Fault Diagnosis For A Team Of Planetary Rovers

Sarah Swinton
James Watt School of Engineering
University of Glasgow
Glasgow, Scotland

Euan McGookin
James Watt School of Engineering
University of Glasgow
Glasgow, Scotland

***Abstract*—The aim of this work is to evaluate the use of a combined fault detection and isolation system; first for a single planetary rover, then applied as a centralised health monitor to a small team of coordinated rovers. Three fault types are modelled: heading sensor faults, actuator faults, and power failure. Testing is carried out on the central health monitor to evaluate its ability to diagnose faults within a simulated environment. The resulting data suggests that fault diagnosis using only top-level telemetry data can successfully diagnose faults within the rover team.**

***Index Terms*—FDIR, Planetary Robotics, Fault Diagnosis**

I. Introduction

Planetary rovers are used to study extremely hazardous locations currently inaccessible to manned-missions. Therefore fault tolerance is central to any planetary exploration system as physical maintenance and part replacement are infeasible. These fault tolerant systems must be able to carry out fault detection, isolation and reconfiguration (FDIR) [1]. Fault detection is the process of identifying any faults in a system. Fault isolation aims to find the source and cause of a fault. The combined process of fault detection and isolation are often referred to as fault diagnosis.

To detect faults, a system must be able to identify the presence of abnormal behaviour. Model based fault detection is one of the most well-studied methods of fault detection [2]. Using model-based fault detection, system properties are predicted by modelling nominal system behaviours [3]. To do this, residuals can be generated by comparing system properties with other estimates or measurements of those properties. Two common residual generation methods are: directional residual generation and structured residual generation [1]. Directional residual generation creates vectors representing residuals. Structured residuals are designed to respond to particular faults, and to have no response to others. Both directional and structured residual generation facilitate the isolation of faults.

In order to diagnose faults, the effect of known faults must be understood. Abci et al. present an informational approach to sensor and actuator fault diagnosis, where a bank of filters is used to compare the probability distributions of a model's predicted and measured values [4]. This approach allowed sensor and actuator faults to be diagnosed, for the occurrence of both single and multiple faults.

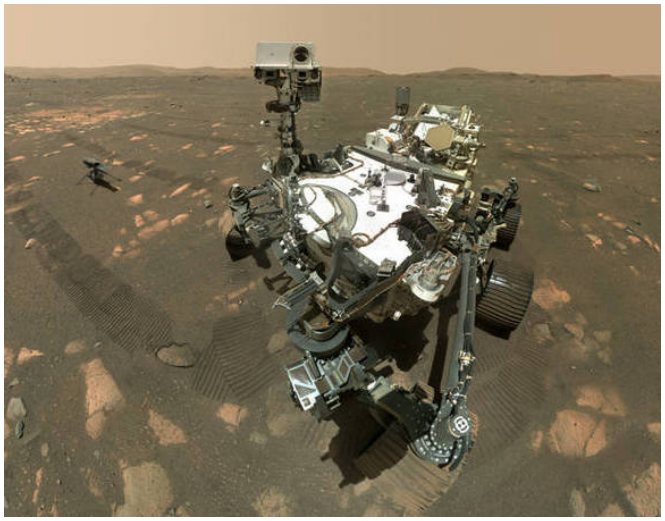

Fig. 1. Perseverance And Ingenuity On Mars

NASA's recent Mars 2020 mission has become the first mission to utilise two vehicles: the Perseverance Rover and the Ingenuity Helicopter [5] (Fig. 1 [6]). In multi-vehicle planetary exploration missions, FDIR is particularly important as faulty vehicles pose a threat not only to their own mission success, but to that of the other vehicles in the group. For example, inter-vehicle collisions could cause the loss of robots and severe degradation to the group's data collection capabilities.

