

SSC21-S1-25

The Fully Automated and Self-Contained Operations Paradigm of the CSIM Mission

Sierra Flynn, Karen Bryant, Emily B. Pilinski, James Paul Mason, Michael Chambliss, Lauren De Moudt, Noah Crisler, Matthew D. Hanley

University of Colorado - Laboratory for Atmospheric and Space Physics (LASP)
1234 Innovation Dr, Boulder, CO 80303, USA; 720-471-1440
sierra.flynn@lasp.colorado.edu

ABSTRACT

The Compact Spectral Irradiance Monitor (CSIM) CubeSat Mission has been collecting solar spectral irradiance (SSI) data for over two years, contributing to 40+ years of multi-mission SSI data collection. CSIM utilizes a fully automated and self-contained operations paradigm developed at the Laboratory for Atmospheric and Space Physics (LASP). LASP efficiently performs the entire operations workflow for CSIM, from planning through data processing, which nominally requires only 15 minutes of staffed operations support per week. Mission operations students at LASP are responsible for the entire planning process. They query for ground station contacts and solar observation times which are input into a suite of software tools to create the onboard stored command table and the weekly uplink plan. An automated ground station script then configures for the upcoming CSIM contacts by querying Space-Track for overflights. Within 2 minutes from the start of a pass, the script commands the UHF or S-Band antenna to point at the spacecraft, brings up the command-and-control software, and performs an initial health-and-safety check upon AOS (acquisition of signal). Automated command scripts then configure the spacecraft and upload the plan using command success logic checks. This ensures that all commands are sent and accepted by the spacecraft in-order, and without overwriting any non-expired scheduling slots. The week's worth of commands is loaded within a few passes, and science collection typically starts soon after. Ground automation will detect major anomalies and notify the flight control team in real-time, allowing the operators to recover the spacecraft on the next contact and prepare a new activity plan for autonomous upload. Additionally, ground automation queries CSIM health and safety data and sends telemetry trends to the operations team for daily, weekly, and monthly health and safety checks. CSIM science data is downlinked during 1 or 2 passes per day via the S-band antenna. This data is processed twice per day via an automated data processing pipeline which requires no regular human intervention. The self-contained and automated nature of the data processing pipeline ensures that LASP scientists can access CSIM data within a few hours of being received on the ground. We discuss how this efficient single-mission, self-contained operations paradigm will be expanded to support multiple missions and external customers in the future.

INTRODUCTION

The CSIM Mission

The Compact Spectral Irradiance Monitor (CSIM) is a 6U CubeSat in a sun-synchronous 575 km orbit that collects solar spectral irradiance (SSI) data. CSIM hosts a two-channel prism spectral radiometer and photodiodes to obtain measurements of the solar spectrum across the 200 – 2800 nm range, encompassing 96% of the solar spectrum¹. CSIM was launched on December 3rd, 2018 and is operated by the Laboratory for Atmospheric and Space Physics (LASP).

The scientific goal of the CSIM mission is to contribute to, and expand upon, the 40+ year SSI solar data record. Data from CSIM is used to understand how solar variability impacts Earth's climate and atmosphere, which can be used to validate or improve climate models¹.

CSIM was also designed to be a flight demonstration model for future solar-focused missions. While previous SIM instruments have been developed for larger-scale missions like the \$100 million SORCE (Solar Radiation and Climate Experiment) satellite² and the TSIS-1 (Total and Spectral Solar Irradiance Sensor) payload, CSIM utilizes a compact, cost-effective, and innovative design to achieve similar scientific goals³: the electrical substitution radiometer (ESR) on CSIM is a novel design that uses vertically aligned carbon nanotube bolometers to calibrate the instrument, helping to produce SSI data consistent with that of TSIS-1 but with a 7.2x reduction in noise across a wider spectral range and with an absolute accuracy within 1%⁴.

LASP's Role in CSIM Operations

The Laboratory for Atmospheric and Space Physics (LASP) at the University of Colorado (CU) in Boulder, CO is a nationally-recognized facility that specializes in atmospheric, solar, planetary, and space physics

research, as well as spacecraft engineering, operations, and data analysis. LASP has contributed to the research, development, and operations behind a multitude of missions exploring every planet in the solar system for over 70 years⁵.

Most programs at LASP support high-class NASA spacecraft, and many operational tools, ground support equipment, and data systems resources can be reused and tailored for each mission. However, CubeSats tend to have smaller budgets and require more compact, cost-efficient development schedules. Therefore, many of the tools available to larger-scale missions at LASP may not be applicable to CubeSat missions.

