Simulators, Software and Small Satellites: Testing in Tight Spaces

Ronnie Killough
Southwest Research Institute
6220 Culebra Rd
San Antonio, TX 78238
210-522-3616
ronnie.killough@swri.org

John Hanley
Southwest Research Institute
6220 Culebra Rd
San Antonio, TX 78238
210-522-2884
john.hanley@swri.org

Alan Henry
Southwest Research Institute
6220 Culebra Rd
San Antonio, TX 78238
210-522-5238
alan.henry@swri.org

Robert Klar
Southwest Research Institute
6220 Culebra Rd
San Antonio, TX 78238
210-522-5052
robert.klar@swri.org

Scott Miller
Southwest Research Institute
6220 Culebra Rd
San Antonio, TX 78238
210-522-4249
scott.miller@swri.org

***Abstract*—In the world of spacecraft integration and test, "Test As You Fly" (TAYF) is the mantra. This is sometimes easier said than done since methods to stimulate the various sensors can be difficult, and commanding many flight actuators while the spacecraft is sitting on the ground is impractical. As such, a mixture of sensor stimulation, sensor emulation and other environment simulations are used to dupe the spacecraft into believing it is flying, thus enabling the flight software (FSW) and control algorithms to be tested in their final flight configurations.**

When building and testing very small satellites, some additional obstacles are present. For example, installing external simulators and emulators necessary for activities such as attitude determination and control (AD&C) testing and mission simulations late in the integration schedule may be precluded due to a lack of physical access. Using special electrical ground support equipment (EGSE) interfaces to stimulate the spacecraft may introduce other challenges - no one wants to stand before a launch readiness review board and say that one set of FSW was used during final mission tests and simulations, but that another version will be loaded just prior to launch!

The Cyclone Global Navigation Satellite System (CYGNSS) mission is a constellation of eight microsatellites that is currently in the integration and test phase. The CYGNSS payload is comprised of a set of Global Positioning System (GPS) receivers, which compare direct and ocean-reflected signals to measure surface wind speeds. Each microsat has a suite of AD&C sensors and actuators that must be simulated or stimulated during test. Designing a simulation and test environment that was cost-effective for a NASA Class D mission and dealt with the limitations of size, while maintaining a TAYF philosophy, presented significant challenges.

This paper discusses how satellite size translated into challenges in the design of the FSW and EGSE, and how this impacted the overall test and simulation approach. Unique and creative solutions developed will be described, such as the use of "man-in-the-middle attack" techniques (commonly used by cyber hackers) to allow the FSW to execute normally even while communicating over non-flight interfaces. Finally, the pros and cons of the various design choices will be discussed.

TABLE OF CONTENTS

1. INTRODUCTION

In the world of spacecraft integration and test (I&T), "Test As You Fly" is the mantra. That is to say, when testing an observatory it is important to use, to the greatest extent possible, the same hardware, the same flight software, the same ground systems, and the same command sequences that will be used on orbit. That also implies that the observatory should "never experience expected operations, environments, stresses or their combinations for the first time during the [on orbit] mission" [1].

Applying TAYF principles is sometimes easier said than done, since methods to adequately stimulate the various sensors can be difficult, and commanding many flight actuators while the spacecraft is sitting on the ground is often impractical. Further, when there are synchronization and closed-loop testing requirements, meaning that the various stimulation inputs and actuator outputs cannot be treated independently, the test environment gets much more complicated. In spite of these challenges, a mixture of sensor stimulation, sensor emulation and actuator and environment simulations are typically used to dupe the spacecraft into believing it is flying and enable the flight software and control algorithms to be tested in their final flight configurations. Some missions have the resources to use special test equipment, facilities and techniques, such as suspending the spacecraft to allow some level of freedom of rotation, and realistic magnetic field emulation. When model-based simulators are used in place of sensors during testing of a flight model observatory, connector savers are utilized to prevent wear on the flight connectors until the simulators are removed from the system for the last time. Most missions also employ a fully-functional “flatsat” that enables physical access to spacecraft components and inter-connections, such that stimulators and emulators can be selectively inserted in place of sensors and actuators to meet the full complement of test and verification requirements.

