Towards a Multi-Function Swarm That Adapts to User Preferences

Sondre Engebråten[1,2], Oleg Yakimenko[3], Jonas Moen[1,2], and Kyrre Glette[1,2]

***Abstract*— This paper describes the current state of our work towards a multi-function swarm of UAVs that adapts to user preferences. To achieve this, we employ software simulation and an evolutionary search method for offline generation of a repertoire of controller parameters, covering a wide range of swarm behaviors. By allowing the user to select the appropriate controller parametrization online, the swarm system can adapt to human preferences at run time. To validate our approach, we have performed real-world validation of a few select controllers, on a small fleet of modified commercial off-the-shelf UAVs. The results so far are encouraging, and we plan to extend the range of swarm behaviors and increase the swarm size in the future.**

I. Introduction

Swarms have the potential to greatly simplify and speed up the solving of a number of tasks simply by utilizing multiple agents to solve problems cooperatively. Example areas of application could be mapping and network relay in dangerous or disaster areas, illustrated by Fig. 1. Traditional challenges in swarm research, and in particular, swarm engineering, have been how to make controllers that scale and provide the desired high level behavior based on low level or individual agent actions [1].

While there have been some works using evolutionary methods to design low-level UAV controllers, most often these use simulations [2], as working with a real-world aerial swarm can pose several practical challenges. In our work, we target a real-world implementation of the swarm, which puts severe limits on the type of sensor information that can reliably be obtained and used. In addition, the implementation of a real-world swarm is of paramount importance in order to visualize and demonstrate the increased capabilities and value a swarm system can have.

In this paper, we give an overview of our current results towards our goal of a multi-function aerial swarm, derived from our work in [3]–[5]. We use software simulations and evolutionary methods for offline generation of a repertoire of controllers corresponding to different high-level swarm behaviors. We also describe how we extend a commercial-of-the-shelf (COTS) drone platform with networking capabilities to enable real-world testing of the swarm behaviors, and report from initial field testing.

The use of evolutionary repertoire generation is to our knowledge new in a swarm context, and the application of evolved rules to real-world flying drones has so far been little explored.

II. Methods / System

A. Applications

For the initial experiments with multi-function swarm systems, we consider two applications to examine further: exploration and networking of agents. Exploring an area is a very basic behavior that might be useful for a number of more specific application, such as search and rescue, perimeter surveillance and 3D reconstruction using images (photogrametry). Networking is essential in order to be able to communicate between agents, for our initial experiments the agents operate in a limited area and can expect to be in communication range at all times, however in the future a large swarm may easily extend over an area that is far greater than the range of a single hop communication link. In addition, the swarm could be considered a network backbone provider, or in other words a way of providing network infrastructure where there may be none or where the existing infrastructure is unavailable.

Fig. 1. Example of our UAV swarm operating environment – exploration and network coverage in an urban area.

B. Controller

A simple swarm controller can be made by weighting of sensor data, in the form of input forces. The idea behind this approach is to allow for each application to contribute some information to the controller, which are then weighted or scaled, in order to form the controller output. This allows the controller to consider multiple applications at once, which is essential for our multi-function swarm.

The currently implemented controller receives 4 inputs; the direction and distance to the closest neighbor, second closest neighbor, third closest neighbor and the direction to the least-visited neighboring field (square). To find the least visited square surrounding the agent, a histogram over visits to each area is collected. This is based on a Moore neighborhood model, i.e least of the eight surrounding squares.

Each parametric controller requires 16 parameters. There are 4 parameters associated with each input. The 4 parameters for each input describe a general attraction and repulsion and the strength, range and center distance affecting the distance held to other agents. Holding a distance to another

[1]Norwegian Defence Research Establishment, P.O. Box 25, 2027 Kjeller, Norway Sondre.Engebraten@ffi.no

