

COLLECTIVE BEHAVIOR

Optimized flocking of autonomous drones in confined environments

Gábor Vásárhelyi^{1,2,*}, Csaba Virág², Gergő Somorjai^{1,2}, Tamás Nepusz³, Agoston E. Eiben⁴, Tamás Vicsek^{1,2}

Copyright © 2018
The Authors, some
rights reserved;
exclusive licensee
American Association
for the Advancement
of Science. No claim
to original U.S.
Government Works

We address a fundamental issue of collective motion of aerial robots: how to ensure that large flocks of autonomous drones seamlessly navigate in confined spaces. The numerous existing flocking models are rarely tested on actual hardware because they typically neglect some crucial aspects of multirobot systems. Constrained motion and communication capabilities, delays, perturbations, or the presence of barriers should be modeled and treated explicitly because they have large effects on collective behavior during the cooperation of real agents. Handling these issues properly results in additional model complexity and a natural increase in the number of tunable parameters, which calls for appropriate optimization methods to be coupled tightly to model development. In this paper, we propose such a flocking model for real drones incorporating an evolutionary optimization framework with carefully chosen order parameters and fitness functions. We numerically demonstrated that the induced swarm behavior remained stable under realistic conditions for large flock sizes and notably for large velocities. We showed that coherent and realistic collective motion patterns persisted even around perturbing obstacles. Furthermore, we validated our model on real hardware, carrying out field experiments with a self-organized swarm of 30 drones. This is the largest of such aerial outdoor systems without central control reported to date exhibiting flocking with collective collision and object avoidance. The results confirmed the adequacy of our approach. Successfully controlling dozens of quadcopters will enable substantially more efficient task management in various contexts involving drones.

INTRODUCTION

Groups of gregarious animals often display an interesting and spectacular collective pattern (1). They establish ordered structures without collisions in a limited amount of time (2, 3). They can also react extremely fast to environmental changes, such as the sudden appearance of a predator or an obstacle (4, 5). Although these systems are enormously complex, they are also perfectly optimized, and thus, their expressed motion patterns remain gracefully natural (6). When these systems are modeled, one tends to focus on the replication of the smooth optimal motion patterns by making idealistic assumptions about the underlying complexity. This simultaneous simplification of the “input” and “output” explains why so many different statistical physical models of swarm behavior can be efficient in reproducing the same natural collective motion patterns with abstract mathematical formalism.

According to early microscopic agent-based models (7), establishing and maintaining collision-free cohesive flocking require only three simple interactions between idealistic agents: repulsion in short range, velocity alignment in middle range, and attraction in long range. On the basis of these general rules, hundreds of models have emerged to describe the synchronized collective motion of animals, humans, or even migrating cells (8–10). We call these systems self-organized because interactions in them are local and decisions are made by the agents themselves.

One of the recent applications of self-organizing flocking models is in collective robotics (11, 12), where decentralized control algorithms for groups of autonomous drones can be developed on the basis of these interactions, as a prerequisite for safe operation. Driving the behavior of such systems toward some desirable pattern is highly non-

trivial. First, the agents (robots and drones) are autonomous and imperfect. That is, every agent has (i) its own onboard computer for performing the calculations needed for controlling its own actions, (ii) its own sensor system for measuring relative positions and velocities, and (iii) its own communication device for data exchange with neighboring agents. These features reflect the current definition of sensory and reactive autonomy described in (13). Second, these systems should work without central control. That is, although agents can observe each other and may exchange information, they do not send and receive direct control commands because there is no leader within the group, nor is there an external supervisor such as a base station or human overseer.

In developing decentralized control algorithms for swarms of flying robots in stochastic environments where communication outages and delays are common, one soon faces a set of severe challenges that are rarely targeted by previous idealistic agent-based models. As an example, 32 representative microscopic flocking models were selected and compared out of more than 100 (9). Fine and Shell state that “there is no consensus on the precise details of the motions needed to produce rich flocking motions under realistic sensing models, actuation, and dynamics constraints”; most works lack completeness and precision in terms of repeatable modeling and validation; only a few included motion constraints and collision avoidance [e.g., (7, 14, 15)]; and none handled motion constraints explicitly. Finally, only one investigated bounded space with obstacles (16).

While aiming for a stable and scalable flocking model for real flying robots, some serious design challenges need to be addressed:

(1) Reality gap. Flocking models that are stable in simulation under idealistic conditions tend to oscillate and destabilize quickly under real-life conditions when delays, uncertainties, and kinematic constraints are present (17–21).

(2) Adaptivity. Flocking models developed for open space or periodic boundary conditions do not necessarily work in confined spaces and with obstacles in the way (9, 22).

¹Statistical and Biological Physics Research Group of the Hungarian Academy of Sciences, Pázmány Péter sétány 1/A, 1117 Budapest, Hungary. ²Department of Biological Physics, Eötvös Loránd University, Pázmány Péter sétány 1/A, 1117 Budapest, Hungary. ³Molde University College, Brivegen 2, 6410 Molde, Norway. ⁴Department of Computer Science, Vrije Universiteit Amsterdam, de Boelelaan 1081a, 1181HV Amsterdam, Netherlands.
*Corresponding author. Email: vasarhelyi@hal.elte.hu

(3) Scalability. Flocking models that are developed for a specific speed or group size might not be scalable; that is, for higher velocities or larger groups, motion patterns may become unstable (16, 23, 24).

(4) High dimensionality. Flocking models that work well in real life generally have a substantial number of parameters with complex nonlinear interactions that need to be tuned for a wide range of conditions in reasonable time (6).

The largest drone swarms so far were developed for show business by Intel (25) and by Ehang (26) with more than 1000 drones each; however, these drones were individually programmed for predefined trajectories or were centrally controlled and did not satisfy the above criteria of autonomy. The music band Metallica recently included dozens of drones in their concerts that seemed to exhibit some kind of partially autonomous swarming behavior by using a dedicated indoor positioning system and central control mechanisms (27). The U.S. military is also experimenting with fixed-wing drone swarms called Perdix (28). The press release stated that the system of 103 autonomous drones performed adaptive formation flying. The published video suggests that the drones received a set of predefined targets, chose one with a collective decision, and followed that individually. Drones also loitered around a common point, although at different heights. Unfortunately, there are no public details about the control mechanism, the communication scheme, or possible collision avoidance behavior for a reliable assessment of the work.

Autonomous drone swarms also appear in the scientific literature, using indoor motion capture-based (29, 30), outdoor Global Positioning System (GPS)-based (24, 31–33), or even vision-assisted (34, 35) navigation. These systems typically have a much smaller flock size than preprogrammed drone swarms. Although motion capture-based indoor systems (with 20 minidrones and 49 nanodrones in the mentioned citations) are remarkably accurate and dynamic, they represent a very different type of system because they do not have to tolerate profound imperfections such as meter-level positioning, external wind turbulence, or long-range communication decay. The mentioned GPS-based outdoor swarms consisted of no more than 10 drones, except for (32), where 50 fixed-wing unmanned aerial vehicles (UAVs) were flown but at different altitudes, without any explicit collision avoidance mechanism. Vision-based solutions have used only a few drones as the state of the art.

In this article, we build on our previous results (24), where an outdoor drone swarm of 10 agents were presented as a proof of concept with flocking and formation flight capabilities. Although the previous work included preliminary results of closed-area flocking, trajectories were quite oscillatory even though they were executed in the simplest arena: a circular one that actually helped to develop smooth turns. Furthermore, the system—due to the improper treatment of acceleration limits—was not scalable to speeds higher than 4 m/s.

Creating a large decentralized outdoor drone swarm with synchronized flocking behavior using autonomous collision and object avoidance in a bounded area is as yet an unresolved task. We filled this gap by presenting real flights of 30 autonomous quadcopters performing tight and stable flocking in a bounded and cluttered environment. To achieve this goal we used a scalable, optimized control framework, based on realistic dynamic modeling and the explicit treatment of motion constraints in the flocking equations.

The overall descriptors that specified a given setup of our system were the number of drones and the predefined flocking speed. The desired swarm behavior was defined as being collision-free and coherent, that is, with strongly correlated velocity values of the individual

drones, and exhibiting a velocity close to the flocking speed. Furthermore, we aimed for stable swarm behavior with persistent global collective motion patterns resembling those of natural systems with collective intelligence.

The explicit treatment of motion constraints was based on a special concept for the velocity alignment interaction. The key idea was to abandon the generally used fixed spatial boundaries of the local interactions. Instead, the alignment interaction range (and magnitude) was determined dynamically, based on the expected optimal relation between distance and velocity difference. Because the acceleration of agents is limited, they need time and space to brake and avoid collisions. Consequently, the amount of allowed velocity difference must be distance-dependent: Close agents should align perfectly, whereas distant agents are allowed to have larger velocity difference up to a certain limit. To find the upper bound of velocity difference for a given distance, we used an acceleration-limited braking curve. The goal of the alignment was thus to reduce velocity difference below this distance-dependent threshold. This workflow was easy to calculate and provided optimal foundations of scalability in the velocity domain because it took into account the limited acceleration of agents, the source of many undesired oscillations.

The model has many independent parameters with which a broad range of emerging behaviors and visually pleasing collective patterns could be generated. However, our requirements of stability and coherence provide quantifiable criteria for the instantiation of the general model with suitable parameter values. This implies a highly nontrivial optimization problem because of the large number of parameters, their complex nonlinear interactions, and the noisy relations between parameter values and collective motion patterns.

An important element of our approach is the focus on model instances, that is, on models together with specific values of their parameters. The rationale is grounded in our interest in system behavior. Having a model is not enough to generate and study motion patterns; to make a model executable, it must be instantiated by parameter values. Blatantly disregarding theoretical benefits of models, we could say that any model is worth as much as its best instance. Therefore, we considered optimizing the parameter setup as an essential part of the model generation.

This view is missing from current flocking models and the realization of the corresponding robotic swarms, although it stands to reason that natural systems operate at the optimal values of their “tunable” parameters (in the spirit of the Darwinian theory). As the complexity of artificial intelligence increases, we will be forced to include more and more optimization into model design.

To solve the optimization problem, we used evolutionary algorithms, population-based stochastic search methods inspired by natural evolution that have proven competitive in solving hard problems in the face of challenging characteristics such as nondifferentiability, discontinuities, multiple local optima, noise, and nonlinear interactions among the variables (36). The family of evolutionary algorithms contains several variants of the main principles, including genetic algorithms, evolution strategies, differential evolution, and particle swarm optimization (37). Evolution strategies, particularly the covariance matrix adaptation evolution strategy (CMA-ES) (38), are considered to be excellent optimizers in continuous parameter spaces; therefore, we used this algorithm to find good settings for our model.

The main contributions of this paper are (i) a flocking model that explicitly treats motion constraints by maintaining an improved balance between distance and velocity difference; (ii) a method to design

individual drone controllers by optimizing self-organized group-level behavior in a complex, noisy, real-world system; (iii) simulations of this system for presenting its scalability for wide velocity ranges and group sizes; and (iv) the demonstration of the framework with a fleet of 30 quadcopters, performing fully autonomous, synchronized outdoor flights with collective collision and obstacle avoidance in a confined space.

