Human-Centered Fault Recovery for Aerospace Control Systems: Tools for Interactive Troubleshooting

Nathaniel Guy AIAA Research Plan

Proposal Overview

Fault detection within aerospace control systems is usually based on either hardware redundancy, where multiple units of the same hardware exist to provide bases for comparison and fallback opportunities, or on software models, where the control plant is simulated and faults can be detected as characteristic discrepancies between the actual output and the simulated output. Once detected, some anticipated classes of faults may be automatically recovered from, but others are more complex and require humans to understand and determine the best resolution strategy. This project proposes to:

1. Determine methods whereby estimated system state, and relationships of a particular fault to specific system output and hardware components, can be more richly and intuitively conveyed to a human operator using a state-of-the-art interactive interface.
2. Demonstrate the efficacy of these methods by implementing them in practice on an actual semi-autonomous aerospace control system, with a generalized interface that can be applied to a multitude of distinct systems.

Relevance

This project responds directly to the AIAA's call to shape the future of aerospace, by working to push the envelope of how semi-autonomous aerospace systems may be operated in the future. This work will contribute to the state-of-the-art for control system fault resolution, and will provide new methods that can be used by ground-based and on-board software throughout the space industry. Additionally, this research will also provide tools to support the goal, laid out in NASA's Science Mission Directorate's Science Plan, of expanding human presence into the solar system and to the surface of Mars in order to advance exploration, science, innovation, benefits to humanity, and international collaboration.

Background and Motivation

The topic of fault detection, isolation and recovery (FDIR) for aerospace control systems is a rich subject that draws from decades of research. In 1976, [1] presented various key analytical methods for system fault detection, including the details of redundancy-based voting systems, which are key to most FDIR systems today. Modern FDIR systems primarily employ either or both of two techniques: the more expensive hardware-based FDIR, and the less expensive but more analytically involved model-based FDIR.

Model-based FDIR systems, which emulate hardware redundancy in software by looking for discrepancies against a simulated model, are capable of detecting faults by generating "residuals," or quantitative indicators of the likelihood of a fault, based on deltas between estimated system performance and actual system output. Furthermore, by establishing a system of rules related to how certain data families correspond to types of faults that can occur, it is possible to isolate specific components of the system in order to prevent further anomalous behavior.

However, many faults are complex, and the traditional automated techniques—whether they employ model-based FDIR, or more advanced statistical or machine-learning-based techniques—may not be up to the task. Cases like these may require extensive human troubleshooting to determine the root cause. In 2012, [2] demonstrated some of the shortcomings of automated approaches for diagnosis of complex faults, and illustrated that visualization-based tools can allow operators to more accurately identify root causes in the event that the automated FD's residuals are incorrect or not specific to the actual issue at hand.

Additionally, for many systems, the recovery system is fail-safe; i.e., when faults are detected and isolated, the system will enter a "safe mode" in which the craft is no longer in danger of further anomalous behavior as a result of the faults, but is also no longer capable of achieving its main mission objectives, at least until intervention by a human operator is possible. [3] acknowledges this fact as a forcing function for building improved automated FDIR techniques, and this issue has no doubt been a motivator for many similar fault recovery system attempts. "Fail-operational" systems—ones in which the system remains fully functional in the presence of faults—are exceedingly difficult to design, as the space of possible anomalies, consisting of varied and complex fault profiles, is so great that time constraints on system development require postponing the resolution of many faults until human operators can perform a thorough diagnostic analysis.

This becomes problematic, however, when a fault's recovery requirements are time-sensitive in nature; if a major mission event such as stage separation or orbital insertion is near, human operators may be required to make time-sensitive decisions based on the limited telemetry that is immediately available. As an extreme example, Range Safety Officers are typically put in the unenviable position of terminating a space launch attempt at the earliest detection of any dangerously off-nominal telemetry. The necessity of presenting an RSO with fault-related telemetry in a clear, unmistakable, and easily comprehensible format should be apparent to the reader.