[2]University of Oslo, P.O. Box 1080, Blindern, 0316 Oslo, Norway

[3]Naval Postgraduate School, 699 Dyer Rd., Monterey, CA 93943, USA

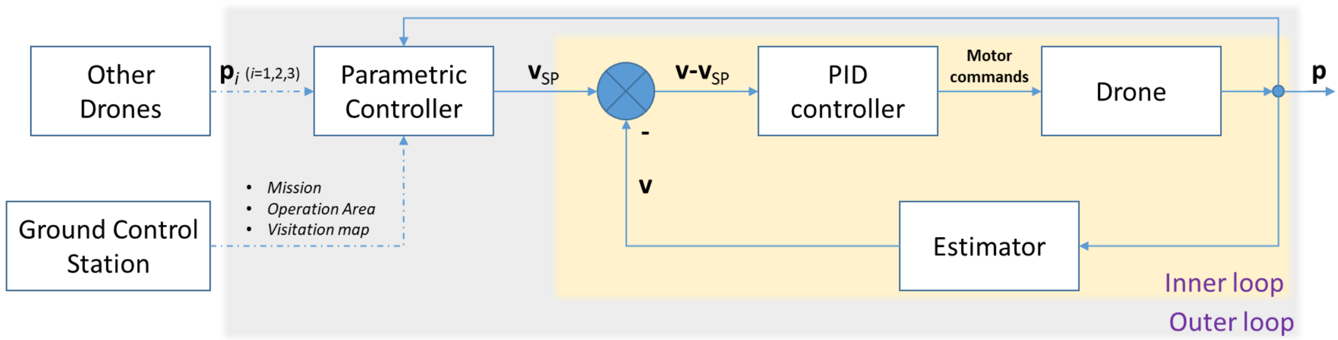

Fig. 2. The implemented hardware setup uses two control loops, an inner classical loop featuring a PID controller and an outer loop which employs our new parametric control structure. Each drone receive telemetry from other nearby drones in the swarm allowing the parametric controller to use this information as input to avoid collisions and optimize behavior performance.

agent is important for our networking application. We defined a new function–the Sigmoid-well function–which incorporates these parameters and allows for a distance dependent weighting of sensor inputs. An example of the Sigmoid-well function can be seen in Fig. 3 where the strength of the force depends on the distance to the sensed object.

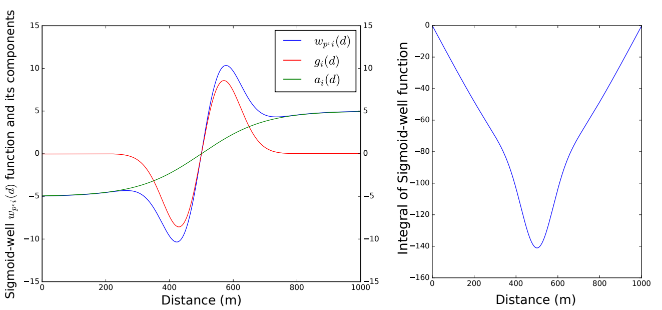

Fig. 3. The Sigmoid-well function is shown in the left part of the figure. The green line represents the Sigmoid component, which gives a persistent attractive or repulsive contribution to the Sigmoid-well function. The red line is the gravity well component, which is responsible for holding a distance to another agent. Added together they form the blue line: the Sigmoid-well function. The right part of the figure depicts the integral of the Sigmoid-well function, which has a clear strong attraction (minimum) around a center at 500.0. The shape of this function can be adapted parametrically.

The final output from our controller is a velocity setpoint to the UAV. For these experiment the velocity setpoint is a 2D vector in a local coordinate frame. The output is calculated by accumulating the contribution from each of the input forces after weighting them using the Sigmoid-well function. A more complete discription of the controller structure can be found in previous works [3]

C. Simulator setup and Evolutionary optimization

In simulation, each agent is a modeled as a simple point mass with limits on velocity and acceleration. While simplistic, this allows to make approximate models for a wide range of platforms including non-holonomic platforms through limiting velocity and acceleration. A simple model is suitable as we are most interesting in examining the high level behaviors of the swarm and the potential interactions between agents that occur. The weakness of this approach is obviously the lack of fidelity, which can pose a challenge when transferring controllers to a real-world swarm.

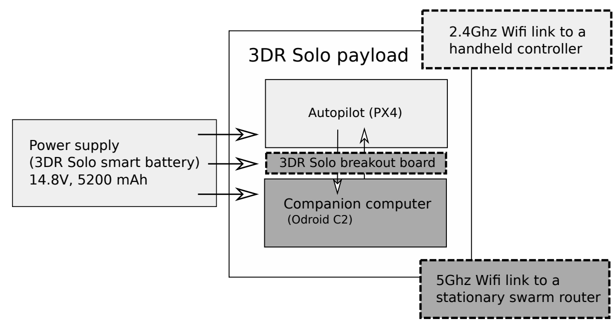

Fig. 4. Enhancement of the 3DR Solo COTS UAV, enabling decentralized swarm operations. Each 3DR Solo was extended with an additional onboard computer, this enables each swarm agent to independently make decisions on run time without relying on a single centralized control structure. The individual controllers communicate over 5Ghz wifi, while the remote used for manual remote piloting of the drone operates in 2.4Ghz.

