HAL
open science

From random process to chaotic behavior in swarms of UAVs

Martin Rosalie, Grégoire Danoy, Serge Chaumette, Pascal Bouvry

To cite this version:

Martin Rosalie, Grégoire Danoy, Serge Chaumette, Pascal Bouvry. From random process to chaotic behavior in swarms of UAVs. 6th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications, Nov 2016, Malte, Malta. pp.9 - 15, 10.1145/2989275.2989281 . hal-01360176

HAL Id: hal-01360176

https://hal.science/hal-01360176v1

Submitted on 5 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

From random process to chaotic behavior in swarms of UAVs

Martin Rosalie
University of Luxembourg
SnT Interdisciplinary Centre
Luxembourg
Univ. Bordeaux
LaBRI, UMR5800
F-33400 Talence, France
martin.rosalie@uni.lu

Grégoire Danoy
University of Luxembourg
SnT/FSTC-CSC
Luxembourg
gregoire.danoy@uni.lu

Serge Chaumette
Univ. Bordeaux
LaBRI, UMR5800
F-33400 Talence, France
serge.chaumette@labri.fr

Pascal Bouvry
University of Luxembourg
SnT/FSTC-CSC
Luxembourg
pascal.bouvry@uni.lu

ABSTRACT

Unmanned Aerial Vehicles (UAVs) applications have seen an important increase in the last decade for both military and civilian applications ranging from fire and high seas rescue to military surveillance and target detection. While this technology is now mature for a single UAV, new methods are needed to operate UAVs in swarms, also referred to as fleets. This work focuses on the mobility management of one single autonomous swarm of UAVs which mission is to cover a given area in order to collect information. Several constraints are applied to the swarm to solve this problem due to the military context.

First, the UAVs mobility must be as unpredictable as possible to prevent any UAV tracking. However the Ground Control Station (GCS) operator(s) still needs to be able to forecast the UAVs paths. Finally, the UAVs are autonomous in order to guarantee the mission continuity in a hostile environment and the method must be distributed to ensure fault-tolerance of the system. To solve this problem, we introduce the Chaotic Ant Colony Optimization to Coverage (CACOC) algorithm that combines an Ant Colony Optimization approach (ACO) with a chaotic dynamical system. CACOC permits to obtain a deterministic but unpredictable system.

Its performance is compared to other state-of-the art models from the literature using several coverage-related metrics, i.e. coverage rate, recent coverage and fairness. Numerical results obtained by simulation underline the performance of our CACOC method: a deterministic method with unpredictable UAV trajectories that still ensures a high area coverage.

Keywords

cooperative UAVs; multilevel swarms; mobility models; ACO; chaotic systems

1. INTRODUCTION

The purpose of the ASIMUT project (Aid to SItuation Management based on MUltimodal, MUltiUAVs, MUltilevel acquisition Techniques), supported by the European Defence Agency (EDA), is to improve the situation awareness of an operator through area coverage and detection of threats based on multi-sensor and multi-source data fusion (see [14] for more details concerning this project). Information is delivered by heterogeneous swarms of autonomous Unmanned Aerial Vehicles (UAVs) flying at different altitudes. One of the objective of the ASIMUT project addressed here is area surveillance that is ensured by means of a swarm of UAVs. These UAVs collect data with their embedded sensors. It should be noted that the operation takes place in a military context.

Even if this paper is dedicated to UAVs, the problem remains the same in the case of other unmanned vehicles like ground, surface or underwater vehicles. To solve this problem two main techniques have been proposed : online and offline path planning. Offline planning consists in precomputing the flight plan of the UAVs. The main asset of this approach is that the UAVs trajectories are easily monitored from the Ground Control Station (GCS). However, these techniques are not adaptive to any change of configuration during the flight: the scheduled path can be irrelevant by the time the UAVs execute it. On the other hand, online methods compute the trajectories of the UAVs at runtime. The advantages and drawbacks of online methods are the opposite of those of to offline systems. As a consequence, in this paper we will propose a method that combines the assets of both online and offline methods.

To build such a system, we choose to use the Ant Colony Optimization method (ACO) introduced by Dorigo [4]. We mainly rely on the work of Kuiper & Nadjm-Tehrani [10] who adapted the ACO algorithm to the coverage problem for UAVs. The latter uses repulsive pheromones to guide the UAVs over the area they have to cover. The UAVs share a map of virtual pheromones that indicate recently visited areas when high pheromone concentrations are present. The UAVs then have a higher probability to move to the least recently visited areas. The drawback of this method is that it uses a random process which prevents any offline path planing.

