Pausing Makes Perfect: Intermittent Pauses for Resilient Swarming

Peleg Shefi[1], Amir Ayali[2], and Gal A. Kaminka[1]

[1] The MAVERICK Group, Computer Science Department, Bar Ilan University, 5290002 Ramat Gan, Israel
pelegshefi@hotmail.com,
galk@cs.biu.ac.il
[2] School of Zoology and Sagol school of Neuroscience, Tel Aviv University, Tel Aviv 6997801, Israel
ayali@tauex.tau.ac.il

Abstract. Collective motion is a widespread natural phenomenon in which individuals move in an ordered fashion, without centralized control. Such movement is often considered to be robust to individual failures, but common models of collective motion fail to display this trait. Inspired by observations of locusts, we introduce intermittent pauses in individual motion, where pause duration is tied to perception of neighbours' movements. Realistic physics-based simulations of swarm robots demonstrate that this leads to significant improvements in the resilience of the swarm, without sacrificing its functionality or coherence.

Keywords: Collective motion, Swarm, Pause and Go, Locust, Resilience

1 Introduction

Swarm collective motion is a widespread phenomenon in nature, found in many organisms, such as schools of fish [15], flocking of birds [10], marching of locusts [2, 3], and others. In these, ordered collective behavior emerges without centralized control; individuals act solely based on local information, through simple repeated interactions. Over the years, numerous models have been developed that replicate this phenomenon [9, 11, 14, 17, 19, 22, 28, 34]. Two popular modeling approaches are *Heading-Alignment* (e.g., [34]) where an agent tries to align speed and heading with its surrounding neighbors, and *Avoid-Attract* (e.g., [5, 11, 26]), where agents try to keep neighbors within preferred distances. These models demonstrated that group leadership, hierarchical control and global information are not necessary for collective behavior [9].

While the inspiration for collective motion is drawn from nature, many of the models are simplified for the sake of analysis. More often than not, these models assume agents are a dimensionless particle, moving with constant speed and in a convex-bounded environment free of obstacles. These unrealistic assumptions do not hold in nature, nor are they relevant to robots; animals and robots may be slowed down or even stop due to failures. Some swarm robotics applications require the robots to be low-cost by design, which exacerbates the rate of failure.

While swarms are often considered to be robust to individual motion failures, Winfield and Nembrini [35], and Bjerknes and Winfield [5] show otherwise: even a small portion of faulty agents in a swarm can break the desired collective behavior. Moreover, the identification of failing robots is a challenge in itself. In principle, ego-motion estimation of its own heading and speed allows a robot to estimate the velocity of its neighbors, and thus recognize robots that stop or are too slow to be considered. Such computation is also used in heading-alignment models [9, 28, 34]. However, robotic ego-motion estimation is highly susceptible to errors [6] and cannot be relied on without occasional calibration vis-a-vis the environment. Moreover, collective motion models that avoid relying on ego-motion estimation, such as avoid-attract models [11, 26], lack this mechanism for detecting failing neighbors, and thus face even greater difficulty.

In this paper, we present a simple method allowing agents to recognize their failing neighbors, without ego-motion estimation. Working with a hybrid locust-robot swarm [4] (Fig. 1), we were inspired by locust marching behavior; individual locusts intermittently change between moving and standing (so-called, *pause-and-go* or *intermittent* locomotion) [2, 3]. The pauses serve as decision points, and are affected by the complexity of the sensory input [1, 27]. The same type of motions are also evident in other swarming animals [18, 21, 24].

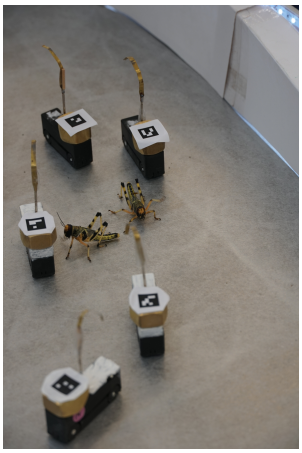


Fig. 1. Our hybrid swarm: A group of gregarious *locust nymphs* interacting with *nymbot* robots, in our lab.

We apply the intermittent locomotion mechanism in a simulated swarm of our robots, in the ARGoS 3 simulator [25], using an avoid-attract model [11, 26]. Each robot individually alternates between pauses and movement states. During a pause, the robot observes its neighbors and marks robots who are moving too slowly or stopping as faulty. It then moves, applying collective motion control selectively, only using robots it considers to be functioning. The cycle begins again (without memory of previous faulty robots).

The decision on movement and pause duration is not trivial, as robots that are pausing may be perceived by others to be failing, and thus the swarm coherence may deteriorate. We evaluate several pause termination criteria, tied to the perception of motion in neighbours, to compensate for this false detection of failing neighbors. Experiments demonstrate that applying pause-and-go using these criteria can maintain swarm coherence, despite significant percentage of failing members.

2 Related Work

Models of collective motion determine the motion of a robot, given its locally-perceived neighbors, without centralized control. Various types of failures in individuals can have catastrophic effects [5, 8, 33, 35], but the most devastating

failure is of a robot who stops permanently, while its active neighbors do not recognize it [5, 35], as its perceived velocity significantly affects their decisions. Christensen et al. [8] suggest a method inspired by fire-flies, where a blinking light is used as a heartbeat signal (*keepalive*) to signal an active robot. When a robot fails catastrophically, the signal stops and its neighbors ignore it. However, this mechanism is of little use in the case where motors fail, but the signal continues. Moreover, the mechanism assumes visual signaling and perception capabilities, potentially at greater energy expense. A more elaborate scheme is presented by Tarapore et al. [33], requiring communication between robots.