A team of LASP professionals and students developed the operational and data analysis tools needed for CSIM from scratch. One of the biggest goals was to design autonomous tools to reduce the cost associated with having operators on-console for routine activities or manually processing and delivering science data products. This same cost-saving approach will be used on LASP's future CubeSat missions. However, CSIM's ground automation tools are highly-tailored and were not designed to be used on other CubeSat missions.

LASP has recently been granted funds for developing and flying many more CubeSat missions, including

CUTE, CIRBE, CTIM, SPRITE, and AEPEX. Many of these missions will be flown simultaneously by a small team of CubeSat Operators at LASP. This has driven the need to take the efficient ground automation architecture and philosophy of CSIM, and apply it to new multi-mission tools and processes for these future missions.

In this paper, we discuss the needs and solutions for automating the ground system, planning process, commanding process, and data analysis for the CSIM mission. We consider the pros and cons of these tools and processes, and explore how the same architecture and philosophies can be expanded and improved for future CubeSat missions.

The LASP Ground Station

Most of LASP's CubeSat missions, including CSIM, will have a UHF antenna for commands and real-time housekeeping telemetry, and an S-Band antenna for large science data downlinks. LASP hosts a 4.2-meter S-Band dish and a Yagi UHF antenna suitable for CubeSat operations, in addition to all of the necessary hardware and software for ground station operations, mission planning, mission operations, and post-pass data analysis (Figure 1).

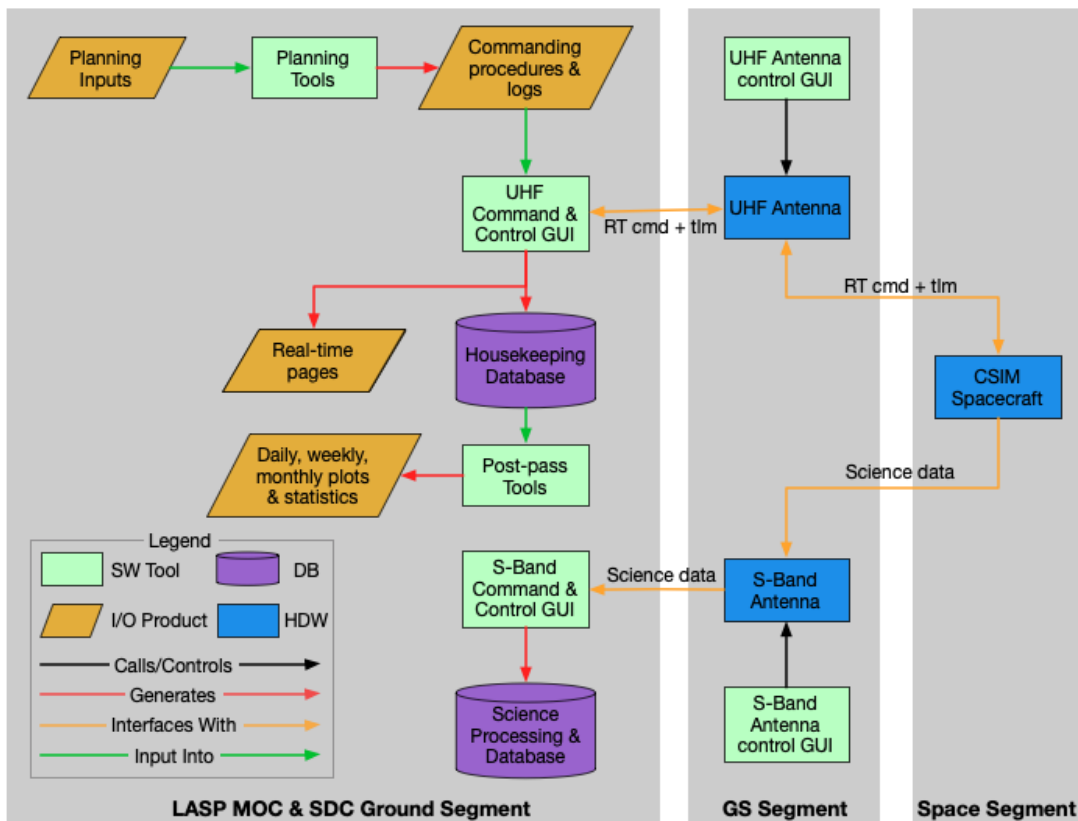


Figure 1: The interfaces between the LASP Mission Operations Center (MOC) and Science Data Center (SDC), the Ground Station (GS) Segment, and the CSIM Spacecraft.

