Trust Repair in Human-Swarm Teams[+]

Rui Liu[1*], Zekun Cai[2], Michael Lewis[2], Joseph Lyons[3], Katia Sycara[1]

***Abstract*— Swarm robots are coordinated via simple control laws to generate emergent behaviors such as flocking, rendezvous, and deployment. Human-swarm teaming has been widely proposed for scenarios, such as human-supervised teams of unmanned aerial vehicles (UAV) for disaster rescue, UAV and ground vehicle cooperation for building security, and soldier-UAV teaming in combat. Effective cooperation requires an appropriate level of trust, between a human and a swarm. When an UAV swarm is deployed in a real-world environment, its performance is subject to real-world factors, such as system reliability and wind disturbances. Degraded performance of a robot can cause undesired swarm behaviors, decreasing human trust. This loss of trust, in turn, can trigger human intervention in UAVs' task executions, decreasing cooperation effectiveness if inappropriate. Therefore, to promote effective cooperation we propose and test a trust-repairing method (*Trust-repair*) restoring performance and human trust in the swarm to an appropriate level by correcting undesired swarm behaviors. Faulty swarms caused by both external and internal factors were simulated to evaluate the performance of the *Trust-repair* algorithm in repairing swarm performance and restoring human trust. Results show that *Trust-repair* is effective in restoring trust to a level intermediate between normal and faulty conditions.**

I. Introduction

Robotic swarms consist of simple, typically homogeneous robots that interact with other robots and the environment. Swarm robots are coordinated via simple control laws to generate emergent behaviors such as flocking, rendezvous, and deployment. Owing to their scalability and natural robustness to individual robot failures, swarms are attractive for large-scale applications such as environmental monitoring [16], exploration [34], search and rescue [22], and agriculture [1].

Human presence is important in swarm applications since humans can recognize and mitigate shortcomings of the swarm, such as limited sensing and communication. Humans can also provide new goals to the swarm as the environment and mission requirements dictate [12]. Intervention, however, carries its own costs. The evolution of swarm behavior has been shown to be highly time dependent with delay in input leading to better outcomes under some conditions, a phenomena termed Neglect Benevolence [19] and shown to be difficult for humans to manage [18]. In addition, perturbing a swarm leads to transient reductions in consensus and corresponding loss of efficiency. By maintaining correct trust calibration, the operator can balance the costs of complacency (not intervening when needed) with those of intervening when it is unnecessary.

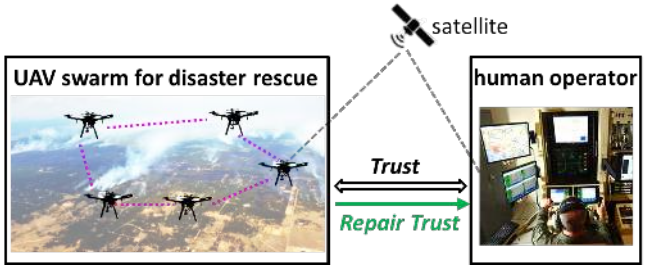

Fig. 1. Illustration of trust repair between swarm and human. When a swarm realizes its faulty behaviors decrease human trust, it will correct these behaviors to regain human trust during swarm-human cooperation.

Human-swarm cooperation can be easily influenced by real-world uncertainty, such as wear to a robot's motor, sensor failures, or wind disturbances. Because swarms coordinate by consensus the presence of faulty robots can lead to increasing divergence of the swarm from its intended behavior. External factors such as wind can also act to disrupt individual behavior and divert the swarm from its intended goal. These faulty behaviors change both the appearance and performance of a swarm.

Prior literature on human-robot trust, has found that trust is influenced primarily by the perceived performance of the robot [8]. However, task performance of swarms is often not intelligible to the operator [25], [28], since swarms perform tasks through complex interactions among the swarm members themselves, such as consensus that takes time to converge. For example, the swarm may not follow the operator's command in a case where it first needs to maintain connectivity which may not be readily apparent to the operator. Such misperceptions make trust modeling and estimation challenging.

These difficulties might be avoided if the swarm could detect anomalies and act to repair itself. In our previous paper [14], a decentralized trust-aware behavior reflection method was demonstrated to effectively correct swarms' faulty behaviors. In this paper we ask whether observers of a faulty swarm being repaired through this algorithm will retain their trust in the swarm.

Prior studies [10], [31] and [20] have reliably modeled human interaction with mobile robots and swarms predicting within trial judgments of trust and interventions from one another and system state. In the present experiment we collect trust judgments for an immediately preceding behavior

[+] This work has been funded by AFOSR award FA9550-15-1-0442 and AFOSR/AFRL award FA9550-18-1-0251.

[1] is with Robotics Institute, School of Computer Science, Carnegie Mellon University, PA, USA. [2] is with School of Computing and Information, University of Pittsburgh. [3] is with Air Force Research Laboratory, Wright-Patterson AFB, Dayton, OH, USA *Rui Liu is the corresponding author ruiliu.robotics@gmail.com. Draft submitted to IEEE ROMAN 2019 at 05.06.2019; shared on Arxiv at 05.08.2019.

sequence but without allowing intervention, relying on the earlier studies to establish that relationship. Because both faulty and repairing swarms exhibit weakened coherence and an earlier study [20] has shown trust judgments to be more influenced by swarm appearance than performance, this experiment investigates whether performance improving repairs will lead to greater trust. If so, a supervising human could be predicted to calibrate trust in the proper direction leading to better human-swarm performance. Otherwise, improvements due to self-repair might be cancelled out by unnecessary human intervention.

II. Related Work

While swarms are largely robust against loss of individual members, failure mode and effects analysis [2] finds them far more vulnerable to partial failures where faulty robots continue to contaminate the swarm's consensus. Research in swarm self-healing, however, has focused largely on replacing lost robots within formations ignoring the greater danger of partial failures likely to be encountered in real-world deployments. Zhang et al. [32], for example, develop methods for mobile robot networks to maintain logical and physical topology of the network when robots fail and must be replaced within a formation. They further demonstrate the stability of motion synchronization under their topological repair mechanism. [33] extend this robot replacement strategy by introducing a gradient for selecting repair robots yielding small improvements over random selection used by [32].

