

Reverse-engineering of Artificially Evolved Controllers for Swarms of Robots

Sabine Hauert, Jean-Christophe Zufferey, Dario Floreano

Abstract— It is generally challenging to design decentralized controllers for swarms of robots because there is often no obvious relation between the individual robot behaviors and the final behavior of the swarm. As a solution, we use artificial evolution to automatically discover neural controllers for swarming robots. Artificial evolution has the potential to find simple and efficient strategies which might otherwise have been overlooked by a human designer. However, evolved controllers are often unadapted when used in scenarios that differ even slightly from those encountered during the evolutionary process. By reverse-engineering evolved controllers we aim towards hand-designed controllers which capture the simplicity and efficiency of evolved neural controllers while being easy to optimize for a variety of scenarios.

I. INTRODUCTION

Artificial Evolution has been extensively used for the development of robot controllers due to its capacity to automatically engineer solutions displaying complex abilities using simple and efficient behaviors [1], [2]. Systems of interest generally can not be solved using conventional programming techniques because they are highly non-linear, stochastic or poorly understood [3]. Subsequently, artificial evolution is particularly well suited for the design of controllers for swarms of robots. Indeed, there currently exists no conventional methodology to deterministically design decentralized controllers which are capable of giving rise to a desired emergent swarm behavior. Overcoming this limitation, genetic algorithms and genetic programming have successfully been used to design controllers for swarms of ground [4], [5] and aerial vehicles [6]–[10] in simulation or on-board physical robots in research environments.

However, evolved controllers are often unable to adapt across different scenarios without being re-evolved. This process takes time and is unrealistic for robot swarms which are intended to be used out-of-the-box in critical applications. Instead we propose to reverse-engineer evolved controllers so as to capture the simplicity and efficiency found through evolution in hand-designed robot controllers whose parameters can easily be optimized for various scenarios. In this paper, the entire process for the design of swarm controllers includes 1) the evolution of neural controllers for robots 2) the analysis of the behaviors performed by the robots to achieve swarming 3) the hand-design of robot controllers inspired from these behaviors 4) the optimization of robot controllers for various environments.

Laboratory of Intelligent Systems, Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland sabine.hauert@epfl.ch

Our endeavor is motivated by an application, whereby a swarm of fixed-wing micro air vehicles (MAVs) must cooperate to serve as a wireless relay for rescuers and victims in disaster areas. In a typical scenario, rescuers rapidly arrive at a disaster site and set up a base which is equipped with a wireless device of limited range. MAVs are launched from this base by hand, one after the other, and must then self-organize to find other rescuers or victims in the environment while maintaining a communication link to the base. Once a user is found, the communication can be established between the base and the user via the MAVs which serve as relay nodes. The MAVs must maintain this communication link active until it is no longer needed.

Flying robots are interesting for such applications because they can easily deploy and spread out over difficult terrain such as flooded areas or debris while providing line-of-sight communication. Our MAVs are not equipped with sensors such as Global Positioning Systems (GPS), cameras, laser range finders or radars because we aim at minimal platforms which are cheap, safe and light-weight. The positioning information derived from these sensors is typically primordial in current research on swarms of flying robots [11]–[23]. Instead of relying on position information, our MAV controllers use only local communication with immediate neighbors and proprioceptive sensors which provide heading, speed and altitude [6], [24].

The following section presents the evolution of controllers for swarms of flying robots and the analysis of the robot behaviors. Sect. III presents the hand-designed controllers inspired from evolution and the optimization of their parameters for various scenarios. In Sect. IV, we test the controller optimization on different scenarios and choose one to show the performance and behavior of the hand-designed swarm in a specific network deployment application. Finally, in Sect. V we summarize the work done so far and discuss future developments.

II. EVOLVED CONTROLLER

As presented in [6] we use artificial evolution as a means of automatically designing neural controllers for a swarm of simulated fixed-wing MAVs. We hereby briefly summarize this work, namely the scenario used to evolve the controllers, the robot neural architecture and the behaviors displayed by the individual MAVs.