We focus on passive detection approaches, requiring no cooperation by the failing robot. We distinguish two families of collective motion models, based on their making (or not making) an assumption that each robot knows its own velocity vector (*ego-motion*). Both families exhibit poor performance when some individuals stop, but their potential for detecting failing robots is different.

Heading Alignment [9, 23, 28, 29, 34] models decide on the robots heading by aligning with the average orientation of surrounding neighbors. This requires either knowing self- and neighbors- global orientation (not available in local-perception simple robots), or the ability to calculate the relative heading based on estimated ego-motion (which is highly erroneous [6]). That said, having access to the ego-motion velocity vector allows detecting of robots that have stopped or slowed down, as their velocity can be computed from the relative velocity and the ego-motion velocity (see Sect. 3.2 for details).

The second type of models avoids ego-motion estimation, achieving ordered movement without explicit alignment [5, 11, 26, 30–32]. This removes the reliance on noisy ego-motion estimation. However, without it, these models cannot detect robots that have stopped or slowed down, as they can no longer compute relative velocity while moving. Among these, a few *Avoid-Attract* models have been implemented and tested on robots [5, 11], and serve as a basis for our work. In Bjerknes et al. [5] the short-range repulsion is manifested by moving away from too close neighbors. While, for long-range attraction, each robot times the duration since its last repel and if the value surpasses a threshold, the robot turns towards the group's center position. In Ferrante et al. [11] the avoid-attract uses a potential function that corresponds to a preferred distance to keep.

Detecting stopped robots is more easily done when the observer is itself motionless, as moving neighbors can be considered functioning. We therefore seek to adopt intermittent pausing (pause-and-go locomotion, P&G) as an approach to detecting stopped faulty robots. P&G regimes are found in locust marching bands [3], though the relation to ego-motion estimation in locust is unknown. *Burst and Coast* is a related natural phenomenon in fish locomotion [7, 16, 21]. Rimer et al. [29] applied a P&G locomotion in a Vicsek heading-alignment model [34]. Their work did not have any method of fault handling.

3 Collective Motion with Intermittent Pauses

We begin by presenting an avoid-attract model governing collective motion (Sect. 3.1). We then discuss how faulty robots affect it, and how they may be detected, and their effects nullified (Sect. 3.2).

3.1 Avoid-Attract Collective Motion: Naive Model

N robots of rectangular shape are each represented by a position vector $\mathbf{p_i^t}$. Their own ego-motion velocity vector $\mathbf{v_i^t}$ is unknown to them; their controls are open-loop and their global orientation and speed is not known. Each robot senses all robots around it, within a distance R_{sense}, and calculates its own force vector $\mathbf{f}$. It then translates this into desired linear and angular velocity. The vector $\mathbf{v_i^t}$ is updated using magnitude-dependent motion control (MDMC) [11]. Time is discretized into *ticks*, each a tenth of a second.

Avoid-Attract Force f. The avoid-attract model encodes the idea of robots preferring to keep a distance d from all robots they perceive. Figure 2 shows the focal rectangular robot centered and heading towards the top of the page. The inner circle shows the *Avoid* zone, limited by the desired distance d. The *Attract* zone is marked by the outer ring, limited by the sensing radius R_{sense}. Dark rectangles show robots outside the sensing range. Also shown: sensed robots (light gray) scattered in both zones, r_{ij} (distance to neighbor j) and the bearing angle.

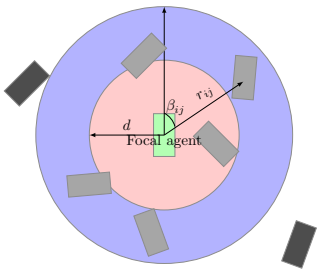


Fig. 2. Focal robot.

The focal robot avoids neighbors closer than the preferred range d, and moves towards neighbors that are farther. To do this, the robot uses Eq. 1 (inspired by Qi et al. [26]) to compute $\mathbf{f}$.

$$\mathbf{f} = \sum_{j=1}^{N_s} \gamma \frac{r_{ij} - d}{r_{ij}^2} \boldsymbol{\beta_{ij}} \tag{1}$$

There are N_s local neighbors, r_{ij} is the distance between the focal robot i and its j-th neighbor, while $\boldsymbol{\beta_{ij}}$ is the unit vector in the direction of neighbor j and γ is a gain parameter (see [26]). Like most previous studies [12], we assume the focal agent can differentiate between neighbors and calculate their distance. The decision to avoid or attract is handled by the fractional part ($\frac{r_{ij}-d}{r_{ij}^2}$): if the fraction is positive the force is applied in the direction of robot j; if the fraction is negative, then r_{ij} is smaller than d, and the force is applied in the opposite direction of neighbor j.