More recently [24] has addressed the problem of partial failures and adversarial robots by protecting the swarm though resilience by restricting robot updates to values of neighbors near their own. Their results for swarms meeting connectivity requirements and based on communication of constant or time varying values by faulty robots showed convergence of the swarm to correct headings. Their scenario, however, did not encompass the variety of real-world conditions such as faulty robots which might continue to be influenced by their neighbors or external influences such as gusts of wind which could disrupt consensus. In this paper we extend their approach through an active *Trust-repair* method that can address these typical robot faults, such as motor degradation or wind disturbance

Current strategies for swarm self-healing passively increase the swarm resilience in order to cancel the negative influence of faulty robots. Passive methods require greater connectivity for the swarm and need to specify fault types and speed/angular ranges, which are difficult to preset in practical environments. Because of the cumulative effect of faulty values on consensus it is necessary to actively correct faulty behaviors when they appear. The proposed *Trust-repair* method corrects faulty robots by updating a robot's motion with reference to its trusted neighbors. *Trust-repair* can correct faulty behaviors that cannot be prevented by other resilience-increasing techniques. Combining both passive and the Trust-repair active methods corrects faulty swarm behaviors more effectively than either alone.

III. Human-Swarm Cooperation

We envision a human-swarm system in which a UAV swarm is remotely supervised by a human operator. The swarm performs "distributed biased flocking", a critical tactic in tasks such as UAV rendezvous, area coverage and search. To focus on the *Trust-repair* algorithm's effects on human trust, the experimental environment is obstacle free.

A UAV swarm consists of n holonomic robots $X_i (i = 1, 2, ..., n)$. Each robot X_i is defined by the tuple $< x_i, y_i, \theta_i >$. x_i and y_i are the robot i's horizontal and vertical positions and θ_i denotes robot i's orientation. The swarm's communication graph is given by $G = (\mathcal{V}, \mathcal{E})$ where every node $v \in \mathcal{V}$ represents a robot. Every robot i only communicates with its direct neighbors $j \in N_i$, where N_i is the set of all neighbors of i within the communication radius, R. If robot j is a neighbor of i, then edge $(v_i, v_j) \in \mathcal{E}$. The connectivity graph is connected and undirected. The dynamic model [17] for each robot is defined as follows

$$\dot{x}_i = u_i^{\mathrm{v}} cos(\theta^i) \tag{1}$$

$$\dot{y}_i = u_i^{\mathrm{v}} sin(\theta^i) \tag{2}$$

$$\dot{\theta}_i = u_i^w, \tag{3}$$

where u_i^v and u_i^w are the linear and angular velocity of robot i. The bearing vector (or heading direction), b_{ij}, between robot i and j is given by

$$b_{ij} = \frac{X_j - X_i}{||X_j - X_i||_2}. \tag{4}$$

The remainder of this paper will consider swarms tasked with using biased flocking to move in an eastward direction. Traditional biased flocking rules update the motion status of a robot i at time step t as follows

$$u_i[t+1] = \frac{1}{N_i + 1}(u_i[t] + \sum_{j \in N_i} u_j[t]). \tag{5}$$

The conventional method in equation 5 is fragile in that, both faulty and failed robots contribute unreliable information for use in their neighbors' estimates of consensus. As these errors accumulate the swarm will increasingly deviate from its correct behavior. These deviations along with loss of coherence due to differences in local estimates across the swarm can lead to a loss of human trust.

In this paper, "Faulty robot" refers to a robot with undesired but correctable behaviors. "Failed robot" refers to a robot with undesired but not correctable behaviors. Influenced by faulty and failed robots, a swarm with abnormal behaviors, such as partial disconnection or heading deviation, is defined as an "untrustworthy swarm". The real-world factors, such as degraded motors on a robot, uncertainty in sensors and mechanical systems, or wind/rain disturbances from the environment can cause abnormal robot behaviors and impair robot performance. These factors are defined as "environmental influence".

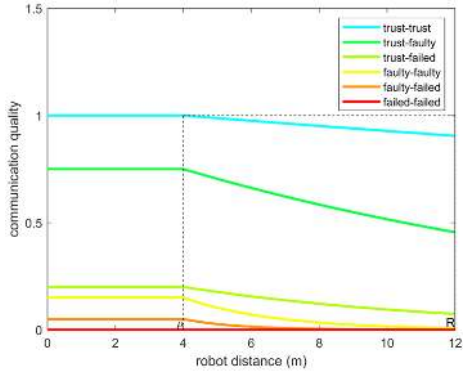

Fig. 2. Trust-aware communication quality assessment. The quality of communication between trusted robots is high, while the quality is low when faulty robots involved.

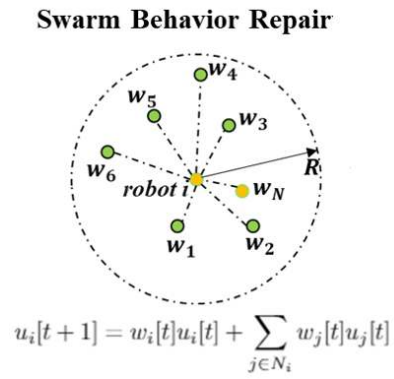

Fig. 3. Overview of the Trust-repair method. A robot's speed is calculated by differently considering its neighbors' speed.

