AAReST: Autonomous Assembly of a Reconfigurable Space Telescope

Ae 105 Final Presentations 2018

Credit: Antonio Pedivellano
Mission Overview

Fabien Royer
Building Large Space Telescopes

• Mirror dia. of current and planned space telescopes limited by constraints of a single launch
  – Hubble (1990): Ø 2.4 m
  – JWST (2018): Ø 6.5 m
  – HDST (2030+): Ø 11.7 m

• New paradigms needed for Ø 30 m+ segmented primary:
  – Autonomous assembly in orbit
  – Active ultralight mirror segments

• Active mirrors relax tolerances for assembly and manufacturing, correct thermal distortions

• Modular, robust, low-cost architecture
Building Large Space Telescopes

• Mirror dia. of current and planned space telescopes limited by constraints of a single launch
  – Hubble (1990): Ø 2.4 m
  – JWST (2018): Ø 6.5 m
  – HDST (2030+): Ø 11.7 m

• New paradigms needed for Ø 30 m+ segmented primary:
  – Autonomous assembly in orbit
  – Active ultralight mirror segments

• Active mirrors relax tolerances for assembly and manufacturing, correct thermal distortions

• Modular, robust, low-cost architecture
AAReST Mission Objectives

• Demonstrate key technologies:
  – Autonomous assembly and reconfiguration of modular spacecraft carrying mirror segments
  – Active, lightweight deformable mirrors operating as segments in a primary

• Operate for as long as necessary to accomplish the objectives (~90 days)

• Gather engineering data to enable development of the next system
AAReST Spacecraft
AAReST Spacecraft

CoreSat
Power, Comm., Telescope ADCS
Caltech
AAReST Spacecraft

CoreSat
Power, Comm., Telescope ADCS
Caltech

Reference Mirrors (× 2)
Fixed figure mirror segments
Caltech
AAReST Spacecraft

- **MirrorSat (×1)**
  - Reconfigurable free-flyers
  - *U. of Surrey*

- **CoreSat**
  - Power, Comm., Telescope ADCS
  - *Caltech*

- **Reference Mirrors (×2)**
  - Fixed figure mirror segments
  - *Caltech*

- **MirrorSat (×1)**
  - Reconfigurable free-flyers
  - *IIST*
AAReST Spacecraft

CoreSat
- Power, Comm., Telescope ADCS
- Caltech

Reference Mirrors (× 2)
- Fixed figure mirror segments
- Caltech

Deformable Mirrors (× 2)
- Active mirror segments
- Caltech

MirrorSat (× 1)
- Reconfigurable free-flyers
- U. of Surrey

MirrorSat (× 1)
- Reconfigurable free-flyers
- IIST
AAReST Spacecraft

- **MirrorSat (×1)**
  - Reconfigurable free-flyers
  - *U. of Surrey*

- **CoreSat**
  - Power, Comm., Telescope ADCS
  - *Caltech*

- **Reference Mirrors (×2)**
  - Fixed figure mirror segments
  - *Caltech*

- **Deployable Boom**
  - Composite structure provides 1.2 m focal length
  - *Caltech*

- **Deformable Mirrors (×2)**
  - Active mirror segments
  - *Caltech*

- **MirrorSat (×1)**
  - Reconfigurable free-flyers
  - *IIST*
Overview – Spacecraft – Mission Requirements – CONOPS – History – CoreSat – Outline

AARest Spacecraft

- **MirrorSat (×1)**
  - Reconfigurable free-flyers
  - U. of Surrey

- **CoreSat**
  - Power, Comm., Telescope ADCS
  - Caltech

- **Reference Mirrors (×2)**
  - Fixed figure mirror segments
  - Caltech

- **Deformable Mirrors (×2)**
  - Active mirror segments
  - Caltech

- **Deployable Boom**
  - Composite structure provides 1.2 m focal length
  - Caltech

- **MirrorSat (×1)**
  - Reconfigurable free-flyers
  - IIST

- **Camera**
  - Imaging, Wavefront Sensing and Control
  - Caltech
Mission Requirements

• Minimum mission
  1. Produce one focused image from a deformable mirror
     • 80% encircled energy radius from point source < 25 µm
  2. Perform at least one in-flight autonomous spacecraft reconfiguration maneuver to demonstrate space assembly capability

• Extended mission
  1. Produce one focused image from a deformable mirror after reconfiguration
  2. Coalign images to improve SNR and demonstrate precursor to co-phasing
  3. Produce at least two images of other sources (e.g. Earth and Moon) for outreach purposes
## Concept of Operations


**Launch**

- **Launch in a compact, stowed volume**
  - \( 46 \text{ cm} \times 34 \text{ cm} \times 30 \text{ cm} \)
## Concept of Operations

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>2.</td>
<td>3.</td>
<td>4.</td>
<td>5.</td>
<td>6.</td>
</tr>
<tr>
<td>Launch</td>
<td><strong>Telescope Deployment</strong></td>
<td>Telescope Calibration &amp; Imaging</td>
<td>Reconfiguration</td>
<td>Telescope Recalibration &amp; Imaging</td>
<td><strong>Extended Mission</strong></td>
</tr>
</tbody>
</table>

- Satellite health check, detumble, antenna deployment
- Deploy boom in two stages
- Uncage deformable mirrors
## Concept of Operations

|----------------------------|-------------------------|-----------------------------------|--------------------|--------------------------------------|---------------------|

- **Telescope points to a bright reference star**
- **Calibrate:**
  - Segment tip/tilt/piston
  - Deformable mirror surface figure
- **Camera provides feedback for segment calibration**
## Concept of Operations

<table>
<thead>
<tr>
<th>1.</th>
<th>2.</th>
<th>3.</th>
<th>4.</th>
<th>5.</th>
<th>6.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Launch</td>
<td>Telescope Deployment</td>
<td>Telescope Calibration &amp; Imaging</td>
<td>Reconfiguration</td>
<td>Telescope Recalibration &amp; Imaging</td>
<td>Extended Mission</td>
</tr>
</tbody>
</table>

- MirrorSats release from CoreSat using electromagnets
- Fly out ~10 – 30 cm
- Re-dock into “wide” configuration
# Concept of Operations

|-----------|--------------------------|-----------------------------------|------------------|-------------------------------------|---------------------|

- Extended Mission

![Telescope Image]
## Concept of Operations

|-----------|--------------------------|-----------------------------------|-----------------|-----------------------------------|------------------|

- Co-align star images from different segments to improve SNR
  - Pre-cursor to co-phasing
- Produce images of extended sources (e.g. Moon, Earth) for outreach
- Perform orbital maneuvers using thrusters
AARReST History

2008 November: Large Space Apertures KISS workshop
2010 June: Ae105
   - Initial mission design
2012 September: Mission Concept Review
2013 September: Preliminary Design Review
2014 September: Detailed Design Review
2015 September: Complete Design Review
2016 June: Ae105
   - Telescope systems
2017 January: Telescope Complete Design Review
2017 June: Ae105
   - Preliminary design of CoreSat
2017 September: Complete Design Review of three satellites
2018 June: Here we are!
   - CoreSat hardware assembly and testing
CoreSat Overview

LEFT STACK
- Electromagnet Board UL
- Payload Interface Computer
- Payload Switch Board
- MirrorSat Frangibolt
- Electromagnet Board LL
- Interface Board
- Z Torque Rods
- Radio
- Antenna Board

CENTRAL STACK
- Star Camera
- Camera Frangibolt
- Interface Boards
- ADCS
- EPS board
- OBC
- Reaction Wheels

RIGHT STACK
- Electromagnet Board UR
- Payload Interface Computer
- Payload Switch Board
- MirrorSat Frangibolt
- Electromagnet Board LR
- Interface Board
- Z Torque Rods
- Battery
- Antenna Board
Acknowledgement

The AAReST project is supported by:
• The Keck Institute for Space Studies
• Division of Engineering and Applied Science
• Provost’s Innovation in Education Fund
• JPL (providing mentoring)
Outline

2:00 pm: Introduction & Mission Overview
2:15 pm: Structures
2:40 pm: Mechanisms
3:55 pm: ADCS
3:20 pm: Thermal
4:35 pm: Onboard Software
4:00 pm: Telecom
4:25 pm: Avionics
5:00 pm: Reception
Structures Team

Yunsang Choi
Kanthasamy Ubamanyu
Haojie Zhuang

Mentor: Antonio Pedivellano
Overview and Objectives

Goal: deliver a launch-qualified flight structure

– Vibration test to ensure the structural soundness of the CoreSat, MirrorSats
  • Design representative mass model
  • Manufacture and assemble mass model
– Study the structural dynamics of the spacecraft through FEM
AAReST Mass Model

CoreSat (Caltech)

MirrorBoxes (Caltech)

MirrorSat (IIST)

MirrorSat (Surrey)

Camera (Caltech)
Task breakdown

• Yunsang and Haojie
  – Manufacture AAReST Model for vibration testing
  – Integrate subsystems and perform fit checks
  – Document procedures and improvements

