Heterogeneity of Faults in a Robot Swarm: Identifying Discriminatory Metrics

Suet Lee[1], and Sabine Hauert[2]
Department of Engineering Mathematics, Bristol Robotics Laboratory,
University of Bristol, United Kingdom
Email: [1]suet.lee@bristol.ac.uk, [2]sabine.hauert@bristol.ac.uk

***Abstract*— Robot swarms operating in real-world conditions will experience a variety of faults, essentially making the swarm heterogeneous over the lifetime of its operations. Our previous work presented a method to identify metrics which can discriminate effectively between normal and faulty states of a robot in the swarm. Here we present our approach to faulty state discrimination through the lens of measuring diversity: can diversity be evaluated through discrimination of states of a system, and can we identify discriminatory metrics to apply to real-time diversity evaluation?**

I. INTRODUCTION

Homogeneity is often considered to be a defining characteristic in the swarm paradigm [1]. In simulation, physical homogeneity can be imposed on the system. In a real-world setting however, heterogeneity arises naturally from mechanical differences in the sensors and actuators of the robots. In particular, faults will inevitably occur over the course of system run-time limiting the capabilities of certain individuals and giving rise to a level of heterogeneity.

We thus see a parallel between the discourse in heterogeneous multi-robot research and faulty swarm systems. In particular, metrics of diversity could be useful in evaluating failure in multi-robot/swarm systems, and vice-versa. From the lens of safety, we have been interested in detecting faults in swarm robot systems. There are many approaches to fault detection and diagnosis, and it is widely studied in other domains under labels of execution monitoring or anomaly detection [2].

The common features in these approaches are the identification, separation and characterization of distinct states of the system. We focus on separation and characterization of two states of a robot swarm: a "normal", faultless state of operation, and a faulty state of operation. In swarm robotics, there are additional dimensions to the fault detection problem arising from two properties of the swarm:

1) The swarm is decentralized: individuals act based on local knowledge of their environment.
2) Local interactions of individuals give rise to emergent behaviours which can be difficult to predict and model.

Together these two properties motivate a decentralized, data-driven method of fault detection where individual robots self-detect faults based on measures of their local environment – either in the range of sensing or on-board. We summarize results for this method which have been reported in previous work published in [3], demonstrating a method for the automatic extraction of measures, or *metrics*, with high discriminatory power between states.

Finally, we discuss how the same method may be useful to measure diversity in a heterogeneous system - can metrics based on local sensing discriminate between a range of given states and can we build a model for the real-time evaluation of diversity in the system?

II. RELATED WORK

We have discussed two states of the swarm, *normal* versus *faulty*, but this is a gross simplification of the possible states of fault. Firstly, faults can be differentiated into categories based on fault type. Secondly, a fault can exist on a gradient - robots moving at a fraction of maximum speed for example. The "bipolar" classification of state as either homogeneous or heterogeneous has been challenged in previous work which presents a quantitative metric of diversity: *hierarchic social entropy* [4]. This is one of many diversity metrics, or indices, which have applications beyond multi-agent robotics: previous work has evaluated a range of the most relevant diversity indices across domains together with their information-theoretic counterparts, with considerations for the impact of sample size [5]. These are metrics evaluated at the *global* level of the system.

In the work we present, we demonstrate a method for extracting discriminatory metrics at the *local* level, in an intra-logistics use-case scenario, which may be used in combination to detect a variety of fault types in a swarm [3]. Faults can be classified as topology or component related [6]. A communication link fault is an example of the former; sensor and actuator faults examples of the latter. We can classify metrics in the same way: for example, component metrics could capture sensor data and topology metrics could capture the distance between robots. We propose an automatic method for metric extraction which could be applied more broadly to different scenarios, looking at both topology and component metrics in a univariate analysis.

III. METHOD

A. Scenario

Swarms have the potential to be used out-of-the-box for intralogistics in areas that have not yet adopted robotics, such as SMEs, or in messy real-world environments [7].

In our use-case scenario, the robots operate in a 5m x 5m bounded arena, referred to as the warehouse: the task is to retrieve and deliver boxes to the drop-off zone, a 25cm-wide vertical strip extending along the length of the right-hand wall. Additionally, the boxes are raised up on tables which the robots can detect and navigate under in order to lift. Robots move stochastically and are able to detect objects (robot, box or wall) via ArUco tags. Following previous work in our team, the parameters of the scenario have been selected to match as closely as possible the robot platform and the arena we have available, our aim being to close the reality gap with real-world tests [8], [9]. Table I and figure 1 summarize the configuration.