IV. Swarm Self-Healing Through Trust-Repair

When a faulty robot appears in a swarm, it becomes unreliable to update other robot's status by referring the faulty robots' motion status (calculated by Equation 5) [9]. Instead, it is more reliable to constrain information sharing between a faulty robot and its neighbors. In particular, if the trust level is high (faultiness is low) then the strategy "accept high-trust information" is employed. On the other hand, if trust level is medium (fault level is medium) then "reduce middle-trust information" is employed; and if trust level is low (faultiness is high) then "refuse low-trust information". We propose a novel information updating method based on the weighted mean subsequence reduced algorithm (WMSR) [23]. Instead of merely averaging values as in the previous update method, our *Trust-repair* method updates information differently based on the communication quality (Equation 6). Weights w_i are calculated in section IV.B.

$$u_i[t+1] = w_i[t]u_i[t] + \sum_{j \in N_i} w_j[t]u_j[t] \tag{6}$$

A. Trust-Aware Communication Quality Assessment

The overall communication graph for robot i is $\mathcal{E} = \{(i,j) \mid j \in N_i\}$. Based on the estimated trust levels of the two robots $\{i,j\}$, communication quality, $f_{ij} \in [0,1]$, is used to measure the reliability of exchanged information. The trust-aware communication quality is dynamically updated to reflect the changing communication graph using Equation 7. The best communication distance between two robots i and j is ρ. Communication within ρ is considered as the communication with the best quality. The communication radius is R. The parameter, η, is used as a weighting factor to discourage the impact of faulty robots on their neighbors.

$$f_{ij} = \begin{cases} 0 & ||x_i - x_j|| \geq R \\ \frac{1}{2}(g_i + g_j)\eta & ||x_i - x_j|| \leq \rho \\ \frac{(g_i+g_j)\eta}{2} \exp \frac{-\gamma(||x_i - x_j|| - \rho)}{R-\rho} & otherwise \end{cases} \tag{7}$$

where g_i is the trust level of the robot i. The above communication quality evaluation method implies that within the communication range, the communication reliability is the average of the two robots' trust values. If both robots are trusted, their communication is the most reliable; if one robot is faulty, the most reliable communication under that connection is the communication from the trusted robot.

The quality assessment for robot communications is visualized in Figure 2. The rationale of designing the trust-aware communication quality is to encourage information sharing with trusted robots by using higher upper limits on their communication quality, while discouraging information sharing with untrusted robots by using lower upper limits on the communication quality. Meanwhile, to encourage a compact swarm with closer distances among robots, the communication quality is decreased if the robot distance increases. Figure 2 shows that the communication quality among trusted robots is close to 1, while the quality among failed robots is 0.

For the curves shown in Figure 2, the g values are (1, 0.5, 0) for trusted robots, faulty robots and failed robots, respectively. η values are (1, 1, 0.4, 0.3, 0.2, 0.2) and γ values are (0.1, 0.5, 1, 3, 5, 7) for communications between trusted-trusted robots (trust-trust), trusted-faulty robots (trust-faulty), trusted-failed robots (trust-failed), faulty-faulty robots (faulty-faulty), faulty-failed robots (faulty-failed), failed-failed robots (failed-failed). g and η are used to set upper limits on the communication quality. γ defines the sensitivity of quality to mutual distance. For the remainder of the paper we set the communication radius to be $R = 12m$ and the best communication distance to be $\rho = 4m$.

The novel trust-weighted Laplacian matrix, $[L]_{ij}$, calculated as $[L]_{ij} = [D]_{ij} - [A]_{ij}$ can then be defined as:

$$[L]_{ij} = \begin{cases} -f_{ij} & i \neq j \\ \sum_j f_{ij} & i = j. \end{cases} \tag{8}$$

The eigenvalues $\{\lambda_i \mid i = 1, 2, ..., n\}$ of L are real and they satisfy $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. The connectivity measure λ_2 is estimated by the equation $Le_2 = \lambda_2 e_2$ and the eigenvector e_2.

B. Trust-Aware Swarm Behavior Correction

A swarm proactively corrects its faulty behaviors using a two step process. First, it corrects the faulty robots by

restraining the negative influence from faulty robots and referring to trusted robots for behavior correction. The failed robots are isolated from other trusted robots, preventing the sharing of unreliable motion information. The connectivity control in Section 5 is then used to reduce the distance between robots and their "normal" neighbors. In doing so, a robot adjusts its behavior – heading direction and speed – using a larger amount of trusted motion information.

$$w_k[t] = \frac{\hat{f}_k[t]}{\hat{f}_i[t] + \sum_{j \in N_i} \hat{f}_j[t]}, k \in [i, N_i] \tag{9}$$

Weights for updating each robot's status are calculated by Equations 6 and 9. The result of the weighted update mechanism is shown on the right side of Figure 3. For updating a robot i, weights w_k are calculated by normalizing all the communication quality values in a communication range, shown in equation 9. When $k = i$, $\hat{f}_k = g_i$ (i.e, the trust level of itself). If $k = j \in N_i$ then $\hat{f}_k = f_{ij}$ (i.e., the communication quality between robots i and j). $\hat{f}_i = g_i$ for all values of k.

With the trust-weighted update, the control input u_v^i and u_w^i for robot motors are changed to $u_{v,trust}^i$ and $u_{w,trust}^i$. The gains K_v and K_w are parameters for adjusting the motor output.

$$u_{i,trust}^{\mathrm{v}} = (K_{\mathrm{v}} + K_{\mathrm{v},trust})(\mathrm{v}_i + q_{N_i})^T b_i \tag{10}$$

$$u_{i,trust}^{w} = (K_w + K_{w,trust})\big(\gamma_i + \phi(b_i, q_{N_i})\big) \tag{11}$$

Let $u_i[t+1]$ denote the actual speed of a robot with abnormal behaviors at the moment $t+1$, then the expected speed calculated by referring to its neighbors is denoted by $u_{i,trust}[t+1]$. The extra trust-gain $K_{\mathrm{v},trust}$ and $K_{w,trust}$ can then be solved to adjust the control output of robot motors. The gains are updated based on the difference between the actual and trusted robot speeds.

$$K_{\mathrm{v},trust}[t+1] = \frac{u_{i,trust}^{\mathrm{v}}[t] - u_i^{\mathrm{v}}[t]}{u_i^{\mathrm{v}}[t]} \tag{12}$$

$$K_{w,trust}[t+1] = \frac{u_{i,trust}^{w}[t] - u_i^{w}[t]}{u_i^{w}[t]} \tag{13}$$

To avoid collision, the safe distance (repulsion radius) for separating robots is set to r. For a pair of robots i and j, their positions at the moment t are x_i and x_j. The overall swarm safety is maintained during the correction period $[0, T]$ by maintaining safety distance $h_{i,j}^{safe}$ for any robot pair i and j. $\mathcal{H}_{i,j}^{safe}$ is the set of all safe distances.