• Uba
  – Design of interface plate for vibration test
  – Finite element simulations
Task Changes

• Fit test and assembly time frame extended
  – Numerous errors in outsourced components
  – Components modified and fixed in house

• Vibration test postponed due to delay in arrival of Mirrorsats from IIST and Surrey
Current Status

IIST MirrorSat
- Received

Camera
- Manufacturing: 100%
- Assembly: 100%
- Integration: 100%

Surrey MirrorSat
- Not yet shipped

MirrorBoxes
- Manufacturing: 100%
- Assembly: 100%
- Integration: 50%

Boom
- Assembly: 100%
- Integration: 100%

CoreSat
- Manufacturing: 100%
- Assembly: 85%
CoreSat Mass Model Design

Flight CAD

Representative Mass Model
CoreSat Internal Stacks

Example of internal representative mass stacks

Threaded rods and frame
Manufacturing Errors

Remanufactured +X Face Plate

Misaligned Holes
MirrorBox Mass Model

Overview – Manufacturing and assembly – FEM - Conclusion and Future Work

Flight CAD

Mass Model Design

Assembly

Integration
Camera Mass Model

- Manufacturing delayed by waterjet breakdown
- Switched to bandsaw manufacturing
- Waiting on Frangibolt interface and mount
## Mass Comparison

<table>
<thead>
<tr>
<th></th>
<th>CAD Mass (g)</th>
<th>Measured Mass (g)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center Stack</td>
<td>1345</td>
<td>1520</td>
</tr>
<tr>
<td>Right Stack</td>
<td>1377</td>
<td>1401</td>
</tr>
<tr>
<td>Left Stack</td>
<td>866</td>
<td>932</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>3588</strong></td>
<td><strong>3853</strong></td>
</tr>
</tbody>
</table>

Mass difference: 265 g (6.7% deviation)

Deviations:
- 80 gram resolution for masses (up to 40 grams ~1% deviation)
- The CAD model lacks fasteners and cable harnessing
MirrorSats

• Representative model required for vibration testing
  – MirrorSats attached to CoreSat via Frangibolt (provided by Caltech)
  – Electromagnet provided by Surrey

• IIST MirrorSat received

• Surrey MirrorSat not yet shipped
Integrated Mass Model

Deployable boom

IIST MirrorSat

Camera

CoreSat

Deformable MirrorBox
Vibration Table Interface

- Mount AAReST to the vibration table
- Requirements
  - Lowest natural frequency > 2000 Hz
  - Match actual boundary conditions
  - 3-axis vibration test

Figure 1: Example of Vibration table [1]

Figure 2: Flight LV interface

Figure 3: a) Proposed interface; b) Interface with AAReST

Design of Interface Plate

• Initial estimation for thickness \(^{[2]}\)
  
  – Rectangular plate simply supported along all edges
    \[
    \omega^2 = \frac{\pi^4 D}{a^4 \rho \left\{ \left( \frac{a}{b} \right)^4 + 2 \left( \frac{a}{b} \right)^2 \right\}}; \quad \text{where, } D = \frac{E h^3}{12(1 - \nu^2)}
    \]
    
    38.1mm (1 ½”) thick rectangular plate of 350mm x 300mm

• 3-axis vibration test
  
  – vibration table often allows only lateral vibration
  
  – 45° angled mount

Numerical Simulation

Motivation

• Efficient tool for evaluating design changes and sensitivity studies
• Easy to obtain stress and strain distributions
• Required by launch provider
  – Modal frequency analysis
  – Random environment analysis
  – Shock analysis

Objective

• Study the structural response of the CoreSat
  – Develop a “representative” FE model
  – Fidelity of FE model correlated against vibration test results
Finite Element Model

Overview – Manufacturing and assembly – **FEM** - Conclusion and Future Work
Finite Element Model

• Simplified geometry of the model
  – Single part for entire external structure
  – Sub systems included as point masses

• Connection type
  – Tie constraint - no relative motion between them
  – Coupling constraint - allowing control over the transmission of forces

• Element type
  – C3D10 - 10 node quadratic tetrahedron

• Boundary condition
  – Clamped at Launch vehicle interface
Modal Analysis

PSLV Stiffness requirements
- For base-fixed conditions
- Lateral > 45 Hz
- Longitudinal > 90 Hz

Mode 1
Lateral Mode (X) 65.1 Hz

Mode 4
Lateral Mode (Y) 109.4 Hz

Mode 5
Longitudinal Mode (Z) 111.4 Hz
Modal Analysis

PSLV Stiffness requirements
- For base-fixed conditions
- Lateral > 45 Hz
- Longitudinal > 90 Hz

Requirements satisfied!
with a margin of 22% for longitudinal and 44% for lateral

Mode 1
Lateral Mode (X)
65.1 Hz

Mode 4
Lateral Mode (Y)
109.4 Hz

Mode 5
Longitudinal Mode (Z)
111.4 Hz
Random Response Analysis

- Random environment load analysis
  - Load analysis using extracted eigenmodes for base-driven random vibration

- Extracted range of modes sufficiently spans the mass of spacecraft
  - 90% of total mass captured along each axis
Random Response Analysis

• Random environment load analysis
  – Load analysis using extracted eigenmodes for base-driven random vibration

• Extracted range of modes sufficiently spans the mass of spacecraft
  – 90% of total mass captured along each axis

• Parameters used for analysis:
  – Damping 5% [2]
  – Apply base motion as PSD along all 3-axes

Results

Maximum Von Mises stress in CoreSat

Yield stress Aluminum 6061-T6: 240 MPa
Deliverables

• Assembly procedure
  – Include information regarding how to assemble and put together AAReST mass model
  – Facilitate future assembly work that would have to be performed

• List of recommendations
  – Suggest possible improvements for flight structure, from design perspectives to outsourcing tips

• Numerical simulations
  – Simplified, representative finite element model
  – Modal analysis results
  – Random environment analysis results
Conclusion

• Internal and external structures of AAReST are assembled
• Awaiting parts from IIST and Surrey
• Satisfied PSLV frequency requirements
• Nominal stress levels are below yield stress

Future work

• Optimize the design of interface for vibration table
• Perform vibration and shock test
• Refine and calibrate the model depending on the results
• Perform shock analysis via FEM
Questions?
Mechanisms Team

Mouadh Bouayad  David Watson

Mentor: Christophe Leclerc
Validation and Characterization of Separation Interface

• Role of separation interface:
  - Holds MirrorSat and camera to CoreSat during launch
  - Release MirrorSat for reconfiguration, release camera for boom extension
  - Responsible for power transfer from CoreSat to MirrorSat

• Upon actuation of separating device (Frangibolt), energy is released into MirrorSat
  - Will Frangibolt actuation cause the ejection of a MirrorSat?
  - How can we determine when the Frangibolt has actuated?
Location on AAReST
Interface System

- MirrorSat Interface
- Frangibolt
- Power Transfer Pad
- CoreSat Interface
- Power Transfer Pin

Overview – Testing Design – Results – Conclusion
Deliverables

- Estimate recoil energy, velocity, impulse of separation
  - Derive MirrorSat motion upon Frangibolt actuation

- Record actuation time for Frangibolt
  - Compare at various input voltages
  - Determine power consumption of Frangibolt

- Measure temperature profile
  - Determine when to cut power to Frangibolt

- Determine the power transfer capability of interface
  - Transfer 4.5V @ 2A
Recoil Energy – Testing Design

- A pendulum was chosen to measure the energy released during Frangibolt actuation
  - Energy chosen for ease of direct measurement
  - Minimal friction during motion
  - Pendulum theory used to easily verify results
- Three tests using only the titanium interfaces, powering the Frangibolt with different input voltages:
  - Minimum voltage: 26.8 V
  - Nominal voltage: 29.6 V
  - Maximum voltage: 33.4 V
- Fourth test using a mass dummy to simulate motion of a full MirrorSat after actuation
Recoil Energy – Constructed Test Setup

Overview – Testing Design – Results – Conclusion

Power Supplies, Wheatstone Bridge, Recording laptop

Pendulum Cable

Titanium Interfaces, Frangibolt

Wood and foam shield

Protractor

Frangibolt

Pendulum cable

Acrylic and foam Frangibolt Containment box
Recoil Energy – Dummy Mass Setup

$m = 3.8 \text{ kg}$
Reloading Frangibolt for Actuation

- Prior to each use, Frangibolt required to be reset
- Requires 3,400 $lb_f$ of compressive force
- Used compression fixture to apply parallel force
- Measured length after compression required within $\pm 0.005''$ specified by Manufacturer
Video Results of All Tests

1st test: Nominal voltage
Recoil Energy – Test Results

• Interface tests:
  – Velocity measured thanks to both protractor and slow motion camera
  – For each test, both velocity measurements concur within 10%
  – Estimate recoil energy as $E_k \approx 0.08 \, J$
  – Initial velocity estimated for full MirrorSat: $v_0^{exp} \approx 0.2 \, m/s$

