Field-Testing of High-Level Decentralized Controllers for a Multi-Function Drone Swarm

Sondre Engebråten[12], Kyrre Glette [12] and Oleg Yakimenko [3]

***Abstract*— This paper describes the results of initial field testing of a high-level decentralized controller for a swarm of multi-rotor drones. This controller allows the intuitive implementation of a wide variety of swarm behaviors. The parameters of this controller were mapped to two possible missions. The proposed controller structure is parameterized to enable the evolution of swarm behaviors in future work. In this paper, a set of hand coded controllers, previously evaluated in simulator were field tested. The simulator used considers each UAV as a simple point-mass model of each agent. As such, the ability to transfer the controller structure from simulated UAVs to real UAVs is investigated. The experiments use a fleet of enhanced networking-capable commercial-of-the-shelf drones that were especially developed to support and demonstrate the proposed high-level controller in the decentralized swarm environment. The paper describes hardware and controller implementation and then proves feasibility and great potential of the proposed approach in a series of flight tests conducted by a swarm of four identical drones.**

I. Introduction

Swarms are not a new concept. In natural biology, swarms have existed for millions of years. In academic terms some of the early work on mimicking biological swarms in computer simulation is over 30 years old [1]. Yet, there is a distinct lack of research presenting the implementation, use and evolution of real-world robotics swarm in realistic environments.

A swarm of unmanned aerial vehicles (UAVs) is, in essence, nothing more than a group of UAVs collaborating to solve a specific task. Collaboration can be achieved through explicit communication, deliberation and interaction (which is common in multi-agent systems), or implicitly, due to more organic interaction between UAVs or agents in the swarm.

For swarms of unmanned vehicles multiple different control structures have been proposed, including; hybrid structures [2], potential field control [3] and artificial physics [4]. In addition, more general control abstraction for groups of robots allowing for the generation of control rules to adapt swarms to more arbitrary shapes, have been proposed [5]. Yet other works show how it is possible to integrate multiple different control structures into a unified behavior framework [6].

Swarms of UAVs have a potential to greatly simplify a number of tasks, for instance in search-and-rescue, disaster recovery or security scenarios [7]–[10]. Previous works have shown that the use of multiple UAVs may be beneficial in providing network coverage [7], [9]. Implementing a full fledged search and rescue system, including ground control stations and multi-mode behaviors, is a challenging research issue that has been tackled using a partially decentralized control structure [8]. Swarms have been proposed to assess forest environments, including a control method for maintaining connectivity and a formation within the swarm [11]. In monitoring tasks, UAVs have the potential to gather data for wildlife monitoring and research that might otherwise be cost-prohibitive to collect [12]. It is also possible to adapt biologically inspired methods for monitoring [13]. For all of these examples, the improvement in capability using swarm systems can be attributed to tackling the dull, dirty and potentially dangerous tasks that are better performed by machines rather than humans.

Obviously, for swarms to truly achieve their full potential, control must be decentralized. Without decentralized control, the swarm becomes nothing more than a set of distributed actuators, all tied to a single vulnerable centralized controller. However, this does not mean that there can be no interaction with a centralized controller, but each agent should be able to operate independently, at least to some degree, of any centralized control structure. In the terms of a two-loop control architecture it is a matter of minimizing the operators input to the outer-loop controller, and relying on own state feedback data from other agents and results of sensing the operational environment.

While decentralized control is important for swarm as a concept, it is equally important to be able to provide a way for humans or operators to interact with the swarm. This does not have to be direct interaction or commands of individual agents, which are common in todays command and control structures, but can be subtler higher-level requests. This enhances swarm scalability by allowing a single operator to control more agents simultaneously. While a single ground control station may be used to control multiple UAVs [14], it does not fully address the issue of scaling a multi-robot or swarm system. Specifically, to further increase scalability, it might be possible to develop behavioral swarm primitives. Each primitive provides a slightly different swarm behavior, that would allow an operator to adapt a swarm to the desired behavior by choosing an appropriate controller. The latter would greatly lessen the need for human interaction, while incorporating a suitable interface for humans to interact with the swarm.

The challenge of coordinating a UAV swarm to establish a communication network has been tackled [15], but previous work does not show the results on real UAVs. The SMAVNET project shows that establishing a communication network is possible without position information [16]. Communication and networking is vital for UAV systems [17].

[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

There are several commercial-of-the-shelf (COTS) network implementations available, but real-world tests are important to determine the real-world performance of these COTS networks [17]. It is also possible to include radio frequency propagation as a consideration into UAV swarm path planning [18]. UAV swarms may also be used as a platform for electronic warfare, or jamming, through an opportunistic beamforming array [19]. This, however, depends on being able to accurately hold a distance between UAVs, and as such, is similar to the requirements for creating a communication network. These are just a few of the numerous applications for swarms [20].

For this study, an implicitly collaborating swarm is considered. This assumes that all participants in the swarm are actively seeking to participate in the swarm, and assumes that the other agents will act and react in a similar way to themselves. This is different from swarms with competing agents [21].

Using a similar on-board computer as these experiments, with the Robot Operating System (ROS) framework, researchers show that it is possible to track and land on an unmanned surface vehicle using on-board vision and controller algorithms [22]. This requires more processing power than a classical autopilot can provide, and serves as an example of how an additional companion computer [23] can extend the capabilities of the platform.