Motion Control. The force vector is broken into linear and angular speeds, using the magnitude-dependent motion control (MDMC) method of Ferrante et al. [11]. Let f_x, f_y be the projections of force $\mathbf{f}$ on the x and y axes of the focal robot body coordinate system (x axis along the length of the body), and U_{max} be the maximal speed. Then, the linear speed u and the angular speed ω of the robot are derived from $\mathbf{f}$ using Equation 2:

$$u = K_1 f_x + U, \qquad \omega = K_2 f_y \tag{2}$$

Here, K_1, K_2 are constants, and U is a forward-moving speed bias, set at half of U_{max} (see [11] for details of how these are set, and Table 1 for nomenclature). This leads the robot to move faster as the f_x is larger, while a larger f_y

leads to a quicker turn. To allow for real-world kinematic constraints on robot motion, u, ω may be truncated. In the experiments reported below, u was allowed in the range $[0, U_{max}]$ (i.e., no reverse motion); ω was allowed within the range $[-\omega_{max}, \omega_{max}]$, where ω_{max} is the maximal turn rate. Then, the robot's differential-drive controller converts u, ω into linear speeds for each wheel.

3.2 Detecting and Handling Faulty Agents

We now turn to examine faulty robots, which stop moving and remain in their last position and heading. Faulty robots do not indicate their status, and thus the naive model cannot distinguish faulty robots from others. As a result, applying the naive model when faulty robots are part of the swarm causes a breakdown in the coherence of the collective motion, as well as in its overall motion (as moving robots remain in the vicinity of the faulty robots). This breakdown is also evident in the experimental results (next section).

Detection of Faulty Agents. Since the robot's own velocity is unknown, it can not distinguish static faulty robots from functioning neighbor robots, other than by noting their motion ($|\mathbf{v_{neighbor}}| > 0$). To do this, normally the focal robot can compute the velocity vector of a neighbor by adding the focal robot's own velocity vector (its ego-motion vector), to the perceived relative velocity vector (Eq. 4).

$$\mathbf{v_{rel}} = \mathbf{v_{neighbor}} - \mathbf{v_{own}}, \tag{3}$$

and therefore,

$$\mathbf{v_{rel}} + \mathbf{v_{own}} = \mathbf{v_{neighbor}}. \tag{4}$$

As discussed, heading-alignment models assume that $\mathbf{v_{own}}$ is known, but this is a problematic assumption [6]. For the model we presented earlier (and variants which avoid the problematic assumption), $\mathbf{v_{own}}$ is not known, with the exception of one case, where $\mathbf{v_{own}}$ is set to $\mathbf{0}$, i.e., the focal robot stops. Then, any observed relative velocity vector ($|\mathbf{v_{rel}}| > 0$) is necessarily caused by the motion of the neighbor, which indicates that it is functioning.

Intermittent Pause and Go Locomotion. Inspired by the observed behavior of individual nymphs in locust marching bands, we require robots to intermittently select between *go* (moving) and *pause* (stopping) states. When a functioning robot is in the *go* state, it moves according to Eq. 5, which is a modified version of the force computation described earlier, which takes into account active moving robots (N_{sm}) and faulty robots (N_o). The duration of this state is randomly chosen from a uniform distribution.

$$\mathbf{f} = \sum_{j=1}^{N_{sm}} \gamma \frac{r_{ij} - d}{r_{ij}^2} \boldsymbol{\beta_{ij}} + \sum_{j=1}^{N_o} \mathbf{f_{obj}}, \tag{5}$$

where

$$\mathbf{f_{obj}} = \gamma \frac{r_{ij} - d}{r_{ij}^2} \theta(d - r_{ij}) \boldsymbol{\beta_{ij}}. \tag{6}$$

The first summation in Eq. 5 is over all sensed functioning neighbors (i.e., all but thought to be faulty neighbors). The second summation is of all the neighbors whom are considered faulty and induce a repel force. In Eq. 6 the $\theta(d - r_{ij})$ is a Heaviside step function (1 for 0 or positive, 0 otherwise), ensuring that only repulsion force is applied.

When the robot is in a *pause* state, it does not move. Instead, it continues to sense its surrounding neighbors, and notes those that are moving as active. Tracking active neighbors to detect movement requires short term memory, a requirement feasible both in robots and in nature [13].

The selection of states is done individually, and for a duration determined by a mechanism which we describe below. It is designed so that stochastically, at any given time, some of the N robots are in a pause state, while the rest are in a go state. Figure 3 illustrates the timeline for a small swarm, where one robot is failing. Others may be mistakenly believed to be failing, because they are in an intermittent pause state when they are being observed by others.

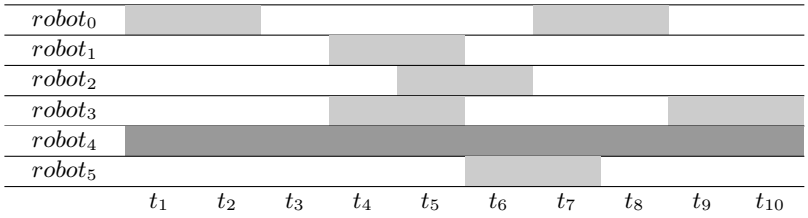


Fig. 3. Example timeline. Each row is a the timeline of a swarm member. Grayed out cells are pause state time-steps, while white cells are go state time-steps. $robot_1$ and $robot_3$ will consider each other faulty after the pause state in t_4–t_5 as they overlap. $robot_4$ is faulty (marked dark gray), hence will be detected correctly as faulty by all.