• Dummy mass test:
  – Only slow motion measurement (protractor not reliable)
  – Initial velocity measured lower than expected:
    \[ v_0^{mes} \approx 0.06 \, m/s < 0.2 \, m/s \]
Recoil Energy – Discrepancies

• Large discrepancy between the mass dummy test and the interface tests
  – Energy measured about 10 times lower than expected

• Possible causes:
  – Rotation of the dummy mass
  – Solid friction between the interfaces
  – Testing setup wasn’t perfectly rigid

• The testing setup can be improved:
  – Improve movement of pendulum
  – Frame can be stiffer
Recoil Energy – MirrorSat Simulations

- Simulation of MirrorSat’s movement produced
- Assuming a magnetic force of 1N at 2mm
- If $v_0 > 0.03 \text{ m/s}$, the MirrorSat is ejected
- $v_0^{exp} \approx 0.2 \text{ m/s}$ expected from tests
- $v_0^{mes} \approx 0.06 \text{ m/s}$ was measured in tests

⇒ MirrorSat is ejected under current configuration
• Actuation time is within [20, 50] seconds
• Decreasing and proportional function of input voltage
Temperature Profiles of All Tests

- Profile varies by input voltage
- Inconsistent temperature at actuation
- No change in profile after actuation
Power Transfer Capability

- MirrorSat interface will transfer power as designed
- CoreSat interface shorts over conduction pins
  - Pins are press-fit
  - Causes removal of anodization when inserted
  - Creates a short across the interface
  - Solution: increase hole size and isolate pins

Pin-Pad resistance: \( R = (0.3 \pm 0.1)\Omega \)
Conclusion

• Frangibolt releases $\approx 0.08 \, J$ of energy upon actuation
  - This results in a MirrorSat exit velocity of $v_0 \approx 0.2 \, m/s$

• As designed, the 1 N at 2 mm permanent magnetic force is not strong enough to hold the MirrorSat upon actuation
  - Electromagnets may be used instead (stronger)

• Power transfer is possible once pins are isolated from CoreSat interface

• Temperature profile data is inconclusive for actuation point
Future Work

• A test in a vacuum to measure the RTD data will provide an accurate temperature profile for actuation in space

• An improved testing setup for the Mass Dummy test will provide more accurate results

• A stronger electromagnet needs to be tested for holding MirrorSat in place

• A new method for determining when Frangibolt actuation has occurred is necessary, since RTD data is inconclusive
Questions?

1st test : Nominal voltage
AAReST Attitude Determination and Control System (ADCS)

Avinash Chandra
Mike O’Connell
Hiroyasu Tsukamoto

Mentor: Michael Marshall
Overview

- Responsibilities
  - Develop high-level ADCS operational architecture
    ▪ CubeSpace ADCS → ADCS specific to this mission
  - Plan ADCS acceptance tests
    ▪ Execute ADCS on-board computer (OBC) health check

- ADCS components (excluding coarse sun sensors)
ADCS Operational Architecture

**Goal:** Architect operational modes and commands to ADCS

- We developed:
  1. Mapping from ADCS operational modes to CubeSpace ADCS
  2. Software architecture for ADCS operational mode transitions
  3. Available sensors in each AAReST configuration
  4. Software architecture for AAReST configuration change

### Operational Architecture Diagram

- **Operational Mode N**
  - Control modes
  - Estimation modes
  - Sensors available

- **Transition**

- **Operational Mode N+1**
  - Control modes
  - Estimation modes
  - Sensors available

- **Configuration Change**
Summary of ADCS Operational Modes and Configurations

Operational Modes

- Ground Testing
- Idle
- Detumble
- Survival (Default)
- Configuration Change
- Slew
- Calibration
- Rendezvous and Docking (RDV)
- Sun Pointing (Coarse Pointing)
- Ground Track
- Science (Fine Pointing)

Configurations

- Wide
- Narrow

Overview – Schedule – Work – Project Discussion - Conclusion & Future Work
Summary of CubeSpace ADCS

Control Modes
- None
- Detumbling
- Y-Thomson spin
- Y-Momentum
- XYZ-Wheel
- Sun tracking
- Target tracking

Estimation Modes
- None
- MEMS rate
- Magnetometer rate
- Magnetometer rate with pitch
- Magnetometer and Fine-sun TRIAD
- Full State EKF
- MEMS Gyro EKF

Sensors used
- None
- MEMS rate sensor
- Magnetometer
- Coarse sun sensors
- Fine sun sensor
- Nadir sensor
- Star tracker
1. Mapping from Operational Modes to CubeSpace ADCS

- Operational Mode N
  - Control modes
  - Estimation modes
  - Sensors available

- Transition

- Operational Mode N+1
  - Control modes
  - Estimation modes
  - Sensors available

- Configuration Change
# Mapping from Operational Modes to CubeSpace ADCS

<table>
<thead>
<tr>
<th>Operational mode</th>
<th>Control mode</th>
<th>Estimation mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Idle</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>Detumble</td>
<td>Detumbling control &amp; Y-Thomson spin</td>
<td>MEMS rate &amp; Magnetometer rate</td>
</tr>
<tr>
<td>Slew</td>
<td>XYZ-Wheel control</td>
<td>MEMS Gyro EKF</td>
</tr>
<tr>
<td>Rendezvous and Docking (RDV)</td>
<td>XYZ-Wheel control</td>
<td>MEMS Gyro EKF</td>
</tr>
<tr>
<td>Sun Pointing (Coarse Pointing)</td>
<td>Sun tracking control</td>
<td>Full State EKF</td>
</tr>
<tr>
<td>Science (Fine Pointing)</td>
<td>Target tracking control</td>
<td>MEMS Gyro EKF</td>
</tr>
<tr>
<td>Ground Track</td>
<td>Target tracking control</td>
<td>MEMS Gyro EKF</td>
</tr>
</tbody>
</table>
2. Software Architecture for Operational Mode Transition

- Control modes
- Estimation modes
- Sensors available

- Control modes
- Estimation modes
- Sensors available
### Software Architecture for Transition

#### Mapping from Required Actions to Commands

<table>
<thead>
<tr>
<th>Actions</th>
<th>Command Name</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Check &amp; clear errors</td>
<td>Clear Errors</td>
<td>12</td>
</tr>
<tr>
<td>2. Enable ADCS</td>
<td>ADCS Run Mode</td>
<td>10</td>
</tr>
<tr>
<td>3. Set control gains</td>
<td>Set CONTROL NAME Parameters</td>
<td>38, 39, 40, 54</td>
</tr>
<tr>
<td>4. Set target reference</td>
<td>Tracking Controller Target Reference</td>
<td>55</td>
</tr>
<tr>
<td>5. Set control mode</td>
<td>Set Attitude Control Mode</td>
<td>13</td>
</tr>
<tr>
<td>6. Distribute power</td>
<td>ADCS Power Control</td>
<td>11</td>
</tr>
<tr>
<td>7. Set estimation mode</td>
<td>Set Attitude Estimation Mode</td>
<td>14</td>
</tr>
<tr>
<td>8. Set estimation noise and sensor mask</td>
<td>Set Estimation Parameters 1 &amp; 2</td>
<td>43, 44</td>
</tr>
<tr>
<td>9. Perform control</td>
<td>Trigger ADCS Loop</td>
<td>18</td>
</tr>
</tbody>
</table>
3. Available Sensors in Each AAReST Configuration

- Control modes
- Estimation modes
- Sensors available
# Sensor Obstructions

- Table shows obstructions by component configuration and impact on sensors
  - Available estimation modes depend on available sensors

- **Star tracker must be moved slightly to avoid obstructions**
- Reduced sensor performance – **further analysis required for CSS**
- Sensor obstructed – not needed during corresponding mission phases
- Mask nadir obstructions in software
- Sensor unobstructed

## Table

<table>
<thead>
<tr>
<th>Component</th>
<th>Boom</th>
<th>Antenna</th>
<th>MirrorSat</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Stowed</td>
<td>Partially Deployed</td>
<td>Fully Deployed</td>
</tr>
<tr>
<td>Fine sun sensor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Star tracker</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nadir sensor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-X_{CAD} CSS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>+Z_{CAD} CSS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. Software Architecture for Configuration Change

- Control modes
- Estimation modes
- Sensors available

- Control modes
- Estimation modes
- Sensors available

Operational Mode N → Transition → Operational Mode N+1 → Configuration Change
Architecture for Configuration Change Event

- Designed ADCS operational architecture for configuration change
- Applies to boom deployment and frangibolt breakage

Check satellite state
- Deployment state
- Battery SOC
- Attitude & spin (telecommand (TC) 146 & 147)
- Reaction wheel spin rate (TC 156)

Safety checks before deployment
- Attitude rates below threshold
- SOC > 80%
- Reaction wheel rates below threshold

Turn off attitude control
- Set to uncontrolled (TC 13)
- Set constant reaction wheel speed (TC 17)
- Attitude estimation continues

Deployment
- Deploy component
- Wait for deployment completion

Update system parameters
- S/C inertias (TC 41, 42)
- Control gains (TC 38-40)
- Estimator parameters (TC 43-44)
- Nadir sensor FOV mask (TC 28-32)
- Ignore obstructed sensors in EKF (TC 44)
- Update available control and estimation modes

Reinitialize control system

Return to survival mode
ADCS Configuration Sheet

- **Goal**: Identify ADCS component orientation relative to S/C body frame
- **Required by CubeSpace for initial firmware delivery**
- **Completed configuration sheet and delivered it to CubeSpace**
- **To be updated after final S/C integration and assembly**

![Diagram of component frame, CAD frame, and S/C body frame with axes labels.