The purpose of our paper is to change the random process of ACO system for a chaotic dynamical system. A chaotic dynamic is the solution to a deterministic system with the following properties. The solution is bounded and sensitive to initial conditions, and consequently, unpredictable on a long-term. We thus want to introduce these deterministic but unpredictable solutions in our UAV mobility model[1]. The recent paper [18] provides a detailed review of various applications of chaotic dynamics for mobile robots. Various systems are used to generate chaotic dynamics. In most cases, the authors use the logistic map $x_{n+1} = \alpha x_n(1 - x_n)$ to do it. A map is an iterative application that does not diverge, neither converges to a point. This logistic map converges to a dense set of points, between 0 and 1. This map generates a fully distributed chaos when $\alpha = 4$ and it produces an output $x_n \in [0 : 1]$. We assume that the simplicity of this equation combined with an easy to use output contributes to find this equation in many papers. For instance Iba & Shimonishi [8] use it to define a chaotic walk (in comparison to the *random walk*). The shape of their process depends on the number of decimals they use to compute the next iteration. This underlines how chaotic processes are sensitive to the initial condition. Gong & Wang [7] also introduce a chaotic process in an ACO algorithm by testing various parameters for the logistic map to improve the results of the algorithm. This map is also successfully used with an Ant Colony Algorithm in order to find best UAV path between threats ([20] or [19]).

However, Arroyo *et al.* [2] detail some of the properties of the logistic map (including probability distributions) while the parameter α is varied. They highlight that this equation exhibits some periodic behavior between the values of the parameters used in [7]. The same ranges of parameters are used by Li *et al.* [11] to solve the Traveling Salesman Problem (TSP). This highlights the fact that the logistic map works as a chaotic generator but for a small range of parameter values. Thus the logistic map is a good start but not sufficient to explore the capacities and richness of chaotic dynamical systems. Some recent works have investigated further the integration of chaotic systems in optimization algorithms. Ahmadi et al. in [1] use Ordinary Differential Equations (ODE) systems to combine chaos and the weed invasive algorithm. Also the authors of [6] use various chaotic maps (tent map, Lozi map, etc.) to explore the performance of their firefly algorithm. We choose to use the Rössler system [15] as a basic system to explore a combination of chaotic behaviour from ODE with Ant Colony Algorithm. This system is a reference in the literature as one of the first ODE systems with a simple chaotic mechanism.

[1]In the frame of the ASIMUT project, a real deployment of this model is planned on UAVs.

This work is organized as follows. We first introduce the two main topics: ACO and chaos. The second part is dedicated to the description of the problem we solve and the related work. In the third section, we present our models: one only based on the Rössler system and then CACOC. The following section contains our experimentations with the metrics and the statistics we produce. We finally give a conclusion and describe our future work.

2. PROBLEM DEFINITION AND RELATED WORK

2.1 Problem definition

In this paper, we consider a problem similar to the one defined by Kuiper and Nadjm-Tehrani [10]: a squared area has to be covered by a swarm of UAVs. In addition to the non predictability constraint induced by the military context, the trajectories of the UAVs still need to be monitored from the Ground Control Station (GCS). It is indeed mandatory for users of this type of system to know and anticipate the positions of their UAVs. We remind that we also need to have an adaptive method that is resilient to UAVs failures or loss of UAV. As a consequence, the problem we address here is at the edge of the path planning for UAVs and the autonomous distributed coverage. This is not a path planning problem because of the resilience constraint.

Because of the non predictability constraint, articles giving the optimized solution of a coverage problem [5] have not been considered. Indeed, in such approaches the UAVs coverage pattern is too explicit to be used in a military context. In this section we present the algorithms from the literature that have been compared to our method.

2.2 Related work

In this paper we address the following problem: an area has to be visited regularly by UAVs in order to collect information. One of the constraints is that an observer should not be able to anticipate the reconnaissance pattern of the swarm. One solution proposed by Kuiper and Nadjm-Tehrani [10] was to introduce a random process to prevent the UAVs trajectories to be predicted. In the same article the authors also proposed one ACO algorithm and compared it to the random appraoch. These two models are presented in detail hereinafter. We also present a third model from the literature, referred to as the *chaotic walk* introduced by Iba & Shimonishi [8].

