Trust-Aware Behavior Reflection for Robot Swarm Self-Healing

Fan Jia

CMU-RI-TR-21-41

August 2021

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Katia Sycara, Chair
Jean Oh
Wenhao Luo

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

For my family and friends

Abstract

The deployment of robot swarms is influenced by real-world factors, such as motor issues, sensor failure, and wind disturbances. These factors cause the appearance of faulty robots. In a decentralized swarm, sharing incorrect information from faulty robots will lead to undesired swarm behaviors, such as swarm disconnection and incorrect heading directions. We envision a system where a human operator is exerting supervisory control over a remote swarm by indicating changes in trust to the swarm via a "trust-signal". By correcting faulty behaviors, trust between the human and the swarm is maintained to facilitate human-swarm cooperation. In this research, a trust-aware behavior reflection method – ***Trust-R*** – is designed based on a weighted mean subsequence reduced algorithm (WMSR). By using Trust-R, detected faulty behaviors are automatically corrected by the swarm in a decentralized fashion by referring to the motion status of their trusted neighbors and isolating failed robots from the others. Based on real-world scenarios, three types of robot faults – degraded performance caused by motor wear, abnormal motion caused by system uncertainty and motion deviation caused by an external disturbance such as wind – were simulated to test the effectiveness of Trust-R. Results show that Trust-R is effective in correcting swarm behaviors for swarm self-healing.

Acknowledgments

Firstly, I am extremely grateful to my advisor Prof. Katia Sycara for her invaluable support, patience, and insightful suggestions. Her encouragement, immense knowledge, and guidance lead to the development of this thesis. Her expertise in multi-agent systems, enthusiasm to work, and trust in students make her the best advisor.

I would also like to thank Prof. Jean Oh and Dr. Wenhao Luo for their valuable feedback and encouragement as members of my thesis committee.

Apart from that, I would like to thank all my lab mates in the Advanced Agent - Robotics Technology Lab for helping me work on various projects. I am specially grateful to Dr. Rui Liu and Dr. Wenhao Luo for their collaboration and contributions of this project.

Finally, I am deeply grateful to my family for their unconditional love.

Contents

List of Figures

Chapter 1

Introduction

Robot swarms use simple, local control laws to achieve a desired global emergent behavior over time. In using only local information, these systems are flexible to changes in the environment conditions and swarm size. The scalability of robot swarms leads to their use in a variety of applications, such as search and rescue [1], disaster relief [17], and environmental monitoring [8].

In many of these applications, human operators use supervisory control interfaces to remotely adjust mission goals and requirements. Human trust in the swarm is critical for effective human-swarm cooperation [21][6]. When a swarm is untrusted due to its unsatisfied performance, unnecessary interventions, such as diverting swarms' paths through the mission space and assigning additional intermediary spots to pass by, from human operators will increase. Unnecessary interventions are often time-consuming and require the attention of a group of robots to receive new inputs from the human operator, leading to delayed goal attainments or a decreased efficiency in human-swarm cooperation [18]. While, when a swarm is trusted, human operators are more willing to rely on automation to perform tasks, thereby reducing unnecessary interventions [12].

However, real-world faults, such as motor degradation, sensor failure or wind disturbance, make maintaining trust between humans and swarms challenging [13][5]. These factors can cause undesirable and uncontrollable robot behavior, such as a robot or group of robots getting disconnected as shown by robot 1 in Figure 1.1. In addition, faulty robots may share incorrect information with other members of

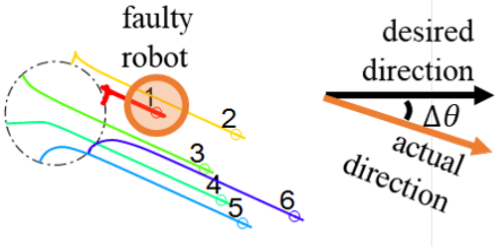


Figure 1.1: Undesired swarm flocking caused by faulty robots sharing incorrect information with their neighbors.

the swarm leading to incorrect behaviors of the swarm as a whole (Figure 1.1). The unpredictable nature of these real-world faults can reduce human trust in the reliability of the swarm. Unlike centralized systems where these faults can be directly corrected by centralized control commands, decentralized systems have a difficult time identifying these issues and are more susceptible to the effect faults have on the overall system behavior [7][2].

We envision a system where the human operator assumes a supervisory control role over the remote swarm. In such a scenario, the tolerance levels for faulty and failed robot behaviors are prescribed by the human's interpretation of the application requirements. Using these prescribed tolerance levels, the human operator monitors the performance of the swarm. It has been found in the literature that if the swarm performance decreases e.g. due to faulty robots, the operator's trust also decreases [18][12]. The operator indicates changes in trust (in particular trust decrease) to the swarm via, a so-called "trust-signal". The trust signal also contains information about expected swarm behavior and any current swarm deviations from that behavior. Therefore, it is critical to correct these faulty behaviors in a timely manner to ensure a high level of trust is maintained between the human and swarm. In this paper, a decentralized trust-aware behavior reflection (Trust-R) method is proposed to correct swarms' faulty behaviors. With Trust-R correction, swarms with faulty robots can repair themselves to attain human-assigned goals and still receive high-level human trust with minimum unnecessary interventions. The contributions are two-fold. First, a novel trust-aware reflection (Trust-R) algorithm is presented to help robots with a semi-automated, self-behavior diagnosis. Instead of merely judging whether it is normal or abnormal, each robot identifies its level of faultiness from a human trust perspective. Second, a reflective correction method is designed. Robots leverage the communicated levels of faultiness from their neighbors to update their motion status using only the information received from their trusted neighbors. As a result, information exchange is encouraged with trusted robots and discouraged with untrusted robots resulting in behavior correction of the whole swarm.

In this thesis, similar to previous self-healing work [16], we assume the faulty robots are a minority. Thus, it is possible to correct the swarm behavior by following the trusted robots.