When building and testing very small satellites (nanosatellites and microsatellites) additional obstacles are present. It is not difficult to identify the mechanical challenges associated with building something small; issues such as component clearances, routing of cables, order of assembly, and limited physical access for the hands and tools of assembly technicians are easy to imagine. What may be less obvious is that these mechanical constraints can also have significant impacts in the design of interface simulators, FSW, and to overall verification and mission simulation planning. The limited budget associated with small satellite missions presents additional challenges. The types of resources available to these programs (such as a dedicated “flatsat”) may be out of reach of small mission budgets.

For example, installing external simulators and emulators necessary for activities such as attitude determination and control (AD&C) testing and mission simulations late in the integration schedule may be precluded due to a lack of physical access, threatening the TAYF approach. Forget connector savers—just putting a tool on the jack screws of a component connector may be impossible once the satellite is “buttoned up”. Using special electrical ground support equipment (EGSE) interfaces to stimulate the spacecraft may introduce other challenges: no one wants to stand before a launch readiness review board and say that one set of FSW was used during final mission tests and simulations, but that another version will be loaded just prior to launch!

The CYGNSS mission has faced additional challenges. First, the CYGNSS mission is comprised of eight flight model (FM) observatories that each have to be tested. While there was budget to build an engineering model (EM) observatory, it was largely intended as a mechanical and electrical pathfinder for the FM observatories. As such, it is not really a “flatsat” in the fullest sense, and being a pathfinder, there are some differences between the EM and FM observatories. Full verification and validation credit, then, must be taken on one or more of the FM observatories, meaning that some way to insert simulators into the flight test environment must be implemented. Requirements that only need to be verified once could be tested on a single FM observatory. However having a consistent test configuration and a single set of well-vetted test scripts that are executed on all eight observatories is attractive when time and money are at a premium.

2. THE CYGNSS MICROSAT

The Cyclone Global Navigation Satellite System (CYGNSS) [2] is a constellation of eight microsatellites (microsats) scheduled to launch in October 2016 and will operate in low-earth orbit (LEO) at an inclination of 35 degrees. The core structure of each microsat, with solar arrays stowed, is approximately 40cm square by 10cm tall. CYGNSS is currently in the integration and test phase. Each microsat contains a delay doppler mapping instrument (DDMI), which receives direct signals from GPS satellites, as well as signals reflected off the ocean surface. The direct signals pinpoint the location of the microsat, while the reflected signals respond to ocean surface roughness, from which wind speed is derived. Figure 1 illustrates the CYGNSS concept.

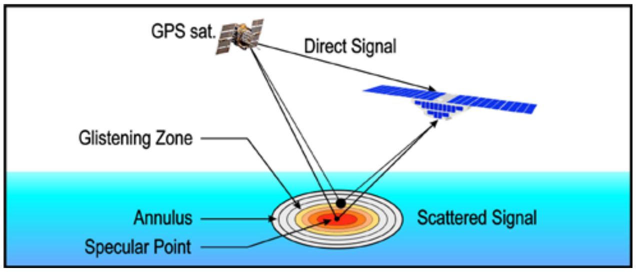

Figure 1. CYGNSS Concept of Operations

CYGNSS is an Earth Venture Mission (EVM) under the National Aeronautics and Space Administration (NASA) Earth System Science Pathfinder (ESSP) program. CYGNSS is a Category 3 Class D mission. This carries with it a lower budget and a provision for tolerating more risk than most missions. Because of the physical size limitations of the microsats, CYGNSS faced the additional challenges that were outlined in the previous section.

Each microsat has a suite of AD&C sensors and actuators that must be simulated or stimulated during ground testing. The single science instrument, the DDMI, is a GPS receiver, and also serves as a sensor input to the AD&C subsystem. Conversely, the instrument requires attitude knowledge from the AD&C subsystem, resulting in coupling between these two subsystems. A partial block diagram of the CYGNSS spacecraft is shown in Figure 2.