This paper presents the results of field testing the high-level (outer-loop) controller for a swarm of multi-function multi-rotor UAVs, aiming at executing several typical real-world missions. The structure of this controller has been developed and tested in computer simulations already [24] and this paper pushes it further, implementing it on a fleet of network-capable COTS drones [23] operating in a realistic mission environment (Fig. 1). The paper proceeds with Section 2 briefly describing multi-rotor UAVs developed using COTS components and adapted for swarm operations. It then follows with a detailed characterization of the proposed controller supporting multiple missions within the same cascaded architecture. The results of tests and evaluation of swarm UAV performance is presented in Section 4. Section 5 concludes the paper.

II. Adaptation of COTS Drone for Swarm Operations

Conventional command-and-control architectures send commands to individual drones or agents. This limits the number of agents one operator can easily control. One way of alleviating this issue is to move the control interface to a higher level of interaction. Instead of sending commands to individual agents, the operator may issue high level commands to the swarm as a whole, directing and adapting the swarm system to the application at hand. In other words, this paper views the swarm as a super organism, where commands are issued to the swarm instead of individual agents. Based on these high-level commands state-of-the-art guidance, navigation and control (GNC) algorithms would execute scheduling and determine individual agent behavior. As such, the human operator to be less concerned with the micro management of the swarm and rather focus on the macro level, or the behavior of the swarm as a whole.

Fig. 1. Example swarm UAV operational environment used in this study.

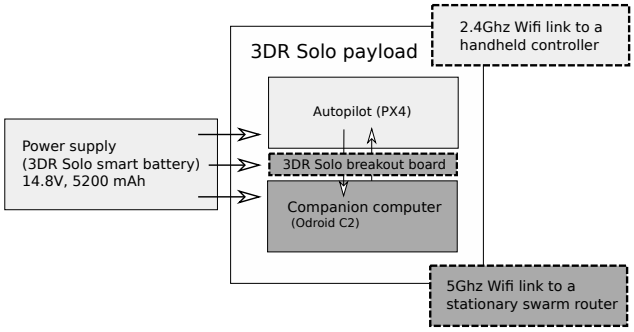


Fig. 2. Enhancement of the 3DR Solo COTS UAV, enabling decentralized swarm operations.

To date, there is no COTS multi-UAV system available for purchase. To enable the swarm system envisioned in this paper, the hardware for this application has to be adapted from existing solutions and products. Towards this goal, the 3DR Solo COTS drone was modified by adding a secondary companion computer: Odroid C2, featuring more processing power and additional extension ports [23]. Adding the secondary companion computer enabled each UAV to be connected to two networks: the standard 2.4GHz controller link to the manual remote controller and a joint 5GHz swarm network (Fig. 2). It is this second joint swarm network that allows the agents to interact and exchange information.

As shown in Fig. 2, the companion computer (high-level controller) is also connected to the drone autopilot (low-level controller). This allows two-way communication, receiving telemetry from autopilot and sending the control commands or setpoints back. Having direct access to the autopilot through a robust direct link allows getting data from both autopilot's sensors and state estimators, while at the same time being able send the GNC commands from the higher-level controller directly to the drone (as opposed to sending them over the network).

Fig. 3. Fleet of networking 3DR Solo drones.

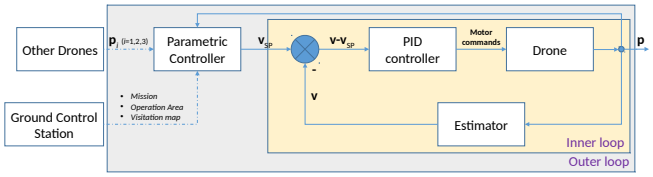


Fig. 4. Two-loop UAV control architecture.

It should be noted that the developed low-cost COTS-drone-based capability is quite unique because very few standard COTS drones offer open-source application programming interface (API) with their autopilots. The few that do offer an API are most commonly offered as more expensive developer UAVs, which is undesirable in a swarm context. By combining additional on-board processing capability and the joint swarm network (in Fig. 2), the developed COTS UAV platform possesses the capability to interact with other UAVs, i.e. communicate and conduct joint missions. All without requiring a centralized control structure. This seems to be one of the first necessary steps towards a true swarm UAV prototype, that has the decentralized control as a key concept. Figure 3 shows a fleet of identical enhanced 3DR Solo drones built with the aforementioned concept in mind, and capable of executing swarm missions.

For the experiments conducted and described in this paper, each of the agents make their own, individual, local decisions at runtime. This is important for a number of reasons, including maintaining scalability, being able to handle loss of link, and minimize the latency for reactions. Further, it is also easy to imagine that with a shared network, it could easily be overloaded as the number of agents increases; especially if all agents only take command from a central controller. In a bandwidth limited scenario, it is preferable to keep sensor data processing as close as possible to the sensor itself. This allows the bandwidth requirement to be minimized by digesting the data before sharing it further.

III. Swarm Missions and Controller Architecture

In this research, each drone uses some behavioral primitives, which an operator can select among, allowing the swarm to adapt to human preferences or needs. It is important, however, to distinguish between the traditional pre-programmed missions, such as path-following, executing search patterns and alike, and more advanced adaptive and reactive behaviors studied here. The proposed behaviors differ from pre-programmed missions in that they are not fully described at initiation time. Instead, the interaction and behavior is determined by a set of rules or a program at run time, depending on agent-to-agent interactions and the environment. This type of behavior can be considered reactive behaviors. While this is more powerful and flexible, it is also much more complicated. Instead of simply specifying a predefined path, this requires a complex internal structure for each agent that is able to process information and react based on perceived sensor data.