Chapter 2

Related Work

Recent swarm self-healing research has focused on simulating faulty robot behaviors. In [22], faulty robots were defined and simulated as the robots not located in the desired position defined by a swarm's network topology. In [16], faulty robots were defined and simulated as robots with incorrect heading directions. By comparing observed behavior with ideally designed robot behaviors, the faulty behaviors were detected and corrected. These two methods are effective in swarm healing. However, neither considers the presence of a range of real-world factors, such as sensor failures or wind disturbance. These factors can greatly influence swarm behaviors in the real world and cannot be ignored. In our trust-R method, frequently observed faults, such as degraded motors, system uncertainty, and wind disturbance, were considered, showing our method to be general and suitable for real world environments. Moreover, Trust-R has the potential to support adaptive swarm deployments.

Additional emphasis has been placed on passive healing strategies that increase the swarm resilience. [16][23][14][9][10]increased the tolerance of faulty robots in the swarm by encouraging larger network robustness. As a result the negative influence of faulty robots on the swarm can be limited. Although this method is able to dilute the negative influence, the passive strategies usually require relatively high swarm connectivity and required specification of tolerable speed values which may be difficult to specify in advance. Therefore, it is necessary to actively correct these faulty behaviors when they appear. When faulty robots mislead the normal robots in a swarm, the proposed Trust-R method corrects the faulty robots by referring to those who are trusted. The failed robots are isolated by lowering the communication quality between the failed robots and the others. Trust-R can correct faulty behaviors that cannot be prevented by other techniques that aim to increase swarm resilience. Combining the passive prevention methods and the Trust-R active correction method, would make a swarm more robust.

In [3][20], faulty behaviors, such as fixed heading directions, were identified by tracking the temporal motion trajectory of a robot. However, the severity (i.e., effect on the whole swarm's behavior) of the faulty robots in different scenarios was not assessed. Without assessment of the severity, appropriate control strategies are difficult to design. In the proposed Trust-R method, degree of severity is diagnosed. Different control strategies, such as "take trusted robots as reliable information sources", "correct the robots with mild fault", "isolate the failed robots from the swarm", are then designed for exchanging information and adjusting connectivity among robots.

Chapter 3

Problem Formulation

We study a problem that the swarm performs "distributed biased flocking" in an obstacle free environment with some faulty robots. Figure 1.1 shows an example that faulty robots may share incorrect information with other members of the swarm leading to incorrect behaviors of the swarm as a whole. Our goal is effectively correct the swarm faulty behaviors. In this thesis, our assumption is the faulty robots are a minority. Thus, it is possible to correct the swarm behavior by following the trusted robots.

Consider a robot swarm of n holonomic robots with positions $X_i \in \mathbb{R}^3$, where $X_i = (\boldsymbol{x}_{i,\mathrm{h}}, \boldsymbol{x}_{i,\mathrm{v}}, \theta_i)$. Each robot is assigned a unique identifier (UID) $i \in \{1, 2, ..., n\}$. The communication graph is given by $\boldsymbol{G} = (\boldsymbol{\mathcal{V}}, \mathcal{E})$. 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 (i.e., $(v_i, v_j) \in \mathcal{E} \Rightarrow (v_j, v_i) \in \mathcal{E}$). The dynamic model [11] for each robot is defined as follows. A robot i is controlled by the linear velocity $\boldsymbol{u}_i^v$ and angular velocity $\boldsymbol{u}_i^w$ generated by motors. $\boldsymbol{x}_i$, θ_i denotes horizontal and vertical positions, and orientation state respectively.

$$\dot{\boldsymbol{x}}_{i,h} = \boldsymbol{u}_i^{\mathrm{v}} cos(\theta^i) \tag{3.1}$$

$$\dot{\boldsymbol{x}}_{i,\mathrm{v}} = \boldsymbol{u}_i^{\mathrm{v}} sin(\theta^i) \tag{3.2}$$

$$\dot{\boldsymbol{\theta}}_i = \boldsymbol{u}_i^w \tag{3.3}$$

Traditional biased flocking rules update the motion status of a robot i at time step t as follows

$$\boldsymbol{u}_i[t+1] = \frac{1}{N_i + 1}(\boldsymbol{u}_i[t] + \sum_{j \in N_i} \boldsymbol{u}_j[t]) \tag{3.4}$$

As seen from the distributed update method above, 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.

In this thesis, "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 "influential factors".

Chapter 4

Methodology

The overall architecture of our self-healing method is shown in Figure 4.1. Based on a human's trust signal that also indicates human's diagnosis and level of faults, (e.g. low, medium or high, of the fault), each robot determines its strategy of communication between itself and its neighbors. When faulty robots appear in a swarm, it becomes unreliable to update a robot's status by considering in the faulty robots' motion status (calculated by Equation 3.4) [4]. 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) [15]. Instead of merely averaging values as in the previous update method, our Trust-R method updates information differentially based on the communication quality (Equation 4.1). Weights w_i are calculated in Sections 3.2.2 and 3.2.3.

$$u_i[t+1] = w_i[t]u_i[t] + \sum_{j \in N_i} w_j[t]u_j[t] \tag{4.1}$$

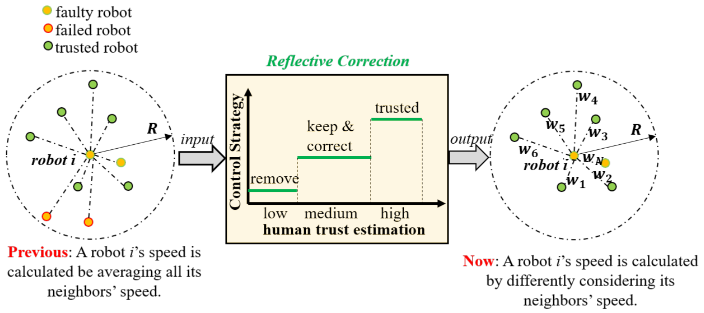


Figure 4.1: Illustration of change in speed calculation in the Trust-Aware Behavior Reflection for Swarm Self-Healing.