To ensure that a faulty robot has enough reliable neighbors to share trusted motion information for behavior corrections, a control law for swarm connectivity maintenance is designed to encourage the relative closer distance between a robot and a trusted robot [14].

V. Evaluation

The effectiveness of *Trust-repair* for a human-swarm system depends both on its effectiveness in protecting the swarm from faulty robots and disturbances and its effectiveness in supporting an appropriate level of trust in its human supervisor. In this section we present both swarm performance data and human ratings of trust for fourteen scenarios.

To validate the effectiveness and generalizability of *Trust-repair* in helping the swarm self-heal, swarm behavior correction was conducted in two types of faulty scenarios {internal influence - motor issue, external influence - wind disturbance}, with four types of swarm configurations {six robots with one/two faulty robot(s), twelve robots with one/two faulty robot(s)}. Normal, faulty and repair conditions were simulated using MATLAB. The experiment includes 14 simulated scenarios (2 normal scenarios, 8 degraded motor scenarios including 4 faulty and 4 repaired, 4 wind disturbance scenarios including 2 faulty and 2 repaired). Simulated faults were chosen as representative of faults commonly found in real-world UAV deployments, such as densely distributed forests/buildings and extreme weather conditions, which can affect robot communication, spatial distributions and system reliability [11] [27]. The task for the swarm in all scenarios was distributed biased flocking with human-desired heading-direction "East". The faulty/failed robots for each degraded motor scenario were 1 or 2 robots, which are the minority in the swarm so that the faulty robots could potentially correct themselves by incorporating sufficient values from trusted robots. Under the influence of abnormal robots, neighboring robots can also became faulty/failed. The map size for the flocking was 60m×60m. The velocity for each robot was set as $1.0m/s$. To observe the misleading effect of one faulty robot on its neighbors, robot locations were randomly initialized but still in a circle with radius of $8m$. The heading direction of all the robots pointed to the circle center. To avoid collision, the repulsion radius securing robot safety was set as $2m$. For all conducted experiments $\beta_1 = 10\%$ and $\beta_2 = 50\%$ were used for the faulty behavior detection.

A. The Evaluation of Swarm Behavior Correction

1) Internal Influence – Robot Motor Wear

Due to a motor wear, the speed of the faulty robots was lower than the normal robots. Through distributed control, the robot's lower speed will be exchanged with its neighbors lowering their speed as well potentially even affecting other robots' headings. These undesired speed and heading changes decrease a swarm's performance potentially reducing human trust on the swarm's capability.

For a normal swarm (the first row of Figure 4), after about 27 time steps ($13.5s$), the velocity of all 6 robots achieved the desired consensus of $1.0m/s$ with a $0.1m/s$ deviation; After about 28 time steps ($14s$), the heading of all 6 robots achieved consensus on the "East" direction. The connectivity λ_2 was 6, which means all robots achieved the best communication in this scenario.