Operations are performed by the LASP Mission Operations Center (MOC) through separate UHF and S-Band Command and Control GUIs called Hydra⁷. Hydra decommutates real-time housekeeping data and displays it in engineering units to the operator, unpacks recorded science data and pushes it to a science processing computer, and allows the operator to send commands. Hydra incorporates elements that makes real-time operations easier for the operator, including a way to run a controllable procedure of commands with telemetry checks, display when telemetry goes out-of-limits or becomes stale, and a real-time paging option for spacecraft anomalies.

In addition to automating CSIM operations, the LASP team also developed tools to automate the ground stations, including scripts to configure and move the ground antennas during overflights, bring up and tear down the command-and-control GUI, send post-pass statistics, and capture and process data downlinks on the ground. Many of these tools work hand-in-and and rely on complex interfaces between each system.

CSIM CONOPS

CSIM nominally operates under a repeating 11-day, orbit-by-orbit science schedule. Normally, Channel A takes solar scans during four orbits per day. Three of these orbits are used to generate a daily solar spectrum covering 200nm-2800nm. The fourth orbit is used to run calibration scans with the ESR. During two days of the 11-day cycle the Channel B didoes are operated. The combination of ESR calibration data on Channel A and the diode data from Channel B allows us to correct for instrument degradation¹. Occasionally, the plan will deviate from this when we want to perform special science activities.

CSIM's orbit ensures that it passes over the LASP ground station about 4-5 times per day for ~10-minute passes. Due to the frequency of data collection, we have reserved about 4 daily passes for UHF operations (real-time housekeeping downlink and command uplink), and 1-2 night passes for S-Band operations (recorded science data downlink only).

The XB1 Bus and Interfaces

CSIM is operated via Blue Canyon Technology's XB1 CubeSat bus. This 1.5U bus and ADCS (attitude control and determination) system provides the flight software, attitude control systems, attitude determination instruments, radios and antennas, solar arrays, and battery required to operate a basic CubeSat⁶. The XB1 is mechanically and electrically integrated to the CSIM instrument, which passes housekeeping and science data to the XB1 bus for future downlink and receives and executes commands from the XB1.

The XB1 provides a volatile Stored Time and Autonomous Command (STAC) Table onboard, which can be used to hold absolutely-timed sequences (ATSs). LASP uses the 400-slot ATS portion of the STAC table as a circular buffer, so the oldest (and presumably executed) slots are overwritten by new, future commands. Typically, a week's worth of ATS commands can fit in this 400-slot table, so the CSIM planning process is done on a 7-day cadence.

PLANNING

The weekly CSIM planning process is nominally the only manual step needed for CSIM operations. Due to the difficulty of accurately predicting satellite orbits long-term, the student planner utilizes AGI's Systems Tool Kit (STK)⁸ to generate CSIM solar periods for the upcoming week. Once STK has been opened, a Python script (csim_stk.py⁹) is run to interface with the STK GUI and automatically generates the report given the input start and stop times (Figure 2). This minimizes the amount of manual effort needed and ensures a consistent report format for later input into a secondary tool.

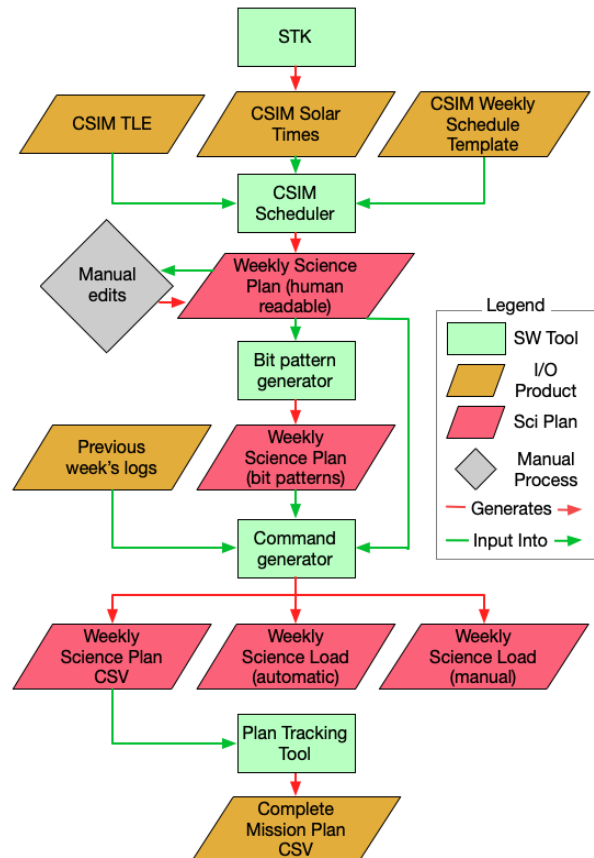


Figure 2: The CSIM Planning Process

The second step involves running the CSIM Scheduler (schedule_csim.py¹⁰) to read in the STK report given the

day of the week. The day of the week is documented in a CSIM workbook that tracks the 11-day science schedule. Oftentimes, a different workbook may be specified for special science activities, such as the twice-annual two-week CSIM Channel B scans. The CSIM Scheduler generates the human-readable weekly science plan (the ATS). This consists of the commands needed to autonomously control the spacecraft and payload for the next week, including science observations and S-Band downlink maneuvers. For each command, a comment is added to specify the absolute time that the command will execute. At this point, the student planner may need to manually modify the ATS for off-nominal or special activities.