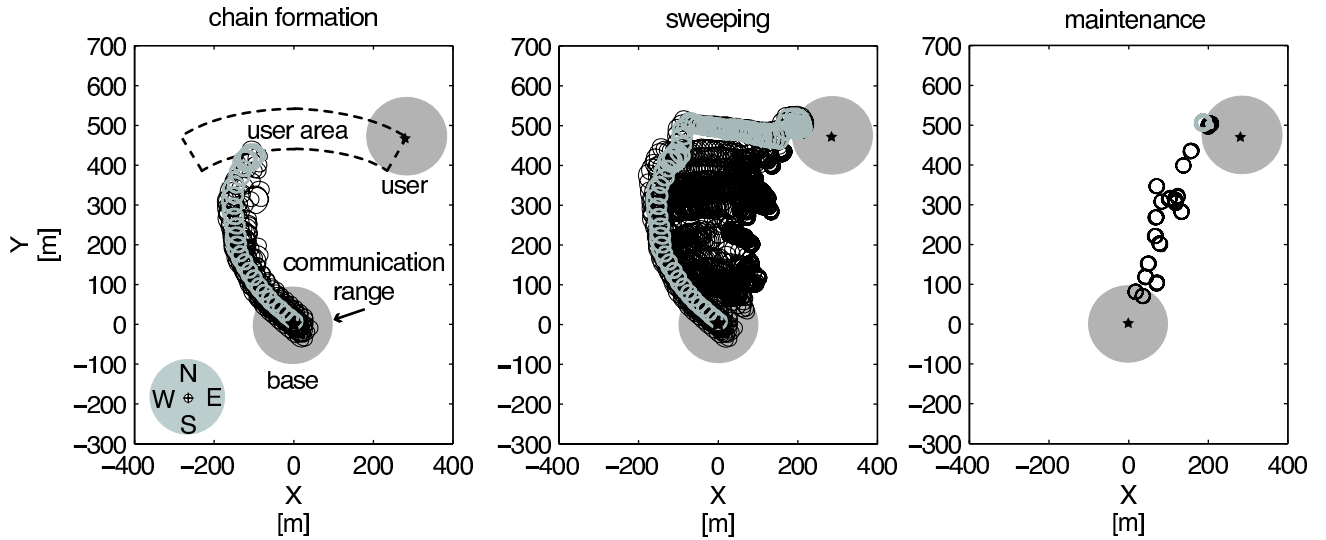


Fig. 1. Trajectories of all the MAVs with the best evolved controllers during a 30 min trial. In this trial, the user has been placed to the North-East with respect to the base in order to show the full extent of the sweeping behavior of the swarm. The trajectory of the first launched MAV is shown by a light grey line.

A. Scenario

The aim is to deploy and maintain an ad-hoc wireless network composed of swarming MAVs to enable communication between a base (BS) from which 20 MAVs are launched every 15 ± 7.5 s and a user (US) positioned in the environment within a $\pm 30^\circ$ angle from North and within 550 m of the base. The deployment and maintenance of the network lasts 30 min and the communication range of the MAVs, base and user is approximately 100 m. These values are based on our acquired experience so far with real platforms designed for our network of swarming MAVs [25].

B. Controller

Each MAV is equipped with identical controllers (homogenous swarm) to allow for scalable swarms composed of interchangeable agents. In our system, MAVs maintain a constant speed and altitude while modulating their turn rate. Constant speed is justified given that fixed wing platforms must always maintain forward motion so as not to stall, as opposed to aircrafts capable of hovering or ground robots which can stop moving when needed. A neural controller with three inputs and four hidden neurons is used to output the turn rate the MAVs should adopt. A low-level flight controller is assumed to make sure the turn rate is achieved within a certain time. Since no positioning information is available, the inputs to the neural controller are exclusively derived from an absolute heading sensor and the messages received from neighboring MAVs every 50 ms. Inputs to the neural controller are defined as follows:

- The heading of the MAV as computed using a magnetic compass.
- The minimum number of network hops that separate the base from the MAV (*BSHopCount*). MAVs which are disconnected from the base are assigned a *BSHopCount* of N where, N is the maximum number

of hops which can be obtained in a given network. For our application, N corresponds to the number of MAVs in the swarm ($N=20$).

- The minimum number of network hops that separate the user from the MAV. When disconnected from the user, MAVs are assigned a value of N .

A genetic algorithm is used to evolve the weights of the neural controllers based on a fitness which aims at maximizing the quality of the connections between the base and user over time. Colony level selection is used to favor inter-agent cooperation [26].

C. Behavior analysis

An example showing the behavior of the best evolved controllers can be seen in Fig. 1. The strategy adopted by the swarm consists of forming a tight chain of MAVs which grows as long as additional MAVs are launched. Once all MAVs have been launched, the MAV chain shifts along the communication range of the base, sweeping the area from West to East until the user is found. The communication link between the base and the user is maintained by having all MAVs turn on the spot with the smallest possible radius given the dynamics of the aircraft.

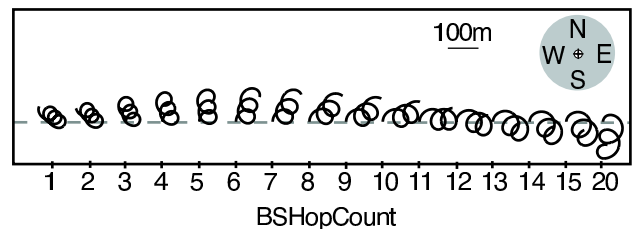


Fig. 2. Effect of the number of hops which separate the base from an MAV (*BSHopCount*) on its trajectory. Here the trajectories of the best evolved MAV controllers are plotted over 30 s. MAVs were disconnected from the user during these experiments.