RESULTS

Flocking model instantiation through evolutionary optimization

Our generic flocking model included proper evaluation metrics, that is, order parameters and fitness functions (see Materials and Methods for detailed equations). The model was instantiated with proper parameter values first in simulation. Evolutionary optimization has been used to find parameter values that maximize flock coherence and speed while minimizing collisions. First, we optimized the parameters under conditions suitable for real experiments: using a square-shaped obstacle-free arena with a side length of $L^{\text{arena}} = 250$ m and three different flocking speed (v^{flock}) values: 4, 6, and 8 m/s. The corresponding values for the maximum allowed speed (v^{max}) were 6, 8, and 10 m/s, respectively. We examined the behavior of 100 simulated agents in all cases. For each of the v^{flock} values, we performed at least three independent, randomly initialized optimization processes to identify possible multiple local optima in the parameter space but found very similar solutions and convergence in the alternative evolutionary runs. Therefore, below, we will only refer to the best (highest fitness) evolutionary run for each flocking speed.

The populations in our evolutionary algorithm consisted of parameter vectors whose fitness was determined by a 10-min-long realistic simulation of the system. In all runs, we used a population size of 100 and terminated after 150 generations. The 15,000 fitness evaluations turned out to be sufficient in all cases. The optimization was performed on the Atlasz supercomputer cluster of Eötvös University,

Budapest, Hungary (39); the overall execution time of a single evolutionary run varied between 2 and 6 days.

The evaluation of the phenotypes was based on a single fitness value that was created as the product of six independent normalized partial fitness values (corresponding to minimized collision risk, minimized collision with walls, maximized velocity correlation, velocity magnitude as close as possible to flocking speed, maximized cluster size, and minimized number of disconnected agents). Each partial fitness, as well as the final fitness value, takes values between 0 (worst case) and 1 (ideal case).

The final fitness of the best solutions after optimization converged to 0.92, 0.87, and 0.8 for $v^{\text{flock}} = 4, 6$, and 8 m/s, respectively. In these best stochastic simulation instances, four of the six partial fitnesses were exactly 1 (corresponding to a perfectly collisionless and fully connected flock), and only the velocity correlation and the average velocity reduced the overall fitness. This is a natural and inevitable tendency because hitting the wall in a bounded area requires the flock to change direction, and this cannot be performed without temporarily reducing the speed and velocity correlation. It is worth noting that these high fitness values have been reached under harsh realistic conditions with a 1-s communication delay and substantial noise, which generally act against perfect synchronization. Optimized parameter values are given in table S1.

The stability of the optimized models was investigated next by executing 100 parallel stochastic simulations for each speed. Detailed results about the statistical evaluations can be found in table S2. The average fitnesses naturally became somewhat lower than the maxima: 0.812 ± 0.101 (SD), 0.776 ± 0.086 , and 0.728 ± 0.075 for $v^{\text{flock}} = 4, 6$, and 8 m/s, respectively, with the appearance of occasional collisions. Note that in simulation, the partial fitness of collisions must be a continuous and not-too-steep function; otherwise, the optimizer cannot find the direction of gradients from suboptimal solutions. This means that having a few collisions can be banned only with a limited decrease of fitness, and the optimizer will not devote a special

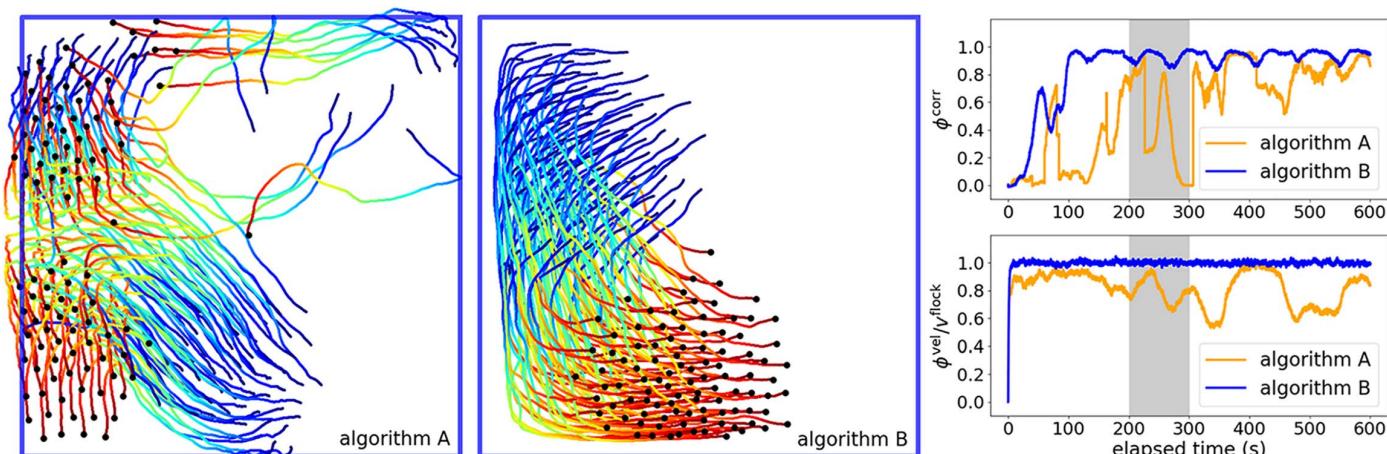


Fig. 1. Comparing previous simulation work with current study. Sample timelines of two order parameters (right, velocity correlation at top and normalized velocity magnitude at bottom) from our previous work (40) (algorithm A) and our novel flocking model (algorithm B). Trajectories corresponding to the gray sections of the timelines are shown for both models on the left, with color mapped to time. Corresponding motion can be seen in movies S1 and S2. Algorithm B performs much better and has a lower transient time. We used the following interaction parameter set for algorithm A: $C^{\text{frict}} = 30 \text{ m}^2$, $r_{\min}^{\text{frict}} = 5 \text{ m}$, $r_0^{\text{rep}} = 20 \text{ m}$, and $p^{\text{rep}} = 1 \text{ s}^{-1}$ [for details on parameters, see (40)]. For algorithm B, we used the optimized parameter set for $v^{\text{flock}} = 4 \text{ m/s}$. Using an average from 10 simulations with the same parameter setup, the order parameters averaged over time were $\phi_A^{\text{corr}} = 0.63 \pm 0.07$ and $\phi_A^{\text{vel}} = 3.37 \pm 0.15 \text{ m/s}$ for algorithm A and $\phi_B^{\text{corr}} = 0.92 \pm 0.002$ and $\phi_B^{\text{vel}} = 3.83 \pm 0.005 \text{ m/s}$ for algorithm B.

priority to entirely collisionless solutions; it will aim for a maximally dense flock implicitly to increase connectedness and velocity correlation, which narrows that stability range with regard to collisions. On the other hand, in reality, collisions have to be eliminated completely at all times as a first rule. Several workarounds exist for this problem: (i) running the optimizer with a larger radius of collision or a larger communication delay to optimize to a solution where minimum interagent distances become larger and (ii) increasing interagent distance (repulsion) manually after optimization with the harmless compromise of reducing overall velocity correlation in the enlarged flock. This time, we chose the second method because, in real flights, one needs to start from an oversecured parameter setup anyways, compared with the optimum with highest possible fitness but lower stability.

Evolutionary optimization produced a huge number of stochastic fitness evaluations, which also contain precious information about the reasonable parameter ranges where fitness is expected to be high. These working ranges are summarized in table S1 to provide tuning information for real drones as well. We also listed all model parameters in table S3 with a detailed explanation on meaning and usage.

Finally, let us note a surprising benefit of evolving parameter settings. The evolutionary algorithm found unexpected parameter settings in both the repulsive and alignment interactions between agents:

(1) Instead of a strong hard-core repulsion (as expected intuitively), a spatially more extended and smoother repulsion is preferable according to the evolutionary optimization.

(2) The velocity alignment between close-by agents should be maximal and mostly distance-independent, allowing only a certain, relatively small velocity difference slack, mainly to speed up the collective turning process.

Overall, we achieved our first goal: The general model could be instantiated well with suitable parameter values in simulation; the optimized setup displayed a stable and efficient flock in the investigated velocity regime, which can serve as the basis for the real field experiments. Comparing our new results to our previous work (40), we can see a substantial increase in flock stability and coherence (see Fig. 1). Movies S1 and S2 show corresponding motion of old and new simulations at 4 m/s.

Scalability in velocity

The most important feature of the acceleration-limited velocity alignment term in the agent-agent and wall-agent interactions is its inherent scalability in the velocity regime. To demonstrate this, we performed

further optimizations with higher flocking speed values: 16 and 32 m/s, beyond most bird migration speeds (41). We changed two parameters in the environmental setup: We increased the communication range from 80 to 160 and 320 m and the size of the arena from 250 to 500 and 1000 m for the two speed values, respectively. The first change was needed because the communication delay remained at 1 s, which created a much larger positional uncertainty and braking distance at higher speeds. This can be compensated only if agents have information about each other at higher distances. The second change was a consequence of the first: With such a large communication delay and speed, the interagent distances became larger, and thus, the flock could not fit into a smaller arena with enough freedom for nice flocking behavior.

The optimized solutions obtained a high fitness again, with a maximum of 0.91 and 0.89 and a statistical average of 0.79 ± 0.12 and 0.63 ± 0.23 for 16 and 32 m/s, respectively. Detailed fitness values of the statistical evaluations are summarized in table S2. Note that for the highest speed of 32 m/s, the lower average of the fitness is mostly a result of the slightly increased number of collisions (3.53 ± 3.61). We investigated the role of communication range and delay in this case and found that collisions disappeared when we reduced the delay below 1 s, assuming that the communication range was large enough (see Fig. 2 for details). The first part of movie S3 shows the optimized and stable flocking behavior of 100 agents in simulation for 16 m/s.

Scalability in agent number with collective obstacle avoidance

Because of the locality of the communication and the interactions, the proposed flocking model provides the foundations of scalability in agent number. However, when more agents synchronize in such a nonequilibrium system, the overall momentum of the flock also scales with flock size, which creates increased “pressure” of agents when the flock bumps into walls. In similar situations, human crowds are prone to injuries or even death during panic events (16) or, for example, around mosh pits at heavy metal concerts (42). To provide collisionless solutions with higher agent numbers, one needs to prevent accumulating pressure of agents, for example, using obstacles inside the arena. Obstacles can be introduced with the same type of interactions as surrounding walls (see Materials and Methods for details).

Without going into statistical details, movie S3 shows some examples of the realistic simulation with flock sizes between 30 and 1000 and flocking speeds between 4 and 32 m/s, with different types of obstacles in the way. Overall, we see that the presented flocking

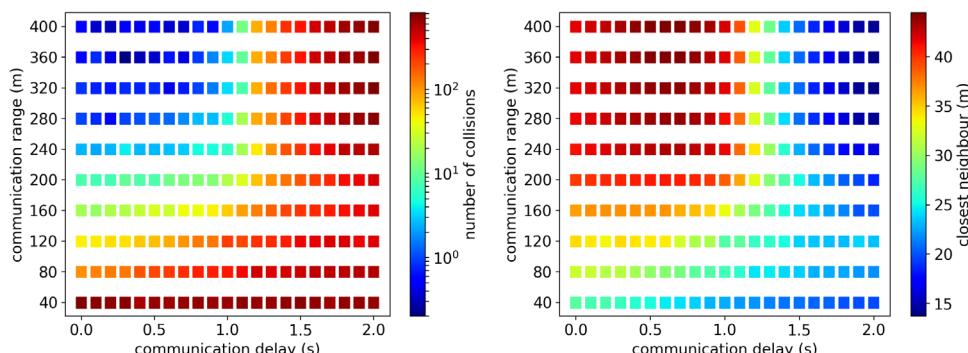


Fig. 2. Distribution of the number of collisions and the average closest-nearest neighbor distance as a function of communication range and delay. Every bin is the average of 20 simulations with the optimized parameter setup for a flocking speed of 32 m/s. As can be seen, safe flocking can be achieved with small enough delay (<1 s) and large enough communication range (>240 m) for this setup.

model can be used as a general framework to handle flocking-type motion in a confined area, with a large number of agents, large flocking speeds, and obstacles.