TABLE I: Configuration

	Property	Value
Warehouse	Dimensions	500 cm x 500 cm
	Number of boxes	10
	Number of robots	10
	Box diameter	25 cm
Robot	Diameter	25 cm
	Cameras	4 x 120 FOV video cameras equidistant on perimeter, 1 x 120 FOV video camera upward-facing to detect boxes
	Camera range	50 cm
	Robot max speed	200 cm/s (real-time)

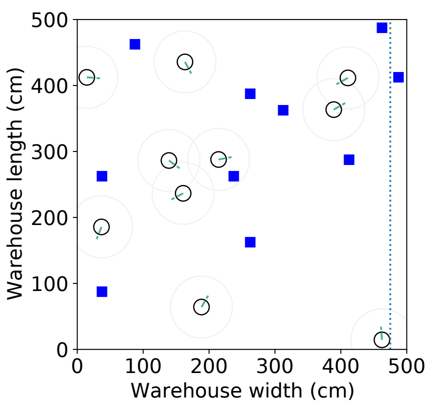

Fig. 1: Warehouse setup: boxes are represented by blue squares, robots by circles. A green pointer shows the current heading. The drop-off zone is marked by a dashed vertical line on the right side of the arena.

B. Simulation environment

We run experiments in a 2D physics-based simulator written in python[1]. The main considerations in developing the simulation test environment include:

- Robot motion: the robots move stochastically and choose a new heading at random at a rate of once per 0.2 seconds.
- Initialization: boxes and robots are placed at random in the arena.
- Table box carrier: we abstract the table carriers so that robots only interact with a “box”.
- Robot sensors: we abstract sensing capabilities and assume a 50 centimeter camera range given the size of the arena. The camera is assumed to perform both object detection and avoidance.

[1]https://bitbucket.org/suet_lee/metric_extraction_ddmefd/src/master/

C. Metric Selection

We have already introduced two motivations in selecting for metrics: capturing information about the local environment of an individual, and self-detection of faults. We are most interested in metrics related to external factors in the environment, which would exploit the “swarmy” and complex nature of the system, rather than focusing on mechanical components or the state of software of an individual. These properties have been covered by single-robot fault detection and diagnosis studies and we assume they can be checked at runtime by an on-board internal checker. Instead we look at properties of the environment and the robot, which may affect, or be affected, by individual behaviour and also the wider system.

The metrics selected were informed by the literature in swarm behaviours [10] and were selected systematically in two steps: first we considered the topology of the swarm from a robot’s local perspective, second we considered all possible measurements available from the sensors and actuators specific to our robot platform. The full list is as follows:

1) Robots in range
2) Boxes in range
3) Walls in range
4) Combined in range
5) Velocity
6) Robot delivery count
7) Robot state
8) Nearest robot distance
9) Nearest box distance
10) Nearest wall distance
11) Nearest combined distance
12) ROC of nearest robot ID (rate of change)
13) ROC of nearest box ID
14) ROC of nearest wall ID
15) ROC of nearest combined ID

D. Faulty States

Potential faults were reasoned about systematically by considering our use-case scenario and by reference to faults covered in the body of related work. We consider the following faults:

F_1: 0% max speed
F_2: 10% max speed
F_3: 50% max speed
F_4: Can’t lift boxes
F_5: Can’t deposit boxes
F_6: 0% camera range

We test discrimination of each fault type independently against the “normal” state. In practice, we have the following configuration for each trial:

- Select a fault type: $F_i \in \{F_1, ..., F_6\}$
- Select the number of faulty robots: $0, ..., n$

Faults are assumed to be consistently present for the duration of a trial. Then for any given trial, we have a population of robots belonging to either the faulty state F_i or the normal state. In particular, the fault types we consider also have a level of granularity: F_1, F_2, F_3 are related to the speed of the robot, reduced at different levels.

E. Statistical Analysis

As we are interested in discriminating between two states, we take the approach of generating dataset samples for each state and evaluating the *group difference* between datasets with a suitable statistical test. The datasets are generated for each of the metrics in section III-C, with a sample size of 100. We use the Mann-Whitney U (MWU) test as it makes no assumption of an underlying distribution in the data. We ensure samples are independent and identically distributed to meet the criteria of the test. Further, we can evaluate the *common language effect size* which takes into account sample size [11], [12].

In sum, we derive a measure E which we apply to the normal and faulty datasets for a single metric. E is dependent on the MWU test statistic U and takes into account sample sizes n_1, n_2 for the datasets under comparison:

$$E = 2\left|\frac{U}{n_1 n_2} - 0.5\right| \tag{1}$$

We call E a measure of *disciminatory power* with values in the range of $[0, 1]$, where 0 indicates no power and 1 indicates high discriminatory power.

IV. RESULTS

The matrix of results is shown in figure 2. At a glance, there is some correspondence between fault types and metric types with largest effect size: this could be seen as a kind of "signature" for fault type.

