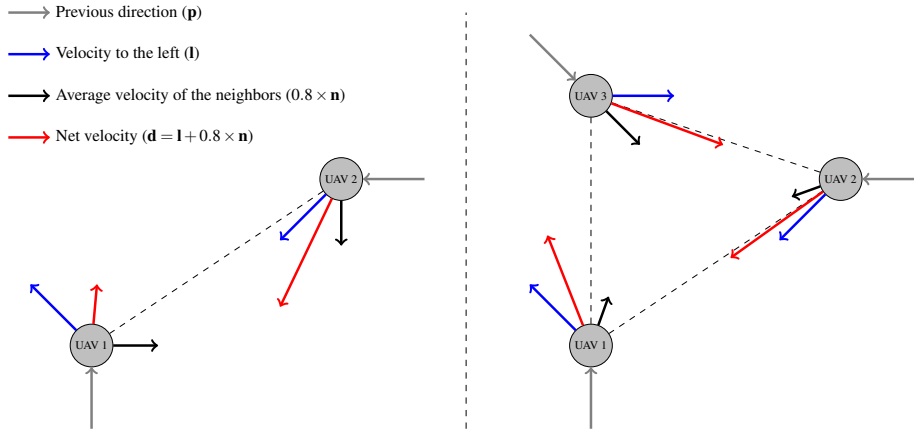
Fig. 3 Velocities imposed to prevent collision: blue arrow represent the velocity to the left (**l**), the black arrow represents the average direction resulting from the other UAVs (**n**) and the red arrow is the direction given (net velocity $\mathbf{d} = \mathbf{l} + 0.8 \times \mathbf{n}$). This schema only represents the direction of the UAV to illustrate how CA is handled. The norm is not proportional to the movement executed by the UAVs (see Fig. 4 for the trajectories of the UAVs) and Eq.(1) for the normalization.

Algorithm 3 CACOC+CA: CACOC with Collision Avoidance mobility model

---

1: **procedure** CACOC
2:     *current state* ← "ahead"
3: *loop*:
4:     **if** there is UAV in the neighbourhood **then**
5:         *current state* ← $\mathbf{d}_s$ # See Fig. 3 and Eq. (1)
6:     **else**
7:         ρ ← *next value in the first return map Fig. 2*
8:         *current state* ← $CACOC(\rho)$ # see Alg. 2
9:     move according to the *current state*

Fig. 4 shows two consecutive maneuvers of CA during a CACOC+CA simulation. According to the classification of the CA system (see Fig. 3 [1]), Fig. 4 is showing the distinctive evasive left turns. As a remark, large turns to the left can occur in CACOC (see Fig. 2). Hence, the large turn to the left permits to include a mechanism of CA without introducing a new complex pattern. In order to evaluate the proposed CACOC+CA approach statistically, its performance is evaluated in long-time simulations. As the goal is to deploy the UAVs at the same altitude, the CACOC+CA mobility model is expected to be slightly less efficient than a CACOC mobility model, due to the inhibiting character of CA. In the long-time simulations the computed CACOC waypoints are serving as target points for MPC controlled UAVs with CA inclusion which is subject of section 3. The consequence is a realistic evaluation scenario which allows to evaluate the performance of CACOC considering realistic UAV dynamics, controllers and CA.

3 UAV control

In order to describe the dynamic behavior of a real quadrotor system, previous work [7] presented an identified state space model of an AR.Drone 2.0 quadrotor. This state space description has been used as prediction model within a model predictive controller which has
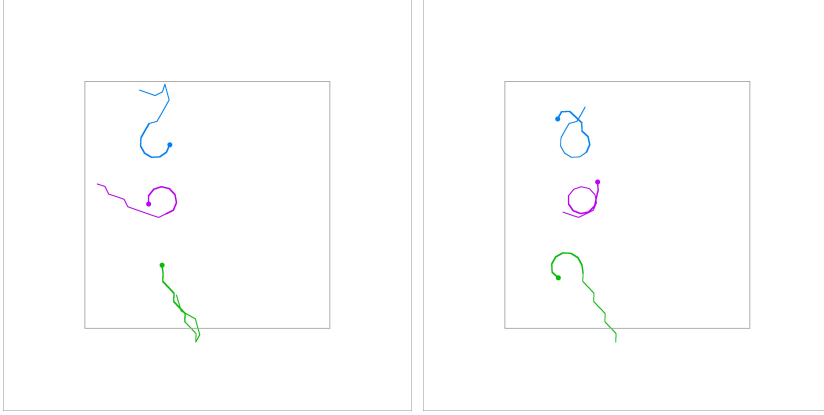
Fig. 4 Trajectories of the three UAVs doing maneuver to avoid collision.

been applied to a real AR.Drone 2.0 system. The resulting control performance validated the quality of the identified model. For this reason, the same model is utilized for the evaluation of the presented CACOC+CA approach. The utilized UAV model is illustrated in Fig.5.
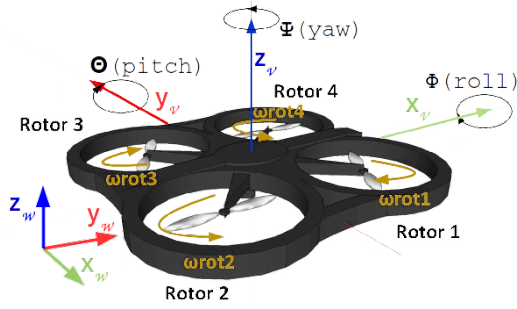

Fig. 5 AR.Drone 2.0 Quadrotor with utilized coordinate conventions.