Experimental results with outdoor flying robots

We implemented the described model in our custom-built multidrone framework as a control algorithm. Details of the drone setup are given in Materials and Methods.

We performed two-dimensional experiments with 30 drones flocking at the same altitude, with a horizontal speed of $v^{\text{flock}} = 4, 6$, and 8 m/s and $L^{\text{arena}} = 200$ to 260 m . Parameter values were initialized mostly within the working range of the corresponding evolutionary optimized results (table S1) with some notable changes from simulation optima based on the following precautions and preventive safety considerations: (i) repulsion strength was somewhat increased (larger gain) to minimize the chance of collisions (compromise: sparser flock); (ii) coefficient of alignment was increased to reduce possible oscillations (compromise: more sluggish motion); (iii) shill interaction strength was reduced (smaller shill velocity), but range was extended (larger distance offset) to avoid very high interaction terms at walls while maintaining overall strength. The final parameter values used in the experiments and detailed comments on their possible changes from simulation optima are summarized in table S4.

To assess the quality of the flights, we calculated a set of order parameters that describe different aspects of the motion. We calculated the cluster-dependent velocity correlation (ϕ^{corr}), the average velocity

(ϕ^{vel}), the average and minimum of interagent distances ($\text{min}r_{ij}$ and $\bar{r}_{ij}^{\text{min}}$), and the average normalized velocity expressed in local angle polar (LAP) coordinates (43):

$$\phi^{\text{LAP}} = \frac{v_y r_x - v_x r_y}{v^{\text{flock}} \sqrt{r_x^2 + r_y^2}} \quad (1)$$

where r_x and r_y represent the average position of agents in the horizontal plane relative to the center of the arena and v_x and v_y are the average velocity components of the agents in the horizontal plane. ϕ^{LAP} is a simple descriptor of rotational behavior: An instantaneous value of zero means no correlated motion tangentially, whereas a value of 1 or -1 represents correlated circular flight in the counterclockwise and clockwise directions, respectively.

Figure 3 shows the order parameters as a function of time for a selection of full 10- to 15-min stable flights (flight time depended on battery and wind conditions, with a maximum tolerated wind of about 40 km/hour). Flights were selected to represent the most common emerging collective patterns, namely, repetitive circular and diagonal flights in an obstacle-free arena and a lively random collective flight with obstacles. Note that the emergent rotational pattern is a universal one (44), appearing in a large variety of flocking systems ranging from elongated rods (45) through locusts (2) and fish (5) to humans (46, 47). A 10-min part of the trajectories from the same flights can be seen in Fig. 4, a long-exposure photo of a shorter section is shown in Fig. 5,

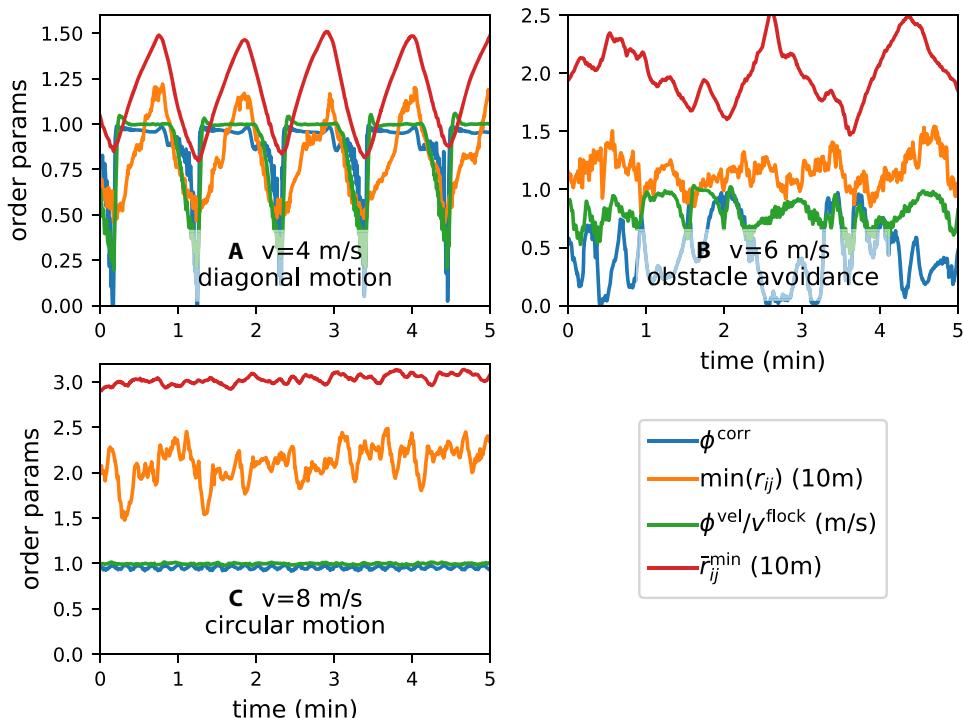


Fig. 3. Order parameters as a function of time for different v^{flock} values during real experiments with 30 drones. ϕ^{corr} is the cluster-dependent velocity correlation, $\phi^{\text{vel}}/v^{\text{flock}}$ is the average normalized velocity, and $\bar{r}_{ij}^{\text{min}}$ represents the average of the closest neighbors, whereas $\text{min}(r_{ij})$ is the minimum of the closest neighbors. The depicted region corresponds to the middle 5 min of Fig. 4. There are two typical, mostly stable behaviors in a square-shaped arena without obstacles: (A) shows mostly linear motion along the main diagonals with a cyclic expansion and shrinking of the flock (cyclic red and orange curves) and sudden turns at corners (blue and green curves dropping to zero), whereas (C) shows circular motion within the boundaries (nearly constant order parameters at all times). (B) The repetitive, trivial patterns were broken and became livelier due to obstacles in the way. Correspondingly, velocity correlation and average velocity magnitude drops, whereas minimal interagent distance remains the same, showing the stability of the flight even in this obstructed case.

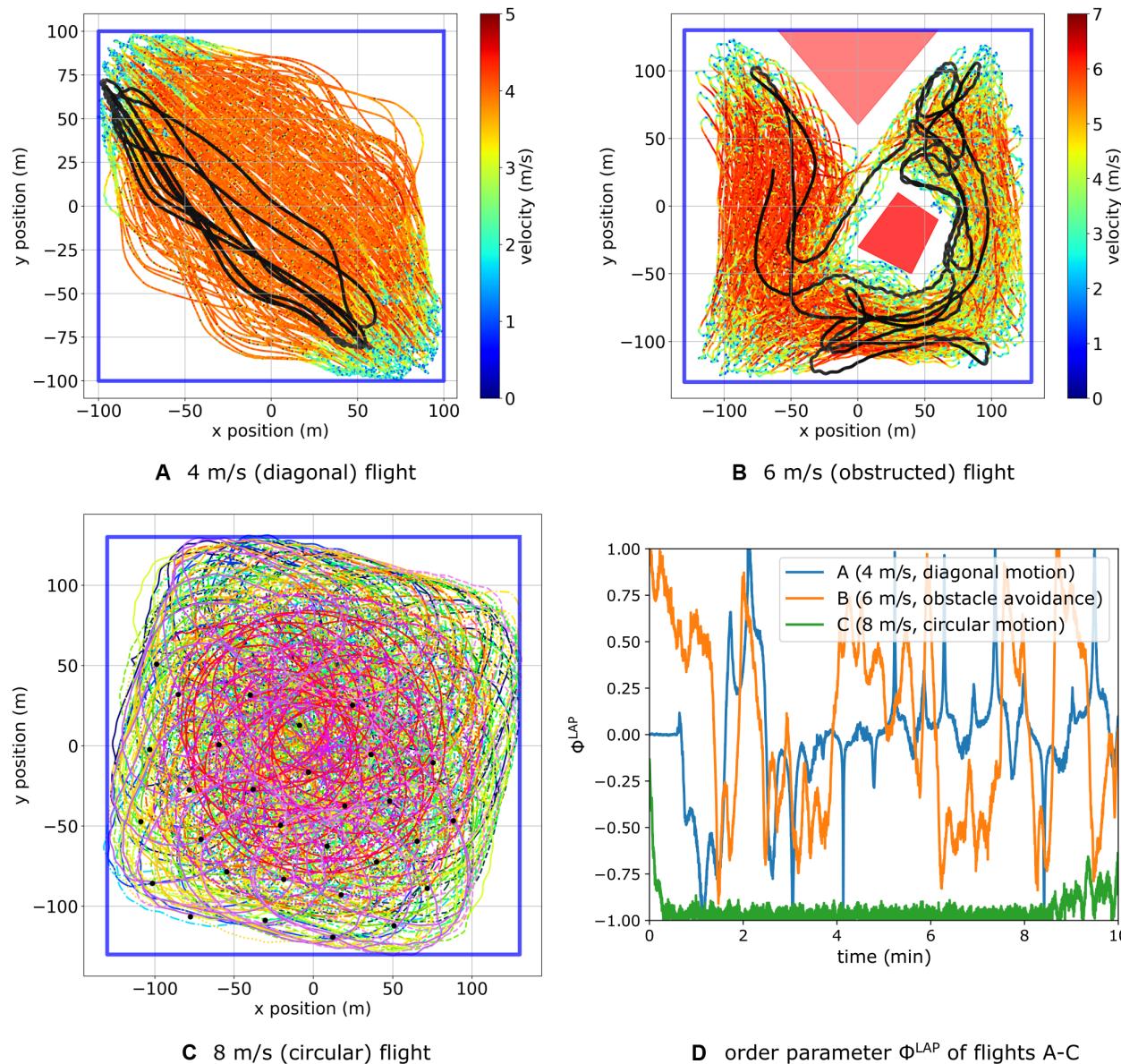


Fig. 4. Multidrone flight trajectories and corresponding order parameters. Ten-minute trajectory sections of 30 drones in the horizontal plane for (A) 4 m/s, (B) 6 m/s and (C) 8 m/s flights, representing a selection of typical flight patterns. Blue squares show the boundaries of the virtual arena. (A) The trajectories show diagonal linear motion of the flock, bouncing back from the right-angled corners. Trajectory colors represent speed in the horizontal plane, whereas a random single trajectory is highlighted in gray scale. (B) The motion is still locally correlated, but the obstacles (red shapes) induce a very rich dynamic pattern, resembling lively flocks of birds or other animals. (C) The trajectories show a highly correlated close-to-circular flight. Colors and line styles are mapped to individual drones here; black dots show terminal positions of drones. (D) Comparison of the three qualitatively different behaviors of (A) to (C) with the timeline of a dedicated order parameter: the average normalized velocity, expressed in local angle polar coordinates (ϕ^{LAP}).

whereas movies S4 to S6 show the corresponding dynamic flight log visualizations for the three selected flights. Finally, movie S7 shows a summary of the results including actual footage of the flights, too.

As can be seen from the experiments, real drones performed well within the whole tested flocking speed regime. Namely, there are no critical oscillations or collisions with each other, with the wall, or with obstacles. Furthermore, motion is smooth and lively during collective turns when the flock hits the walls, flies into the right-angled corner of the arena, or splits when obstacles are in the way. On the basis of Fig. 3, we can see that, for the flocking speed regime of 4 to 8 m/s, the average closest-neighbor distance varied between 12 and 30 m and the

minimum interagent distance remained between about 5 and 15 m. With Global Navigation Satellite System (GNSS) positioning errors in the range of 2 and 3 m and possible communication outages in the order of 1 s, we believe that these results show a tight and stable flock.