Many studies have been carried out on methods of FDIR for mobile robots. Ireland et al. present an inverse simulation method for fault detection and isolation [3]. This is a model-based method which can be used to generate a suite of residuals from several locations within a rover system by inverting the rover model using an iterative algorithm such as the Newton-Raphson algorithm.

Further studies have described FDIR methods for use within robot teams. Daigle et al. propose a distributed diagnosis algorithm for a rigid formation of mobile robots [7]. This approach utilises fault signatures to predict the time derivative effects of faults on system model measurements.

Meskin et al. present three architectures for multi-vehicle FDIR: centralised, decentralised and semi-decentralised [8]. Within the centralised architecture, all FDIR data is sent to a central unit to be processed. The semi-decentralised

architecture proposes that vehicles in the group should be able to detect not only their own faults (endogenous faults), but faults in neighbouring vehicles (exogenous faults). Within the decentralised architecture each vehicle is programmed to detect and isolate its own faults. A decentralised approach to fault detection has the benefit of being scalable as no one robot needs to process the data of every other robot in the group, and may instead only monitor its closest neighbours, making it well suited to FDIR within robot swarms [9].

This work proposes a centralised health monitoring system to diagnose faults within a group of planetary exploration rovers. In doing so, the system mitigates the expense of separate health monitoring systems for each individual rover. By using top level telemetry data (the position and heading of each rover) the computational load is further reduced.

This paper is set out as follows. Section II discusses the methodologies employed for system and fault modelling, alongside the design of the fault detection and isolation systems. Section III details the full-environment simulation where tests are carried out using a rover team to evaluate the fault diagnosis algorithm. Section IV discusses the results of the previous tests. Finally, Section V sets out the conclusions drawn from this work.

II. Single-Rover Simulation

A. System Modelling

A simple, 4-wheel symmetrical rover is described. This mathematical model of a rover has been developed and validated at the University of Glasgow in [10]. Two frames of reference are utilised in this project (Fig. 2). The first is the Earth-fixed frame, which has an inertially fixed origin, and axes denoted X_E, Y_E, Z_E. The second is the rover body frame, which rotates with the rover's motion i.e., is fixed to the rover's axes denoted X_B, Y_B, Z_B. The origin of the rover body frame is the rover's centre of gravity.

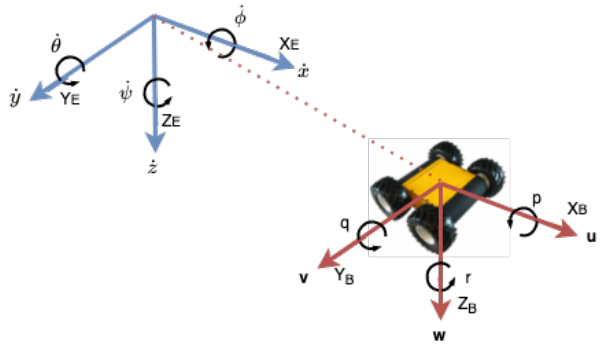

Fig. 2. Earth-Fixed And Rover Body Axes

The rover's equations of motion, with reference to the rover body frame and Earth-fixed frame, can be described by the matrix relationship shown in Equation (1).

$$\begin{bmatrix} \dot{v} \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} \frac{-(C(v)+D(v))}{M} & \frac{-g(v)}{M} \\ J(\eta) & 0 \end{bmatrix} \begin{bmatrix} v \\ \eta \end{bmatrix} + \begin{bmatrix} \frac{-\tau'}{M'} \\ 0 \end{bmatrix} \quad (1)$$

Here, vectors v and η represent the velocity and displacement variables. M is the mass and inertia matrix, $C(v)$ is the Coriolis matrix, $D(v)$ is the damping matrix, $g(v)$ represents the gravitational forces and moments, $J(\eta)$ is an Euler matrix representing the trigonometric transformation from the body fixed reference frame to the earth fixed reference frame, and the τ vector represents the control inputs. This model is fully described in [10] and [11].

