ROSALIE, M., DENTLER, J.E., DANOY, G., BOUVRY, P., KANNAN, S., OLIVARES-MENDEZ, M.A. and VOOS, H. 2017. Area exploration with a swarm of UAVs combining deterministic chaotic ant colony mobility with position MPC. In *Proceedings of the 2017 International conference on unmanned aircraft systems (ICUAS 2017), 13-16 June 2017, Miami, USA*. Piscataway: IEEE [online], pages 1392-1397. Available from: https://doi.org/10.1109/ICUAS.2017.7991418

Area exploration with a swarm of UAVs combining deterministic chaotic ant colony mobility with position MPC.

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

2017

© 2017 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

OpenAIR @RGU

This document was downloaded from https://openair.rgu.ac.uk

SEE TERMS OF USE IN BOX ABOVE

DISTRIBUTED UNDER LICENCE

2017 International Conference on
Unmanned Aircraft Systems (ICUAS)
June 13-16, 2017, Miami, FL, USA

Area exploration with a swarm of UAVs combining deterministic Chaotic Ant Colony Mobility with position MPC

Martin Rosalie[1], Jan E. Dentler[1*], Grégoire Danoy[2], Pascal Bouvry[2],
Somasundar Kannan[1], Miguel A. Olivares-Mendez[1] and Holger Voos[3]

***Abstract*— The recent advances in Unmanned Aerial Vehicles (UAVs) technology permit to develop new usages for them. One of the current challenges is to operate UAVs as an autonomous swarm. In this domain we already proposed a new mobility model using Ant Colony Algorithms combined with chaotic dynamics (CACOC) to enhance the coverage of an area by a swarm of UAVs. In this paper we propose to consider this mobility model as waypoints for real UAVs. A control model of the UAVs is deployed to test the efficiency of the coverage of an area by the swarm. We have tested our approach in a realistic robotics simulator (V-Rep) which is connected with ROS. We compare the performance in terms of coverage using several metrics to ensure that this mobility model is efficient for real UAVs.**

I. Introduction

The extensive operational space and sensing capabilities of UAVs facilitate the exploration of areas. By using multiple autonomously operating UAVs, the response time in search and rescue scenarios within large areas can be reduced significantly. 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 [1] and [2]. A detailed recent state of art of the multi-UAV area exploration problem from a perspective of optimization and artificial intelligence is summarized in [3]. 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 [4] as Ant Colony Optimization with pheromones to coordinate a swarm of UAVs. This deterministic path planning is in contrast to Dynamic Data Driven Application Systems (DDDAS, see [5] for an example applied on swarm of UAVs), where the UAVs are coordinated according to the data they collected.

Kuiper and Nadjm-Tehrani [6] define an area exploration scenario with ten UAVs with a pheromone based approach for a path planning. Their study compares a random mobility model to a pheromone-based mobility model using few metrics for coverage. Some recent studies propose to replace the random part of such metaheuristics to increase their performance. For instance Gandomi *et al.* use this principle for Particle Swarm Optimization method [7] and Firefly Algorithm [8]. Thus, in our previous work [9] we use chaotic dynamic to create the Chaotic Ant Colony Optimization to Coverage (CACOC) mobility model to increase the area coverage performance of an Ant Colony Algorithm. However, the high-level simulations used to evaluate CACOC do not take into account the dynamics of a real quadrotor, which are limiting the unpredictability of a real UAV trajectory. To analyze the influence of the UAV dynamics, the major contribution of this paper is a stochastic evaluation of the performance of CACOC in terms of coverage in a realistic scenario.

The structure of this use case exploration scenario is illustrated in Fig.1. The latter consists of ten UAV models which are approximating a real quadrotor motion as introduced in some of the authors previous work [10]. The positions of these UAVs are controlled with model predictive control (MPC) as presented in the same paper. The system dynamics are equal to the MPC inherent prediction model which has been applied in [10] on real quadrotors and therefore justifies its representative usage. To finally assess the influence of the closed-loop UAV dynamics, the waypoints which are generated with the CACOC mobility model (indicated as *mobility model*) are given as target position for the MPC controlled UAV (indicated as *control*).

