University of Colorado
Department of Aerospace Engineering Sciences
ASEN 4018
Project Definition Document

Lockheed Martin Fault Testbed
Sunday 1st October, 2017

Approvals

<table>
<thead>
<tr>
<th>Customer</th>
<th>Name</th>
<th>Affiliation</th>
<th>Approved</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>Dr. Jim Chapel</td>
<td>LMSS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Course Coordinator</td>
<td>Dr. Dale Lawrence</td>
<td>CU/ AES</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Project Customers

Name: Dr. Jim Chapel
Email: jim.d.chapel@lmco.com
Phone: 303-977-9462

Team Members

<table>
<thead>
<tr>
<th>Name: Dalton Anderson</th>
<th>Email: <a href="mailto:dalton.anderson@colorado.edu">dalton.anderson@colorado.edu</a></th>
<th>Phone: 720-438-5449</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name: Sam O’Donnell</td>
<td>Email: <a href="mailto:samuel.odonnell@colorado.edu">samuel.odonnell@colorado.edu</a></td>
<td>Phone: 858-699-5201</td>
</tr>
<tr>
<td>Name: Dan Greer</td>
<td>Email: <a href="mailto:daniel.greer-l@colorado.edu">daniel.greer-l@colorado.edu</a></td>
<td>Phone: 720-320-6505</td>
</tr>
<tr>
<td>Name: Zach Reynolds</td>
<td>Email: <a href="mailto:zare0851@colorado.edu">zare0851@colorado.edu</a></td>
<td>Phone: 303-263-0231</td>
</tr>
<tr>
<td>Name: Ben Hutchinson</td>
<td>Email: <a href="mailto:behu1036@colorado.edu">behu1036@colorado.edu</a></td>
<td>Phone: 308-250-0077</td>
</tr>
<tr>
<td>Name: Kristyn Sample</td>
<td>Email: <a href="mailto:kristyn.sample@colorado.edu">kristyn.sample@colorado.edu</a></td>
<td>Phone: 573-819-9956</td>
</tr>
<tr>
<td>Name: Kent Lee</td>
<td>Email: <a href="mailto:Kent.lee@colorado.edu">Kent.lee@colorado.edu</a></td>
<td>Phone: 502-523-6219</td>
</tr>
<tr>
<td>Name: Corwin Sheahan</td>
<td>Email: <a href="mailto:corwin.sheahan@colorado.edu">corwin.sheahan@colorado.edu</a></td>
<td>Phone: 661-912-2204</td>
</tr>
<tr>
<td>Name: Andrew Levandoski</td>
<td>Email: <a href="mailto:Andrew.Levandoski@colorado.edu">Andrew.Levandoski@colorado.edu</a></td>
<td>Phone: 303-885-8519</td>
</tr>
<tr>
<td>Name: Pol Sieira</td>
<td>Email: <a href="mailto:Pol.Sieira@colorado.edu">Pol.Sieira@colorado.edu</a></td>
<td>Phone: 720-210-3314</td>
</tr>
<tr>
<td>Name: Andrew Mezich</td>
<td>Email: <a href="mailto:Andrew.Mezich@colorado.edu">Andrew.Mezich@colorado.edu</a></td>
<td>Phone: 907-903-4141</td>
</tr>
<tr>
<td>Name: Zack Toelkes</td>
<td>Email: <a href="mailto:zato0770@colorado.edu">zato0770@colorado.edu</a></td>
<td>Phone: 785-393-3762</td>
</tr>
</tbody>
</table>
1. Problem Statement

The motivation of this project is grounded in the identification and mitigation of operational faults, more specifically, those which occur in the avionics of satellites. Numerous satellites have fallen victim to faulty avionics systems since the beginning of the space age. The ability to detect, analyze, and respond to faults in a way that maintains the satellite’s operational integrity is becoming an increasingly desirable technology. This technology, however, is quite expensive and requires a long developmental process to create. Additionally, the process of testing a satellite’s avionics for fault responses is very much satellite specific. This means that once a significant amount of resources have been expended developing a sound process for testing the avionics of one satellite, that process will likely not be applicable to other satellites. It is therefore highly desirable to develop modular testing environments for fault management. This project will aim to develop a modular fault management test bed.