The simplest example of a reactive behavior is an implementation of dynamic collision avoidance. While conducting a mission, the agent may recalculate its path if the preset path is blocked by another agent or an obstacle. In order to achieve this, the use of local rules and distributed control is exceedingly important. Without local on-board processing for each agent, the delay in sending a request to a centralized control structure and receiving a response may not be able to assure collision-free flight.

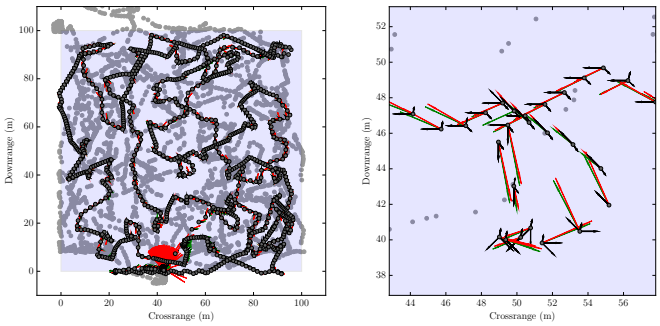
The developed high-level controller described in this section is supposed to support multiple applications, such as area search, and establishing and maintaining a communication network (while assuring reactive collision avoidance). The overall architecture of the two-loop control architecture is shown in Fig. 4, featuring the inner loop as implemented on a standard Ardupilot autopilot and a parametric weighted controller programmed as the outer-loop [24]. In this figure, vector $\mathbf{p} = [x, y]^T$ represents a planar position in the local tangent coordinate frame and $\mathbf{v} = [v_x, v_y]^T$ is the speed vector (for the initial evaluation of the developed algorithms all drones operate in the horizontal plane).

Each robot is controlled by setting a velocity setpoint $\mathbf{v}_{\text{SP}}$ (outer-loop control). The goal of inner-loop controller is then to change acceleration to match the velocity setpoint, i.e. minimize the norm $||\mathbf{v} - \mathbf{v}_{\text{SP}}||$. If needed, the cascaded control architecture allows setting a position setpoint as well, which would then be transferred to a velocity setpoint.

The way the velocity setpoint is set for each agent is defined by the current swarm configuration. Specifically, at every instance of time, each drone receives four candidate inputs used to generate a single controller output. These inputs are

1) Direction and distance to the closest neighbor
2) Direction and distance to the second closest neighbor
3) Direction and distance to the third closest neighbor
4) Direction to the least-visited neighboring field (square)

Figure 5 illustrates this concept graphically. Specifically, Fig. 5a shows a birds-eye view of a joint mission flown by four UAVs. At each instance of time each UAV processes the available information shared via a joint swarm network and defines the four aforementioned directions as for example shown in Fig. 5b. In the latter figure, these four directions are depicted with four smaller arrows. These four directions are then blended into a single command input using the weighting coefficients as discussed next. In Fig. 5b this



(a) (b)

Fig. 5. Illustration of the four inputs during a real flight.

direction is indicated with a green arrow. The red arrow denotes the final velocity setpoint sent to the autopilot; this also includes any necessary corrections made by the underlying collision avoidance algorithm.

To find the least-visited neighboring square the developed algorithm analyses a histogram of visits to each area. In computer simulations, a shared blackboard structure was used, but this seemed needlessly complicated for the real-world experiment. Hence, for the experiments described in this paper, each agent keeps its own histogram of visited areas without sharing this with the other agents, which removes the need for complicated synchronization algorithms to keep the map consistent across the swarm.

The four aforementioned inputs are coded as four two-dimensional (position) difference vectors, $\mathbf{F}_i$, $i = 1, 2, 3, 4$, the vectors defining direction and distance relative to the current agents position. In this case, if one of the inputs is transferred to the controller with no change the agent will move towards one of the three neighboring agents or the least-visited neighboring field. In practice however, each of the four inputs are weighted with a weighting coefficient w_i^p = const, $i = 1, 2, 3, 4$ - so that the resultant desired direction may not coincide with any particular input. This direction serves as a velocity setpoint

$$\mathbf{v}_{\text{SP}} = \frac{1}{4}\sum_{i=1}^{4} w_i^p \frac{\mathbf{F}_i}{||\mathbf{F}_i||} \quad (1)$$

The parametrically weighing coefficients, in Eq.(1) are composed of the two parts, depending on the distance (computed differently for each input) to the sensed object (agent).