Through a qualitative analysis of the best evolved controllers, the following simple behaviors were extracted to account for the chain formation, sweeping and communication maintenance performed by the swarm [6]:

1) MAVs continuously turn in the same direction, modulating their turn rate based on their heading measurements and their desired advancement direction and speed.

2) MAVs which are connected to the base ($BSHopCount < 20$) proceed following the trajectories shown in Fig. 2. The typical number of hops separating an MAV from the base is generally lower than 10, meaning that the MAVs move to the North, away from the base.

3) MAVs which are disconnected from the base ($BSHopCount = 20$) backtrack in the direction of the base to tentatively reconnect (Fig. 2).

4) MAVs which are connected to the base turn with a different average turn rate than when disconnected (Fig. 2).

5) MAVs which are connected to both the base and the user turn on the spot following the smallest possible turn radius given by the dynamics of the platform (Fig. 1, right).

The first behavior allows the MAVs to simply modify the global advancement speed and direction of their trajectories by slightly modulating their turn rate with respect to their current heading measurements. The second and third behaviors allow for the formation of an MAV chain with agents moving away from the base as long as they are connected to it and backtracking when disconnected. During the chain formation phase, MAVs are sequentially launched to further extend the MAV chain. Once all MAVs are launched, the swarm periodically disconnects and reconnects to the base, alternating behaviors 2 and 3. Given that the turn rates are different when connected and disconnected (behavior 4), the MAVs are able to sweep the area from one side to the other. Because of the slight difference in trajectory depending on the hop information of the MAVs (Fig. 2), the sweeping is radial and agents closer to the base sweep slightly slower than agents further away. Once the user is found, behavior 5 ensures a minimal maintenance of the communication link between the base and the user by having all MAVs turn on the spot.

These five simple behaviors are responsible for the deployment and maintenance of a swarming MAV network between a base and a user.

III. HAND DESIGNED CONTROLLER

Our interest lies in taking inspiration from behaviors found through evolution to design novel swarm controllers which capture their simplicity and efficiency. We do not

however exactly copy the evolved solution, in particular, our proposed hand-designed controller contains a simplification which makes it suitable for applications whereby a chain of MAVs is able to sweep in parallel (i.e. not radially like in the evolved solution). We then show that this controller can be rapidly adapted to a variety of different scenarios.

A. Controller

We hereby show how the five behaviors discovered through evolution (Sect. II) can be implemented as parameterized controllers.

1) Similar to the evolved controllers, MAVs with speed v can perform trajectories which follow a global direction ψ by adopting different turn rates depending on their heading measurements ψ_{MAV} . As shown in Fig. 3 this is achieved by having an MAV perform a small turn radius r_1 when its heading forms an obtuse angle with ψ and a large turn radius r_2 when the angle is acute. Intuitively, if r_1 and r_2 are identical, the MAV will perform circular trajectories and remain on the spot. Finally, for a trajectory described by ψ , r_1 and r_2 , the turn rate $\omega_{r_1, r_2, \psi}(\psi_{MAV})$ of an MAV is set as follows:

$$\omega_{r_1, r_2, \psi}(\psi_{MAV}) = \begin{cases} -\frac{v}{r_1} & \text{if } obtuse(\psi_{MAV}, \psi) \\ -\frac{v}{r_2} & \text{if } acute(\psi_{MAV}, \psi) \end{cases}$$

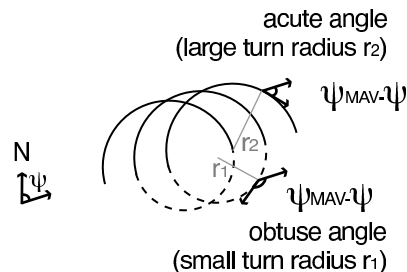


Fig. 3. Example of an MAV trajectory formed by the alternation of circular trajectories of radius r_1 and r_2 based on the heading of the MAV ψ_{MAV} and the global advancement direction of the trajectory ψ . When the angle between the MAV's heading and the direction ψ is obtuse, the MAV performs a circle of radius r_1 while an acute angle yields a circle of radius r_2 .

2) MAVs which are connected (even indirectly) to the base adopt turn rate $\omega_{r_1, r_2, \psi_{US}}(\psi_{MAV})$ where ψ_{US} reflects the approximate direction in which a user might be found. This parameter is given by the scenario specifications (Fig. 4 A).

3) MAVs which are disconnected from the base adopt turn rate $\omega_{r_3, r_4, \psi_{BS}}(\psi_{MAV})$ where ψ_{BS} is the direction opposite to ψ_{US} which will allow the MAVs to advance in the direction of the base (Fig. 4 B).

4) Given the difference between an MAV's average turn radius when connected (r_1, r_2) or disconnected from the base (r_3, r_4), the MAV chain will translate along the

communication range of the base (sweeping) (Fig. 4 C).