The fault management test bed will comprise of two physical elements: a platform (TestTable) that provides an environment with limited restrictions on the planar dynamics and motion of a body placed on the TestTable, and a mock satellite (MockSat) which will serve as the active component of the test bed. The MockSat will be equipped with an attitude determination and control system (ADCS) that approximately replicates the 0.04 Hz bandwidth response of the GOES-16 satellite. The GOES-16 satellite is a geostationary satellite developed by Lockheed Martin and serves as the next generation of weather forecasting. The MockSat will allow for the injection of a predefined fault in order to evaluate the fault response of the flight software. The resulting flight data of the MockSat will be retained for future analysis.

The ADCS in modern space vehicles are designed to be single fault tolerant, which is characterized by the following:

1. Zero single points of failure (SPOF). No single failure in any one component should be so catastrophic as to disable the system.

2. Identification of the failed component. The fault should be classified to the greatest extent possible in order to optimize the fault response strategy.

3. Isolation/containment of the failure. In order to protect against fault propagation, the faulty component should be isolated in the most comprehensive manner possible.

4. The ability to revert to a safe mode. No failure should initiate a cascade failure that results in the system being disabled.

Since single fault tolerant flight software is complex and difficult to test, a simple, modular system is desirable for rapid testing of flight software configurations. By separating the ADCS software from fault management software, fault testing becomes increasingly modular, allowing for simpler and less costly design.

The GOES-16 satellite, being 3-axis stabilized, uses reaction wheels to control its attitude, therefore the MockSat will utilize reaction wheels to control attitude about one axis. This project will demonstrate single fault tolerance with the fatal operating fault of a reaction wheel. Within the scope of this project, a fatal operating fault is one which hinders the performance of a component to the point where the component no longer operates in a nominal fashion, preventing it from performing its specific task. Once implemented, the MockSat will detect the fatal operating fault and respond, entering a safe-mode and alerting the ground station operator. Once the operator has been alerted, the user can command the MockSat to resume operational capabilities by transferring attitude control responsibilities to the redundant reaction wheel. Additionally, a fatal operating fault will be injected into the attitude determination sensors, responding by again entering a safe-mode and allowing the ground station operator to determine how to resume normal satellite operation. Beyond demonstration of fault management, a fault management flowchart outlining possible failures and their corresponding responses will be created to understand possible tests with the MockSat and TestTable.

The purpose of the TestTable is to provide an environment with limited restrictions on the planar motion of the MockSat. The MockSat will not have the ability to actively control translation, therefore the TestTable will serve as not only a platform on which the MockSat will operate, but it will also restrict the MockSat motion to three degrees of freedom (DOF). The TestTable will allow for two degrees of freedom in translation, passively limited to relatively small portion of the TestTable, and one in rotation. The MockSat will have, as a baseline, redundant reaction wheels. Both the TestTable and the MockSat provide the infrastructure for testing fault management. The essence of the project lies with the ability to test a series of different faults with hardware developed for the TestTable and MockSat.
2. Previous Work

2.1. Existing Low-Friction, 3-DOF Test Beds and Sims

To test the guidance, navigation, and control systems for satellites, it is useful to do so in an environment indicative of space. This environment can be modeled simply in two dimensions when isolating planar motion. There are a few options for achieving this simulated environment that have been employed by other engineering projects. The European Space Agency has used air bearings and an extremely smooth epoxy floor. Alternatively, a prior CU Aerospace senior design team, DRAFTSat, used an air hockey table like design to achieve this environment. These are some of the options that will be considered in future trade studies.

2.2. Fault Management

Fault management (FM) is a critical feature to any space craft system. As such, there are thousands of examples of FM systems in use today, and they are too many to enumerate here. Additionally, FM is often particular to the system that is being designed, however, NASA has compiled a Fault Management Handbook with the purpose of standardizing FM practices. This handbook aggregates knowledge from numerous NASA missions and provides a standardized best practice approach to FM.