Misclassification of faulty robots. Naively, one may consider the pause-and-go mechanism to be a simple and elegant solution to the problem of detecting (and avoiding) faulty robots. However, as Fig. 3 demonstrates, deciding on motion and pause duration is critical to its success, and is not trivial. Longer duration of the go state allow the robots to move farther, and more opportunities for others in pause state to correctly identify moving robots as functioning. However, as faulty robots are not detected during movement, their negative effects are exacerbated during movements of longer duration. In contrast, longer pauses allow more opportunities for robots to identify others as faulty or functioning (giving them a higher chance to switch from a pause to a go), but also causes others to misidentify the pausing robot as faulty.

We therefore require a mechanism for determining the length of pauses. At minimum, the pause is at least two (2) ticks long, so that there is an opportunity to indicate a change in two subsequent sensor readings, indicating motion. Also, if all observed robots move, then obviously the pause can be terminated. But given that not all robots may be moving, any early termination of the pause

state may risk misclassifying a functioning robot as a faulty, which may disrupt the collective motion, as it will be avoided in the subsequent go state.

Aidan et al. [1] report that in locust, pause duration is affected by the complexity of the sensorial input. Inspired by this observation, we proposed a majority rule for early termination of the pause state. Thus the pause state terminates either when a parameterized duration has been reached, or when at least X percent of observed robots are moving. Next, we report experiments with different pause termination rules.

4 Experiments and Results

We first introduce the experimental environment and settings (Sect. 4.1). Then we report on the results and their implications (Sect. 4.2).

4.1 Experimental setup

To evaluate the P&G model, we use the ARGoS 3 platform [25], a physics-based robot simulator, to simulate the *Nymbots* of our hybrid swarm. These are rectangular two-wheeled differential-drive robots; their size, maximum speed, and turn kinematics are designed to approximate locust nymphs.

We compare the *naive* model (Sect. 3.1) with different variants on of the model with P&G motions (Sect. 3.2). Table 1 shows the model parameters and their values. Rows with multiple values show all values used in the experiments; values in bold denote defaults. The sensing range approximates locust visual range, and d is set based on the recommendation in [11]. The robots are deployed in a $0.5 \times 0.5\ m^2$ area in random position and random headings. The duration of the experiment is 8000 ticks (equivalent to 800 seconds).

The expectation is that the collective motion model achieves ordered (aligned headings), as quickly as possible. We use the *Order* metric presented in Vicsek et al. [34] and used in many collective motion studies (Eq. 7). It is the absolute value of the average normalized velocity, measuring the global alignment: when the group is completely aligned the order will be 1 while for a disordered group the value will near 0. The calculation ignores faulty robots.

$$\psi = \frac{1}{N} \left| \sum_{i=1}^{N} \mathbf{v_i} \right| \tag{7}$$

We typically show the evolution of the order metric with time. In all the graphs below, the solid lines show means over 25 trials, and the lightly-shaded envelope around it marks the standard error of the mean. All model variations were tested in 7 different swarm densities (by changing the swarm size, see N values in table 1). For each model and swarm density, five proportions of faulty robots (see N_f values in table 1) were tested with ($\frac{1}{N}$ is always one faulty robot). Faulty agents remain faulty for the entire experiment, starting from tick 0.

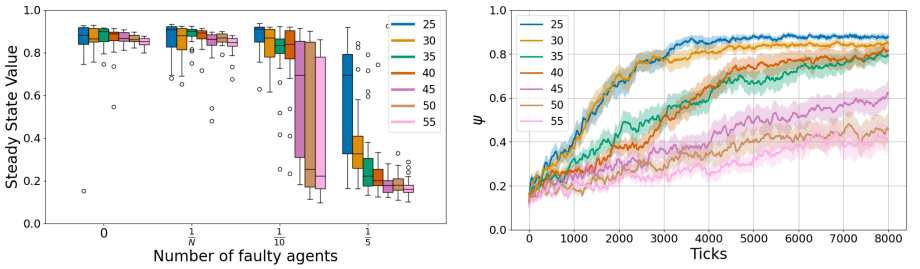
We additionally utilize the *steady-state value* (SSV) metric, presented in Ferrante et al. [11], and defined as the average value of the order metric in the last 10% frames (800 ticks in our case).

Parameter	Explanation	Values tested [units]
Body length	Robots length	0.05 [m]
Body width	Robots width	0.02 [m]
K_1	MDMC linear gain	0.5
K_2	MDMC angular gain	0.06
N	Number of robots	{25, 30, 35, 40, 45, 50, 55}
N_p	Proportion of initial pause robots	$\frac{1}{3}$
N_f	Proportion of failed robots	$\{0, \frac{1}{N}, \frac{1}{10}, \frac{1}{5}\}$
R_{sense}	Sensing radius	0.19 [m]
d	Wanted distance	0.11 [m]
γ	force gain	1.3
U_{max}	Maximum speed	0.035 [m/s]
ω_{lim}	Maximum turn rate	30 [Deg/s]
Pause duration	Pause state duration	{0.2, **0.5**, 0.7} [s]
Go duration	Go state duration range	{2-12, 2-22, **12-22**}[ticks]
T	Experiment duration	800 [s]
Tick	Tick duration	$\frac{1}{10}$ [s]
Trials	Number of runs per setting	25

Table 1. Nomenclature and robot parameters, and values. Defaults in **bold**.