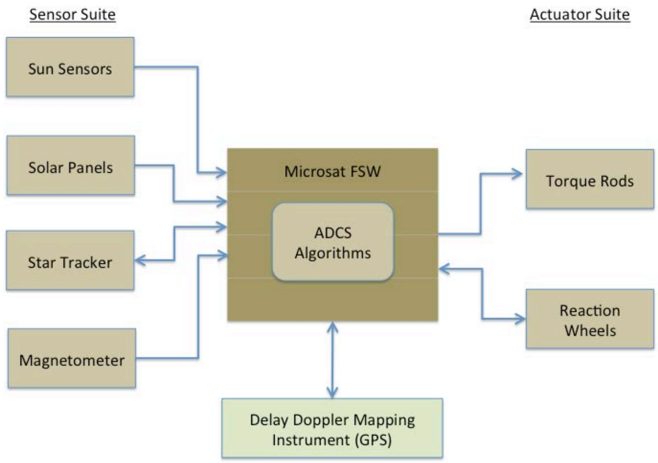

Figure 2. CYGNSS Partial Block Diagram

The FSW communicates with the AD&C sensors and actuators through a combination of Universal Asynchronous Receiver-Transmitter (UART) and Inter-Integrated Circuit (I^2C) interfaces. The interfaces to the DDMI include a UART for commanding and housekeeping telemetry and a SpaceWire interface for science data.

3. TESTING THE CYGNSS MICROSAT

Designing a simulation and test environment that was cost-effective for a Class D mission, dealt with the limitations of size, was adequate to test each fully-integrated microsat, and maintained a “test as you fly” philosophy presented significant challenges.

These challenges were further exacerbated when, to streamline the schedule and meet the Class D budget constraints associated with this class of mission, a goal was established of using the same dynamics simulation system from AD&C algorithm testing to initial software testing, through I&T, and into mission simulations and rehearsals. Since CYGNSS is comprised of eight microsats, the simulator had to be inexpensive and easy to replicate. Several copies of the dynamics simulator would be needed to meet schedule because multiple FM microsats would have to be tested in parallel. A copy of the simulator would also be needed in the AD&C laboratory, the FSW laboratory, the Observatory I&T laboratory, and the Mission Operations Center (MOC).

To meet these challenges, a Spacecraft Dynamics Simulator (SDS) was conceived that had the following basic requirements:

- Emulated each of the sensor and actuator I/O interfaces per the vendor Interface Control Documents (ICDs)
- Modeled the behavior and performance of each of the sensors and actuators
- Implemented environment models appropriate for the CYGNSS mission design

As described in the prior section, most of the interfaces to the AD&C sensors and actuators are via a UART. This allowed the SDS hardware design to be relatively simple and easy to replicate—it consists largely of a rack-mounted Linux PC with a multi-port RS-422 I/O card. The SDS can be connected directly in place of the individual AD&C sensors and actuators.

Figures 3, 4, and 5 illustrate several scenarios for how the SDS was used throughout the development and test of the CYGNSS observatories.

In the AD&C laboratory, a commercial evaluation board is used to execute the FSW. The SDS is connected directly to the individual AD&C component interface ports on the evaluation board. Because the AD&C algorithms only require GPS data (and not the full science data) the truth model in the SDS is used as the GPS data source. The interface to the torque rods on the observatory is via an I^2C bus, but neither the commercial evaluation board nor the SDS contains such an interface. As a workaround, the torque rod commands are transmitted over a separate RS-422 interface to the SDS when used in the AD&C laboratory.