For these reasons, it's imperative to design aerospace ground software such that it presents data in the clearest possible fashion, with detected faults being displayed as obvious alarms which require resolution, and modern ground software has made some strides in this area, in part thanks to NASA efforts [4]. However, while this software is designed to display fault detection results clearly and unambiguously, the troubleshooting aspect of fault recovery is often left entirely to the user; a highly trained operator must review telemetry channels and cross-reference fault timestamps with data around that time, usually pouring through mountains of plotted channels and alarm lists to attempt to make some sense of the fault. Finding the core reason behind a system anomaly—be it a failed pressure transducer causing off-nominal thrust-vectoring, or evaporated lubricant causing an antenna deployment failure—can take days or months of after-the-fact data review.

Promisingly, a considerable body of research work has accumulated which shows the benefits of interactive 3D interfaces for quick comprehension of complex systems. Ever since Brooks' seminal publication of [5] in 1988, there has been a large push to turn interactive multimedia technology to the problems of science and engineering, rather than just entertainment. In 2001, it was shown that graphical simulations are not only effective for training operators for space telerobotic operations, but also allow for robotic telemetry visualization that would not otherwise be possible on live video, and simplify identification of system anomalies [6]. More recently, empirical results demonstrated that human operators trying to understand 3D laser scanner data demonstrated increased perceptual speed and recognition accuracy when viewing data in an interactive, stereoscopic 3D format, as compared with a traditional 2D format [7].

The benefits of 3D interactive interfaces for quick comprehension of complicated scenes are well-demonstrated, but there is further work to be done. The space industry on the whole has a vested interest in reducing complex anomaly diagnosis time by using modern data visualization and interactive 3D interfaces to present the operator with a greater array of tools to diagnosis the reasons behind system faults and recover from them confidently. By providing users with greater insight into fault detection and isolation systems, and creating a reusable, generalized framework with hooks into existing control system models, this research will attempt to facilitate the human diagnosis of system faults, and greatly decrease the time required for a human operator to determine the proper recovery steps to take for a non-trivial fault.

Proposed Research

This proposal is for a year-long research project, and given its applicability to a large number of aerospace systems, should result in at least one publication and a number of presentations about this work at conferences such as IEEE VIS or CDC. The project also anticipates the development of an extensible, generalized, and open-source software toolkit for application to a variety of aerospace control systems. The project will be broken up into two primary phases: the problem characterization phase and the problem solution phase.

In the **problem characterization phase**, this project will endeavor to answer the question of how we can best inform and assist the fault recovery process through proactive fault-detection system design, and by application of interactive tools. I will focus primarily on model-based fault-detection systems, and study how fault isolation and residual generation can be customized to present a clearer picture to the operator during human-assisted fault recovery. I will compare techniques for fully-autonomous IR, fully human-operated IR, and semi-autonomous, human-assisted IR, in order to better isolate those types of faults that would most benefit from the approaches that will be employed in the second phase. I plan to augment and refine this data by conducting a series of interviews with operators and developers of aerospace systems about operational pain points. I anticipate that the area where these proposed techniques will show the greatest benefit is that of complex mechanical systems where physical layout is very relevant and where a physical understanding of the problem is highly beneficial to the operator: one such example would be a planetary rover with a stuck wheel assembly.

In the **problem solution phase**, I will work closely with system operators and developers to build tools that address these issues by a) better illustrating faults that have occurred, through intelligent analysis and design of their generated residuals, and b) presenting operators with an immersive and intuitive 3D interface for the immediate recognition and facilitated troubleshooting of these faults. I will draw upon previous work in 3D data visualization and interactive interface design, both in academia as well as in the video game industry, to construct these tools using state-of-the-art software techniques and frameworks. In order to test the efficacy of the APIs and interactive tools developed, user studies will be conducted on actual operational aerospace systems, and I will iterate on and refine the tools in response to the user feedback. I anticipate that our user studies will show improved insight and accuracy in problem diagnosis as a direct result of using the tools developed as a part of this research.