The NASA FM Handbook provides useful examples of how to and how not to approach FM. One example given is that the CALIPSO satellite’s PROTEUS propulsion bus had ambiguous fault tolerance requirements. It was unclear which documents provided the requirements, and some documents even contradicted themselves. Another example cited was the Mars Observer program, in which the fault management was assumed to have been accomplished by low-level hardware redundancy. Both of these projects suffered from the lack of a top-down systems engineering approach. This project will provide an avenue to explore different paradigms of fault management from the system level.

2.3. GOES-16

Many satellites in development or use today incorporate a common design aspect, single fault tolerance. These systems use redundant sensors and actuators to avoid mission complications stemming from a single fault. Implementing exhaustive fault management systems is monetarily expensive and complex as each satellite requires its own customized approach to fault management.

This project will reference the GOES-16 satellite for developmental purposes regarding the ADCS system. The software and hardware chosen for this project will include measures to replicate the bandwidth of GOES-16. The GOES-16 satellites is 3-axis stabilized, using gyros, reaction wheels, and star-trackers to achieve this stabilization. The MockSat will model the functionality of the GOES-16 ADCS with a single plane of attitude control.

3. Specific Objectives

Table 1 breaks down this project’s success into level 1, 2, and 3 objectives; level 1 indicates the minimum definition of success, and level 3 is the primary success metric for each major subsystem. This project will have two primary mechanical components: (1) TestTable, (2) MockSat. The TestTable will allow 3DOF for the MockSat (2 translational and 1 rotational). The MockSat translation will be passively controlled, while its rotation will be actively controlled. The TestTable/MockSat system must be low enough friction to provide practical freedom of movement in these directions to simulate zero-g, planar movement. The TestTable will have a mobile attitude reference to verify MockSat attitude pointing both before and after fault injection/recovery.

The TestTable must be transportable between labs and buildings withing the University of Colorado Engineering Center. As the standard exterior door is 80 inches by 36 inches, the dimensions of the TestTable will not exceed 72 x 72 x 28 inches. According to OSHA guidelines, an individual lifting weights greater than 50 pounds increases the risk of injury. Therefore, the TestTable will weigh less than 100 pounds and be transportable by two people.
Table 1: Level 1, 2, and 3 Objectives

<table>
<thead>
<tr>
<th>TestTable</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Construct a TestTable to allow for 2D translation dynamics with passive control, 1D rotation dynamics, support weight of MockSat, stationary attitude reference</td>
<td>30 minute constant operating time</td>
<td>Moving attitude reference</td>
</tr>
<tr>
<td>MockSat Hardware</td>
<td>Power source, position sensor, coarse orientation sensor (analogous to sun sensor), fine orientation sensor (analogous to star tracker), redundant reaction wheels (1 set of 2), ADCS/fault injection processor, data storage, 15 minute constant operating time</td>
<td>60 minute constant operating time</td>
<td></td>
</tr>
<tr>
<td>Fault Injection</td>
<td>Inject fatal operating fault into primary reaction wheel after a predetermined time from testing start</td>
<td>Inject fatal operating fault into coarse orientation sensor</td>
<td>Inject fatal operating fault into fine orientation sensor</td>
</tr>
<tr>
<td>Fault Management</td>
<td>Upon fault injection, the satellite will recognize the presence of the fault and enter a safe mode</td>
<td>Upon user command, satellite responds to injected fault in a way that maintains operational integrity</td>
<td></td>
</tr>
<tr>
<td>MockSat Control</td>
<td>Active planar rotational control with passive translational control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comm/Data Handling</td>
<td>Flight software uploaded prior to testing; fault uploaded prior to testing; telemetry data stored onboard MockSat; Ground Station data analysis post-test</td>
<td>Wired, real-time telemetry and fault injection</td>
<td>Wireless, real-time telemetry and fault injection</td>
</tr>
</tbody>
</table>

The MockSat will have redundant reaction wheels (primary and secondary) for attitude actuation, at a minimum. The reaction wheels will provide an attitude control bandwidth response of ± 10% of 0.04 Hz, which mimics the pointing response of the GOES-16 satellite. For attitude determination, there will be two sensors: one coarse and one fine. These are analogs to sun sensors and star trackers, respectively. A coarse sensor will have a wider angle of view, but less resolution of attitude determination, and the fine sensor will have a more narrow field of view and will provide a higher degree of accuracy. The coarse/fine sensor array will allow the system to replicate the pointing precision of a star tracker, without the need to include a catalog of stars. As part of a successful demonstration, the MockSat will undergo at least two attitude verification tests before fault injection, then will go through at least two attitude verification tests after fault recovery. The attitude determination tests have not been designed specifically, however, they are expected to be on the order of 15 minutes to an hour. As such, it is desired that the MockSat be able to run for up to 60 minutes autonomously as a Level 3 objective. The MockSat will also need an onboard power source to power the flight control and fault injection systems for this time frame.