4.1 Human Trust in Faulty Behaviors

Abnormal robot behaviors inside a swarm decrease human trust in the swarm [12]. A human operator knows the swarm behavior requirements of the mission she is pursuing, such as requirements of connectivity and heading direction, and therefore can estimate a relation between current performance of the swarm and expected performance. $\delta(u_{actual}, u_{expect})$ is defined as the difference between expected speed/heading direction and actual speed/heading direction. u_{expect} is calculated using Equation 4.1 by referring to a robot's neighbors. u_{actual} is read directly from a robot's motion sensors. δ is used to calculate the trust score. If $\delta(u_{actual}, u_{expect})$ is smaller than a small percentage β_1, robot behaviors are normal with "high trust". If $\delta(u_{actual}, u_{expect})$ is larger than a small percentage β_1 and smaller than a big percentage β_2, robot behaviors are faulty with "medium trust". If $\delta(u_{actual}, u_{expect})$ is larger than the big percentage β_2, robot behaviors have completely failed with "low trust". The β values are found by examining the differences between the speed and heading-direction according to UAV control requirements in different scenarios.

4.2 Trust-Aware Communication Quality Assessment

The overall communication graph for robot i is $\boldsymbol{\mathcal{E}} = \{(i, j) \mid j \in N_i\}$. 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 4.2. 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.

$$f_{ij} = \begin{cases} 0 & ||\boldsymbol{x}_i - \boldsymbol{x}_j|| \geq R \\ \frac{1}{2}(g_i + g_j)\eta & ||\boldsymbol{x}_i - \boldsymbol{x}_j|| \leq \rho \\ \frac{(g_i+g_j)\eta}{2} \exp \frac{-\gamma(||\boldsymbol{x}_i - \boldsymbol{x}_j|| - \rho)}{R-\rho} & otherwise \end{cases} \tag{4.2}$$

where g_i is the trust level of 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 4.2. 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 4.2 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 4.2, 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 $\rho = 4m$.

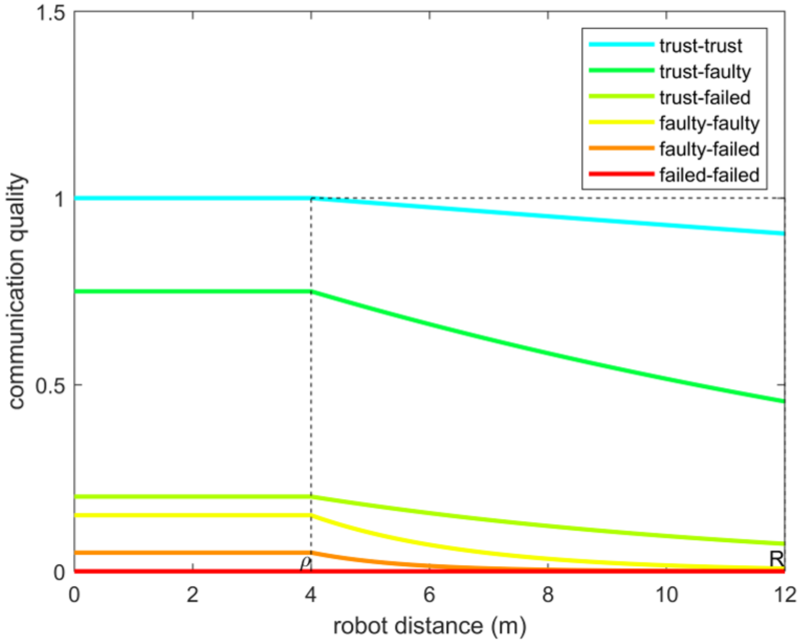


Figure 4.2: Illustration of the Trust-aware communication quality assessment. Information shared by trusted robots is encouraged with higher upper limits, while untrusted information is discouraged with lower upper limits.

The adjacency matrix, $\boldsymbol{A}$, that describes the communication graph is given by:

$$[\boldsymbol{A}]_{ij} = \begin{cases} 0 & i \neq j \\ f_{ij} & i = j. \end{cases} \tag{4.3}$$

The degree matrix, $\boldsymbol{D}$, is:

$$[\boldsymbol{D}]_{ij} = \begin{cases} 0 & i \neq j \\ \sum_j f_{ij} & i = j. \end{cases} \tag{4.4}$$

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

$$[\boldsymbol{L}]_{ij} = \begin{cases} -f_{ij} & i \neq j \\ \sum_j f_{ij} & i = j. \end{cases} \tag{4.5}$$

The eigenvalues $\{\boldsymbol{\lambda}_i \mid i = 1, 2, ..., n\}$ of $\boldsymbol{L}$ are real and they satisfy $0 = \boldsymbol{\lambda}_1 \leq \boldsymbol{\lambda}_2 \leq \ldots \leq \boldsymbol{\lambda}_n$. The connectivity measure $\boldsymbol{\lambda}_2$ is estimated by the equation $\boldsymbol{L}e_2 = \boldsymbol{\lambda}_2 e_2$ and the eigenvector e_2.

