

GLOBAL TASK-AWARE FAULT DETECTION IDENTIFICATION FOR ON ORBIT MULTI-SPACECRAFT COLLABORATIVE INSPECTION

Akshita Gupta*, Yashwanth Kumar Nakka[†], Changrak Choi[‡], and Amir Rahmani^{§, ¶}

In this paper, we present a global-to-local task-aware fault detection and identification algorithm to detect failures in a multi-spacecraft system performing a collaborative inspection (referred to as global) task. The inspection task is encoded as a cost functional \mathcal{H} that informs global (task allocation and assignment) and local (agent-level) decision-making. The metric \mathcal{H} is a function of the inspection sensor model, and the agent full-pose. We use the cost functional \mathcal{H} to design a metric that compares the expected and actual performance to detect the faulty agent using a threshold. We use higher-order cost gradients \mathcal{H} to derive a new metric to identify the type of fault, including task-specific sensor fault, an agent-level actuator, and sensor faults. Furthermore, we propose an approach to design adaptive thresholds for each fault mentioned above to incorporate the time dependence of the inspection task. We demonstrate the efficacy of the proposed method empirically, by simulating and detecting faults (such as inspection sensor faults, actuators, and sensor faults) in a low-Earth orbit collaborative spacecraft inspection task using the metrics and the threshold designed using the global task cost \mathcal{H} .

INTRODUCTION

Spacecraft swarms enable a new class of flexible and adaptive missions including collaborative inspection,^{1,2} and in-orbit construction.³ Progress in miniaturization and CubeSat technology⁴ has further boosted the interest in a technology demonstration of distributed measurement on missions like MarCo⁵ and GRACE.⁶ The multi-spacecraft mission approach offers intrinsic robustness to faults through reconfiguration and improved science data returns through cooperation and collaboration. While multi-spacecraft missions have many advantages, they require complex communication architectures, leading to potential fault propagation. For example, in a leader-follower configuration, any large disturbance in the motion trajectory of one of the followers propagates through the network, leading to network shape deformation. Another example is in a distributed sensing application; if one of the agents behaves in an adversarial manner, the neighboring agents follow the misbehavior due to the underlying consensus framework.

A fault detection, isolation, and recovery (FDIR) architecture is essential to accommodate potential faults at both the network and individual agent levels to continue the mission with graceful

*Ph.D. Student, Department of Industrial Engineering, Purdue University.

[†]JPL Postdoctoral Scholar, Maritime and Multi-Agent Autonomy group, Jet Propulsion Laboratory, California Institute of Technology

[‡]Robotics Technologist, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91104, USA

[§]Supervisor, Maritime and Multi-Agent Autonomy group, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91104, USA

[¶]Authors Akshita and Yashwanth contributed equally to this work.

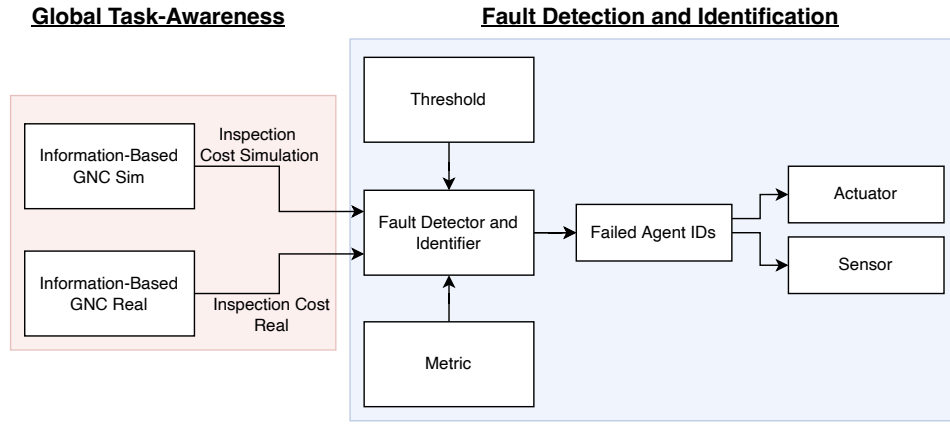


Figure 1: Global task-aware fault detection and isolation for a distributed spacecraft network.

degradation. In this work, we present a new FDI method, as shown in Fig. 1, that detects the failure at the network level using an abstraction of the global task objective \mathcal{H} and local sensing information and informs the agent level FDIR algorithm to perform necessary actions for recovery. For example, a fault at the network level could be due to communication loss, a global task sensor fault (for inspection task), a global task actuator fault (for on-orbit construction), and a fault at the agent level could be due to thruster or reaction wheels. We propose a simulation vs. real comparison using the \mathcal{H} and its higher-order gradients. We detect the fault by computing the off-nominal behavior from the expected global task objective \mathcal{H} by monitoring the individual task using a residual vector sensitive to the agent's faults. The global task objective \mathcal{H} is designed to be a function of the state of agents in the network and a model of the task sensor (for inspection) or actuator (for construction). The residual vector is a function of the local relative state estimates of the agents in the network. We propose a metric that computes the deviation of the \mathcal{H} from the expected performance, which is used as an indicator for faults at the network level, and uses higher-order derivatives of \mathcal{H} to infer if the agent-level faults, as shown in Fig. 1.