Overview – Schedule – Work – Project Discussion - Conclusion & Future Work

06/05/2018

AE 105 Final Presentation

Caltech
Acceptance Test Methodology & Setups
Acceptance Test: Overview

• Needed to verify proper functioning of CubeSpace ADCS components

• Developed test plans and setups

• Test philosophy
  - Health check
  - Polarity test
  - Primarily verifying trends in data
  - Not verifying performance quantitatively
# Summary of Test Setups

<table>
<thead>
<tr>
<th>Component</th>
<th>Test Setup</th>
</tr>
</thead>
<tbody>
<tr>
<td>All components: voltage/current/power</td>
<td>Shunt resistance circuit</td>
</tr>
<tr>
<td>Sun sensor</td>
<td>Halogen lamp- pinhole setup</td>
</tr>
<tr>
<td>Mangetorquer</td>
<td>Polarity: suspend magnetorquer Functional: determine magnetic moment</td>
</tr>
<tr>
<td>Reaction wheel</td>
<td>Polarity: suspend reaction wheel Functional: angular rate data from software</td>
</tr>
<tr>
<td>Nadir sensor</td>
<td>Earth and space movie on laptop screen using Celestia</td>
</tr>
<tr>
<td>Star tracker</td>
<td>Star-field images on laptop screen using Celestia</td>
</tr>
<tr>
<td>Gyroscope</td>
<td>Swivel table</td>
</tr>
<tr>
<td>Magnetometer</td>
<td>Earth magnetic field, mobile app for magnetic field measurement</td>
</tr>
</tbody>
</table>
Sun Sensor

- Setup designed to match sun's:
  - Angular diameter
  - Radiation flux density
  - Radiation spectrum

- Test: compare trend in data
Magnetorquer

**Polarity test**
- Setup designed to reduce counter torque on magnetorquer
- Test: magnetorquer should align with earth's magnetic field

**Functional test**
- Setup designed to calculate magnetic moment, $m$
- Test: compare $m$ with specification sheet

\[ B = \frac{\mu_0}{4\pi r^3} m \]
\[ \| m \| = m = NIA \]

- $B$: magnetic field
- $m$: magnetic moment
- $r$: distance from rod center
- $\mu_0$: absolute permeability of free space
- $I$: current
- $N$: number of wire turns
- $A$: loop area
Reaction Wheel

Polarity test

• Setup designed to reduce counter torque on reaction wheel box
• Test: rotation of reaction wheel box should match desired direction

Functional test

• Obtain angular speed data from software
• Compute torque; compare with specification sheet

Point of suspension above center of gravity of system

Not to scale

Reaction wheel box manufactured for vibration test

A

B

C

\[ \frac{1}{1 + sT} \]

Measurement

Low Pass Filter

Angular Velocity

Smoothed \( \omega \)

Derivative

Resultant torque

\( \tau \)

\( J \)

\( \dot{\omega} \)
Nadir Sensor

- Setup designed to match earth’s angular diameter
- Not addressing earth/space luminance explicitly

Not to scale

Diameter of earth in image: about 15 cm

Test:
- Show earth images/video (Celestia)
- Compare trend in data
Star Tracker

- Setup designed to match:
  - Angular separation of stars
  - Parallel rays
- Not addressing magnitude of stars (algorithm makes no distinction)

- Test:
  - Show starfield images/video (Celestia)
  - Compare trend in data
Overview – Schedule – Work – Project Discussion - Conclusion & Future Work

Gyroscope

- Setup designed to rotate gyro at an angular rate
- Test:
  - Rotate swivel at about 2°/s
  - Compare polarity and value of gyro reading
Magnetometer

- Test:
  - Place magnetometer in earth magnetic field on ground
  - Compare magnetometer readings with *MagnetMeter* app

*MagnetMeter* app on App Store (Apple) to compare magnetometer readings
OBC Health Check

- Confirmed basic OBC functions working upon delivery
- Health check procedure provided by CubeSpace
- All health checks successful
  - Watch dog tests displayed failure due to glitch in interface program, CubeSpace reviewed log files and confirmed test was successful
Conclusion

• Responsibilities
  – Develop high level ADCS operational architecture
  – Plan ADCS component acceptance tests
  – Perform OBC health check

• Completed
  – Mapping between operation modes, sensor obstructions, and available estimation and control modes
  – ADCS command flow for operation mode transition and configuration change
  – Procedure, setup, and parts required for ADCS component acceptance tests
  – Performed OBC health check

• Going forward
  – Perform ADCS component acceptance tests
  – Develop ADCS software around outlined architecture
Questions?
Thermal Overview

Objectives:

• Identify components in danger of exceeding operational and survival temperatures
  – Document and determine critical components
  – Update previous thermal model and run numerical simulations
  – Identify critical components that exceed thermal requirements

• Design passive thermal control system
  – Recommend optical coatings and/or pointing restraints

• Determine impact on mechanical architecture (boom thermal expansion)
1. Spacecraft requirements and critical components
2. Model description
   a. Physical model
   b. Orbits and experimental cases
3. Results
   a. Mirrors
   b. CoreSat
   c. Camera
   d. Boom
4. Conclusions
5. Future Work
**Critical Thermal Requirements**

<table>
<thead>
<tr>
<th>Component</th>
<th>Survival Min. (°C)</th>
<th>Operational Min. (°C)</th>
<th>Operational Max. (°C)</th>
<th>Survival Max. (°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deformable Mirrors</td>
<td>-60</td>
<td>-15</td>
<td>+20</td>
<td>+50</td>
</tr>
<tr>
<td>ADCS hardware</td>
<td>-50*</td>
<td>-10</td>
<td>+60</td>
<td>+100*</td>
</tr>
<tr>
<td>Camera</td>
<td>-100</td>
<td>-20†</td>
<td>+20†</td>
<td>+100</td>
</tr>
</tbody>
</table>

- Critical components were identified from list of spacecraft components based on narrowest operational temperature limits and initial simulations

(*) Estimated survival temperatures
(†) Theoretical operational temperatures (experimental data has suggested operational limits are larger)
# Boom Requirements

<table>
<thead>
<tr>
<th>Error Source</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Boom deflection: translation (mm)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X translation</td>
<td>-0.625</td>
<td>+0.625</td>
</tr>
<tr>
<td>Y translation</td>
<td>-0.625</td>
<td>+0.625</td>
</tr>
<tr>
<td>Z translation (Axial)</td>
<td>-0.127</td>
<td>+0.127</td>
</tr>
<tr>
<td><strong>Boom deflection: rotation (°)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tip</td>
<td>-0.04</td>
<td>+0.04</td>
</tr>
<tr>
<td>Tilt</td>
<td>-0.04</td>
<td>+0.04</td>
</tr>
</tbody>
</table>

(Boom requirements provided by Daniel Türk)
Thermal Desktop Model

- 4 W heat load applied to outside surface of camera
- 30 W applied to internal nodes of Coresat (split into 10 W per stack)
- 6 W applied to outside surfaces of MirrorSats
Mirror Modelling

- Deformable mirrors are made of borosilicate glass
- Mirror is modelled as thin shell cylinder, 0.508mm thick
- Mirror is coated with polished aluminum on the front and black paint on the back
- 1 central node, 6 anterior nodes, and 6 edge nodes
Mirror Modelling

- PCB added below deformable mirror to examine the effects of power dissipation on mirror

- Mirror actuation requires 3 W for 10 minute science period
Thermal Desktop Model

• Model contains a mix of solid prisms and thin shells to model different components
  – Each element has material and optical thermal properties assigned to it
  – Each element is composed of a set of nodes at which radiation calculations are computed
• Model incorporates solar radiation, earth radiation, earth albedo
Spacecraft Hot Case

**Hot Case**: No eclipse period \( (\Omega = 90^\circ) \)

As viewed from sun
Spacecraft Cold Case

**Cold Case:** Eclipse period ($\Omega = 180^\circ$)

As viewed from sun
Deformable Mirrors, Spacecraft Cold Case

- Mirrors are in direct sunlight, experience eclipse
- Displays irregular temperature jumps, but confidence in min. and max. values due to convergence when tilted towards cold case
- >90% of eclipse period within operational range
Deformable Mirrors, Spacecraft Hot Case