$$w_i^p = a_i(d_i) + g_i(d_i) \quad (2)$$

These two parts, $a_i(d_i)$ and $g_i(d_i)$, are responsible for static repulsion/attraction forces, and forces accounting for keeping a predefined distance, respectively

$$a_i(d_i) = k_i * \left(\frac{2}{1 + e^{-(d_i - c_i)/\sigma_i}} - 1 \right) \quad (3)$$

$$g_i(d_i) = -t_i * 2 * (d_i - c_i) * e^{-(d_i - c_i)^2/\sigma_i^2} \quad (4)$$

Specifically, $a_i(d_i)$ contributes a fixed attractive or repulsive force across a greater area enabling pure attraction or repulsion behaviors, which are useful for collision avoidance or exploration. This component is based on a smooth Sigmoid activation function centered around a distance $c_i > 0$ to an object and varying between $-k_i$ and k_i. The second component, $g_i(d_i)$ - the gravity well, enables holding the distance c_i to an object. The appearance of $g_i(d_i)$ function of Eq.(4) is caused by the usage of the normal distribution $N(c_i, \sigma_i)$. Since the desired setpoints are specified for velocity rather than position, $g_i(d_i)$ is a scaled derivative of the normal distribution. Parameters k_i, t_i and σ_i, in Eqs.(3),(4) allow adjusting forces' strength and range. Additionally, the sign of in Eq.(4) allows representing both repulsive and attractive behaviors.

The choice of specific values for the four four-element vectors **k**, **t**, **c** and $\boldsymbol{\sigma}$, defining w_i^p, in Eq.(1), to be tested to the series of field experiments, described in this paper, is based on computer simulations and hand-coded examples discussed in previous works [24]. The full parameter list for four experiments conducted to date is shown in Tab. I.

As seen from Tab. I, for the main experiments three different controllers, or parametrizations of controllers, were tested. One controller (Search) forced the swarm to explore the area by focusing on visiting previously rarely visited areas. Another controller (Network) enforced network maintenance or, in other words, an ability to hold the distance to other UAVs in the swarm. The third controller (Combination) tested the swarm's ability to combine the two aforementioned components to a behavior that awards exploration while staying in a compact formation.

IV. Controller Test and Evaluation

This section describes the results of the field-testing of different swarm behaviors using the detriment sets of controller gains. It starts with describing the setup of experiments and then proceeds with analysis of each experiment as outlined in Tab. I.

A. Test Setup and Scenarios

Tests were conducted in an area replicating a real-world operational environment, as shown in Fig. 1. Each of the drones were equipped with an Odroid board, which connects to a joint swarm network as was presented in Fig. 2. Initially, the drones were positioned out on the ground with an approximately even spacing. The controller software was started manually on each drone, as launch and recovery scheduling is a challenge that was not tackled in this work.

As described in Section 3, the proposed controller for each UAV relies on four inputs to generate one velocity setpoint output. As discussed, three of these inputs are related to a position of other nearby agents. As such, the minimum, number of agents for this controller algorithm is four. For each test, each agent requires telemetry from at least three other agents in order to operate nominally. This means that for all tests, even those testing scaling of the swarm algorithm, only four (out of 20 available) networking UAVs were used. This was done as these initial experiments serve

TABLE I

COEFFICIENTS CHOSEN TO DEMONSTRATE DIFFERENT BEHAVIORS FOR A SWARM OF UAVS.

Experiment	**k** (weights)	**t** (scales)	**c** (centers)	$\boldsymbol{\sigma}$ (spread)
Scaling	[-0.5,0,0,1]	[0,0,0,0]	[15,15,15,100]	[10,10,10,10]
Search	[-0.5,0,0,1]	[0,0,0,0]	[30,30,30,100]	[20,20,20,10]
Network	[-0.5,0.5,0,0.1]	[-0.05,-0.05,-0.05,-0.05]	[30,30,30,100]	[20,20,20,10]
Combination	[-0.5,0.5,0,1]	[-0.05,-0.05,-0.05,-0.05]	[30,30,30,100]	[20,20,20,10]

Fig. 6. Swarm UAVs on a mission during one of the described experiments.

to validate the viability of the controller approach, in future work the swarm will be extended to include more operational UAVs. In the case of less than four drones flying, which is the minimum for the proposed controller, the remaining drones were on the ground essentially feeding the algorithm with dummy data required for the normal operation.

As the launch process is not automated, the time between launches varies. In the experiments, it was found that the most unreliable process was actually the launch of the drones. Quite frequently, one or more drones would have a problem delaying the launch. This included "motion detected" (even though drones were standing still on the ground) or failure to arm (start spinning the propellers). Both seemed to be one of the downsides of a COTS based swarm system. Initial tests, gradually scaled up from 1 to 4 drones in the air at any given time. This was to make sure the computer simulation results [24] could reliably be transferred to the real swarm. Figure 6 shows a snapshot of three UAVs conducting a swarm mission.

B. Scalability tests

For the initial scalability tests the goal was simply to determine if the proposed controller algorithm could scale from 1 to 4 agents and if the controller parameters used in simulations [24] were at all transferable to the real-world. Figure 7 shows the trace of the behavior of 1 through 4 agents in the scalability tests with the controller defined by the gain shown in the second row of Tab. I.

It should be noted that the original set of gains; as derived from computer simulations in [24] - was different from those shown in Tab. I. To be specific, the original gains were larger. However, using the original set of weights led to too aggressive behaviors that endangered swarm operations. Reducing the gains allowed for a more moderate behavior that was still agile, but reduced the risk of losing UAVs.