Earlier work^{7,8} includes FDI architectures for distributed sensor networks using a local decentralized observer to detect internal agent sensors or actuator failure. Recent work^{9,10} uses adaptive or reconfiguration control with minimal notion of network task to achieve fault tolerance. We instead focus on incorporating global task objectives to inform the local FDIR to react or respond appropriately to continue the mission autonomous to the best possible capability of the network while maximizing the global task objective \mathcal{H} . Furthermore, while most of the earlier work^{11,12} on FDI for distributed systems focuses on simple linear dynamical systems, we focus on low-earth orbit formation flying dynamics that includes periodic orbits.

The main contributions of this work are as follows: 1) we propose an architecture for FDI in a multi-agent spacecraft system that integrates global-task objectives and local-agent level behaviors for task awareness, 2) we derive a global cost functional that is decomposable to cost functions that inform local progress and intermediate consensus on global progress, 3) we propose a novel FDI metric based on the global cost \mathcal{H} and the high-order derivatives of the \mathcal{H} to detect and identify both the global and local faults.

We apply our FDI architecture to a recently proposed multi-agent collaborative spacecraft inspection mission^{1,2} in a low Earth orbit to detect failures in the inspection sensors and individual agent

sensing. The cost function \mathcal{H} defines the global inspection progress by fusing individual agent sensor data measurements. The inspection data fusion runs at a fixed frequency ω_g , and the network fault diagnosis is run at frequency ω_{FDI} . We assume that agents communicate with each other only when within the communication radius, leading to a time-varying communication topology and sensing graph. The proposed method is capable of handling the time-varying graph and intermittent communication. We demonstrate that the proposed method can detect and identify the faults while keeping track of the global task. This approach is essential to inform the recovery procedure, described in our recent work,¹³ for designing new orbits and pointing trajectories to complete the mission.

The paper is divided into four sections: 1) overview of the GNC architecture for the multi-agent collaborative inspection; 2) problem description and an outline of the potential faults at both the global task level and agent level; 3) discussion of the proposed approach and the derivation of the metrics and the threshold for fault detection and identification; and 4) preliminary implementation and simulation results.

OVERVIEW OF GNC ARCHITECTURE FOR INSPECTION

In this section, we give a brief overview of the collaborative low-earth orbit inspection framework proposed in our earlier work.^{2,14} Using this framework we design optimal Passive Relative Orbits^{2,15} (PROs) and attitude trajectories for N observer spacecrafts, inspecting M Points of Interest (POIs) on a target spacecraft, by solving the following information-based optimal control problem.

Problem 1 Information-Based Optimal Control Problem

$$\min_{\mathbf{p}, \mathbf{u}_i} \int_0^{t_f} \left(\sum_{j=1}^M \mathcal{H}(\mathbf{p}, \mathbf{s}_j) + \sum_{i=1}^N \|\mathbf{u}_i\| \right) dt \quad (1)$$

$$\text{s.t.} \quad \begin{cases} \text{Dynamics Model :} & \dot{\mathbf{p}}_i = \mathbf{f}(\mathbf{p}_i, \mathbf{u}_i) \\ \text{Safe Set :} & \mathbf{p}_i \in \mathcal{P}, \forall i \in \{1, \dots, N\} \\ \text{Inspection Sensor Model :} & \mathbf{z}_{i,j} = h(\mathbf{p}_i, \mathbf{s}_j) + \xi, \xi \sim \mathcal{N}(0, \Sigma_h(\mathbf{p}_i, \mathbf{s}_j)), \end{cases} \quad (2)$$

$$\text{Points of Interest :} \quad \mathbf{s}_j \forall j \in \{1, \dots, M\} \quad (3)$$

where $\sum_j \mathcal{H}(\mathbf{p}, \mathbf{s}_j)$ is the information cost, $\sum_i \|\mathbf{u}_i\|$ is the fuel cost, \mathbf{p}_i is the full-pose of the observer spacecraft, \mathbf{s}_j is the full-pose of the j^{th} POI on the target spacecraft. The inspection sensor model in Eq. (3) outputs the value of interest $\mathbf{z}_{i,j}$, when the i^{th} observer with pose \mathbf{p}_i is inspecting a POI at \mathbf{s}_j . Minimizing the information cost $\sum_j \mathcal{H}(\mathbf{p}, \mathbf{s}_j)$ ensures that the inspection task is complete.