The state of the i-th UAV can be described by vector

$$\mathbf{x}_i = \left[x_i, y_i, z_i, \Psi_i, v_{f,i}, v_{s,i} \right]. \tag{2}$$

For means of visualization, the time dependency of states and controls (e.g., $\mathbf{x}_i(t)$) is not explicitly shown here. The system state (2) consists of the quadrotor's x_i, y_i, z_i position in Cartesian world coordinates ($\mathcal{W}$), its yaw angle Ψ_i and the forward $v_{f,i}$ and side ward velocities $v_{s,i}$ in the vehicle frame ($\mathcal{V}$). The UAV inputs are concatenated in its control vector

$$\mathbf{u}_i = \left[u_{f,i}, u_{s,i}, u_{z,i}, u_{\Psi,i} \right], \tag{3}$$

containing forward $u_{f,i}$, side ward $u_{s,i}$, upward $u_{z,i}$ and yaw $u_{\Psi,i}$ velocity. The identified system model of [7] is given as

$$\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i, \mathbf{u}_i) = \begin{bmatrix} v_{f,i}\cos(\Psi_i) - v_{s,i}\sin(\Psi_i) \\ v_{f,i}\sin(\Psi_i) + v_{s,i}\cos(\Psi_i) \\ u_{z,i} \\ 1.6u_{\Psi,i} \\ -0.5092v_{f,i} + 1.458u_{f,i} \\ -0.5092v_{s,i} + 1.458u_{s,i} \end{bmatrix}. \tag{4}$$

This model is based on a second order model for the x and y-channel and single integrator systems for z and Ψ. This assumption of single integrator systems in z and Ψ implies that the target velocity in this channel is reached instantaneously. This is a suboptimal estimation, but does not influence the evaluation of CACOC as the UAVs are flying on the same altitude and do not change their orientation. As further remark, the AR.Drone 2.0 quadrotor behavior shows nonlinearities in the z-axis, due to unexpected behavior in the inherent sensing solution combining cameras and ultrasonic-sensors. However, except the nonlinear mapping of the forward and sideward velocities to $\dot{x}$ and $\dot{y}$ no further nonlinearities are considered by the model. Despite, the model is sufficiently precise to serve as prediction model for a real AR.Drone 2.0. The scalar parameters in (4) have been identified from a real AR.Drone 2.0 system by means of a motion capture system.

As commercial drones are typically operated by velocity commands, a position controller is required to reach a desired position. [7] is using a fast MPC for this purpose which is based on the computation of optimal controls by solving an optimal control problem (OCP). In [27], this approach has been applied to ten UAVs using OCP

$$\min_{\mathbf{u}_i} J_i = \int_{t_0}^{t_f} (\mathbf{x}_i^* - \mathbf{x}_i)^\top \mathbf{Q}(\mathbf{x}_i^* - \mathbf{x}_i) + \mathbf{u}_i^\top \mathbf{R}\mathbf{u}_i \ d\tau \tag{5}$$

s.t.

$$\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}_i, \mathbf{u}_i) \tag{6}$$

$$\mathbf{c} \le \left[u_f(t)^2 - 1 \ u_s(t)^2 - 1 \ u_z(t)^2 - 1 \ u_\Psi^2 - 1 \right]^\top \tag{7}$$

$$\mathbf{x}_i(0) = \left[50, 0, i, 0, 0, 0 \right] \tag{8}$$

with

$$\mathbf{Q} = \mathscr{D}\{ \left[1.5, 1.5, 1.6, 0.1, 0, 0 \right] \}, \tag{9}$$

$$\mathbf{R} = \mathscr{D}\{ \left[1, 1, 1, 1 \right] \}. \tag{10}$$

The idea of MPC is to predict the UAVs behavior within a horizon $\tau = [t_0, t_f]$ by means of the system dynamics (6) which are equal to the model (4). Based on this prediction, the control inputs $\mathbf{u}_i$ are determined in order to minimize cost functional J_i while satisfying constraints (7) and initial conditions (8). The position tracking in J_i is achieved by a quadratic penalty (9) of the state error, while energy optimality is imposed by a quadratic penalty (10) of the controls.

The advantage of MPC is its ability to directly consider constraints in the OCP (e.g The AR.Drone 2.0 input limitations $-1 \le u \le 1$ in (7)) and a scenario independent control policy. In [7] this advantage has been used to avoid the collision of a UAV with a static obstacle by imposing the constraint as additional cost term

$$l_{CA}(\mathbf{x}_i, \mathbf{x}_j, d_{min}) = \frac{a}{1 + e^{-b\left(d_{min}^2 - (\mathbf{x}_i - \mathbf{x}_j)^\top(\mathbf{x}_i - \mathbf{x}_j)\right)}} \tag{11}$$

in J_i. The idea is to impose costs of height a, if the Euclidean distance between the drone positions $\mathbf{x}_i$ and $\mathbf{x}_j$ is smaller than distance d_{min}. For larger drone distances the costs vanish to 0. This switching behavior is approximated by a sigmoid with the steepness parameter b. The result is a repulsive cost-gradient around the violation distance. The disadvantage of this constraint handling is its cost trade-off with the trajectory tracking and energy optimality. Unfortunately, this cost gradient cannot be chosen arbitrarily steep, as it worsens the OCP condition. Therefore, the cost trade-off is also a trade-off between repulsion from the violation and solution feasibility, which does provide CA only for limited tracking errors and system velocities. Within this work the CA distance is set chosen to $d_{min} = 1.5$m.

As l_{CA} is considering the positions of two UAVs, the cost functions cannot be treated separately for each UAV as in (5)-(10). As shown in previous work [5] for distributed systems and in [6] for UAVs, the CA constraints (11) are coupling the individual OCPs. Combining UAV dynamics, MPC for coupled multi agent systems and CA cost functions for three UAVs ($i = 0, 1, 2$) results in OCP

$$\min_{\mathbf{u}_0,\mathbf{u}_1,\mathbf{u}_2} J = \int_{t_0}^{t_f} \sum_{i=0}^{2} \left(\mathbf{x}_i^* - \mathbf{x}_i\right)^\top \mathbf{Q}\left(\mathbf{x}_i^* - \mathbf{x}_i\right) + \mathbf{u}_i^\top \mathbf{R}\mathbf{u}_i$$
$$+ l_{CA}\left(\mathbf{x}_0, \mathbf{x}_1, 1.5\right) + l_{CA}\left(\mathbf{x}_0, \mathbf{x}_2, 1.5\right) + l_{CA}\left(\mathbf{x}_1, \mathbf{x}_2, 1.5\right) \ \mathrm{d}\tau \qquad (12)$$
$$\text{with } l_{CA}\left(\mathbf{x}_i, \mathbf{x}_j, d_{min}\right) = \frac{1.0}{1 + e^{-3.0\left(d_{min}^2 - \left(\mathbf{x}_i - \mathbf{x}_j\right)^\top \left(\mathbf{x}_i - \mathbf{x}_j\right)\right)}}. \qquad (13)$$

s.t.