4.2 Results

We first experimented with the naive avoid-attract model with different swarm sizes and proportions of faulty robots. In Fig. 4 the SSV and evolving order of the naive model are plotted. In Fig. 4(a) there is a comparison of the SSV of different swarm sizes with changing faulty proportions. As seen it drops dramatically with the growth of swarm size and faulty robots. The faulty robots trap active (functioning) robots, that stay in their vicinity, circling around them. As the number of failing robots grow in number, their *"gravitational pull"* increases until the swarm is anchored and unable to escape.



(a) SSV of different swarm sizes, at different faulty robot proportions.

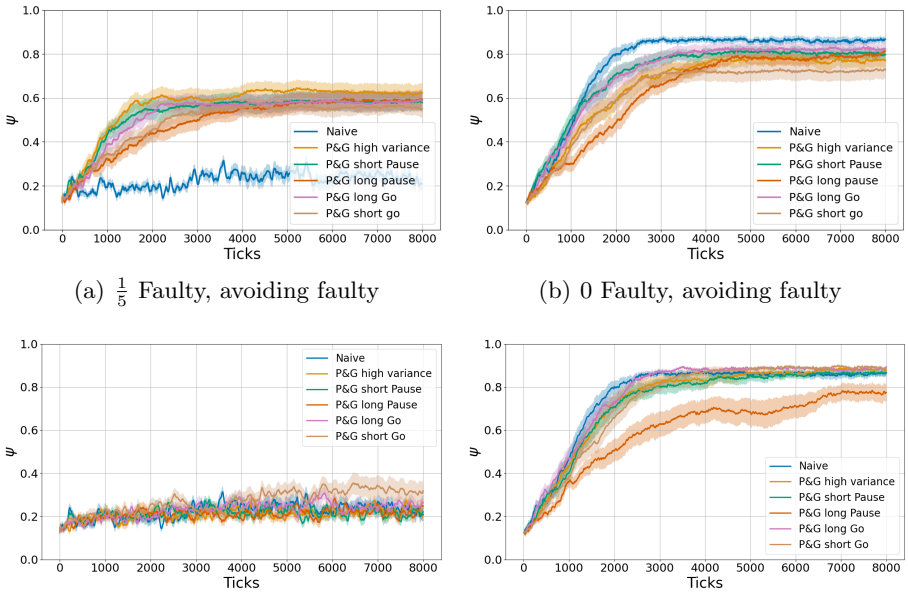
(b) Evolution of order over time, with $\frac{1}{10}$ of its robots being faulty

Fig. 4. The steady state value (a) and evolving order (b) of the naive model.

Interestingly, different swarm densities break down (become trapped) with different proportion of faulty robots. At the lower density of 25 robots in the

area, the swarm can slowly escape even from $\frac{1}{5}$ of faulty robots, while the higher density of 55 is already affected by 1 faulty robot. This is seen in Fig. 4(b) showing different swarm sizes, all with $\frac{1}{10}$ faulty robots. As the swarm gets larger (and hence denser) the swarm clearly does not converge to an ordered motion within the experiment duration.

To combat the deterioration in the swarm coherence, we evaluate several P&G variants, each with a different pause and go regime. In each regime, the pause duration is fix and the go duration is uniformly chosen from a specific range: *high variance* (P: 5 ticks, G: 2-22 ticks), *short pause* (P: 3 ticks, G: 2-22 ticks), *long pause* (P: 7 ticks, G: 2-22 ticks), *short go* (P: 5 ticks, G: 2-12 ticks) and *long go* (P: 5 ticks, G: 10-12 ticks). Figure 5(a) shows promising results. All P&G variants manage to converge to some order, though a fifth of the robots are faulty. In comparison, the naive model clearly fails to synthesize ordered motion. However, this comes at a cost. Figure 5(b) shows that compared to the naive model, P&G variants clearly under-perform when no faulty robots exist. The rate of misclassification is sufficiently high to cause noticeable decline in order and convergence rate, compared to the naive model.