The XB1 commands and their associated execution times need to be converted to the appropriate stored time command (STC) so they can be saved into the onboard, 400-slot absolutely-timed command table. The STC command requires the byte definition of the command, which can be obtained using Hydra's bit pattern generator functionality. The student planner executes the ATS in Hydra, which outputs the bit patterns for all of the commands.

In order to extract the command bytes into a ATS load product, the planner then references the last planning cycle to see what the next available slot number is. They run a command generator tool (`stored_cmd_gen.py`¹¹) given the next available slot and the command bit patterns to generate two versions of the ATS load script: a manual load (rarely used), and an automated load. A nicely-formatted human-readable CSV of the week's activities is also generated.

Lastly, the planned CSIM activities need to be documented. This is done via the Plan Tracking Tool (`csim_activities.py`¹²), which verifies that all science scans have been planned and documents them in the complete mission plan CSV.

This entire planning process typically takes less than 15 minutes. At this point, the resulting ATS load products are moved to the CSIM Operations machine, where ground automation can pick them up and begin autonomously loading them to the spacecraft via Hydra.

GROUND AUTOMATION AND COMMANDING

Ground automation is initially manually kicked off via two scripts: `start_gpredict`¹³ and `start_automation`¹⁴. An operator need only start these scripts one time, and they will control the LASP antenna hardware and CSIM automation for each pass.

`Start_gpredict` configures the UHF Antenna Control Software (Figure 3). It brings up `gpredict`¹⁵, a satellite tracking and radio control GUI, and displays the GNU Radio¹⁶ software defined radio (SDR) "waterfall", a plot of CSIM's UHF signal frequency and amplitude over time¹⁷.

The Automation Software is kicked off by `start_automation`, which uses the CSIM TLE from `space-track.org` to determine the pass times over LASP. It then waits until 2 minutes before expected AOS to configure the SDR and interfacing software. It interfaces with `gpredict`, which also has access to CSIM's TLEs, to engage LASP's UHF antenna hardware and point it at CSIM throughout its overflight. `Gpredict` also controls the radio frequency with respect to Doppler shifts, and the SDR radio software, GNU Radio. In summary, `start_gpredict` starts and configures `gpredict` for establishing the interface between the ground station antennas and the spacecraft, so that the ground can receive telemetry and send commands.

`Start_automation` will also execute `auto_pass_manager`¹⁸, a tool that will then open Hydra 1.5 minutes prior to expected AOS. Hydra automatically establishes telemetry and command connections with the GNU Radio. While Hydra is an interactive GUI that an operator can use to send commands, start scripts, and

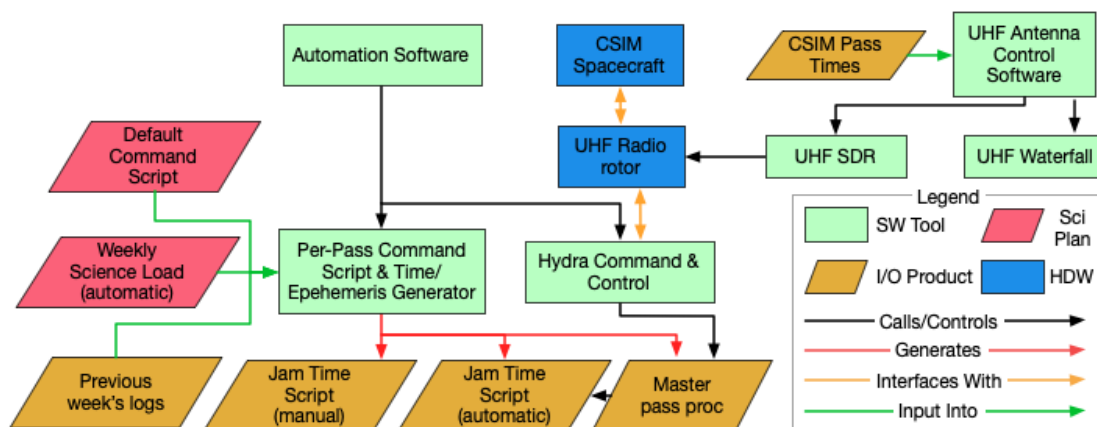


Figure 3: The CSIM Commanding Process

review telemetry, it is primarily controlled autonomously by ground automation and configuration files. Therefore, spacecraft operators are not required to be on-console for the majority of CSIM operations. In the case where an operator needs to be on-console for commanding or monitoring, they can remotely access the Hydra computer and easily halt and restart automation as-needed.