4.3 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 4.4 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.

$$w_k[t] = \frac{\hat{f}_k[t]}{\hat{f}_i[t] + \sum_{j \in N_i} \hat{f}_j[t]}, k \in [i, N_i] \tag{4.6}$$

Weights for updating each robot's status are calculated by Equations 4.1 and 4.6. The result of the weighted update mechanism is shown on the right side of Figure 4.1. For updating a robot i, weights w_k are calculated by normalizing all the communication quality values in a communication range, shown in equation 4.6. 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 $\boldsymbol{u}_v^i$ and $\boldsymbol{u}_w^i$ for robot motors are changed to $\boldsymbol{u}_{v,trust}^i$ and $\boldsymbol{u}_{w,trust}^i$. The gains K_v and K_w are parameters for adjusting the motor output.

$$\boldsymbol{u}_{i,trust}^{\mathrm{v}} = (K_{\mathrm{v}} + K_{\mathrm{v},trust})(\mathbf{v}_i + \boldsymbol{q}_{\boldsymbol{N_i}})^T \boldsymbol{b}_i \tag{4.7}$$

$$\boldsymbol{u}_{i,trust}^{w} = (K_w + K_{w,trust})\big(\boldsymbol{\gamma}_i + \phi(\boldsymbol{b}_i, \boldsymbol{q}_{N_i})\big) \tag{4.8}$$

Let $\boldsymbol{u}_i[t+1]$ 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 $\boldsymbol{u}_{i,trust}[t+1]$. 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 the human-trusted robot speeds.

$$K_{\mathrm{v},trust}[t+1] = \frac{\boldsymbol{u}^{\mathrm{v}}_{i,trust}[t] - \boldsymbol{u}^{\mathrm{v}}_{i}[t]}{\boldsymbol{u}^{\mathrm{v}}_{i}[t]} \tag{4.9}$$

$$K_{w,trust}[t+1] = \frac{\boldsymbol{u}^{w}_{i,trust}[t] - \boldsymbol{u}^{w}_{i}[t]}{\boldsymbol{u}^{w}_{i}[t]} \tag{4.10}$$

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 $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. The overall swarm safety is maintained during the correction period $[0, T]$ by maintaining safety distance $h^{safe}_{i,j}$ for any robot pair i and j. $\mathcal{H}^{safe}_{i,j}$ is the set of all safe distances.

$$h_{ij,safe}(t) = ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 - \rho^2, \forall i, j \tag{4.11}$$

$$\mathcal{H}_{ij,safe}[t] = \Big\{\boldsymbol{x} \in \mathbb{R}^2, t \in [0, T] : h^{safe}_{ij}(t) \geq 0\Big\} \tag{4.12}$$

4.4 Trust-Aware Connectivity Maintenance for Motion Consensus

To further correct faulty swarm behaviors, connectivities between a faulty robot and the other trusted robots are strengthened by Equation 4.13. The graph's Laplacian matrix is $\boldsymbol{L}$ and the algebraic connectivity is $\boldsymbol{\lambda}_2$. The connectivity (described by the second smallest eigenvalue $\boldsymbol{\lambda}_2$ of $\boldsymbol{L}$, e_2 is the corresponding eigenvector) between a faulty – correctable – robot and a neighboring – trusted – robot is improved by reducing the distance between the faulty robot and the trusted neighbor. The more reliable information provides a mechanism for correcting the faulty robot's behavior. $\boldsymbol{x}_{i,\psi}$ is the position component on direction ψ (horizontal or vertical direction) for the robot i. $\frac{\alpha \boldsymbol{L}(\boldsymbol{x})}{\boldsymbol{\alpha x}_{i,\psi}}$ is computed by calculating the difference between the reliability values, f_{ij}, at adjacent time steps as shown in Equation 4.14.

$$\boldsymbol{u}_i = \bigtriangledown_{i,\psi} \boldsymbol{\lambda}_2 \tag{4.13}$$

$$= \frac{\boldsymbol{\alpha}\boldsymbol{\lambda}_2(\boldsymbol{L})}{\boldsymbol{\alpha}\boldsymbol{x}_{i,\psi}} = \frac{\boldsymbol{\alpha}\boldsymbol{\lambda}_2(\boldsymbol{L})}{\boldsymbol{\alpha}\boldsymbol{L}(\boldsymbol{x})}\frac{\boldsymbol{\alpha}\boldsymbol{L}(\boldsymbol{x})}{\boldsymbol{\alpha}\boldsymbol{x}_{i,\psi}} = \textbf{Trace}\left\{\left[\frac{e_2e_2^T}{e_2^Te_2}\right]^T\left[\frac{\boldsymbol{\alpha}\boldsymbol{L}(\boldsymbol{x})}{\boldsymbol{\alpha}\boldsymbol{x}_{i,\psi}}\right]\right\} \tag{4.14}$$

Theorem 1 *The novel method, Trust-R, reduces untrusted information and encourages the trusted information among robots.*

Proof. For two robots $a, b \in \boldsymbol{G}$, a is an abnormal robot, while b is a normal robot. To update the motion status of a target robot i, weights of its neighbors a and b $(a, b \in N_i)$ for information exchange are w_a and w_b respectively.

$$w_a[t] = \frac{\hat{f}_a[t]}{\hat{f}_i[t] + \sum_{j\in N_i}\hat{f}_j[t]}, w_b[t] = \frac{\hat{f}_b[t]}{\hat{f}_i[t] + \sum_{j\in N_i}\hat{f}_j[t]}$$

An abnormal robot's trust level is lower than that of a normal robot. As a result, $\hat{f}_a[t] \leq \hat{f}_b[t] \Rightarrow w_a[t] < w_b[t]$. Therefore, with trust awareness, Trust-R reduces the untrusted information given by abnormal robots, while encouraging the sharing of trustworthy information from normal robots.