5) MAVs which are connected to both the base and the user adopt turn rate $\omega_{r_{min}, r_{min}, null}$ so as to turn on the spot following the smallest possible turn radius r_{min} . This allows the swarm to maintain a communication link between the base and the user (Fig. 4 D).

As a summary, each MAV will adopt a turn rate $\omega(\psi_{MAV}, BS, US)$ based on information concerning its connection to the base ($BS = 1$ if the MAV is connected to the base and 0 otherwise) or the user ($US = 1$ if the MAV is connected to the user and 0 otherwise) and its heading ψ_{MAV} following equations

$$\omega(\psi_{MAV}, BS, US) = \begin{cases} \omega_{r_{min}, r_{min}, null}(\psi_{MAV}) & \text{if } BS \wedge US \\ \omega_{r_1, r_2, \psi_{US}}(\psi_{MAV}) & \text{if } BS \wedge !US \\ \omega_{r_3, r_4, \psi_{BS}}(\psi_{MAV}) & \text{otherwise} \end{cases}$$

An example of possible trajectories obtained by an MAV controller with parameters chosen as $r_{min} = 10$, $r_1 = 10$, $r_2 = 30$, $r_3 = 20$, $r_4 = 30$, $\psi_{US} = North$ and $\psi_{BS} = South$ can be seen in Fig. 4. When using this controller, the swarm can perform behaviors similar to those discovered through evolution to deploy and maintain a wireless network, namely chain formation, sweeping and communication maintenance.

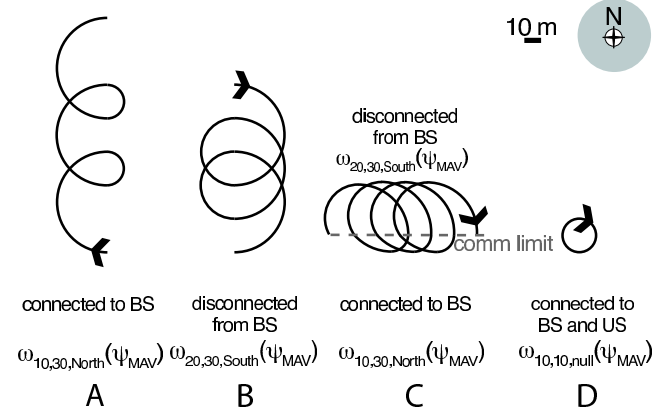


Fig. 4. MAV trajectories depending their heading and connectivity to the base (BS) and user (US).

B. Scenario

We aim at showing that our controller can quickly be adapted to a number of different scenarios. In particular, we look at a scenario where rescuers, when arriving by road, deposit several wireless beacons along the way at a small enough interval for the beacons to be directly or indirectly interconnected. Here, beacons with a communication range of 100 m are dropped from a rescue vehicle, every 50 m, along a straight road which extends from West to East. The result is an enlarged base as can be seen in (Fig. 5). A rescuer will then sequentially launch 20 MAVs every

15 ± 7.5 s by simply throwing them into the air from the West-most beacon. The swarm must establish a communication link between the base and a single user positioned to the North of the base and within a $d \times w$ area. Once the user is found, the communication relay must be maintained until the end of the mission duration which is of 30 min.

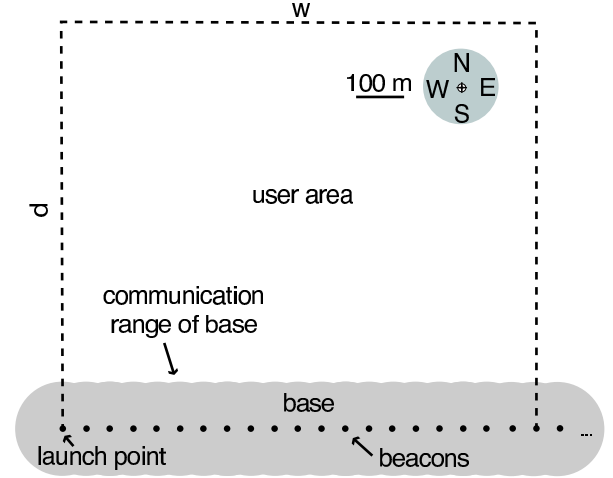


Fig. 5. The swarm composed of 20 MAVs must be able to find any user positioned to the North of the base and within a $d \times w$ area. The beacons forming the base have a communication range of 100 m and are positioned every 50 m along a straight road. MAVs are launched from the West-most beacon.

C. Parameter optimization

For real world applications, it is interesting to be able to rapidly optimize the parameters of a robot controller given the requirements of a scenario. For the scenario described in Sect. III-B we look to optimize the parameters r_1 , r_2 , r_3 and r_4 of the robot controllers (Sect. III-A) for a desired area coverage $d \times w$. The remainder of the parameters are set by the scenario specifications ($\psi_{US} = North$, $\psi_{BS} = South$ and r_{min} is defined by the limitations of the platform).