Once the UHF Hydra is up and has established communications with the ground antennas and CSIM with the help of gpredict, the ATS can be loaded to the 400-slot circular STAC table on the XB1 bus. One of the trickiest parts of automated ATS loading is determining what has already been loaded and what is OK to overwrite onboard the spacecraft. Auto_pass_manager calls the per-pass Command Script Generator (auto_pass_gen.pro¹⁹), which generate the final, one-time-use Master Pass Procedure, 1.5 minutes before AOS. This one-time procedure includes beacon commands, time and ephemeris commands, and a subset of ATS load commands. Given the complete ATS load script, the previous pass's event logs, and the previous week's slot schedule, the command generator is able to pick up loading where the last pass left off, while ensuring that no future commands onboard are overwritten. Each command is automatically verified and re-attempted if it does not succeed, and spacecraft health is continuously monitored. If automation hits the end of the cyclical buffer, at which time all 400 onboard commands have not yet executed, it halts loading until commands start executing again and their slots become available for overwriting. Without commands to load, it uses a default command script to increase the downlink beacon rate and set the time and ephemeris. If needed, the back-up manual weekly science load can be run by an operator in place of this automated loading process.

In the case that the spacecraft is in safe mode upon the start of the pass, it doesn't make sense to load the ATS, since a) stored time commands become disabled in this mode; b) the spacecraft time and ephemeris have reset; and c) the fault that sent CSIM to safe mode may be indicative of a hardware or software failure that needs analysis before restarting science observations. In this scenario, ground automation will immediately detect that the spacecraft is in safe mode, and will abort loading the ATS. It will instead utilize the Default Command Script to simply obtain the spacecraft UHF beacon and ensure that state of health (SOH) data is downlinked for an operator to analyze.

Hydra is able to send real-time pages in the form of text messages or emails in the event that anomalous telemetry is detected during a pass. This, in combination with the remote desktop set-up of the Hydra machine, provides

the opportunity for an operator to log-in remotely and analyze telemetry or begin the recovery process within the same 10-minute pass. Strategic long-term engineering data analysis is aided by an automatic cron that will query the CSIM database on a daily, weekly, and monthly cadence, and send out the plots and statistics to the operations team for review.

Abnormal or special operations are typically handled by pre-written and pre-verified Hydra scripts that are run by CSIM operators on-console. These scripts have a combination of automatic and manual command and telemetry verification throughout and are simple for an operator to follow, increasing efficiency and decreasing risk.

At the end of the pass, ground automation will terminate the command and telemetry connections, close Hydra, and generate a shift report of all the commands sent and loaded during the pass and emails it out to the CSIM team. Two minutes before the next CSIM pass, this whole process automatically starts over again.

While gpredict, GNU Radio, and Hydra are also used on the S-Band side, they are running constantly on a separate computer and do not require as much automation or manual interaction. TrackGUI waits for the pass and moves the LASP S-Band antenna as needed, gpredict sets the downlink frequency for the GNU Radio, and Hydra receives the raw recorded science telemetry before passing along to a science data processing computer.

SCIENCE DATA PROCESSING

The science data processing on CSIM is run on a Linux virtual machine (VM) operated at LASP. Twice per day an automated job runs which performs all steps of the data processing pipeline from decoding raw binary data to generating the irradiance and wavelength values for any newly acquired scans. The VM also serves as a data archive with regular backups for all raw and processed CSIM data. The processing pipeline is completely automated and nominally requires no human intervention.

The automated process begins by copying any new binary files to the VM via a shared mounted drive on the LASP internal network. This includes files generated via the UHF and S-band antennas. We also query packets from the SatNOGS amateur ground station network. This is an open-source network of UHF antennas which monitors satellites using amateur bands²⁰. The collected state of health data from SatNOGS has been invaluable for investigating spacecraft anomalies.

Next, all science and housekeeping packets are decoded to engineering units via a custom decoder written in the IDL language. The decoded data are sorted into data structures associated with each packet type and written to files which are stored locally on the VM. These files are used for analysis of the raw telemetry and are the inputs to the science processing pipeline.