B. Guidance, Navigation and Control

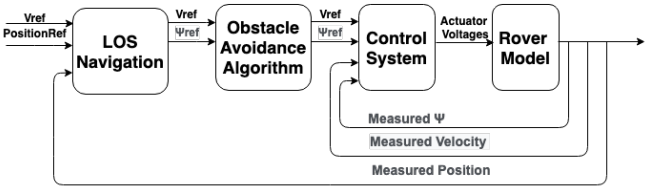

Fig. 3. Guidance, Navigation and Control Architecture

Each rover is equipped with a guidance system providing control, navigation and obstacle avoidance (Fig. 3). The control system consists of two PID controllers for heading and velocity, respectively. The interaction of these two controllers is described in Fig 4.

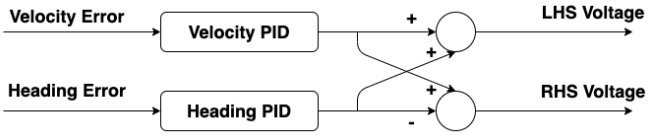

Fig. 4. Control System Block Diagram

The navigation algorithm is a Line-of-Sight algorithm [12], allowing each rover to navigate towards its current waypoint. A simple obstacle avoidance algorithm adjusts the heading of the rover in response to the detection of a static obstacle [11]. If an object is identified as visible and is deemed to be a hazard (i.e., if it is within 1m of the rover and it intercepts the rover's path), the waypoint following algorithm is adjusted to prioritise manoeuvring around the obstacle. Only once this manoeuvre is complete will the rover return to following its target waypoints.

C. FDIR Architecture

To carry out fault detection and isolation, a reference model is introduced to the simulation. Faults are not applied to the reference model. Fig. 5 shows the architecture of the FDIR system. The rover's control input is fed to both the reference model and the measured rover model, where the measured rover is represented by actuators, plant dynamics

and sensors. The outputs of the reference and measured rover models are compared in order to generate residuals to be used for fault detection. A fault isolation filter is used to determine the cause of any detected faults.

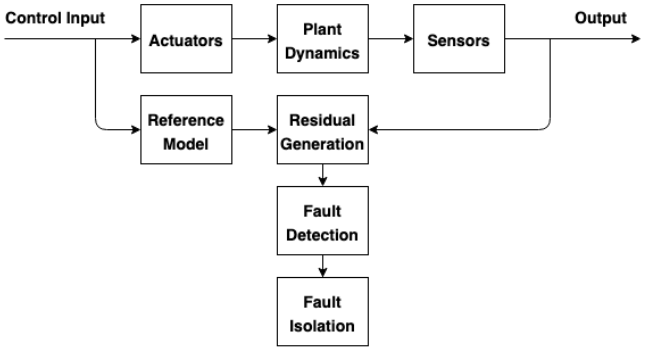

Fig. 5. FDIR System Architecture

D. Fault Modelling

Three faults have been modelled: power failure, wheel actuator faults and heading sensor faults. Power failure represents the event in which a rover is unable to continue motion, and is modelled by reducing the torque applied to the rover's motors to zero. Actuator faults have been modelled to affect either the left-hand side (LHS) or right-hand side (RHS) of the rover's actuators. These faults are injected as a reduction by half in the torque applied to faulty actuators. A fault magnitude of +1°for the modelled heading sensor fault causes a sufficiently large change in reported heading angle without causing instability in the system.