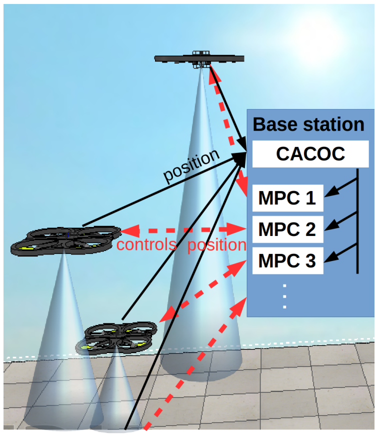

Fig. 1. Use case: implementation structure of area exploration scenario.

[1]SnT, University of Luxembourg, Luxembourg
martin.rosalie@uni.lu
jan.dentler@uni.lu
somasundar.kannan@uni.lu
miguel.olivaresmendez@uni.lu
[2]SnT/FSTC-CSC, University of Luxembourg, Luxembourg
gregoire.danoy@uni.lu
pascal.bouvry@uni.lu
[3]SnT/FSTC-RUES, University of Luxembourg, Luxembourg
holger.voos@uni.lu

[*]Supported by FNR ”Fonds National de la Recherche” (Luxembourg) through AFR ”Aides à la Formation-Recherche” Ph.D. grant scheme No. 9312118.

The remainder of this paper is organized as follows. We first provide the description of the CACOC UAV mobility model and the control model of the UAVs in section II. Then the experimentation environment, the metrics used to compare the model and statistical analysis are given in section III. In section IV we finally provide conclusions and future work perspectives that this approach provides.

II. UAV Mobility Model

This section first provides a brief description of the CACOC mobility model used to generate the waypoints to be reached by the UAVs. The second part introduces the model predictive position control (MPC) of a real quadrotor used in the experiments.

A. CACOC mobility model (Waypoints)

In order to generate the different waypoints to be followed by the UAVs, the CACOC mobility model introduced in one of our previous works [9] has been used. In a high level simulation process, CACOC is 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 [9]. We empirically demonstrated that CACOC permits an increase of performance for several aspects of coverage (overall coverage, recent coverage and fairness of the coverage).

This section provides a brief description of this mobility model, for more details the interested reader should refer to [9]. CACOC is based on the idea that the chaotic behavior of a dynamical system can enhance the performance of optimization algorithms. More precisely, CACOC replaces the random parts of the ant colony optimization algorithm [11] by a chaotic behavior in such a way that the exploration capabilities of the UAV swarm is improved. The CACOC performance on the coverage problem was empirically demonstrated using metrics as described below in Sec. III.

For the chaotic dynamics, we used the Rössler system [12] because it is a well-known dynamical system that can exhibit various chaotic mechanisms [13]. CACOC works as follows: UAVs have a constant speed and choose at each step a direction: A for ahead, R for 45° right and L for 45° left. If there is no virtual pheromone to guide the UAV (pheromones are deposited by each UAV to indicate areas they already visited), then the next choice is given by the first return map (Fig. 2). This map underline 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;
- if $1/3 < \rho_n < 2/3$ then direction is left;
- else the direction is ahead.

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). If there are pheromones, they have repulsive properties and the next choice 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 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 $p_R + p_L + p_A = 1$. 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 $p_R < \rho_n < p_R + p_L$ next direction is left;
- else the direction is ahead.

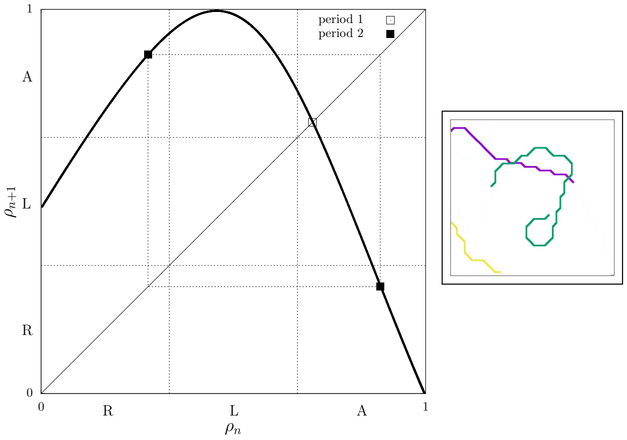

Fig. 2. Details of the CACOC mobility model with the first return maps from the Rössler system and the pattern it generates when there is no pheromone to guide the UAVs. The next choice depends on the previous one with: A for ahead, L for left and R for right. The period 1 orbits leads to symbols AAAAA.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

