Paper ID: 32540
oral
student

SPACE SYSTEMS SYMPOSIUM (D1)
Interactive Presentations (IP)

Author: Mr. Peter Schulte
School of Aerospace Engineering, Georgia Institute of Technology, United States

Mr. David Spencer
Georgia Institute of Technology, United States
Dr. Neil Smith
King Abdullah University of Science and Technology (KAUST), Saudi Arabia
Prof. Matthew McCabe
King Abdullah University of Science and Technology (KAUST), Saudi Arabia

# DEVELOPMENT OF A FAULT PROTECTION ARCHITECTURE BASED UPON STATE MACHINES

**Abstract**

This paper describes an advance in the state-of-the-art of spacecraft fault protection through development of an architecture that utilizes state machines for Fault Detection, Isolation, and Recovery. Through the application of state machine logic, the architecture actively responds to hardware and software faults, allowing autonomous recovery to a safe state. The study leverages a MATLAB/Simulink six degree-of-freedom simulation environment, allowing the evaluation of the fault detection algorithms in flight-like mission scenarios. The modularity of the simulation environment allows the investigator to define the sensor/actuator suite and software modules to test various combinations of algorithms and hardware models.

Within Simulink, a tool called Stateflow is used to implement complex logical relationships by using state charts, also known as state machines, to represent the current state of different spacecraft hardware or software components. The fault protection architecture is developed as a Stateflow block that receives measurements of state variables from spacecraft software and hardware models in Simulink to estimate the current state of the system. Based on that state, the fault protection algorithms determine if any faults are present (detection), determine the type of fault and likely location (isolation), and command actions to contain or prevent further faults (recovery). Outputs from the fault protection Stateflow charts will issue commands back to the spacecraft software and hardware models, allowing an automated response to spacecraft faults.

This fault protection architecture is based on several requirements; it is designed to be generic, modular, and portable to flight software. The simulation environment allows setting parameters such as physical dimensions and orbit elements, is applicable to a multitude of possible mission scenarios and allows alternate configurations, such as multiple cooperative or non-cooperative spacecraft. The visual block diagram environment offered by MATLAB/Simulink can be reconfigured to test many combinations of software and hardware components. Finally, the capability to easily convert into flight software code (i.e. autocoding) is available through the MATLAB/Simulink platform.

The study advances the state-of-the-art in fault protection and builds on previous work by bringing together capabilities including Stateflow decision logic, autocoding to flight software, and model-based design into a single generic, modular architecture that is portable to embedded systems. The resulting architecture is intended to be broadly applicable for space flight missions, advancing flight system capabilities for automated mission operations.