The MockSat will also have an onboard processor for flight software and fault injection control. The MockSat control software will autonomously maintain a specified orientation to within ± of 2.5° of a desired pointing angle, and be able to autonomously adjust orientation with an attitude control bandwidth response of ± 10% of 0.04 Hz. The attitude control system will be single-fault tolerant, meaning that the MockSat flight software will be able to respond to a fatal operating fault in a reaction wheel, the coarse attitude sensor, or the fine attitude sensor (but not more than one fault happening concurrently).

Lastly, the MockSat will have a separate control system onboard to create faults. This need not lie on a separate processor as long as it is able to inject a fault without alerting the flight control software that a fault is occurring. This system will be able to cause a fatal fault in the primary reaction wheel on the MockSat, as well as fault the coarse and fine attitude sensors. This system will be completely separate from the MockSat flight control system, and will not interfere with the attitude and location control of the MockSat (other than by causing fatal operating faults). As a
The bare minimum, the fault injection system will cause a fatal operating fault in the primary reaction wheel after a pre-programmed amount of time. This fault needs only to prevent the proper operation of the reaction wheel. Therefore, in order to minimize component cost, it is desired to have the fault not permanently damage the primary reaction wheel and ADCS system. As a Level 2 objective, the fault will be able to be triggered by a user on demand, rather than after a predetermined time frame. Along with triggering the fault on demand, it will be able to inject a fault into either the reaction wheel or coarse orientation sensor. Finally, as a Level 3 objective, the fault injection system will have the capability of triggering fatal faults in the fine orientation sensor.

4. Functional Requirements

1. The TestTable shall allow for two degrees of freedom in translation and one degree of freedom in rotation.

2. The MockSat shall be equipped with an attitude determination and control system (ADCS) that replicates the 0.04 Hz bandwidth response of the GOES-16 satellite to within ±10%.

3. The MockSat shall have the ability to maintain a controlled attitude relative to a point of reference within ±2.5°.

4. The system shall have the ability to introduce a fatal operating fault in either the MockSat’s primary reaction wheel, the coarse orientation sensor, or the fine orientation sensor (but not more than one fault at a time).

5. The MockSat flight control software shall recover from a fatal operating fault in either the MockSat’s primary reaction wheel, the coarse orientation sensor, or the fine orientation sensor (but not more than one fault at a time) by entering a safe mode.

The major functional subsystems of the functional block diagram (FBD) are denoted by the rounded rectangles. The fundamental components of each subsystem are highlighted in the rectangular boxes. The red shading indicates the components that will have to be designed or obtained through the design process. Some components may possibly be obtained through available resources at no cost to the team, but there are not any components currently decided on.
The main purpose of the TestTable is to provide a “frictionless” environment for the MockSat to operate in 3DOF. A reference source, monitored by rough and fine precision sensors incorporated in the MockSat’s sensor array, will ultimately provide a point of reference for the MockSat’s ADCS system. This reference source will provide the MockSat the ability to demonstrate its pointing and control capability during testing, both before and after fault introduction.

The final piece of this system is the MockSat, which will integrate several elements. First, the MockSat will have the ability to orient itself relative to an external point of reference. Once attitude determination and control has been demonstrated, a fault will be injected into either the reaction wheel, coarse sensor, or the fine sensor. The fault injection system will initiate this fault, which the fault management software will recognize and respond to. The response will either force the MockSat to enter a safe mode or resolve the fault and re-enter operational mode. The command to re-enter operational mode will be commanded by the ground station at the request of the ground station operator. The flight data will be recorded as well, and will likely require external memory capability, which will be addressed through additional research.