E. Fault Detection System

In order to detect the occurrence of a fault, residuals have been generated. The general equation for the generation of a residual is shown in Equation (2), where the residual of an output, R_{signal}, is the difference between the measured output signal, $Y_{measured}$, and the estimated output signal, $Y_{estimated}$, at time, t.

$$R_{signal}(t) = Y_{measured}(t) - Y_{estimated}(t) \qquad (2)$$

To minimise computational complexity, residuals are generated using top level data, i.e. the rover's heading and position, by comparing real-time data from both the reference model and the faulty rover model. Once residuals have been generated, a threshold is used to detect the presence of a fault. Since a power failure would not produce a non-zero heading residual in all cases, a fault detection threshold of 0.01m was placed on the position residual as all three faults would trigger a detection. This low threshold was possible as no noise was added to the system - the detection threshold would require re-calibration in the presence of noise. The fault isolation routine is triggered when the a position residual exceeds the detection threshold.

F. Fault Manifestation

The diagnosis system was developed using a script representing a single rover, which traverses a path consisting of four waypoints. The motion of the rover, subject to no faults, can be seen in Fig. 6. All fault residuals in Section III have been generated using this array of waypoints, where the fault is injected 40 seconds into path traversal.

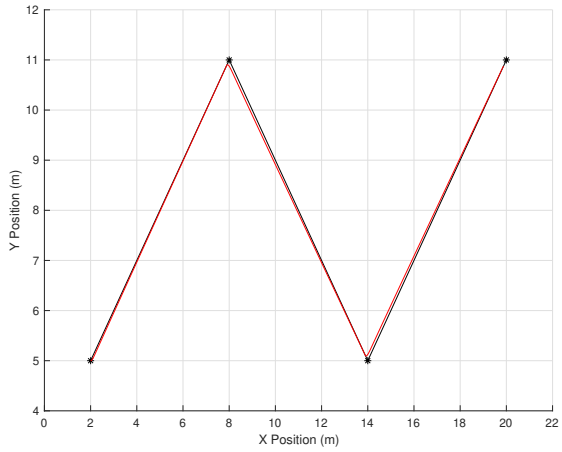

Fig. 6. Rover Test Path - Simple

The residuals generated by the injection of a power failure can be seen in Fig. 7. The rover's position residual increases linearly with time as the rover halts and the reference model continues along its path. Step changes in heading residual occur as the reference model traverses each path segment.

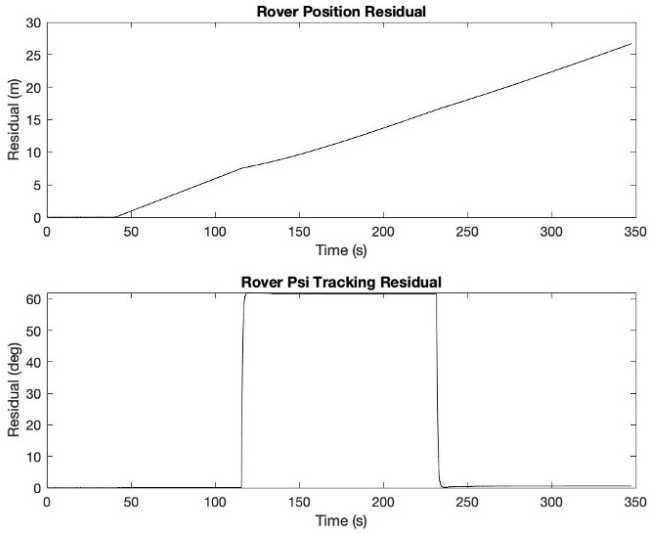

Fig. 7. Residuals for Power Failure

The residuals generated by the injection of an actuator fault can be seen in Fig. 8. A spike in the position residual occurs upon waypoint acquisition. The position residual slowly decreases before the next waypoint, as the control

system attempts to minimise heading error. Upon waypoint acquisition, the heading residual experiences a spike. The PID heading controller is able to compensate for this spike quickly.