We decompose the Problem 1 to derive a hierarchical GNC algorithm (for details refer to our earlier work¹⁴). The hierarchical algorithm uses the information-cost and the sensor model to select the informative PROs and attitude pointing vector for each agent. We optimize the informative PROs and attitude plan for optimal orbit insertion, reconfiguration, and attitude tracking using an optimal control problem formulation that computes minimum fuel trajectory using sequential convex programming approach. In this work, we use the information cost to keep track of the task progress and detect off-nominal behaviour of the multi-agent system and the individual agents. We describe the cost functional used to compute the information gain in the following.

Information Gain. To quantify the information, a prior model of the target spacecraft is used along with sampled *points of interest* (POIs) on the surface of the spacecraft. The cost function \mathcal{H} is designed to minimize the total variance on the knowledge of POIs. We use the cost function \mathcal{H} designed in,¹⁶ and is a function of POIs as given below.

$$\mathcal{H}_{\text{POI}}(\mathbf{s}) = \left(w^{-1} + \sum_{\mathbf{p} \in \mathcal{P}} \sigma(\mathbf{p}, \mathbf{s})^{-1} \right)^{-1}$$

$$\mathcal{H} = \sum_{\mathbf{s} \in \text{POIs}} \mathcal{H}_{\text{POI}}(\mathbf{s}) \phi(\mathbf{s}), \quad (4)$$

where $\mathbf{s} \in \mathbb{R}^3$ is a POI on the target spacecraft's surface, $w \in \mathbb{R}$ is the initial variance based on the prior model of the target spacecraft, $\mathbf{p} \in SE(3)$ is the pose of a sensor mounted on a spacecraft such as a camera, \mathcal{P} is the set of all sensor poses, $\sigma(\mathbf{p}, \mathbf{s})$ estimates the variance of estimating POI at \mathbf{s} with the sensor at \mathbf{p} , and $\phi(\mathbf{s}) \in \mathbb{R}$ is the relative importance of POI \mathbf{s} .

The function $\sigma(\cdot, \cdot)$ corresponds to information per pixel. It incorporates sensor characteristics such as the current uncertainty of the spacecraft's pose estimate, the accuracy of the sensor based on the distance between \mathbf{p} and \mathbf{s} , or the lighting conditions. Here, we use a simple RGB camera sensor and no environmental noise:¹⁶

$$\sigma(\mathbf{p}, \mathbf{s}) \propto \begin{cases} \text{dist}^2(\mathbf{p}, \mathbf{s}) & \mathbf{s} \text{ visible from } \mathbf{p} \\ \infty & \text{otherwise} \end{cases}, \quad (5)$$

where $\text{dist}(\mathbf{p}, \mathbf{s})$ is the Euclidean distance between POI \mathbf{s} and pose \mathbf{p} . We compute σ using visibility checking. The offline solution to problem 1 is used to predict the nominal system behavior in terms of the information-based cost \mathcal{H}_{nom} over a finite time interval (1 or 2 orbits). As described in Fig. 1, we precompute the nominal behaviour \mathcal{H}_{nom} and compare it to the real-time behaviour \mathcal{H} over the time horizon t as follows:

$$\int_0^t (\mathcal{H} - \mathcal{H}_{\text{nom}}) dt \geq \Delta \mathcal{H}_{\text{threshold}} t. \quad (6)$$

If the real-time value deviates from the nominal behavior by a threshold $\Delta \mathcal{H}_{\text{threshold}}$ then a fault is detected. In the following section, we discuss on how we modify the cost function to construct the FDI architecture in Fig. 1.

PROBLEM DESCRIPTION

The goal of the work is to detect the global and local faults in a multi-spacecraft system performing global tasks such as: on-orbit inspection and on-orbit construction. The global behaviour faults and local faults detected using the proposed approach are described in the following Figure 2. Note that the global behaviour fault for inspection task could lead to both deteriorating and improved performance. The improved performance is due two reasons: 1) failure in the controller at the agent level that leads to better exploration of the inspection target, and 2) spurious signals being communicated by the neighbours. The deteriorating performance is due to failure in the inspection sensor or the pointing controller of the spacecraft. The behaviour fault detection is used as an input to identify the agent and the type of fault. The fault detection and identification problem addressed in this work is summarized as follows:

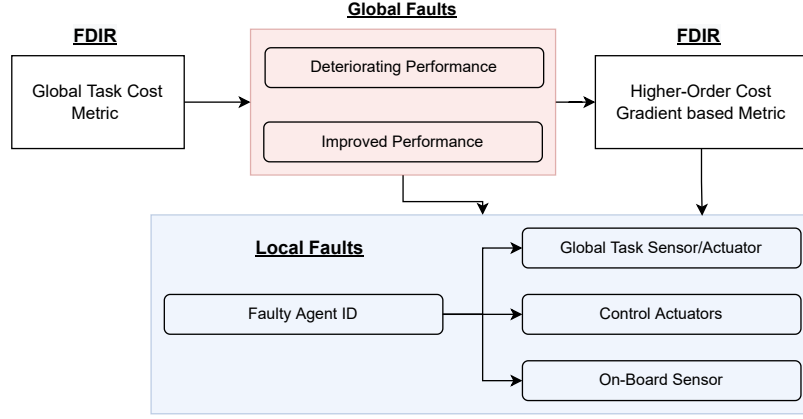


Figure 2: An overview of the type of global and local faults detected and identified using the proposed FDI architecture, metrics and the threshold.