We demonstrated that the performance of the mobility model is increased with this chaotic behavior instead of a random one (see [9] for details). CACOC relies on a simplified quadrotor model, this article therefore proposes to use a realistic one to evaluate CACOC performance in a physical environment. In this work, the CACOC mobility model provides waypoints that UAVs will try to reach. For the sake of clarity, the mobility model is referred to as *Waypoints* in the remainder of this article.

B. UAV control (Controlled)

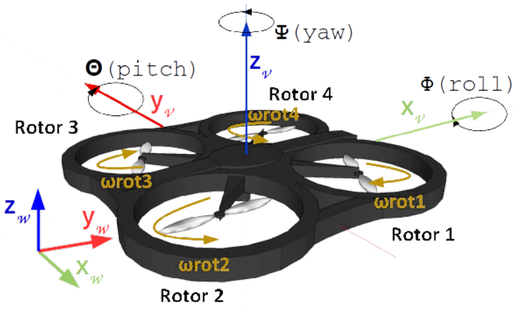

Fig. 3. Quadrotor system.

As representation of a real quadrotor system, the motion model of an AR.Drone 2.0 is used in this work. Its system dynamics according to Fig. 3 and model predictive control (MPC) have been presented in some of the authors previous work [10]. Considering multiple quadrotors, the dynamics of quadrotor i can be described, using state $\mathbf{x}_i$ and control vector $\mathbf{u}_i$:

$$\mathbf{x}_i = \left[x_i, y_i, z_i, \Psi_i, v_{f,i}, v_{s,i}\right] \quad (1)$$
$$\mathbf{u}_i = \left[u_{f,i}, u_{s,i}, u_{z,i}, u_{\Psi,i}\right]. \quad (2)$$

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 (1) consists of the quadrotor's x_i, y_i, z_i position, its yaw angle Ψ_i and the forward $v_{f,i}$ and sidewards velocities $v_{s,i}$. Its control vector (2) concatenates forward $u_{f,i}$, sidewards $u_{s,i}$, upward $u_{z,i}$ and yaw $u_{\Psi,i}$ velocity. The approximation of the real quadrotor's dynamics by means of system identification, results in:

$$\dot{\mathbf{x}}_i = \mathbf{f}_i\left(\mathbf{x}_i, \mathbf{u}_i\right) = \begin{bmatrix} v_{f,i}\cos\left(\Psi_i\right) - v_{s,i}\sin\left(\Psi_i\right) \\ v_{f,i}\sin\left(\Psi_i\right) + v_{s,i}\cos\left(\Psi_i\right) \\ u_{z,i} \\ 1.6u_{\Psi,i} \\ -0.5092v_{f,i} + 1.458u_{f,i} \\ -0.5092v_{s,i} + 1.458u_{s,i} \end{bmatrix}. \quad (3)$$

The scalar parameters in (3) have been identified from a real AR.Drone 2.0 system by means of a motion capture system. This system model is used in an MPC approach, which computes optimal controls by solving the following optimal control problem (OCP):

$$\min_{\mathbf{u}_i} J_i = \int_{t_0}^{t_f} \left(\mathbf{x}_i^* - \mathbf{x}_i\right)^\top \mathbf{Q}\left(\mathbf{x}_i^* - \mathbf{x}_i\right) + \mathbf{u}_i^\top \mathbf{R}\mathbf{u}_i \;\, \mathrm{d}\tau \quad (4)$$

s.t.