The controllers are defined by 4 vectors of length 4, which gives the weighting of each input force. This is a structure that is highly suitable for optimization using evolutionary algorithms. We chose MAP-elites [6] for this task, as it makes it possible to evolve a repertoire of solutions in a predefined and discretized space of multiple behavior characteristics. The result from a single run of the algorithm is a grid containing a wide range of controller behaviors. The controllers are characterized according to two behavior characteristics: the degree of exploration at the swarm level, the network coverage of the swarm. Further, there is a performance metric related to energy usage, which describes the quality of the solution in each grid cell.

D. Hardware platform

In order to validate our control approach, we implement a swarm using 3DR Solo COTS drones. In order to enable agent to agent communication, we extend the 3DR Solo with an onboard computer that communicates over a shared swarm network. This allows the agents to exchange telemetry information, and enables reactive collision avoidance between agents. A brief outline of the controller integration can be seen in Fig. 4.

The Solo drone is based on the open source flight controller PX4 and uses a modified version of the Ardupilot open source flight controller firmware. Previous work flying multiple UAVs do so with a laptop or other centralized computer as the main hub for communication and control. In these experiments however we fully decentralize the swarm control structure. Initial experiments also employed mesh networking, however to reduce interference the swarm network had to be moved to 5Ghz, which unfortunately does not support mesh networking. As such, even the swarm system described here, has one centralized weak point; the router required for agent to agent communication. In the future, it should be possible replace the swarm network with an implementation that supports true mesh/ad-hoc networking. The full details on the integration with the 3DR Solo to enable swarm networking can be found in [5].

III. Experiments

A. *Evolutionary results from simulation*

In [3], we used simulation in combination with the MAP-elites algorithm to generate a behavior repertoire (Fig. 6). This repertoire contains a wide variety of controllers and associated swarm behaviors, as defined the behavior characteristics, and shows that it is possible to evolve behaviors for a multi-function swarm using our proposed controller structure. Some examples, both hand coded and evolved, can be seen in Fig. 5.

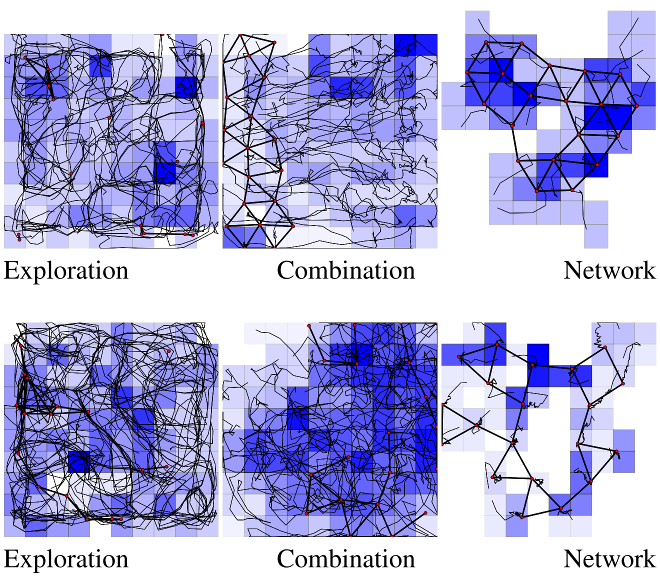

Fig. 5. Example hand designed (top) and automatically designed (bottom) controllers, both using the same controller structure. From left to right are examples of exploration focus (left), combination of exploration and network creation (middle) and a controller that results in a static network (right). All these plots are the results of simulation a swarm. Videos of these behaviors can be found at `https://www.youtube.com/playlist?list=PL18bqX3rX5tQN2HKdHSCna8ysbX9lUeSM`

B. *Real-world testing*

Preliminary results on real-world UAVs, reported in [4], show the promise of the proposed control method. Fig. 7 shows an example of our multi-function controller running on the real-world UAVs. These results are preliminary and do not yet fully address the issue of transferring a controller from simulation to real UAVs. In particular, the real UAVs appear to have a slower response compared to the simulated UAVs. As such, the parameters of the controllers had to be reduced in order to safely use the simulated controller on the real UAVs. In the future, this may be solved by updating the simulated UAV model, to make the two more similar.