$$\dot{\mathbf{x}}_i = \mathbf{f}_i\left(\mathbf{x}_i, \mathbf{u}_i\right), \ i = 0, 1, 2 \qquad (14)$$

$$\mathbf{c}_i \leq \left[u_f, i(t)^2 - 1 \ u_s, i(t)^2 - 1 \ u_z, i(t)^2 - 1 \ u_{\Psi,i}^2 - 1\right]^\top, \ i = 0, 1, 2 \qquad (15)$$

$$\mathbf{x}_0\left(0\right) = \left[0, 0, 1, 0, 0, 0\right] \qquad (16)$$

$$\mathbf{x}_1\left(0\right) = \left[0, 15, 1, 0, 0, 0\right] \qquad (17)$$

$$\mathbf{x}_2\left(0\right) = \left[15, 0, 1, 0, 0, 0\right] \qquad (18)$$

$$\mathbf{Q} = \mathscr{D}\{\left[1.5, 1.5, 1.6, 0.1, 0, 0\right]\}, \qquad (19)$$

$$\mathbf{R} = \mathscr{D}\{\left[1, 1, 1, 1\right]\}. \qquad (20)$$

For means of visibility of the area coverage, the desired trajectory waypoints in the given scenario are limited to changes in the xy-plane, while the initial $z_i = 1$ and $\psi_i = 0$ are tracked. The resulting closed loop behavior is referred to as MPC+CA within this work.

Analog to previous work [7], a Condensed Multiple Shooting Continuation Generalized Minimal Residual (CMSCGMRES) [30] approach is used to solve the MPC with OCP (12)-(20). The CMSCGMRES is parameterized with a maximum number of iterations $k_{max} = 6$, horizon length $T_{MPC} = 1$s, control update interval of $\Delta t_{MPC} = 0.1$s, $n_{hor} = 10$ steps within the horizon, a forward difference step of $h = 0.001$s, solution tolerance of $\varepsilon = 10^{-4}$, continuation factor $\zeta = 10$ and the horizon expansion factor $\alpha = 2$. More details about the parameters are given in [21].

The resulting position tracking and CA is shown in Fig. 6. MPC+CA with inherent UAV model serves as realistic UAV simulator for the CACOC+CA evaluation which is subject of the next section 4.
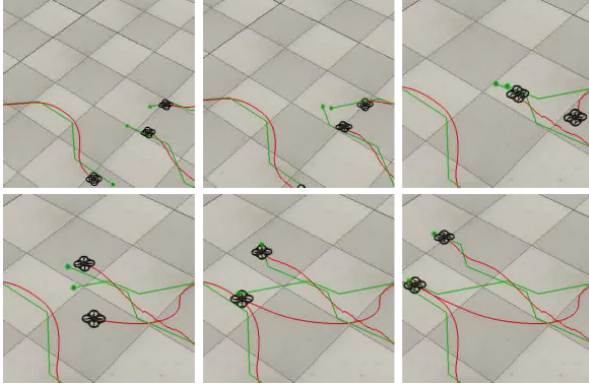
Fig. 6 MPC+CA: MPC position tracking for UAVs (AR.Drone 2.0) with CA. Target positions marked green, UAV positions red. The sequence is showing two intersecting UAV targets and the resulting evasion maneuver imposed by the MPC+CA.

4 Experiments

This section introduces the simulation framework and experimental setup, including all parameters. Subsequently, all metrics for the analysis of the experiment are detailed. Finally, the statistical results of the analysis are given. For the experimental evaluation of CACOC, respectively CACOC+CA, MPC+CA and three AR.Drone 2.0 UAVs V-REP [24] models are implemented according to Fig. 7 using ROS [23] as communication interface.
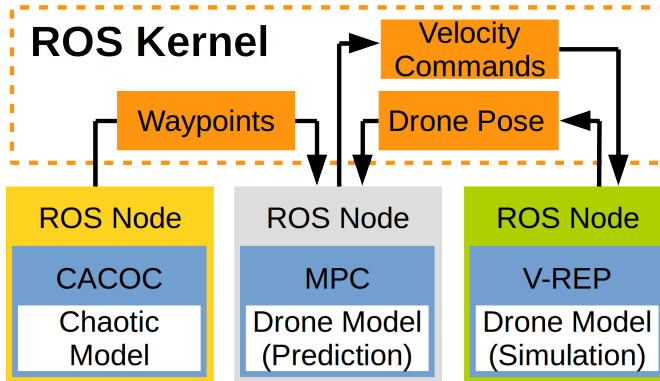

Fig. 7 Implementation setup with ROS as communication interface between CACOC, MPC and V-REP

The CACOC waypoints are updated every second $\Delta t_{CACOC} = 1$s, wherefore one CACOC step equals to $\Delta t_{sim} = 1$s of simulation. The UAV positions in V-REP are updated every $\Delta t_{VREP} = 0.1$s using the dynamics (4). As the controller is synchronized with V-REP, the control update interval is equally $\Delta t_{MPC} = 0.1$s. The simulation environment is illustrated in Fig. 8[1]. The simulation is performed using the parameters shown in Tab. 2.

---

[1] A video illustrating the framework simulator is available at `https://martinrosalie.gforge.uni.lu/_downloads/jint.mp4`
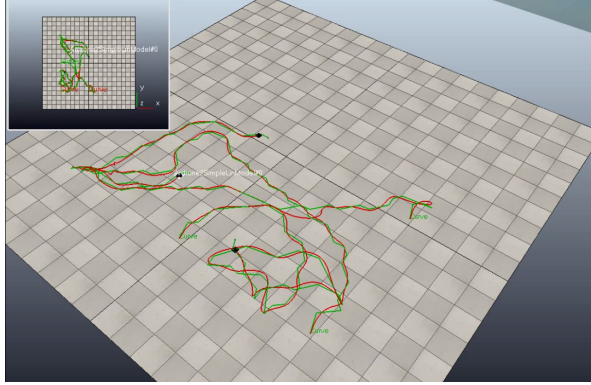
Fig. 8 Three MPC controlled UAVs in the V-Rep simulation environment following waypoints provided by CACOC.