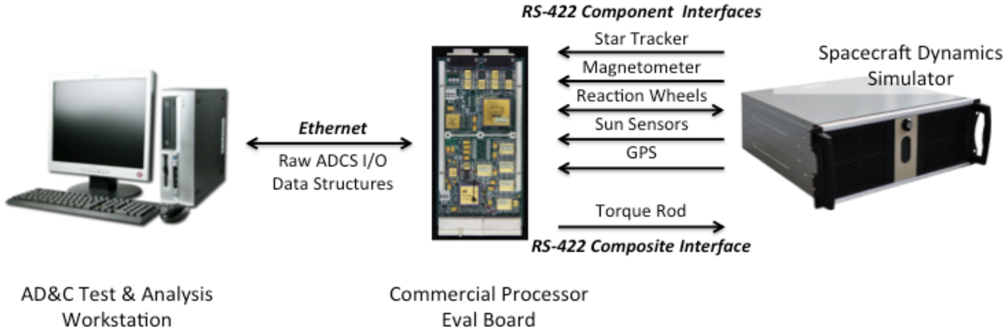
RS-422 Component Interfaces
Star Tracker
Magnetometer
Reaction Wheels
Sun Sensors
GPS
Torque Rod
RS-422 Composite Interface
Spacecraft Dynamics Simulator
Ethernet
Raw ADCS I/O Data Structures
AD&C Test & Analysis Workstation
Commercial Processor Eval Board

Figure 3. AD&C Algorithm Development and Test Lab Configuration

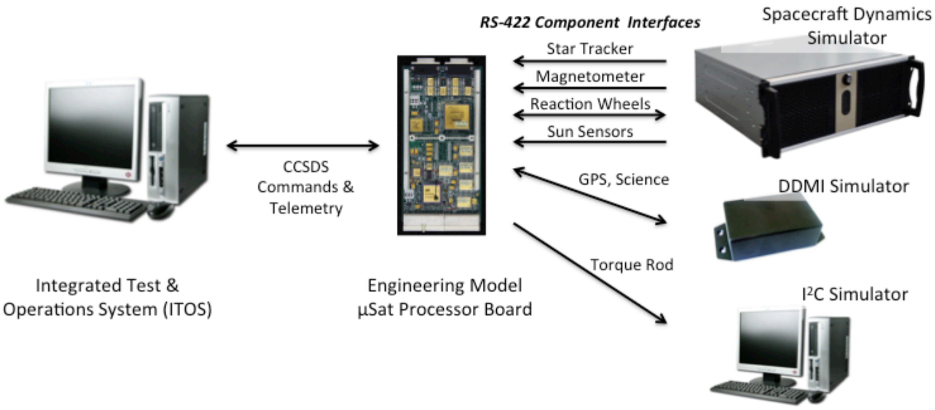
RS-422 Component Interfaces
Spacecraft Dynamics Simulator
Star Tracker
Magnetometer
Reaction Wheels
Sun Sensors
CCSDS Commands & Telemetry
GPS, Science
DDMI Simulator
Torque Rod
I²C Simulator
Integrated Test & Operations System (ITOS)
Engineering Model µSat Processor Board

Figure 4. FSW Development and Test Lab Configuration

In the FSW laboratory, an engineering model of the processor board is used that contains the full complement of features and interfaces. This allows the torque rod commands to be sent over the I^2C bus to a simple I^2C simulator, in this configuration. Because the FSW must be tested with the full science data, GPS data is sourced from another instrument simulator instead of the SDS.

In the Observatory I&T laboratory, the microsat is "buttoned-up" and access to the individual component connectors is no longer possible. In this configuration, simulators and stimulators cannot be connected directly in place of flight components. Nevertheless, the Comprehensive Performance Tests (CPTs), Mission Simulations and other tests need to exercise the full functionality of the microsat. Consequently, the SDS had to be designed to support two modes:

- Component Mode – this is the default mode, in which each of the sensor/actuator models is connected directly in place of the real component, and sensor telemetry and actuator commands traverse the same physical interface as the real component.
- Composite Mode – in this mode, all sensor telemetry and actuator commands traverse a single separate EGSE interface.

Use of Composite Mode is illustrated in Figure 5. In this mode, a single UART interface multiplexes the data that is normally relayed by several individual UART interfaces.