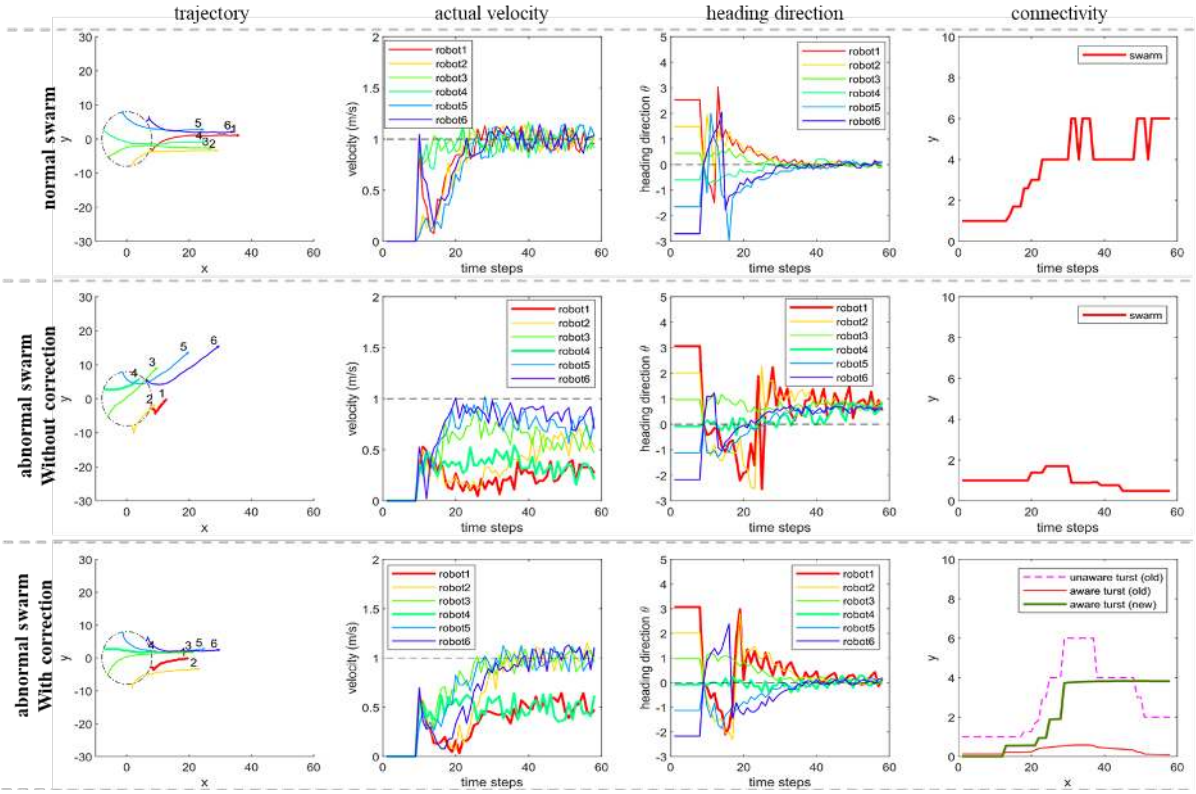

Fig. 4. System responses in three situations, with two faulty robots under motor-wear influence. In a normal situation, the swarm flock to east with a motion consensus on both linear speed and heading direction; while when a robot has a faulty motor, swarm consensus is destroyed; after corrections using Trust-repair, the swarm's faulty behaviors were corrected to reach a motion consensus again.

Figure 4 shows a scenario in which Robot 1 and 4 had worn motors. As shown, despite faulty robots in the swarm, the velocity consensus was maintained (Figure 4, figures of row 2 trajectory, actual velocity and heading direction). Robot 1 and 4 were disconnected from the swarm with connectivity finally reaching 0 (Figure 4, connectivity). The heading direction of the swarm shifted to $0.8rad$ after 30 time steps ($15s$) (Figure 4,heading direction).

Trust-repair uses trust awareness to help a robot to identify faulty behaviors of itself and its neighbors. In this case, Robot 1, whose speed is 50% lower than the expected speed, was considered as an untrusted and failed robot (Figure 4, row2 trajectory), thereby decreasing swarm performance and potentially human trust. The communication quality between Robot 1 and other normal robots decreased as calculated by the "trust-failed" curve in Figure 2. With the *Trust-repair* correction, the information exchanged with the robot 1 and 4 was tightly constrained. After 28 time steps ($14s$), robot 1 and 4 were disconnected from the normal robots. The swarm with only trusted robots achieved velocity consensus after 28 time steps ($14s$), and achieved consensus on heading after 38 time steps ($19s$) with only a 0.2 rad deviation (Figure 4, row 2, the figure of actual velocity and heading direction) from the east direction. This demonstrates that *Trust-repair* was effective in correcting the faulty behaviors of the swarm. After behavior correction (Figure 4, row 3 connectivity), the swarm which constrained the information exchanged with the faulty robots 1 and 4 had connectivity that increased to a high level of 3.8, showing the effectiveness of *Trust-repair* in encouraging connectivity among trusted robots.

Similar effectiveness in behavior corrections are verified for other scenarios { 6 robots with 1 faulty robot, 12 robots with 1 faulty robots, 12 robots with 2 faulty robots}, with the an east-flocking goal accomplishment with 0.2 rad derivation and 2 4 increment of the connectivity value.

2) External Influence – Wind Disturbance

When robots in a swarm cross a wind zone, the wind imparts extra linear and angular velocity to the robots. For this experiment, a wind region with size of 15×15 was located in the convex hull formed by the following set of vertices ((15,4), (30,4), (30,19),(15,19)). Before reaching the region, the swarm had already achieved motion consensus. Some robots will cross the wind region and gain an extra $0.25 \sim 0.75m/s$ linear velocity along the "North" direction and an angular deviation of $0.1rad/s$.

As Figure 5(row 1 trajectory) shows, Robots 3 and 4 crossed the wind region first. They then attracted Robots 5, 6, 1, 2, 7, and 8 into the wind zone. Without correction, a motion consensus was not achieved (Figure 5, row 1). The connectivity decreased to 0 after about 30 time steps ($15s$)(Figure 5 row 1 connectivity). With the *Trust-repair* correction, misleading information from the untrusted Robots 3 and 4 was quickly constrained and their influence on the other robots was largely reduced. The new swarm without faulty robots achieved velocity consensus of $1.0m/s$ with a $0.1m/s$ deviation at after about 15 time steps ($7.5s$) and

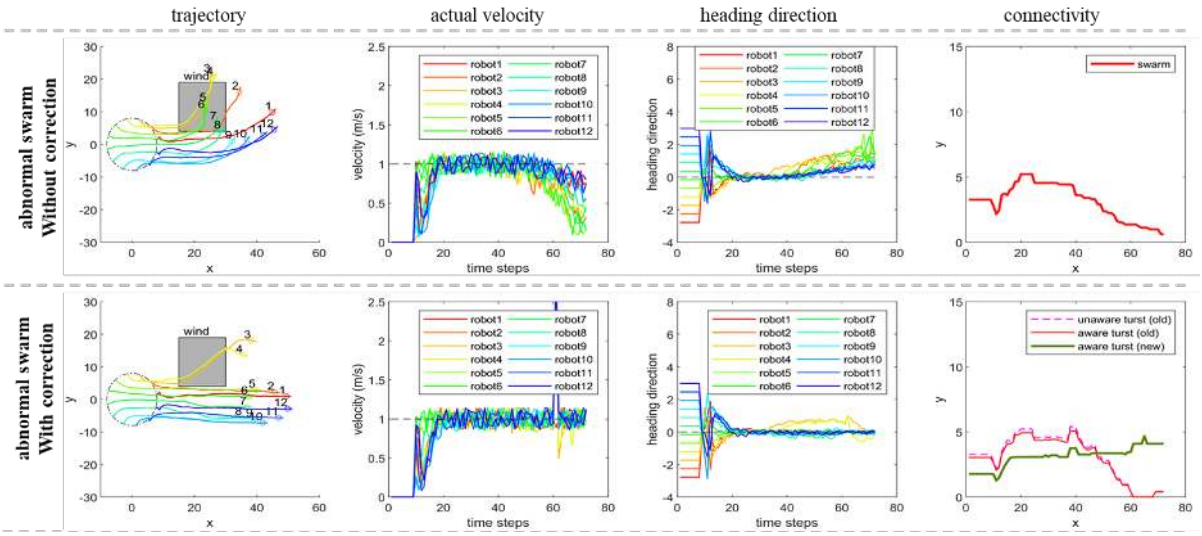

Fig. 5. System responses of a swarm under wind disturbance. With wind disturbance, a swarm's motion consensus was destroyed; while after correction, the swarm's motion consensus was achieved.