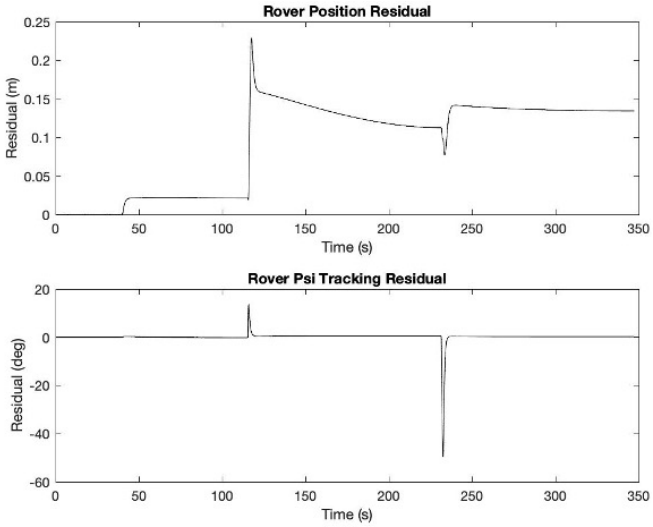

Fig. 8. Residuals for Actuator Fault

Fig. 9 shows the residuals generated by the injection of a heading fault. After an initial spike, the position residual increases between the acquisition of waypoints, until the control system is able to compensate for the fault and drive the rover towards its desired waypoints. The magnitude of the position fault is significantly larger than that caused by an actuator fault. The heading residual spikes upon acquisition of a waypoint whilst the control system attempts to compensate for the fault.

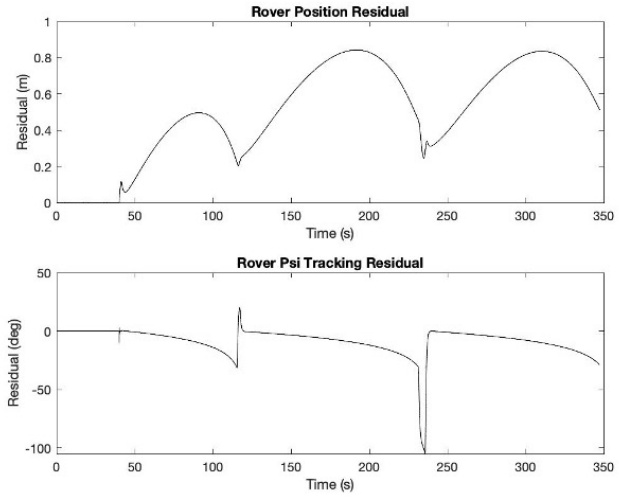

Fig. 9. Residuals for Heading Sensor Fault

G. Fault Isolation System

A key aim of the fault isolation logic is global diagnosability, meaning that all three modelled faults can be uniquely identified given the data available to the fault isolation system. To achieve this aim, a set of structured residuals have been defined such that each residual responds to particular faults, without responding to others [1]. The first is ΔR_H, which represents the change in a rover's heading residual between the time of detection and the time of isolation. Time of isolation has been selected arbitrarily as one second after fault detection in order to allow the rate of change of the fault signatures to be captured within the isolation logic. The second is ΔR_P, which represents the change in a rover's position residual between the time of detection and the time of isolation. The final structured residual is ΔP. While the ΔR_H and ΔR_P thresholds are calculated using the residual data from the reference and measured models, the value of ΔP is simply the measured rover's change in position between the time of detection and time of isolation.

To calibrate each residual's isolation threshold, the magnitude of the heading and position residuals associated with each fault type were recorded, at both the time of detection and time of isolation. A ΔR_H threshold of 0.1°, and a ΔR_P threshold of 0.01m were implemented. The threshold for the ΔP isolation variable has been defined as 25% of the distance that the non-faulty rover should travel at its desired velocity, $V_{desired}$, in the time between fault detection and isolation, $T_{diagnosis}$ (Equation (3)).

$$\Delta P_{threshold} = 0.25 \times V_{desired} \times T_{diagnosis} \tag{3}$$

The fault isolation matrix shown in TABLE I presents which isolation thresholds are exceeded by particular faults. This shows that each fault model exceeds a unique set of thresholds. The faults modelled can be uniquely identified and are, therefore, globally diagnosable.