Lastly the science processing software uses the instrument model, various ground calibration values, and on-orbit telemetry from the decoded packet files to generate irradiance and wavelength values for each science scan. The processed data are separated into data structures for the ESR and diode scans. Each entry in the data structures contains a timestamp, irradiance and wavelength value along with other related meta data related to the measurement. Files are generated containing these data structures and saved to the VM. These post-processed science files are available for CSIM team members on the LASP local network to access.

CURRENT CHALLENGES & FUTURE PROSPECTS

In addition to up-scaling the LASP ground system architecture and hardware to allow for multi-mission use and sufficient uptimes, there are a number of other ideas that can improve the ground system automation for future missions.

Ground Station Automation Updates

While the upcoming CubeSat missions will be operated by the LASP Mission Operations team and primarily utilize the LASP ground station, future missions may involve external customers and heavier use of the backup ground stations at Parker, CO and Fairbanks, AK. This drives the need to establish the LASP SmallSat Network (LSN), a system integrating the three ground systems and providing interfaces for external mission operations centers (MOCs) or science data centers (SDCs) (Figure 4). The ultimate goal is to have a gateway for external MOCs to control their spacecraft, as well as a user portal for the MOC or SDC to retrieve mission or ground station data and schedule ground station contacts.

Currently, there is no established ground system team at LASP. CSIM operators are usually the ones to first detect issues with the ground stations, and they must contact a ground system engineer to resolve the problem. With more missions being added to the ground station schedule, a more efficient means to monitor and control the ground stations must be established. An automated system can be set up to monitor the ground system, including the antenna hardware statuses and pass statistics, and send real-time pages and weekly reports to a future LSN Team. This will increase ground station uptime and help the LSN keep track of the ground station health and performance. Additionally, a tool will need to be developed to allow ground station scheduling and detect and fix any scheduling conflicts between spacecraft.

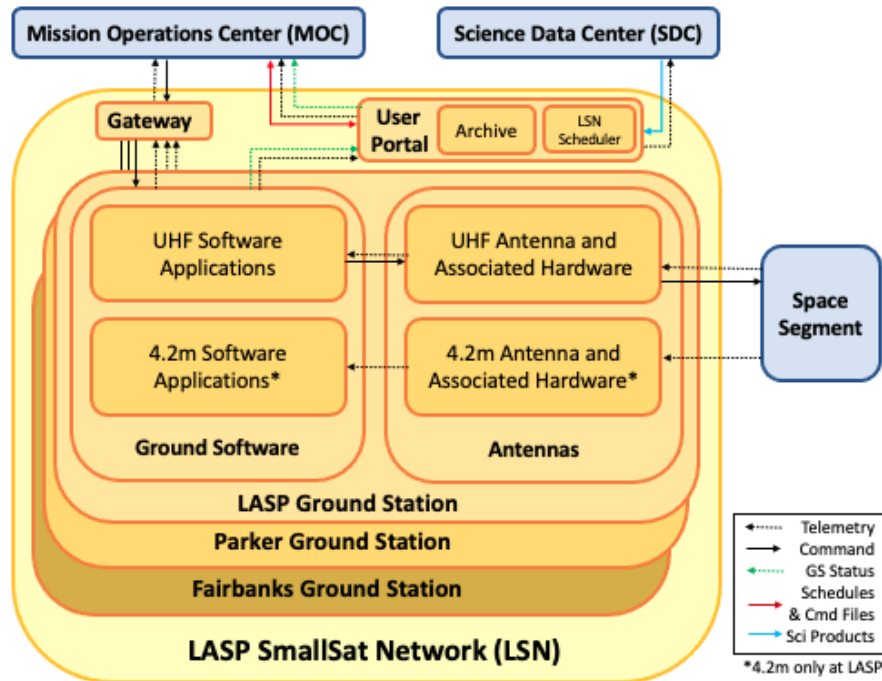


Figure 4: The Future LASP SmallSat Network

Planning Automation Updates

While the full planning process typically only takes about 15 minutes, there are many confusing interfaces involved, utilizing at least 5 different tools, 6 different inputs or configuration files, and 10 different output files per week. This can lead to planning errors and is neither scalable nor flexible for use on future CubeSat missions.

The LASP CubeSat Operations Team is developing a more simplified and streamlined planning process that can be applied to multiple CubeSat missions. While the same 5 tools will be used, they will be updated to make them more automated and produce fewer temporary output files. For example, the final commanding script need not be selected between 3 different scripts at runtime; instead, simple logic can be used in a single script to tell automation which of the three sets of commands to use. Additionally, the tools are being written in such a way that they can be easily tailored for different missions.