achieved heading direction consensus after 30 time steps (about 15s), shown in Figure 5, row 2. As shown in Figure 5, row 2 connectivity, connectivity of the old swarm without the *Trust-repair* correction was decreased to 0 finally, due to the disconnection of the faulty Robot 2, 3, 4, 5, 6 and 7. In contrast, the connectivity of the new swarm (swarm after removing the faulty Robot 3 and 4) increased to 4.5, showing the effectiveness of *Trust-repair* in correcting abnormal swarm behaviors caused by disturbances such as wind.

Similar effectiveness in behavior corrections were verified in the other scenario {6 robots with wind disturbance}, with the east-flocking goal accomplishment with 0.05 rad derivation and 3.8 increment of the connectivity value.

B. User Study: Effects of Trust-Repair Algorithm on Human Trust

1) Methods: To measure the effects on human trust of observing the repair algorithm in action we conducted a study on the crowd-sourcing platform Amazon Mechanical Turk [3]. 123 English speaking Volunteers were recruited and paid $3.00 to assess trust levels and answer additional questions about swarm behavior portrayed in brief 15 sec videos. The average time of an experimental session was approximately 30 minutes. Data for one of the fourteen conditions, large swarm single motor failure, was lost for 24 participants due to logging difficulties.

Videos were made for each of the 14 scenarios described in the previous section. A brief tutorial in which participants viewed sample videos of each type (Normal, Faulty, Repair) and were introduced to the questions and scales, preceded data collection. On experimental trials after initially viewing the video, participants were asked whether they had detected a fault. They were then allowed to view the video again before rating their trust in the swarm. Trust was rated on a five point scale: Completely Distrust; Distrust; Neutral; Trust; Completely Trust. On trials in which participants reported a fault they were once again allowed to view the video and asked to identify the faulty robots. This was followed by an opportunity to rate their trust in the robots they had identified as faulty and comment on the features leading to their detection of a fault. At the end of the trial they were asked to rate the swarm's performance on a 5 point scale.

Table I Faulty and Repaired conditions
*Mann Whitney U

Swarm Scenarios Faulty robot's Number/swarm size	Median Trust Level		
	Faulty	Repaired	P*
Motor 1/6	Neutral	Trust	<.001
Motor 2/6	Distrust	Neutral	<.001
Motor 1/12	Trust	Trust$^+$	.069
Motor 2/12	Neutral	Neutral$^+$	n.s.
Wind 6	Distrust	Distrust$^+$	n.s.
Wind 12	Distrust	Neutral	<.001
Motor	Neutral	Trust	<.001
Wind	Distrust	Distrust$^+$	<.001
All	Distrust	Neutral	<.001

2) Results: Participants were more likely to report faults in faulty conditions (Mdn=no fault) than in normal ones (Mdn=fault) (U=37899, $p < .001$). Participants also expressed higher levels of trust (U=45660, $p < .001$) under normal conditions (Mdn=5, completely trust) than when faults occurred (Mdn=1, completely distrust). Participants, however, were not significantly more likely to report a fault in Faulty conditions than in Repair conditions indicating that the *Trust-repair* algorithm did not mask the occurrence of failures. Participants, however, expressed significantly higher trust (U=132,524, $p < .0001$) in repair conditions (Mdn=3, neutral) than in Faulty ones (Mdn=1, completely distrust). Participants in fault conditions experiencing motor failures reported (U=106707, p=.002) slightly more failures

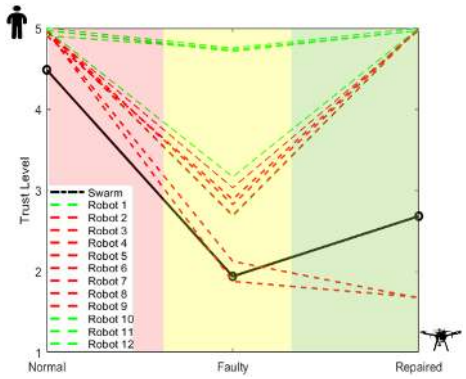

Fig. 6. The trust levels for a swarm and individual robots in different stages normal, faulty, repaired.

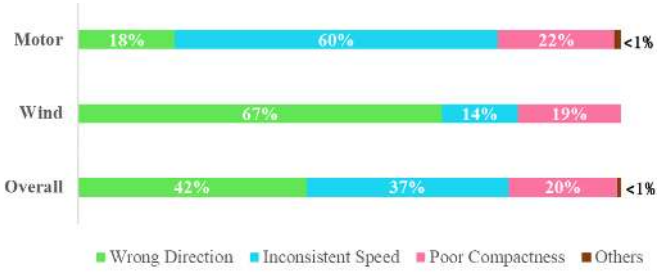

Fig. 7. Performance characteristics influencing a human's trust on a swarm.

(mean=1.13 vs. 1.21) than those in the repair condition who also reported significantly higher trust (U=80850, p $<$ .001) (Mdn=4 trust) than those without repair (Mdn=3 neutral). Results are more equivocal for trust in the wind disturbance for which Faulty conditions (mean=2.03) differed only slightly (U=24502, p $<$ .001) from repair ones (mean=2.36). Effects of repair for maintaining trust were more evident for the smaller swarm (ϕ=.255, p $<$ .001) than the larger swarm where they were only marginally (ϕ=.111, p=.066) significant. Results for trust and number of faulty robots were also small with differences in trust of one (mean=3.03) and two (mean=2.75) faulty robots (U=23612, p=.007) found for unrepaired failures, while a greater difference (U=24574, p $<$ .001) was found in the repair condition between one (Mdn=2, distrust) and two (Mdn=3, neutral) faulty robots.