$$\dot{\mathbf{x}}_i = \mathbf{f}_i\left(\mathbf{x}_i, \mathbf{u}_i\right) \quad (5)$$
$$\mathbf{c} \leq \left[u_f\left(t\right)^2 - 1 \;\, u_s\left(t\right)^2 - 1 \;\, u_z\left(t\right)^2 - 1 \;\, u_\Psi^2 - 1\right]^\top \quad (6)$$
$$\mathbf{x}_i\left(0\right) = \left[50, 0, i, 0, 0, 0\right] \quad (7)$$
$$\mathbf{Q} = \mathcal{D}\{\left[1.5, 1.5, 1.6, 0.1, 0, 0\right]\}, \quad (8)$$
$$\mathbf{R} = \mathcal{D}\{\left[1, 1, 1, 1\right]\} \quad (9)$$

for each drone $i = 1...10$ over a receding horizon. The system dynamics (3) are represented as constraints (5). The system inputs are defined between $-1 \leq u \leq 1$, which is taken into account with constraint (6). To avoid collisions, the quadrotors initial positions $\mathbf{x}_i\left(0\right)$ differ in their height (7). The tracking of the desired state $\mathbf{x}_i^*$ is achieved by a quadratic state penalty with $\mathbf{Q}$ (8) in cost-function J_i (4). 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 $z_i = i$ and $\psi_i = 0$ are tracked. Furthermore the quadratic control penalty with $\mathbf{R}$ (9) is minimizing the control effort and thus the energy consumption.

Analog to the previous work [10], a Condensed Multiple Shooting Continuation Generalized Minimal Residual (CMSCGMRES) [14] approach is used to solve the MPC with OCP (4)-(9). The CMSCGMRES is parameterized with a maximum number of iterations $k_{max} = 6$, horizon length $T = 1$s, control update interval of $\Delta T = 0.1$s, $n_{hor} = 10$ steps within the horizon, a forward difference step of $h = 0.001$s, solution tolerance of $\epsilon = 10^{-4}$, continuation factor $\zeta = 10$ and the horizon expansion factor $\alpha = 2$. More details about the parameters are given in [15].

III. Experimentation

The first part of this section provides information on the simulation framework used for the experiments. Then the experimental setup, including all parameters and metrics used are detailed. Finally the last part contains the experimental results and their analysis.

A. Simulation framework

For the experimental validation, the quadrotor dynamics (3) are implemented in the V-Rep simulation environment [16]. The MPC is interfaced via ROS. An example of a scenario with three controlled UAVs is shown in Fig. 4.

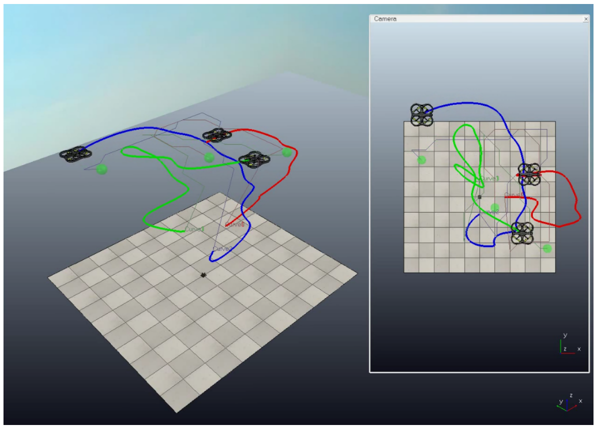

Fig. 4. Three MPC controlled UAVs in the V-Rep environment following waypoints provided by CACOC for three UAVs.

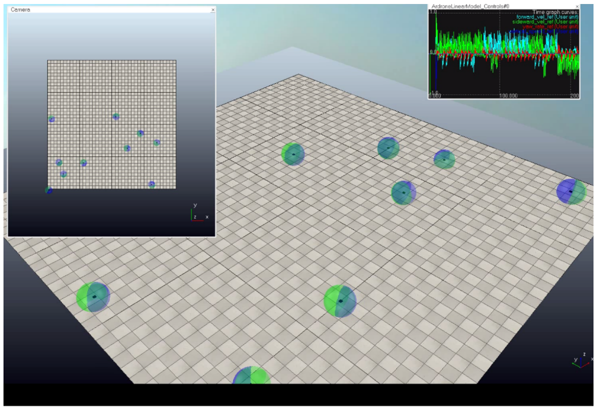

Fig. 5. Simulation environment of use case of CACOC exploration with 10 MPC controlled UAVs.