4. A MAN-IN-THE-MIDDLE HIJACK

Implementing Composite Mode in the SDS is only half of the battle because the FSW expects to receive sensor telemetry and send actuator commands via the individual component UART interfaces, not via the EGSE interface. If a separate special version of the FSW is used for observatory-level AD&C tests, this would be significant departure from a TAYF philosophy, resulting in an uncomfortable level of risk even for a Class D mission. To resolve this issue, some techniques were applied to the design of the FSW that are similar to the ones used by cyber hackers in attacking software applications: a combination of a man-in-the-middle attack and a dynamically linked library (DLL) hijack attack.

A man-in-the-middle (MITM) attack is one in which a third party inserts himself as a relay/proxy into the communication path between two or more systems. A MITM attack exploits the real-time transfer and processing of data. A MITM attack allows an attacker to intercept, send and receive data without the communicating parties knowing [3]. The concept of a MITM attack is illustrated in Figure 6.

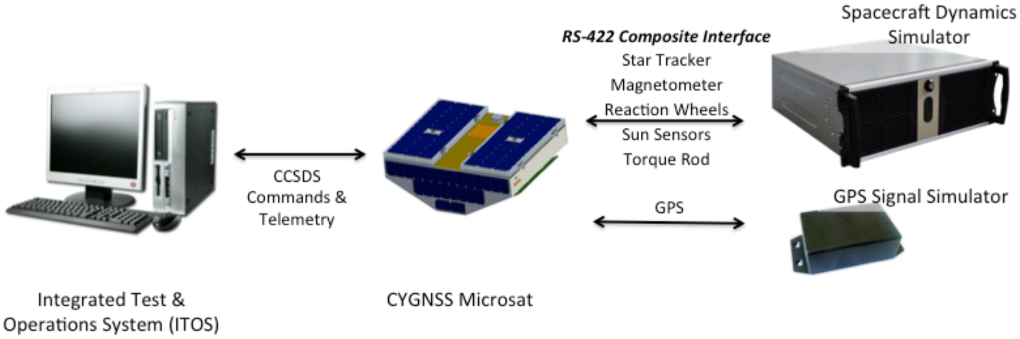
RS-422 Composite Interface
Spacecraft Dynamics Simulator
Star Tracker
Magnetometer
Reaction Wheels
Sun Sensors
Torque Rod
CCSDS Commands & Telemetry
GPS
GPS Signal Simulator
Integrated Test & Operations System (ITOS)
CYGNSS Microsat

Figure 5. Observatory I&T and Mission Simulation Configuration

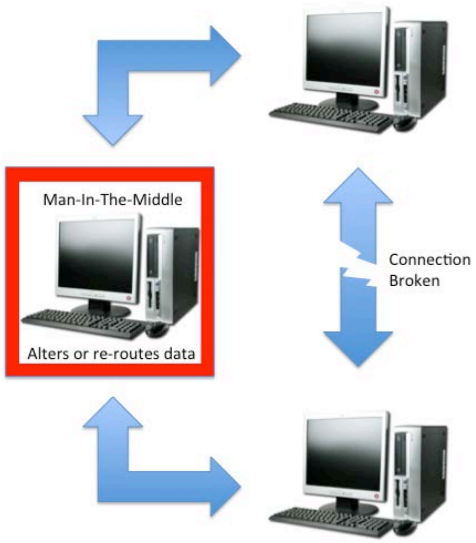

Figure 6. Man-In-The-Middle Attack Concept

One way to accomplish a MITM attack from a software perspective is to impersonate part of an application. A DLL hijack attack is one in which a software application links to a "fake" DLL that has been installed by an attacker [4]. The fake DLL may mimic the same timing and behavior as the real DLL, but in fact may be performing additional functions and/or re-routing the data to an alternate destination.

Leveraging these two techniques, a Composite Hijacker was developed that, when activated and configured, will inject itself into the communications path between the sensors and actuators and the AD&C component I/O processing software. Commands that are sent to an actuator get intercepted and re-routed by the Composite Hijacker to the simulated actuator in the SDS. Similarly, any requests for telemetry from an AD&C sensor are intercepted by the Composite Hijacker and rerouted to the sensor model in the SDS, and the response from the modeled sensor is routed back to the AD&C algorithm. The technique used is such that the core FSW execution and timing is essentially unaffected. The microsat FSW calls the exact same functions as it does when the Composite Hijacker is not executing, and the core FSW is completely agnostic to the fact that those functions have been "hijacked" by the Composite Hijacker.