Table 1 shows median trust levels for scenarios with and without correction.

To examine the effects of *Trust-repair* on human trust more closely figure 6 shows trust ratings for individual robots in the 12 robot wind disturbance scenario. Six of 8 faulty robots, which were considered faulty by more than 50% subjects in the Faulty condition (ranging from 61% $\sim$ 89% with a mean of 73%), were trusted in the Repair condition. Only the two most influenced robots remained untrusted. This illustrates the effects of the *Trust-repair* algorithm which actively isolates faulty robots in order to reduce their influence on the swarm. While reduction in trust for individual robots was largely avoided through the repair, trust in the swarm itself remained depressed with a median of rating of Neutral.

3) Explanation of Trust Loss

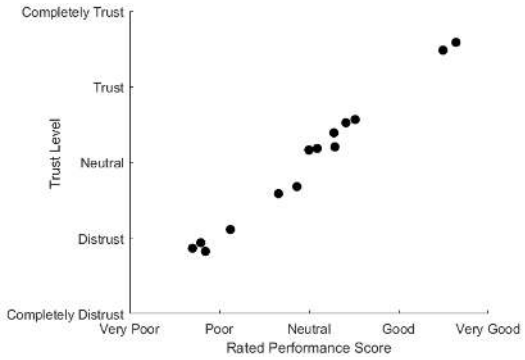

Fig. 8. The linear relation indicates human trust heavily depends on the performance score.

The first 20 participants were asked to describe in a free text comment how they had identified the reported faults. We classified the three most common responses as: a) wrong heading direction, b) inconsistent speed among robots, c) poor compactness of the swarm. The remaining participants were asked to select from among these three causes or provide an alternative in free text.

Figure 7 shows the contributions of these factors to the participants' trust judgments. Direction and speed proved to be the primary determinants of these judgments with inconsistent speed dominating ratings where motor degradation reduced speed of affected robots while 'wrong headings' was the most commonly cited cause for robots blown off course by wind disturbances.

Given the feedback provided by subjects, trust loss was consistent with the heading-direction deviation of the swarm influenced by motor issue (mean=0.33 *rad/s*) and wind disturbance (mean=0.74 *rad/s*), indicating the strong role of performance in the decision to trust. As figure 8 shows, subjective estimates of performance were highly correlated with trust judgments (ρ=.82, p $<$ .001).

VI. Conclusion & Future Work

In prior work [14], we developed a method, *Trust-repair*, for protecting a swarm from the influence of faulty members or external disturbances and demonstrated its effectiveness. In the present paper we report an experiment in which human participants observed brief vignettes of swarm behavior under normal, faulty, and repair conditions in which the *Trust-repair* algorithm acted to reduce the effects of faulty behavior. *Trust-repair* effectively corrected faulty swarm behavior by maintaining an original heading direction within an error of $\pm 0.2 rad$ and maintaining a desired flocking speed of $1.0 m/s$ with a $0.1 m/s$ deviation. Participants rated performance more highly and showed significantly greater trust when the Trust-repair algorithm was employed to protect the swarm. However, despite the small deviation in swarm parameters when protected by *Trust-repair*, reductions from completely trust to trust or neutral were observed in comparison to conditions without failures. So, *Trust-repair* did not completely avoid loss of trust. From the present experiment we cannot determine whether the observed loss of trust was due to the brevity (10 sec) of the videos and would have been

fully restored had the swarm (less its quarantined members) continued to exhibit correct behavior for an extended time. We also cannot tell how the ranges of trust observed in our experiment would affect decisions to intervene were the swarm under active control. To remedy these problems, we plan an additional study in which participants will actively control a flocking swarm performing a foraging task. Failures and disturbances will be introduced at predetermined points and either repaired or allowed to persist until the participant intervenes. Trust ratings will be collected at regular intervals on an interactive slider [20] and these data used to model the dynamic effects of failure and *Trust-repair* on ratings of trust and the performance of the human-swarm system.

VII. Acknowledgements

Thanks for the valuable inputs from our colleagues Fan Jia, Huao Li, Meghan Chandarana, Wenhao Luo.

References

[1] S. Berman, V. Kumar, and R. Nagpal. 2011. Design of control policies for spatially inhomogeneous robot swarms with application to commercial pollination. IEEE International Conference on Robotics and Automation (ICRA), 2011, pp 378-385.

[2] J. Bjerknes and A. Winfield. On Fault Tolerance and Scalability of Swarm Robotic Systems. In Springer Tracts in Advanced Robotics, 2013, pp 1-13.

[3] M. Buhrmester, T. Kwang, T. and Gosling. Amazons Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data? Perspectives on Psychological Science, 6(1), 2011, pp 35.

[4] H. Chen, X. Wang, and Y. Li, A survey of autonomous control for UAV, In Artificial Intelligence and Computational Intelligence, 2009. AICI09. International Conference on, Vol. 2. IEEE, 267-271.

[5] R. Cooley, S. Wolf, and M. Borowczak, Secure and Decentralized Swarm Behavior with Autonomous Agents for Smart Cities. arXiv preprint arXiv:1806.02496(2018).

[6] M. Desai, P. Kaniarasu, M. Medvedev, A. Steinfeld, and H. Yanco. Impact of robot failures and feedback on real-time trust. In Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction. IEEE Press 2013, pp. 251-258

[7] E. J. de Visser, R. Pak, T. H. Shaw. From automationto autonomy: the importance of trust repair in humanmachine interaction. Ergonomics 61,10(2018), pp. 1409-1427.