Problem 2 *Given the nominal expected global task performance \mathcal{H}_{nom} and the real-time performance \mathcal{H} , detect the global and agent-level faults in the multi-spacecraft system performing a global task (collaborative inspection or construction). The global and agent-level fault tree is described in the Fig. 2.*

In the following sections, we describe the derivation of the metrics and threshold used in the two FDIR blocks shown in Figure 2 for fault detection and identification.

GLOBAL TASK AWARE FAULT DETECTION AND ISOLATION

In this section, we describe the different components of the global task aware fault detection and isolation algorithm that solves the Problem 2. This section describes the different components of the fault detection system. We first describe the derivation of the attribute required to detect a faulty spacecraft and the gradient-based fault metric is derived. Finally, we discuss the design of thresholds for detecting and identifying different types of fault.

Global Task Cost Functional

We use a centralized monitoring system that utilizes only the information cost updates and implicitly the variance updates of the POIs being shared on the communication network during the collaborative inspection process by the individual spacecraft to detect off-nominal behaviour and identify the faults. At any given time instant t , each spacecraft i shares its local information cost value $\mathcal{H}_i(t)$ with the central computing system. To determine the value of $\mathcal{H}_i(t)$ observe that from (4), the centralized cost function is given as follows:

$$\mathcal{H} = \sum_{s \in \mathcal{S}} \mathcal{H}_{POI}(s) \phi(s), \quad (7)$$

$$\text{Where: } \mathcal{H}_{POI}(s) = (w^{-1} + \sum_{p \in \mathcal{P}} f(p, s)^{-1})^{-1}.$$

The above cost functional for individual POI can be decomposed into contributions from individual agent H_i and a consensus term $\psi(s)$ that is updated over fixed intervals of time.

$$\begin{aligned}\mathcal{H}_{POI}(s) &= (w^{-1} + \sum_{p \in \mathcal{P}} f(p, s)^{-1})^{-1} \\ \mathcal{H}_{POI}(s) &= \psi(s)(w^{-1} + \sum_{p \in \mathcal{P}} f(p, s)^{-1})\end{aligned}\tag{8}$$

where, $\psi(s) = \frac{1}{(w^{-1} + \sum_{p \in \mathcal{P}} f(p, s)^{-1})^2}$. Therefore,

$$\begin{aligned}\mathcal{H} &= \sum_{s \in \mathcal{S}} \phi(s) \psi(s) (w^{-1} + \sum_{p \in \mathcal{P}} f(p, s)^{-1}) \\ \mathcal{H} &= \sum_{s \in \mathcal{S}} \phi(s) \psi(s) w^{-1} + \underbrace{\sum_{s \in \mathcal{S}_1} \phi(s) \psi(s) f(p_1, s)^{-1}}_{H_1} + \dots + \underbrace{\sum_{s \in \mathcal{S}_N} \phi(s) \psi(s) f(p_n, s)^{-1}}_{H_N} \\ \mathcal{H} &= \sum_{s \in \mathcal{S}} \phi(s) \psi(s) w^{-1} + \sum_{p_i \in \mathcal{P}} \underbrace{\sum_{s \in \mathcal{S}_i} \phi(s) \psi(s) f(p_i, s)^{-1}}_{H_i}\end{aligned}\tag{9}$$

Using (9), the information cost for each spacecraft i is computed as $\mathcal{H}_i(t) = \sum_{s \in \mathcal{S}_i} \phi(s) \psi(s) f(p_i(t), s)^{-1}$ where $\psi(s)$ is the normalization factor. This attribute not only tracks the performance of individual spacecrafts, but also prevents the need to explicitly share information about the state of spacecrafts over the communication channel inbetween the agents.

Note: The new formulation in Eq. (9) is a linear combination of the individual contributions by each agent i at high frequency updates and intermediate consensus term $\psi(s)$ that is updated at low frequency. This decomposition, is useful in computing the performance of individual agent, and design the metric and threshold using the individual agent contribution H_i .