Figure 7 illustrates the operation in Component Mode, using the reaction wheels as an example. In this case the SDS is not connected, the Composite Hijacker has not been activated, and the FSW communicates directly with the reaction wheels in the nominal fashion. The Component Mode configuration is used during interface checks and component functional tests, to ensure that the microsat can successfully communicate with the real sensors and actuators, and that they correctly respond to the required command set. Figure 8 illustrates how the Composite Hijacker, when activated, intercepts and reroutes reaction wheel commands to the simulated reaction wheel in the SDS. The Composite Mode configuration is used for tests that require flight-like execution of the AD&C algorithms after the observatory has been fully integrated.

This technique is applied to all of the sensors and actuators. The Composite Hijacker was designed so that each sensor and actuator interface could be individually configured to traverse the Component interface or Composite interface. In addition, the Composite Hijacker is capable of sending actuator commands to both the real and simulated actuator simultaneously. This allows the FSW to execute the AD&C algorithms using the simulated inputs and outputs, while allowing for external measurements and observations to ensure the real actuators are responding to the commands. For example, a magnetometer can be used to measure the torque rod output even when using the modeled torque rod in the SDS.

There are some obvious potential problems with this approach. From a TAYF perspective, at first blush it appears that a special build of the FSW would be required to utilize Composite mode. However, this was not necessary—the Composite Hijacker is compiled into the actual FSW image. The same FSW binary used to test the observatories is the same FSW image that CYGNSS will launch with. However, the Composite Hijacker is not activated except when conducting tests with one or more simulated sensors and actuators.

The risk then shifts to whether the Composite Hijacker could ever be activated on orbit. CYGNSS employs several layers of protection to ensure this does not occur. The Composite Hijacker can only be activated when the FSW is booted—there is no logic flow that allows it to be activated