- **Speed related faults** are discriminated by *velocity*, metrics related to *proximity to walls and other robots*, and *robot delivery count*. We reason that slower robots collect fewer boxes, thus having lower delivery count, and reduced speed is reliably detected on-board a robot.
- **Lifter faults** are discriminated by *robot delivery count* and *robot state*. Issues with the lifter impacts a robot's ability to retrieve and deliver boxes. Topological metrics have low discriminatory power however.
- **Camera faults** are discriminated by metrics related to *wall proximity*.

V. A SIMPLE THRESHOLD MODEL

We propose a simple threshold model to demonstrate how the extracted metrics are useful to differentiate between normal and faulty. For a particular fault, we select metrics with the highest discriminatory power. For each metric, we will need a threshold to determine whether a data sample deviates from normal. Our model then uses a linear combination of thresholds to evaluate whether a robot is in a faulty state or not. We build a model to detect a single fault type in the following steps:

1) Discard any metric where $E < s$ and s is a cut-off value to be specified.
2) Take n metrics with the highest discriminatory power E.
3) Find a threshold τ for each metric: take the mean of the faulty and normal datasets for this metric, μ_F and μ_N respectively. We set $\tau = (\mu_F + \mu_N)/2$.
4) We also note which side of the threshold normal and faulty lie: if $\mu_F < \tau$, then a random sample x is considered faulty if $x < \tau$ and normal if $x >= \tau$. For $\mu_F \geq \tau$, x is considered faulty if $x >= \tau$.
5) Specify a number of thresholds, k, to be passed for a robot to be declared as faulty.

We apply these steps to build a model for each selected fault: we select the best combination of parameters n, k and s for performance. The faults selected for analysis are F_1: 0% max speed, F_4: Can't lift boxes, and F_6: 0% camera range. We do not focus on faults F_2 and F_5 as they belong to the same fault types as F_1 and F_4 respectively. Table II lists the selected parameters, including metrics, for the models.

TABLE II: Selected model parameters

	n	k	s	Metrics selected
F_1	5	4	0.15	Velocity
				Robot delivery count
				Nearest combined distance
				ROC Nearest wall ID
F_4	3	2	0.15	Robot delivery count
				Robot state
F_6	5	3	0.15	Combined in range
				Nearest wall distance
				Nearest combined distance

TABLE III: Mean measures

	F_1	F_4	F_6
Mean A	0.96	0.90	0.97
Mean SE	1.0	1.0	1.0
Mean SP	0.92	0.84	0.94

We test the models varying the proportion of faulty robots with 10 repetitions. The model under test is applied to each robot in the scenario and metric values are taken as an average over 50 samples (taken in a 1 second time interval). Finally, we compute an accuracy score *A* at each simulation timestep and additionally, we compute specificity and sensitivity scores: *SP* and *SE*.

Table III shows that the models perform well overall with mean accuracy scores greater or equal to 0.9 and maximum *sensitivity* scores at 1.0 (to 2 d.p.). However, mean *specificity* scores are slightly lower, in the range of 0.84 to 0.94, which means we detect false positives. We have shown that through a data-driven analysis we are able to find metrics where faulty and normal data distributions have large group difference, which allows for a single value threshold to be found easily.

VI. DISCUSSION

We have demonstrated metric extraction to discriminate between faulty and normal states. However, the metrics selected for analysis in section III-C should generalize to other states beyond failure modes, and in particular, they