Another tool has been developed to read the as-run command logs and keep track of the onboard ATS STAC table on the ground, a feature which is not implemented at the CSIM MOC. This new tool allows a snapshot of the STAC table to be queried quickly in real-time by tools generating the next set of commands to load. This will remove the need for the planner to manually check which slots to load to, and will allow the operations team to view the STAC table at any given time in history.

Commanding Automation Updates

The OASIS-CC (Operations and Science Instrument Support – Command and Control) software is an alternative to Hydra²¹. In addition to having more support and enhanced GUI elements compared to Hydra, OASIS-CC is used across all other missions at LASP and is well-known by all operations personnel. While Hydra will still be used as A) a shim between the ground station antennas and OASIS-CC for stripping/adding telemetry/command headers, B) an interface for sorting and passing along different telemetry files, and C) a flight software development tool, OASIS-CC will take over as the primary command and control GUI for both automated and manual operational use. The Mission OASIS Environment (MOE) toolset will streamline the pre-launch development of OASIS-CC on a per-mission basis and allow for a fast mission set-up²². Although OASIS-CC has its benefits, the addition of another interface complicates the operations and automation process and will require additional time for risk-mitigation and ground testing.

Pre-launch tools have also been developed to automatically generate individual command scripts that will log whether a command was successful or rejected,

verify a telemetry response if applicable, and wait for operator intervention if necessary. The only input needed for this tool to generate scripts for all commands is the pre-existing OASIS-CC command and telemetry database. By splitting each command into its own automated script that performs command verification, commanding efficiency is increased and risk lowered as even a manual operations process will benefit from the added automation.

Operational Updates

CSIM experienced an SD card anomaly in 2019 that limits its data-taking capabilities. Because of this anomaly, CSIM only turns on the SD card when it is collecting science data. Thus, the onboard storage does not fill up as quickly as originally designed. This makes data re-dumps much easier for an operator to add into the ATS, because there is very low risk of overwriting old data onboard or missing newly collected data. The operator simply sends a command to downlink the last X minutes of data, with plenty of buffer. During nominal operations our downlink capability is several times larger than the amount of data we acquire. This ensures that CSIM can play back data several times for any given time range without much extra work for the MOC. The repeated playbacks allow us to capture packets which may have been thrown out due to transmission bit errors. Future CubeSat missions, however, will have different data collection rates and science requirements. As the data collection rate approaches the S-Band data downlink rate, the risk of overwriting lost data increases, and the MOC must have quick and efficient ways to downlink the missed data without impacting the new data. The LASP Operations Software group is developing a multi-mission tool to automatically detect data gaps and develop the necessary commands to downlink the lost data in as short an amount of time as possible.

The CSIM XB1 experiences about 50 resets per year due to suspected solar energetic particle hits. When the XB1 resets, it loses the volatile ATS and reconfigures for safe mode. Hydra automatically detects this and notifies an operator, who must then run a script (typically taking 2 separate passes) to recover the spacecraft. A CSIM planner then has to determine where the reset occurred in the science schedule and replan the ATS. Once CSIM exits sun safe mode, Hydra will automatically start loading the new ATS. These resets may cause up to a 3-day loss in science data collection. This is not ideal, but CSIM is primarily a technology demonstration mission and does not have any set requirements for acquiring a given quantity of science data per month. However, future CubeSat missions will have stricter science requirements, which will drive the need for a quicker and

more automated reset-recovery and ATS-replan and reload process.

Engineers at LASP are developing the LASP Bus, which will run the multi-mission LASP Common Code²³ and will be used for future CubeSat missions such as CTIM and AEPEX. By flying a bus built in-house and of the same software suite running on the payloads, many of our operational processes and automation tools can be reused and streamlined for future missions.

Science Data Processing Updates

The CSIM data processing pipeline was developed specifically and rapidly for this mission and was not designed to be modular enough to use for different missions. While the specific science data processing architecture and software used by CSIM will not be used by future CubeSat missions, there are many lessons that have been learned from this mission.

With the numerous upcoming CubeSat missions, repeating a similar one-off approach would not be possible given budget and time constraints. The LASP data systems team is currently developing data processing systems specifically for CubeSats which will be more modular and easy to setup. The general design will involve several key features:

- A standard method of pushing or pulling the raw binary files for a specific mission to a data processing VM.
- A standardized CCSDS packet decoder which will use a specific packet definition format. The decoder will output the decoded data into one of several standard file formats (e.g., netCDF, HDF5, or FITS) or write the data to a database.
- A standard process for the science team to deploy mission-specific science processing code to a VM where it can easily access decoded telemetry files or query decoded data from a database.