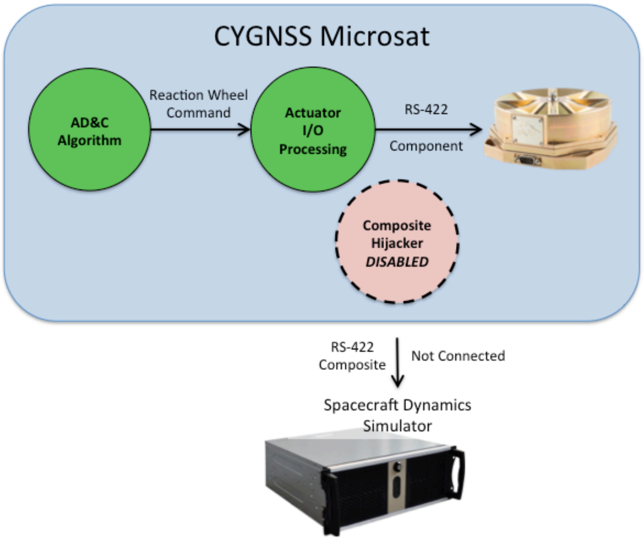

Figure 7. Normal Component Mode I/O - Reaction Wheel Assembly

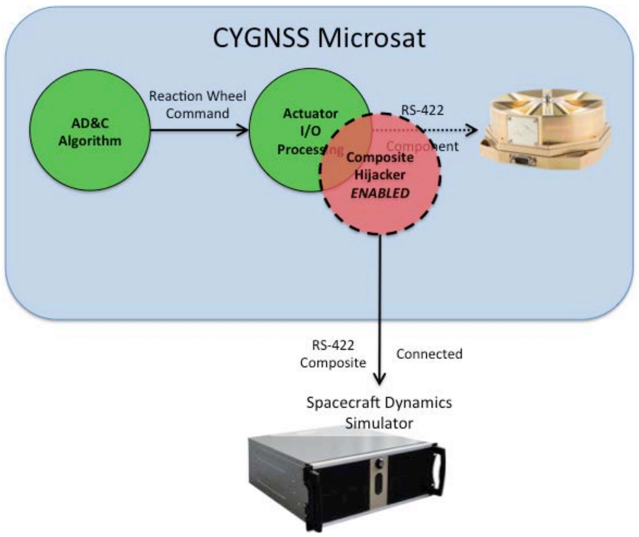

Figure 8. Composite Mode I/O - Reaction Wheel Assembly

once the FSW is already executing. Further, when the FSW boots, the Composite Hijacker is not activated unless (a) voltage is detected on the physical RS-422 interface that serves as the Composite interface and (b) a special pattern is present in a “level zero” configuration register. In spite of these protections, if the Composite Hijacker is somehow activated, it checks a level zero register to determine which sensor and actuator interfaces to reroute. If the register contents are zero (which is always true at power-up) or do not reflect a valid configuration, the Composite Hijacker takes no action.

The last issue also relates to TAYF principles. Rerouting the sensor and actuator messages would appear to have the potential to significantly change the FSW timing. However, the design is such that the FSW timing is affected at a negligible level and well within required timing margins. The Component interfaces are mostly UARTs and the Composite interface is also a UART. This permits using similar hardware and driver software at the lowest level. Because the FSW commands actuators and polls sensors in a synchronous manner, the time to transmit and receive data over the Composite interface is the same as that required to transmit and receive data over the Component interfaces. Since the Composite Hijacker functions effectively replace the nominal Component functions (similar to a DLL hijack), FSW task switching and stack operations are nearly identical regardless of being in Component Mode or Composite Mode. For the torque rods, which are not normally accessed through a UART interface, the command length is only a few bytes and timing differences are negligible.

5. CONCLUSION

This paper describes a novel approach to testing a constellation of satellites faced with budget and resource constraints common to small missions and spacecraft. The approach utilizes a single multi-mode spacecraft dynamics simulator coupled with a special "man-in-the-middle" attack software module that is built right into the flight software. The simulator was used from early development, and continues to be used as CYGNSS is completing the integration and test phase. The “one simulator” goal was largely achieved, except for the Monte Carlo simulations that needed to be done in a software-only environment in order to run faster-than-real-time. The approach resulted in significant cost and schedule savings, which was needed on a small Class D mission budget, because it provided a sophisticated level of testing without requiring expensive test equipment and fixtures. Moreover, the approach successfully dealt with the special challenges of size and physical access inherent in microsat designs, that would otherwise preclude the use of emulators in place of sensors and actuators on the flight observatories during assembly I&T and mission simulations.

For those considering using a similar approach, it is important to note that it requires significant advanced planning and coordination among the various teams and organizations that will be involved in developing and using the simulator. While the potential savings is great, those savings can be eaten up in cleaning up the mess created by a lack of early and adequate planning & inter-organizational communication.

The solution, while effective, does have some disadvantages. First, using the same simulator throughout design, development and I&T means less independence. Using different test environments at various phases can sometimes reveal problems that a single test environment would not. CYGNSS dealt with this risk by having a third party conduct an independent review of the simulator models.

Second, while the design goes a long way towards adhering to TAYF principles, the execution of the FSW is slightly modified during certain tests. However, the same can be said for most any mission—TAYF is best seen as a guiding principle and not a requirement, since such a requirement could not be met by any pre-launch test program if interpreted in the strictest sense.

REFERENCES

[1] White, J., L. Tilney (2013). Applying the “Test Like You Fly” (TLYF) Process to Flight Software Testing, ©Aerospace Corporation. http://flightsoftware.jhuapl.edu/files/2013/talks/FSW-13-TALKS/TLYF_Apply2FSW_Dec2013r1.pdf

[2] Ruf, C., S. Gleason, Z. Jelenak, S. Katzberg, A. Ridley, R. Rose, J. Scherrer, V. Zavorotny . The NASA EV-2 Cyclone Global Navigation Satellite System (CYGNSS)

Mission, Presented at 93rd American Meterological Society Annual Meeting, Austin, TX, January 2013, 7 p.