(a) $\frac{1}{5}$ Faulty, avoiding faulty

(b) 0 Faulty, avoiding faulty

(c) $\frac{1}{5}$ Faulty, not avoiding faulty

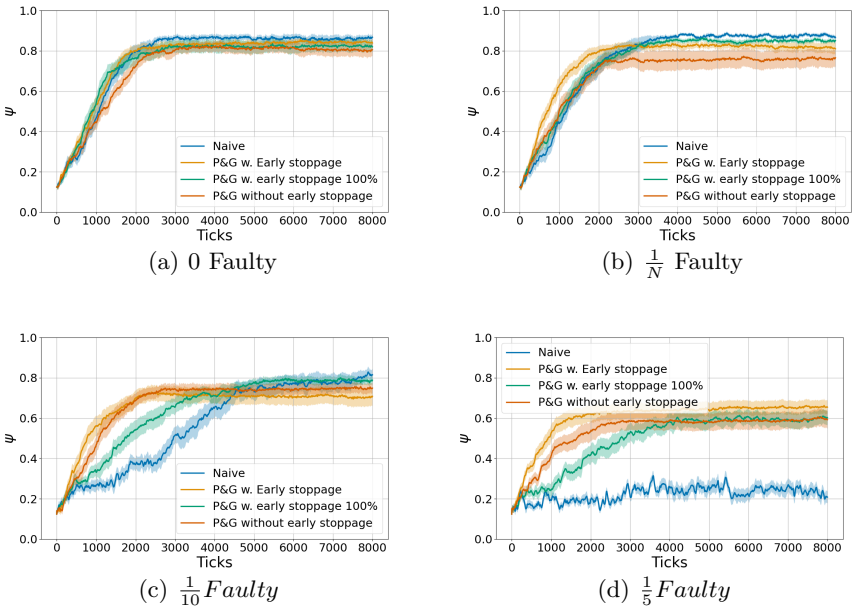
(d) 0 Faulty, not avoiding faulty

Fig. 5. The plots illustrate the evolving order of a swarm consisting of 40 robots under 0 and $\frac{1}{5}$ faulty robots. The first row shows the results when faulty robots are avoided. The second row shows what happens when they are not avoided.

This raises the question of the cause of the decline in the P&G performance in the 0-fault scenario. Is it due to the intermittent motion itself, or is it influenced by the rule regarding the exclusion of static robots (Eqs. 5–6)? To answer this

question, we examine the results from swarms employing a nullified P&G variant, where robots intermittently pause, but they do not ignore others. Figures 5(c)–5(d) show that this nullified model is practically indistinguishable from the naive model. In other words, the exclusion of misclassified faulty robots is the cause for the subpar performance of the P&G in the case of no faulty robots.

We therefore turn to focus on reducing the misclassification rate, using the parameterized pause termination rule discussed above. Figure 6 shows the results for a swarm of 40 robots, of which different proportions of robots are faulty. The figure contrasts the naive model, with three variants of the P&G mechanism: *Early stoppage* (2-tick minimal pause, 70% majority for terminating a pause), *Early stoppage 100%* (same, but 100% of observed robots must move to cause early termination of the pause), fixed pause (without the option for early termination—this is the best-performing variant of the previous experiment). The results show the 70% majority persistently outperforms competing alternatives in both order and convergence time.



(a) 0 Faulty (b) $\frac{1}{N}$ Faulty

(c) $\frac{1}{10}Faulty$ (d) $\frac{1}{5}Faulty$

Fig. 6. Plots show the order parameter of the naive model (*dark blue line*) and different variants of the P&G model. The swarm has 40 robots, and each plot has a different number of faulty robots. The chosen model is the *light orange line*.

Next, we evaluated the P&G-70 mechanism (2-tick minimal pause, 70% majority) in all fault proportions, in all swarm densities. The results are shown in Figure 7. They demonstrate that P&G-70 is able to escape from all tested proportions of faulty robots in all the swarm densities. Comparing Fig. 7(a) to the corresponding figure for the naive model (Fig. 4(a)), we see that P&G-70

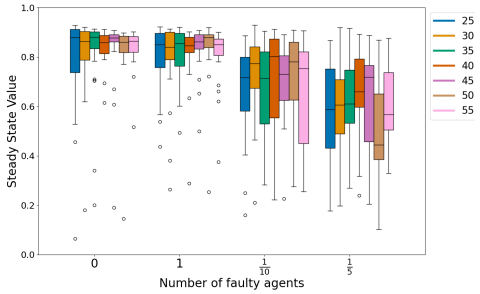
degrades much more gracefully. Fig. 7(b) shows a simulation screenshot from one of the trials.

Despite the clear improvement over the naive model, the SSV variance is considerable. Preliminary analysis of the causes yielded an interesting observation: The P&G allows exclusion of failing members, but causes *fragmentation*, splitting the swarm into ordered sub-swarms that head in different directions (see screenshot in Fig. 8(a)). This behavior occurs as the connectivity graph of the robots is broken by the faulty robots. A possible solution is to have a larger R_{sense}, allowing the robot to consider farther active robots. However, as reported by Ferrante et al. [11], this leads to either a form of milling or a collapse of the swarm (which we have witnessed).

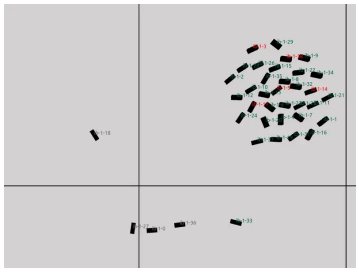
We were surprised to discover that a larger R_{sense} seems to work well, when sensor occlusions are taken into account. Such occlusions have been reported to sometimes interfere with collective motion [20], but here they seem to compensate for the negative effects of the larger R_{sense} radius. Adding occlusions changes the process of neighbor selection [12], which has a dramatic effect. Contrast Fig. 8(a), with the corresponding screenshot where P&G-70 is used with a larger R_{sense} of 0.25 (before: 0.19), and a simple occlusion rule, whereby a robot can perceive a neighbor (active or faulty) if there is a direct line of sight between body centers. The effects can also be seen in Fig. 8(c) and Fig. 8(d). The results for the naive model had not changed because of the larger R_{sense}, but P&G-70 performance clearly improved (Fig. 8(d)). We caution that these promising results are preliminary.