Table 2 Main experimental parameters.

Parameter Name	Parameter Value
Simulation area	
Geographical Area	30 m × 30 m
Number of cells	30 × 30
Simulation Setup	
Update interval of CACOC waypoints	$\Delta t_{CACOC} = 1$s
Update interval of UAV dynamics in V-Rep	$\Delta t_{VREP} = 0.1$s
Control update interval	$\Delta t_{MPC} = 0.1$s
Time duration	$T_{sim} = 4000$s for 4000 steps
UAV Waypoint	
UAVs speed	1 m/s
Possible UAV actions	ahead, 45° left, 45° right
Initial UAVs position	middle of the bottom of the map
Experiments	
Mobility models	CACOC with MPC+CA, CACOC+CA with MPC+CA
Number of UAVs	3
Simulation steps	4000
Independent runs	30

4.1 Metrics

For the evaluation of the experiment the same three area coverage metrics as introduced in [27] are used. A short description is provided hereinafter. Fig. 9 underlines how the metrics are calculated for one simulation. The three UAVs start in the bottom left corner with the following coordinate on a grid: (0, 0), (0, 15) and (15, 0).

4.1.1 Coverage

The coverage metric is the percentage of cells of the total area visited during the whole simulation. The environment is a 30 m by 30 m square area. The coverage value varies during the whole simulation. To have a representative value of the coverage, we choose to compare the coverage value after 4000 steps for each model. This indicates the efficiency of the models to visit the total area. On the other hand, we want to evaluate the first steps of each model to compare their initial behavior. For this purpose the first 500 steps are used to extracted the slope of a linear regression $a \times x$ (Fig. 9).
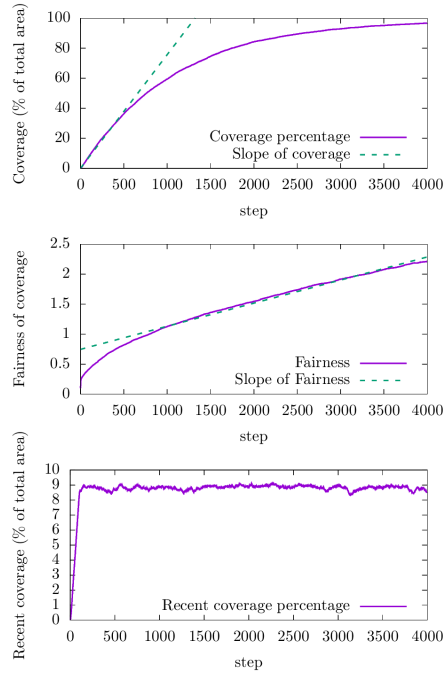
Fig. 9 Metrics measurements for one simulation. Two values are extracted for the coverage: the coverage after 4000 steps and the slope of coverage for the 500 first steps. For the fairness the slope of fairness is computed after the 500 first steps. The average value of the recent coverage is calculated after the 100 first steps.

4.1.2 Fairness of the coverage

The fairness measures, if all cells are regularly and equally scanned. This is computed as the standard deviation (STD) of their respective number of scans [29]. To evaluate the fairness during the whole simulation, a linear regression $a \times x + b$ is performed using the last 3500 steps (Fig. 9). This measure is complementary to the coverage initial slope that only evaluates the initial UAV trajectories. Here, the slope value is considered as a measure to evaluate the fairness of the models without this initialization part.

4.1.3 Recent coverage ratio

This metric introduced in [4] represents the percentage of coverage during the last 100 iterations. These 100 steps correspond to the pheromones' evaporation time. For the computation of the recent coverage mean value (Fig. 9), the 100 first iterations are excluded from the simulation.

4.1.4 Distance between each couple of UAVs

The metrics used to quantify the efficiency of the CA mechanism are dedicated to run where only one collision is scheduled on purpose [17]. However, here the goal is to cover a wide area where multiple collisions can occur. As a consequence, new metrics are necessary to
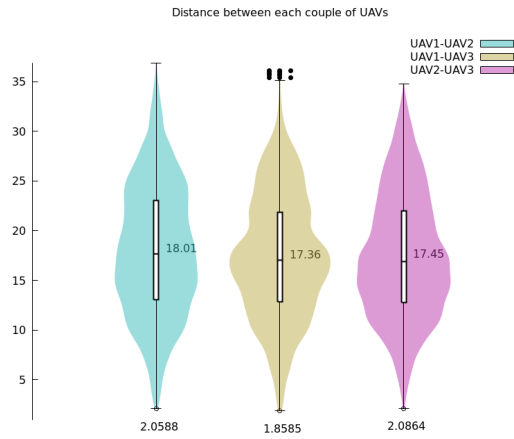
Fig. 10 For a given simulation, using CACOC+CA with MPC+CA, the distance between each couple of UAV is shown in meters. An estimation of the distribution is given with the mean value as well as the minimum.
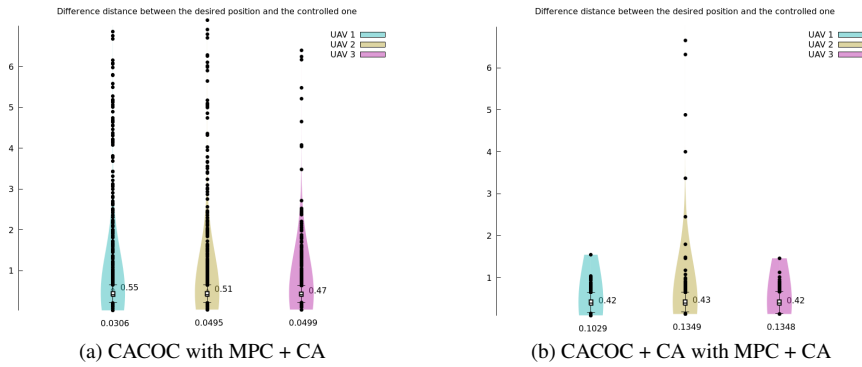

(a) CACOC with MPC + CA

(b) CACOC + CA with MPC + CA

Fig. 11 For a given simulation, using CACOC+CA with MPC+CA, the distance between the desired position and the controlled one is shown in meters. An estimation of the distribution is given with the mean value as well as the minimum.

quantify the CA mechanism efficiency on a long-time simulation. An intuitive way of evaluating the CA mechanism is to measure the distance between each couple of UAVs during the whole simulation. This includes the mean value and the minimal value of this metric to make a comparison between CACOC+CA with MPC+CA and CACOC with MPC+CA. For a given simulation, Fig. 10 represents the estimated distribution of the measured values.