- Mirrors receive no direct sunlight
- \(~5\) °C temperature variations due to Earth radiation and albedo reaching mirrors
- >95% of orbit within operational limits, PCB heat during science brings mirrors within limits
Spacecraft Pointing Considerations

Considering the mirror thermal requirements, what elevation angles of AAReST are permitted and where can science be performed in each case?
Pointing Restrictions: Hot case

- For deformable mirrors, PCB below mirrors keeps mirrors operational when pointed away from sun (increases temp. by 10°C)

PCB and solar radiation makes mirrors too hot for science

Incoming solar radiation

- Operational
- Survival
- Damage
Pointing Restrictions: Cold case

- When pointed towards sun (< +/- 30°), science must be performed towards end of eclipse period
- When pointed away from sun (> +/- 30°), science must be performed towards end of sunlight period
CoreSat: ADCS

- 50% margin below hot limit
- Cold case reaches exactly lower limit
- Has heat load applied according to internal power
Camera Coating: Chromic Paint Covers All Cases

- ~25% margin below hot temp, ~50% margin above cold temp
- Camera coated in chromic paint ($\alpha=0.44$, $\varepsilon=0.56$)
- Camera is running in the cold case (4 W load)
Boom Thermal Distortion

● Changes in temperature cause physical expansion or contraction of materials

● Will consider two problems of thermal distortion
  – Axial expansion (or contraction) as temperature changes
  – Translation as a result of temperature differences between sides of the boom (hot case only)
  – Camera tilt as a result of temperature differences between sides of the boom (hot case only)
Boom Distortion Req. Met

- ~25% margin before reaching upper limit, ~65% margin above lower limit
- Difference of ~5 μm between sides of the boom at steady temperature in hot case, causes camera tilt of ~0.02°
## Comparing to Requirements

<table>
<thead>
<tr>
<th>AAReST Component</th>
<th>Operational Restriction</th>
<th>Survival Restriction</th>
<th>Results</th>
<th>Smallest Margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>CoreSat/ADCS System</td>
<td>-10/+60°C</td>
<td>-50/+100°C</td>
<td>-15/+45°C</td>
<td>0%</td>
</tr>
<tr>
<td>Deformable Mirrors</td>
<td>-15/+20°C</td>
<td>-60/+50°C</td>
<td>-15/+45°C</td>
<td>20%*</td>
</tr>
<tr>
<td>Boom</td>
<td>±0.127 mm (axial)</td>
<td>-0.03/+0.09 mm (axial)</td>
<td>25%†</td>
<td></td>
</tr>
<tr>
<td></td>
<td>±0.625 mm (translation)</td>
<td>0.005 mm (translational)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Camera</td>
<td>±20°C</td>
<td>±100°C</td>
<td>-8/+15°C</td>
<td>25%†</td>
</tr>
</tbody>
</table>

* margin to survival temps.
† margin to operational temps.
Conclusion

• Deformable mirror temperatures remain within operational limits long enough to perform science and do not exceed survival temperatures
• ADCS remains within operational limits (no margin in cold case)
• Boom distortion is within requirements
• Camera can be coated to keep it in operational requirements, but might need to run while in cold scenarios
Future Work

• Refine camera model
• Improve internal modelling complexity to simulate internal temperature profile more accurately
• Experiment with azimuthal angles for pointing considerations
• Extend analysis and adapt model for wide configuration
Questions?
Onboard Software

Lucas Benoit-Maréchal
Jean-Sébastien Spratt     Alexander Wen

Mentor: Fabien Royer
Presentation Overview

- Team Overview
- Software Development Process
- Hardware Test Setup Development
- Testing Process
- Conclusion
Project Responsibilities and Scope
Scope and responsibilities

- **OBC responsible for communication between AAReST subsystems**
  - Interface with existing subsystem software through I2C

- **Main focus: two modules**
  - Mirror
  - ADCS

- **Software:**
  - Driver files that implement telecommand and telemetry requests
  - C Language
  - Communication channel I2C

- **Hardware:**
  - EFM32 board to emulate OBC
  - Arduino to emulate modules
Deliverables

• **Code for Mirror + ADCS functions**
  — 75% of functions written (290/390)

• **Documentation on functions’ writing process**
  — Compilable file template
  — Documented pitfalls and solutions

• **Hardware development and test setup**
  — Software configuration for emulators (EFM32 and Arduino boards)
  — Hardware modifications
Module Development
Software architecture

- **Hardware Abstraction Layer (HAL)**
  - Takes care of communication

- **Modules**
  - Black boxes

- **Source file .c**
  - Functions: format and transmit messages

- **Header file .h**
  - Variables and functions declaration

- **EFM32 board**
  - Emulates OBC

- **Drivers**
  - ADCS.c
  - ADCS.h
  - Mirror.c
  - Mirror.h
Anatomy of a telecommand

- Creating Message
- ID Number
- Data Byte
- Writing

- Interface Control Document (ICD) example:

<table>
<thead>
<tr>
<th>ID</th>
<th>Parameters Length (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>47</td>
<td>8</td>
</tr>
</tbody>
</table>

**Table 64: Set SGP4 Orbit Eccentricity Command Format**

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>Parameters</th>
<th>Parameters Length (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>47</td>
<td>Set SGP4 Orbit Eccentricity</td>
<td>Offset (bits) Length (bits) Name Data Type Description</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
<td>64</td>
</tr>
</tbody>
</table>
Work organization

• Multi developer configuration management: GitHub platform
  – Code reviews
  – Clear GUI
  – Homogenized nomenclature
Test Setup Development
Additional work is needed to make driver functions run and execute on the development board.
We Developed Test Setups That Simulate Flight Config

**ADCS Flight Config**

**CubeSpace OBC**

**Mirror Flight Config**

**CubeSpace OBC**

**CubeSpace ADCS**

**I2C - UART Bridge**

**Mirror Box**
We Developed Test Setups That Simulate Flight Config

ADCS Simulator

EFM 32 STK3700

I2C

ADCS Simulator

EFM 32 STK3700

Mirror Simulator

I2C

EFM 32 STK3700

I2C - UART Bridge

Mirror Box
More Software Needed: Development Board Requires Drivers and GPIO Config Files to Setup for I2C
More Software Needed: Development Board Requires Drivers and GPIO Config Files to Setup for I2C
Even with all of the software configured, some hardware modifications needed as well

**Rework Procedure**

**Required Tools:**
- Soldering Iron

**Rework Instructions:**
- Remove the 0 ohm resistor R198 of the LC-Sense Circuit.

**Notes:**
The location of the resistor is indicated below. It is the center, black package component in a column of three components.
We Built our Test Setups Incrementally: Easy to Debug, Leverage previous work at all stages

- Config. STK3700 for I2C
- Interface: I2C Sim
- Dev Board
- I2C Sim
- Interface: I2C-UART Bridge
- Mirror Box
- Bridge
- ADCS Sim
- Mirror Sim
ADCS Simulator: Working Final Setup

1. I2C Read from device

```c
set_led(0,1);
uint16_t value = i2c_read_register(117);

// Set an LED on the Starter Kit if success
if (value == DEVICE_ID) {
    set_led(1,1);
}
```

2. If device read correct, turn on LED

Two lights = working
Mirror Simulator: Working Final Setup

1

```
USR_Mirror_I2C_Start_App();
```

```
USR_Mirror_I2C_Move_Pico2(100000);
```

Move PicoMotor 2

Mirror Sim Setup

I2C-UART Bridge

Eng. Mirror

Dev Board

Motor

Button press commands motor to turn
Testing Process
Test Flow

1. Independent visual code review
2. Telecommand Testing (Write only)

3. Telemetry Request Testing (Write and Read)

4. Integration Testing
2. Telecommand Testing

• Create test inputs for each data type

• Write a list of expected messages using the Interface Control Document (ICD) and test inputs

• Run commands and check obtained message
Compiling Expected Message list

Table 10: Reset Command Format

<table>
<thead>
<tr>
<th>ID</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
<th>Parameters Length</th>
<th>Parameters Offset (bits)</th>
<th>Parameters Length (bits)</th>
<th>Name</th>
<th>Data Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform a reset</td>
<td>1</td>
<td>0</td>
<td>8</td>
<td>Magic number</td>
<td>UINT</td>
<td>Magic number to make sure it is a valid reset command. Should equal 0x5A</td>
</tr>
</tbody>
</table>

Message:

1 90

Repeat for each command (ADCS Example):