The concept of operations, shown in Figure 2, depicts the MockSat tracking some reference target on a low-friction TestTable. The MockSat initially awaits upload of flight software until it begins operation. The MockSat communicates wirelessly with ground station for downlinking its telemetry. When the MockSat demonstrates proper pointing (to above pointing accuracy), the user injects a specific fault into a component of the ADCS. The MockSat detects this fault and responds by going into a safe mode and alerting the GSU. A command is then given by the user to resume operations, and the MockSat returns to nominal operating conditions.

![Figure 2: Concept of Operations](image-url)
5. **Critical Project Elements**

5.1. **TestTable**

One of the critical project elements will be the TestTable, which needs to allow for semi-unperturbed near frictionless movement with 3DOF. In order to truly isolate the dynamics of the MockSat, the TestTable must be large enough to allow for this freedom of motion without interference from the TestTable itself. The TestTable must also allow for passive transnational control of the MockSat.

Another aspect, which may prove to be difficult, will be limiting any attitude change on the pitch and roll axes. Any method that allows translation in a third dimension, however slight, will constitute a source of error to the flight software. The team will strive to develop a method with confidence that this error can be disregarded.

5.2. **Mock Satellite**

In order to implement attitude control, the MockSat will need to determine and control its attitude through the use of sensors and actuators. These sensors and actuators will need an onboard power supply so as not to interfere with the frictionless offloading, which will introduce limitations to factors such as weight and test run time. In order to interface each of these components, experience with PCB design will be necessary, as well as a strong understanding of electronics and power systems. In addition, depending upon the sensors and actuators chosen, these subsystems could become quite expensive. A feasibility analysis of different solutions will need to be conducted in order to ensure that these subsystems fit within the design constraints.

5.3. **ADCS Software**

The ADCS software will be written to control motion in a single plane and will require in-depth knowledge of dynamics and P(I)D controllers. Additionally, depending on the method for onboard processing, knowledge of embedded systems may be necessary. This presents a technical challenge, although many resources at the university are available to the team to overcome this challenge.

5.4. **Fault Injection System**

Developing a way to inject faults into the MockSat while it is operating is a key challenge of this project. The fault injection system will be independent from the flight control system. A trade study must be conducted to determine if both systems can either be on the same processor or on their own dedicated processors. Effective means of initiating these faults will also need to be determined. Designing and implementing the fault injection system in an intelligent and forward thinking way will be a key element to this project.

5.5. **Fault Management Software**

One of the most critical elements of the project will be the software aboard the MockSat that detects and manages faults based on a fault management algorithm. This software will work with the ADCS software to detect and mitigate faults as they occur. Due to the TBD fault management algorithm and response mechanisms, this software will be technically and logistically difficult until the specific faults and allowed types of responses are solidified. For this purpose, specifically one fault will be demonstrated, while others can simply be researched. Even then, it may be very difficult, from a programming perspective, to manage the fault that the MockSat will experience, independently from the flight software. Although the project only requires a demonstration of managing one fault, thought must be given toward the ability for the MockSat to demonstrate management of other faults.

5.6. **Communications and Data Handling**

A critical element of completing this project will be the communications and data handling systems. This system will allow the MockSat to either store flight data onboard, to be read at a later time, or communicate this data to the ground
station during the test. The data will be either saved on some removable memory disk, or transmitted to the ground station through a wired or wireless connection. This system is critical because it allows the flight data of the MockSat to be analyzed, which can help improve ADCS software. Logistically, this system will largely depend on the type and numbers of sensors chosen, as this will affect the amount of data that needs to be saved or transmitted.