Fault Metric

In this work, a gradient based metric is used to compare the actual progress made by individual spacecrafts to the expected task progress in a finite time interval Δt . Let $\mathcal{H}_i^{pred}(t)$ denote the expected information cost for spacecraft i at time t , computed by the central computing system using a simulation setup. Then the fault detection metric for a particular spacecraft is given as follows:

$$\mathcal{H}_{m_i}(t) = abs \left(1 - \frac{\Delta \mathcal{H}_i(t)}{\Delta \mathcal{H}_i^{pred}(t)} \right) = abs \left(1 - \frac{\mathcal{H}_i(p_i[t]) - \mathcal{H}_i(p_i[t - \Delta t])}{\mathcal{H}_i^{pred}(p_i[t]) - \mathcal{H}_i(p_i[t - \Delta t])} \right).\tag{10}$$

The above metric can be used to distinguish between whether a spacecraft is exhibiting faulty behavior or not as follows:

$$\mathcal{H}_{m_i}(t) \begin{cases} = 0 & \text{No fault} \\ > 0 & \text{Fault has occurred.} \end{cases}\tag{11}$$

Note that the occurrence of a fault can also improve the information gain, i.e., a faulty spacecraft can perform better than expected. However, there is still a need to identify this type of fault because

the spacecraft is not behaving as expected. The fault metric in (10) can also be used to determine if the performance of a spacecraft has improved or deteriorated. Defining $x := \frac{\Delta \mathcal{H}_i(t)}{\Delta \mathcal{H}_i^{pred}(t)}$, table 1 below specifies the criteria to analyze the performance of a faulty spacecraft i .

Fault Case	Condition
Deteriorating performance	$sign(\Delta \mathcal{H}_i(t)) \neq sign(\Delta \mathcal{H}_i^{pred}(t));$ $sign(\Delta \mathcal{H}_i(t)) = sign(\Delta \mathcal{H}_i^{pred}(t))$ and $x < 1$
Improved performance	$sign(\Delta \mathcal{H}_i(t)) = sign(\Delta \mathcal{H}_i^{pred}(t))$ and $x > 1$

Table 1: Identifying spacecraft performance under fault.

Fault Type

The primary goal of the centralized FDIR system is to identify the faulty observer spacecrafts in the network. This work is primarily focused on detecting actuator and sensor faults in the spacecraft. In the GNC architecture, an actuator fault can either occur during the state propagation of a spacecraft or during the optimal sensor placement to observe the target POIs.

In the former case, the faulty spacecraft doesn't follow its assigned orbit, thereby changing the set of POIs which it observes. Such a faulty observer behaves as a rogue spacecraft or can collide with another spacecraft in the vicinity. In the latter case, the onboard sensor fails to accurately point at the POI with maximum variance. Both these fault cases will cause a change in the information gain computed by the observer. In the case of a sensor fault, the variance in observing a POI is different than expected.

Figures 3 and 4 demonstrate the effect of different types of actuator faults on the global information gain \mathcal{H} , as well as the fault signal for different observer spacecrafts. The actuator faults are simulated on the observer spacecraft 0 by adding a random noise to the state of the observer and the pose of the onboard sensor, respectively. In Fig. 3 the occurrence of actuator fault improves the global system performance since the true value of information gain is lower than the expected value. The corresponding plot for the fault signal shows a value of 0 for non-faulty observers and a value of 1 for the faulty observer. This is consistent with the conditions in (11). Figure 4 shows a similar behavior when the actuator fault is implemented during sensor pointing. In this case the global system performance deteriorates.

The above fault cases show that the proposed fault metric performs as expected. However, there is still a need to determine an appropriate threshold to distinguish between system noise and a fault. The occurrence of actuator faults cause a change in the pose $p_i(t)$ of a spacecraft. From the formulation of the global cost \mathcal{H} in (9), it is explicitly clear that a change in pose $p_i(t)$ of a spacecraft will effect the variance in observing a target POI. There is also a more subtle dependence of the set of visible POIs, $\mathcal{S}_i(t)$, on the pose of spacecraft, since the field of view of the onboard sensor will change w.r.t. the spacecraft pose.

Let $\mathcal{S}_i^{pred}(t)$ be the expected set of visible POIs for spacecraft i at time t . Then, to compute a fault threshold, it is first necessary to construct a set of POIs $\mathcal{S}_i'(t)$, such that

$$0 < |\mathcal{H}_i(\mathcal{S}_i'(t)) - \mathcal{H}_i(\mathcal{S}_i^{pred}(t))| \leq |\mathcal{H}_i(\mathcal{S}_i(t)) - \mathcal{H}_i(\mathcal{S}_i^{pred}(t))|, \quad \forall \mathcal{S}_i(t) \subset \mathcal{S}. \quad (12)$$

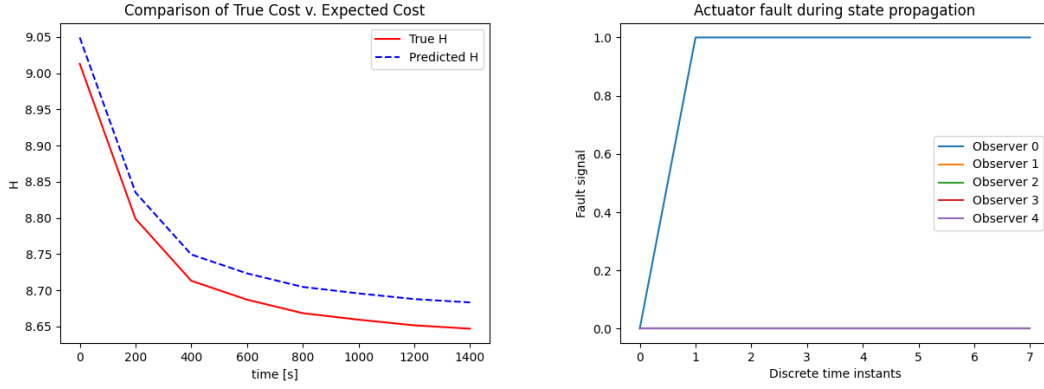


Figure 3: real-time vs. Expected cost under actuator attack I (left); Behavior of fault signal (right).