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Expected Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 90</td>
</tr>
<tr>
<td>2</td>
<td>2 170 170 170 170 170 170</td>
</tr>
<tr>
<td>3</td>
<td>3 1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Run Commands and Read Message

```c
// Calling functions with test values inputs:
USR_ADCS_I2C_Reset(); //1
USR_ADCS_I2C_CurrentUnixTime(testu32, testu16); //2
USR_ADCS_Cache_Enable_State(testbool); //3
USR_ADCS_Reset_Log_Pointer(); //4
USR_ADCS_I2C_DeployMagnetometerBoom (testu8); //7
USR_ADCS_I2C_AdcsPowerControl(testenum, testenum, testenum, testenum, testenum, testenum); //7
USR_ADCS_I2C_ClearErrors(testbool, testbool); //12
USR_ADCS_I2C_SetAttitudeControlMode(testbool, testbool, testu16); //13
USR_ADCS_I2C_SetAttitudeEstimationMode(testenum); //14
USR_ADCS_I2C_CommandedAttitudeAngles(testu16, testu16, testu16); //15
USR_ADCS_I2C_SetMagnetorquerOutput(testu16, testu16, testu16); //16
USR_ADCS_I2C_SetWheelSpeed(testu16, testu16, testu16); //17
USR_ADCS_I2C_TriggerADCSLoop(); //18
```

- **Dev Board**
- **I2C**
- **ADCS Sim**
- **Run**
- **Print**
Compare to Expected Messages

Developed test excel spreadsheet

<table>
<thead>
<tr>
<th>Command ID</th>
<th>Expected Message</th>
<th>Message</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1 90</td>
<td>1 90</td>
<td>TRUE</td>
</tr>
<tr>
<td>2</td>
<td>2 170 170 170 170 170 170 170</td>
<td>2 170 170 170 170 170 170 170</td>
<td>TRUE</td>
</tr>
<tr>
<td>3</td>
<td>3 1</td>
<td>3 1</td>
<td>TRUE</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>4</td>
<td>TRUE</td>
</tr>
<tr>
<td>7</td>
<td>7 170</td>
<td>7 170</td>
<td>TRUE</td>
</tr>
<tr>
<td>11</td>
<td>11 170 170 128</td>
<td>11 170 170 128</td>
<td>TRUE</td>
</tr>
<tr>
<td>12</td>
<td>12 192</td>
<td>12 192</td>
<td>TRUE</td>
</tr>
<tr>
<td>13</td>
<td>13 2 128 170 170</td>
<td>13 2 128 170 170</td>
<td>TRUE</td>
</tr>
<tr>
<td>14</td>
<td>14 2</td>
<td>14 2</td>
<td>TRUE</td>
</tr>
<tr>
<td>15</td>
<td>15 170 170 170 170 170 170 170</td>
<td>15 170 170 170 170 170 170 170</td>
<td>TRUE</td>
</tr>
<tr>
<td>16</td>
<td>16 170 170 170 170 170 170 170</td>
<td>16 170 170 170 170 170 170 170</td>
<td>TRUE</td>
</tr>
<tr>
<td>17</td>
<td>17 170 170 170 170 170 170 170</td>
<td>17 170 170 170 170 170 170 170</td>
<td>TRUE</td>
</tr>
</tbody>
</table>

Expected

Obtained

True if messages match
3. Telemetry Request Testing

- Set up Arduino to respond with test register
- Set up dev board test to check responses
- Run commands and check stored values
Additional Testing Code

Dev Board

ADCS Sim

Telemetry Request

Response

Run Command:

```c
USR_ADColumns_I2C_GetEstimatedAttitudeAngles(); //146
```

Test Response:

```c
//146
if (TCVal_ADColumns.estimates_oll_angle != testi16)
{
    setLed(1,1); 
    errors = errors + 1;
} 
if (TCVal_ADColumns.estimates_pitch_angle != testi16)
{
    setLed(1,1); 
    errors = errors + 1;
} 
if (TCVal_ADColumns.estimates_yaw_angle != testi16)
{
    setLed(1,1); 
    errors = errors + 1;
}
```

Simulate Command response:

```c
// USR_ADColumns_I2C_GetEstimatedAttitudeAngles
if (register_address == 146)
{
    Wire.write(simulated_registers, 6);
    Serial.print("146 Transmitted");
    Serial.print("\n");
}
```

LED stays off: SUCCESS!
4. Integration Testing

- ADCS: Hardware was received too late
- Mirror Module:
  1. MATLAB Function Testing on Module
  2. Telecommand Testing
Mirror Box Telecommand Testing

Dev Board

I2C - UART Bridge

Mirror Box

Run Command (example):

```
USR_Mirror_I2C_Move_Pico2(100000);
```

I2C to UART Conversion code*

- Observe Mirror Box
- Probe Mirror Directly

*Provided by Thibaud Talon
Work Accomplished

Mar 4, 2018 – May 31, 2018

Contributions to master, excluding merge commits

293 Functions written
Work Accomplished

Functions Written

- **ADCS**
  - Completed: 186
  - Testing Procedure Outlined: 186
  - Tested: 20

- **Mirror**
  - Completed: 107
  - Testing Procedure Outlined: 107
  - Tested: 10
Conclusion

Accomplishments:
• Coded all planned ADCS and Mirror functions
• Developed test setup for the ADCS and Mirror modules
• Performed unit testing on representative functions, and laid out testing procedure for the rest
• Performed integration testing of a few functions on the Mirror module, and laid out the procedure for full integration testing

Future Work:
• Code other modules using documented coding procedure as template
• Finish testing of ADCS and mirror module using the developed test setup and documented testing procedure
• Use testing template to test other modules
QUESTIONS?
Telecom Subsystem

Kai Matsuka, Conor Martin, Edward Chu

Mentors:
Michael Marshall, Maria Sakovsky
AAReST Telecom Overview

Previous Ae105 classes:
• Selected desired frequencies
• Selected transceiver and designed antennas
• Tested antenna deployment mechanisms

Our primary Ae105 tasks:
• Configure and test the AstroDev radio
  ➔ Verify functionality of radio
• Experimentally characterize the antennas
  ➔ Validate antenna design and optimize transmitted power
Telecom Team Objectives

**Antenna Responsibilities (Kai, Edward)**
1. Perform sensitivity analysis via antenna simulation
2. Fabricate antenna assemblies
3. Fabricate EM-representative spacecraft model
4. Characterize antenna performance
   ➔ End goal: Test and validate the antenna design

**Radio Responsibilities (Conor)**
1. Test radio commands
2. Configure the radio
3. Plan power tests
   ➔ End goal: Test and verify functionality of the radio
Antenna Sensitivity Analysis

- Need to build EM-representative AAReST model for antenna experiments
- Need to determine which components to include in model

➔ Vary geometry in simulation and compare the antenna characteristics

Conclusion:
- Camera/boom not required
- Frame thickness not important
- Coresat, MirrorSats, Launch Vehicle Interface (LVI) required
EM-Representative SC Model

Background:

• Ideal monopole antenna assumes infinitely large “ground plane”
• Spacecraft chassis acts as ground plane

Objective: Build EM-representative model for antenna experiments

Approach: Simple but sufficiently representative of model

• Used sensitivity analysis results to design model
• Waterjetted 1/16” thick aluminum plates with foam core
• Used vibration test LVI
Antenna Fabrication

Objectives:
• Develop the antenna fabrication procedure
• Fabricate antenna assemblies for testing

Results:
• Fabricated 6 antennas, including 2 flight-like antennas
• Iterated, refined, and documented fabrication procedure
Antenna Experiment Setup

Objective: Find the antenna length that minimizes $s_{11}$ at the desired frequency

Procedure:
1. Calibrate the Network Analyzer (NA)
2. Measure $s_{11}$ vs. frequency
3. Trim the antenna in ¼” increments
4. Repeat 2 and 3

Experiment Parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAReST Comm Freq:</td>
<td>146 MHz (VHF), 436.5 MHz (UHF)</td>
</tr>
<tr>
<td>Meas. Freq. Range:</td>
<td>50 – 700 MHz</td>
</tr>
<tr>
<td>Predicted Lengths:</td>
<td>18.1 in (VHF), 5.7 in (UHF)</td>
</tr>
<tr>
<td>Varied Length Range:</td>
<td>12.5-24 in (VHF), 6-10 in (UHF)</td>
</tr>
</tbody>
</table>
Antenna Experiment Results

Results:
- Experimentally determined the optimal VHF antenna length
- Observed expected trend for minimum $s_{11}$ vs frequency
- Unexpected $s_{11}$ values for UHF case → future task is troubleshooting
Telecom Team Objectives

Antenna Responsibilities (Kai, Edward)
1. Perform sensitivity analysis via antenna simulation
2. Fabricate antenna assemblies
3. Fabricate EM-representative spacecraft model
4. Characterize antenna performance
   ➔ End goal: Test and validate the antenna design

Radio Responsibilities (Conor)
1. Test radio commands
2. Configure the radio
3. Plan power tests
   ➔ End goal: Test and verify functionality of the radio
Radio Test Plan

1. **Testing and configuration**
   - Create program to send test commands

2. **Power input**
   - Measure voltage and current in receiving and transmitting operating modes

3. **Power output**
   - Measure RF output power

4. **Inrush current**

5. **Functional test with telecom system**
   - Verify radio communication with mock ground station