2.2.1 Model 1: Random-based mobility model [10]

The next direction of the UAV depends on the previous one and is chosen using the probability rules given in Table 1. The initial state of each UAV is considered as "straight ahead".

Model 1	Probability of action		
Last action	Left	Ahead	Right
Left	0.70	0.30	0
Ahead	0.10	0.80	0.10
Right	0	0.70	0.30

Table 1: Random action table for Model 1.

2.2.2 Model 2: Pheromone-based mobility model [10]

This method is a distributed pheromone repel mobility model. While UAVs fly, they also deposit virtual pheromones that indicate recently visited geographical areas and that evaporate over time. If there is pheromone in its neighbourhood, the UAV chooses its next direction using the probability indicated in Table 2. If there is no pheromone to guide the UAV, it will use the random action of Model 1 based on its previous action.

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

Table 2: Pheromone action table for Model 2. $left$ **is the amount of pheromone sensed at the left of the UAV,** $ahead$ **is the amount of pheromone sensed in front of the UAV and** $right$ **is the amount of pheromone sensed at the right of the UAV; finally** $total = left + ahead + right$**.**

2.2.3 Model 3: Chaotic walk [8]

The process described here was introduced by Iba & Shimonishi [8]. The authors name it *chaotic walk* because it only uses the logistic map to determine the next angular direction of an entity

$$x_{n+1} = 4x_n(1 - x_n)$$
$$\theta_{n+1} = 2\pi x_{n+1} \ .$$

The logistic map makes it possible to obtain $x_n \in [0 : 1]$. In our case, UAVs have constant speed and the angle is given by θ_{n+1}. Even if this method is not dedicated to coverage, we implemented it to evaluate its performance as we did for the previous models. As a consequence this simple chaotic walk serves as a basic chaotic mobility model to compare our more developed method.

3. UAV MOBILITY MODELS WITH CHAOS

In the previously presented methods, ACO is a good candidate for the coverage problem because the algorithm is designed in a distributed way. However, this algorithm uses probabilities to let the UAV choose its next action; this implies to have random variables for each step. On the other hand the algorithm presented by Iba & Shimonishi using the chaotic logistic map permits to make a chaotic walk. This structure does not permit an efficient coverage but it satisfies the UAVs deterministic but unpredictable constraints. This is one of the properties of chaotic dynamical systems. In this section we propose two novel chaotic mobility models for UAV swarms to tackle the coverage problem: (1) the first one is based on the Rössler system which is a different chaotic system and (2) the second one is CACOC, that combines the chaotic dynamics of the Rössler system with an ACO approach.

In methods using meta-heuristics, random processes are used to generate variability and explore the possible solutions. In order to reproduce the simulations, the authors mainly use a seed to generate pseudo-random numbers. These pseudo-random numbers satisfy several properties such as independence and non periodicity. Here the specific *Models 1* and *2* require to use the previous values to decide on the next value. We do not want to use chaotic process to generate pseudo-random numbers as it as been done by Lozi [12]. Instead, we will build our models using the properties of chaotic dynamical systems. The fact that we can reproduce our simulation, as it is possible with seeds, is also a plus to compare the chaotic behavior versus randomness through the performance of the models. These two approaches have enough common points to be compared efficiently. In the following section we will use the specificities of chaotic systems to build a model using a chaotic process instead of a random one.

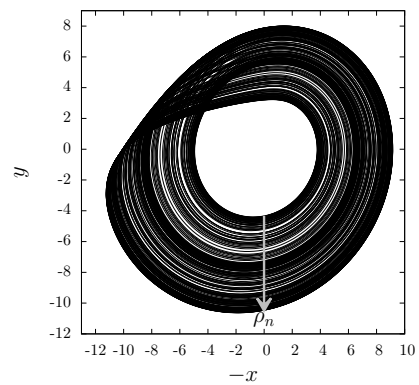

Figure 1: Rössler attractor solution to (1) **for parameters** $a = 0.1775$**,** $b = 0.215$ **and** $c = 5.995$ **with the Poincaré section** (2)**.**