DISCUSSION

We have presented a tunable distributed flocking model for a large group of autonomous flying robots with which they were capable of maintaining stable, collision-free collective motion in a closed space with or without obstacles, within a large velocity regime. The solution is based on the



Fig. 5. Long-exposure photo of a flight with multiple drones. [Credit: Zsolt Bézsényi]

simplest force-based rules presented by the earliest self-propelled particle models, repulsion and alignment, but uses a form of alignment that takes into account the desired acceleration regime of agents. The model works in a noisy environment, with inaccurate sensors and short-range communication devices, and in the presence of substantial communication delay and with possible local communication outages—these are usual features of current outdoor multidrone experimental set-ups. The model produces a very rich dynamic of motion, especially in an obstructed space, with a variety of emergent collective motion patterns, resembling lively natural flocks.

The model has 11 tunable parameters that call for automated and efficient optimization methods, such as the CMA-ES. With the introduced single-objective fitness function taking into account several important order parameters, we could instantiate our model to find working ranges and optimal parameter setups for a wide range of velocities. Both our optimized simulation results with 30 to 1000 drones and 4 to 32 m/s flocking speeds and our real experiments on 30 drones and 4 to 8 m/s flocking speeds showed stable flocking behavior of agents.

Because of the concept behind the new alignment term, we believe that much higher velocity regimes can be targeted with the same approach, if needed. For this, one would need to have larger interagent distances, larger radii of interactions, and thus proper large-distance communication methods.

Limitations of the general usage of the model could arise from further scaling in the velocity regime and in the number of drones. With 30 drones, we demonstrated an order-of-magnitude scalability relative to the smallest drone swarms of only a few agents; however, further scaling in numbers also implies an increase in emergent pressure among frontal agents facing walls and obstacles, such as that in human crowds, resulting in smaller, possibly dangerous interagent distances. This issue has to be solved in systems with a very high number of agents. A related limiting factor is the large number of necessary parameters that need to be optimized for every system separately. Although we have selected the parameters with special care to have independent meaning and significance, in such a complex system, a deep understanding of the rich dynamic behavior and substantial experience is needed before safely applying the results to other vehicles with different characteristics. A final shortcoming of the present study is the lack of rigorous quantitative analysis of stability, because it is not straightforward

to do in such a high-dimensional parameter space. We avoided this by analyzing the fitness evaluations statistically and gave approximate, independent ranges for the parameters above which fitness is expected to be high. For certain applications, though, a more sophisticated analysis would be more appropriate. Despite these limitations, we believe that the presented concept of velocity alignment, the model in general, together with the fitness evaluation method, can be used optimally in a wide range of multidrone scenarios requiring sophisticated cooperation and/or collaboration.

MATERIALS AND METHODS

A tunable self-propelled flocking model

On the basis of statistical-physical methods, a basic flocking model was introduced in (40) and (24). This is a minimal realistic approach of flocking behavior that was demonstrated to work with up to 10 flying robots with a maximum flocking speed of 4 m/s. In this model, the three interaction terms yield a momentary desired velocity vector \mathbf{v}^d , which has to be achieved by the agents. Of course, maintaining the desired velocity is generally hard due to several robot-specific deficiencies such as communication delays and reaction times, inaccuracy of the onboard sensors, effects of wind, sensor signal outages, inertia etc. The question we examine here is whether there are interactions that can guarantee more stable behavior than previously published attempts under these conditions for larger flocking speeds and also in confined spaces. In the subsections below, we present the exact mathematical formulation of our novel generic flocking model, taking into account the realistic limitations of autonomous flying agents with the explicit treatment of motion constraints in the equations.

Repulsion

For local repulsion, we choose a simple half-spring model, that is, a linear distance-dependent central velocity term with a cutoff at a maximum interaction range, r_0^{rep} , under which agents start to repulse each other:

$$v_{ij}^{\text{rep}} = \begin{cases} p^{\text{rep}} \cdot (r_0^{\text{rep}} - r_{ij}) \cdot \frac{r_i - r_j}{r_{ij}} & \text{if } r_{ij} < r_0^{\text{rep}} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

In the equation above, p^{rep} is the linear gain of the pairwise repulsion and $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ is the distance between agents i and j . The total repulsion term calculated for agent i with respect to the other agents is

$$\mathbf{v}_i^{\text{rep}} = \sum_{j \neq i} \mathbf{v}_{ij}^{\text{rep}} \quad (3)$$

where j is iterated for all other agents. Note that we have experimented with more complex repulsion functions in the $v(r)$ plane (40, 48), but according to our experience, the half-spring model is sufficiently simple and effective at the same time.

Velocity alignment

Pairwise velocity alignment can be obtained with a velocity term that depends on the difference of the velocity vectors of nearby agents. Previous works typically used a power law of the velocity difference of the interacting agents decaying in space asymptotically to zero (49, 50). These models work fine in some specific conditions, within a lower velocity regime. However, our objectives regarding the velocity alignment are complex. It is the very term that synchronizes motion to achieve collective flocking behavior, but it also has to serve as a damping medium, reducing self-excited oscillations emerging due to the delayed and noisy response to for example, repulsion. It has to be local, but it also has to be scalable for large velocities (and therefore large possible velocity differences) at the same time. This last condition implies that if the acceleration of the agents is limited, large velocity differences should be relaxed at larger distances to avoid collisions.

To fulfill all the requirements above, as a theoretical basis for the velocity alignment term, we have chosen an ideal braking curve, that is, a smooth velocity decay function in space [denoted by $D(\cdot)$], with constant acceleration at high speeds and exponential approach in time at low speeds (51):

$$D(r, a, p) = \begin{cases} 0 & \text{if } r \leq 0 \\ rp & \text{if } 0 < rp < a/p \\ \sqrt{2ar - a^2/p^2} & \text{otherwise} \end{cases} \quad (4)$$

where r is the distance between an agent and an expected stopping point, a is the preferred acceleration, and p is a linear gain also determining the crossover point between the two phases of deceleration.

The rationale behind our velocity alignment term is to prohibit two agents having a larger velocity difference at a given distance than what is allowed by this ideal braking curve and, thus, to serve as kind of a motion planning term in the otherwise momentary force-based equations (see Fig. 6 for a visual representation):

$$v_{ij}^{\text{frictmax}} = \max\left(v_{ij}^{\text{frict}}, D(r_{ij} - r_0^{\text{frict}}, a^{\text{frict}}, p^{\text{frict}})\right), \quad (5)$$

$$v_{ij}^{\text{frict}} = \begin{cases} C^{\text{frict}}(v_{ij} - v_{ij}^{\text{frictmax}}) \cdot \frac{v_i - v_j}{v_{ij}} & \text{if } v_{ij} > v_{ij}^{\text{frictmax}} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

In the equations above, C^{frict} is a linear coefficient of the velocity alignment error reduction, v_{ij}^{frict} is a velocity slack to allow for certain amount of velocity difference independently of interagent distance, r_0^{frict} is the distance of the stopping point for agent i relative to and in

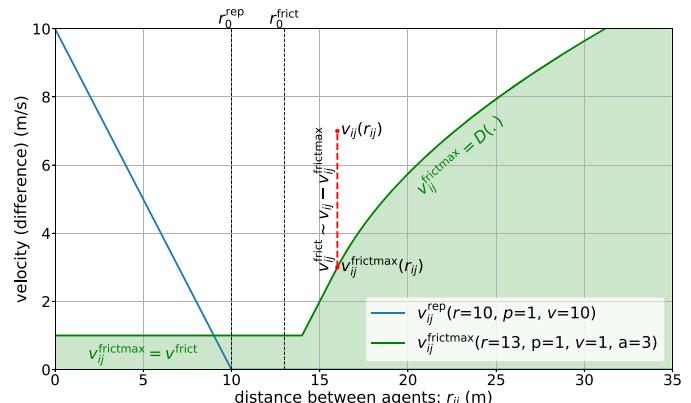


Fig. 6. Visual explanation of the interaction terms. The blue line depicts repulsion between agents as a function of interagent distance. The green line is the maximum allowed velocity difference between agents as a function of interagent distance. The velocity alignment term is proportional to the difference between this and the actual velocity difference between agents (red dashed line). All exemplary parameter values are in SI units.

front of agent j , p^{frict} and a^{frict} are the linear gain and the acceleration parameters of the pairwise alignment, and $v_{ij} = |\mathbf{v}_i - \mathbf{v}_j|$ is the amplitude of the velocity difference between agents i and j . The total velocity alignment term calculated for agent i with respect to the other agents—similarly to the repulsion term—is

$$\mathbf{v}_i^{\text{frict}} = \sum_{j \neq i} \mathbf{v}_{ij}^{\text{frict}} \quad (7)$$

where j is iterated for all other agents. Note that the superscript “frict” comes from the concept that velocity alignment should be a strong local velocity-damping term, analogous to viscous friction (24).

In addition, the locality condition of the velocity alignment in this form is only implicitly included: The interaction range is upper bounded by the distance where $D(\cdot) = 2v^{\text{max}}$. On the other hand, this solution allows for flexible scalability in the velocity domain. If the flocking speed is much higher, then it is obviously preferable to start reducing velocity difference at a much larger distance (as an analogy, compare the deceleration behavior and braking distances of a toy drone against an object and a large manned aircraft reaching its destination).

Interaction with walls and obstacles

Long-range attraction (7) is not explicitly part of our flocking system. To keep agents together, we instead define a bounded flight arena for the agents surrounded with soft repulsive virtual walls. One of the ideal ways to define such repulsion is to define virtual “shill” agents near the arena walls (52). These virtual agents are heading toward the arena with a certain speed, v^{shill} . The real agents close to the walls should relax their velocity to the velocity of the shill agents. We do this here with the velocity alignment term introduced before:

$$v_{is}^{\text{shillmax}} = D(r_{is} - r_0^{\text{shill}}, a^{\text{shill}}, p^{\text{shill}}) \quad (8)$$

$$v_{is}^{\text{wall}} = \begin{cases} \mathbf{v}_{is}^{\text{frict}}(C - 1) & \text{if } v_{is} > v_{is}^{\text{shillmax}} \\ \left(v_{is} - v_{is}^{\text{shillmax}}\right) \cdot \frac{v_i - v_s}{v_{is}} & \text{if } v_{is} > v_{is}^{\text{shillmax}} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

These equations are very similar to Eqs. 5 and 6, with two simplifications: We do not allow velocity slack for the wall and keep the error proportional term (C^{shill}) at 1 to have the strongest shill alignment possible. In the equations above, the index s refers to the shill agents defined for all wall polygon edges separately; $r_{is} = |\mathbf{r}_i - \mathbf{r}_s|$, where \mathbf{r}_s is the position of the shill agent, located at the closest point of the given edge of an arbitrarily shaped convex wall polygon relative to agent i ; $v_{is} = |\mathbf{v}_i - \mathbf{v}_s|$, where \mathbf{v}_s is the velocity of the shill agent, pointing perpendicularly to the wall polygon edge inward the arena, with magnitude v^{shill} .

Convex obstacles inside the arena can be avoided with the same concept, but with shill agents moving outward from the obstacle, not inward, as described above for the arena. Another difference is that, whereas all wall polygon edges generate a separate shill agent inside the arena, obstacles are represented with a single shill agent located at the closest point of the obstacle polygon relative to the agent. Thus, for every agent i and obstacle s , we can define a velocity component $\mathbf{v}_{is}^{\text{obstacle}}$ similarly to Eq. 9, using the same shill parameters as for the wall.