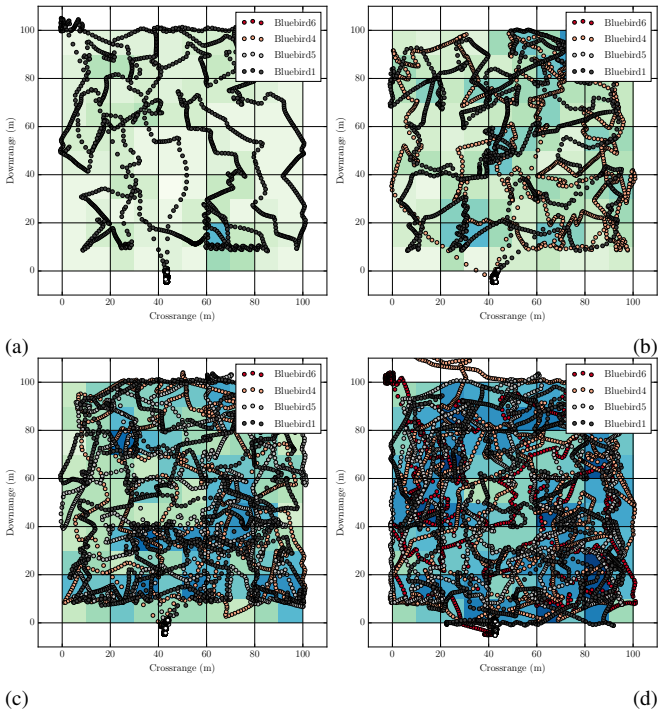


Fig. 7. Scalability test using 1 (a), 2 (b), 3 (c) and 4 (d) drones.

C. Area search

The results of exploring the area search behavior with the gains of Tab. I are presented in Fig. 8. Examining the birds-eye view of the swarm search pattern indicates that it is quite similar to that seen in computer simulation Fig. 9 [24]. While not explicitly seen in the field experiment log, four UAVs explore their surroundings in much the same manner as in simulations.

This particular set of controller parameters has no focus on creating and maintaining a communication network (the scale vector in Tab. I is zero), and as such, does not cluster up or seek out other agents. Each agent is, rather, repulsed by other swarm members, which contributes to spreading the agents out and assisting the search. This is clearly seen in Fig. 10, where the distance to all other agents, as viewed from Drone 1 is presented. As seen, there is no tendency to hold at any distance and the distance varies greatly within 20-140m with the mean value of 63m. This shows that this parametrizations is poor at providing optimal network coverage, as that requires the agents to maintain a persistent

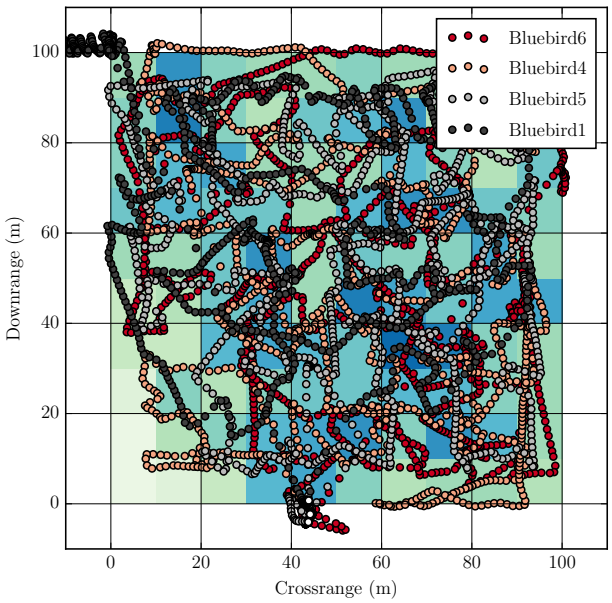


Fig. 8. Birds-eye view of UAV paths for the exploration experiment. While stochastic the paths taken by the UAVs qualitatively resemble the results seen in simulation by visual inspection. Experiments show that while there is some differences between simulated UAVs and real UAVs the general behaviors are the same.

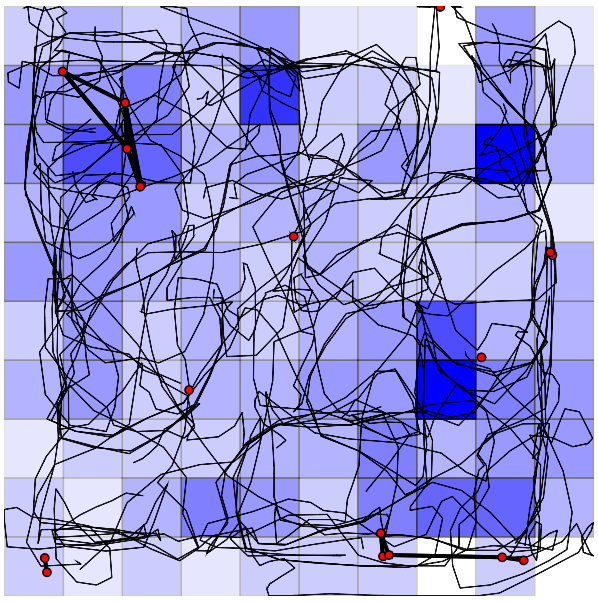


Fig. 9. Simulated trace of UAV paths for a controller focused on exploring an area.