Fig. 8 shows the separation distance between individual UAVs during a flight, both for an exploration-focused controller as well as a network-focused one. In the exploration experiment, the controller had no incentive to maintain any fixed distance, as such the distance between agents vary greatly. Considering instead the controller where the focus was on maintaining distance between agents in order to facilitate a communication network, the distances between agents are much more stable once the agents have reached an equilibrium. Our real-world experiments also included a test with a controller that is tailored to a combination of the two applications. This controller exhibits a distance that varies less than the exploration controller but more than the network focused controller, as expected. This shows that the behaviors can be adjusted to either application or a combination of the two.

C. *Discussion*

Results so far on repertoire generation are promising. We are able to generate a wide range of behaviors through evolution, based only on the controller structure and the defined characteristics. The evolutionary optimization method used, MAP-elites, is greedy. This is problematic when dealing with noisy evaluations, and we were forced to take measures to reduce the variance in the fitness evaluation.

While the controller structure used in these experiment are enough to demonstrate the concept of a multi-function swarm, it may be necessary to augment the number of inputs and potentially the type of inputs in order to satisfy more complex applications. One simple extension could be to include fixed inputs, in the form of a simulated senses location, which would allow for the specification of areas of interest within the current framework.

For the real world experiment we have found that with current COTS technology, the weak link of the swarm is communication. There are two potential approaches to mitigating this weakness; to improve on the network used for the swarm or to reduce the reliance on the swarm network.

Finally, while the 3DR Solo COTS drones used in this experiments allow for rapid prototyping of a UAV swarm system, it is also clear that flying numerous drones at once was not a design requirement in the design process of this drone. In particular, the requirement for having one controller per drone scales poorly when operating a swarm of UAVs.

IV. Conclusion and future work

In this paper we have presented the current state of our approach to a multi-function swarm, summarizing the work in [3]–[5]. On the software side we have proposed a parametric controller automatically generated a repertoire of low-level controllers which correspond to different degrees of high-level swarm behavior. To deploy the swarm controller on a fleet of real-world drones we have extended a COTS platform with networking capabilities, and performed initial field tests of the automatically designed controllers.

Further work on the system including studying the effects of increasing the swarm size and investigating the

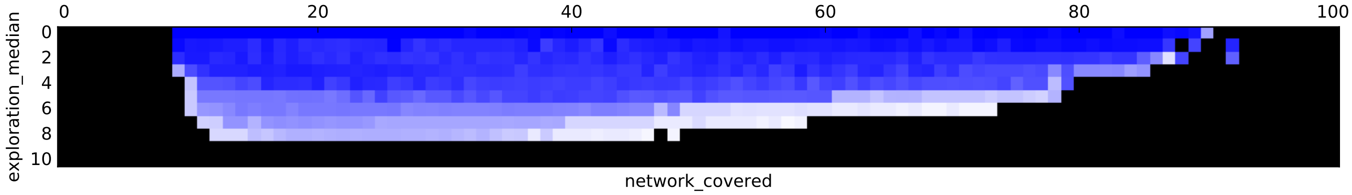

Fig. 6. The automatically generated repertoire of controllers after epoch 200 using the MAP-elites evolutionary method. Behaviors range from low to high exploration and low to high network coverage. Black regions indicate that no controller was found for that exploration-network combination. Controller performance is color coded from white to dark blue, ranging from lowest to highest energy efficiency. Each cell in the repertoire represents one potential controller. All cells that are filled are valid controller structure, but due to the difference between simulated and real UAVs not all controllers may transfer equally well. This is a topic for future research.