Self-propelling term

Besides the agent-agent and agent-wall interactions introduced above, a simple self-propelling term is added to the desired velocity of the agents. For the i th agent, this term is parallel with the actual velocity vector, \mathbf{v}_i and has a certain constant magnitude, v^{flock} .

Final equation of desired velocity

To calculate the desired velocity, we take the vectorial sum of all the interaction terms introduced before:

$$\tilde{\mathbf{v}}_i^d = \frac{\mathbf{v}_i}{|\mathbf{v}_i|} v^{\text{flock}} + \mathbf{v}_i^{\text{rep}} + \mathbf{v}_i^{\text{frict}} + \sum_s \mathbf{v}_{is}^{\text{wall}} + \sum_s \mathbf{v}_{is}^{\text{obstacle}} \quad (10)$$

After this superposition, we also introduced a cutoff at v^{max} , keeping the direction of the desired velocity but reducing its magnitude if it is over the limit:

$$\mathbf{v}_i^d = \frac{\tilde{\mathbf{v}}_i^d}{|\tilde{\mathbf{v}}_i^d|} \cdot \min \left\{ |\tilde{\mathbf{v}}_i^d|, v^{\text{max}} \right\} \quad (11)$$

In the flocking model above, we have introduced a substantial number of parameters to give the necessary degree of freedom to the general model. To help readers understand complex model behavior, we provide an overview of the parameters with detailed descriptions on meaning and usage (table S3).

Tuning the above model means that we choose an optimal set of parameters for a fixed flocking speed v^{flock} and maximal speed v^{max} for a given arena with characteristic size L^{arena} . The other parameters (namely, r_0^{rep} , r_0^{frict} , C^{frict} , v^{frict} , p^{frict} , a^{frict} , r_0^{shill} , v^{shill} , p^{shill} , and a^{shill}) have to be optimized. Note that the parameter space is 11-dimensional; therefore, manual tuning, global optimization methods, or parameter sweeping would be generally too time-consuming.

General model of a flying robot

For testing any flocking algorithm in a realistic environment before actual flights, we used a simulation framework, which was originally developed for modeling special features of flying robots based on second-order ordinary differential equations. In this subsection, we

present only the main features of this framework, without details. For further details, see (40) or download the simulation framework from <https://github.com/csviragh/robotsim>. The following general features of flying robots can be taken into account with our framework:

(1) Communication delay. The position and velocity data received by an agent from neighboring agents are old due to the necessary time for data transmission and processing. In the simplest case, we modeled this effect with a constant time delay t^{del} .

(2) Inertia. A flying robot cannot change its velocity immediately because of its mass, aerodynamic effects, and specific features of its low-level control algorithm. We assumed that the real velocity \mathbf{v}_i converges to the desired velocity \mathbf{v}_i^d exponentially with a characteristic time τ^{CTRL} . A maximal acceleration of the units (a^{max}) is also assumed.

(3) Refresh rate of the sensors. The agents cannot update their sensory data continuously, only with a nonzero time period t^s . For simplicity, in the simulation framework, this parameter is constant and uniform for all agents.

(4) Locality of the communication. If two agents are too far from each other, they cannot exchange messages; that is, they do not see each other. This is a common feature of any decentralized, radio-based communication device. For modeling this effect, a maximum communication range r^c is defined in our approach.

(5) Inaccuracy of the onboard sensors. We also had to model the fluctuating behavior of measured positions and velocities. This behavior can be described as a stochastic process. For the i th agent, this process can be chosen as a fictive Langevin equation with a Gaussian noise term and a parabolic potential centered at \mathbf{r}_i . The noise can be characterized by its SD σ^s .

(6) Outer noises. To take into account the environmental effects such as wind compensation of the low-level control algorithm, we added a delta-correlated Gaussian noise term with SD σ to the acceleration of the robots.

According to the list above, one can define a simulated realistic homogeneous multirobot system by giving certain values to all elements of the set $\{t^{\text{del}}, \tau^{\text{CTRL}}, a^{\text{max}}, r^c, t^s, \sigma^s, \sigma\}$. We presented the optimization through a realistic example based on measurements performed with quadcopters [for further details, see (24, 40)]; we chose the values listed in table S5.

Note that we prefer and tend to overestimate the errors compared with their real mean value to simulate worst-case scenarios. This makes model selection and optimization harder, but once a proper solution is found, it will ensure a larger stability range in real experiments.

Order parameters

In this subsection, the quantitative requirements of stability and coherence of a flock will be discussed. To this end, we used three measures: coherence, collision avoidance, and obstacle avoidance. Coherence is frequently described by the spatial velocity correlation, which is a commonly accepted order parameter of collective motion. However, on a large area, correlated subflock clusters without global coherence can also be treated as a reasonably good solution for flocking. Therefore, it is practical to measure the correlation function only within connected clusters. To define clusters, we defined a communication graph that contains the agents as nodes. An edge exists between two nodes if the agents referred by the nodes are closer to each other than a given r^{cluster} , typically defined by the range of interactions. We used the value

$$r^{\text{cluster}} = \max \left(r_0^{\text{rep}}, r_0^{\text{frict}} + \tilde{D}(v^{\text{flock}}, a^{\text{frict}}, p^{\text{frict}}) \right) \quad (12)$$

where $\tilde{D}(v, a, p)$ is the braking distance r for which $D(r, a, p) = v$.

Let N be the number of agents, N_i be the number of agents in the cluster that contains the i th agent, and let J_i refer to the set of indices of agents that are in the same cluster as the i th agent. On the basis of these notations, the expression of cluster-dependent velocity correlation takes the following form:

$$\phi^{\text{corr}} = \frac{1}{T N} \int_0^T \sum_{i=1}^N \frac{1}{N_i - 1} \sum_{j \in J_i} \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{|\mathbf{v}_i| |\mathbf{v}_j|} dt \quad (13)$$

This value needs to be maximized. Besides high velocity correlation inside clusters, one can characterize the flock with properties of the communication graph itself. For example, the number of disconnected points (N^{disc} , referring to agents that cannot communicate with any other agents) can be measured, or a minimum cluster size (N^{min}) can be defined as an error threshold for avoiding the situation where only very small groups of agents are present in the system. Of course, the minimum cluster size should depend on the total number of agents. For intuitive reasons, we chose $N^{\text{min}} > N/5$ as an experimentally good lower threshold.

The next important requirement is collision-free motion. We defined a characteristic distance $r^{\text{coll}} = 3$ m, which refers to a dangerous zone around agents. If two agents are closer to each other than r^{coll} , a dangerous situation (collision) occurs. During algorithm tuning, the number of collisions should be minimized, which is similar to minimizing the following parameter, the so-called collision risk:

$$\phi^{\text{coll}} = \frac{1}{T N(N-1)} \int_0^T \sum_{i=1}^N \sum_{j \neq i} \Theta(r_{ij}(t) - r^{\text{coll}}) dt \quad (14)$$

where $\Theta(\cdot)$ is the Heaviside step function.

With the wall-agent interaction velocity term defined as Eq. 9, one can restrict the motion of the flocking agents into a closed space, which can be viewed as a method for maintaining the cohesiveness of the group (this is a general criterion of flocking behavior) but also can be treated as a general approach of obstacle avoidance, which is a common task in collective robotics. Below, we define an order parameter for calculating the possible collisions with the walls of the arena or obstacles (this parameter has to be minimized):

$$\phi^{\text{wall}} = \frac{\int_0^T \sum_{i=1}^N \Theta(\tilde{r}_{is}(t)) \tilde{r}_{is}(t) dt}{\int_0^T \sum_{i=1}^N \Theta(\tilde{r}_{is}(t)) dt} \quad (15)$$

where \tilde{r}_{is} is the signed form of r_{is} , taking a positive value outside, and a negative value inside, the arena (and the opposite for obstacles), assuring that the calculation of the average is performed only at the points where the agents are outside of the arena (or inside obstacles).

With the parameters presented above, one can define a quantitative criterion for safe flocking behavior for the simulated (or real) robots, namely, $\phi^{\text{corr}} \rightarrow 1$, $\phi^{\text{coll}} \rightarrow 0$, $\phi^{\text{wall}} \rightarrow 0$, and $N^{\text{min}} > N/5$.

Finally, we also require the flock to move with a certain flocking speed:

$$\phi^{\text{vel}} := \frac{1}{T N} \int_0^T \sum_{i=1}^N |\mathbf{v}_i(t)| dt \rightarrow v^{\text{flock}} \quad (16)$$

Fitness function

Instead of parameter sweeping or any relatively slow global optimization method, the search for the optimal values of the 11 model parameters

(r_0^{rep} , p^{rep} , r_0^{frict} , C^{frict} , v^{frict} , p^{frict} , a^{frict} , r_0^{shill} , v^{shill} , p^{shill} , and a^{shill}) is known to be much more efficient by using evolutionary optimization. Within such a framework with the defined set of order parameters, one can choose between single- or multiobjective evolutionary algorithms. We chose using a single-valued fitness function that contains several criteria about the order parameters presented in the previous subsection. According to these assumptions, a global fitness function can be defined using three different types of transfer functions (the value of these functions should be between 0 and 1).

The first type is a monotonically growing function, $F_1(\phi)$, which converges to 1 with increasing ϕ :

$$F_1(\phi, \phi_0, d) = 1 - S(\phi, \phi_0, d) \quad (17)$$

where $S(x, x_0, d)$ is a sigmoid function with a smooth sinusoidal decay from $x_0 - d$ to x_0 :

$$S(x, x_0, d) = \begin{cases} 1 & \text{if } x < x_0 - d, \\ \frac{1}{2} \left(1 - \cos\left(\frac{\pi}{d}(x - x_0)\right) \right) & \text{if } x_0 - d < x < x_0 \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

The second transfer function is derived from the probability density of the normal distribution, with a single maximum at $\phi = 0$ and a smooth decay around it:

$$F_2(\phi, s) = \exp\left(-\frac{\phi^2}{s^2}\right) \quad (19)$$

Finally, the third transfer function is a sharp peak, which gives a harsh constraint to the fitness around $\phi = 0$:

$$F_3(\phi, a) = \frac{a^2}{(\phi + a)^2} \quad (20)$$

A visual illustration of the shape of the three transfer functions is given in Fig. 7. With these transfer functions, we can construct a single-objective fitness function that takes into account all necessary requirements of safe flocking behavior:

$$F = F^{\text{speed}} \cdot F^{\text{coll}} \cdot F^{\text{disc}} \cdot F^{\text{cluster}} \cdot F^{\text{wall}} \cdot F^{\text{corr}} \quad (21)$$

where

$$\begin{cases} F^{\text{speed}} = F_1(\phi^{\text{vel}}, v^{\text{flock}}, v^{\text{tol}}), \\ F^{\text{coll}} = F_3(\phi^{\text{coll}}, a^{\text{tol}}), \\ F^{\text{disc}} = F_3(N^{\text{disc}}, N/5), \\ F^{\text{cluster}} = F_3(N^{\text{min}}, N/5, N/5), \\ F^{\text{wall}} = F_2(\phi^{\text{wall}}, r^{\text{tol}}), \\ F^{\text{corr}} = \Theta(\phi^{\text{corr}}) \phi^{\text{corr}} \end{cases} \quad (22)$$

and v^{tol} , a^{tol} , and r^{tol} are tolerance values for speed, collision risk, and collisions with walls and obstacles, respectively. These tolerance

values can be chosen arbitrarily, depending on the absolute and relative importance of the partial fitness components in the optimization. We chose $v^{\text{tol}} = \frac{1.5}{4} v^{\text{flock}}$ m/s, $a^{\text{tol}} = 0.00003$, and $r^{\text{tol}} = 2$ m, which gave a balanced weighting to the overall fitness function. Note that the order parameter ϕ^{corr} is present in the fitness function only as a multiplicative term with a cutoff at 0 because its values are originally between -1 and 1 .