Theorem 2 *The novel Trust-R method encourages a relatively closer distance between a robot and other trusted robots, and encourages a relatively farther distance between a robot and other untrusted robots. The adjustment will be reduced to zero once the flocking consensus is reached.*

Proof. When using the trust-aware communication quality to adjust the distance of robot i to other robots, the adjustment along a direction ψ is

$$\boldsymbol{u}_i = \textbf{Trace}\left\{\left[\frac{e_2e_2^T}{e_2^Te_2}\right]^T\left[\frac{\boldsymbol{\alpha}\boldsymbol{L}(\boldsymbol{x})}{\boldsymbol{\alpha}\boldsymbol{x}_{i,\psi}}\right]\right\} = \textbf{Trace}\left\{\left[\frac{e_2e_2^T}{e_2^Te_2}\right]^T\left[\frac{\boldsymbol{\alpha}[\boldsymbol{L}]_{ij}}{\boldsymbol{\alpha}\boldsymbol{x}_{i,\psi}}\right]\right\}$$

For the off-diagonal elements in the Laplacian matrix, $\boldsymbol{L}$, $\frac{\boldsymbol{\alpha}[\boldsymbol{L}]_{ij}}{\boldsymbol{\alpha}\boldsymbol{x}_{i,\psi}}$ is solved by

$$\sum_K -\frac{\boldsymbol{\alpha}f_{ij}}{\boldsymbol{\alpha}x_{k,\psi}}\boldsymbol{u}_{k,\psi} = \frac{\boldsymbol{\alpha}f_{ij}}{\boldsymbol{\alpha}x_{i,\psi}}(\boldsymbol{u}_{j,\psi} - \boldsymbol{u}_{i,\psi})$$

For the diagonal elements in $\boldsymbol{L}$, $\frac{\boldsymbol{\alpha}[\boldsymbol{L}]_{ij}}{\boldsymbol{\alpha x}_{i,\psi}}$ is solved by

$$\sum_k \left(\sum_j \frac{\boldsymbol{\alpha} f_{ij}}{\boldsymbol{\alpha x}_{k,\psi}} \right) \boldsymbol{u}_{k,\psi} = \sum_j \frac{\boldsymbol{\alpha} f_{ij}}{\boldsymbol{\alpha x}_{i,\psi}} (\boldsymbol{u}_{i,\psi} - \boldsymbol{u}_{j,\psi})$$

Since

$$\frac{\boldsymbol{\alpha} f_{ij}}{\boldsymbol{\alpha x}_{i,\psi}} = -\frac{\gamma\eta(g_i + g_j)(\boldsymbol{x}_{i,\psi} - \boldsymbol{x}_{j,\psi})}{2(R-\rho)||\boldsymbol{x}_i - \boldsymbol{x}_j||} \exp \frac{-\gamma(||\boldsymbol{x}_i - \boldsymbol{x}_j|| - \rho)}{R - \rho}$$

$\frac{\boldsymbol{\alpha} f_{ij}}{\boldsymbol{\alpha x}_{i,\psi}}$ is bounded by the robot distance which is smaller than communication radius R. For a desired flocking direction $\boldsymbol{q}_0$, the adjustment degree $\boldsymbol{u}_i$, between two robots i and j, is positively correlated with their average trust score $\frac{(g_i+g_j)\eta}{2}$. A larger trust score leads to a larger adjustment. Therefore, Trust-R encourages a relatively closer distance between a robot and other trusted robots, and encourages relatively farther distances to abnormal robots. The abnormal faulty robots are gradually abandoned by the swarm.

As robots reach the motion consensus along the heading direction, $\boldsymbol{u}_i$ will be equal to $\boldsymbol{u}_j$ within a limited time. Therefore, $\frac{\boldsymbol{\alpha} f_{ij}}{\boldsymbol{\alpha x}_{i,\psi}}$ will be 0, stopping the adjustment when the consensus is reached.

Chapter 5

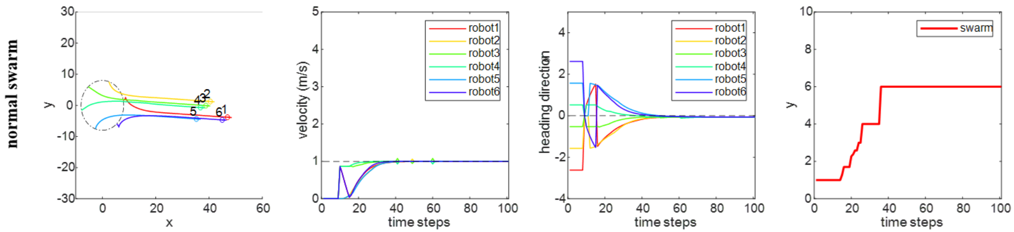
Simulation Results

To validate the effectiveness of Trust-R in helping the swarm self-heal, three real-world faults were simulated using MATLAB: a degraded motor, system uncertainty (sensor dysfunction) and wind disturbance. These faults commonly happen in complex environments, such as densely distributed forests/buildings and extreme weather conditions, which can affect robot communication, spatial distributions and system reliability [5][19]. Our goal in using Trust-R is to repair the untrusted swarm misled by faulty robots by improving the swarms' environmental adaptation. The task for the swarm in all experiments is distributed biased flocking. All results reported are the performance of the system for a single run. The non-stochastic nature of the algorithm results in the same behavior each time a specific parameter configuration is run.

To focus on "correction" of faulty swarms with different faults and to reduce the difficulties in analyzing the behaviors of individual robots, the number of robots was set to a small number – 6 – and the biased heading-direction was fixed to "East". The initial number of faulty/failed robots for each scenario was chosen to be either 1 or 2 robots. Under the influence of these abnormal robots, several of the 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 initialized 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.

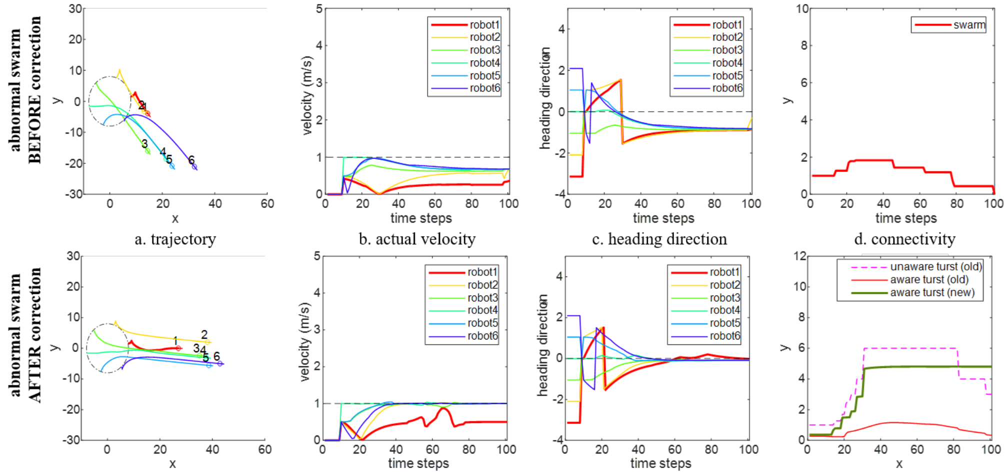
5.1 Limited Speed – Degraded Motor(s)