Many preexisting software tools will be used to enable and accelerate implementation of the methods that this research will investigate, and I plan on leaning heavily on industry-tested tools for interactive simulation as a starting point for development. One such tool is the Unity 3D engine, a game and visualization engine that enables rapid prototyping and development and currently has 45% of the worldwide game engine market share [8]. I have experience with this tool and many others from my work at Nintendo and at JPL, and am confident in my ability to use them to develop fully-featured applications capable of interacting with aerospace systems.

For the primary subjects of my user research, I plan to work with two distinct academic labs and their testbeds. The University of Washington’s RAIN Lab has already made impressive progress building a formation control testbed consisting of multiple remote-operated and semi-autonomous two-wheeled vehicles, and is actively working on development of unmanned aerial vehicles for formation control testing as well. In addition, the Tohoku University Space Robotics Lab has developed two- and four-wheeled micro-rover testbeds for an ongoing lunar exploration project, and is very receptive to advanced visualization systems for human operators, as their prior work shows [9]. I will be an active member of both of these labs over the next year, and I anticipate innumerable opportunities to iterate over methods developed in the course of this study, and, by applying to them to actual testbed operations, to support a number of other ongoing research projects.

I anticipate that it will also be necessary for the controls or safety engineer, when initially configuring the fault detection system, to add semantic “hooks” that allow the system to provide greater insight to the human operator. To give an example: in a certain fault detection system, assume that a certain residual is generated from telemetry from a motor actuator, and when the residual exceeds a certain threshold, an alarm is sounded, and the system enters a fail-safe mode. A human operator presented with the task of recovery in this situation would benefit from an immediate understanding of a) the telemetry channels that factor into the off-nominal residual, b) the electromechnical components directly related to the fault, c) the state of the system immediately before the anomaly occurred, and d) if this is a well-known fault, advice from system designers to assist with the troubleshooting process. This project plans to investigate the best methods to establish these semantic links during the system design phase, and to fuse together all of this data into a single interface. This interface could allow a FDIR flow similar to the sample flow shown in Figure 1. A suggestive mockup of what this interface might look like is shown in Figure 2.

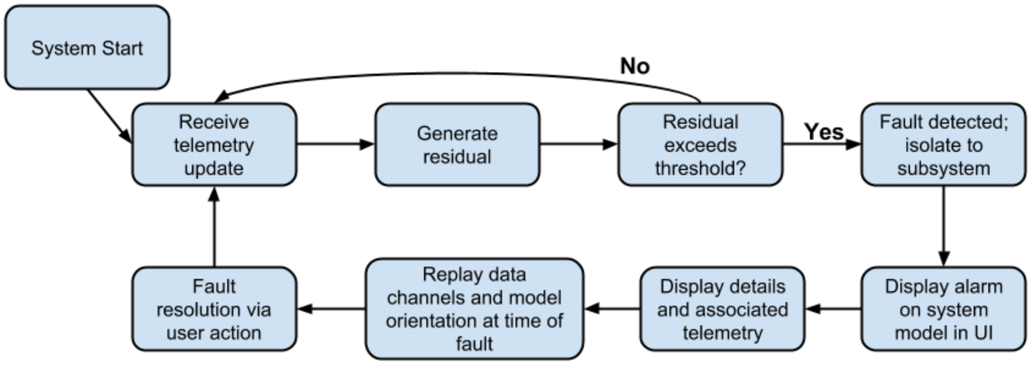
System Start
Receive telemetry update
Generate residual
Residual exceeds threshold?
No
Yes
Fault detected; isolate to subsystem
Display alarm on system model in UI
Display details and associated telemetry
Replay data channels and model orientation at time of fault
Fault resolution via user action

Figure 1: One possible flow for FDIR troubleshooting. Note that some transitions are in response to UI interaction.