With this method, we created a single-objective optimization scenario, which can be solved using state-of-the-art evolutionary

algorithms such as the CMA-ES (38, 53). To perform this task, we used an open-source Python library (54), with default settings (see table S6). Parameters were initialized at mid-value, with an initial SD of one-sixth of their allowed range.

Drone setup

Our quadcopters use the Pixhawk autopilot (55) for controlling the rotors with a slightly modified ArduPilot controller [see our open-source code that runs on the Pixhawk system at (56)]. We also used an onboard, Linux-based companion minicomputer (Odroid C1+) through which we gave desired velocity commands at 20 Hz to the autopilot. The desired velocity was calculated onboard using the flocking model presented above as the control algorithm.

We used two independent, parallel wireless modules for interagent communication in the 2.4-GHz range, both broadcasting the same status packets. One is an XBee module broadcasting through its own proprietary protocol at 1 Hz; the other one is a small universal serial bus (USB) wifi dongle (Odroid Wifi Module 0) transmitting user datagram protocol (UDP) packets through a local ad hoc wireless network at 10 Hz. The two modules are complementary in bandwidth and range (XBee being small bandwidth and longer range and WiFi being large bandwidth but shorter range). Packets contained an absolute time stamp, geodetic position, and velocity principally measured by onboard GNSS receivers and other safety-related status info about the actual state of the drone that was not relevant to the main control algorithm. Relative position and velocity were calculated by the differences of GNSS-based absolute measurements. The net payload size of a status packet was 46 bytes.

Because of the properties of the wireless media and the broadcasting transport protocols, packet delivery was unreliable; at a transmission

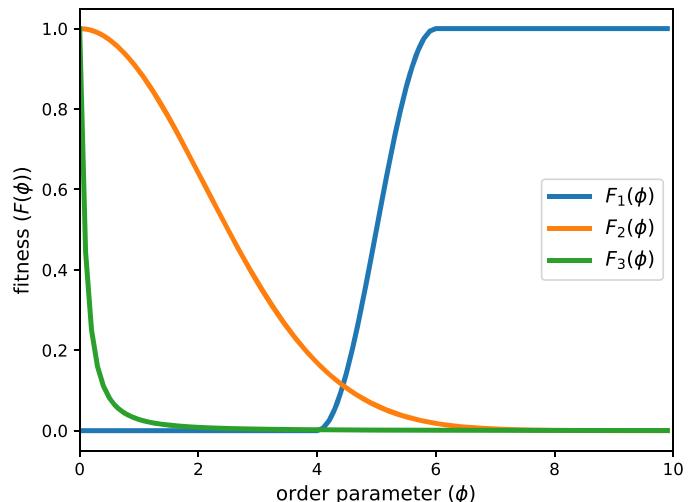


Fig. 7. Three different types of transfer functions with a codomain of [0,1]. A global single-objective fitness value can be defined as the multiplication of several partial fitness functions based on these transfer functions.

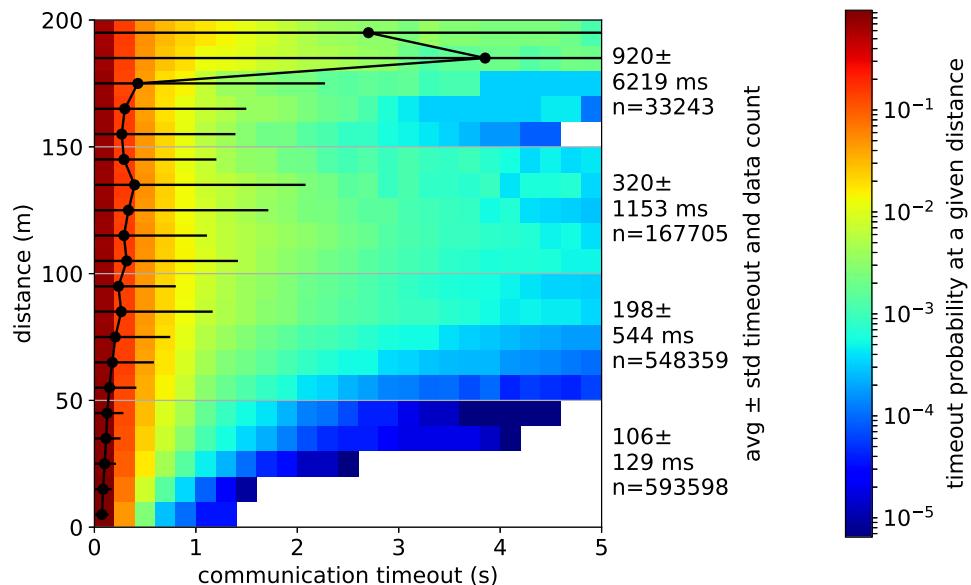


Fig. 8. Probability distribution of the communication outages as a function of distance. The database was gathered from a 5-min section of a general flight with 32 drones in a remote open-air setting. Each drone logged a 5-Hz sampling of the elapsed time since the reception of the last status packet from all other drones. This value [we call it timeout for simplicity but it actually also contains a small (<0.2 s) processing delay] was matched later with the position of the drones recorded accurately by each drone onboard. The distribution shows logs from all drones (1,349,490 data points in total), and it is normalized for each row (distance) separately. Color indicates timeout probability in each bin for a given distance. Average timeouts with SD and with the number of data points are indicated on the right for 50-m distance binning, whereas the black line on the plot indicates average and SD of timeout for each distance bin of 10 m. Database is very sparse and thus less accurate above 150 m, but as a general tendency, communication is most stable between close-by drones, whereas outages were more frequent and longer with increasing distance.

frequency of $10 + 1$ status packets sent per second per UAV, typically only around 40 to 80% of them were received by close-by peers on average in a general flight with 30 drones, where the reception rate was dependent on many factors, such as the bandwidth usage of other nearby wireless solutions or the details of the surrounding environment. Because signal power decays quickly with space [ideally at $1/r^2$ in open space (57)], reception from nearby agents (<20 m) was nearly perfect, whereas large communication outages started to occur at larger distances (>50 m). As an illustration of the actual communication characteristics, a detailed log of the distance dependence of the reception quality from a 5-min sample flight with 32 drones can be seen in Fig. 8. Note that the observed spatial decay fits naturally into our targeted distributed communication approach for two reasons: (i) the most critical information comes from the closest agents, which is always the most reliable, and (ii) the communication network naturally becomes scalable with flock size as the bandwidth overlap decays with distance.

There is a trivial reality gap in the communication aspects: Simulations contain constant delay and communication range, whereas the real setup was more stochastic and distance-dependent. Furthermore, in the real setup, we compensated for communication outages to some extent with linearly extrapolating neighboring drone positions using the global time stamps and velocity. With this approach, our overall aim was to have the safest real system possible and an underestimated communication quality through the model design phase to be prepared for a worst-case communication scenario. As a result, with this setup, we did not experience any mission-critical communication outage so far.

SUPPLEMENTARY MATERIALS

robotics.sciencemag.org/cgi/content/full/3/20/eaat3536/DC1

Movie S1. Simulation of the old flocking model (algorithm A) with 100 agents.

Movie S2. Simulation of the new flocking model (algorithm B) after evolutionary optimization with 100 agents.

Movie S3. Simulation of flocking for different speeds (4 to 32 m/s), flock sizes (30 to 1000 agents), and scenarios.

Movie S4. Flight log visualization of 30 drones at 4 m/s in a diagonal flight pattern.

Movie S5. Flight log visualization of 30 drones at 6 m/s with obstacles.

Movie S6. Flight log visualization of 30 drones at 8 m/s in a circular flight pattern.

Movie S7. Summarizing documentary with simulation, flight log visualization, and footage on real flights.

Table S1. Optimized model parameter values and working ranges in simulation.

Table S2. Statistic evaluation of optimized simulations.

Table S3. Explanation of flocking model parameters.

Table S4. Model parameter values used on real drones.

Table S5. Environmental parameters of the realistic setup.

Table S6. Parameter settings of the evolutionary optimization.