3.1 Model 4: Chaos using the Rössler system

The Rössler system [15] is a well known Ordinary Differential Equation (ODE) system

$$\begin{cases} \dot{x} = -y - z \\ \dot{y} = x + ay \\ \dot{z} = b + z(x - c) \end{cases} \tag{1}$$

producing chaotic dynamics. This system is a good candidate to evaluate the impact of chaos for our systems because it has been studied several times. This system is used to validate tools and methods whose purpose is to analyse chaotic systems. As detailed in [13] when a parameter of the Rössler system is varied, it leads to several non equivalent dynamics. For this three-dimensional system with the parameters $a = 0.1775$, $b = 0.215$ and $c = 5.995$ (attractor $\mathcal{A}$ of [13]), the dynamics properties are clearly identified: the chaotic mechanism is the classical horseshoe with stretching and folding mechanisms as illustrated in Figure 1.

This system is a well documented source of various chaotic regimes. The analysis of a chaotic attractor is performed with a Poincaré section

$$P = \{(y_n, z_n) | x_n = 0, \dot{x}_n > 0\} \ . \tag{2}$$

This tool permits to discretize the continuous solution. From the Poincaré section, a first return map details the dynamical properties of the system. From the ODE system, we obtain a map (Figure 2) $\rho_{n+1} = f(\rho_n)$ with $\rho_n \in [0 : 1]$ is the normalization of the value y_n in the Poincaré section.

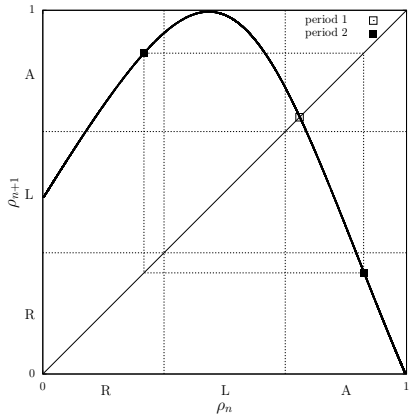

Figure 2: First return map of the Rössler attractor (Figure 1). This map is partitioned in three parts indicating the UAV directions: L (left), A (ahead) and R (right). Orbits of period 1 and 2 illustrate patterns (AAAAA...) and (ARARA...).

The arrow indicates the map orientation from the center to the outside, it corresponds of the orientation from 0 to 1.

The use of the Poincaré section permits to avoid considering what happens in the phase space (x, y, z). Instead we only focus on one value after the passage of the solution through the Poincaré section: ρ_n. This map details the dynamic properties of the Rössler system. Contrary to random processes, the independence properties imply that any value can be obtained at any iteration. Here the value is directly obtained from the previous value. To obtain the direction of the UAV from this return map, we build a partition:

- 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.

The shape of the first return map presented in Figure 2 indicates the possible transitions of the UAVs. This figure display ρ_{n+1} depending on ρ_n. It indicates the next position in the Poincaré section depending on the previous one. Thus, from a previous ahead direction (A), transition to A, left (L) and right (R) are possible. From a previous L direction, transition to A is the only possible transition. Finally, from a previous R direction, only transition to L or A are possible. We intentionally remove the symmetry process (left symmetric to right) introduced in the probabilities repartition proposed for *Model 1*. Based on some experiments that we have conduced, we believe that this absence of symmetry is in favor of a better coverage method.

To initialize the system, we choose an initial condition for each UAV to solve the Rössler system. Giving non equal initial conditions lead to the same attractor as solution and to the same first return map. However the points constituting the return maps are not the same both in terms of values and of order of appearance.

This model satisfies the problem constraints because it uses a chaotic system to have unpredictable trajectories. The GCS uses the same hardware and runs the exact same programme as the UAVs. This ensures to obtain the same decimals as the UAVs. As a consequence, operators in the GCS can monitor the UAVs positions.

3.2 Model 5: Chaotic Ant Colony Optimi-zation for Coverage (CACOC)

To satisfy all our problem constraints, we will combine the pheromone method [10] with the previous Rössler model. In the *Model 2*, random processes are used for any iteration. We propose to replace the uses of random processes with the first return map ($\rho_{n+1} = f(\rho_n)$) as we do for *Model 4*.