6. Team Skills and Interests

<table>
<thead>
<tr>
<th>Critical Project Elements</th>
<th>Team Members and Associated Skills/Interests</th>
</tr>
</thead>
</table>
Control Systems/Algorithms, Analytic/Numerical Optimization,  
Applied Math 2nd Major, High Fidelity Simulations |
| TestTable, MockSat, Fault Injection System | Dan Greer (Manufacturing Lead)  
Aerospace department machine shop employee with two years of experience in production life cycle, including but not limited to design synthesis, fabrication, and test/verification. |
| ADCS Software, MockSat, Fault Injection System | Ben Hutchinson  
Team Leader for Northrop Grumman Internship (8 Members), Software Development, Controls Work, Systems Writing and Presentations |
| MockSat Avionics & Power Supply (Sensors, Actuators, Telemetry) | Kent Lee (Electronics Lead)  
Two years electronics shop experience (PCB design, DAQ software, embedded systems)  
One year EPS lead for COSGC’s Polarcube (Developed solar system for 3U cubesat) |
| TestTable (Design and Manufacturing), MockSat, Finance | Andrew Levadoski (Safety Lead)  
6 years Range Safety Officer, 4 years Ordinance handling safety, 3 years tactical team training and safety po, Two years as metal parts fabricator using cad software and machine shop tools, CNC certified |
| TestTable, MockSat Harware, MockSat Dynamics | Andrew Mezich  
One year experience as Structures Lead for Colorado Space Grant Consortium Polar Cube cubesate project. Orbital ATK Intern - Structural Analysis.  
Skills: Solidworks, ANSYS, CAM. |
| MockSat Flight Hardware/Software | Sam O’Donnell (Systems Lead)  
3 years experience in construction project management.  
Experienced with Microsoft Projects.  
Systems Engineering Lead for Gateway to Space balloon satellite RadSat3000 |
| Comm./Software | Zach Reynolds  
Coursework/interest in electronics, communications, programming, control systems |
| Project Manager MockSat Dynamics | Kristyn Sample  
Leadership skills learned and practiced through 6 years military experience.  
Program management experience through starting and developing new student organization at CU. Interest in system dynamics and controls. |
| ADCS Software, Fault Injection System, Fault Management Software | Corwin Sheahan  
Three summers of internship experience working within both the software and aerospace industries on software solutions to engineering problems |
| Testing, ADCS Software, Fault Management Software | Pol Sieira  
Experience with testing different projects with student group CUSEDS, process oriented from military experience, consistently took software lead in previous ASEN labs |
| TestTable, MockSat, Avionics, Testing | Zack Toelkes  
One year experience designing and writing software for Telerobotic rover at CASA.  
Experience designing, developing, and implementing avionics payload for Colorado Space Grant Consortium Rocket-Sat project. Experienced in electronics, mechanical, sensors |

7. Resources

This design project requires emulating certain components of the GOES-16 satellite, as outlined above. Therefore, technical data on GOES-16 bandwidth, response times, inertia, sample rate of controller, and more will be gleaned from reports provided by the customer.

New attitude control spin-modules are being developed by the university’s aerospace electronics shop to support the undergraduate curriculum course “ASEN 3200 Attitude Determination & Control.” These wireless modules allow for
a single rotational degree of freedom. The data acquisition software has already been developed, and uses National Instrument’s LabVIEW platform with a myRIO for software deployment. Trudy Schwartz, the director of the electronics shop, has offered the existing architecture for these spin modules as a basis for further modifications to achieve our design goals.

Faculty, as well as historical projects, will be resources for developing the flight software that is to be incorporated in the MockSat. Since the team does not necessarily have experience or expertise in this matter, this is something that will need to be addressed.

Finally, a CU Boulder senior project from 2013 named TRACSat used a medium-sized (3 x 2 ft) air-cushion table to demonstrate autonomous translation and attitude control using only thrusters and electro-optical sensors. This project will likely be similar in size magnitude and shape, and will have similar functionality should it be decided to use an air-cushion table to simulate a zero-g 3DOF environment. This table, including cart and accompanying air compressor, will be available for our use as the team sees fit, again, if it is decided to pursue an air-cushion solution for modeling zero-g offloading and this fits logistic requirements such as weight and size.

![TRACSat air-cushion table](image)

**Figure 3: TRACSat air-cushion table**

**References**


[3] ESA, “2d zero-g testing for space robotics.” [http://www.esa.int/Our_Activities/Space_Engineering_Technology/2D_zero-g_testing_for_space_robotics](http://www.esa.int/Our_Activities/Space_Engineering_Technology/2D_zero-g_testing_for_space_robotics), Dec 2015.