REFERENCES AND NOTES

- P. DeLellis, G. Polverino, G. Ustuner, N. Abaid, S. Macri, E. M. Bolt, M. Porfiri, Collective behaviour across animal species. *Sci. Rep.* **4**, 3723 (2014).
- J. Buhl, D. J. T. Sumpter, I. D. Couzin, J. J. Hale, E. Despland, E. R. Miller, S. J. Simpson, From disorder to order in marching locusts. *Science* **312**, 1402–1406 (2006).
- A. Attanasi, A. Cavagna, L. Del Castello, I. Giardina, T. S. Grigera, A. Jelić, S. Melillo, L. Parisi, O. Pohl, E. Shen, M. Viale, Information transfer and behavioural inertia in starling flocks. *Nat. Phys.* **10**, 615–698 (2014).
- M. Ballerini, N. Cabibbo, R. Candelier, A. Cavagna, E. Cisbani, I. Giardina, V. Lecomte, A. Orlandi, G. Parisi, A. Procaccini, M. Viale, V. Zdravkovic, Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proc. Natl. Acad. Sci. U.S.A.* **105**, 1232–1237 (2008).
- U. Lopez, J. Gautrais, I. D. Couzin, G. Theraulaz, From behavioural analyses to models of collective motion in fish schools. *Interface Focus* **2**, 693–707 (2012).
- H. Hildenbrandt, C. Carere, C. K. Hemelrijk, Self-organized aerial displays of thousands of starlings: A model. *Behav. Ecol.* **21**, 1349–1359 (2010).
- C. W. Reynolds, Flocks, herds, and schools: A distributed behavioral model. *ACM SIGGRAPH Comput. Graph.* **21**, 25–34 (1987).
- T. Vicsek, A. Zafeiris, Collective motion. *Phys. Rep.* **517**, 71–140 (2012).
- B. T. Fine, D. A. Shell, Unifying microscopic flocking motion models for virtual, robotic, and biological flock members. *Auton. Robots* **35**, 195–219 (2013).
- E. Méhes, T. Vicsek, Collective motion of cells: From experiments to models. *Integr. Biol.* **6**, 831–854 (2014).
- M. Brambilla, E. Ferrante, M. Birattari, M. Dorigo, Swarm robotics: A review from the swarm engineering perspective. *Swarm Intell.* **7**, 1–41 (2013).
- D. Floreano, R. J. Wood, Science, technology and the future of small autonomous drones. *Nature* **521**, 460–6 (2015).
- International Organization for Standardization (ISO), *ISO 8373:2012, Robots and Robotic Devices—Vocabulary* (ISO, 2012).
- A. E. Turgut, H. Çelikkannat, F. Gökçe, E. Şahin, Self-organized flocking in mobile robot swarms. *Swarm Intell.* **2**, 97–120 (2008).
- F. Gökçe, E. Şahin, To flock or not to flock: The pros and cons of flocking in long-range “migration” of mobile robot swarms, in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1* (International Foundation for Autonomous Agents and Multiagent Systems, 2009), pp. 65–72.
- D. Helbing, I. Farkas, T. Vicsek, Simulating dynamical features of escape panic. *Nature* **407**, 487–490 (2000).
- B. Liu, T. Chu, L. Wang, Stability and oscillation of swarm with interaction time delays, in *2007 American Control Conference (ACC'07)* (IEEE, 2007), pp. 4600–4605.
- E. Forgoston, I. B. Schwartz, Delay-induced instabilities in self-propelling swarms. *Phys. Rev. E* **77**, 035203 (2008).
- Z. Yang, Q. Zhang, Z. Jiang, Z. Chen, Flocking of multi-agents with time delay. *Int. J. Syst. Sci.* **43**, 2125–2134 (2012).
- B. Lindley, L. Mier-y Teran-Romero, I. B. Schwartz, Noise induced pattern switching in randomly distributed delayed swarms, in *2013 American Control Conference (ACC'13)* (IEEE, 2013), pp. 4587–4591.
- S. K. Armah, S. Yi, Analysis of time delays in quadrotor systems and design of control, in *Time Delay Systems* (Springer, 2017), pp. 299–313.
- R. Olfati-Saber, Flocking for multi-agent dynamic systems: Algorithms and theory. *IEEE Trans. Autom. Contr.* **51**, 401–420 (2006).
- R. Olfati-Saber, R. M. Murray, Consensus problems in networks of agents with switching topology and time-delays. *IEEE Trans. Autom. Contr.* **49**, 1520–1533 (2004).
- G. Vásárhelyi, Cs. Virág, G. Somorjai, N. Tarcai, T. Szörényi, T. Vicsek, Outdoor flocking and formation flight with autonomous aerial robots, in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)* (IEEE, 2014), pp. 3866–3873.
- K. Kaplan, “500 drones light night sky to set record,” *iQ by Intel*, 4 November 2016; <https://iq.intel.com/500-drones-light-show-sets-record/>.
- Ehang, China; www.ehang.com/formation/.
- M. Waibel, B. Keays, F. Augugliaro, “Drone shows: Creative potential and best practices,” *Verity Studios*, January 2017; <http://veritystudios.com/whitepaper/>.
- “Department of Defense Announces Successful Micro-Drone Demonstration,” *U.S. Department of Defense*, 9 January 2017; www.defense.gov/News/News-Releases/News-Release-View/Article/1044811/department-of-defense-announces-successful-micro-drone-demonstration.
- A. Kushleyev, D. Mellinger, C. Powers, V. Kumar, Towards a swarm of agile micro quadrotors. *Auton. Robots* **35**, 287–300 (2013).
- J. A. Preiss, W. Honig, G. S. Sukhatme, N. Ayanian, CrazySwarm: A large nano-quadcopter swarm, in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2017), pp. 3299–3304.
- S. Hauert, S. Leven, M. Varga, F. Ruini, A. Cangelosi, J.-C. Zufferey, D. Floreano, Reynolds flocking in reality with fixed-wing robots: Communication range vs. maximum turning rate, in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2011), pp. 5015–5020.
- A. Pospischil, “Mapping ad hoc communications network of a large number fixed-wing uav swarm,” thesis, Naval Postgraduate School, Monterey, California (2017).
- R. G. Braga, R. C. da Silva, A. C. Ramos, F. Mora-Camino, Collision avoidance based on Reynolds rules: A case study using quadrotors, in *Information Technology-New Generations* (Springer, 2018), pp. 773–780.
- S. A. Quintero, G. E. Collins, J. P. Hespanha, Flocking with fixed-wing UAVs for distributed sensing: A stochastic optimal control approach, in *2013 American Control Conference* (IEEE, 2013), pp. 2025–2031.
- J. L. Sanchez-Lopez, J. Pestana, P. de la Puente, R. Suarez-Fernandez, P. Campoy, A system for the design and development of vision-based multi-robot quadrotor swarms, in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)* (IEEE, 2014), pp. 640–648.
- A. E. Eiben, J. Smith, From evolutionary computation to the evolution of things. *Nature* **521**, 476–482 (2015).
- A. E. Eiben, J. E. Smith, *Introduction to Evolutionary Computing* (Springer, ed. 2, 2015).

38. N. Hansen, A. Ostermeier, Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.* **9**, 159–195 (2001).
39. Atlasz Supercomputer Cluster Service; <http://hpc.iig.elte.hu/>.
40. C. Virág, G. Vásárhelyi, N. Tarcai, T. Szörényi, G. Somorjai, T. Nepusz, T. Vicsek, Flocking algorithm for autonomous flying robots. *Bioinspir. Biomim.* **9**, 025012 (2014).
41. C. J. Pennycuick, S. Åkesson, A. Hedenstrom, Air speeds of migrating birds observed by ornithodolite and compared with predictions from flight theory. *J. R. Soc. Interface* **10**, 20130419 (2013).
42. A. Bottinelli, D. T. J. Sumpter, J. L. Silverberg, Emergent structural mechanisms for high-density collective motion inspired by human crowds. *Phys. Rev. Lett.* **117**, 228301 (2016).
43. N. Tarcai, C. Virág, D. Ábel, M. Nagy, P. L. Várkonyi, G. Vásárhelyi, T. Vicsek, Patterns, transitions and the role of leaders in the collective dynamics of a simple robotic flock. *J. Stat. Mech.* **2011**, P04010 (2011).
44. T. Vicsek, Universal patterns of collective motion from minimal models of flocking, in *2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems* (IEEE, 2008), pp. 3–11.
45. J. Deseigne, S. Léonard, O. Dauchot, H. Chaté, Vibrated polar disks: Spontaneous motion, binary collisions, and collective dynamics. *Soft Matter* **8**, 5629–5639 (2012).
46. I. J. Farkas, T. Vicsek, Patterns in the collective behavior of humans. *AIP Conf. Proc.* **779**, 1–15 (2005).
47. J. L. Silverberg, M. Bierbaum, J. P. Sethna, I. Cohen, Collective motion of humans in mosh and circle pits at heavy metal concerts. *Phys. Rev. Lett.* **110**, 228701 (2013).
48. C. Virág, M. Nagy, C. Gershenson, G. Vásárhelyi, Self-organized UAV traffic in realistic environments, in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1645–1652.
49. F. Cucker, S. Smale, Emergent behavior in flocks. *IEEE Trans. Automat. Contr.* **52**, 852–862 (2007).
50. S.-Y. Ha, T. Ha, J.-H. Kim, Emergent behavior of a cucker-smale type particle model with nonlinear velocity couplings. *IEEE Trans. Automat. Contr.* **55**, 1679–1683 (2010).
51. L. Meier, P. Tanskanen, F. Fraundorfer, M. Pollefeys, PIXHAWK: A system for autonomous flight using onboard computer vision, in *2011 IEEE International Conference on Robotics and Automation* (IEEE, 2011), pp. 2992–2997.
52. J. Han, M. Li, L. Guo, Soft control on collective behavior of a group of autonomous agents by a shill agent. *J. Syst. Sci. Complex* **19**, 54–62 (2006).
53. N. Hansen, S. D. Müller, P. Koumoutsakos, Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evol. Comput.* **11**, 1–18 (2003).
54. CMA-ES Python Library; <https://github.com/CMA-ES/pycma>.
55. Pixhawk AutoPilot; <https://pixhawk.org>.
56. CollMot Branch of the ArduCopter codebase on GitHub; <https://github.com/collmot/ardupilot/tree/CMCCopter-3.4>.
57. S. Hayat, E. Yanmaz, R. Muzaffar, Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint. *IEEE Commun. Surv. Tut.* **18**, 2624–2661 (2016).

Acknowledgments: The long-exposure photo was taken by Zs. Bézsényi. We are grateful for the help of other members of the robotics team, namely, B. Badár, I. Donkó, and N. Gál, who are not present here as authors, but have made valuable contribution to make our drones fly. **Funding:** This work was partly supported by the following grants: U.S. Air Force grant no. FA9550-17-1-0037, K_16 Research Grant of the Hungarian National Research, Development and Innovation Office (K 119467), and János Bolyai Research Scholarship of the Hungarian Academy of Sciences (BO/00219/15/6). **Author contributions:** The project was conceived by T.V. Cs.V., G.V., and T.V. developed the concept and algorithms. Cs.V. and G.V. developed the simulation. T.N. and G.V. developed the software for drones. G.S. and G.V. developed the drone hardware. A.E.E. and G.V. implemented the evolutionary optimization framework. T.N., G.S., Cs.V., and G.V. performed drone experiments. The paper was written by A.E.E., Cs.V., G.V., and T.V. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** The project website is <http://hal.elte.hu/drones/>. Flight logs related to the results presented are available at <https://doi.org/10.5061/dryad.mq85r61>. Source code of our robotic simulation framework is available at <https://github.com/csviragh/roboticsim>. All other data needed to evaluate the conclusions in the paper are present in the paper or the Supplementary Materials.

Submitted 17 February 2018

Accepted 1 June 2018

Published 18 July 2018

10.1126/scirobotics.aat3536

Citation: G. Vásárhelyi, Cs. Virág, G. Somorjai, T. Nepusz, A. E. Eiben, T. Vicsek, Optimized flocking of autonomous drones in confined environments. *Sci. Robot.* **3**, eaat3536 (2018).

Optimized flocking of autonomous drones in confined environments

Gábor Vásárhelyi, Csaba Virág, Gergo Somorjai, Tamás Nepusz, Agoston E. Eiben and Tamás Vicsek

Sci. Robotics **3**, eaat3536.
DOI: 10.1126/scirobotics.aat3536

ARTICLE TOOLS

<http://robotics.science.org/content/3/20/eaat3536>

SUPPLEMENTARY MATERIALS

<http://robotics.science.org/content/suppl/2018/07/16/3.20.eaat3536.DC1>

REFERENCES

This article cites 32 articles, 4 of which you can access for free
<http://robotics.science.org/content/3/20/eaat3536#BIBL>

PERMISSIONS

<http://www.sciencemag.org/help/reprints-and-permissions>

Use of this article is subject to the [Terms of Service](#)

Science Robotics (ISSN 2470-9476) is published by the American Association for the Advancement of Science, 1200 New York Avenue NW, Washington, DC 20005. 2017 © The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works. The title *Science Robotics* is a registered trademark of AAAS.