---

Future Work
Radio Commands for Testing

- Known commands and expected responses
- 15 total commands
  - 5 with no payload
  - 10 different payload sizes

<table>
<thead>
<tr>
<th>Command Code</th>
<th>Command Description</th>
<th>Payload</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x1001</td>
<td>No-Op</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x2001</td>
<td>No-Op Ack</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x1002</td>
<td>Reset</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x2002</td>
<td>Reset Ack</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x1003</td>
<td>Transmit</td>
<td>Bytes</td>
<td>N</td>
</tr>
<tr>
<td>0x2003</td>
<td>Transmit Ack</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x2004</td>
<td>Received Data</td>
<td>Bytes</td>
<td>N</td>
</tr>
<tr>
<td>0x1005</td>
<td>Get Transceiver Configuration</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x2005</td>
<td>Transceiver Configuration</td>
<td>Structure</td>
<td>N</td>
</tr>
<tr>
<td>0x1006</td>
<td>Set Transceiver Configuration</td>
<td>Structure</td>
<td>N</td>
</tr>
<tr>
<td>0x2006</td>
<td>Set Transceiver Acknowledge</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x1007</td>
<td>Telemetry</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x2007</td>
<td>Telemetry Structure</td>
<td>Structure</td>
<td>N</td>
</tr>
<tr>
<td>0x1008</td>
<td>Write Flash</td>
<td>16 Byte MD5</td>
<td>16</td>
</tr>
<tr>
<td>0x2008</td>
<td>Write Flash Ack</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x1009</td>
<td>RF Configure</td>
<td>RF Structure</td>
<td>N</td>
</tr>
<tr>
<td>0x2009</td>
<td>RF Configure Ack</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x1010</td>
<td>Beacon Data</td>
<td>Bytes</td>
<td>N</td>
</tr>
<tr>
<td>0x2010</td>
<td>Beacon Data Ack</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x1011</td>
<td>Beacon Configure</td>
<td>Beacon Structure</td>
<td>N</td>
</tr>
<tr>
<td>0x2011</td>
<td>Beacon Config. Ack</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x1012</td>
<td>Read Firmware Rev</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x2012</td>
<td>Firmware Rev</td>
<td>Bytes</td>
<td>4</td>
</tr>
<tr>
<td>0x1013</td>
<td>DIO Key Write</td>
<td>Bytes</td>
<td>16</td>
</tr>
<tr>
<td>0x2013</td>
<td>DIO Key Write Ack</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x1014</td>
<td>Firmware Update</td>
<td>16 Byte MD5</td>
<td>16</td>
</tr>
<tr>
<td>0x2014</td>
<td>Firmware Update Ack</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x1015</td>
<td>Firmware Packet</td>
<td>Bytes</td>
<td>TBD</td>
</tr>
<tr>
<td>0x2015</td>
<td>Firmware Packet Ack</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>0x1020</td>
<td>Fast Set PA</td>
<td>Byte</td>
<td>1</td>
</tr>
<tr>
<td>0x2020</td>
<td>Fast Set PA Ack</td>
<td>-</td>
<td>0</td>
</tr>
</tbody>
</table>
Radio Commands for Testing

- Known commands and expected responses
- 15 total commands
  - 5 with no payload
  - 10 different payload sizes

---

**Generic message structure**

<table>
<thead>
<tr>
<th>Header (8 Bytes)</th>
<th>Payload (0 to 255 Bytes)</th>
<th>Payload Check Sum A (1 Byte)</th>
<th>Payload Check Sum B (1 Byte)</th>
</tr>
</thead>
</table>

**Figure 1**--Packet structure for the Command and Data Interface (CDI).

<table>
<thead>
<tr>
<th>Sync Characters (2 Bytes)</th>
<th>Command Type (2 Byte)</th>
<th>Payload Size (2 Bytes)</th>
<th>Header Check Sum A (1 Byte)</th>
<th>Header Check Sum B (1 Byte)</th>
</tr>
</thead>
</table>

**Figure 2**--Description of the packet header used in the CDI.
Radio Commands for Testing

- Known commands and expected responses
- 15 total commands
  - 5 with no payload
  - 10 different payload sizes

```
uint_1 interface_baud_rate;   //Radio Interface Baud Rate (9600=0x00)
uint_1 tx_power_amp_level;    //Tx Power Amp level (min = 0x00 max = 0xFF)
uint_1 rx_rf_baud_rate;      //Radio RX RF Baud Rate (9600=0x00)
uint_1 tx_rf_baud_rate;       //Radio TX RF Baud Rate (9600=0x00)
uint_1 rx_modulation;         //0x00 = GFSK)
uint_1 tx_modulation;         //0x00 = GFSK)
uint_4 rx_freq;               //Channel Rx Frequency (ex: 450000000)
uint_4 tx_freq;               //Channel Tx Frequency (ex: 450000000)
unsigned char source[6];      //AX25 Mode Source Call Sign (default NOCALL)
unsigned char destination[6]; //AX25 Mode Destination Call Sign (default CQ)
uint_2 tx_preamble;           //AX25 Mode Tx Preamble Byte Length (0x00 = 20 flags)
uint_2 tx_postamble;           //AX25 Mode Tx Postamble Byte Length (0x00 = 20 flags)
uint_2 function_config;       //Radio Configuration Discrete Behaviors
uint_2 function_config2;      //Radio Configuration Discrete Behaviors #2
```

“Set Configuration” message structure
Radio Interface Program

- Wrote interface program (MATLAB)
  - Enabled serial communication
  - Created user prompt
- Tested 8 important commands
  - Also tested external reset pin

```matlab
%% Define preset list of commands, expected outputs

%% Define all possible commands (no checksums)
commands(1,:)=[72; 101; 16; 2; 0; 0]; %'Reset'
commands(2,:)=[72; 101; 16; 3; 0; 0]; %'Transmit' - need payload
commands(3,:)=[72; 101; 16; 6; 0; 0]; %'Set Configuration' - need payload
commands(4,:)=[72; 101; 16; 32; 0; 0]; %'Fast Set PA' - need payload
commands(5,:)=[72; 101; 16; 5; 0; 0]; %'Get Configuration'
commands(6,:)=[72; 101; 16; 7; 0; 0]; %'Get Telemetry'
commands(7,:)=[72; 101; 16; 16; 0; 0]; %'Set Beacon Data' - need payload
commands(8,:)=[72; 101; 16; 17; 0; 0]; %'Set Beacon Configuration' - need payload
commands(9,:)=[72; 101; 16; 1; 0; 0]; %'No-Op'
commands(10,:)=[72; 101; 16; 8; 0; 0]; %'Write Flash' - need payload
commands(11,:)=[72; 101; 16; 9; 0; 0]; %'RF Configure' - need payload
commands(12,:)=[72; 101; 16; 18; 0; 0]; %'Read firmware revision'
commands(13,:)=[72; 101; 16; 19; 0; 0]; %'DIO Key Write' - need payload
commands(14,:)=[72; 101; 16; 20; 0; 0]; %'Firmware Update' - need payload
commands(15,:)=[72; 101; 16; 21; 0; 0]; %'Write Firmware Packet' - need payload

%% Which command to send (1-15)?
```

Command Window

Radio Test Comm Program:

1. Reset
2. Transmit
3. Set Transceiver Configuration
4. Fast Set Power Amplifier
5. Get Transceiver Configuration
6. Get Telemetry
7. Set Beacon data
8. Set Beacon Configuration
9. No-Op
10. Write Flash
11. RF Configure (RF Structure)
12. Read Firmware revision #
13. DIO Key Write
14. Firmware Update
15. Write Firmware Packet
Power Requirements

Capabilities:
• Radio can draw up to 16 W and transmit up to 7 W

Requirements:
• Power budget: **2 W input** for UHF transceiver allocated
• Link budget: 0.8 W RF output power
  – Requires ~2.5 W RF input power

Representative plots for AstroDev radios
Power Requirements

Capabilities:
- Radio can draw up to 16 W and transmit up to 7 W

Requirements:
- Power budget: **2 W input** for UHF transceiver allocated
- Link budget: 0.8 W RF output power
  - Requires ~2.5 W RF input power

**Expected radio efficiency ~ 30-40% at 2 W power input level**
Radio Power Test Setup

Simultaneous input and output power test

• Measure output power with spectrum analyzer
• Measure input power with shunt resistors: $R_1$ and $R_2$

Results still to come!
Conclusions

Antenna:
• Fabricated antennas and improved fabrication procedure
• Delivered EM-representative spacecraft model
• Performed preliminary antenna tests
  – Partially confirmed the simulation results

Radio:
• Created interface program for serial radio communication
  – Verified functionality of relevant commands
• Evaluated power requirements
• Developed power test plan
  – Purchased components (attenuators, shunt resistors)
Future Work

Antenna:
• Identify why UHF antenna has unexpected $s_{11}$
• Measure $s_{11}$ parameter with different spacecraft configurations (e.g. wide configuration)