TABLE I
ISOLATION MATRIX

Fault	**Threshold**		
Type	ΔR_H	ΔR_P	ΔP
Power Failure	0	1	0
Actuator Fault	1	0	1
Sensor Fault	1	1	1

III. FULL-ENVIRONMENT SIMULATION

A. Environment Model

A simulated environment model has been designed to provide a sufficient path planning and traversal challenge to both individual rovers and the group [11]. Individual rovers must follow a safe path which avoids high risk areas and hazards. The group must be able to sufficiently coordinate their paths such that rovers can pass through narrow areas without collisions. There are three layers of information within the environmental model: large obstacles/impassable areas, terrain classification, and small obstacles/hazards.

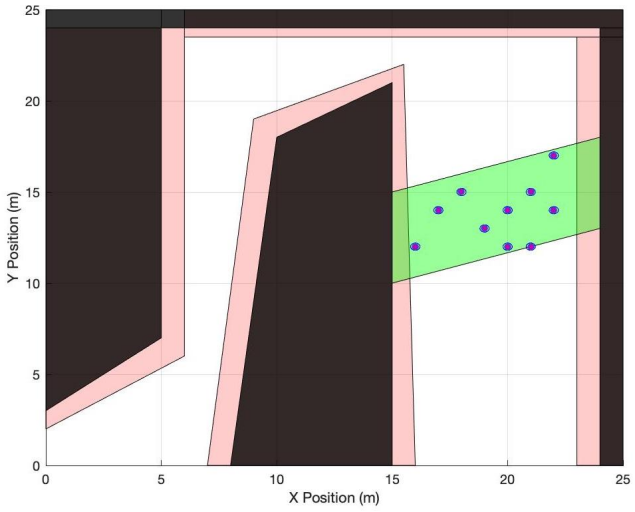

Fig. 10. Simulated Environment

Fig. 10 shows the environmental model; where black regions are impassable, pink regions represent steep slopes, green regions represent a rough regolith rock field, magenta points represent small obstacles surrounded by a blue safety radius, and white areas represent smooth regolith.

B. Multi-Rover Method

A simulation has been developed representing five rovers. In this work, the team of rovers is directed by a centralised planner with a decoupled architecture [11]. This planner carries out all health monitoring and coordination for the team of rovers. This central planner could be contained within a lander, orbiter or a larger 'parent' rover. No communication takes place between rovers as all communications are sent via the central planner. The use of a central health monitor reduces the expense of the team's FDIR system by removing the need for individual health monitoring systems in each rover. Within the simulation, each rover is instructed to navigate towards a unique target, with the central planner coordinating the motion of each rover to mitigate the risk of collisions.

C. Multi-Rover Results

Tests have been carried out using multiple rovers within the full simulation environment, to evaluate the ability of the health monitor to correctly detect and isolate faults whilst monitoring the rover group. The fault modes evaluated are power failure, LHS actuator faults, RHS actuator faults, and a heading sensor fault. Tests have been carried out using one set of coordinated paths. First, a control test was run, in order to evaluate whether the health monitor system would experience any false positive detections. No detections occurred during the control test.

For each of the four tests modes, twenty-five diagnosis scenarios were run. During each test scenario, the current fault mode was applied to a single rover at a pre-decided fault injection time (40 seconds into the simulation). Five fault injection times were tested so that diagnosis tests could be carried out at various points along each of the five rover paths. First, each fault mode has been tested to ensure that its manifestation exceeds the fault detection threshold. This is shown in Fig. 11, where the threshold is shown in blue, with the residual manifestation shown in black.