Thanks to the simple geometry of the MAV trajectories, we are able to design a model which can be used to predict the area coverage of the swarm given by the distance d_{pred} reached by an MAV chain and the translation w_{pred} of this chain sweeping along the communication range of the base (Fig. 5). The values d_{pred} and w_{pred} are computed using Eq. 1 and Eq. 2 in the Appendix.

To optimize the parameters, we compute the summed square error $e = (d - d_{pred})^2 + (w - w_{pred})^2$ for each combination of r_1 , r_2 , r_3 , r_4 in the range of natural numbers from r_{min} to r_{max} with the constraint that $r_1 < r_2$ and $r_3 < r_4$. The combination with the smallest error e is selected as the optimized parameter set. To ensure that the turn radius of the MAV remains small with respect to the communication range r_{comm} we define the maximum turn radius r_{max} as equal to a quarter of r_{comm} .

IV. RESULTS

Experiments are run in a realistic event-based simulator which implements 802.11b communication models, physics-based wave propagation and a first order model of an MAV platform which flies at 10 m/s, has a minimum turn radius of 10 m and is affected by sensor and actuator noise as described in [24].

A. Parameter Optimization

Using the approach presented in Sect. III-C we are able to determine the parameters r_1 , r_2 , r_3 and r_4 of the robot controllers described in Sect. III-A for a desired area coverage $d \times w$. We test our approach on five different coverages with the corresponding parameters listed in Table I. For each desired area coverage, 100 deployments are done with no user present in the environment to allow the swarm to reach out as far as possible. Fig. 6 shows the distances d_{sim} and w_{sim} reached in simulation for each desired area coverage. As can be seen, the optimized robot controllers are successful since the simulated swarms are able to achieve the desired area coverages. For the remainder of this paper we will consider the robot controller optimized for an area coverage of 500 m x 500 m.

TABLE I
OPTIMIZED PARAMETERS FOR VARYING AREA COVERAGES $d \times w$

$d \times w$ [m x m]	r_1 [m]	r_2 [m]	r_3 [m]	r_4 [m]
250 x 500	11	13	12	15
500 x 750	14	22	23	24
500 x 500	16	25	23	25
750 x 500	10	21	10	25
500 x 250	14	22	16	22

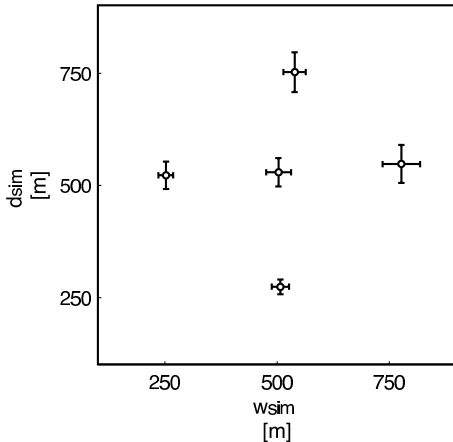


Fig. 6. Chain length d_{sim} and sweep translation w_{sim} along the communication range of the base (Fig. 5) reached by the swarm over 100 trials for desired area coverages of 250 m x 500 m, 500 m x 500 m, 750 m x 500 m, 500 m x 250 m and 500 m x 750 m. No users are positioned in the environment to determine the full extent of the swarm coverage. For each of the five desired area coverage, we plot the mean coverage obtained in simulation with a point and the standard deviations as bars extending from this point.

B. Behavior

The trajectories of all the MAVs during a 30 min trial can be seen in Fig. 7. As in the evolved system, the MAVs are able to form a tight chain which reaches out from their initial launching site. The MAVs in the chain are then able to translate along the communication range of the base, sweeping the area from West to East. Once an initial connection between the base and the user in the environment is created, it is maintained by having all MAVs turn following the smallest possible turn radius.

C. Performance

All the users positioned within the desired search area of 500 m x 500 m are found by the swarm. Once a connection between the base and a user is established, the probability of receiving a data packet, sent every second from the base, at the user end is given in Fig. 8. Results show that the median probability is of 81%, this is sufficient to achieve usable communication networks. The probability is not maximal because the MAVs generally navigate at the edge of the communication range of the base where they are subject to noise and disconnections. Finally, although intermittent, the communication links are maintained in 100% of the cases to the end of the trial durations.

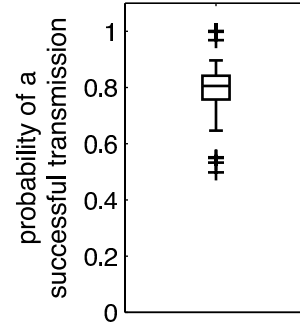


Fig. 8. Probability of receiving a data packet, sent every second from the base, at the user end, when tested over 100 trials with users randomly positioned in a 500 m x 500 m as shown in Fig. 5. Data packets are only sent after the swarm has created a first connection from the base to the user. The box has lines at the lower quartile, median, and upper quartile values. The whiskers extend to the farthest data points that are within 1.5 times the interquartile range. + symbols denote outliers.