CONCLUSION

LASP is able to perform all the needs of the CSIM mission, including ground station control, planning, uplink and downlink, and data processing and storage, using an interconnected suite of automated tools normally requiring only 15 minutes of manual intervention per week. This has made CSIM a highly successful mission for over two years despite its small budget and mission life expectancy, paving the way for future CubeSat missions.

Although the current LASP CubeSat architecture works solely for CSIM, the tools and philosophies can be applied to LASP's upcoming fleet of CubeSats. By

improving and simplifying the planning process, developing multi-mission tools, creating a highly-customizable and reusable command and control GUI, and developing a LASP SmallSat Network, future internal or external CubeSat missions utilizing the LASP ground stations, MOC, or SDC can expect efficient pre-launch development and a streamlined, flexible, and autonomous mission process, all the way from planning to data processing.

ACKNOWLEDGEMENTS

We would like to thank the LASP MinXSS Operations Team for their early work establishing a LASP ground station. We would also like to thank the entire CSIM Operations Team, including the flight controllers, engineers, and University of Colorado students, who all worked tirelessly writing planning and automation code and performing spacecraft operations. Thank you to everyone involved in making the CSIM mission a scientific and engineering success.

REFERENCES

1. Richard, E., Harber, D., Drake, G., Rutkowski, J., Castleman, Z., Smith, M., ... Lehman, J, "Compact spectral irradiance monitor flight demonstration mission", SPIE Proceedings, vol. 11131, August 2019. 10.1117/12.2531268.
2. "CU-Boulder Returns Nearly \$3 Million In Cost Savings to NASA for Design and Operation of Satellite", Laboratory for Atmospheric and Space Physics, June 2008. lasp.colorado.edu/home/2008/06/17/cu-boulder-returns-3-million-to-nasa-for-satellite/.
3. "Small Satellite Missions: Designed, Built, Tested, & Operated by LASP", Laboratory for Atmospheric and Space Physics, July 2017. https://lasp.colorado.edu/home/wp-content/uploads/2017/07/Small-Satellite-Missions_2017.pdf.
4. Richard, E., Harber, D., Drake, G., Castleman, Z., Rutkowski, J., Fowle, M., ... Pilewskie, P., "The Compact Spectral Irradiance Monitor Flight Demonstration", Earth Science Technology Forum 2019, Mountain View, CA, June 2019.
5. "About LASP", Laboratory for Atmospheric and Space Physics, 2021. <https://lasp.colorado.edu/home/about/>.
6. Stafford, G., "XB1: Enabling a New Realm of CubeSat Science", Blue Canyon Technologies, Boulder, CO, 2012.
7. Cervelli, B., "Hydra User's Guide", Laboratory for Atmospheric and Space Physics, 2020.

8. Analytical Graphics, Inc., Systems Tool Skit (STK), version 11, 2016. [GUI]. <https://www.agi.com/products/stk>.
9. Chakraborty, I., `csim_stk.py`, 2020. [Source code]. Laboratory for Atmospheric and Space Physics.
10. Hanley, M., and Beerman, G., `schedule_csim.py`, 2020. [Source code]. Laboratory for Atmospheric and Space Physics.
11. Beerman, G., and Krumvieda, R., `stored_cmd_gen.py`, 2020. [Source code]. Laboratory for Atmospheric and Space Physics.
12. Beerman, G., and Krumvieda, R., `csim_activities.py`, 2020. [Source code]. Laboratory for Atmospheric and Space Physics.
13. Mason, J. P., `start_gpredict`, 2019. [Source code]. Laboratory for Atmospheric and Space Physics.
14. Mason, J. P., `start_automation`, 2019. [Source code]. Laboratory for Atmospheric and Space Physics.
15. “About Gpredict”, October 2020. <http://gpredict.oz9aec.net/index.php>.
16. GNU Radio project, “About GNU Radio”, 2021. <https://www.gnuradio.org/about/>.
17. Mason, J. P., “CubeSat Ground Station Software Introduction”, Laboratory for Atmospheric and Space Physics.
18. Mason, J. P., `auto_pass_manager`, version 3.0.0, 2019. [Source code]. Laboratory for Atmospheric and Space Physics.
19. Bryant, K., `auto_pass_gen.pro`, 2020. [Source code]. Laboratory for Atmospheric and Space Physics.
20. “About SatNOGS”, Libre Space Foundation. <https://satnogs.org/about/>.
21. OASIS-CC, 2021. [Source code]. Laboratory for Atmospheric and Space Physics.
22. Forsyth, B., Mission OASIS Environment (MOE), 2021. [Source code]. Laboratory for Atmospheric and Space Physics.
23. LASP Common Code, 2021. [Source code]. Laboratory for Atmospheric and Space Physics.