Due to a degraded motor, the speed of a robot was constrained such that the velocity and angular speeds could not reach the designed speeds. In this case, the speed of the faulty robots was lower than the normal robots. Because of the exchange of motion statuses with the faulty robots, the speed of some robots was influenced. The upper limits of linear velocity and angular velocity were set as $0.5m/s$ and $0.5rad/s$. For comparison, normal biased flocking without faulty robots was simulated as a baseline.



a. trajectory b. actual velocity c. heading direction d. connectivity

Figure 5.1: System response given by normal flocking. With a distributed control method, the swarm without a faulty robot flocks to the human assigned direction: "East". with consensus on both heading and motion.



a. trajectory b. actual velocity c. heading direction d. connectivity

e. trajectory f. actual velocity g. heading direction h. connectivity

Figure 5.2: System response given by untrusted flocking caused by a robot with faulty motor. Under the influence of faulty, robot 1, the motion consensus cannot be achieved.

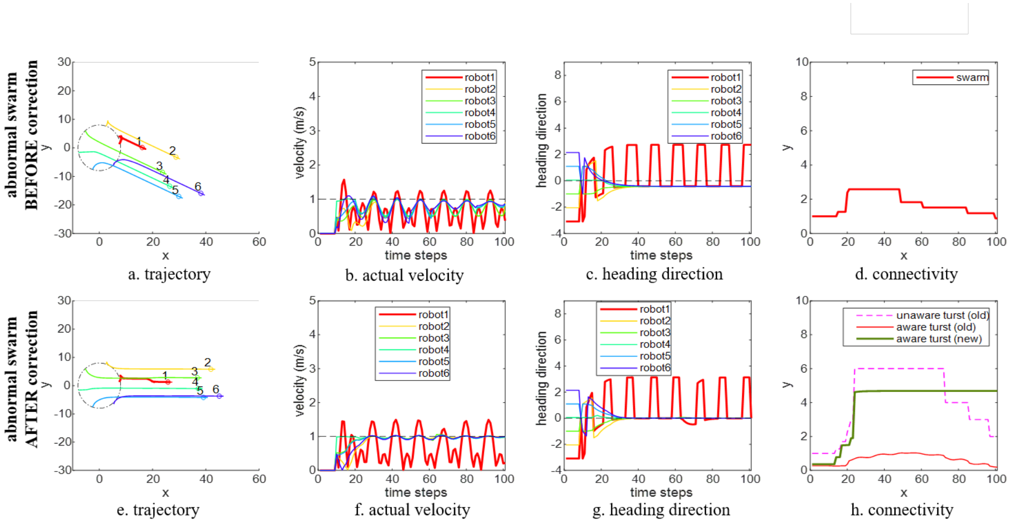
For a normal swarm (Figure 5.1), after about 38 time steps ($19s$), the velocity of all 6 robots achieved the desired consensus of $1.0m/s$; After about 50 time steps ($25s$), the heading of all 6 robots achieved consensus on the "East" direction. The connectivity $\boldsymbol{\lambda}_2$ was 6, which means all robots achieved the best communication in this scenario.

Figure 5.2 shows a scenario in which Robot 1 had a degraded motor. As shown, with the faulty robot in the swarm, the velocity consensus was not achieved within 100 time steps ($50s$), and the faulty Robot 1 was disconnected from the swarm (Figure 5.2(a,b)). The heading direction of the swarm shifted to $1rad$ after 64 time steps ($32s$) (Figure 5.2(c)). Connectivity with the faulty robot was decreased to 0 after 100 time steps ($50s$)(Figure 5.2(d)).

Trust-R provided robots in the swarm additional awareness by assessing the motion statuses of a robot and its neighbors. In this case, Robot 1, whose speed is 70% lower than the expected speed, was considered an untrusted and failed robot (Figure 5.2e), thereby decreasing swarm performance and human trust. The communication quality between Robot 1 and other normal robots decreased as calculated by the "trust-failed" curve in Figure 4.2. With the Trust-R correction, the information exchanged with Robot 1 was tightly constrained. After 70 time steps ($35s$), Robot 1 was disconnected from the normal robots. The swarm with only trusted robots achieved velocity consensus after 32 time steps ($16s$), and achieved consensus on heading after 50 time steps ($25s$) with only a -0.1 rad deviation (Figure 5.2(f,g)). This demonstrates that Trust-R was effective in correcting the faulty behaviors of the swarm. Shown as Figure 5.2(h), the connectivity, $\boldsymbol{\lambda}_2$, of the old swarm without Trust-R maintained a low-level of connectivity and decreased to 0 after 80 time steps ($40s$). In contrast, the swarm which constrained the information exchanged with the faulty Robot 1 had connectivity that increased to a high level of 4.8, showing the effectiveness of Trust-R in encouraging connectivity among trusted robots.