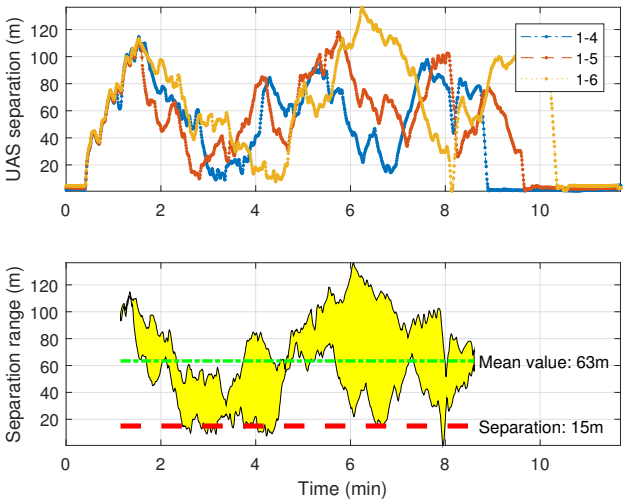


Fig. 10. Drone separation distance time histories, as seen from Drone 1 for the exploration experiment.

distance in order to maximize coverage. The distance drop down to 0m at t=8min was likely caused by a near miss collision.

The color of the 10m by 10m cells within the 100m by 100m operational area in Fig. 8 corresponds to the number of visitations, darker color corresponds to more visitations. As mentioned earlier, in this initial set of field tests of the developed UAV swarm, the visitation map was not shared among all drones, so what is shown in Fig. 8 represents a sum on individual visitations maintained by each drone individually.

Overall, the area search experiment proved that the controller gains proposed in previous works and verified in computer simulations can be transferred to real world experiments. Minor adjustments had to be made to account for slower vehicle response, but overall, the ratios for the different parameters remain similar to those used for the controllers tested in simulator [24]. Generally, all the weights of **k** had to be reduced by a factor of 3-4 in order to account for the slower response of the actual UAVs as apposed to simulated UAVs. The operation area for the real UAVs is also only one tenth of the size of the simulation area. To accommodate this, the center and spread parameters were reduced by a factor of 5.

D. Network Maintenance

As seen from Tab. I, the network maintenance oriented controller includes a non-zero scale parameter. This forces a controller to keep all agents in the swarm at a specific distance from each other. The desired distance itself is defined by the center parameter (30m). As a result (see Fig. 11), during the flight test, all four agents clustered and maintained a relative distance to each other. It should be noted that, compared to the area search experiment, the network maintenance experiment was cut short, and the agents retrieved before their batteries ran out. This was done as the agents quickly converged to and hold a stable stationary formation representing a square pattern. As seen from Fig. 12, the average distance between the drones was kept at the 30m level, with a spread lesser than that of Fig. 10.

E. Combination controller

The combination controller of Tab. I exhibited a combination of the two behaviors: exploration and network maintenance (see Fig. 13). This can be seen from the usage of the weight vector enabling both behaviors and a non-zero scale vector (last row in Tab. I). As seen in Fig. 13, the drones explore the area, much like the purely search-

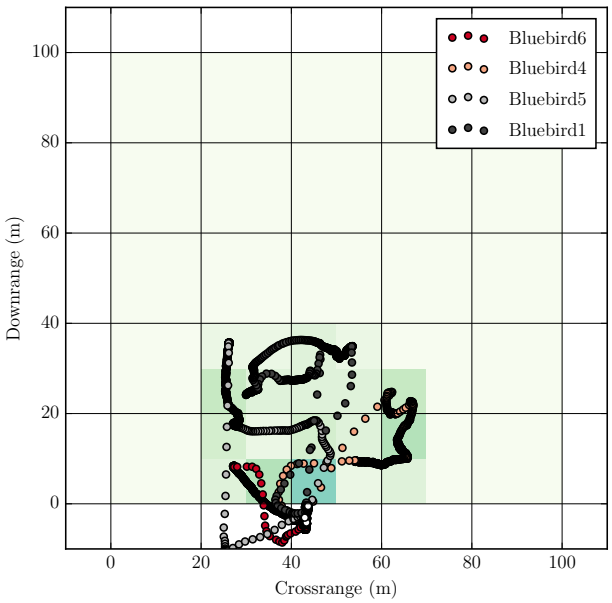


Fig. 11. Birds-eye view of UAV paths for the networking experiment.

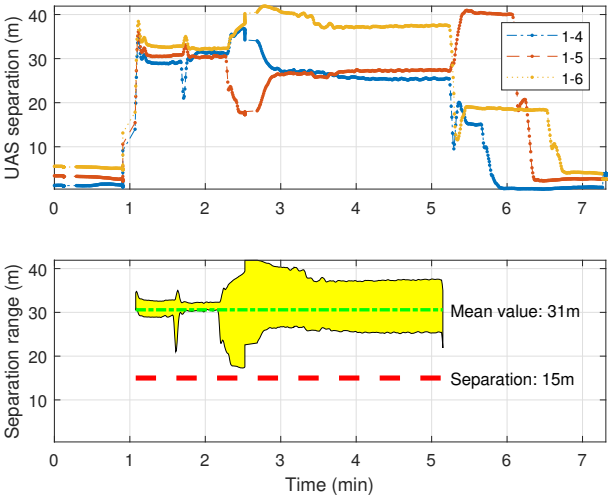


Fig. 12. Drone separation distance time histories, as seen from Drone 1 for the networking experiment.