4.1.5 Distance between the desired and the current position

In order to quantify few CA occurrences in a simulation, the authors of [17] use the excess separation deviation from the path and the graph status (indicating if the UAV is doing CA). We introduce a new metric by computing the distance between the waypoint of a UAV and the position given by the MPC before receiving the new waypoint. During the whole simulation it indicates how far the UAV is from the current waypoint. We collect the minimum value and the mean value of this metric to quantify the CA.

For the CACOC+CA with MPC+CA example simulation, Fig. 11(b) shows that only few of the UAV2 values are far from the median value. This indicates that despite the CA included in CACOC, the CA of the MPC is used. For the CACOC with MPC+CA example simulation in Fig. 11(a), only the CA of the MPC is used resulting in several points far from the median value for all UAVs. We collect both mean value and minimum values to have an average value of the distance combined with the shorter distance reached during the simulation. As a remark, the static offset in the tracking error is expected, due to the constantly changing target position and missing integral part in the MPC control policy.

4.2 Experimental results

The previously described metrics are obtained with the same program to avoid bias. First, they are computed from the waypoints and secondly using the synchronized positions of the UAVs from the V-Rep simulation using ROS. As CACOC is a chaotic deterministic mobility model, only the initial conditions of the dynamical system are varied to create a wide range of simulations with the same mobility model without changing its parametrization as it has already been done previously [26].

4.2.1 Comparison of the coverage metrics

Table 3 details the coverage metrics statistics to compare CACOC with MPC+CA and CACOC+CA with MPC+CA. Data for *coverage after 4000 steps* follow the same distribution with 95% confidence according to a Wilcoxon test (p–value $= 0.1357$). This permits to confirm that for both, CACOC with MPC+CA and CACOC+CA with MPC+CA, we fulfill our objective by covering almost the whole area: for both the median value is 1 meaning that 100% of the area has been covered in more than half of the simulations.

Table 3 Coverage metrics statistics. For each metric, the first line details CACOC with MPC+CA and the second line details CACOC+CA with MPC+CA and the best mean values are highlighted.

Metrics	CA	mean	min	median	max	STD
Coverage after 4000 steps		0.99967	0.99778	1	1	0.00066
	✓	0.99944	0.99778	1	1	0.00070
Slope of coverage		0.00188	0.00179	0.00187	0.00196	3.8377e-05
	✓	0.00183	0.00173	0.00183	0.00195	4.2101e-05
Slope of the Fairness of the coverage		8.0783e-4	7.2143e-4	8.0077e-4	9.1150e-4	4.7068e-05
	✓	8.3092e-4	7.3900e-4	8.2478e-4	9.2471e-4	4.4969e-05
Recent coverage		0.26730	0.26606	0.26727	0.26894	9.0553e-4
	✓	0.26051	0.25781	0.26046	0.26315	1.3482e-3

Fig. 12 shows that the slope of coverage values of CACOC+CA with MPC+CA values are lower than the values of CACOC with MPC+CA. Data for *slope of coverage* follow a different distribution with 95% confidence according to a Wilcoxon test (p–value $= 0.0002189$). This small difference is due to the introduction of the CA mechanism that prevents a very good spreading of the UAVs at the beginning of the simulation compared to the CACOC mobility model. However, this metric mainly quantifies the early steps of the algorithm. Hence, the performance reduction is not significant as the starting points of the UAVs are not in close proximity.
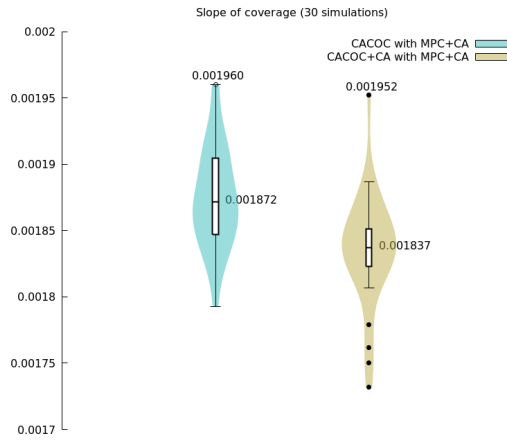
Fig. 12 Slope of the coverage (see Fig.9 for details about the linear regression).
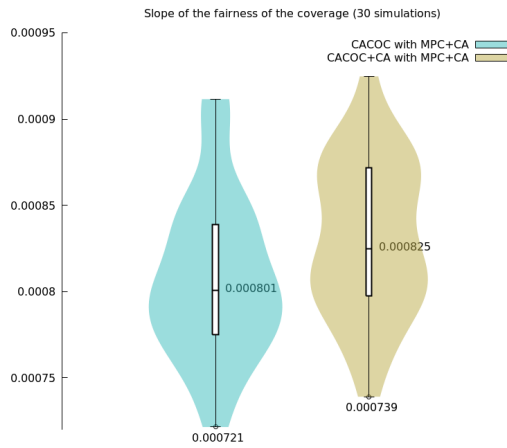

Fig. 13 Fairness slope of the coverage (see Fig.9 for details about the linear regression).

We remind that the lower the fairness, the better the performance. Data for *fairness* follow different distributions with 95% confidence according to a Wilcoxon test (p–value = 0.04146). The p–value is close to the threshold value to accept the hypothesis, meaning that the results are similar. Fig. 13 details how CACOC+CA with MPC+CA is slightly less efficient than CACOC with MPC+CA. Although the two resulting distributions are not the same, the difference between the mean values is only 3%. As this metric evaluates the global performance of the algorithm, we can conclude that the usage of a CA mechanism does not significantly impact the coverage of the area.