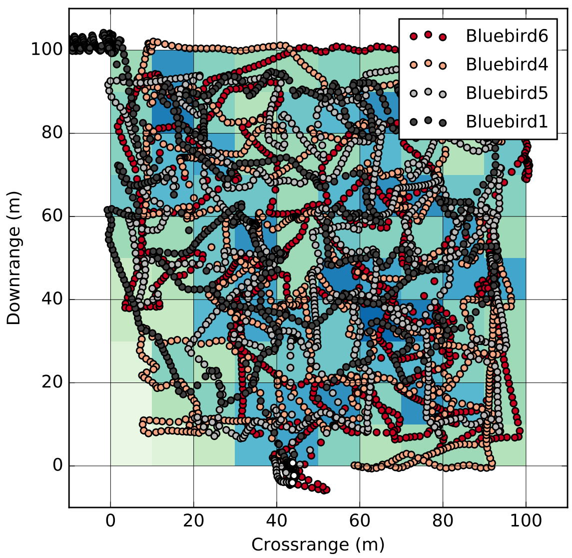

Fig. 7. Bird's-eye view of UAV paths for the exploration experiment. While stochastic, the paths taken by the UAVs qualitatively resemble the results seen in simulation by visual inspection. Experiments show that while there is some differences between simulated UAVs and real UAVs the general behaviors are the same. Experiments also show that a number of UAVs are able to operate in a single horizontal plane without collisions due to a reactive collision avoidance implementation.

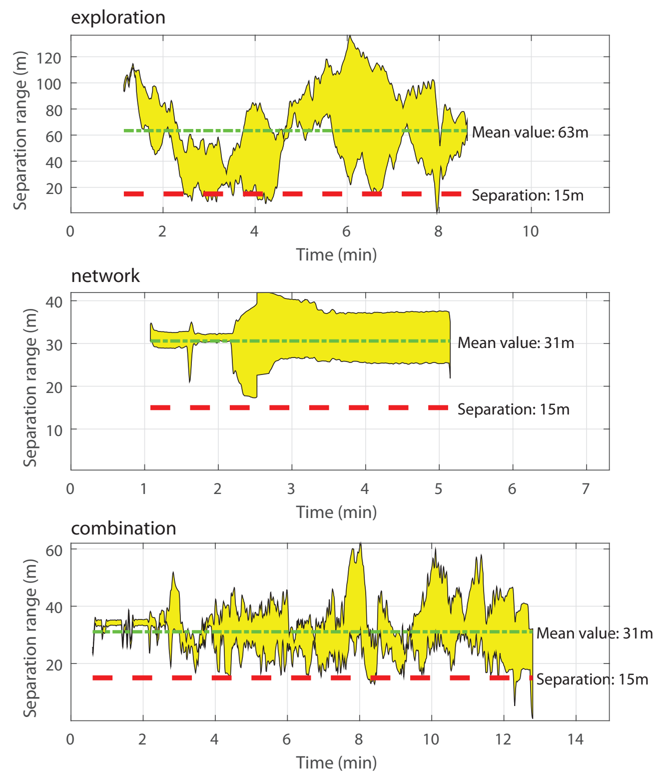

Fig. 8. Drone separation distance time histories, as seen from Drone 1 for experiments with the exploration (top), network (middle) and combined (bottom) controllers. These show how the three different controller tested have different behaviors which would result in different performance in the applications described. The separation line is a desired minimum separation required and is specified indirectly through the parameters of the controller. Specifically the exploration controller does not maintain any fixed distance, while the networking controller limits movement as to maintain a static lattice structure and finally the combination controller tries to achieve both. The combination controller varies more than the networking controller but still is able to maintain a minimum separation while exploring the area.

dynamics when switching between behaviors. We also want to extend the functionalities of the swarm by introducing new behaviors such as source seeking. We believe it can be very useful to have a repertoire of controllers, to select appropriate behaviors in complex scenarios, however this will require exploration of efficient ways for humans to control the swarm.

Acknowledgments

The authors would like to acknowledge the Norwegian Defense Research Establishment, the Naval Postgraduate School's Naval Research Program and Consortium for Robotics and Unmanned Systems Education and Research along with the Fulbright Scholar Program for support of this research.

References

[1] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[2] S. Hauert, J.-C. 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.

[3] S. A. Engebråten, J. Moen, O. Yakimenko, and K. Glette, "Evolving a repertoire of controllers for a multi-function swarm," in *Applications of Evolutionary Computation - 21st International Conference, EvoApplications 2018*, ser. Lecture Notes in Computer Science. Springer, 2018, vol. 10784, pp. 734–749.

[4] S. Engebråten, K. Glette, and O. Yakimenko, "Field-testing of high-level decentralized controllers for a multi-function drone swarm," in *Control & Automation (ICCA), 11th IEEE International Conference on*. IEEE, 2014, to appear.

[5] ——, "Networking-enabling enhancement for a swarm of COTS drones," in *Control & Automation (ICCA), 14th IEEE International Conference on*. IEEE, 2014, to appear.

[6] J.-B. Mouret and J. Clune, "Illuminating search spaces by mapping elites," *arXiv preprint arXiv:1504.04909*, 2015.