5 Summary

Taking a step towards considering physical robot constraints and properties in swarming, we focus on the challenging problem of making collective motion resilient to failing agents that stop permanently, but cannot be identified as such by any means other than measuring their velocity. This is a serious open challenge [5,8,33,35]. Inspired by the behavior of desert locusts, we modified the

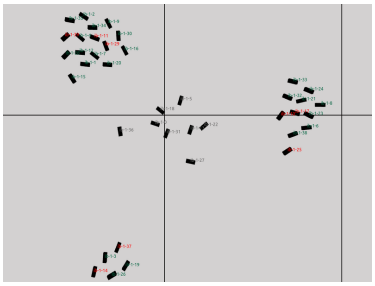


(a) SSV of different swarm sizes, at different faulty robot proportions.

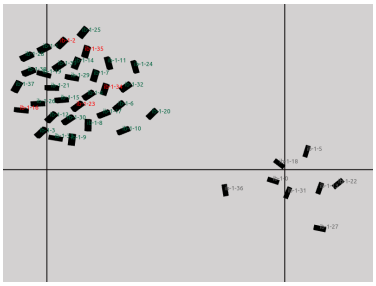


(b) Simulation screenshot. Failed robots left behind.

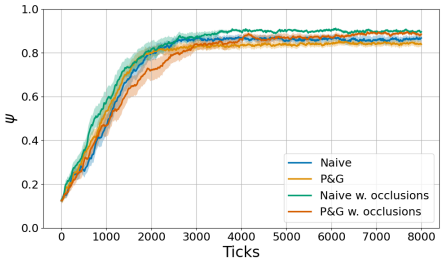
Fig. 7. Results P&G (70% majority).



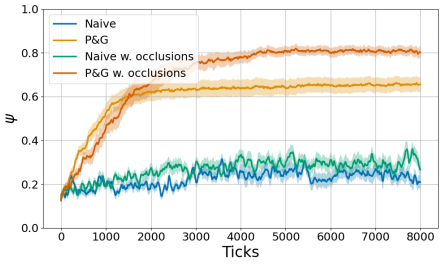
(a) P&G-70 - $\frac{1}{5}$ Faulty



(b) P&G-70 w. occlusions - $\frac{1}{5}$ Faulty



(c) 0 Faulty



(d) $\frac{1}{5}$ Faulty

Fig. 8. Simulation screenshots of a fragmented swarm without occlusions **(a)** and a coherent swarm with larger R_{sense} with occlusions **(b)**. In both, green labels mark a *go* state, red labels mark a *pause* state and gray labels mark *faulty* robots. Second row figures **(c), (d)** show the evolving order parameter of the original P&G-70 and naive models, and same with increased R_{sense} and occlusions.

individual locomotion from continuous to intermittent pausing (*pause and go*). By carefully considering the pause termination conditions, resilient swarming is achievable, as the pauses allow exclusion of stopped robots (even in large numbers), but maintain the coherence when no faults occur. The method was tested in hundreds of simulation trials, with different swarm sizes and portions of faulty robots.

A negative side effect of the P&G mechanism is that it may cause the swarm to split into smaller (ordered) groups. We plan to investigate this in depth.

Acknowledgements. This work was supported in part by ISF grant #1373/24. As always, thanks to K. Ushi.

References

1. Y. Aidan, I. Bleichman, and A. Ayali. Pausing to swarm: locust intermittent motion is instrumental for swarming-related visual processing. *Biology Letters*, 20(2):20230468, Feb. 2024. Publisher: Royal Society.