5.2 Abnormal Motion – System Uncertainty

Due to the system uncertainties such as sensor failures, lost GPS signals and internal disturbances from the mechanical systems, robots may show abnormal behaviors such as sinusoidal motion, random motion, or fixed-direction motion. For this case study,



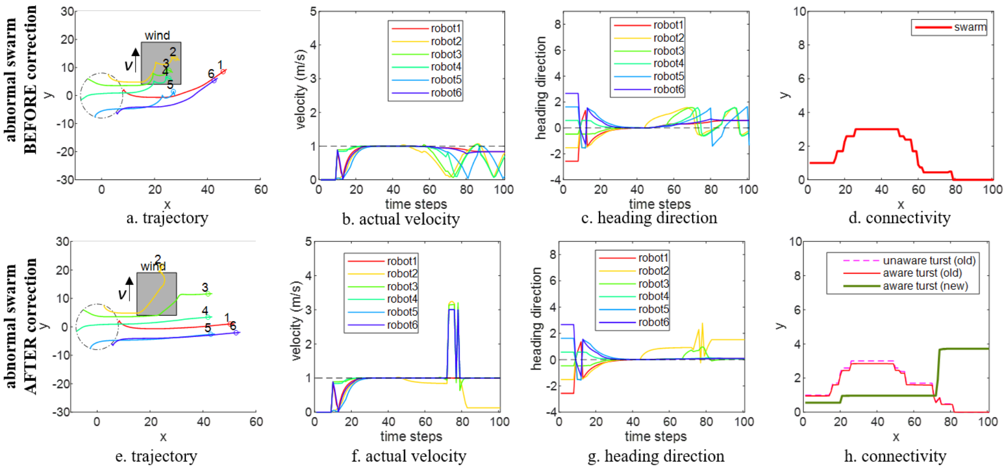
a. trajectory
b. actual velocity
c. heading direction
d. connectivity
e. trajectory
f. actual velocity
g. heading direction
h. connectivity

Figure 5.3: System response given by untrusted behaviors caused by system uncertainty (faulty sensor).

a sinusoidal motion was investigated. Robot 1 had abnormal sinusoidal velocity and angular velocity (shown as Figure 5.3(a,b)), with amplitude of $1.5m/s$. Without correction, the motion consensus was not achieved (Figure 5.3(b,c)). The connectivity decreased to 0.8 after about 100 time steps (50 seconds), shown in Figure 5.3(d). With the Trust-R correction, misleading information from the untrusted robot 1 was quickly constrained. The new swarm without faulty robots achieved velocity consensus after 30 time steps ($15s$) and achieved heading direction consensus after 30 time steps ($15s$) with $0rad$ deviation, shown in Figure 5.3(e,f,g). As shown in Figure 5.3(h), connectivity of the swarm without Trust-R correction remained low as a farther distance between the faulty Robot 1 and other trusted robots was encouraged. On the other hand, the connectivity of the swarm isolating the faulty Robot 1 increased to 4.9 by using Trust-R, showing again the effectiveness of Trust-R in correcting abnormal swarm behaviors.

5.3 Motion Deviation – Wind Disturbance

When some robots in a swarm cross into a wind zone, the wind will give the robots extra linear and angular velocity. 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 $0.25m/s$ linear velocity along the "North" direction and an angular deviation of $0.1rad/s$.



a. trajectory b. actual velocity c. heading direction d. connectivity

e. trajectory f. actual velocity g. heading direction h. connectivity

Figure 5.4: System response given by untrusted flocking caused by a wind disturbance.

As Figure 5.4(a) shows, Robots 2 and 3 crossed the wind region first. They then attracted Robots 4 and 5 into the wind zone. Without correction, a motion consensus was not achieved (Figure 5.4(a-c)). The connectivity decreased to 0 after about 78 time steps ($39s$)(Figure 5.4d). With the Trust-R correction, misleading information from the untrusted Robots 2 and 3 was quickly constrained and their influence on the other robots was largely reduced. The new swarm without faulty robots achieved velocity consensus after about 30 time steps ($15s$) and achieved heading direction consensus after 40 time steps (about $20s$), shown in Figure 5.4(e,f,g). As shown in Figure 5.4(h), connectivity of the old swarm without the Trust-R correction was decreased to 0 after 80 time steps ($40s$), due to the disconnection of the faulty Robot 2. In contrast, the connectivity of the new swarm (swarm after removing the faulty Robot 2) increased to 4, showing the effectiveness of Trust-R in correcting abnormal swarm behaviors caused by disturbances such as wind.

Once the wind disturbance has passed, the difference between the actual velocity and the expected velocity will decrease. If the robot is already disconnected, given the characteristics of the distributed control, the robot is no longer reachable and will be ignored by the swarm (behavior correction of robot 2 in this case). If the robot is still within the communication range of the other robots after the wind disturbance has passed, the previously faulty robot will then be considered a normal robot with a high communication quality. The new high-level communication can then be used to correct the previously faulty robot's behavior (robots 3 and 4).

Chapter 6

Conclusion and Future Work

We presented a trust aware behavior reflection method – Trust-R – to help a swarm self-heal. This allows the swarm to repair its overall behavior when faulty behaviors of its members occur. Three types of robot faults – limited performance caused by a degraded motor, abnormal motion caused by system uncertainty such as sensor failure and motion deviation caused by wind disturbance – were simulated. With Trust-R, the motion status of the robots are estimated and the corresponding communication quality is determined. The robots are encouraged to communicate with trusted robots and discouraged from communicating with untrusted robots. In doing so the negative influence caused by misleading information is largely reduced and swarm behaviors are corrected. The simulation results for the three faulty scenarios demonstrate the effectiveness of Trust-R in correcting a range of faulty swarm behaviors.

In this thesis, the relation between human trust and swarm performance was presumed linear. Future research will focus on user studies to assess the effects on trust of self-healing swarm strategies which preserve swarm performance despite observably faulty behavior of individual robots. Additional studies will be conducted to characterize the type and severity of faulty behaviors necessary to decrease a human's trust to the point of intervention despite self-healing behavior. In this thesis faults from different categories were remediated using the same mechanism in Trust-R which leads to expulsion of faulty robots from the swarm. In cases such as the wind disturbance where the robot's communications are faulty but not the robot itself or where neighboring robots might assist we hope to develop compensatory strategies to salvage those faulty robots we can. Lastly, we plan to develop signatures from faulty behavior detection data to analyze the temporal status of robot motion and interactions. Based on these models, faulty robot behaviors can be automatically detected during the swarm deployments and the time needed to mitigate or expel faulty robots can be reduced. More complex scenarios (e.g., obstacles) will also be considered. The system's ability to mitigate the effect of faulty and failed robots on the performance of the swarm in these complex scenarios will be evaluated.