Data for *recent coverage* follow different distribution with 95% confidence according to a Wilcoxon test (p–value $< 2.2e - 16$) and Fig. 14 details the recent coverage values. In our case, for 3 UAVs over a 30×30 area with pheromones' duration of 100 steps, the optimal value is $3 \times 100/(30 \times 30) = \frac{1}{3}$. The mean value of the recent coverage of CACOC+CA with MPC+CA values is 0.22% of the optimal value and accordingly lower than the average value of CACOC with MPC+CA. This marginal difference implies that even with a CA mechanism, the mobility model is efficient.
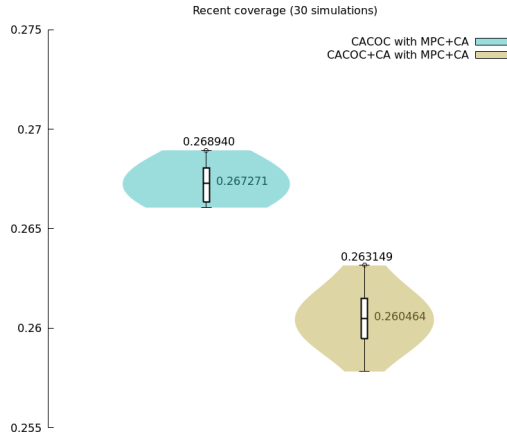
Fig. 14 Recent coverage (see Fig.9 for details about the average value).

Considering the overall coverage performance of the CACOC+CA with MPC+CA compared to CACOC with MPC+CA, we can say that even if the values are not following the same distribution, they are close to the value without CA. This small offset is reasonable comparing to the added security by CA. In the next section, we will compare the CA metrics to see the difference between the two methods.

4.2.2 Comparison of the collision avoidance metrics

The Figs. 15–18 illustrate the value distribution for each metric dedicated to the CA. We thus perform a pairwise non-parametric test (using the Wilcoxon signed-rank test): where we observe that the data follows different distributions with 95% confidence. Tab. 4 synthesizes the statistics for these CA metrics.

Table 4 CA metrics statistics. For each metric, the first line details CACOC with MPC+CA and the second line details CACOC+CA with MPC+CA and the best mean values are highlighted.

Metrics	CA	mean	min	median	max	STD
Average distance between UAV and waypoint [m]		0.50657	0.45554	0.49252	0.71484	0.04770
	✓	0.50572	0.41288	0.47357	0.90022	0.09822
Minimal distance between UAV and waypoint [m]		0.05954	0.00334	0.05376	0.15001	0.02980
	✓	0.10562	0.01600	0.11084	0.18513	0.04506
Average distance between couple of UAVs [m]		15.327	14.112	15.339	16.395	0.49876
	✓	17.192	10.636	17.462	18.567	1.2416
Minimal distance between couple of UAVs [m]		0.82445	0.26817	0.79422	1.3867	0.24397
	✓	1.5546	0.06670	1.6684	2.8743	0.67708

For the average distance between the actual position of a UAV and its waypoint (Fig. 15), the average values are similar, but the standard deviation (STD) is almost doubled for the CACOC+CA with MPC+CA. This implies that CACOC+CA with MPC+CA with the lower mean performs slightly better, but the variation of its tracking performance is higher. This variation is explainable with a border lock phenomenon occurring when UAVs come close to border and CA constraints. As the UAVs should stay in the the given search area they are
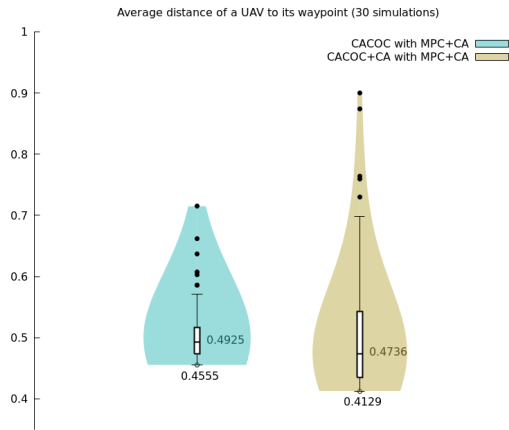
Fig. 15 Average distance in meters between the UAV and its waypoint.

forced back into it. As a result, the UAV trajectories start to oscillate and stay consequently close together for a while. We remind that the CA of the MPC is rarely used when there is already a CA in the mobility model (see Fig. 11a where a significant amount of outliers is visible).
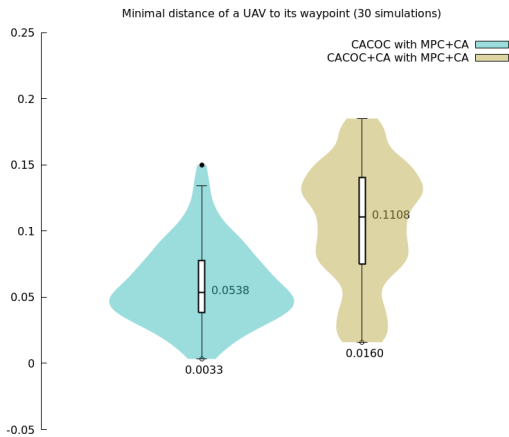

Fig. 16 Minimum distance in meters between the UAV and its waypoint.

Considering the minimum distance between the UAV and its waypoint (Fig. 16) the distributions are not the same but the range of values is similar. Taking into account the mean values (0.06 for CACOC with MPC+CA and 0.11 for CACOC+CA with MPC+CA in Tab. 4) we can consider that in average, the second method does not permit two UAVs to get very close to waypoints. We explain this shift because of the CA mechanism in CACOC with MPC + CA that can cause the UAVs to avoid each other and reach positions far or close to the given waypoints. This happens less frequently in CACOC+CA with MPC+CA because the second CA mechanism (MPC+CA) is typically not used.
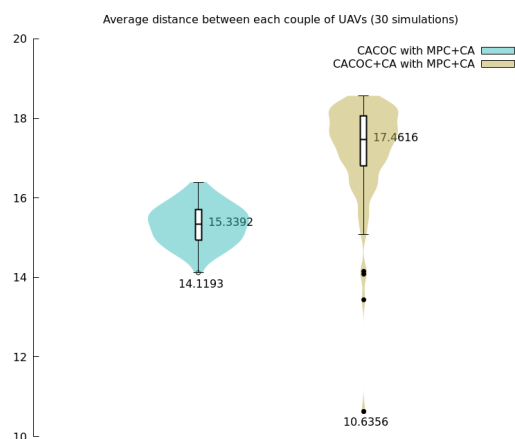
Fig. 17 Average distance in meters between each couple of UAVs.