V. CONCLUSION

Artificial evolution has proven to be a powerful mechanism for the development of simple and yet efficient controllers for robot swarms [4]–[10]. In order to extend the applicability of these controllers to a wider range of scenarios than those used during evolutionary experiments, it is possible to reverse-engineer the simple evolved strategies. Reverse-engineered controllers have the advantage of being easy to parameterize for various scenarios.

Our aimed application envisions the deployment of a swarm of fixed wing robots to serve as communication relay between a base from which the robots are launched

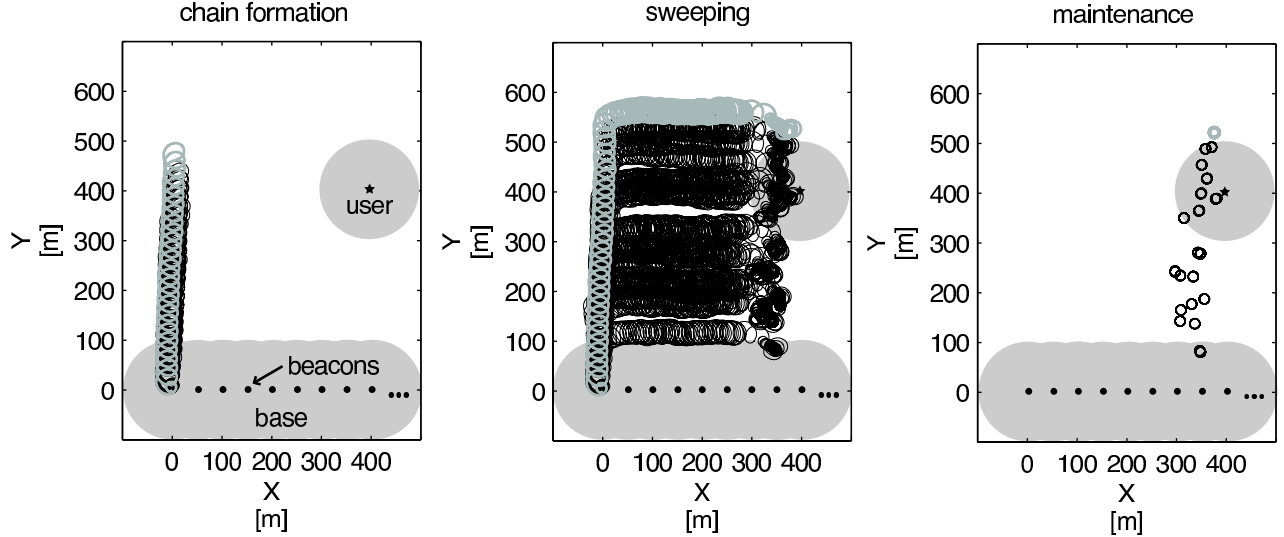


Fig. 7. Trajectories of all the MAVs during a 30 min trial. The user is located in (400, 400). The trajectory of the first launched MAV is shown by a light grey line. Notice the chain formation, sweeping and maintenance of the communication link between the base and the user.

and a single user present on the ground. Artificial evolution was used to automatically design MAV controllers, resulting in swarm strategies to create chains which can sweep over a given area while maintaining a connection with the base. After the swarm is able to establish a communication link between the base and the user in the environment, it is efficiently maintained throughout the duration of the experiment [6]. We then reverse-engineer these three strategies, namely chain formation sweeping and maintenance. The resulting MAV controllers are very simple and the value of their parameters can be rapidly optimized given a desired area coverage.

In the future, we aim at taking advantage of our understanding of reverse-engineered robot controllers to provide the basis for a safeness and liveness analysis of the swarm system [27], [28]. Furthermore, we are investigating the effect of wind and its mitigation on the flying robots as described in [24]. Finally, current developments are aimed towards the implementation of the robot controllers presented in this paper on actual MAV platforms.

APPENDIX

A. Area coverage model

We aim at predicting the distance d_{pred} reached by an MAV chain and the translation w_{pred} of the swarm sweeping along the communication range of the base in Fig. 5.