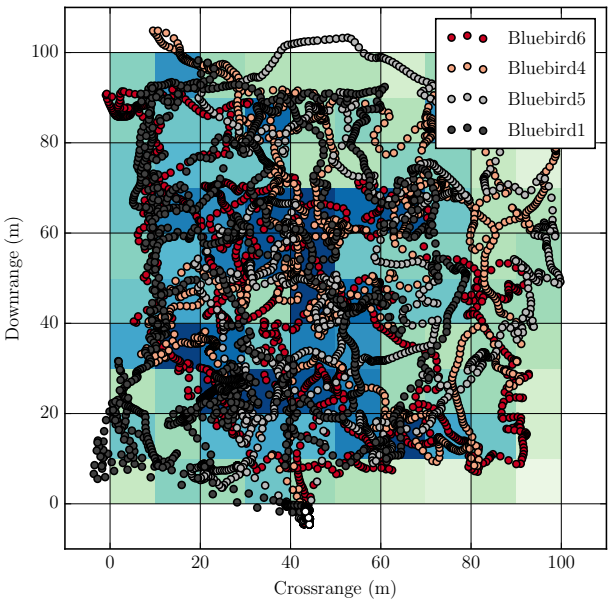


Fig. 13. Birds-eye view of UAV paths for the combination controller experiment.

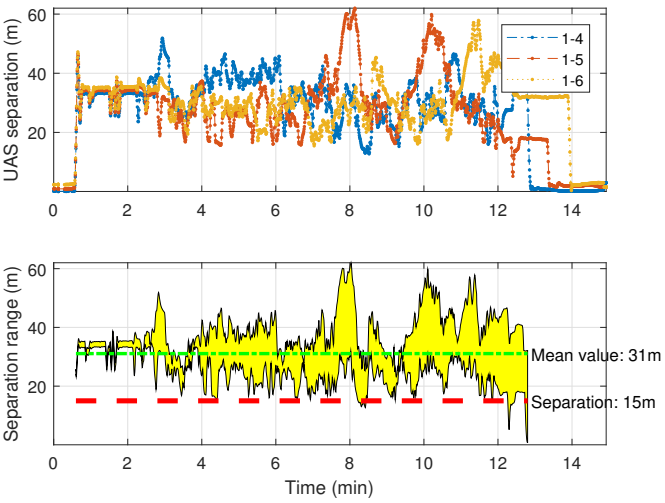


Fig. 14. Drone separation distance time histories, as seen from Drone 1 for the combination controller experiment.

focused controller (cf. Fig. 8). However, Fig. 14 shows that this controller also restricts, or tries to keep the desired distance between the agents, as was the case in Fig. 12 (cf. Fig. 10). With the combination controller, each UAV was given a fair amount of freedom to violate the distance constraint, which resulted in a wider spread of a distances between the agents as compared to Fig. 12. This is an effect or having a low absolute value for the scale vector. With a higher absolute value on the scale vector there would likely be less of a search behavior and more consistent distance holding behavior. In other words, there might be a trade-off between the two applications when/if needed.

V. Concluding Remarks

This paper described the real-world test of novel decentralized outer-loop controller for the swarm of networking UAVs. It demonstrated that the behavior of a swarm UAV can be quickly changed by varying a few parameters in the parametrically-weighed controller utilizing four understandable control inputs drivers. Field testing proved the ability of the proposed controller to describe several different trade-offs between two (in this particular test series) distinct applications, giving a gradient of swarm behaviors capable of solving the search task, network maintenance or both, while assuring collision free operations. Specifically, the scaling experiments showed how the behavior scales from 1 to 4 UAVs without adaptation. The area search experiment showed how the proposed method is successfully able to explore the given search area. The network maintenance experiment show that the swarm can achieve a stable configuration for the purpose of maintaining a communication

network. Finally, the combination experiment showed how it is also possible to both search and maintain a communication network simultaneously. This is a step towards the development of multi-function swarm systems capable of tackling a variety of missions without much human interaction. Moreover, the human interaction with a swarm can be conducted in a much more natural way compared to todays multi-UAV operations. Rather than micromanaging the swarm, the proposed approach allows an operator to simply select an applicable behavior component in order to tackle the specific task at hand. This frees up valuable human resources and allows a single operator to control more platforms than before.

While the proposed parametric controller has previously been tested in computer simulations, its transition to the real UAV swarm was not direct. This is a known problem, often descried as the reality gap, or the difference between simulated behavior and behavior on real platforms. For the experiments described in this paper, this issue was addressed by adapting the parameters of controller to moderate an otherwise too aggressive behavior. This showcases the ability for the proposed controller to be adapted to a particular mission or scenario on the fly. While this challenge has been resolved, it does pose an interesting topic for future research on how to better tackle this transition and find the ways to make simulations more realistic for better transferability.

Finally, it should be emphasized that the proposed decentralized controller was implemented on an Odroid single-board computer on-board each swarm agent, which is significant and of major importance as a swarm. In essence, a swarm derives the desirable attributes of fault tolerance, scalability and redundancy by not relaying on a single point of failure. Future research will include improvement of the developed swarm, its network and communication infrastructure, to be able to share larger amounts of synchronized information.

ACKNOWLEDGMENTS

The authors would like to acknowledge the Norwegian Defense Research Establishment, the Naval Postgraduate Schools Naval Research Program and Consortium for Robotics and Unmanned Systems Education and Research along with Fulbright Scholar Program for support of this research. They would also like to thank Jørgen Nordmoen and Espen Skjærvold for their contributions to the initial draft of the controller software.