Bibliography

[1] Hai Chen, Xin-min Wang, and Yan Li. A survey of autonomous control for uav. In *Artificial Intelligence and Computational Intelligence, 2009. AICI'09. International Conference on*, volume 2, pages 267–271. IEEE, 2009. 1

[2] Rafer Cooley, Shaya Wolf, and Mike Borowczak. Secure and decentralized swarm behavior with autonomous agents for smart cities. *arXiv preprint arXiv:1806.02496*, 2018. 1

[3] Yuan-Shun Dai, Michael Hinchey, Manish Madhusoodan, James L Rash, and Xukai Zou. A prototype model for self-healing and self-reproduction in swarm robotics system. In *Dependable, Autonomic and Secure Computing, 2nd IEEE International Symposium on*, pages 3–10. IEEE, 2006. 2

[4] Ali Jadbabaie, Jie Lin, and A Stephen Morse. Coordination of groups of mobile autonomous agents using nearest neighbor rules. *IEEE Transactions on automatic control*, 48(6):988–1001, 2003. 4

[5] Belkacem Khaldi, Fouzi Harrou, Foudil Cherif, and Ying Sun. Monitoring a robot swarm using a data-driven fault detection approach. *Robotics and Autonomous Systems*, 97:193–203, 2017. 1, 5

[6] Andreas Kolling, Phillip Walker, Nilanjan Chakraborty, Katia Sycara, and Michael Lewis. Human interaction with robot swarms: A survey. *IEEE Transactions on Human-Machine Systems*, 46(1):9–26, 2016. 1

[7] Eric J Leaman, Brian Q Geuther, and Bahareh Behkam. Hybrid centralized/decentralized control of bacteria-based bio-hybrid microrobots. In *2018 International Conference on Manipulation, Automation and Robotics at Small Scales (MARSS)*, pages 1–6. IEEE, 2018. 1

[8] Ali Marjovi and Lino Marques. Optimal spatial formation of swarm robotic gas sensors in odor plume finding. *Autonomous Robots*, 35(2-3):93–109, 2013. 1

[9] Nithin Mathews, Anders Lyhne Christensen, Rehan O'Grady, Francesco Mondada, and Marco Dorigo. Mergeable nervous systems for robots. *Nature communications*, 8(1):439, 2017. 2

[10] Nithin Mathews, Anders Lyhne Christensen, Alessandro Stranieri, Alexander Scheidler, and Marco Dorigo. Supervised morphogenesis: Exploiting morphological flexibility of self-assembling multirobot systems through cooperation with aerial robots. *Robotics and Autonomous Systems*, 112:154–167, 2019. 2

[11] Sasanka Nagavalli, Nilanjan Chakraborty, and Katia Sycara. Automated sequencing of swarm behaviors for supervisory control of robotic swarms. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2674–2681. IEEE, 2017. 3

[12] Changjoo Nam, Phillip Walker, Michael Lewis, and Katia Sycara. Predicting trust in human control of swarms via inverse reinforcement learning. In *Robot and Human Interactive Communication (RO-MAN), 2017 26th IEEE International Symposium on*, pages 528–533. IEEE, 2017. 1, 1, 4.1

[13] James O'Keeffe, Danesh Tarapore, Alan G Millard, and Jon Timmis. Towards fault diagnosis in robot swarms: An online behaviour characterisation approach. In *Conference Towards Autonomous Robotic Systems*, pages 393–407. Springer, 2017. 1

[14] Andrzej Pelc and David Peleg. Broadcasting with locally bounded byzantine faults. *Information Processing Letters*, 93(3):109–115, 2005. 2

[15] David Saldana, Amanda Prorok, Shreyas Sundaram, Mario FM Campos, and Vijay Kumar. Resilient consensus for time-varying networks of dynamic agents. In *American Control Conference (ACC), 2017*, pages 252–258. IEEE, 2017. 4

[16] Kelsey Saulnier, David Saldana, Amanda Prorok, George J Pappas, and Vijay Kumar. Resilient flocking for mobile robot teams. *IEEE Robotics and Automation Letters*, 2(2):1039–1046, 2017. 1, 2

[17] Nathan Schurr, Janusz Marecki, Milind Tambe, Paul Scerri, Nikhil Kasinadhuni, and John P Lewis. The future of disaster response: Humans working with multiagent teams using defacto. In *AAAI spring symposium: AI technologies for homeland security*, pages 9–16, 2005. 1

[18] Tina Setter, Andrea Gasparri, and Magnus Egerstedt. Trust-based interactions in teams of mobile agents. In *American Control Conference*, pages 6158–6163, 2016. 1, 1

[19] Andreas Steyven, Emma Hart, and Ben 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*, pages 155–162. ACM, 2017. 5

[20] Danesh Tarapore, Anders Lyhne Christensen, and Jon Timmis. Generic, scalable and decentralized fault detection for robot swarms. *PloS one*, 12(8):e0182058, 2017. 2

[21] Xiaotian Wang and Yue Wang. Co-design of control and scheduling for human–swarm collaboration systems based on mutual trust. In *Trends in Control and Decision-Making for Human–Robot Collaboration Systems*, pages 387–413. Springer, 2017. 1

[22] Fei Zhang and Weidong Chen. Self-healing for mobile robot networks with motion synchronization. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 3107–3112. IEEE, 2007. 2

[23] Haotian Zhang and Shreyas Sundaram. Robustness of information diffusion algorithms to locally bounded adversaries. In *American Control Conference (ACC), 2012*, pages 5855–5861. IEEE, 2012. 2