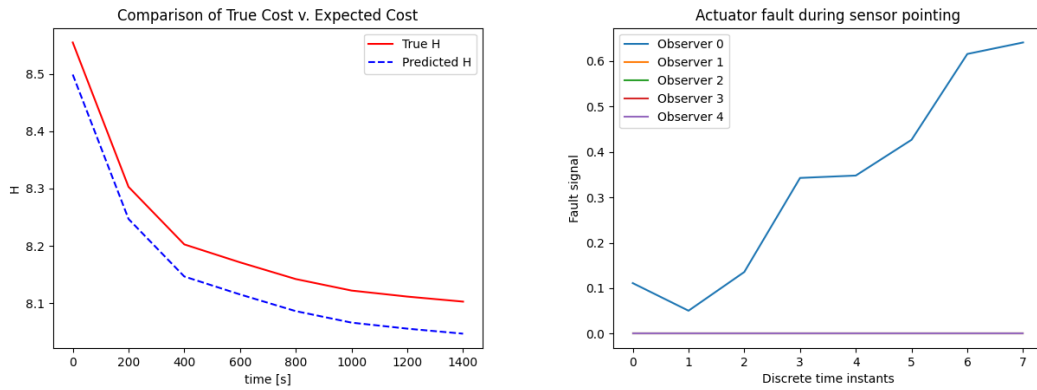


Figure 4: Real-time vs. Expected cost under actuator attack II (left); Behavior of fault signal (right).

The fault threshold for individual spacecrafts can be computed as

$$\tau_i(t) = \text{abs} \left(1 - \frac{\mathcal{H}_i(\mathcal{S}'_i(t)) - \mathcal{H}_i(\mathcal{S}_i(t - \Delta t))}{\mathcal{H}_i(\mathcal{S}_i^{\text{pred}}(t)) - \mathcal{H}_i(\mathcal{S}_i(t - \Delta t))} \right). \quad (13)$$

Equation (13) is used to construct adaptive fault thresholds for individual spacecrafts, where an actuator fault is detected in spacecraft i if $\mathcal{H}_{m_i}(t) > \tau_i(t)$.

IMPLEMENTATION AND RESULTS

Computing the set $\mathcal{S}'_i(t)$ in (12) can become computationally intractable with increasing number of POIs. Therefore, a sampling based approach is taken to approximate the set $\mathcal{S}'_i(t)$ by randomly pointing the onboard sensor within an ϵ -neighborhood of the target POI. Figure 5 demonstrates the construction of this ϵ -neighborhood, where the onboard sensor vector is randomly pointed to any point in the ϵ -neighborhood, thereby changing its FOV and the visible set of POIs. For actuator faults, the value of ϵ can be estimated by analyzing $\frac{\partial p_i}{\partial u_i}$ for the system.

The fault detection framework was incorporated in the simulation setup for the hierarchical planning algorithm.^{2,14} During the orbit assignment phase at time t , the central agent determines the

expected set of visible POIs, $\mathcal{S}_i^{pred}(t)$, for each spacecraft i . This is used to compute the nominal behavior, $\mathcal{H}_i(\mathcal{S}_i^{pred}(t))$, for each spacecraft during a fixed time period of 2 orbits. The fault threshold, $\tau_i(t)$, for each spacecraft is also computed during the orbit assignment phase. In the following plots, 10 target POIs were sampled in an ϵ -neighborhood around the POI with maximum variance, for each spacecraft. The sampled set $\mathcal{S}_i'(t)$ which gives the minimal value for $\tau_i(t)$ determines the fault threshold for spacecraft i at time t . At the local agent level, each spacecraft receives its orbit assignment and tracks the progress of its local information cost, $\mathcal{H}_i(t)$, while propagating the next 2 orbits. At the end of this fixed time interval, each spacecraft transmits its local information cost to the central agent where the centralized FDIR algorithm detects any faulty spacecraft behavior using the metric in (10).