In the first case, it happens when there is no pheromone to guide the UAV, *Model 2* uses the random processes defined in *Model 1*. In that case, we use the *Model 4* to explore a non visited area of our CACOC model (*Model 5*). If there are some pheromones in *Model 2*, the UAV chooses its next direction with probabilities. For CACOC, we propose to replace the random choice with the next value in the return map using the probabilities repartition of Table 2 (p_L, p_A and p_R).

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

Thus, the return map values are used to choose the next direction for each step of CACOC model. The use of pheromones permits to have a system resilient to UAVs failure or loss. If a UAV is out of the system, it will no more deposit pheromones.

From the point of view of the GCS operator(s), UAVs are monitored and if any failure is detected, the operator can remove the UAV in the simulator reproducing the UAVs position from the GCS. This permits to maintain the monitoring at the GCS while keeping unpredictable UAVs trajectories.

3.3 Orbits and regular structure

For our model, we use a solution of the Rössler system that is an attractor (Figure 1). Unstable periodic orbits structure attractor dynamics and topology. From the initial condition, after the transitory time, the solution converge to the attractor. It corresponds to a trajectory that evolves in the attractor. In details, the trajectory visit the orbits of the attractor. However these orbits are unstable and the trajectory reach one of them for a moment before visiting another one.

In Figure 2 we present two low period orbits: a period 1 and a period 2 orbits. We represent these two orbits in the first return map but several others orbits can be founded for this specific attractor [13]. We focus our attention to these orbits because it explains two simple patterns for our UAVs. First, the period 1 orbits is located in the area associated to A. This means that when the system momentarily converge close to the period 1 orbits, it corresponds to a long straight trajectory. The period 2 orbits lead to an alternation of A and R (ARARA...) that corresponds to a half-turn for the UAV. These state (close to the orbits) do not persist because, these orbits are unstable. As a consequence, these regular patterns sometimes occur for our UAVs and lead them to an efficient unpredictable exploration pattern.

4. EXPERIMENTATIONS

In order to evaluate the performance of our proposed CACOC mobility model, we have compared it to the other four models presented in the previous sections using simulations. The setup used for the experiments, including the different metrics, is presented in the next subsection, while the experimental results are presented and analyzed in section 4.2.

4.1 Experimental setup

The simulation area is a 100 km × 100 km square, divided in square cells of 1 km × 1 km. The UAVs have a constant speed of 1km/s and they all depart from the base station located in the middle of the bottom edge of the map, i.e. position (50,0). At each simulation step, each UAV can do one of the following three actions: (1) ahead: the UAV keeps the same direction; (2) left: the UAV turns left with a 45° angle; (3) right: the UAV turns right with -45° angle. To prevent collisions between our UAVs they all have non equal flight altitudes [10].

Each of the five mobility models has been evaluated with a swarm of 10 autonomous UAVs equipped with wireless communication capabilities. We ran 30 independent simulations of 7000 steps eachs, to obtain significant results and applied statistical tests in order to compare the five models. We want to perform unpaired multiple comparisons [9].

Parameter Name	Parameter Value
Simulation area	
Geographical Area	100 km × 100 km
Number of cells	100 × 100
UAV Autopilot	
UAVs speed	1 km/min
Possible UAV actions	ahead, 45° left, 45° right
Initial UAVs position	middle of the bottom of the map
Experiments	
Mobility models	$[1, 2, 3, 4, 5]$
Number of UAVs	10
Simulation steps	7000
Independent runs	30

Table 3: Main experimental parameters.

In order to evaluate the performance of the different models in terms of area coverage and network connectivity, the following four metrics from [3] have been used: coverage, recent coverage ratio, fairness and connectivity. These are detailed hereinafter.

Coverage.

The coverage is the portion of the total area visited during the whole simulation. The environment is a 100 km by 100 km square area. The coverage value variates during the whole simulation. To have a representative value of the coverage, we choose to compare the coverage value after 5000 steps for each model. This indicates how efficient the models are to visit the total area. On the other side, we want to evaluate the first steps of each model to compare their initial behaviour while there is no guidelines for UAVs (e.g. no pheromones for Models 2 and 5). We will extract the slope of a linear regression ($a \times x$) considering the 500 first steps.

Recent coverage ratio.

This metric introduced in [3] 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.

Fairness.

The fairness measures if all cells are regularly and equally scanned. This is computed by the standard deviation of their respective number of scans [16]. To evaluate the fairness during the whole simulation, we perform a linear regression ($a \times x + b$) using the last 6500 steps. This comes to balance 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.