REFERENCES

[1] C. W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," in *ACM SIGGRAPH computer graphics*, vol. 21, pp. 25–34, ACM, 1987.

[2] M. Duarte, S. M. Oliveira, and A. L. Christensen, "Hybrid control for large swarms of aquatic drones," in *Proceedings of the 14th International Conference on the Synthesis & Simulation of Living Systems. MIT Press, Cambridge, MA*, pp. 785–792, Citeseer, 2014.

[3] L. Barnes, M. Fields, and K. Valavanis, "Unmanned ground vehicle swarm formation control using potential fields," in *Control & Automation, 2007. MED'07. Mediterranean Conference on*, pp. 1–8, IEEE, 2007.

[4] Y. Fu, X. Wang, L. Huan, and H. Zhu, "Multi-UAV formation control method based on modified artificial physics," in *Control and Decision Conference (CCDC), 2016 Chinese*, pp. 2523–2529, IEEE, 2016.

[5] C. Belta and V. Kumar, "Abstraction and control for groups of robots," *IEEE Transactions on robotics*, vol. 20, no. 5, pp. 865–875, 2004.

[6] B. G. Woolley and G. L. Peterson, "Unified behavior framework for reactive robot control," *Journal of Intelligent and Robotic Systems*, vol. 55, no. 2-3, pp. 155–176, 2009.

[7] M. Asadpour, D. Giustiniano, K. A. Hummel, and S. Egli, "UAV networks in rescue missions," in *Proceedings of the 8th ACM international workshop on Wireless network testbeds, experimental evaluation & characterization*, pp. 91–92, ACM, 2013.

[8] J. Scherer, S. Yahyanejad, S. Hayat, E. Yanmaz, T. Andre, A. Khan, V. Vukadinovic, C. Bettstetter, H. Hellwagner, and B. Rinner, "An autonomous multi-UAV system for search and rescue," in *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pp. 33–38, ACM, 2015.

[9] P. Bupe, R. Haddad, and F. Rios-Gutierrez, "Relief and emergency communication network based on an autonomous decentralized uav clustering network," in *SoutheastCon 2015*, pp. 1–8, IEEE, 2015.

[10] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue," *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 46–56, 2012.

[11] M. R. Brust and B. M. Strimbu, "A networked swarm model for uav deployment in the assessment of forest environments," in *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on*, pp. 1–6, IEEE, 2015.

[12] J. Fortuna, F. Ferreira, R. Gomes, S. Ferreira, and J. Sousa, "Using low cost open source UAVs for marine wild life monitoring-Field Report," *IFAC Proceedings Volumes*, vol. 46, no. 30, pp. 291–295, 2013.

[13] J. Zelenka and T. Kasanickỳ, "Outdoor UAV control and coordination system supported by biological inspired method," in *Robotics in Alpe-Adria-Danube Region (RAAD), 2014 23rd International Conference on*, pp. 1–7, IEEE, 2014.

[14] J. del Arco, D. Alejo, B. Arrue, J. Cobano, G. Heredia, and A. Ollero, "Multi-UAV ground control station for gliding aircraft," in *Control and Automation (MED), 2015 23th Mediterranean Conference on*, pp. 36–43, IEEE, 2015.

[15] A. Sivakumar and C. K.-Y. Tan, "UAV swarm coordination using cooperative control for establishing a wireless communications backbone," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 3-Volume 3*, pp. 1157–1164, International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[16] 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.

[17] D. Hague, H. Kung, and B. Suter, "Field experimentation of cots-based UAV networking," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pp. 1–7, IEEE, 2006.

[18] W. L. Teacy, J. Nie, S. McClean, and G. Parr, "Maintaining connectivity in UAV swarm sensing," in *GLOBECOM Workshops (GC Wkshps), 2010 IEEE*, pp. 1771–1776, IEEE, 2010.

[19] P. Cevik, I. Kocaman, A. S. Akgul, and B. Akca, "The small and silent force multiplier: a swarm UAVelectronic attack," *Journal of Intelligent & Robotic Systems*, vol. 70, no. 1-4, pp. 595–608, 2013.

[20] L. Bayindir, "A review of swarm robotics tasks," *Neurocomputing*, vol. 172, pp. 292–321, 2016.

[21] S. Mitri, D. Floreano, and L. Keller, "The evolution of information suppression in communicating robots with conflicting interests," *Proceedings of the National Academy of Sciences*, vol. 106, no. 37, pp. 15786–15790, 2009.

[22] J. N. Weaver, D. Z. Frank, E. M. Schwartz, and A. A. Arroyo, "UAV performing autonomous landing on USV utilizing the robot operating system," in *Proc. of the ASME District F-Early Career Technical Conference*, Citeseer, 2013.

[23] S. A. Engebråten, K. Glette, and O. Yakimenko, "Networking-Enabling Enhancement for a Swarm of COTS Drones." submitted to the 14th IEEE International Conference on Control and Automation, Anchorage, AK, June 12-15, 2018.

[24] S. A. Engebråten, J. Moen, O. Yakimenko, and K. Glette, "Evolving a Repertoire of Controllers for a Multi-Function Swarm," in *European Conference on the Applications of Evolutionary Computation*, Springer, 2018.