Radio:
• Conduct radio power tests
• Complete telecom system functional test
  – Simulate ground station
  – Send and receive signals to radio via RF
Questions?
Avionics

Kira Headrick
Daniel Jamison
Thibaud Talon
Ashish Goel

Lewis Jones
Zoey Flynn
Team Structure

- **Kira - Solar panels**
  - Generate power from solar radiation

- **Lewis, Dan, Zoey - Interface boards**
  - Route power and signals within the CoreSat

Overview
- Solar Panel
- PCB Wiring
- Conclusion & Future Work

+X Face

Central

Y-

Y+
Solar Panel (SP) Overview

- Solar Panel Responsibilities
- Cell Placement
- PCB Wiring
- Conclusion
SP Responsibilities

- Solar Cell Placement
  - Fit as many solar panels as possible
  - Finalize PCB design
- PCB Wiring
  - Create wiring architecture
  - Create PCB in Eagle
SP Work: Overview

- Place solar cells
  - Majority of work
  - Changes made on rolling basis
- Place additional components
  - Temp. sensor
  - Sun sensor
  - Connectors
  - Access holes
  - Mounting screws
- Holes for mounting PCB to coresat
  - Number, Clearance
- Avoid reserved areas
- Match number of cells from last year
  - Revisit power consumption studies
SP Work: +X

- Solar cells: 23
- Challenges
- Placement of additional components prevented horizontal symmetry
- Reserved Area for LEDs
SP Work: +X

Last Year (23 cells)

Current (23 cells)
SP Work: -X

- Solar cells: 24
  - Primary power generating face
  - 1 more cell than last year

Challenges
- Placement of additional components
  - Cannot overlap with solar cells
  - Spacers for antenna
    - Burn wire screws
  - Coarse sun sensors moved
    - Greater field of view for ADCS
SP Work: -X

Last Year (23 cells)

Current (24 cells)
SP Work: +/-Y

- Solar cells: 6
- Challenges
  - Reserved area for LEDs
- Mirror Box connector placed
  - Prevent line twisting

Mirror Box connector

LED Area

horizontally
SP Work: +/- Y

Last Year (6 cells)

Current (6 cells)
SP Work: Mirror Box

- Solar cells: 2
- 4 styles of mirror box PCB: 
- Due to changes in
  - Screw size
  - Number of lines on connector
  - Placement of coarse sun sensor
SP Work: Mirror Box

- 4 styles of Mirror Box PCB

MB on Y Faces

MB on X- Face

MB on X+ Face Left Side

MB on X+ Face Right Side
Cell Wiring Diagram

- Per cell: 2.7V, 0.5A
- Per input channel: 16V, 2A
- Combine +/-X and +/- Y channels for geometric reasons
Eagle: Solar Cell Component

- Eagle: Software to make PCBs
- No physical datasheet for Solar Cell
  - Cell measured to create component
SP Conclusion & Future Work

- Solar Cell PCBs
  - Sufficient number of cells per face
  - Primary power face (-X) has one more cell than last year

- Wiring (Eagle)
  - PCB wiring architected
  - Component libraries created

- Future Work
  - Wire PCBs in Eagle
Interface Boards

- Project Overview and Progress
- Y+ Interface Board
- Development Process
- Central Interface Board
- Flight Preparation Panel
- Y- Interface Board
- Conclusion

CoreSat
Interface Boards Overview

Objectives:
• Design, manufacture, and test four separate interface boards: Y-, Y+, and two Central

Interface Board Responsibilities:
• Routes connections
  – Solar panels power
  – Sensor signals
  – Power, ground
  – I2C, UART, CAN
  – Payload
  – Radio
• Provide Switching
  – Frangibolts
  – Burn wires
  – Inter-satellite power link
  – Power lines
• Provide voltage conversion
• Analog-digital conversion
  – Temperature sensors
Interface Boards Progress

Completed:
- Routing diagrams
- Connector/component selection
- Schematics
- Layouts
- Hardware assembly for some boards

Current Status:
- Y+ Interface Board Assembled
- Central Interface Boards Being Assembled
- Y- Interface Board In Final Stages of Layout
Y+ Interface Board

Y+ Interface Board Responsibilities:

Routing:
- Route all power and signal between central and Y+ electronics stacks of CoreSat
- Interface with Y+, X+, and X- solar panels
- Route all solar panel power to EPS
- Route all Coarse Sun Sensor (CSS) signals to ADCS

Signal Conversion/Switching:
- Convert analog signals from solar panel temperature sensors to digital
- Perform switching for Inter-Satellite Power Link (ISPL), which allows CoreSat to transfer power to MirrorSat

Provide mounting for antenna burnwire
Y+ Interface Board Progress

Completed:
- Schematic
- Design/Layout
- Fabrication
- Assembly

Ready for:
- Verification Testing
Assembled Y+ Interface Board

Y+ IB, Top Face
Assembled Y+ Interface Board

- PC104 Header
- MirrorBox Connector
- MirrorSat - ISPL Connector
- Central IB Connector
- Y- IB Connector
- ISPL Switch
- Analog Digital Converter
- ADCS Connector
- Temperature Sensor

Y+ IB, Top Face
Assembled Y+ Interface Board

- X- Solar Panel Connector
- Notch for Nadir Sensor
- Antenna Burnwire Mounting
- Y+ IB, Bottom Face

- Z- CSS Connector
- EPS Connector
- X+ Solar Panel Connector
- Y+ Solar Panel Connector
Development Process

Routing Diagrams

Schematic/Layout

PCB Fabrication/Assembly
Routing Diagram

Objectives:
- Assign Pins
- Select Connectors and Components

Timeline:
Week 1 - 7

Routing Diagram of Y+ IB

Schematic/Layout

Objectives:
- Place components and connectors
- Route connections

Timeline:
Week 8 - 12

5th Iterations of Y+ Interface Board Layout
PCB Fabrication/Assembly

Objectives:
- Send design for fabrication
- Place orders for components
- Assemble (solder) in clean room

Timeline:  Week 12-14
Central IB Responsibilities:

- Provide interface between the central stack and science instruments, Y+ and Y- interface boards
- Provide power, switching, and data circuitry for the burn wires and frangibolts
- Provide mounting points for fine sun sensor and magnetometer
Central Interface Boards

Completed:
- Schematic
- Design/Layout
- Fabrication

In Progress:
- Assembly

Future steps:
- Verification Testing
Bottom Central Interface Board

- Purpose: power management, burn wire circuitry, camera connector, mounting points for fine sun sensor and magnetometer
Bottom Central Interface Board

- Purpose: power management, burn wire circuitry, camera connector, mounting points for fine sun sensor and magnetometer
Top Central Interface Board

• Purpose: Y+ and Y- interface board connectors, frangibolt circuitry, temperature sensor
Top Central Interface Board

- Purpose: Y+ and Y- interface board connectors, frangibolt circuitry, temperature sensor
Central IB Fabricated PCBs

Bottom Board:

Top Board:
Flight Preparation Panel

Purpose:
- Allows system to be tested thoroughly after assembly and before flight

Design:
- Interfaces with all critical power and signal lines and also remove before flight, kill switch, and charging lines
- Consists of 100 lines that route to the interface boards
  - 16 lines to Y+ IB
  - 34 line to Y- IB
  - 50 lines to Central IB
Y- IB Responsibilities:

- Route connections between the Y- stack
- Interface with the Central and Y+ interface boards
- Provide power and data circuitry for the radio
- Connect FPP to EPS
Y- Interface Board Progress

Completed:
- Schematic
- Multiple iterations of design/layout

In Progress:
- Final layout

Future Steps:
- Final checks
Y- Board Bottom

FPP

KILL Switch

DC-DC Converters

RESET

Charging

ADC

EPS Power

SP Connector
Conclusion & Future Work

• Y+ Board complete
• Central Board assembly in progress
• Y- Board awaiting final checks
• Future Work
  – Finalize Y- layout
  – Complete Central Board assembly
  – Order remaining parts
Thank You!

**Ae105 Class of 2017-2018**

**Instructors:** Daniel P. Scharf, Prof. Sergio Pellegrino

**TAs:** Thibaud Talon, Fabien Royer

- **Mechanisms:** Chistophe Leclerc, Mouadhi Bouayad, David Watson
- **Thermal:** Daniel Türk, Jonah Krop, Henry Steiner
- **Onboard Software:** Fabien Royer, Lucas Benoit-Marechal, Joel Kosmatka, Jean-Sebastien Spratt, Alexander Wen
- **Telecom:** Michael Marshall, Maria Sakovsky, Kai Matsuka, Conor Martin, Edward Chu
- **Structures:** Antonio Pedivellano, Yunsang Chol, Kanthasarny Ubamanyu, Haojie Zhuang
- **ADCS:** Michael Marshall, Avinash Chandra, Mike O’Connell, Hiroyasu Tsukamoto
- **Avionics:** Thibaud Talon, Kira Headrick, Lewis Jones, Daniel Jamison, Zoey Flynn