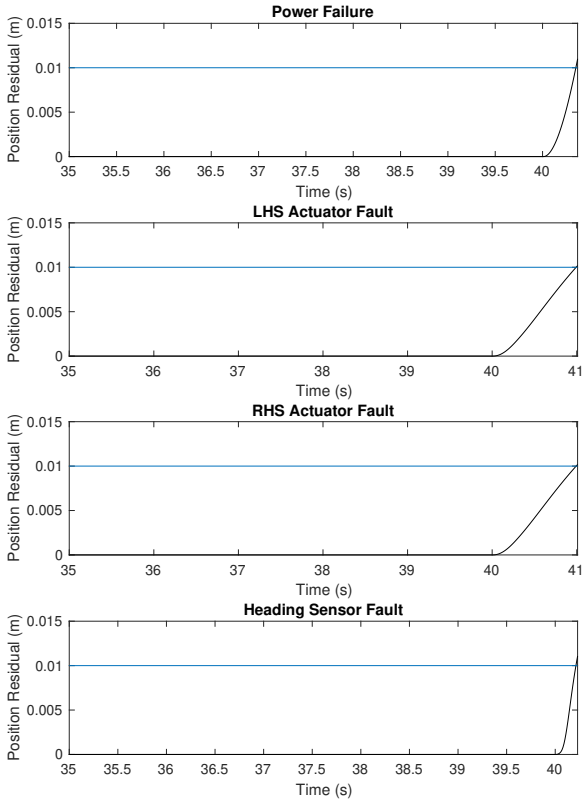

Fig. 11. Fault Detection Verification

Table II shows the results of these tests for each of the four fault modes. Here, diagnosis time is the time taken to diagnose the fault after its initial injection.

TABLE II
MULTI-ROVER TESTING DIAGNOSIS DATA

Fault Type	**Correct Diagnoses**	**Avg. Diagnosis Time** (s)
Power Failure	25	0.40
Actuator Fault (LHS)	23	1.45
Actuator Fault (RHS)	23	1.43
Sensor Fault	22	0.23

Table III shows the average residual values for each of the four fault modes.

TABLE III
MULTI-ROVER TESTING RESIDUAL DATA

Fault Type	**Avg. Residual Magnitude**		
	$\Delta R_H(^\circ)$	$\Delta R_P(m)$	$\Delta P(m)$
Power Failure	2.286	0.075	0.017
Actuator Fault (LHS)	0.086	0.008	0.088
Actuator Fault (RHS)	0.052	0.016	0.089
Sensor Fault	5.075	0.054	0.048

IV. DISCUSSION

During the power failure tests, all faults were diagnosed correctly. In the average time taken to diagnose a power failure, the maximum distance travelled by the faulty rover was 0.04 m. As a power failure incurred a very small change in heading, it is likely that the rover continued to follow its planned path before it would come to rest. Therefore, the faulty rover poses minimal threat to the safety of the non-faulty rovers and no further changes are proposed in order to adjust the performance of the diagnosis of power failures.

In each of the two actuator fault test sets, two of the injected faults were misdiagnosed. In the LHS actuator fault test set, both of the misdiagnosed faults were diagnosed as heading sensor faults. These were due to a heading change which exceeded the isolation threshold for actuator faults. In order to resolve this, the fault isolation thresholds could be tuned further. In the RHS fault actuator test set, one fault was similarly misdiagnosed as a heading sensor fault. The other case misdiagnosed the actuator fault as a power failure because the fault was injected as the rover crossed the boundary between the smooth regolith and the rock field. This increases the distance between the reference and measured rovers as they have different desired velocities therefore exceeding the isolation threshold. This could be resolved by checking the rover's location during the fault isolation process.

During the heading sensor fault tests, three faults were misdiagnosed. All three misdiagnoses identified a power failure due to changes in desired velocity affecting the rover's movement during fault isolation.

In the vast majority of cases, faults have been correctly diagnosed. In these cases, an appropriate recovery method could be applied in order to mitigate the negative fault effects on the overall system. Where sensor and actuator faults have been misdiagnosed as power failures, the rover would be considered non-functional when, in reality, it could have continued its operation with some function under an appropriate recovery strategy. Sensor faults misdiagnosed as actuator faults, and vice versa, could lead to the implementation of an inappropriate recovery strategy, which could mean that the faulty rover poses a danger to its team mates.