Connectivity.

Eventually, we evaluate the network connectivity of our system by measuring the number of clusters of the swarm, i.e. the number of connected components in the induced graph. We compute the mean value of the connectivity during the whole simulation to have a representative value for the different mobility models.

4.2 Experimental results

We executed 30 runs for each model and we thus obtain a distribution of 30 values for each algorithm. We produce five statistical comparisons to evaluate our algorithms. These tests also permit to assess how our method is statistically relevant compared to the others.

Models 3, 4 and 5 are deterministic. As a consequence, if we run them 30 times to compare the metrics, the solutions will be the same. We only need to introduce variability to perform statistical analysis of our models. For Models 4 and 5, we change the initial conditions of the Rössler system. We add a random value $\varepsilon \in [0 : 0.1]$ to obtain the following initial conditions $(x_0, y_0, z_0) = (-0.4 + \varepsilon + 0.1 \times \mathrm{Id}, 0, 0)$ where $\mathrm{Id} = 1, 2, \ldots, 10$ for the ten UAVs. This change is enough to obtain non identical maps for each UAV due to the sensitivity to the initial condition properties of chaotic dynamical systems. We also use the $\varepsilon + 0.1 \times \mathrm{Id}$ value as initial condition value for the logistic map (Model 3).

For the five metrics studied in this work, the 30 results do not follow normal distributions. As a consequence we performed a Kruskal-Wallis rank sum test [9] to make an unpaired multiple comparison of our model. All the results for the five metrics hereinafter have statistically significant differences with 95% confidence between any two algorithms (i.e. with a *p-value* < 0.05). We explain the good results of our Models 4 and 5 by the appearance of structured patterns when dynamics evolve close to orbit (Section 3.3).

a. Coverage after 5000 iterations.

Table 4 presents the mean coverage values after 5000 steps, together with the corresponding standard deviation, for all five mobility models. The worst is Model 3 based on the logistic map. Comparing the Kuiper and Nadjm-Tehrani methods (Models 1 and 2), the pheromo-ne based model is better. The most surprising result of this comparison is that the proposed Rössler model only using chaos without pheromone, is better than Models 1 and 2. Finally, the

combination of chaos and pheromone gives not only the best result but also the smallest standard deviation.

Stat	Model 1	Model 2	Model 3	Model 4	Model 5
Mean	0.9579	0.9841	0.5966	0.9875	0.9917
Std Dev	0.0171	0.0059	0.0759	0.0018	0.0015

Table 4: Coverage after 5000 iterations.

b. Slope a of the 500 first iterations.

This value evaluates the initialization phase of our system: the transition from the initial group of UAVs to a swarm. Here again Model 5 is better than Model 4, the latter being better that the three other models, as can be seen from the results in Table 5. The combination of the Rössler system with pheromone also permits an efficient coverage at the beginning of the simulation.

Stat	Model 1	Model 2	Model 3	Model 4	Model 5
Mean	5.094e-04	6.775e-04	1.787e-04	7.344e-04	8.000e-04
Std Dev	3.752e-05	3.335e-05	1.960e-05	2.283e-05	2.176e-05

Table 5: Slope of the first 500 iterations.

c. Mean value of the recent coverage.

Regarding the recent coverage metrics, the 100 first iterations are not taken into account because it is the maximal time necessary to evaporate pheromones dropped in an area. Also, the maximum value of this metrics is equal to an optimal solution where, during 100 iterations, the UAVs never deposit pheromone in the same area. As a consequence the optimal recent coverage rate is $(10 \times 100)/(100 \times 100) = 0.1$.

Stat	Model 1	Model 2	Model 3	Model 4	Model 5
Mean	0.07673	0.09047	0.05100	0.08574	0.09221
Std Dev	3.904e-04	2.907e-04	7.238e-04	2.657e-04	2.260e-04

Table 6: Mean value of the recent coverage.

For this metrics, we found that Model 4 is less efficient than the model using pheromones, as presented in Table 6. This underlines the need of using the ACO algorithm to be efficient for the coverage problem. These values also highlight that the combination of pheromones and chaos (Model 5) is more efficient than Model 2.

d. Fairness slope a of the 6500 last iterations.