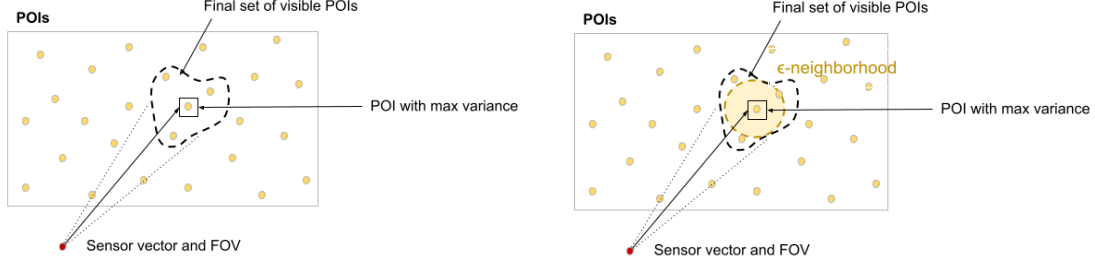


Figure 5: Visible set of POIs for a spacecraft (left); ϵ -neighborhood constructed around POI with maximum variance (right).

The performance of the above technique is tested on different types of actuator faults, as shown in Fig. 6 and 7. In Fig. 6, an actuator fault is implemented during the state propagation of spacecraft 4. In this case, the overall task performance improves and the proposed fault threshold successfully detects the actuator fault. In Fig. 7, an actuator fault is implemented during the sensor pointing phase causing the overall system performance to deteriorate. The proposed adaptive threshold detects the fault after first few time instants.

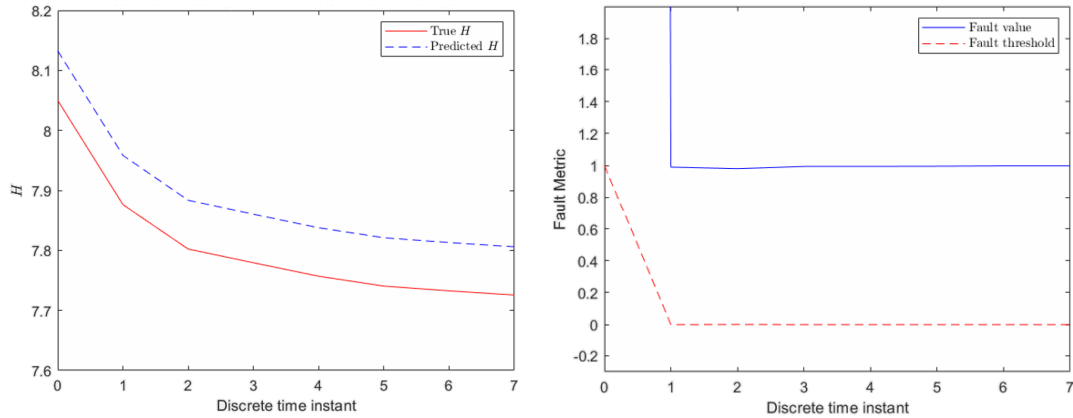


Figure 6: Real-time vs. Expected cost under actuator attack I (left); Adaptive threshold for faulty spacecraft 4 (right).

It is noted that the sampling based threshold design is susceptible to false negatives if the samples are not uniformly distributed or the number of samples are not sufficient to construct a tight estimate.

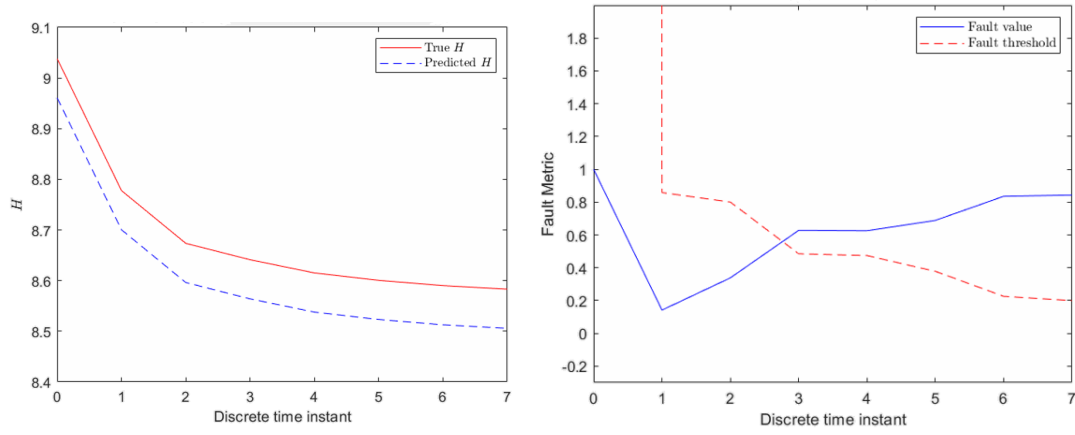


Figure 7: Real-time vs. Expected cost under actuator attack II (left); Adaptive threshold for faulty spacecraft 3 (right).

Finally, designing a threshold for sensor faults depends on the hardware specifications.