[8] P. Hancock, D. Billings, K. Schaefer, J. Chen, E. De Visser, R. Parasuraman. A meta-analysis of factors affecting trust in human-robot interaction. Human Factors: The Journal of the Human Factors and Ergonomics Society, vol 53, 2011, pp 517-527.

[9] A. Jadbabaie, J. Lin, and A. S. Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. IEEE Transactions on automatic control 48, 6 (2003), pp. 988-1001

[10] P. Kaniarasu, A. Steinfeld, M. Desai and H. Yanco. Potential measures for detecting trust changes. ACM/IEEE Int. Conference on Human-Robot Interaction, 2012, pp 241-242.

[11] B. Khaldi, F. Harrou, F. Cherif, and Y. Sun. Monitoring a robot swarm using a data-driven fault detection approach. Robotics and Autonomous Systems 97 (2017), pp. 193-203.

[12] A. Kolling, P. Walker, N. Chakraborty, K. Sycara and M. Lewis. Human Interaction with Robot Swarms: A Survey. IEEE Transactions on Human-Machine Systems, vol 46, 2016, pp 9-26.

[13] E. J. Leaman, B. Q. Geuther, and B. Behkam. Hybrid Centralized/Decentralized Control of Bacteria-Based Bio-Hybrid Microrobots. In 2018 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS). IEEE, pp. 1-6.

[14] R. Liu, F. Jia, W. Luo, M. Chandarana, C. Nam, M. Lewis, and K. Sycara, Trust-Aware Behavior Reflection for Robot Swarm Self-Healing, International Conference on Autonomous Agents and Multi-agent Systems (AAMAS2019), 2019.

[15] K. Marinaccio, S. Kohn, R. Parasuraman, and E. J. De Visser. A framework for rebuilding trust in social automation across health-care domains. In Proceedings of the International Symposium on Human Factors and Ergonomics in Health Care, Vol. 4. SAGE Publications Sage India: New Delhi,India, pp. 201-205

[16] A. Marjovi and L. Marques. Optimal spatial formation of swarm robotic gas sensors in odor plume finding, Autonomous robots, vol 35, 2-3, 2013, pp 93-109.

[17] S. Nagavalli, N. Chakraborty, and K. Sycara. Automated sequencing of swarm behaviors for supervisory control of robotic swarms. In Robotics and Automation (ICRA), 2017 IEEE International Conference on. IEEE, pp. 2674-2681.

[18] S. Nagavalli, S. Chien, M. Lewis, N. Chakraborty and K. Sycara. Bounds of Neglect Benevolence in Input Timing for Human Interaction with Robotic Swarms. Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, 2015, pp 197-204.

[19] S. Nagavalli, L. Luo, N. Chakraborty and K. Sycara. Neglect benevolence in human control of robotic swarms. 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp 6047-6053.

[20] C. Nam, P. Walker, M. Lewis, and K. Sycara. Predicting trust in human control of swarms via inverse reinforcement learning. 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2017, pp 528-533.

[21] C. Nam, P. Walker, H. Li, M. Lewis, and K. Sycara. Models of Trust in Human Control of Swarms With Varied Levels of Autonomy. IEEE Transactions on Human-Machine Systems(2019)

[22] J. Penders, L. Alboul, U. Witkowski, A. Naghsh, J. Saez-Pons, S. Herbrechtsmeier, and M. El-Habbal. Advanced Robotics, vol 25, 2011, pp 93-117.

[23] D. Saldana, A. Prorok, S. Sundaram, M. FM. Campos, and V. Kumar. Resilient consensus for time-varying networks of dynamic agents. In American Control Conference (ACC), 2017. IEEE, pp. 252-258.

[24] K. Saulnier, D. Saldana, A. Prorok, G. J. Pappas, and V. Kumar. Resilient flocking for mobile robot teams. IEEE Robotics and Automation Letters 2, 2 (2017), pp. 1039-1046.

[25] A. Seiffert, S. Hayes, C. Harriott and J. Adams. Motion perception of biological swarms. Annual Meeting of the Cognitive Science Society, 2015.

[26] T. Setter, A. Gasparri, and M. Egerstedt. Trust-based interactions in teams of mobile agents. In American Control Conference 2016. pp. 6158-6163.

[27] A. Steyven, E. Hart, and B. Paechter. An investigation of environmental influence on the benefits of adaptation mechanisms in evolutionary swarm robotics. In Proceedings of the Genetic and Evolutionary Computation Conference. ACM 2017, pp. 155-162.

[28] P. Walker, M. Lewis and K. Sycara. Characterizing Human Perception of Emergent Swarm Behaviors, IEEE International Conference on Systems, Man, and Cybernetics, 2016.

[29] X. Wang and Y. Wang. Co-design of Control and Scheduling for HumanSwarm Collaboration Systems Based on Mutual Trust. In Trends in Control and Decision-Making for HumanRobot Collaboration Systems. Springer 2017, pp. 387-413.

[30] A. Xu and G. Dudek. Trust-driven interactive visual navigation for autonomous robots. In 2012 IEEE International Conference on Robotics and Automation. IEEE, pp. 3922-3929.

[31] A. Xu and G. Dudek. OPTIMo: Online Probabilistic Trust Inference Model for Asymmetric Human-Robot Collaborations. Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction, 2015, pp 221-228.

[32] F. Zhang and W. Chen. Self-healing for mobile robot networks with motion synchronization. In Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on. IEEE, pp. 3107-3112.

[33] L. Zhe, J. Jianjun, C. Weidong, F. Xiangyu and W. Hesheng. A Gradient-Based Self-Healing Algorithm for Mobile Robot Formation. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). 2015, pp 3395-3400.

[34] C. Zhu, S. Zhang, A. Dammann, S. Sand, P. Henkel, and C. Günther. Return-to-base navigation of robotic swarms in Mars exploration using DoA estimation, International Symposium on Electronics in Marine, 2013, pp 349-352.