[3] "Man in the Middle Tutorial: Learn About Man-in-the-Middle Attacks, Vulnerabilities and How to Prevent MITM Attacks," written by Neil DuPaul, http://www.veracode.com/security/man-middle-attack

[4] "Analyzing DLL Hijacking Attacks," written by Chris Sanders, last modified on 13 Oct. 2010, http://www.windowsecurity.com/articles-tutorials/windows_os_security/Analyzing-DLL-Hijacking-Attacks.html

BIOGRAPHIES

Ronnie Killough *is a Program Director in the Space Sciences and Engineering Division at Southwest Research Institute (SwRI). He has 25 years experience on numerous NASA programs including ground systems development for the shuttle/station mission control center and flight software for IMAGE, Swift, Deep Impact, and New Horizons. He is currently serving as a deputy systems engineer for software, data systems and fault management on the Cyclone Global Navigation Satellite System (CYGNSS) mission. He is a 1987 graduate of Angelo State University and a 1990 graduate of Texas A&M University. He has bachelors and masters degrees in computer science, with minor concentrations in electrical engineering, accounting, and finance.*

John Hanley *is a Staff Engineer in the Space Sciences and Engineering Division at Southwest Research Institute (SwRI). He has over 15 years of experience on NASA and European Space Agency (ESA) programs for developing flight software, flight digital systems and ground systems; and being a member of systems engineering and project management teams for science instruments and avionics systems. He has worked on: Deep Space 1, IMAGE, Swift, Deep Impact, New Horizons, Van Allen Probes (formerly RBSP), MMS and Solar Orbiter. He is currently serving on the systems engineering team for the Cyclone Global Navigation Satellite System (CYGNSS), and supporting the Rosetta, IBEX, Juno, Solar Probe Plus and Europa (NASA) missions. He has a BSEE from the University of Texas at Austin, and MSEE from St. Mary's University in San Antonio, TX.*

Alan Henry *is a Staff Engineer in the Space Science and Engineering Division at Southwest Research Institute. He has more than 25 years of experience in spacecraft design, development, integration, operations, and management. He is currently the AI&T lead for CYGNSS. He served as the I&T Manager for the MMS Instrument Suite for four years, and successfully delivered the first Instrument Suite to NASA prior to transitioning to CYGNSS. At Orbital Sciences Corp, he supported the DAWN Interplanetary Spacecraft program currently in orbit around Ceres. He has a Bachelors Degree in Aerospace Engineering from the University of Texas.*

Robert Klar *is a Principal Engineer in the High-Reliability Systems Section at Southwest Research Institute. He has over 18 years of experience in the areas of software engineering, real-time operating systems, embedded systems, signal processing, computer networking, and communications. Mr. Klar was the technical lead and developer of avionics and instrument flight software for the Imager for Magnetopause to Aurora Global Exploration (IMAGE), Swift Gamma Ray Burst Explorer, Fermi, Orbital Express, Wide-field Infrared Survey Explorer (WISE), Kepler, and Magnetospheric Multiscale (MMS) programs. He is currently developing flight software for the CYGNSS mission. He has a Bachelors Degree in Computer Engineering from Texas A&M University and a Masters Degree in Electrical Engineering from St. Mary's University.*

Scott Miller *is a Senior Research Engineer in the High Reliability Systems Section at Southwest Research Institute. He has nine years of experience in the areas of embedded systems, data analysis, and computer networking. He has worked as a developer and System Engineer for A-10 military jet avionics and for the Instrument Suite (IS) for the NASA Magnetospheric Multiscale (MMS) Mission. He has a B.S. in Computer Engineering from Texas A&M University. He currently serves as the technical lead and project manager for the CYGNSS flight software. He has a Bachelors Degree in Computer Engineering from Texas A&M University.*