For the evaluation of the large area coverage with ten UAVs, as illustrated in Fig. 5, the 3D environment data is mapped onto the xy-plane as shown in Fig. 6 for 10 UAVs[1].

[1] A video illustrating the framework simulator is available at https://martinrosalie.gforge.uni.lu/_downloads/icuas.mp4

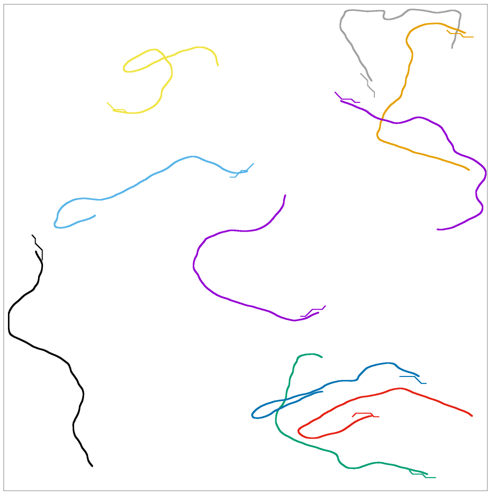

Fig. 6. Ten controlled UAVs trajectories in the squared area where the thickest straight lines are links between the waypoints they have to reach.

This figure permits to highlight the smoother trajectory of the controlled UAVs (thick line) in comparison to the discontinuous velocity trajectories (thin line) of the *CACOC mobility model*.

B. Experimental setup

The simulation is performed using the environment shown in Tab. I. The same three area coverage metrics as introduced in [9] are used. A short description is provided hereinafter. Fig. 7 underlines how the metrics are calculated for one simulation.

TABLE I
MAIN EXPERIMENTAL PARAMETERS.

Parameter Name	Parameter Value
Simulation area	
Geographical Area	100 m × 100 m
Number of cells	100 × 100
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	Waypoint, Controlled
Number of UAVs	10
Simulation steps	4000
Independent runs	30

1) Coverage: The coverage is the percentage of the total area visited during the whole simulation. The environment is a 100 m by 100 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. We extract the slope of a linear regression $a \times x$ considering the 500 first steps (Fig. 7).

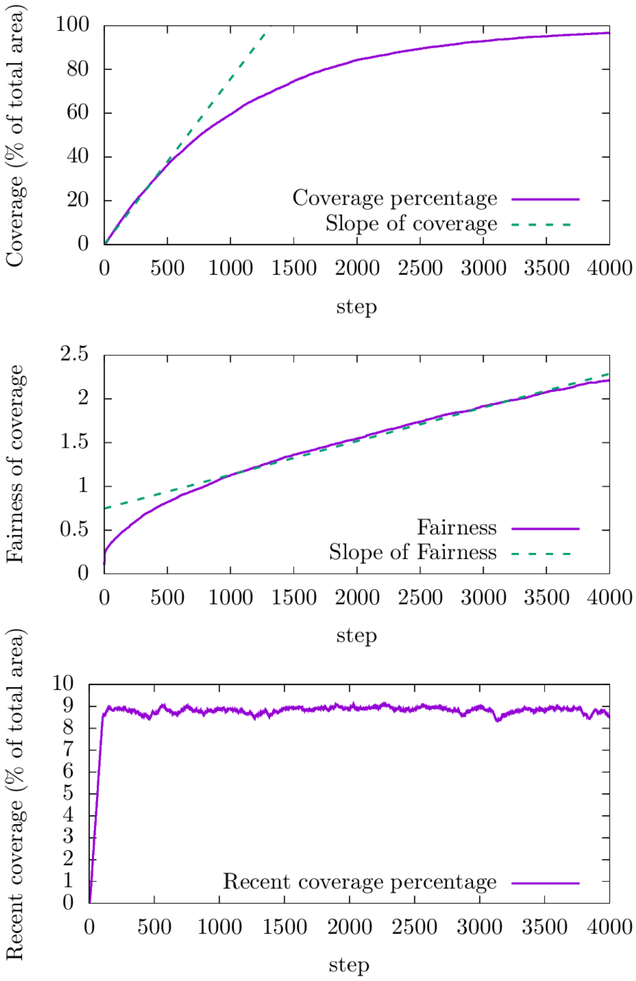

Fig. 7. Metrics measurements for one simulation. Two values are extract for the coverage: the coverage after 4000 steps and the slope of coverage for the 500 first steps. For the fairness we compute the slope of the fairness after the 500 first steps. The average value of the recent coverage is calculated after the 100 first steps.