Supplementary Material for Optimized flocking of autonomous drones in confined environments

Gábor Vásárhelyi,^{1,2*} Csaba Virágħ,² Gergő Somorjai,^{1,2}
Tamás Nepusz,³ Agoston E. Eiben,⁴ Tamás Vicsek^{1,2}

¹MTA-ELTE Statistical and Biological Physics Research Group,
Pázmány Péter sétány 1/A, 1117 Budapest, Hungary

²Eötvös Loránd University, Department of Biological Physics,
Pázmány Péter sétány 1/A, 1117 Budapest, Hungary

³Molde University College,
Britvegen 2, 6410 Molde, Norway

⁴Vrije Universiteit Amsterdam, Department of Computer Science,
de Boelelaan 1081a, 1181HV Amsterdam, The Netherlands

*To whom correspondence should be addressed; E-mail: vasarhelyi@hal.elte.hu

Supplementary Movies

- Movie S1. Simulation of old flocking model (algorithm A) with 100 agents, available at
<https://youtu.be/viEfowBXzho>
- Movie S2. Simulation of new flocking model (algorithm B) after evolutionary optimization with 100 agents, available at <https://youtu.be/t8kr79k3DUQ>
- Movie S3. Simulation of flocking for different speeds (4-32 m/s), flock sizes (30-1000 agents) and scenarios, available at <https://youtu.be/KPVfi9Pwuq8>
- Movie S4. Flight log visualization of 30 drones at 4 m/s in a diagonal flight pattern, available at <https://youtu.be/JMMGIQm7Ris>
- Movie S5. Flight log visualization of 30 drones at 6 m/s with obstacles, available at
<https://youtu.be/YW5zDD70x8o>
- Movie S6. Flight log visualization of 30 drones at 8 m/s in a circular flight pattern, available at <https://youtu.be/GoiunzowSG4>
- Movie S7. Summarizing documentary with simulation, flight log visualization and footage on real flights, available at <https://youtu.be/loemTAEGWfg>

Supplementary Tables

Table S1: Optimized model parameter values and working ranges in simulation. F^{\max} represents the highest achieved fitness values during optimization, while \bar{F} shows average (\pm std) fitness values calculated from 100 stochastic evaluations of the optimized solutions. Optimized model parameter values are shown below. Numbers in brackets indicate the range where solutions existed during the optimization with fitness above \bar{F} , representing a reasonable working range of the parameter space. The number of these high-fitness evaluations (out of 150000) is shown by N . Note that as r_0^{frict} is optimized to a high value, the value of p^{frict} and a^{frict} has no effect.

param	unit	$v^{\text{flock}} = 4 \text{ m/s}$	$v^{\text{flock}} = 6 \text{ m/s}$	$v^{\text{flock}} = 8 \text{ m/s}$	$v^{\text{flock}} = 16 \text{ m/s}$	$v^{\text{flock}} = 32 \text{ m/s}$
F^{\max}	-	0.918	0.873	0.798	0.910	0.892
\bar{F}	-	0.812 ± 0.101	0.776 ± 0.086	0.728 ± 0.075	0.789 ± 0.120	0.629 ± 0.227
$N(F \geq \bar{F})$	-	1835	2118	2030	2515	4364
Rep.	r_0^{rep}	m	25.6 [17.3, 28.2]	41.1 [30.8, 51.0]	52.9 [46.0, 63.5]	103 [85.7, 120]
	p^{rep}	1/s	0.13 [0.08, 0.37]	0.07 [0.03, 0.16]	0.06 [0.04, 0.09]	0.03 [0.02, 0.06]
Friction	r_0^{frict}	m	85.3 [51.3, 87.8]	88.5 [58.5, 100]	86.8 [52.2, 114]	182 [95.6, 198]
	C^{frict}	-	0.05 [0.02, 0.27]	0.05 [0.03, 0.22]	0.06 [0.04, 0.17]	0.07 [0.04, 0.23]
	v^{frict}	m/s	0.63 [0.07, 1.88]	0.93 [0.30, 2.70]	1.48 [0.81, 3.10]	2.47 [1.60, 4.99]
	p^{frict}	1/s	3.20 [0.0, 10.0]	5.32 [0.38, 9.67]	6.87 [1.66, 9.97]	5.92 [0.02, 7.72]
	a^{frict}	m/s/s	4.16 [0.0, 10.0]	9.94 [5.04, 10.0]	0.98 [0.0, 2.89]	2.57 [0.0, 5.25]
Wall	r_0^{shill}	m	0.3 [0.0, 15.1]	0.5 [0.0, 9.3]	0.1 [0.0, 3.5]	0.5 [0.0, 14.2]
	v^{shill}	m/s	13.6 [9.4, 15.0]	19.7 [17.5, 20.0]	22.8 [17.8, 34.1]	49.2 [39.2, 56.3]
	p^{shill}	1/s	3.55 [0.26, 10.0]	5.44 [0.48, 9.96]	8.12 [4.32, 10.0]	9.49 [3.42, 10.0]
	a^{shill}	m/s/s	3.02 [1.24, 9.99]	3.54 [2.53, 5.12]	4.18 [2.39, 9.99]	9.50 [6.72, 10.0]

Table S2: Statistic evaluation of optimized simulations. Table contains average and standard deviation of measured order parameters (average speed, average distance from closest neighbor, collision risk and number of collisions) and fitness values at model optima. Values are gathered from 100 stochastic evaluations of the simulations for all tested flocking speeds. Other conditions of the simulations, like arena size and communication range are also given for each case.

v^{flock}	4 m/s	6 m/s	8 m/s	16 m/s	32 m/s
L	250 m	250 m	250 m	500 m	1000 m
r^C	80 m	80 m	80 m	160 m	320 m
$\phi^{\text{vel}} \text{ (m/s)}$	3.78 ± 0.10	5.60 ± 0.02	7.26 ± 0.02	15.01 ± 0.04	29.6 ± 0.15
$\bar{r}_{ij}^{\min} \text{ (m)}$	12.2 ± 0.4	13.4 ± 0.06	14.7 ± 0.07	23.2 ± 0.23	41.7 ± 0.70
$\Phi^{\text{coll}} \text{ (E-6)}$	0.94 ± 1.3	1.6 ± 1.9	1.0 ± 1.8	2.1 ± 2.8	6.0 ± 8.1
N^{coll}	1.18 ± 1.25	1.65 ± 1.34	0.89 ± 1.19	1.57 ± 2.08	3.53 ± 3.61
F^{disc}	1 ± 0				
F^{coll}	0.945 ± 0.071	0.911 ± 0.098	0.943 ± 0.094	0.892 ± 0.133	0.769 ± 0.235
F^{wall}	0.997 ± 0.008	1 ± 0	0.997 ± 0.018	0.998 ± 0.015	0.941 ± 0.171
F^{cluster}	1 ± 0				
F^{corr}	0.916 ± 0.015	0.923 ± 0.003	0.904 ± 0.005	0.949 ± 0.003	0.957 ± 0.003
F^{speed}	0.938 ± 0.062	0.923 ± 0.006	0.857 ± 0.007	0.934 ± 0.005	0.908 ± 0.012
F	0.812 ± 0.101	0.776 ± 0.086	0.728 ± 0.075	0.789 ± 0.120	0.629 ± 0.227

Table S3: Explanation of flocking model parameters

param		name and description
Repulsion	r_0^{rep}	Repulsion range. The distance at which local repulsion kicks in. Larger values create sparser flocks with less collision. Optimization tends to increase this above intuitive levels together with decreased gain value, resulting in a smoother and spatially extended repulsive interaction. Suggested value scales linearly with flocking speed according to the optimization.
	p^{rep}	Repulsion gain. The strength of repulsion (analogous to the spring constant of the half-spring repulsion model). Higher values result in more emphasized repulsion which acts against collisions in free space but also induces oscillations in confined spaces. Should be tuned together with repulsion range.
Friction	r_0^{frict}	Stopping point offset of alignment. The distance of the stopping point in front of agents according to the optimal velocity reduction curve. Below this value alignment reduces all velocity difference above the given small velocity slack threshold. Optimization tends to increase this value above intuitive levels to maximize inter-agent alignment in the whole communication range without spatial dependence.
	C^{frict}	Coefficient of velocity alignment. Linear coefficient of the velocity difference error reduction in the velocity alignment term. Higher values create stronger damping between agents which helps reducing repulsion-induced oscillations but makes motion more sluggish. Optimization tends to decrease this value below intuitive levels.
	v^{frict}	Velocity slack of alignment. This parameter sets the velocity difference level agents are allowed to have at all times. Having a non-zero value reduces local overdamped dynamics a bit and thus increases the information spreading capability within the flock during turns, which results in increased flock-level agility around obstacles and at walls. Some velocity slack also helps eliminating roll and pitch oscillations arising in real systems due to strong alignment and the delayed response between tilting and velocity change. Should be kept at small levels as too large values disable alignment completely.
	p^{frict}	Gain of braking curve. Linear gain of the optimal braking curve used in determining the maximal allowed velocity difference. Note that linearity is only expressed in the v-x plane, while the parameter has a nonlinear effect in behaviour. Large values approximate the braking curve to the curve of constant acceleration. Small values elongate the final part of braking (at small speeds) with decreasing acceleration and smoother stops.
	a^{frict}	Acceleration of braking curve. The maximal allowed acceleration in the optimal braking curve used for determining the maximal allowed velocity difference between agents. Higher values assume that agents can brake quicker and thus make alignment more local. Too high values result in the inability of agents to react to too large velocity differences in time and thus lead to collisions.

param		name and description
Wall	r_0^{shill}	Stopping point offset of walls. This parameter is the same as r_0^{frict} but for wall alignment interactions. Larger values make agents start to brake at larger distances from walls.
	v^{shill}	Velocity of virtual shill agents. Wall alignment uses shill agents with this velocity magnitude. These virtual agents move inwards from walls and outwards from obstacles. Shill agent velocity is typically at least twice as large as the flocking speed as wall-agent interactions are not symmetric (shill agents do not align) thus agents need to take care of reducing the full velocity difference themselves. Larger values push agents away from walls with higher intensity and implicitly also at higher distances already. Suggested value scales linearly with flocking speed according to the optimization.
	p^{shill}	Gain of braking curve for walls. This parameter is the same as p^{frict} but for wall alignment interactions.
	a^{shill}	Acceleration of braking curve for walls. This parameter is the same as a^{frict} but for wall alignment interactions.
Spp	v^{flock}	Flocking speed. The preferred speed of the agents. Higher values in general require stronger repulsion and larger shill agent velocity.
	v^{max}	Maximum speed. The maximal allowed speed of the agents. Should be higher than flocking speed. If too high, cumulated interaction terms can speed up agents too much, which can be dangerous.

Table S4: Model parameter values used on real drones. Values are mostly within the optimal working ranges of the simulation with some notable exceptions (highlighted with bold face), mostly due to the cautious attitude necessary in real experiments. The changes from simulation optima are explained for each parameter in the last column. Values are shown for different flocking speeds in different columns.

	param	unit	4 m/s	6 m/s	8 m/s	comments on tuning
Rep.	r_0^{rep}	m	15	35	50	At 4 m/s system was safe enough to reduce repulsion range and get a tighter flock.
	p^{rep}	1/s	0.3	0.1	0.1	Gain was somewhat increased as a precaution to prevent collisions.
Friction	r_0^{frict}	m	45	65	80	Increased substantially to minimize possible oscillations.
	C^{frict}	-	0.5	0.5	0.5	
	v^{frict}	m/s	1	2	2	Some reasonable value was selected as optimization neglected this parameter with high r_0^{frict} .
	p^{frict}	1/s	1	1	1	
	a^{frict}	m/s/s	3	3	3	
Wall	L^{arena}	m	200	260	260	Increased distance from walls and obstacles as a precaution. Decreased shill velocity is enough when r_0^{shill} is higher, resulting in lower interaction magnitudes at walls.
	r_0^{shill}	m	0	5	30	
	v^{shill}	m/s	10	14	14	
	p^{shill}	1/s	2	2	2	
	a^{shill}	m/s/s	2.5	2.5	2.5	

Table S5: Environmental parameters of the realistic setup

parameter name	symbol	value	unit
communication delay	t^{del}	1	s
velocity relaxation time	τ^{CTRL}	1	s
maximal acceleration	a^{\max}	6	m/s^2
communication range	r^c	80	m
sensor update rate	t^s	0.2	s
sensor noise	σ^s	0.005	m^2/s^2
outer noise	σ	0.2	m^2/s^3

Table S6: Parameter settings of the evolutionary optimization. The used CMA-ES optimizer algorithm was obtained from <https://pypi.python.org/pypi/cma>). The most important parameter values are shown below. Parameters not listed here were used with default values.

parameter name (meaning)	value
population size	100
termination condition	150 generations
elitist survivor selection	False
variable mutation step sizes	True
dimension of parameter space	11