CONCLUSION

This work presents a global-to-local task-aware fault detection and identification method for a multi-spacecraft system. When coupled with a replanning method, the algorithm provides a framework for graceful degradation in performance while ensuring mission completion. We derived a gradient-based fault metric used by a central computing system to detect faulty spacecraft in the system in a multi-agent collaborative task such as on-orbit inspection and construction. The fault metric is derived using a global task cost functional that encodes the task of the swarm as a decomposable cost functional for individual agents, while retaining the capability of having an intermediate consensus among locally neighboring agents. This metric is implemented on a low-earth on-orbit collaborative inspection to detect performance deviation from faults and identify the fault type. Further, an adaptive fault threshold is designed to identify actuator faults in individual spacecraft. We demonstrate the approach on the inspection mission by detecting faults using only the global cost metric and identifying the fault type using the higher order gradients of the cost metric.

ACKNOWLEDGEMENT

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration © 2022. California Institute of Technology. Government sponsorship acknowledged.

REFERENCES

- [1] Y. K. Nakka, W. Hönig, C. Choi, A. Harvard, A. Rahmani, and S.-J. Chung, “Information-Based Guidance and Control Architecture for Multi-Spacecraft On-Orbit Inspection,” *AIAA Guidance Navigation Control Conference*, 2021, 10.2514/6.2021-1103.
- [2] Y. K. Nakka, W. Hönig, C. Choi, A. Harvard, A. Rahmani, and S.-J. Chung, “Information-based guidance and control architecture for multi-spacecraft on-orbit inspection,” *Journal of Guidance, Control, and Dynamics*, 2022, pp. 1–18.
- [3] R. C. Foust, E. S. Lupu, Y. K. Nakka, S.-J. Chung, and F. Y. Hadaegh, “Autonomous in-orbit satellite assembly from a modular heterogeneous swarm,” *Acta Astronautica*, Vol. 169, 2020, pp. 191–205.

- [4] B. Yost, S. Weston, G. Benavides, F. Krage, J. Hines, S. Mauro, S. Etchey, K. O'Neill, and B. Braun, "State-of-the-art small spacecraft technology," 2021.
- [5] J. Schoolcraft, A. Klesh, and T. Werne, "MarCO: interplanetary mission development on a CubeSat scale," *Space Operations: Contributions from the Global Community*, pp. 221–231, Springer, 2017.
- [6] F. Flechtner, P. Morton, M. Watkins, and F. Webb, "Status of the GRACE follow-on mission," *Gravity, geoid and height systems*, pp. 117–121, Springer, 2014.
- [7] F. Arrichiello, A. Marino, and F. Pierri, "Observer-Based Decentralized Fault Detection and Isolation Strategy for Networked Multirobot Systems," *IEEE Transactions on Control Systems Technology*, Vol. 23, No. 4, 2015, pp. 1465–1476, 10.1109/TCST.2014.2377175.
- [8] J. L. Paneque, J. Martinez-deDios, and A. Ollero, "Robust Decentralized Context-Aware Sensor Fault Detection with In-Place Self-Calibration," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 3130–3136.
- [9] P. Panagi and M. M. Polycarpou, "Distributed fault accommodation for a class of interconnected non-linear systems with partial communication," *IEEE Transactions on Automatic Control*, Vol. 56, No. 12, 2011, pp. 2962–2967.
- [10] R. K. Ramachandran, N. Fronda, and G. Sukhatme, "Resilience in multi-robot multi-target tracking with unknown number of targets through reconfiguration," *IEEE Transactions on Control of Network Systems*, 2021.
- [11] F. Boem, R. Carli, M. Farina, G. Ferrari-Trecate, and T. Parisini, "Distributed fault detection for interconnected large-scale systems: A scalable plug & play approach," *IEEE Transactions on Control of Network Systems*, Vol. 6, No. 2, 2018, pp. 800–811.
- [12] F. Boem, R. M. Ferrari, C. Keliris, T. Parisini, and M. M. Polycarpou, "A distributed networked approach for fault detection of large-scale systems," *IEEE Transactions on Automatic Control*, Vol. 62, No. 1, 2016, pp. 18–33.
- [13] C. Choi, Y. K. Nakka, A. Rahmani, and S.-J. Chung, "Resilient Multi-Agent Collaborative Spacecraft Inspection," *IEEE Aerospace Conference*, 2023, pp. 1–13.
- [14] Y. K. Nakka, W. Hönig, C. Choi, A. Harvard, A. Rahmani, and S.-J. Chung, "Information-based guidance and control architecture for multi-spacecraft on-orbit inspection," *AIAA Scitech 2021 Forum*, 2021, p. 1103.
- [15] K. T. Alfriend, S. R. Vadali, P. Gurfil, J. P. How, and L. Breger, *Spacecraft formation flying: Dynamics, control and navigation*, Vol. 2. 2009.
- [16] M. Schwager, D. Rus, and J. E. Slotine, "Unifying Geometric, Probabilistic, and Potential Field Approaches to Multi-Robot Deployment," *International Journal of Robotics Research*, Vol. 30, No. 3, 2011, pp. 371–383, 10.1177/0278364910383444.