2) Fairness: The fairness measures if all cells are regularly and equally scanned. This is computed as the standard deviation of their respective number of scans [17]. To evaluate the fairness during the whole simulation, we perform a linear regression $a \times x + b$ using the last 3500 steps (Fig. 7). 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.

3) Recent coverage ratio: This metric introduced in [18] represents the percentage of coverage during the last 100 iterations. These 100 steps correspond to the pheromones' evaporation time. We exclude the 100 first iterations of our simulation to compute the mean value of the recent coverage (Fig. 7).

C. 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 deterministic mobility model, we will only vary the initial conditions of the dynamical system to create a wide range of simulations with the same mobility model without changing its parametrization as it has already been done previously [9].

TABLE II

$p-$VALUES OF TESTS TO HIGHLIGHT STATISTICAL DIFFERENCE BETWEEN *Waypoints* AND *Controlled* METHODS.

Metric	Test	*Waypoints*	*Controlled*
Coverage after 4000 steps	Shapiro-Wilks	0.06068	0.07211
	F-test	0.3859	
	Student's test	9.095e-06	
Slope of Coverage	Shapiro-Wilks	0.06752	0.4839
	F-test	0.9323	
	Student's test	4.367e-08	
Slope of Fairness	Shapiro-Wilks	0.9276	0.7917
	F-test	0.7721	
	Student's test	0.02913	
Recent Coverage	Shapiro-Wilks	0.5773	2.539e-05
	F-test	5.696e-07	
	Wilcoxon test	$\leq$ 2.2e-16	

Fig. 8 gives the box plot for each metric and method with a smooth density estimation. It should be noted that some of the results presented here are not comparable to our previous simulations [9] because their duration is shorter due to the time necessary to compute the *controlled* model. Associated with the Fig. 8 we perform pairwise statistical tests. The purpose of the Shapiro-Wilks test is to evaluate if the values are normally distributed. With 95% confidence, this hypothesis is satisfied for our results (except the recent coverage of the *controlled* model) because the $p-$value ≤ 0.05.

The F-test evaluates the equality of the variance with the same mechanism for the hypothesis for a given $p-$value. The variances are similar for all the metrics except for the recent coverage one with 95% confidence. Finally, if the data are normally distributed with equality of the variance, we can perform a Student's test to check if the data follow the same distribution. Else, a Wilcoxon signed rank test is performed for not normally distributed data. With 95% of confidence we can say that the metric data of the *Waypoints* and *Controlled* model do not follow the same distribution. Even though, Tab. II indicates that the *slope of fairness* is the metric that is the most similar: the $p-$value of the Student's test is equal to 0.02913. We assume that the values for this metric are close because of the use of a linear regression over 87% of the simulation time. For this metric, the average value is higher for the *Controlled* model than for the *Waypoints* the average value is higher (Fig. 7). This means that the coverage is more fair for the *Waypoints* method than for the *Controlled* one. This indicates that despite some small change in the trajectories of the UAVs, the models remain globally efficient considering this metric.