2. G. Ariel and A. Ayali. Locust Collective Motion and Its Modeling. *PLOS Computational Biology*, 11(12):e1004522, Oct. 2015. Publisher: Public Library of Science.
3. G. Ariel, Y. Ophir, S. Levi, E. Ben-Jacob, and A. Ayali. Individual Pause-and-Go Motion Is Instrumental to the Formation and Maintenance of Swarms of Marching Locust Nymphs. *PLOS ONE*, 9(7):e101636, Feb. 2014. Publisher: Public Library of Science.
4. A. Ayali and G. A. Kaminka. The hybrid bio-robotic swarm as a powerful tool for collective motion research: a perspective. *Frontiers in Neurorobotics*, 17:1215085, 2023.
5. J. D. Bjerknes and A. F. T. Winfield. On Fault Tolerance and Scalability of Swarm Robotic Systems. In *Distributed Autonomous Robotic Systems: The 10th International Symposium*, pages 431–444. Springer, Berlin, Heidelberg, 2013.
6. J. Borenstein, H. R. Everett, and L. Feng. Where am i? — systems and methods for mobile robot positioning. Technical report, University of Michigan, 1996.
7. D. S. Calovi, A. Litchinko, V. Lecheval, U. Lopez, A. P. Escudero, H. Chaté, C. Sire, and G. Theraulaz. Disentangling and modeling interactions in fish with burst-and-coast swimming reveal distinct alignment and attraction behaviors. *PLOS Computational Biology*, 14(1):e1005933, Jan. 2018. Publisher: Public Library of Science.
8. A. Christensen, R. O'Grady, and M. Dorigo. From Fireflies to Fault-Tolerant Swarms of Robots. *IEEE Transactions on Evolutionary Computation*, 13(4):754–766, Aug. 2009.
9. I. D. Couzin, J. Krause, R. James, G. D. Ruxton, and N. R. Franks. Collective Memory and Spatial Sorting in Animal Groups. *Journal of Theoretical Biology*, 218(1):1–11, Sept. 2002.
10. J. T. Emlen Jr. Flocking behavior in birds. *The Auk*, 69:160–170, 1952. Place: US Publisher: Ornithological Society of North America.
11. E. Ferrante, A. E. Turgut, C. Huepe, A. Stranieri, C. Pinciroli, and M. Dorigo. Self-organized flocking with a mobile robot swarm: a novel motion control method. *Adaptive Behavior*, 20(6):460–477, 2012. Publisher: SAGE Publications.
12. B. T. Fine and D. A. Shell. Unifying microscopic flocking motion models for virtual, robotic, and biological flock members. *Autonomous Robots*, 35(2):195–219, Oct. 2013.
13. N. Geva, M. Guershon, M. Orlova, and A. Ayali. Memoirs of a locust: Density-dependent behavioral change as a model for learning and memory. *Neurobiology of Learning and Memory*, 93(2):175–182, Feb. 2010.
14. I. Giardina. Collective behavior in animal groups: theoretical models and empirical studies. *HFSP Journal*, 2(4):205–219, Aug. 2008.
15. A. Huth and C. Wissel. The simulation of the movement of fish schools. *Journal of Theoretical Biology*, 156(3):365–385, June 1992.
16. V. Joshi, S. Popp, J. Werfel, and H. F. McCreery. Alignment with neighbours enables escape from dead ends in flocking models. *Journal of the Royal Society Interface*, 19(193):20220356, July 2022.
17. Y. Katz, K. Tunstrøm, C. C. Ioannou, C. Huepe, and I. D. Couzin. Inferring the structure and dynamics of interactions in schooling fish. *Proceedings of the National Academy of Sciences*, 108(46):18720–18725, Nov. 2011. Publisher: Proceedings of the National Academy of Sciences.
18. D. L. Kramer and R. L. McLaughlin. The Behavioral Ecology of Intermittent Locomotion1. *American Zoologist*, 41(2):137–153, Apr. 2001.

19. D. L. Krongauz, A. Ayali, and G. A. Kaminka. Vision-based collective motion: A locust-inspired reductionist model. *PLOS Computational Biology*, 20(1):e1011796, 2024.
20. D. L. Krongauz, A. Ayali, and G. A. Kaminka. Vision-based collective motion: A locust-inspired reductionist model. *PLOS Computational Biology*, 20(1):e1011796, Jan. 2024. Publisher: Public Library of Science.
21. L. Lei, R. Escobedo, C. Sire, and G. Theraulaz. Computational and robotic modeling reveal parsimonious combinations of interactions between individuals in schooling fish. *PLOS Computational Biology*, 16(3):e1007194, Mar. 2020. Publisher: Public Library of Science.
22. M. J. Mataric. Behaviour-based control: examples from navigation, learning, and group behaviour. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2-3):323–336, Apr. 1997.
23. N. Moshtagh, A. Jadbabaie, and K. Daniilidis. Distributed Geodesic Control Laws for Flocking of Nonholonomic Agents. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 2835–2840, Dec. 2005. ISSN: 0191-2216.
24. R. Nathan, W. M. Getz, E. Revilla, M. Holyoak, R. Kadmon, D. Saltz, and P. E. Smouse. A movement ecology paradigm for unifying organismal movement research. *Proceedings of the National Academy of Sciences*, 105(49):19052–19059, Dec. 2008. Publisher: Proceedings of the National Academy of Sciences.
25. C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo. ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295, Dec. 2012.
26. J. Qi, L. Bai, Y. Xiao, Y. Wei, and W. Wu. The emergence of collective obstacle avoidance based on a visual perception mechanism. *Information Sciences*, 582:850–864, Jan. 2022.
27. E. Reches, D. Knebel, J. Rillich, A. Ayali, and B. Barzel. The Metastability of the Double-Tripod Gait in Locust Locomotion. *iScience*, 12:53–65, Feb. 2019.
28. C. W. Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '87, pages 25–34, New York, NY, USA, Aug. 1987. Association for Computing Machinery.
29. O. Rimer and G. Ariel. Kinetic order-disorder transitions in a pause-and-go swarming model with memory. *Journal of Theoretical Biology*, 419:90–99, Apr. 2017.
30. P. Romanczuk and L. Schimansky-Geier. Swarming and Pattern Formation due to Selective Attraction and Repulsion. *Interface Focus*, 2(6), May 2012. arXiv:1205.3406 [cond-mat].
31. J. Strefler, U. Erdmann, and L. Schimansky-Geier. Swarming in three dimensions. *Physical Review E*, 78(3):031927, Sept. 2008.
32. D. Strömbom. Collective motion from local attraction. *Journal of Theoretical Biology*, 283(1):145–151, Aug. 2011.
33. D. Tarapore, A. L. Christensen, and J. Timmis. Generic, scalable and decentralized fault detection for robot swarms. *PLOS ONE*, 12(8):e0182058, Aug. 2017.
34. T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet. Novel Type of Phase Transition in a System of Self-Driven Particles. *Physical Review Letters*, 75(6):1226–1229, Aug. 1995. Publisher: American Physical Society.
35. A. F. Winfield and J. Nembrini. Safety in numbers: fault-tolerance in robot swarms. *International Journal of Modelling, Identification and Control*, 1(1):30, 2006.