During chain formation, n MAVs with a communication range r_{comm} are launched on average every t_{launch} seconds and advance at speed v m/s. As long as MAVs are connected to the base, they continue to advance as shown in Fig. 4 A. Once disconnected, they backtrack as shown in Fig. 4

B. As a result, the maximum distance reached by the chain is equal to nr_{comm} . When MAVs are launched before the chain has had time to advance sufficiently to disconnect from the base, the overall distance d reached by the MAVs is dependent on the average advancement speed of an MAV $v' = \frac{2v(r_2-r_1)}{\pi(r_1+r_2)}$ when performing $\omega_{r_1, r_2, \psi_{US}}(\psi_{MAV})$ (Sec. III). Assuming $r_1 < r_2$, MAVs turn clockwise and the chain is given sufficient time to entirely deploy, the maximum distance d_{pred} reached by the chain is given by

$$d_{pred} = \min(nr_{comm}, (n-1)t_{launch}v' + r_{comm}) \quad (1)$$

As shown in Fig. 4 C, MAV chains sweep by alternating between turn radii r_1, r_2, r_3 and r_4 based on their heading and connection status to the base. It typically takes t_{dis} seconds for an MAV to realize that it is disconnected from the base (time-out or update time). Because the MAVs move together, we consider the advancement speed v'' of the sweeping chain as being identical to that of a single MAV. The geometry of the trajectories gives us the maximum translation w_{pred} along the communication range of the base, reached during a deployment of duration t_{trial} seconds:

$$w_{pred} = (t_{trial} - \frac{r_{comm}}{v'} - nt_{launch})v'' \quad (2)$$

where

$$\begin{aligned}\gamma &= \frac{vt_{dis}}{r_2} \\ y_{dis} &= r_2 \sin(\gamma) \\ \alpha &= \cos^{-1}\left(\frac{2r_1r_4 - r_1r_3 - r_2r_4 - y_{dis}r_1 + y_{dis}r_4}{r_1r_3 - r_2r_4}\right) \\ \beta &= \cos^{-1}\left(\frac{2r_2r_3 - r_1r_3 - r_2r_4 + y_{dis}r_2 - y_{dis}r_3}{r_1r_3 - r_2r_4}\right) \\ s &= r_1(\pi - \beta) + r_2\alpha + r_3(\pi - \alpha) + r_4\beta \\ t &= \frac{s}{v} \\ s' &= -r_1 \sin(\pi - \beta) - r_2 \sin(\alpha) \\ &\quad + r_3 \sin(\pi - \alpha) + r_4 \sin(\beta) \\ v'' &= \frac{s'}{t}\end{aligned}$$

ACKNOWLEDGEMENTS

The authors are grateful to Severin Leven who is currently developing the platform and has provided his insight on all related aspects. This work is supported by armasuisse, competence sector Science + Technology for the Swiss Federal Department of Defense, Civil Protection and Sports.