This metrics is used as a complement to our coverage metrics: we use the last 6500 iterations to remove the initial steps. The obtained values are given in Table 7. In that case, we obtain that the Model 5 is the best model with the lower slope. Indeed, if the standard deviation grows slower during a simulation, it indicates that cells are homogeneously visited during the whole simulation.

e. Mean value of connectivity.

This last metrics helps us to evaluate how UAVs are globally organised over the area. Here again Model 5 gives the best result in terms of means but not for standard deviation, as illustrated in Table 8. A larger value means that

Stat	Model 1	Model 2	Model 3	Model 4	Model 5
Mean	4.000e-04	3.081e-04	7.809e-04	3.513e-04	3.034e-04
Std Dev	5.032e-05	4.134e-05	1.471e-04	2.618e-05	1.409e-05

Table 7: Fairness of the coverage.

the network is more disconnected which also indicates that UAVs are well distributed over the area.

Stat	Model 1	Model 2	Model 3	Model 4	Model 5
Mean	5.916	6.044	4.543	6.142	6.179
Std Dev	0.203	0.156	0.538	0.101	0.122

Table 8: Mean value of connectivity.

5. CONCLUSION AND PERSPECTIVES

In this article we have tackled the problem of area coverage by a swarm UAVs in a military context. To this end, we have proposed the Chaotic Ant Colony Optimization to Coverage (CACOC) algorithm that combines an Ant Colony Optimization approach (ACO) with a chaotic dynamical system, to control the mobility of the UAVs. We have empirically demonstrated that CACOC improves the system behaviour of the Kuiper and Nadjm-Tehrani model when there is no pheromone to guide the UAVs. This has been achieved by replacing the random process with a deterministic chaotic process based on a Rössler system. Moreover, the deterministic properties of such chaotic systems permit to monitor the swarm behavior from the Ground Control Station while the UAVs movements remain unpredictable to any observer. CACOC has proven to efficiently tackle the coverage problem and satisfy its constraints.

To conclude on the impact of chaotic dynamics on our system, we insist on the richness of ODE system compared to chaotic maps as Lozi, Ikeda etc. For a given ODE system, several chaotic mechanisms can be found by modifying one parameter of the system. As a consequence, a parameter of an ODE system can be considered as a parameter of our model we can tune to optimize the solution. For future work, we plan to compare the performance of our algorithm varying a parameter of the Rössler system with the corresponding partition of the return maps.

To conclude, the keystone of this paper is that we replace transition probabilities with a chaotic return map detailing the possible transitions. In our future work, we not only propose to use a chaotic process to increase the performance of our system, but also to find the best chaotic mechanism (i.e. return maps shape) that will make our system even more efficient. CACOC is an original approach that combines assets of offline and online methods. The mobility is decided online in a deterministic and reproducible way but the trajectories of the UAVs remains unpredictable. The main advantage of CACOC is that the position of the UAVs are known on the ground without need of communication between the ground and the UAVs.

Acknowledgement

The authors acknowledge the support of the ASIMUT project A-1341-RT-GP, which is coordinated by the European

Defence Agency (EDA) and partially funded by 8 contributing Members (Austria, France, Germany, Italy, Luxembourg, The Netherlands, Poland and Sweden) in the framework of the Joint Investment Programme on Innovative Concepts and Emerging Technologies 2. The ASIMUT project consortium is composed of Thales, Fraunhofer IOSB, Fly-n-Sense, University of Bordeaux and University of Luxembourg.

The experiments presented in this paper were carried out using the HPC facilities of the University of Luxembourg [17] (see http://hpc.uni.lu).

6. REFERENCES

[1] M. Ahmadi and H. Mojallali. Chaotic invasive weed optimization algorithm with application to parameter estimation of chaotic systems. *Chaos, Solitons & Fractals*, 45(9-10):1108–1120, sep 2012.

[2] D. Arroyo, G. Alvarez, and V. Fernandez. On the inadequacy of the logistic map for cryptographic applications. *ArXiv e-prints*, May 2008.

[3] G. Danoy, M. R. Brust, and P. Bouvry. Connectivity stability in autonomous multi-level UAV swarms for wide area monitoring. In *Proceedings of the 5th ACM Symposium on Development and Analysis of Intelligent Vehicular Networks and Applications - DIVANet'15*. Association for Computing Machinery (ACM), 2015.