V. CONCLUSIONS

Planetary exploration rover missions are one of the most impactful, high-cost, and high-risk engineering feats of the 21st century. The ability to diagnose faults in such rovers in remote and hazardous locations is critical. To achieve this, a centralised health monitor system for a small group of planetary exploration rovers was proposed within this work.

This health monitor is comprised of a fault detection threshold and a fault isolation filter. The fault detection threshold indicates that a fault is present if the position of a measured rover deviates from that of its fault-free reference model. The fault isolation filter uses the position and heading data of each rover to isolate the source of a fault. By using this top-level data to characterise faults at component level, the health monitoring system is able to reduce the computational load required to correctly diagnose the modelled faults: power failures, actuator faults, and heading sensor faults. The health monitor was found to correctly diagnose faults during functionality testing in 93% of cases.

In this work, the use of a central health monitoring system which carries out fault detection and isolation, using top-level data, has been found to correctly diagnose heading sensor faults, actuator faults, and power failures for a small group of rovers in a simulated environment.

REFERENCES

[1] Hwang I, Kim S, and Seah C.E. A Survey of Fault Detection, Isolation, and Reconfiguration Methods. *IEEE Transactions on Control Systems Technology*. 2010;18(3):636-653

[2] Isermann R. Supervision, Fault-Detection and Fault-Diagnosis Methods - An Introduction. *Control Engineering Practice*. 1997;5:639-652

[3] Ireland M.L., Mackenzie R, Flessa T, Worrall K, Thomson D.G., and McGookin E. Inverse Simulation as a Tool for Fault Detection and Isolation in Planetary Rovers. In Proceedings of 10th International ESA Conference on Guidance, Navigation and Control Systems, Salzburg, Austria; 2017

[4] Abci B, El Najjar M.E.B., Cocquempot V and Dherbomez G. *An Informational Approach for Sensor and Actuator Fault Diagnosis for Autonomous Mobile Robots*. 2020;99(2):387–406

[5] Farley K, Williford K, Stack K, Chen A, Torre M, Hand K, et al. Mars 2020 Mission Overview. *Space Science Reviews*. 2020;216:142

[6] NASA/JPL-Caltech/MSSS. Looking at the WATSON Camera [image on the internet]. 7 Apr 2021 [cited 28 Sep 2021]. Available from: https://mars.nasa.gov/resources/25790/perseverances-selfie-with-ingenuity/

[7] Daigle M.J., Koutsoukos X.D., Biswas G. Distributed Diagnosis in Formations of Mobile Robots. *IEEE Transactions on Robotics*. 2007;23(2):353-369

[8] Meskin N, and Khorasani K. Actuator Fault Detection and Isolation for a Network of Unmanned Vehicles. *IEEE Transactions on Automatic Control*. 2009;54(4):835-840

[9] Tarapore D, Christensen A.L., and Timmis J. Generic, Scalable and Decentralized Fault Detection For Robot Swarms. 2017. PLoS ONE [Internet] 12(8): e0182058. Available from: https://doi.org/10.1371/journal.pone.0182058

[10] Worrall K.J. Guidance and Search Algorithms For Mobile Robots: Application and Analysis Within The Context of Urban Search and Rescue [PhD Thesis]. Glasgow, Scotland. *University of Glasgow*. 2008

[11] Swinton S. Fault Tolerant Coordination of Multiple Rovers For Planetary Exploration [MEng Dissertation]. Glasgow, Scotland. *University of Glasgow*. 2021

[12] Breivik M. Nonlinear Maneuvering Control of Underactuated Ships [Dissertation]. Trondheim, Norway. *Norwegian University of Science and Technology*. 2003