This is not the case for the *recent coverage* because the gap between the two average values represent 10% of the optimal value (see [9] for details). This metric evaluates the performance on a short term giving the percentage of

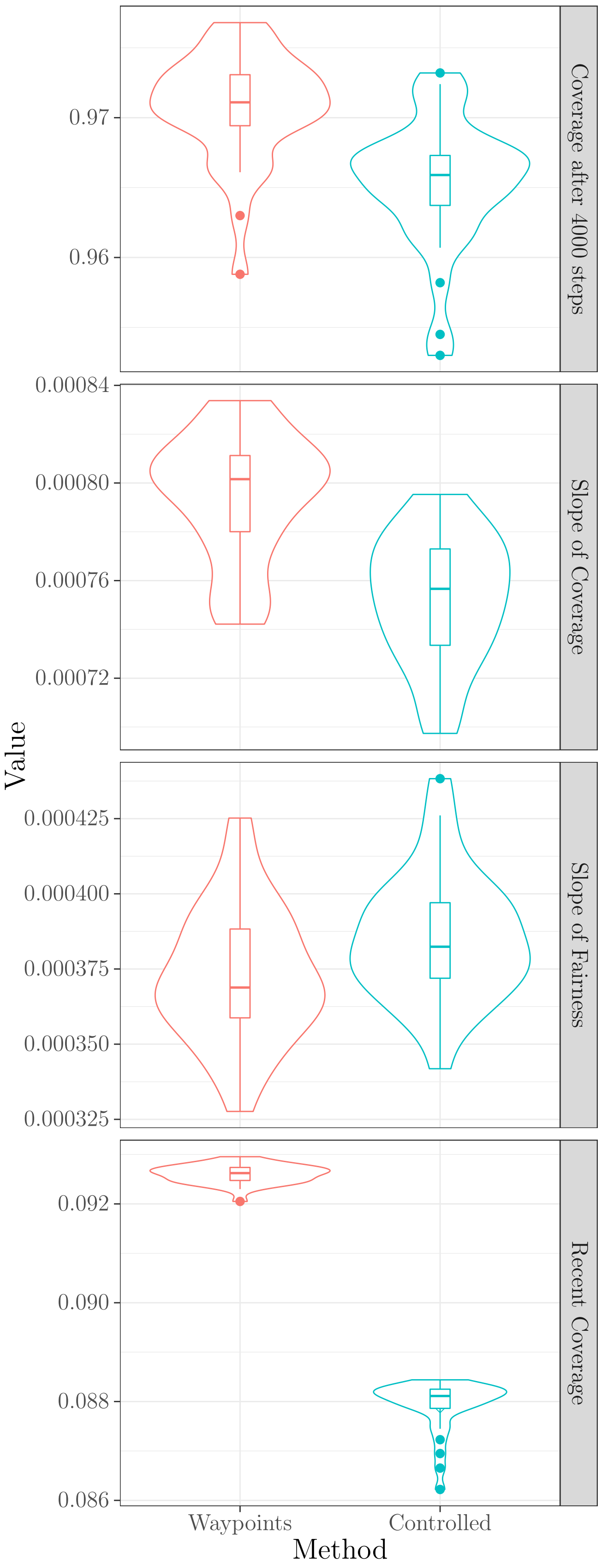

Fig. 8. Metrics comparison between *Waypoints* and *Controlled* methods. The smooth density estimation is given in addition of the box plot.

area covered during the last 100 steps. If we only consider this metric we can erroneously consider that the controlled method does not perform as well as the *Waypoints* model.

Considering the whole statistics, we can conclude that the global performance of the controlled method is very satisfactory. As previously mentioned, the most noticeable performance degradation is for the recent coverage performance. Conversely, the quality of slope of fairness is very similar between the two methods. In addition, it seems that the shift between the expected position and the real one are compensated on a long term for the coverage. Due the position tracking, the quadrotor asymptotically approaches the desired target points. For moving targets as in the present work, this delay leads to a position tracking error. This explains the difference in the slope of coverage. Furthermore this position tracking error results in less observed boundary areas, as all target points lie within the given rectangular region. Accordingly, the coverage in the studied scenario is slightly lower for the controlled system. As this difference in coverage is the direct result of the closed loop dynamics of the controlled quadrotor system, this difference is supposed to decrease with an increasing size of the observed area.

IV. Conclusion

In this paper, we compared the coverage performance of a state-of-the-art UAV swarm mobility model (CACOC) against its physical simulation in a robotic dedicated environment. An empirical evaluation of both the initial *Waypoints* model and of the *Controlled* model was conducted on a 10 UAVs scenario. The difference between these two simulations was evaluated using four area coverage metrics and assessed with statistical tests. The obtained results show that the presented CACOC-MPC approach implemented in the robotics simulator was able to accomplish the covering task efficiently. Indeed, the high similarity of the results between the *Waypoints* and *Controlled* demonstrates that it can be expected that it will perform well in a real environment with the only consideration of a fine tuning of the MPC approach.