[4] M. Dorigo. Optimization, learning and natural algorithms. *Ph. D. Thesis, Politecnico di Milano, Italy*, 1992.

[5] E. Galceran and M. Carreras. A survey on coverage path planning for robotics. *Robotics and Autonomous Systems*, 61(12):1258–1276, dec 2013.

[6] A. Gandomi, X.-S. Yang, S. Talatahari, and A. Alavi. Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, 18(1):89–98, jan 2013.

[7] W. Gong and S. Wang. Chaos ant colony optimization and application. In *2009 Fourth International Conference on Internet Computing for Science and Engineering*. Institute of Electrical & Electronics Engineers (IEEE), dec 2009.

[8] T. Iba and K. Shimonishi. The origin of diversity: Thinking with chaotic walk. *Unifying Themes in Complex Systems*, 8:447–461, 2011.

[9] W. H. Kruskal and W. A. Wallis. Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47(260):583–621, dec 1952.

[10] E. Kuiper and S. Nadjm-Tehrani. Mobility models for UAV group reconnaissance applications. In *2006 International Conference on Wireless and Mobile Communications (ICWMC'06)*. Institute of Electrical & Electronics Engineers (IEEE), jul 2006.

[11] H. Li, S. Wang, and M. Ji. An improved chaotic ant colony algorithm. In *Advances in Neural Networks – ISNN 2012*, pages 633–640. Springer Science & Business Media, 2012.

[12] R. Lozi. Emergence of randomness from chaos. *International Journal of Bifurcation and Chaos*, 22(02):1250021, feb 2012.

[13] M. Rosalie. Templates and subtemplates of Rössler attractors from a bifurcation diagram. *Journal of Physics A: Mathematical and Theoretical*, 49(31):315101, jun 2016.

[14] M. Rosalie, G. Danoy, P. Bouvry, and S. Chaumette. UAV multilevel swarms for situation management. In *Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, DroNet '16, pages 49–52, New York, NY, USA, 2016. ACM.

[15] O. E. Rössler. An equation for continuous chaos. *Physics Letters A*, 57(5):397–398, 1976.

[16] J. Schleich, A. Panchapakesan, G. Danoy, and P. Bouvry. UAV fleet area coverage with network connectivity constraint. In *Proceedings of the 11th ACM international symposium on Mobility management and wireless access - MobiWac'13*. Association for Computing Machinery (ACM), 2013.

[17] S. Varrette, P. Bouvry, H. Cartiaux, and F. Georgatos. Management of an academic hpc cluster: The ul experience. In *Proc. of the 2014 Intl. Conf. on High Performance Computing & Simulation (HPCS 2014)*, pages 959–967, Bologna, Italy, July 2014. IEEE.

[18] X. Zang, S. Iqbal, Y. Zhu, X. Liu, and J. Zhao. Applications of chaotic dynamics in robotics. *International Journal of Advanced Robotic Systems*, page 1, 2016.

[19] D. Zhang, Y. Xian, J. Li, G. Lei, and Y. Chang. UAV path planning based on chaos ant colony algorithm. In *2015 International Conference on Computer Science and Mechanical Automation (CSMA)*. Institute of Electrical & Electronics Engineers (IEEE), oct 2015.

[20] Z. Zhou, H. Duan, P. Li, and B. Di. Chaotic differential evolution approach for 3d trajectory planning of unmanned aerial vehicle. In *2013 10th IEEE International Conference on Control and Automation (ICCA)*. Institute of Electrical & Electronics Engineers (IEEE), jun 2013.

APPENDIX

A. EXPERIMENTAL PROBABILITIES OF MODEL 4 (RÖSSLER SYSTEM)

For the reader interested in the probabilistic method instead of the deterministic one with chaos, we provide here the transition probabilities experimentally obtained from Model 4 using the Rössler system. We remind the reader that this model obtains better results than the probabilistic method (Model 1) [10]. Even if it is not our main goal, we are able to provide a better random model with a better coverage rate (Tab. 9). In our future work, we will compare this model to Model 4 to highlight differences between the probabilistic approach and the chaotic based one.

Experimental (from Model 4)	Probability of action		
Last action	Left	Ahead	Right
Left	0.0	1.0	0.0
Ahead	0.26	0.13	0.61
Right	0.60	0.40	0.0

Table 9: Experimental probabilities resulting from the partition of the Rössler first return map.