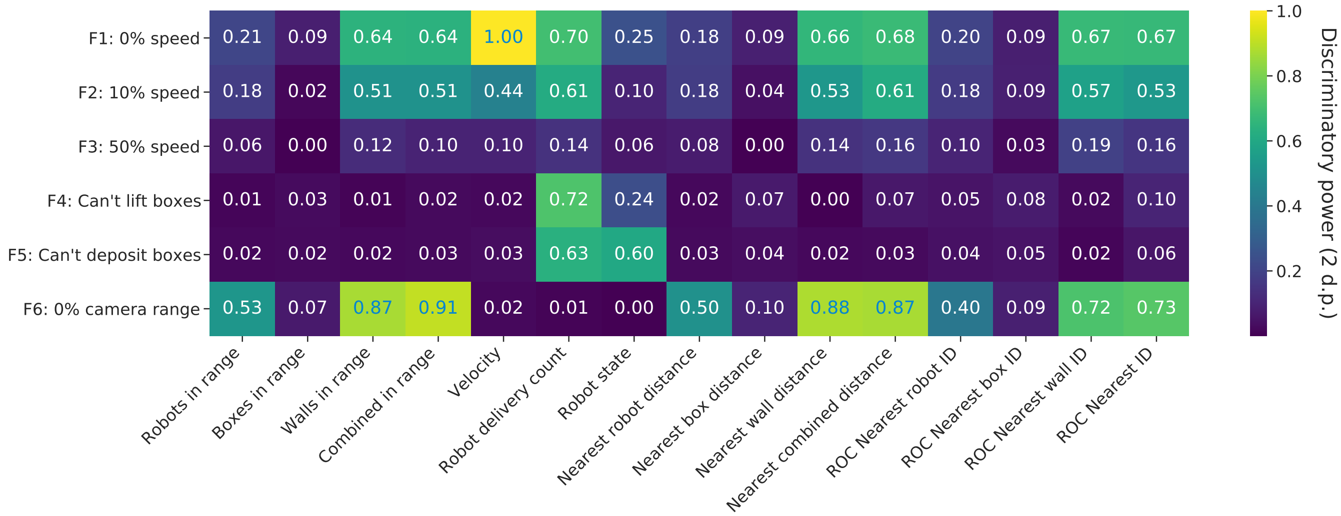

Fig. 2: Discriminatory power (DP) matrix for faults vs. metrics: we evaluate the DP for each metric across all fault types. Effect size values sit in the range [0, 1] with 1 corresponding to the largest effect size. We can see a metric "signature" emerging where a correspondence between fault type and the highest DP metrics may be identified [3].

have potential for diversity extraction. Essentially, the metrics capture information about the local environment of an individual in the swarm.

In order to extend our method to a measure of diversity in a heterogeneous swarm, we would have to know the physical or behavioural characteristics of each subgroup. Given a characterization of each subgroup, we can proceed with the same method - instead of trials across fault types, we can run trials for pairwise comparison of subgroups. Whilst we have designated a default state of "normal" in the context of fault detection, we recognize there may not be an analogous default subgroup in a heterogeneous system. Our method is predominately applicable for comparison between two main groups (or states), one of which may be divided into subgroups (in our case fault types). Otherwise, the number of pairwise comparisons between subgroups could increase significantly.

Further questions for exploration include:

- How does this approach compare to clustering methods, principal component analysis and other multivariate methods?
- Can we reduce the sample size and still produce good results?
- Univariate analysis lends itself to a human-understandable interpretation of metric/fault correlation: is this useful in the context of measuring diversity?

Finally, we have demonstrated the power of our univariate analysis in detecting transitions between two states of a system in real-time with a relatively small sample size. In particular, the ability for real-time discrimination of states may be useful when heterogeneity arises as a dynamic element, with changes in agents arising over time. In future, we would like to test the method on a real-world swarm where collecting a large sample of data may be difficult due to experimental constraints.

VII. Acknowledgements

This work was supported by the US Air Force Office of Scientific Research, European Office of Aerospace Research and Development, the UKRI Trustworthy Autonomous Systems Node in Functionality (EP/V026518/1), and the European Union under Grant Agreement 101070918 and UKRI grant number 10038942.

References

[1] E. Şahin, "Swarm robotics: From sources of inspiration to domains of application," in *Swarm Robotics*, E. Şahin and W. M. Spears, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 10–20.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, 07 2009.

[3] S. Lee, E. Milner, and S. Hauert, "A data-driven method for metric extraction to detect faults in robot swarms," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 746–10 753, 2022.

[4] T. Balch, "Hierarchic social entropy: An information theoretic measure of robot group diversity," *Autonomous Robots*, vol. 8, 03 2000.

[5] S. Xu, L. Böttcher, and T. Chou, "Diversity in biology: definitions, quantification and models," *Physical Biology*, vol. 17, no. 3, p. 031001, mar 2020. [Online]. Available: https://dx.doi.org/10.1088/1478-3975/ab6754

[6] L. Qin, X. He, and D. Zhou, "A survey of fault diagnosis for swarm systems," *Systems Science Control Engineering*, vol. 2, pp. 13–23, 01 2014.

[7] S. Jones, E. Milner, M. Sooriyabandara, and S. Hauert, "Distributed situational awareness in robot swarms," *Advanced Intelligent Systems*, vol. 2, p. 2000110, 08 2020.

[8] E. Milner, M. Sooriyabandara, and S. Hauert, "Stochastic behaviours for retrieval of storage items using simulated robot swarms," *Proceedings of the SWARM2019: The 3rd International Symposium on Swarm Behavior and Bio-Inspired Robotics*, p. 297–308, 2021.

[9] S. Jones, E. Milner, M. Sooriyabandara, and S. Hauert, "DOTS: An open testbed for industrial swarm robotic solutions," 2022. [Online]. Available: https://arxiv.org/abs/2203.13809

[10] M. Schranz, M. Umlauft, M. Sende, and W. Elmenreich, "Swarm robotic behaviors and current applications," *Frontiers in Robotics and AI*, vol. 7, 2020.

[11] D. Kerby, "The simple difference formula: An approach to teaching nonparametric correlation," *Comprehensive Psychology*, vol. 3, 01 2014.

[12] K. O. McGraw and S. P. Wong, "A common language effect size statistic." *Psychological Bulletin*, vol. 111, pp. 361–365, 1992.