The distribution of the average distance between each couple of UAVs (Fig. 17) is totally different and shows outliers. These outliers are also a result of the border lock phenomenon, inducing oscillations around the area border constraints. They can furthermore be found in the minimum value plot of CACOC+CA with MPC+CA in Fig. 17. However, without this specific behavior, the average distance is higher by 15%. This implies that the UAVs are more often far from each other than width CACOC with MPC+CA.
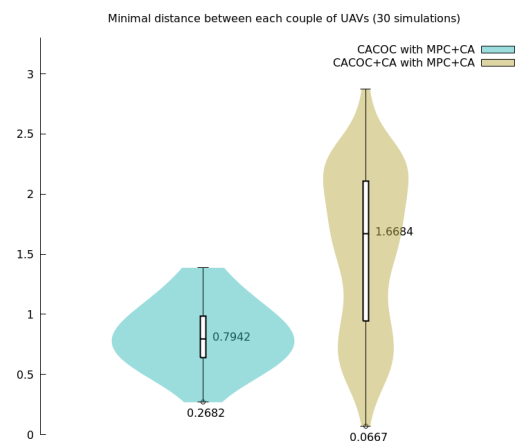

Fig. 18 Minimal distance in meters between each couple of UAVs.

Finally, considering the minimum distance between each couple of UAVs (Fig. 18), the average value of CACOC+CA with MPC is twice as high as for CACOC with MPC+CA for CACOC+CA and reaches a mean value of more than 1.5m. Thus, CACOC+CA complements the CA implemented in MPC to reach the desired minimum distance of $d_{min} = 1.5$m. The higher variance can be explained by the border lock phenomenon and is subject to further research.

5 Conclusion

To support the real deployment of UAVs for coverage purposes, we are aiming to design robust and resilient systems. The paper is extending the Chaotic Ant Colony Optimization to Coverage mobility model by a CA mechanism (CACOC+CA). Its coverage and CA performance is compared to the mobility model without CA (CACOC) in physical simulations within a dedicated robotic environment. The physical simulation includes a dynamic UAV model controlled with MPC including a CA mechanism (MPC+CA). The empirical evaluation of both "CACOC with MPC+CA" and "CACOC+CA with MPC+CA" was conducted on an area coverage scenario with 3 UAVs. In this context, three area coverage metrics and four CA metrics have been used to evaluate and compare the algorithm's performance. The obtained results show that CACOC+CA is able to accomplish covering tasks efficiently while complementing the potential field CA mechanism of the MPC controller, reducing the average tracking error of UAVs and increasing the mean minimal distance between UAVs close to the chosen MPC CA distance $d = 1.5$m. The variance of the minimal distance is assumed to be a result of the border-lock phenomenon and subject to further research. In terms of coverage, the high similarity of the results between CACOC with MPC+CA and CACOC+CA with MPC+CA demonstrates that it performs well in a real environment under consideration of a fine-tuning of the MPC approach. Thus, the implementation of CA with CACOC is advantageous to deal with collisions of UAVs, while the CA of MPC can be used to deal with unknown obstacles (eg. birds).

In order to transfer the proposed approach to real UAVs in future, the CA bounds have to be adjusted according to the localization precision. For scenarios with more UAVs, taking into consideration a CA constraint for every UAV pair does not scale well. This issue can be addressed by using a distributed MPC controller which only considers CA constraints to direct neighbors. Further work on the UAV control part will treat the reduction of the position tracking error, for instance, with the introduction of a disturbance model, the adaptation of the MPC control law, or a target position controller as already presented in [7]. Furthermore, more research work will be dedicated to address the border lock phenomenon. The presented two-sided simulator will also permit to work on the CACOC parameters (pheromone evaporation time, covered area by step, etc.) to increase the performance of the mobility model by comparing the theoretical and realistic results. Finally, we also planed to develop a time-efficient software to handle the MPC+CA algorithm for a swarm of 10 and 20 UAVs.

References

1. Albaker, B.M., Rahim, N.A.: A survey of collision avoidance approaches for unmanned aerial vehicles. In: 2009 International Conference for Technical Postgraduates (TECHPOS). IEEE (2009). DOI 10.1109/techpos.2009.5412074
2. Albaker, B.M., Rahim, N.A.: A conceptual framework and a review of conflict sensing, detection, awareness and escape maneuvering methods for UAVs. In: Aeronautics and Astronautics. InTech (2011). DOI 10.5772/26567
3. Bertrand, S., Marzat, J., Piet-Lahanier, H., Kahn, A., Rochefort, Y.: MPC Strategies for Cooperative Guidance of Autonomous Vehicles. AerospaceLab (8), 1–18 (2014). DOI 10.12762/2014.AL08-11