REFERENCES

- [1] D. Floreano and J. Urzelai, "Evolutionary robotics: the next generation," in *Evolutionary robotics III*, T. Gomi, Ed. Ontario, Canada: AAI Books, 2000.
- [2] K.-J. Kim and S.-B. Cho, "A comprehensive overview of the applications of artificial life," *Artificial Life*, vol. 12, no. 1, pp. 153–182, 2006.
- [3] P. J. Fleming and P. R.C., "Evolutionary algorithms in control systems engineering: a survey," *Control Engineering Practice*, vol. 10, no. 11, pp. 1223–1241, 2002.
- [4] D. Floreano, S. Mitri, S. Magnenat, and L. Keller, "Evolutionary conditions for the emergence of communication in robots," *Current biology*, vol. 17, pp. 514–519, 2007.
- [5] M. Dorigo, V. Trianni, E. Sahin, R. Groß, T. H. Labella, G. Baldassarre, S. Nolfi, J.-L. Deneubourg, F. Mondada, D. Floreano, and L. M. Gambardella, "Evolving self-organizing behaviors for a swarm-bot," *Autonomous Robots*, vol. 17, no. 2-3, pp. 223–245, 2004.
- [6] S. Hauert, J. Zufferey, and D. Floreano, "Evolved swarming without positioning information: an application in aerial communication relay," *Autonomous Robots*, vol. 26, no. 1, pp. 21–32, 2009.
- [7] P. Gaudiano, E. Bonabeau, and B. Shargel, "Evolving behaviors for a swarm of unmanned air vehicles," in *Proceedings of the IEEE Swarm Intelligence Symposium*. Piscataway, NJ: IEEE Press, 2005, pp. 317–324.
- [8] J. Soto and K.-C. Lin, "Using genetic algorithms to evolve the control rules of a swarm of UAVs," in *Proceedings of the IEEE International Symposium on Collaborative Technologies and Systems*. Piscataway, NJ: IEEE Press, 2005, pp. 359–365.
- [9] M. D. Richards, D. Whitley, and J. R. Beveridge, "Evolving cooperative strategies for UAV teams," in *Proceedings of the Genetic And Evolutionary Computation Conference*, vol. 2. New York: ACM Press, 2005, pp. 1721–1728.
- [10] A. S. Wu, A. C. Schultz, and A. Agah, "Evolving control for distributed micro air vehicles," in *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation*. Piscataway, NJ: IEEE Press, 1999, pp. 174–179.
- [11] J. Elston and E. W. Frew, "Hierarchical distributed control for search and tracking by heterogeneous aerial robot networks," in *Proceedings of the IEEE International Conference on Robotics and Automation*. Piscataway, NJ: to appear: IEEE Press, 2008.
- [12] M. Flint, M. Polycarpou, and E. Fernández-Gaucherand, "Cooperative control for multiple autonomous UAVs searching for targets," in *Proceedings of the 41st IEEE Conference on Decision and Control*, vol. 3. Piscataway, NJ: IEEE Press, 2002, pp. 2823–2828.
- [13] D. J. Pack and G. W. P. York, "Developing a control architecture for multiple unmanned aerial vehicles to search and localize RF time-varying mobile targets: part I," in *Proceedings of the IEEE International Conference on Robotics and Automation*. Piscataway, NJ: IEEE Press, 2005, pp. 3954–3959.
- [14] H. V. D. Parunak, S. A. Brueckner, and J. Sauter, "Digital pheromones for coordination of unmanned vehicles," in *Environments for Multi-Agent Systems*, ser. Lecture Notes in Computer Science. Berlin: Springer, 2005, vol. 3374, pp. 246–263.
- [15] A. Ryan, J. Tisdale, M. Godwin, D. Coatta, D. Nguyen, S. Spry, R. Sengupta, and J. K. Hedrick, "Decentralized control of unmanned aerial vehicle collaborative sensing missions," in *Proceedings of the American Control Conference*. Piscataway, NJ: IEEE Press, 2007, pp. 4672–4677.
- [16] J. A. Sauter, R. Matthews, H. V. D. Parunak, and S. A. Brueckner, "Performance of digital pheromones for swarming vehicle control," in *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multi-Agent Systems*. New York: ACM Press, 2005, pp. 903–910.
- [17] P. Vincent and I. Rubin, "A framework and analysis for cooperative search using UAV swarms," in *Proceedings of the ACM Symposium on Applied computing*. New York: ACM Press, 2004, pp. 79–86.
- [18] Y. Yang, A. A. Minai, and M. M. Polycarpou, "Evidential map-building approaches for multi-UAV cooperative search," in *Proceedings of the IEEE American Control Conference*. Piscataway, NJ: IEEE Press, 2005, pp. 116–121.
- [19] L. Merino, F. Caballero, J. R. Martínez-de Dios, J. Ferruz, and A. Ollero, "A cooperative perception system for multiple UAVs: application to automatic detection of forest fires," *Journal of Field Robotics*, vol. 23, pp. 165–184, 2006.
- [20] O. Holland, J. Woods, R. De Nardi, and A. Clark, "Beyond swarm intelligence: the UltraSwarm," in *Proceedings of the IEEE Swarm Intelligence Symposium*. Piscataway, NJ: IEEE Press, 2005, pp. 217–224.
- [21] P. Basu, J. Redi, and V. Shurbanov, "Coordinated flocking of UAVs for improved connectivity of mobile ground nodes," in *Proceedings of the IEEE Military Communications Conference*, vol. 3. Piscataway, NJ: IEEE Press, 2004, pp. 1628–1634.
- [22] B. A. Kadrovach and G. B. Lamont, "Design and analysis of swarm-based sensor systems," in *Proceedings of the 44th IEEE Midwest Symposium on Circuits and Systems*, vol. 1. Piscataway, NJ: IEEE Press, 2001, pp. 487–490.
- [23] E. Kuiper and S. Nadjm-Tehrani, "Mobility models for UAV group reconnaissance applications," in *Proceedings of the IEEE International Conference on Wireless and Mobile Communications*. Piscataway, NJ: IEEE Press, 2006.
- [24] S. Hauert, L. Winkler, J. Zufferey, and D. Floreano, "Ant-based swarming with positionless micro air vehicles for communication relay," *Swarm Intelligence: Special Issue on Swarm Robotics*, vol. 2, no. 2-4, pp. 167–188, 2008.
- [25] S. Leven, J.-C. Zufferey, and D. Floreano, "A simple and robust fixed-wing platform for outdoor flying robot experiments," in *Proceedings of the International Symposium on Flying Insects and Robots*, 2007, pp. 69–70.
- [26] M. Waibel, L. Keller, and D. Floreano, "Genetic team composition and level of selection in the evolution of multi-agent systems," *IEEE Transactions on Evolutionary Computation*, 2009, to appear.
- [27] A. F. T. Winfield, J. Sa, M.-C. Fernández-Gago, C. Dixon, and M. Fisher, "On formal specification of emergent behaviours in swarm robotic systems," *International Journal of Advanced Robotic Systems*, vol. 2, no. 4, pp. 363–370, 2005.
- [28] A. F. T. Winfield, C. J. Harper, and J. Nembrini, "Towards dependable swarms and a new discipline of swarm engineering," in *Swarm robotics*, ser. Lecture Notes in Computer Science. Berlin: Springer, 2005, vol. 4433, pp. 126–142.