Future 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 [10]. This 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 plan to work on collision avoidance for the UAVs. In this work we consider that each UAV has its own altitude but we want to compare the performance of CACOC waypoints with controlled UAVs at the same altitude. The main question we would like to address is the following: is it statistically efficient to implement a collision avoidance mechanism for the *Waypoints* or the collision avoidance included in the *Controlled* method is sufficient?

Acknowledgments

The experiments presented in this paper were carried out using the HPC facilities of the University of Luxembourg [19] (see http://hpc.uni.lu).

References

[1] V. Gazi and K. M. Passino, *Swarm Coordination and Control Problems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 15–25.

[2] S. Bertrand, J. Marzat, H. Piet-Lahanier, A. Kahn, and Y. Rochefort, “MPC Strategies for Cooperative Guidance of Autonomous Vehicles,” *AerospaceLab*, no. 8, pp. 1–18, 2014.

[3] N. Nigam, “The multiple unmanned air vehicle persistent surveillance problem: A review,” *Machines*, vol. 2, no. 1, pp. 13–72, 2014.

[4] P. Gaudiano, B. Shargel, E. Bonabeau, and B. T. Clough, “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.

[5] R. R. McCune and G. R. Madey, “Swarm control of UAVs for cooperative hunting with DDDAS,” *Procedia Computer Science*, vol. 18, pp. 2537–2544, 2013.

[6] E. Kuiper and S. Nadjm-Tehrani, “Mobility models for UAV group reconnaissance applications,” in *Proc. of IEEE International Conference on Wireless and Mobile Communications (ICWMC’06)*, 2006.

[7] A. H. Gandomi, G. J. Yun, X.-S. Yang, and S. Talatahari, “Chaos-enhanced accelerated particle swarm optimization,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 2, pp. 327–340, 2013.

[8] A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, “Firefly algorithm with chaos,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 1, pp. 89–98, 2013.

[9] M. Rosalie, G. Danoy, S. Chaumette, and P. Bouvry, “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.

[10] J. Dentler, S. Kannan, M. A. Olivares-Mendez, and H. Voos, “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.

[11] M. Dorigo, “Optimization, learning and natural algorithms,” Ph.D. dissertation, Politecnico di Milano, Italy, 1992.

[12] O. E. Rössler, “An equation for continuous chaos,” *Physics Letters A*, vol. 57, no. 5, pp. 397–398, 1976.

[13] M. Rosalie, “Templates and subtemplates of rössler attractors from a bifurcation diagram,” *Journal of Physics A: Mathematical and Theoretical*, vol. 49, no. 31, p. 315101, 2016.

[14] Y. Shimizu, T. Ohtsuka, and M. Diehl, “A real-time algorithm for nonlinear receding horizon control using multiple shooting and continuation/krylov method,” *International Journal of Robust and Nonlinear Control*, vol. 19, no. 8, pp. 919–936, 2009.

[15] T. Ohtsuka, “A continuation/gmres method for fast computation of nonlinear receding horizon control,” *Automatica*, vol. 40, no. 4, pp. 563–574, 2004.

[16] E. Rohmer, S. P. N. Singh, and M. Freese, “V-rep: a versatile and scalable robot simulation framework,” in *Proc. of IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[17] J. Schleich, A. Panchapakesan, G. Danoy, and P. Bouvry, “UAV fleet area coverage with network connectivity constraint,” in *Proc. of ACM International Symposium on Mobility Management and Wireless Access (MobiWac’13)*, 2013.

[18] G. Danoy, M. R. Brust, and P. Bouvry, “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.

[19] S. Varrette, P. Bouvry, H. Cartiaux, and F. Georgatos, “Management of an academic hpc cluster: The UL experience,” in *Proc. of IEEE International Conference on High Performance Computing & Simulation (HPCS)*, Bologna, Italy, 2014.