4. Danoy, G., Brust, M.R., Bouvry, P.: Connectivity stability in autonomous multi-level UAV swarms for wide area monitoring. In: Proc. of ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications (DIVANet'15) (2015). DOI 10.1145/2815347.2815351
5. Dentler, J., Kannan, S., Mendez, M.A.O., Voos, H.: A modularization approach for nonlinear model predictive control of distributed fast systems. In: 2016 24th Mediterranean Conference on Control and Automation (MED), pp. 292–297 (2016). DOI 10.1109/MED.2016.7535973
6. Dentler, J., Kannan, S., Mendez, M.A.O., Voos, H.: Implementation and validation of an event-based real-time nonlinear model predictive control framework with ROS interface for single and multi-robot systems. In: 2017 IEEE 1st IEEE Conference on Control Technology and Applications (CCTA), pp. 1000–1006 (2017). DOI 10.1109/CCTA.2017.8062590
7. Dentler, J., Kannan, S., Olivares-Mendez, M.A., Voos, H.: A real-time model predictive position control with collision avoidance for commercial low-cost quadrotors. In: Proc. of IEEE Multi-Conference on Systems and Control (MSC) (2016). DOI 10.1109/CCA.2016.7587882
8. Dorigo, M.: Optimization, learning and natural algorithms. Ph.D. thesis, Politecnico di Milano, Italy (1992)
9. Gandomi, A.H., Yang, X.S., Talatahari, S., Alavi, A.H.: Firefly algorithm with chaos. Communications in Nonlinear Science and Numerical Simulation **18**(1), 89–98 (2013). DOI 10.1016/j.cnsns.2012.06.009
10. Gandomi, A.H., Yun, G.J., Yang, X.S., Talatahari, S.: Chaos-enhanced accelerated particle swarm optimization. Communications in Nonlinear Science and Numerical Simulation **18**(2), 327–340 (2013). DOI 10.1016/j.cnsns.2012.07.017
11. Gaudiano, P., Shargel, B., Bonabeau, E., Clough, B.T.: Swarm intelligence: a new C2 paradigm with an application to control of swarms of UAVs. In: Proc. of International Command and Control Research and Technology Symposium (ICCRTS) (2003)
12. Gazi, V., Passino, K.M.: Swarm Coordination and Control Problems, pp. 15–25. Springer Berlin Heidelberg, Berlin, Heidelberg (2011). DOI 10.1007/978-3-642-18041-5_2
13. Geng, L., Zhang, Y.F., Wang, J.J., Fuh, J.Y.H., Teo, S.H.: Mission planning of autonomous UAVs for urban surveillance with evolutionary algorithms. In: 2013 10th IEEE International Conference on Control and Automation (ICCA). IEEE (2013). DOI 10.1109/icca.2013.6564992
14. Han, S.C., Bang, H., Yoo, C.S.: Proportional navigation-based collision avoidance for UAVs. International Journal of Control, Automation and Systems **7**(4), 553–565 (2009). DOI 10.1007/s12555-009-0407-1
15. Kuiper, E., Nadjm-Tehrani, S.: Mobility models for UAV group reconnaissance applications. In: Proc. of IEEE International Conference on Wireless and Mobile Communications (ICWMC'06) (2006). DOI 10.1109/ICWMC.2006.63
16. Lalish, E., Morgansen, K.A.: Distributed reactive collision avoidance. Autonomous Robots **32**(3), 207–226 (2012). DOI 10.1007/s10514-011-9267-7
17. Leonard, J., Savvaris, A., Tsourdos, A.: Distributed reactive collision avoidance for a swarm of quadrotors. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering **231**(6), 1035–1055 (2016). DOI 10.1177/0954410016647074
18. Mahjri, I., Dhraief, A., Belghith, A.: A review on collision avoidance systems for unmanned aerial vehicles. In: Lecture Notes in Computer Science, pp. 203–214. Springer International Publishing (2015). DOI 10.1007/978-3-319-17765-6_18
19. McCune, R.R., Madey, G.R.: Swarm control of UAVs for cooperative hunting with DDDAS. Procedia Computer Science **18**, 2537–2544 (2013). DOI 10.1016/j.procs.2013.05.436
20. Nigam, N.: The multiple unmanned air vehicle persistent surveillance problem: A review. Machines **2**(1), 13–72 (2014). DOI 10.3390/machines2010013
21. Ohtsuka, T.: A continuation/gmres method for fast computation of nonlinear receding horizon control. Automatica **40**(4), 563–574 (2004). DOI 10.1016/j.automatica.2003.11.005
22. Pluhacek, M., Senkerik, R., Davendra, D.: Chaos particle swarm optimization with eensemble of chaotic systems. Swarm and Evolutionary Computation **25**, 29–35 (2015). DOI 10.1016/j.swevo.2015.10.008
23. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: ROS: an open-source robot operating system. In: Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics. Kobe, Japan (2009)
24. Rohmer, E., Singh, S.P.N., Freese, M.: V-rep: a versatile and scalable robot simulation framework. In: Proc. of IEEE International Conference on Intelligent Robots and Systems (IROS) (2013). DOI 10.1109/IROS.2013.6696520
25. Rosalie, M.: Templates and subtemplates of Rössler attractors from a bifurcation diagram. Journal of Physics A: Mathematical and Theoretical **49**(31), 315,101 (2016). DOI 10.1088/1751-8113/49/31/315101
26. Rosalie, M., Danoy, G., Chaumette, S., Bouvry, P.: From random process to chaotic behavior in swarms of UAVs. In: Proc. of ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications (DIVANet'16) (2016). DOI 10.1145/2989275.2989281

27. Rosalie, M., Dentler, J.E., Danoy, G., Bouvry, P., Kannan, S., Olivares-Mendez, M.A., Voos, H.: Area exploration with a swarm of UAVs combining deterministic chaotic ant colony mobility with position MPC. In: Proc. of IEEE International Conference on Unmanned Aircraft Systems (ICUAS) (2017). DOI 10.1109/icuas.2017.7991418

28. Rössler, O.E.: An equation for continuous chaos. Physics Letters A **57**(5), 397–398 (1976). DOI 10.1016/0375-9601(76)90101-8

29. Schleich, J., Panchapakesan, A., Danoy, G., Bouvry, P.: UAV fleet area coverage with network connectivity constraint. In: Proc. of ACM International Symposium on Mobility Management and Wireless Access (MobiWac'13) (2013). DOI 10.1145/2508222.2508225

30. Shimizu, Y., Ohtsuka, T., Diehl, M.: A real-time algorithm for nonlinear receding horizon control using multiple shooting and continuation/krylov method. International Journal of Robust and Nonlinear Control **19**(8), 919–936 (2009). DOI 10.1002/rnc.1363

31. Tripathi, A.K., Padhi, R.: Reactive collision avoidance of UAVs with simple pin-hole camera based passive stereovision sensing. Unmanned Systems **04**(02), 129–153 (2016). DOI 10.1142/s2301385016500023

32. Varrette, S., Bouvry, P., Cartiaux, H., Georgatos, F.: Management of an academic HPC cluster: The UL experience. In: Proc. of IEEE International Conference on High Performance Computing & Simulation (HPCS). Bologna, Italy (2014). DOI 10.1109/HPCSim.2014.6903792