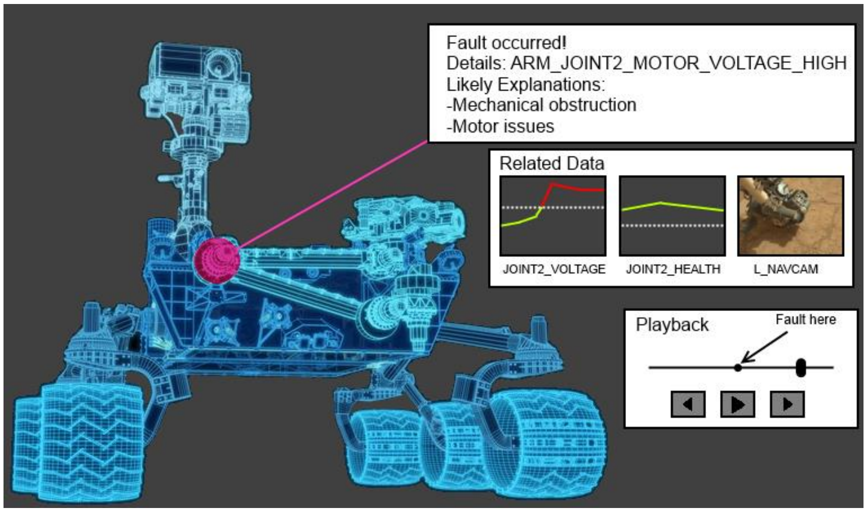
Fault occurred!
Details: ARM_JOINT2_MOTOR_VOLTAGE_HIGH
Likely Explanations:
-Mechanical obstruction
-Motor issues

Figure 2: A mockup of what a data-fused interface for fault troubleshooting might look like. [10]

Proposal Summary

At the conclusion of this proposed work, I will have developed methods to gain insight and introspection into aerospace control system faults and their recovery methods, and developed a generalized, extensible toolkit that allows these methods to be applied to a variety of semi-autonomous systems. I expect to address the following questions:

1. How can a model-based FDIR system be designed to allow a greater fraction of complex faults to be quickly recoverable via human interaction?
2. Once designed, how can a fault-tolerant system be augmented by an advanced 3D interface so that a human operator can optimally assist in fault recovery?
3. Which user interaction systems are most conducive to the quick diagnosis and resolution of a fault that has occurred?

Since this work will produce tools applicable to FDIR systems for generic aerospace control systems, it may help to enhance the operational capabilities of a number of systems, including those which will serve as testbeds for the research. It is also hoped that the application of modern 3D graphics and interactive visualization techniques to aerospace control system problems will encourage further research at the intersection of these two still relatively disconnected disciplines.

References

[1] A. Willsky, "A Survey of Design Methods for Failure Detection in Dynamic Systems," *Automatica,* 1976.

[2] E. Garduno, S. P. Kavulya, J. Tan, R. Gandhi and P. Narasimhan, "Failure Diagnosis of Complex Systems," in *USENIX Large Installation System Administration (LISA) Conference*, 2012.

[3] N. Holsti and M. Paakko, "Towards Advanced FDIR Components," in *Data Systems in Aerospace (DASIA)*, 2001.

[4] J. Fox, J. Breed, K. Moe, R. Pfister, W. Truszkowski, D. Uehling, A. Donkers and E. Murphy, "User-Centered Design of Spacecraft Ground Data Systems," in *Second International Symposium on Spacecraft Ground*, 1999.

[5] F. Brooks, "Grasping Reality Through Illusion—Interactive Graphics Serving Science," in *SIGCHI Conference on Human Factors in Computing Systems*, 1988.

[6] J. Lane, C. Carignan and D. Akin, "Advanced Operator Interface Design for Complex Space Telerobots," *Autonomous Robots,* vol. 11, no. 1, 2001.

[7] T. Fujiwara, T. Kamegawa and A. Gofuku, "Stereoscopic Presentation of 3D Scan Data Obtained by Mobile Robot," in *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2011.

[8] "Unity - Fast Facts," [Online]. Available: http://unity3d.com/public-relations. [Accessed January 2015].

[9] N. Britton, K. Yoshida, J. Walker, K. Nagatani, G. Taylor and L. Dauphin, "Lunar Micro Rover Design for Exploration through Virtual Reality Tele-operation," in *Field and Service Robotics (FSR)*, 2013.

[10] *Mars Curiosity Rover wireframe image is property of National Geographic Channels/ Maas Digital.*